Subjective-Driven Complexity Control Approach for HEVC

Xin Deng, *Student Member, IEEE*, Mai Xu, *Member, IEEE*, Lai Jiang, Xiaoyan Sun, *Senior Member, IEEE*, and Zulin Wang, *Member, IEEE*

***Abstract*— The latest High Efficiency Video Coding (HEVC) standard significantly increases the encoding complexity for improving its coding efficiency, compared with the preceding H.264/Advanced Video Coding (AVC) standard. In this paper, we present a novel subjective-driven complexity control (SCC) approach to reduce and control the encoding complexity of HEVC. Through reasonably adjusting the maximum depth of each largest coding unit (LCU), the encoding complexity can be reduced to a target level with minimal visual distortion. Specifically, the maximum depths of different LCUs can be varied through solving the proposed optimization formulation of complexity control, based on two explored relationships: 1) the relationship between the maximum depth and encoding complexity and 2) the relationship between the maximum depth and visual distortion. Besides, the subjective visual quality is favored with a novel subjective-driven constraint imposed in the formulation, on the basis of a visual attention model. Finally, the experimental results show that our approach can achieve a wide range of encoding complexity control (as low as 20%) for HEVC, with the smallest complexity bias being 0.2%. Meanwhile, our SCC approach outperforms other two state-of-the-art complexity control approaches, in terms of both control accuracy and visual quality.**

***Index Terms*— Complexity control, High Efficiency Video Coding (HEVC), maximum depth.**

I. Introduction

RECENTLY, high-resolution videos and large-sized screens have been flooding into the daily life of humans, bringing about perfect visual enjoyment but at the same time huge challenge on communication bandwidth. It is labored for the once-booming H.264/Advanced Video Coding (AVC) to complete this challenge with its available coding efficiency, thus motivating the birth of a new video coding standard called High Efficiency Video Coding (HEVC). The draft of HEVC was issued in January 2013, and many works have been done to constantly improve its coding efficiency.

Compared with H.264/AVC, HEVC can save bitrates by about 50% with comparable video quality [1]. This eminent coding efficiency mostly benefits from the new coding tree unit (CTU) block partitioning structure, diverse intra-/inter-prediction modes, and other cutting-edge techniques [2], [3]. Along with the remarkable performance of coding efficiency, the encoding complexity of HEVC increases dramatically, ranging from 9% to 502% higher than H.264/AVC [4]. However, most of the multimedia-ready devices, such as portable computers, pads, and smartphones, do not have the ability to sustain such massive complexity due to their limited computational powers. Therefore, the development and research on video coding should not only fasten on enhancing the coding efficiency but also consider the computational complexity.

Complexity control is rather important for HEVC. On the one hand, with the continuous demand for videos at higher resolutions, the complexity of video coding tremendously increases from one standard to the next, with HEVC climbing the highest so far. On the other hand, the amount of multimedia devices has been growing explosively from 2007 onward (when the first generation of iPhone appeared). More and more portable devices featured by encoding and decoding videos are favored by consumers, such as smartphones, pads, carPCs, and other mobile devices. However, their computational powers vary from one device to another. Thus, for better development and implementation of HEVC on various platforms with disparate computing capability, it is necessary to develop an efficient complexity control approach for HEVC.

For the past decades, extensive work has been done to control the encoding complexity for different video coding standards [5]–[7]. More specifically, for H.263, Ismaeil *et al.* [5] evaluated the complexity of four most time-consuming components in encoding process, including the motion estimation (ME), discrete cosine transform, quantization, and mode selection. Each of the four components can be configured to reduce the encoding complexity. By globally assembling the four components with optimal configurations, complexity control can be achieved. Afterward, dealing with the macroblock in H.264/AVC, Kannangara *et al.* [6] proposed an early skip macroblock mode prediction algorithm on the basis of a Bayesian framework. Through reasonably modifying the Bayesian maximum-likelihood criterion that decides the threshold for early skip, the actual complexity of [6] can be reduced to a target complexity. For HEVC, the complexity

Manuscript received November 2, 2014; revised April 17, 2015; accepted July 8, 2015. Date of publication August 27, 2015; date of current version January 6, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61573037, 61202139 and 61471022. This paper was recommended by Associate Editor J.-R. Ohm. *(Corresponding author: Mai Xu.)*

X. Deng, M. Xu, and L. Jiang are with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China (e-mail: cindydeng1991@gmail.com; maixu@buaa.edu.cn; jianglai.china@aliyun.com).

X. Sun is with Microsoft Research Asia, Beijing 100080, China (e-mail: xysun@microsoft.com).

Z. Wang is with the School of Electronic and Information Engineering, Beihang University, Beijing 100191, China, and also with the Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China (e-mail: wzulin@buaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2015.2474075

control mechanism may be related to the new CTU partitioning scheme. For example, Corrêa *et al.* [7] explored the relationship between the coding unit (CU) depth and coding complexity, and then they proposed to constrain the maximum largest CU (LCU) depths of certain frames the same as those of the previous frames, rather than exhaustive rate-distortion optimization (RDO) search. By means of adjusting the number of those constrained frames, the encoding complexity can match the target complexity. However, the above methods do not take visual distortion optimization into consideration during the process of complexity control. Besides, with the deeper study of the human visual system (HVS), it has been found that human attention does not focus on the whole picture when watching a video, but merely a small region around fixation points [8]. In the research fields of rate control and video quality assessment, the subjective quality of reconstructed videos has been considered as an important criterion, in addition to the traditional objective rate-distortion (RD) performance. However, to the best of our knowledge, for HEVC, there is no complexity control approach that considers the subjective factors for ensuring the subjective quality of compressed videos.

In this paper, we propose a subjective-driven complexity control (SCC) approach, based on the visual attention model developed in [9] and [10], to effectively control the encoding complexity of HEVC. The focus of our SCC approach is to vary the maximum depth of each LCU, with the constraint on a given target complexity and minimal objective distortion. Beyond, different from the previous complexity control approaches, the subjective visual quality can be favored in our SCC approach, since the LCUs with larger weights have priority on less distortion, at the same optimal objective quality.

The main contributions of our SCC approach in this paper are presented as follows.

1) We investigate the influence of the maximum depth of LCU on encoding complexity and visual distortion. In this paper, we explore the relationship between the maximum depth of LCU and the overall encoding complexity. Besides, generally speaking, smaller maximum depth may result in larger visual distortion for each LCU. In this paper, we also build the relationship between the maximum depth of LCU and visual distortion.
2) We model the encoding complexity control problem of HEVC by proposing an optimization formulation with a constraint on subjective quality. Specifically, the visual distortion caused by reducing maximum depths of LCUs can be minimized, while decreasing encoding complexity to a target. Meanwhile, subjective quality can be favored by integrating the visual attention model. Liu *et al.* [11] employed a similar algorithm to allocate more complexity resources to salient regions. However, it is used to decrease the decoding complexity of H.264/AVC, but not to control encoding complexity.
3) We propose a solution to our optimization formulation. Our solution yields various maximum depths for LCUs, to achieve target encoding complexity with minimal visual distortion. Corrêa *et al.* [7] also utilized the LCU partition to control the encoding complexity. Actually, the complexity is controlled by predicting maximum LCU depths, based on the similarity of LCU partitions between neighboring frames. This frame-level control may result in large fluctuations of visual distortion and encoding complexity. Our approach controls the encoding complexity at LCU level, which is able to avoid such a disadvantage.

This paper is organized as follows. In Section II, we briefly review the related work. Section III introduces the visual attention model used in our approach. In Section IV, the details about the proposed SCC approach are discussed. Then, the experimental results are shown in Section V to verify the effectiveness of our approach. Finally, Section VI concludes this paper.

II. Previous Work on Complexity Control

Almost every new video coding standard was developed, followed by numerous works on reducing or controlling its encoding complexity, since the encoding complexity of a new standard is usually multiplied compared with its previous generation. For the past few years, a plenty of work has been delivered, with the intention to reduce encoding complexity or achieve complexity control. In the following, we briefly review the relevant work on this direction.

A. Encoding Complexity Reduction Work

Effective complexity reduction is the premise of complexity control. Many studies have been devoted to complexity reduction for both H.264/AVC and HEVC.

For H.264/AVC, most of the encoding complexity reduction studies focused on the ME [12]–[15] and mode decision (MD) [16]–[20] processes, which are the two most time-consuming functions in H.264/AVC. Xu and He [12], Mak *et al.* [13], and Chen *et al.* [14] developed various ME early termination schemes to reduce the complexity of ME process. Li *et al.* [16] and Kannangara *et al.* [17] proposed several early termination methods for MD complexity reduction, by predicting whether a macroblock is of skip mode or not. Different from the early termination-related methods, Huang *et al.* [15] proposed a context-based ME complexity reduction approach. In this approach, the complexity can be reduced by adaptively decreasing the number of searched reference frames. In [18]–[20], various approaches were developed to shrink the optional inter/intra modes in the RDO process, thus decreasing MD complexity. Wang *et al.* [21] employed a hybrid approach by jointly optimizing MD and ME processes to save the encoding time of H.264/AVC.

For HEVC complexity reduction, extensive studies pay attention to the new block partitioning scheme, which leads to huge encoding complexity. Among these studies, [22]–[24] are devoted to finding ways of reducing the encoding complexity on exhaustively searching for optimal CU sizes in the block partitioning process. Specifically, Leng *et al.* [22] proposed an early CU depth prediction approach at frame level. The basic

idea of this approach is to skip some CU depths that are rarely used in the previous frames, thus simplifying the RDO search process to save the encoding complexity. Corrêa *et al.* [23] developed similar approaches at CU level, with the central idea to narrow the current CU depth search range, by virtue of the depth information of adjacent CUs. In addition, some early prediction unit (PU) and transform unit (TU) size decision methods were proposed in [25]–[27] to speed up the PU and TU size selection process. Specifically, Yoo and Suh [26] checked the code block flag and RD cost of the current PU to terminate the prediction process of the next PU for complexity reduction. Corrêa *et al.* [28] used data mining tool to early terminate the RDO process for determining the CU, PU, and TU partitions. Except for CU, PU, and TU size decisions, there are still other components in HEVC affecting the encoding complexity, such as in-loop filtering, and multidirectional intra predictions. From these aspects, [29]–[31] provide several methods to reduce the encoding complexity of HEVC.

B. Encoding Complexity Control Work

Benefitting from the above encoding complexity reduction methods, it is feasible to achieve complexity control for video coding. There exist lots of complexity control approaches, which can be classified into the following three main categories.

The first complexity control philosophy stems from the prevailing complexity allocation thought. Given a target complexity, through rational allocation of complexity resources, the actual running complexity can approach the target, while keeping RD performance well. The approaches in [32] and [33] were proposed for H.264/AVC to allocate the encoding complexity in terms of the distortion (e.g., sum of absolute difference) of each macroblock, in which higher distortion corresponds to more complexity resources. A rate-control-like procedure was developed in [34] for complexity control of H.264/AVC. It employed a novel Lagrange complexity-RD cost model to make a tradeoff between the video quality and complexity cost. Unlike the aforementioned complexity allocation approaches, our SCC approach distributes the complexity resources according to the pixel-wise weights for visual saliency. In this way, the subjective visual quality can be favored.

The heuristic of the second complexity control category is exploiting several encoding parameters to make complexity configurable. Typically, Su *et al.* [35] proposed to manage the complexity of H.264/AVC through adjusting parameters to control ME and MD processes, based on complexity configurable ME and complexity configurable MD algorithms. Most recently, Zhao *et al.* [36] have proposed a flexible mode selection approach for HEVC using a global complexity control factor. Through a hierarchical complexity allocation scheme, the overall RD performance can be maximized. However, it is hard to reach a specific target complexity, as only a few parameters can be configured. In this paper, we propose a complexity control approach, in which the complexity is controlled through reasonably varying the maximum depth of each LCU. Due to the large number of LCUs in each frame, the freedom of degree for complexity control is large such that the encoding complexity can be reduced to a specific target.

The last kind of approach leverages various early termination algorithms to control complexity. Corrêa *et al.* [7] achieved HEVC complexity control by means of predicting the maximum depths of LCUs. In this approach, frames are divided into two categories: 1) unconstrained frames (Fu) and 2) constrained frames (Fc). The maximum depths of LCUs in Fc are early determined, the same as the maximum depths in its previous Fu frame, thus decreasing the RDO process complexity to a specific target. Ren *et al.* [37] proposed to make use of spatial and temporal information to early terminate the MD process for H.264/AVC complexity control. Through setting different early terminate thresholds, the encoding complexity can be reduced to different levels. In this paper, by adjusting the maximum depths of LCUs, we can early terminate the RDO process for selecting the optimal CU sizes of certain LCUs. In this way, the encoding complexity can be saved and meanwhile controlled. The advantage of our approach is that the distortion optimization is considered when decreasing the encoding complexity.

Actually, there are few complexity control approaches proposed for HEVC, since this new standard was just launched not so long ago. Furthermore, to the best of our knowledge, the existing approaches do not take into account subjective quality when controlling the encoding complexity for HEVC. In this paper, we propose an SCC approach to precisely control the encoding complexity of HEVC. By utilizing the bottom-up/top-down visual attention model, our approach can not only control encoding complexity with minimal visual distortion but also preserve subjective quality well.

III. Visual Attention Model

This section describes the visual attention model, as the foundation of the proposed SCC approach. According to the HVS, there exists much perceptual redundancy that can be further exploited to improve coding efficiency without significant perceived quality degradation. For instance, when a person looks at a video, he/she may not pay attention to the whole scene. In other words, a small region around a point of fixation is concerned most [38], while the peripheral region is captured at low resolution. In light of this phenomenon, the computational complexity can be saved in our SCC approach via decreasing the visual quality in the peripheral region with high priority.

From now on, we mainly focus on the visual attention model, which predicts where human looks at a video, as the preliminary of the proposed SCC approach. To predict human visual attention, both bottom-up and top-down models can be utilized for yielding the pixel-wise weight map of each video frame, reflecting the saliency values of different pixels. Specifically, in light of study on the HVS, several low-level features have been developed for the bottom-up model of saliency detection. A representative bottom-up model is [9], in which the low-level features of color, intensity, and motion are integrated to yield saliency maps of videos. However, for the conversational videos, face is a consensus top-down visual cue for saliency detection. As such, in [10], the face is used

as a high-level feature for yielding the saliency maps of video frames. In this paper, we therefore apply bottom-up [9] and top-down [10] models in our SCC approach to provide the weights of saliency, for encoding generic and conversational videos, respectively.

For the bottom-up visual attention model, we apply phase spectrum of quaternion Fourier transform (PQFT) algorithm [9] to calculate the weight maps for all the frames in a video, due to its high accuracy and low complexity. For example, [9] has shown that the PQFT algorithm consumes less than 1 ms for computing the weight map of an 800×600 video frame on a Windows PC with C/C++ platform. Note that the videos may be downsampled for detecting saliency with the PQFT algorithm, once videos have high resolutions or there is the requirement on less computational time. The normalized weight map $\mathbf{v} = \{v_n\}_{n=1}^{N}$ ($v_n \in [0,1]$) for an N-pixel video frame can be computed in the PQFT algorithm by

$$\mathbf{v} = g * \sum_{t=0}^{3} \boldsymbol{\rho}_t^2 \tag{1}$$

where g is a 2-D Gaussian filter to smooth the weight map. In addition, $\boldsymbol{\rho}_t$ is the quaternion representation on four feature maps over the video frame, reconstructed by the PQFT on these maps. Note that the four feature maps include two color channels, one intensity channel, and one motion channel. For more details, refer to [9].

For the top-down visual attention model, we simply utilize our hierarchical perception model of face [10] to output the weight maps when encoding conversational video frames in our SCC approach. It is because the face is an evident cue for the visual attention model. It has been pointed out in [10] that facial regions are much more salient than background. Among the facial regions, facial features (e.g., eyes and mouth) are more visually significant than others, bringing about greater saliency. Therefore, the facial features should have the greatest weight, followed by the facial regions and then the background. Accordingly, pixel-wise weight map $\mathbf{v} = \{v_n\}_{n=1}^{N}$ of a conversational video frame can be estimated as

$$v_n = \begin{cases} 1, & n \in \mathbf{R}_1 \\ v', & n \in \mathbf{R}_2 \\ v'', & n \in \mathbf{R}_3 \end{cases} \tag{2}$$

where $\mathbf{R}_1$, $\mathbf{R}_2$, and $\mathbf{R}_3$ are the facial feature region, other facial region, and background. Here, these regions can be obtained using the same way as [10], in which the face and facial features are extracted according to the 66 landmarks of the real-time face alignment [39]. For less complexity, the 3000-frame/s face alignment [40] may be used for obtaining these regions instead of [39]. In (2), v' and v'' ($v'' < v' < 1$) are the weights of regions $\mathbf{R}_2$ and $\mathbf{R}_3$, respectively. In this paper, we follow [10] to set $v' = 0.4$ and $v'' = 0.2$. Note that these values are based on the eye tracking results on viewing conversational videos. Besides, $\mathbf{v}$ needs to be smoothed with 2-D Gaussian filter g as well. See [10] for more details.

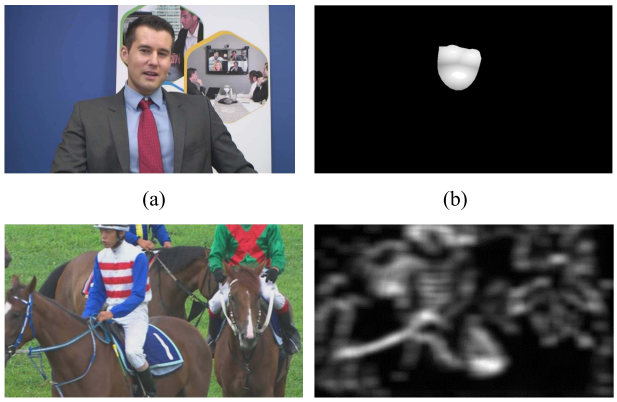


Fig. 1. Examples of weight maps. The intensities in saliency maps, ranging from 0 to 1, represent the output weights of pixels. Note that the map of *Johnny* is the output of the top-down visual attention model [10], whereas the map for *RaceHorses* is detected by the bottom-up model [9]. (a) *Johnny* (the 272nd frame). (b) Weight map of (a). (c) *RaceHorses* (the fourth frame). (d) Weight map of (c).

Fig. 1 shows examples for both top-down and bottom-up weight maps. From Fig. 1, we can observe that the weight map of the visual attention model utilized in this paper is able to roughly reflect the saliency of a video frame. However, the top-down map may miss some important regions in the background. Thus, the integration of bottom-up and top-down models for determining the weights is possible to make the saliency prediction more effective. As this paper mainly works on the complexity control for HEVC, the refinement of the visual attention model may hold for the future work.

Using the visual attention model, we can obtain the saliency weight for each pixel, which can be used to calculate the weight for each LCU. The weights of LCUs have an important effect on favoring subjective quality in our SCC approach, since they influence the maximum depth allocation to each LCU. The next section shows the details about how to incorporate the visual attention models in our approach.

IV. Proposed Method

In this section, we move to our SCC approach, based on the visual attention model of Section III, for the complexity control in HEVC. Since the quad-tree-based CTU partition takes a majority of encoding complexity in HEVC [41], it is possible to control encoding complexity of HEVC by setting various maximum depths $\{d_i\}_{i=1}^{I}$ ($d_i \in \{3, 2, 1, 0\}$) in advance to all I LCUs in each video frame.

Before introducing our work, we first briefly review the LCU partition structure in HEVC. During the partitioning process of the ith LCU, the RDO algorithm is exhaustively executed to select the optimal depth of each CU under the constraint of maximum depth d_i. We illustrate an example of LCU partitioning structure in Fig. 2. As shown in Fig. 2, the RDO algorithm is executed in the following repeated way: if the RD cost of root CU is larger than the aggregated RD cost of its leaf node CUs, the splitting of root CU is implemented; otherwise, the root CU is not allowed to be split. Thus, the RDO process needs to compute the RD costs of CU sizes at

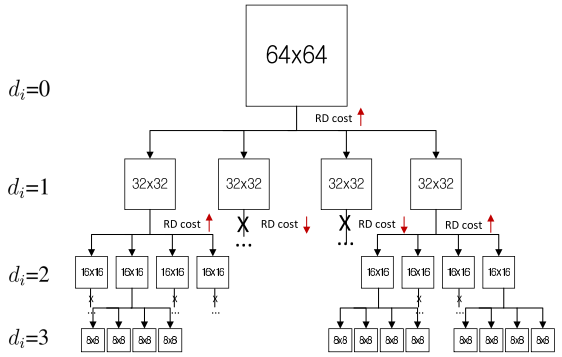


Fig. 2. Example of LCU partitioning structure. The upward arrow of RD cost indicates that the RD cost of root CU is larger than the accumulated RD cost of its leaf CUs, and thus, further splitting can be executed. The downward arrow indicates that further splitting is inhabited.

all the four depths (i.e., 3, 2, 1, and 0), once d_i is set to be the largest as 3, to finally choose the best one. This selection process indeed consumes huge complexity. However, if d_i is set to be 2, the splitting process has to be terminated once the CU sizes are reduced to 16×16 (i.e., splitting is stopped at layer $d_i = 2$ in Fig. 2). As such, only the RD costs of CUs at three depths (i.e., 2, 1, and 0) need to be compared by RDO process, resulting in less complexity. The smaller the d_i is set, the larger the complexity can be saved. Thus, the maximum depth d_i has a crucial effect on reducing the encoding complexity of HEVC [7].

The main cost on the above encoding complexity reduction is the visual distortion, defined by $\Delta D(d_i)$, which indicates the increased distortion of the ith LCU, caused by reducing its maximum depth to d_i. Actually, such visual distortion can be minimized in our approach. Beyond the minimization of visual distortion, subjective quality can be favored with the following constraint:

$$\forall w_i \geq w_l, \quad \Delta D(d_i) \leq \Delta D(d_l) \tag{3}$$

where w_i and w_l are the averaged weights of the ith and lth LCUs, respectively. They can be calculated on the basis of the pixel-wise weight map introduced in Section III using

$$w_i = \frac{1}{M} \sum_{n \in \mathbf{n}_i} v_n \tag{4}$$

where $\mathbf{n}_i$ stands for the set of pixel indices in the ith LCU of a video frame and M is the number of pixels in each LCU. Recall that v_n is the weight of each pixel in the video frame, ranging from 0 to 1, determined by the aforementioned visual attention model. Therefore, as formulated in (3), the visually salient LCUs can be imposed with higher visual quality (i.e., less visual distortion).

Therefore, in combination with the constraint on subjective visual quality, the complexity control in our SCC approach can be formulated by the following optimization model:

$$\underbrace{\min_{\{d_i\}_{i=1}^{I}} \sum_{i=1}^{I} \Delta D(d_i)}_{\text{Visual distortion minimization}} \quad \text{s.t.} \quad \underbrace{\frac{1}{I} \sum_{i=1}^{I} C(d_i) = T_c}_{\text{Encoding complexity constraint}} \quad \underbrace{\forall w_i \geq w_l, \Delta D(d_i) \leq \Delta D(d_l)}_{\text{Subjective-driven constraint}} \tag{5}$$

where T_c is the normalized target complexity, I is the total number of LCUs in each frame, and $C(d_i)$ is the normalized complexity of encoding the ith LCU with its maximum depth being d_i. $\Delta D(d_i)$ is the quality loss when the maximum depth of an LCU is decreased from the largest to d_i. From this equation, we can observe that the objective of complexity control in our SCC approach is to minimize visual distortion at a given target complexity. Meanwhile, it favors the subjective visual quality with the proposed subjective-driven constraint in (3). Note that (5) simply assumes the visual distortion and encoding complexity of each LCU to be unified at the same maximum depth. This assumption is reasonable as only aggregated visual distortion and averaged encoding complexity are required in (5).

To solve the complexity control problem formulated in (5), Section IV-A first explores the relationship between the maximum depth and encoding complexity to work out $C(d_i)$. Then, Section IV-B evaluates the influence of maximum depth on visual distortion to find out $\Delta D(d_i)$. Finally, Section IV-C proposes a solution to (5) and achieves the encoding complexity control in our SCC approach.

A. Relationship Between Maximum Depth and Encoding Complexity

As aforementioned, the quad-tree-based CTU partitioning scheme contributes to a large part of the encoding complexity in HEVC, on which the maximum depth of each LCU exerts an important effect. Therefore, it is interesting to investigate the relationship between the maximum depth and encoding complexity.

1) Training Sequences: For analyzing the relationship between the maximum depth and encoding complexity, we trained four video sequences at four different maximum depths (i.e., 3, 2, 1, and 0) on HEVC test model reference software (HM) 14.0. The training video sequences were selected from the standard HEVC test sequence database, including two 1920×1080 sequences *ParkScene* and *BQTerrace* from Class B and two 1280×720 sequences *KristenAndSara* and *Vidyo1* from Class E. The frame numbers trained for *BQTerrace*, *Vidyo1*, and *KristenAndSara* are 300, while *ParkScene* has 240 frames. Note that Class E contains conversational sequences with human faces. Also, note that these training sequences are different from the test video sequences used in the experiments of Section V.

2) Training Procedure: Given the training video sequences, we used a 64-bit Windows PC with Intel Core i7-4770 processor @3.40 GHz to investigate the encoding time at various maximum depths. Here, the low-delay P main configuration was used for the training. The training procedure is as follows. First, the video sequences were compressed with maximum depths of all the LCUs being the largest as 3. The encoding time[1] of each LCU was then recorded[2] as the reference time. Second, the training sequences were encoded with maximum

[1]In this paper, we use encoding time as the effective measurement of encoding complexity.

[2]Here, the encoding time is recorded by the *clock* function in visual C++ platform, which counts the number of CPU clock cycles to record the encoding time, during the execution of a program.

TABLE I
COMPLEXITY CONSUMPTION OF FOUR TEST SEQUENCES AT 1 Mbit/s

Sequences	$\widetilde{C}(2) \pm sd$	$\widetilde{C}(1) \pm sd$	$\widetilde{C}(0) \pm sd$
KristenAndSara	0.647 ± 0.039	0.385 ± 0.034	0.187 ± 0.016
Vidyo1	0.648 ± 0.058	0.365 ± 0.036	0.160± 0.013
BQTerrace	0.639 ± 0.031	0.381 ± 0.033	0.198± 0.014
ParkScene	0.674 ± 0.058	0.417 ± 0.057	0.224± 0.029

depths being other values (i.e., 2, 1, and 0). The encoding time of each LCU was also recorded. Suppose that $C_j(d_j^*)$, $d_j^* \in \{3, 2, 1, 0\}$, denotes the time of encoding the jth LCU at maximum depth d_j^*. Next, the time on encoding the jth LCU was normalized using reference time C_j (3)

$$\widetilde{C_j}(d_j^*) = \frac{C_j(d_j^*)}{C_j(3)}, \quad d_j^* \in \{3, 2, 1, 0\}. \tag{6}$$

Finally, the encoding complexity at each maximum depth can be obtained by averaging all $\widetilde{C_j}(d_j^*)$

$$\begin{bmatrix} \widetilde{C}(3) \\ \widetilde{C}(2) \\ \widetilde{C}(1) \\ \widetilde{C}(0) \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{1}{J}\sum_{j=1}^{J} \widetilde{C_j}(d_j^* = 2) \\ \frac{1}{J}\sum_{j=1}^{J} \widetilde{C_j}(d_j^* = 1) \\ \frac{1}{J}\sum_{j=1}^{J} \widetilde{C_j}(d_j^* = 0) \end{bmatrix} \tag{7}$$

where J is the total number of LCUs in each training sequence. $\widetilde{C}(3)$, $\widetilde{C}(2)$, $\widetilde{C}(1)$, and $\widetilde{C}(0)$ are the normalized encoding complexity in average with maximum depth being 3, 2, 1, and 0.

3) Training Results: Table I shows the training results of $\widetilde{C}(2)$, $\widetilde{C}(1)$, and $\widetilde{C}(0)$, for four sequences at 1 Mbit/s, using the above training procedure. Note that sd in Table I stands for standard deviations. From Table I, we can observe that the averaged encoding complexity at the same maximum depth is similar for all the four training sequences, and the corresponding standard deviations are relatively small. Furthermore, the complexity consumption under the same maximum depth is nearly unchanged across different bitrates, as shown in Fig. 3. Therefore, we can combine all the encoding time results of four sequences together, and then they are averaged to obtain a generic encoding complexity $C(d_i)$ at each maximum depth

$$C(d_i) = \begin{cases} 1, & d_i = 3 \\ 0.647, & d_i = 2 \\ 0.382, & d_i = 1 \\ 0.190, & d_i = 0 \end{cases} \tag{8}$$

for the complexity control in HEVC.

B. Relationship Between Maximum Depth and Visual Distortion

We have established the relationship between the maximum depth and encoding complexity in Section IV-A, as the basis

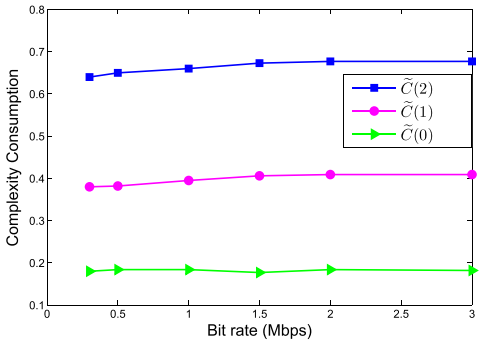


Fig. 3. Training sequence *Vidyo1*: influence of bitrates on complexity consumption of different maximum depths. The bitrates range from 300 kbits/s to 3 Mbits/s. The results of other video sequences are similar to that of *Vidyo1*.

of our SCC approach. However, there is no way to guarantee the visual quality after complexity control. From now on, we concentrate on exploring the influence of maximum depth on the visual distortion, for optimizing the visual distortion in our SCC approach. It is evident that the maximum depth reduction usually results in visual distortion, as only larger sized CUs are allowed. However, to the best of our knowledge, there is no study conducted on investigating the specific relationship between these two factors. Next, we devote to find out this relationship through training on a series of video sequences at different maximum depths.

1) Training Sequences: The video sequences used for investigating visual distortion are the same as those for training complexity, including two 1920 × 1080 sequences *ParkScene* and *BQTerrace* and two 1280×720 sequences *KristenAndSara* and *Vidyo1*.

2) Training Procedure: The distortion for training in this paper is measured in terms of mean square error (MSE).[3] The training procedure is similar to that in Section IV-A. First, we examined the MSE of each LCU at the largest maximum depth (i.e., 3), denoted by $\mathrm{MSE}_j(3)$ as reference. Then, the MSEs under other maximum depths (i.e., 2, 1, and 0) were also examined for each LCU, defined as $\mathrm{MSE}_j(d_j^*)$. Next, the normalized visual distortion of the jth LCU can be computed as

$$\Delta\widetilde{D_j}(d_j^*) = \frac{\mathrm{MSE}_j(d_j^*) - \mathrm{MSE}_j(3)}{\mathrm{MSE}_j(3)}. \tag{9}$$

By averaging $\Delta\widetilde{D_j}(d_j^*)$ of all the LCUs, we can obtain the visual distortion at different maximum depths

$$\begin{bmatrix} \Delta\widetilde{D}(3) \\ \Delta\widetilde{D}(2) \\ \Delta\widetilde{D}(1) \\ \Delta\widetilde{D}(0) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{J}\sum_{j=1}^{J} \Delta\widetilde{D_j}(d_j^* = 2) \\ \frac{1}{J}\sum_{j=1}^{J} \Delta\widetilde{D_j}(d_j^* = 1) \\ \frac{1}{J}\sum_{j=1}^{J} \Delta\widetilde{D_j}(d_j^* = 0) \end{bmatrix}. \tag{10}$$

[3]MSE is defined by $\|\mathbf{B}' - \mathbf{B}\|_2^2$, where $\mathbf{B}'$ and $\mathbf{B}$ are the pixel-wise intensities of the compressed and original video frames.

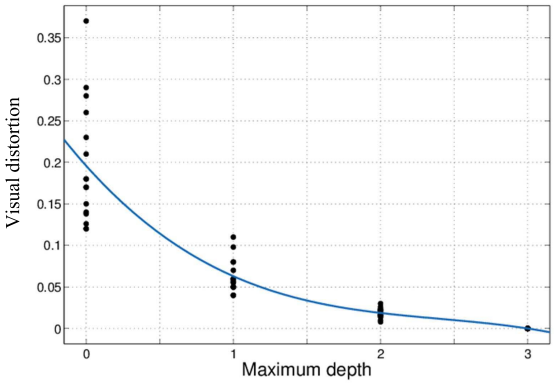


Fig. 4. Fitting curve for relationship between the maximum depth and visual distortion. The R-square for the fitting curve is 0.8073. There are 16 points for each maximum depth, which are obtained from four training sequences at four different bitrates: 500 kbits/s, 1 Mbit/s, 1.5 Mbits/s, and 3 Mbits/s.

Recall that J is the LCU number in each video sequence. $\Delta\widetilde{D}(2)$, $\Delta\widetilde{D}(1)$, and $\Delta\widetilde{D}(0)$ are the overall normalized visual distortion when the maximum depths are 2, 1, and 0. However, different from training encoding complexity, the visual distortion is not unified across different training sequences at various bitrates. Therefore, the following employs a least-square fitting scheme to work out $\Delta D(d_i)$.

3) Training Results: We examined $\Delta\widetilde{D}(2)$, $\Delta\widetilde{D}(1)$, and $\Delta\widetilde{D}(0)$ for the four video sequences at bitrates of 500 kbits/s, 1 Mbit/s, 1.5 Mbits/s, and 3 Mbits/s. The normalized visual distortion of kth compressed video under different maximum depths can be denoted by $\Delta D_k(3)$, $\Delta D_k(2)$, $\Delta D_k(1)$, and $\Delta D_k(0)$, with $k \in \{1, \ldots, K\}$. Here, K is 16 (i.e., four sequences at four bitrates) in our training, as shown in Fig. 4. The polynomial fitting is employed in our approach, to model the relationship between the maximum depth and visual distortion, with the least-square error on these data. In fact, the following lemma holds for the least-square polynomial fitting in our work.

Lemma 1: Given four groups of data $\{(0, \Delta D_k(0))\}_{k=1}^K$, $\{(1, \Delta D_k(1))\}_{k=1}^K$, $\{(2, \Delta D_k(2))\}_{k=1}^K$, and $\{(3, \Delta D_k(3))\}_{k=1}^K$, the optimal least-square polynomial fitting function of $\Delta D(d_i)$, $d_i \in \{3, 2, 1, 0\}$, is

$$\Delta D(d_i) = a_0 + a_1 d_i + a_2 d_i^2 + a_3 d_i^3 \tag{11}$$

where

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0^2 & 0^3 \\ 1 & 1 & 1^2 & 1^2 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(0) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(1) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(2) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(3) \end{bmatrix}. \tag{12}$$

Proof: Refer to the Appendix.

According to Lemma 1, given the above training data on visual distortion of four sequences at four bitrates, we have the following relationship between the maximum depth and visual distortion:

$$\Delta D(d_i) = 0.200 - 0.203 d_i + 0.077 d_i^2 - 0.0105 d_i^3 \quad d_i \in \{3, 2, 1, 0\}. \tag{13}$$

The fitting curve of the maximum depth and visual distortion is shown in Fig. 4. The R-square value of this fitting curve is 0.8073, verifying the effectiveness of the fitting.

C. Encoding Complexity Control

The goal of complexity control in our SCC approach is twofold: 1) reducing the actual encoding complexity to any given target level with minimal visual distortion and 2) ensuring the subjective video quality while reducing the encoding complexity. Given the established $C(d_i)$ and $\Delta D(d_i)$, we can achieve the first goal for each video frame by removing the subjective-driven constraint away from (5)

$$\min_{\{N_p\}_{p=0}^3} \sum_{p=0}^{3} \Delta D(p) N_p \quad \text{s.t. } \frac{1}{I}\sum_{p=0}^{3} C(p) N_p = T_c, \quad \sum_{p=0}^{3} N_p = I \tag{14}$$

where N_p is the number of LCUs with maximum depth $d_i = p$ in the frame and I is the total number of LCUs in the frame. Note that the sum of N_p is equivalent to I. After solving (14), $\{N_p\}_{p=0}^3$ can be obtained. Once $\{N_p\}_{p=0}^3$ are achieved, the objective distortion is minimized and fixed at a given target complexity. Next, the subjective quality needs to be favored according to the subjective-driven constraint, at the same optimized objective distortion. To be more specific, LCUs with greater weight w_i are the salient regions, and their visual quality should be preferred by assigning larger maximum depth d_i. Assuming that $S(w_i)$ indexes the sorted w_i with an ascending order, the following can be obtained to make d_i satisfy the subjective-driven constraint in (5):

$$d_i = \begin{cases} 0, & S(w_i) \le N_0 \\ 1, & N_0 < S(w_i) \le N_0 + N_1 \\ 2, & N_0 + N_1 < S(w_i) \le N_0 + N_1 + N_2 \\ 3, & S(w_i) > N_0 + N_1 + N_2 \end{cases} \tag{15}$$

where N_0, N_1, and N_2 have been obtained by solving (14), representing the numbers of LCUs with maximum depth being 0, 1, and 2 in each frame. From (15), we can observe that the LCU with greater weight is endowed with larger maximum depth, indicating higher visual quality. In this way, the subjective quality can be favored. Through working out (14) and (15), the optimization problem of (5) can be solved to output various maximum depths for LCUs in a video frame. Now, the remaining tasks are to solve (14) and to sort the averaged weight w_i of each LCU.

1) Solving (14): Actually, the problem in (14) can be seen as a general multiconstrained knapsack problem [42] in integer programming. Thus, we can utilize the existing tools for the knapsack problem to solve (14). As we know, the knapsack problem is $\mathcal{NP}$ − hard [42], meaning that it is intractable to directly obtain an optimal solution to (14).

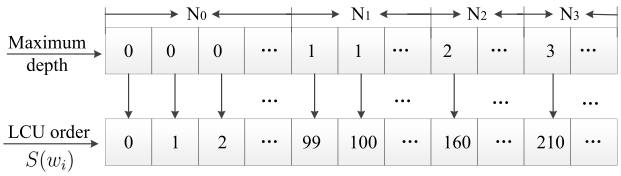
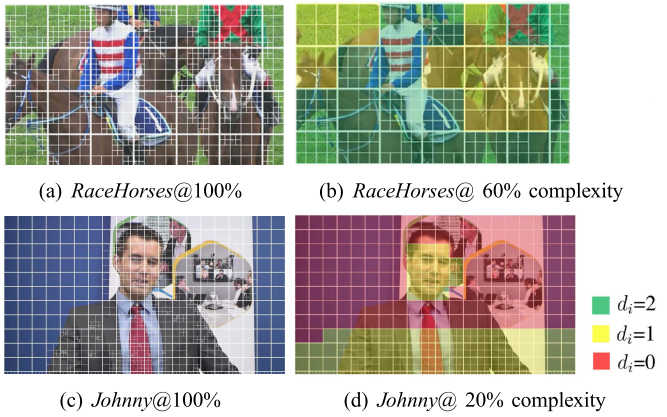


Fig. 5. Example of allocating maximum depth to each LCU in terms of the sorted weight.



(a) *RaceHorses*@100% (b) *RaceHorses*@ 60% complexity

(c) *Johnny*@100% (d) *Johnny*@ 20% complexity

Fig. 6. Examples for the maximum depth distribution of LCUs in two frames. (a) and (b) Fourth frame of *RaceHorses*. (c) and (d) 272nd frame of *Johnny*. Note that the maximum depths of LCUs in (a) and (c) are all equivalent to 3.

TABLE II
TEST VIDEO SEQUENCES

Sequences	Class	Resolution	Frames	Rate control
BasketballDrive	B	1920×1080	300@50fps	Enable
Cactus	B	1920×1080	300@50fps	Enable
Kimono	B	1920×1080	240@24fps	Disable
BQMall	C	832×480	300@60fps	Enable
PartyScene	C	832×480	300@50fps	Enable
Racehorses	C	832×480	300@30fps	Disable
BasketballDrill	C	832×480	300@50fps	Disable
RaceHorses	D	416×240	300@30fps	Enable
BasketballPass	D	416×240	300@50fps	Enable
BlowingBubbles	D	416×240	300@50fps	Disable
BQSquare	D	416×300	300@60fps	Disable
Johnny	E	1280×720	300@60fps	Enable
Vidyo4	E	1280×720	300@60fps	Enable
Fourpeople	E	1280×720	300@60fps	Disable

TABLE III
PARAMETER SETTINGS OF HM 14.0

GOP structure	IPPP
LCU size	64 × 64
Maximum LCU depth	3
Motion Search Range	64
SAO	1
FEN	1
FDM	1
TransformSkip	1
Intra period	-1

Fortunately, in the past decades, several tools were proposed for solving the knapsack problem [43], including the branch-and-bound, dynamic programming, state-space relaxation, and preprocessing algorithms. In this paper, we employ the branch-and-bound algorithm to solve (14), as it is widely applied for solving the knapsack problem. The running time of the branch-and-bound algorithm in our work is less than 0.1 s, far less than the encoding time of the whole sequence. Note that the branch-and-bound algorithm needs to be conducted only once, as its results can be used for all the frames at a given target complexity.

The branch-and-bound algorithm was first proposed in [44]. Its key idea is to branch a large solution set into several small subsets and to find the optimal solution of each small subset, as the upper/lower bound of the ultimate optimal solution. By continuously branching the subsets, the upper and lower bounds are gradually approaching with each other, until the final solution is obtained. The solving process of (14) using the branch-and-bound tool is presented in the Appendix. Finally, the ultimate optimal solution $\{N_p\}_{p=0}^{3}$ can be obtained. As such, we can get the optimal amounts of LCUs under four different maximum depths, i.e., $p = 0, 1, 2,$ or 3. However, the explicit assignment of maximum depths to different LCUs has not been worked out yet. Actually, it can be solved upon (15) with the following scheme on sorting w_i.

Sorting w_i: As can be observed in (15), the final output of our SCC approach can be acquired, once the averaged weights w_i of all the LCUs are sorted with an ascending order. Here, the quicksort algorithm [45] is used for sorting w_i. In our work, the running time of quicksort algorithm for a 1920 × 1080 video frame is less than 1 ms. Due to its high computing speed, the consuming time of sorting can be ignored compared with the encoding time of video frames. Next, based on the sorted index $S(w_i)$, the corresponding LCU can be allocated with a maximum depth d_i according to (15). Fig. 5 shows an example of allocating maximum depths to LCUs, in terms of the sorted index $S(w_i)$. Our SCC approach can guarantee that larger maximum depth is allocated to the LCUs with larger weights. In this way, the subjective quality can be favored.

Finally, the complexity control in formulation (5) can be solved, and we can obtain the maximum depth for each LCU, as the output of our approach. Fig. 6 shows examples of the maximum depth distribution of LCUs using our SCC approach, based on the weight maps of the two video frames shown in Fig. 1. From Fig. 6, we can observe that the encoding complexity is decreased via reducing the maximum depths of LCUs with relatively smaller weights. For instance, the grass region in *RaceHorses* has smaller weights (as it draws little human attention) such that smaller maximum depths are imposed to the LCUs of grass.

V. EXPERIMENTAL RESULTS

In this section, experiments were conducted to validate the effectiveness of the proposed SCC approach. The effectiveness was evaluated from four aspects: 1) the control accuracy; 2) objective visual quality; 3) subjective peak signal-to-noise ratio (PSNR); and 4) Bjøntegaard delta (BD)-rate/BD-PSNR. In the experiments, we compared our approach only with the state-of-the-art approaches [7] and [23]. Note that we did not compare with [36], which is also a state-of-the-art approach, as it cannot precisely control the complexity.[4]

[4]In [36], a global complexity factor, ranging from 0 to 1, needs to be set for different encoding complexities. However, the time saving is proportional only to the predefined complexity factor, but does not satisfy the specific target.

TABLE IV
CLASS B: THE RESULTS OF HEVC ENCODING COMPLEXITY CONTROL FOR OUR AND OTHER TWO APPROACHES

Video sequence	Target bit-rate (Kbps)	Target complexity (%)	Actual complexity (%)			Y-PSNR (dB)			Actual bit-rate (Kbps)		
			SCC	[7]	[23]	SCC	[7]	[23]	SCC	[7]	[23]
BasketballDrive	5000	100	100	100	100	36.59	36.59	36.59	5002.1	5002.1	5002.1
		80	84.6	85.5	**77.2**	**36.53**	36.47	36.38	5002.2	5002.1	5002.1
		60	**60.8**	57.8	58.3	**36.37**	36.32	36.34	5002.2	5002.2	5002.2
		40	**38.3**	43.2	43.5	36.18	36.20	**36.25**	5002.4	5002.5	5002.2
		20	**19.7**	39.8	42.3	35.69	–	–	5002.3	5002.4	5002.4
	10000	100	100	100	100	38.06	38.06	38.06	10002.6	10002.6	10002.6
		80	**82.8**	76.5	77.0	**38.02**	37.97	37.98	10002.5	10003.0	10002.6
		60	**59.6**	58.8	58.9	**37.90**	37.87	37.87	10002.2	10002.8	10002.7
		40	**38.5**	42.8	43.6	37.77	37.78	**37.83**	10002.6	10002.7	10002.7
		20	**18.3**	38.9	43.2	36.90	–	–	9998.2	10002.6	10002.7
Cactus	5000	100	100	100	100	36.24	36.24	36.24	5001.4	5001.4	5001.4
		80	83.5	83.2	**77.5**	**36.21**	36.14	36.10	5001.4	5001.5	5001.4
		60	**62.0**	57.1	57.8	**36.08**	35.95	36.02	5001.3	5001.6	5001.5
		40	**37.9**	42.9	44.8	35.82	35.81	**35.84**	5001.2	5001.6	5001.5
		20	**19.8**	39.1	42.5	35.15	–	–	5001.4	5001.7	5001.6
	10000	100	100	100	100	37.46	37.46	37.46	10001.5	10001.5	10001.5
		80	**82.4**	83.2	77.0	**37.43**	37.38	37.36	10001.5	10001.6	10001.5
		60	**60.9**	57.9	57.8	**37.33**	37.25	37.27	10001.4	10001.5	10001.6
		40	**37.5**	43.1	45.3	37.14	37.14	**37.20**	10000.4	10001.8	10001.5
		20	**19.2**	39.5	41.0	36.70	–	–	9998.2	10001.7	10001.5

A. Test Sequences and Parameter Setting

The test video sequences were chosen from Classes B–E in the standard HEVC test sequence database, as shown in Table II. For thoroughly evaluating the performance of our approach, all the test sequences were divided into two sets: with and without rate control. In the rate-control set, the video sequences were compressed at two target bitrates, by the default rate-control scheme. The quantization parameter (QP) value for the first I frame was set to be 32 by default, and the QP values for other frames were determined by the rate-control scheme. In the nonrate-control set, the video sequences were compressed with four fixed QPs (i.e., 22, 27, 32, and 37), to evaluate the performance of BD-rate and BD-PSNR.

The encoder configuration we used in all the experiments is low-delay *P* main,[5] following the test configurations of [7] and [23]. Since HEVC common test conditions do not test Class A in the low-delay case, this class was not included in our experiments. Class F was not tested as well, as it is with noncamera captured content. Note that all the test sequences are different from those for training in Section IV.

All the experiments were implemented on HM 14.0 platform, with the typical parameter settings presented in Table III. From Table III, we can observe that the maximum LCU depth was initially set to 3. The LCU size was chosen to be 64×64 to allow all possible CU sizes, i.e., 64×64, 32×32, 16×16, and 8×8, for optimizing the distortion in HEVC. Some fast encoding modes such as fast encoder setting, fast decision for merge RD cost, and TransformSkip were enabled in our approach. Our approach may be incorporated with other fast encoding methods, with the relationships of Section IV being retrained. It is worth pointing out that the approach in this paper and the approaches in [7] and [23] all used the same parameter settings on the HM 14.0 platform.

[5]Our approach can also be applied to other configurations, such as random access, but the relationships of Section IV need to be retrained.

B. Evaluation of Control Accuracy

The experimental results in Tables IV–VII demonstrate the accuracy of controlling encoding complexity in the proposed SCC approach. In Tables IV–VII, the encoding complexity is normalized to be the form of percentage, via being divided by the encoding time of conventional HM 14.0. Note that the approach in this paper and the approaches in [7] and [23] are all reduced to the conventional HM 14.0, when the encoding time is set to 100%. From Tables IV–VII, one can observe that our SCC approach is capable of precisely controlling the encoding complexity of HEVC. Specifically, the actual encoding complexity is quite close to the target complexity. Among all the test sequences across different bitrates, the least error is only 0.2% (*Cactus* 20% @5 Mbits/s). Except from Class D sequences at 20% complexity, the largest control error is 4.6% (*BasketballDrive* 80% @5 Mbits/s). Our approach does not perform well in Class D sequences at 20% target complexity, since the number of LCUs is such so small (i.e., each frame has only 18 LCUs with size being 64×64) that the control accuracy is difficult to be guaranteed.

We also compare the accuracy of complexity control among our SCC and two other approaches [7], [23] in Tables IV–VII. The comparison results demonstrate that our SCC approach outperforms other two approaches in terms of control accuracy. Given the same target complexity, the actual encoding complexity of our SCC approach has smaller errors than other two approaches, for almost all the cases. In addition, our approach is capable of controlling encoding complexity with a larger range. More specifically, as can be seen from Tables IV, V, and VII, our SCC approach is able to control the complexity as low as 20% with little deviations from the target, while [7] and [23] cannot succeed in this task.

Apart from the accuracy, the complexity control stability is another component in evaluating the control accuracy. Fig. 7 compares the actual encoding complexity per frame for the approach in this paper and the approaches in [7] and [23]

TABLE V
CLASS C: THE RESULTS OF HEVC ENCODING COMPLEXITY CONTROL FOR OUR AND OTHER TWO APPROACHES

Video sequence	Target bit-rate (Kbps)	Target complexity (%)	Actual complexity (%)			Y-PSNR (dB)			Actual bit-rate (Kbps)		
			SCC	[7]	[23]	SCC	[7]	[23]	SCC	[7]	[23]
BQMall	500	100	100	100	100	31.20	31.20	31.20	501.9	501.9	501.9
		80	**83.0**	76.6	75.0	**31.10**	31.06	31.08	501.9	502.1	502.0
		60	**61.1**	64.5	58.2	**31.05**	30.90	30.88	502.0	502.3	502.0
		40	**40.3**	42.3	43.6	**30.50**	30.47	30.49	501.9	502.2	502.1
		20	**20.9**	39.2	42.3	29.56	–	–	501.9	502.2	502.2
	1000	100	100	100	100	34.05	34.05	34.05	1002.0	1002.0	1002.0
		80	83.5	83.8	**77.9**	**33.92**	33.89	33.89	1002.0	1002.3	1002.0
		60	**60.5**	56.5	58.6	**33.64**	33.42	33.60	1002.0	1002.3	1002.0
		40	**39.6**	42.1	44.3	**33.10**	33.07	33.08	1002.0	1002.2	1002.1
		20	**20.3**	38.6	44.1	31.98	–	–	1002.1	1002.4	1002.2
PartyScene	2000	100	100	100	100	30.96	30.96	30.96	2002.0	2002.0	2002.0
		80	83.7	77.8	**78.5**	**30.82**	30.73	30.77	2001.9	2002.1	2002.0
		60	**58.7**	55.0	58.0	**30.53**	30.32	30.50	2001.9	2002.2	2002.1
		40	**39.2**	42.0	43.4	30.03	30.01	**30.05**	2002.2	2002.1	2002.1
		20	**18.9**	35.6	43.2	28.84	–	–	2002.3	2003.3	2002.2
	4000	100	100	100	100	33.76	33.76	33.76	4001.8	4001.8	4001.8
		80	**81.7**	78.1	78.2	**33.59**	33.48	33.51	4001.8	4002.0	4001.8
		60	**58.6**	55.7	58.3	**33.26**	33.02	33.17	4001.9	4002.3	4001.9
		40	**38.7**	42.3	45.6	32.64	32.61	**32.65**	4001.9	4002.4	4002.1
		20	**18.0**	36.4	45.3	31.33	–	–	4002.3	4002.8	4002.2

TABLE VI
CLASS D: THE RESULTS OF HEVC ENCODING COMPLEXITY CONTROL FOR OUR AND OTHER TWO APPROACHES

Video sequence	Target bit-rate (Kbps)	Target complexity (%)	Actual complexity (%)			Y-PSNR (dB)			Actual bit-rate (Kbps)		
			SCC	[7]	[23]	SCC	[7]	[23]	SCC	[7]	[23]
BasketballPass	200	100	100	100	100	30.94	30.94	30.94	201.6	201.6	201.6
		80	**75.8**	72.1	73.6	**30.69**	30.61	30.64	201.6	201.7	201.6
		60	**62.1**	63.5	56.7	**30.46**	30.32	30.27	201.7	201.7	201.7
		40	**44.3**	44.7	53.6	**30.34**	30.28	30.25	201.7	201.7	201.7
		20	**29.4**	36.2	53.5	30.11	–	–	201.8	201.9	201.9
	500	100	100	100	100	34.94	34.94	34.94	501.7	501.7	501.7
		80	**76.0**	72.3	74.5	**34.54**	34.43	34.45	501.7	501.7	501.7
		60	**61.1**	57.1	57.9	**34.21**	33.87	33.89	501.7	501.7	501.7
		40	**43.8**	44.5	54.7	33.62	33.60	**33.66**	501.8	501.8	501.8
		20	**28.3**	35.8	54.2	33.57	–	–	501.9	501.9	501.9
RaceHorses	300	100	100	100	100	32.12	32.12	32.12	301.0	301.0	301.0
		80	**82.1**	74.5	75.8	**31.82**	31.61	31.68	301.0	301.0	301.0
		60	62.4	57.5	**57.7**	**31.53**	31.15	31.20	301.0	301.0	301.0
		40	**44.4**	**44.4**	56.2	**31.21**	31.08	31.18	301.1	301.1	301.1
		20	**28.5**	38.9	56.0	30.92	–	–	301.1	301.1	301.1
	500	100	100	100	100	34.50	34.50	34.50	500.9	500.9	500.9
		80	**78.1**	75.5	76.8	**34.08**	33.94	33.99	500.1	501.0	500.9
		60	**60.9**	58.4	58.6	**33.67**	33.35	33.38	501.0	501.0	501.0
		40	**44.3**	44.6	57.3	**33.29**	33.20	33.26	501.1	501.1	501.1
		20	**29.2**	39.7	57.1	33.04	–	–	501.1	501.1	501.1

The variances of actual encoding complexity across different frames are also provided in Fig. 7, to quantify the stability of complexity control. From Fig. 7, we can find that our approach is far more steady than other two approaches, in controlling the encoding complexity of HEVC across frames.

C. Assessment of Objective Quality

Tables IV–VII also present the Y-PSNRs of all the test sequences for our approach. It can be observed from Tables IV–VII that our SCC approach has relatively small Y-PSNR reductions for all the video sequences when the encoding complexity decreases to 80%. Specifically, the Y-PSNR reduction is up to 0.06 dB for Class B (*BasketballDrive*@5 Mbits/s), 0.17 dB for Class C (*PartySecne*@4 Mbits/s), 0.42 dB for Class D (*Racehorses*@500 kbits/s), and 0.02 dB for Class E (*Johnny* and *Vidyo4*@500 kbits/s). Besides, the averaged Y-PSNR reduction is 0.04 dB for Class B, 0.14 dB for Class C, 0.34 dB for Class D, and 0.015 dB for Class E. When the encoding complexity drops to 60%, there exists larger Y-PSNR decrease. For example, the Y-PSNR reduction is up to 0.22 dB for Class B (*Cactus*@10 Mbits/s), 0.50 dB for Class C (*PartySecne*@4 Mbits/s), 0.83 dB for Class D (*RaceHorses*@500 kbits/s), and 0.07 dB for Class E (*Johnny*@500 kbits/s). In addition, the Y-PSNR averagely drops by 0.17, 0.37, 0.66, 0.055, for Classes B–E. Similarly, the larger Y-PSNR loss is incurred when we set 40% and 20% encoding complexity. In a word, we can find out that the loss of both maximal and averaged Y-PSNRs increases along with the reduced encoding complexity.

Tables IV–VII also compare the Y-PSNR results among our SCC and the approaches in [7] and [23]. We can observe that our approach in most cases can provide higher Y-PSNRs than other two approaches, given the same target complexity. How-

TABLE VII

CLASS E: THE RESULTS OF HEVC ENCODING COMPLEXITY CONTROL FOR OUR AND OTHER TWO APPROACHES

Video sequence	Target bit-rate (Kbps)	Target complexity (%)	Actual complexity (%)			Y-PSNR (dB)			Actual bit-rate (Kbps)		
			SCC	[7]	[23]	SCC	[7]	[23]	SCC	[7]	[23]
Johnny	500	100	100	100	100	40.57	40.57	40.57	502.1	501.9	501.9
		80	**83.6**	85.3	75.6	**40.55**	40.53	40.51	502.0	502.1	502.0
		60	**61.1**	62.3	57.9	**40.50**	40.47	40.45	501.9	502.0	502.0
		40	**38.8**	41.5	42.7	**40.35**	40.32	40.34	501.9	501.9	502.1
		20	**24.5**	36.7	42.2	40.05	–	–	501.9	501.9	502.1
	1000	100	100	100	100	41.66	41.66	41.66	1002.1	1002.1	1002.1
		80	**81.3**	78.3	77.3	**41.65**	41.64	41.63	1002.0	1001.9	1002.0
		60	**59.4**	65.5	57.9	**41.62**	41.60	41.61	1002.0	1001.9	1002.0
		40	**40.5**	42.6	41.3	**41.54**	41.50	41.52	1002.0	1002.0	1002.1
		20	**23.3**	37.2	40.5	41.27	–	–	1002.0	1002.2	1002.1
Vidyo4	500	100	100	100	100	38.88	38.88	38.88	502.1	502.1	502.1
		80	**83.3**	84.5	75.6	**38.86**	38.80	38.77	502.0	502.0	502.1
		60	**59.6**	59.2	58.2	**38.83**	38.72	38.72	502.1	502.0	502.1
		40	**39.4**	38.0	42.1	**38.61**	38.55	38.60	502.0	502.1	502.1
		20	**24.5**	35.0	41.5	38.29	–	–	502.1	502.1	502.1
	1000	100	100	100	100	40.61	40.61	40.61	1002.2	1002.2	1002.2
		80	82.3	78.2	**78.8**	**40.60**	40.55	40.53	1002.1	1002.2	1002.2
		60	**60.4**	63.3	61.8	**40.55**	40.49	40.50	1002.2	1002.2	1002.3
		40	**41.3**	41.8	42.7	**40.41**	40.39	**40.41**	1002.2	1002.2	1002.3
		20	**24.4**	38.4	42.2	40.11	–	–	1002.2	1002.1	1002.3

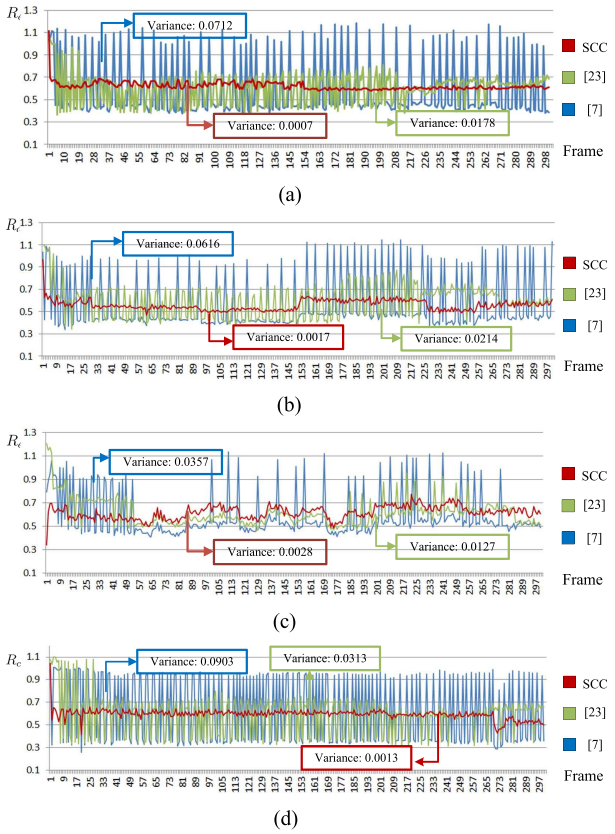


Fig. 7. Actual encoding complexity (R_c) comparisons per frame among the approach in this paper and the approaches in [7] and [23]. The target complexity for all the four sequences is 60%. For other targets, the fluctuations are similar. (a) *Cactus*. (b) *BQMall*. (c) *BasketballPass*. (d) *Johnny*.

ever, for some cases with 40% complexity, the Y-PSNRs of our approach are a bit smaller than [23]. This is due to the fact that the actual encoding complexity of [23] is much larger than that of our approach, thus resulting in a bit higher Y-PSNRs. It can be further observed from Tables IV–VII that both [7] and [23] are incapable of reducing the encoding complexity as low as 20% (all above 35%), and thus their corresponding PSNRs at 20% are not presented.

Fig. 8 shows the comparison of ΔPSNR per frame for the approach in this paper and the approaches in [7] and [23]. In Fig. 8, ΔPSNR is the Y-PSNR reduction when the target complexity is decreased from 100% to 60%. To quantify the quality fluctuation across different frames, the variances of ΔPSNR are provided in Fig. 8. We can observe from Fig. 8 that our approach can always offer steady Y-PSNR reduction for each frame, with relatively small variances. In particular, ΔPSNR of our SCC approach across different frames remains nearly constant for *Cactus* and *Johnny*, as their variances are only 0.0010 and 0.0005, respectively. However, for other two approaches, ΔPSNR across frames fluctuates at a large range, especially for *BasketballPass*, the variances of which are 0.1109 and 0.1302 for [7] and [23], respectively. In fact, this kind of large quality fluctuation has an adverse effect on the overall visual quality.

D. Assessment of Subjective PSNR

We follow the way of [46] to define the subjective Y-PSNR for reflecting the subjective quality. Wang and Li [47] have proved that with saliency weighting, the conventional PSNR metric can be converted to a competitive metric called subjective Y-PSNR, which has a much higher correlation with the subjective quality. Therefore, the subjective Y-PSNR, which weighs Y-PSNR according to the saliency value of each pixel, is adopted in this paper. According to [47], the subjective Y-PSNR S_{PSNR} can be calculated as

$$S_{\mathrm{MSE}} = \frac{\sum_{n=1}^{N} v_n (I'_n - I_n)^2}{\sum_{n=1}^{N} v_n} \tag{16}$$

and

$$S_{\mathrm{PSNR}} = 10 \log \left(\frac{255^2}{S_{\mathrm{MSE}}} \right) \tag{17}$$

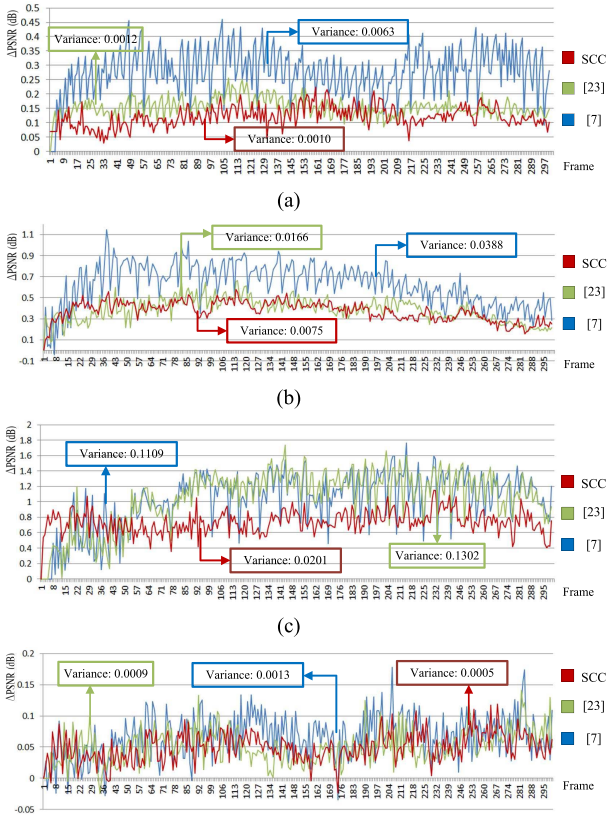


Fig. 8. Comparisons of Y-PSNR reductions per frame among the approach in this paper and the approaches in [7] and [23]. The target complexity for all the four sequences is 60%. (a) *Cactus*. (b) *BQMall*. (c) *BasketballPass*. (d) *Johnny*.

TABLE VIII
AVERAGED SUBJECTIVE AND OBJECTIVE Y-PSNR IMPROVEMENT OVER [7] AND [23]

Target complexity	Subjective Y-PSNR (dB)		Objective Y-PSNR (dB)	
	over [7]	over [23]	over [7]	over [23]
80%	**0.085**	0.057	0.072	**0.071**
60%	**0.166**	**0.164**	0.138	0.116
40%	**0.061**	**0.022**	0.040	-0.004

where I_n and I'_n are the intensities of the nth pixel in the original and encoded frames, respectively. Besides, recall that N is the total amount of pixels in one frame and that υ_n is the saliency weight of the nth pixel.

Then, we tabulate in Table VIII the averaged subjective Y-PSNR improvement of our approach over [7] and [23]. As we can observe from Table VIII, our approach is able to yield higher subjective Y-PSNRs than [7] and [23] for all the cases. This verifies the effectiveness of the subjective-driven constraint. Actually, as aforementioned, the actual encoding complexity of [7] and [23] is far more than 20% (all above 35%) when the target complexity is 20%. Thus, their subjective PSNRs at 20% complexity are not reported in Table VIII.

Figs. 9 and 10 show some frames of video sequences compressed by the approach in this paper and the approaches in [7] and [23]. From Figs. 9 and 10, we can observe that there exist evident block effects and severe visual distortion in [7] and [23]. By contrast, our approach has a better visual quality. For example, we can easily find in Fig. 9 that the left eye of the horse nearly disappears for [7] and [23], which may cause wretched subjective feelings.

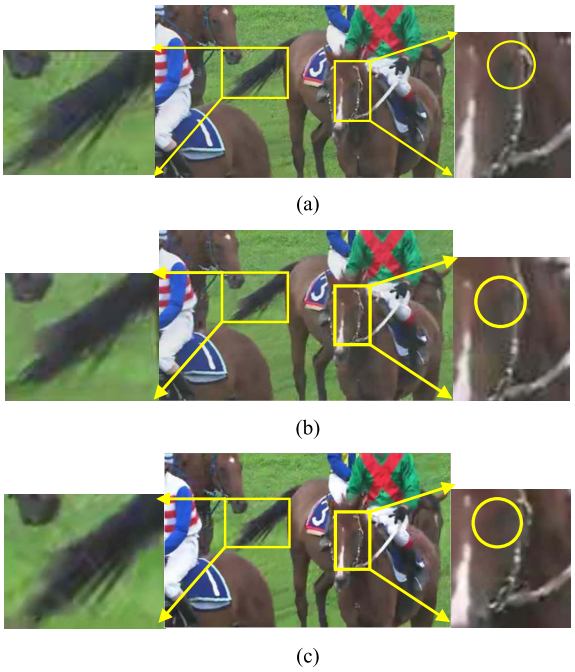


Fig. 9. The 32nd frames of sequence *Racehorses* compressed by the approach in this paper and the approaches in [7] and [23], at a 60% encoding complexity. (a) *Racehorses* with our approach at 300 kbits/s. (b) *Racehorses* with [7] at 300 kbits/s. (c) *Racehorses* with [23] at 300 kbits/s.

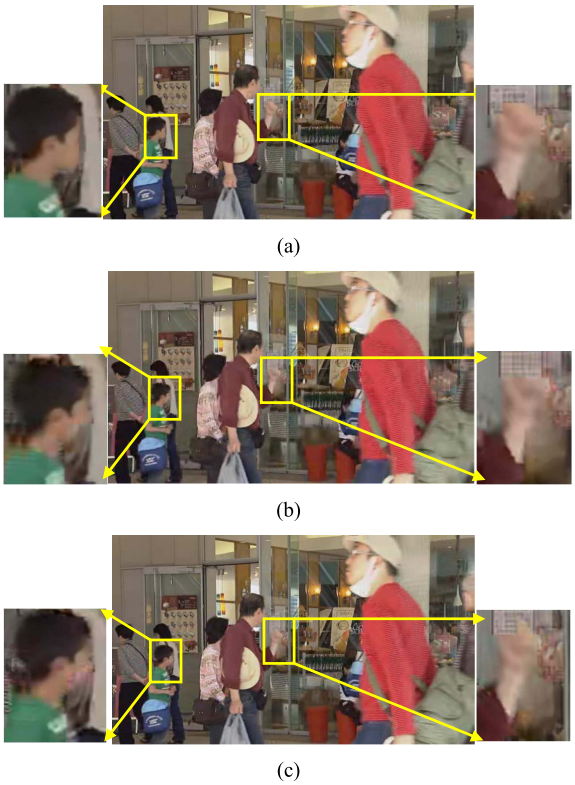


Fig. 10. The 54th frames of sequence *BQMall* compressed by the approach in this paper and the approaches in [7] and [23], at a 60% encoding complexity. (a) *BQMall* with our approach at 500 kbits/s. (b) *BQMall* with [7] at 500 kbits/s. (c) *BQMall* with [23] at 500 kbits/s.

TABLE IX
BD-RATE AND BD-PSNR COMPARISONS, WITH HM 14.0 AS AN ANCHOR

Target Complexity	Our Approach			Approach [7]			Approach [23]		
	BD-PSNR[dB]	BD-Rate[%]	Accuracy	BD-PSNR[dB]	BD-Rate[%]	Accuracy	BD-PSNR[dB]	BD-Rate[%]	Accuracy
80%	-0.047	0.841	**2.4**	-0.065	1.279	4.3	-0.052	1.151	3.1
60%	-0.095	2.268	**0.6**	-0.223	5.155	2.0	-0.206	5.034	1.4
40%	-0.372	7.301	**1.0**	-0.489	11.944	4.6	-0.459	10.600	4.5
20%	-0.756	17.487	**5.7**	–	–	15.4	–	–	16.8

E. Assessment of BD-Rate and BD-PSNR

In this section, we focus on evaluating the performance of our approach with fixed QPs and disabled rate control. We tested six sequences from different classes. As shown in Table II, these sequences include *Kimono* from Class B, *PartyScene* and *Racehorses* from Class C, *BlowingBubbles* and *BQSquare* from Class D, and *Fourpeople* from Class E. The six sequences were tested with different target complexities (80%, 60%, 40%, and 20%), at four fixed QPs (22, 27, 32, and 37). The BD-rate and BD-PSNR were then calculated for each sequence at different encoding complexities, using the method in [48]. Table IX reports the results of BD-rate and BD-PSNR averaged over all the six test sequences. Since [7] and [23] cannot control the complexity to 20%, their corresponding BD-rate and BD-PSNR results at 20% are not reported. It can be observed from Table IX that compared with [7] and [23], our approach can achieve higher Y-PSNRs at the same bitrates, or save some bitrates at the same distortion. Besides, our approach, especially at low complexity, can significantly improve the control accuracy over [7] and [23].

VI. CONCLUSION

In this paper, we proposed a novel approach, namely, SCC, for encoding complexity control in HEVC. It was argued that the maximum depth of each LCU exerts an important effect on the encoding complexity of HEVC. Therefore, with numerical analysis, we first investigated the relationship between the maximum depth and encoding complexity. The reduced maximum depth can decrease encoding complexity, but at the cost of visual distortion. Thus, we further explored the influence of maximum depth on visual distortion. Accordingly, we proposed an optimization formulation to control the encoding complexity of HEVC with minimal visual distortion. Based on the visual attention model, this formulation also favors subjective visual quality, by allocating more complexity resources to LCUs with greater weights. The experimental results verify the effectiveness of our approach. On the one hand, in comparison with other two approaches, our approach is capable of steadily controlling the encoding complexity of HEVC with higher accuracy. On the other hand, our approach can offer superior visual quality over other two state-of-the-art approaches.

There may exist three directions for the future work.

1) Our work in this paper considers only two simple visual attention models. Currently, with the advances in the area of visual attention modeling, much work related to the SCC approach remains to be developed for further improving subjective visual quality. For example, one future goal of our SCC approach should include the integration of bottom-up and top-down visual attention models to provide more reasonable weight maps.
2) Our work in its present form focuses only on allocating complexity resources to LCUs. Yet, there is no complexity budget scheme to adjust the complexity allocation in frame level, which may further improve the accuracy of complexity control. This provides a promising trend for the future work.
3) Many fast encoding and early termination methods are proposed for HEVC. It is quite an interesting future work to incorporate our approach with those fast encoding methods, to further decrease and control the encoding complexity of HEVC.

APPENDIX

A. Proof of Lemma 1

The 1-D optimal fitting x_0 for data group $\{\Delta D_k(0)\}_{k=1}^{K}$ can be calculated via solving

$$\min_{x_0} \sum_{k=1}^{K} (\Delta D_k(0) - x_0)^2 \tag{18}$$

and we have

$$\min_{x_0} \left\{ K x_0^2 - 2 \sum_{k=1}^{K} \Delta D_k(0) x_0 + \sum_{k=1}^{K} \Delta D_k(0)^2 \right\}. \tag{19}$$

Taking the derivative of x_0 in (19), we can obtain the optimal value of x_0 satisfying the least-square error of (18)

$$x_0 = \frac{1}{K} \sum_{k=1}^{K} \Delta D_k(0). \tag{20}$$

Similarly, we have

$$\frac{1}{K} \sum_{k=1}^{K} \Delta D_k(1), \quad \frac{1}{K} \sum_{k=1}^{K} \Delta D_k(2) \quad \text{and} \quad \frac{1}{K} \sum_{k=1}^{K} \Delta D_k(3)$$

for the optimal fitting of $\{\Delta D_k(1)\}_{k=1}^{K}$, $\{\Delta D_k(2)\}_{k=1}^{K}$, and $\{\Delta D_k(3)\}_{k=1}^{K}$.

Next, we extend the optimal fitting to be 2-D. Given the above optimal fitting data $(0, (1/K) \sum_{k=1}^{K} \Delta D_k(0))$, $(1, (1/K) \sum_{k=1}^{K} \Delta D_k^k(1))$, $(2, (1/K) \sum_{k=1}^{K} \Delta D_k(2))$, and $(3, (1/K) \sum_{k=1}^{K} \Delta D_k(3))$, we need to fit only on these four 2-D data with minimal error. According to Lagrange

interpolation principle, a third-order polynomial equation is able to fit the four 2-D data with zero error

$$\Delta D(d_i) = a_0 + a_1 d_i + a_2 {d_i}^2 + a_3 {d_i}^3, \quad d_i \in \{3, 2, 1, 0\}. \tag{21}$$

Upon (21), coefficients a_0, a_1, a_2, and a_3 can be calculated through solving the following matrix equation:

$$\begin{bmatrix} 1 & 0 & 0^2 & 0^3 \\ 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(0) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(1) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(2) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(3) \end{bmatrix} \tag{22}$$

where $0, 1, 2$, and 3 in the 4×4 matrix (except the first column) at the left-hand side represent four different values of d_i. Note that such a 4×4 matrix is a Vandermonde matrix, and obviously, it is of full rank. Thus, there exists one and only one solution to (22). The only solution for coefficients a_0, a_1, a_2, and a_3 can be obtained by solving (22)

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0^2 & 0^3 \\ 1 & 1 & 1^2 & 1^3 \\ 1 & 2 & 2^2 & 2^3 \\ 1 & 3 & 3^2 & 3^3 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(0) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(1) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(2) \\ \frac{1}{K}\sum_{k=1}^{K} \Delta D_k(3) \end{bmatrix}. \tag{23}$$

Finally, this lemma is proved.

B. Solving (14) With the Branch-and-Bound Algorithm

1) Step 1—Relax: Reduce the integer programming problem to a linear programming problem. By removing the constraint that all the variables $\{N_p\}_{p=0}^3$ are integers, the original integer programming problem of (14) can be relaxed to a common linear programming problem, denoted by $\mathcal{L}$. The optimal solution to $\mathcal{L}$ can be yielded as $(\gamma_0, \gamma_1, \gamma_2, \gamma_3)$. If all the elements of the solution are integers, the algorithm stops for outputting the final solution to (14). Otherwise, we need to record the objective value of $(\gamma_0, \gamma_1, \gamma_2, \gamma_3)$ as $\underline{\mathcal{Z}}$, the lower bound of the optimal objective value $\mathcal{Z}$. Meanwhile, we can find one feasible solution to (14), the objective value of which is recorded as $\overline{\mathcal{Z}}$, the upper bound of $\mathcal{Z}$.

2) Step 2—Branch: Split the solution space into smaller subspaces. Specially, one noninteger element of $(\gamma_0, \gamma_1, \gamma_2, \gamma_3)$ needs to be chosen to construct two new constraints. Taking the example of choosing γ_1, the two constraints are $N_1 \leq \lfloor \gamma_1 \rfloor$ and $N_1 \geq \lfloor \gamma_1 \rfloor + 1$, where $\lfloor \gamma_1 \rfloor$ indicates the largest integer not exceeding γ_1. Then, the two constraints are added to $\mathcal{L}$, respectively, constructing two new subproblems $\mathcal{L}_1$ and $\mathcal{L}_2$. Similarly, the optimal solutions to $\mathcal{L}_1$ and $\mathcal{L}_2$ are calculated by linear programming.

3) Step 3—Bound: Calculate the upper/lower bound of each branch. Specially, there exist three cases, in terms of the solution to $\mathcal{L}_1$ and $\mathcal{L}_2$.

Case 1: Either $\mathcal{L}_1$ or $\mathcal{L}_2$ has an integer solution. If the objective value of the branch with an integer solution is smaller than $\overline{\mathcal{Z}}$, then this value is recorded as a new $\overline{\mathcal{Z}}$. The objective value of the other branch is compared with $\overline{\mathcal{Z}}$. If it is larger than $\overline{\mathcal{Z}}$, this branch is discarded or pruned. Otherwise, the algorithm moves to Step 2, in which this branch is recursively divided.

Case 2: Neither $\mathcal{L}_1$ nor $\mathcal{L}_2$ has an integer solution. The branch whose objective value is larger than $\overline{\mathcal{Z}}$ is discarded. The other branch moves to Step 2, and its objective value is recorded as a new $\underline{\mathcal{Z}}$. If both objective values of $\mathcal{L}_1$ and $\mathcal{L}_2$ are less than $\overline{\mathcal{Z}}$, the two values are compared with each other, of which the smaller one is recorded as a new $\underline{\mathcal{Z}}$.

Case 3: There is no solution (both integer and noninteger) to $\mathcal{L}_1$ or $\mathcal{L}_2$. Then, the branch with no solution is discarded directly.

Steps 2 and 3 are recursively executed for $(\gamma_0, \gamma_1, \gamma_2, \gamma_3)$ until all the branches cannot be further branched. Finally, the ultimate optimal solution $\{N_p\}_{p=0}^3$ can be obtained.

REFERENCES

[1] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] M. T. Pourazad, C. Doutre, M. Azimi, and P. Nasiopoulos, "HEVC: The new gold standard for video compression: How does HEVC compare with H.264/AVC?" *IEEE Consum. Electron. Mag.*, vol. 1, no. 3, pp. 36–46, Jul. 2012.

[3] W.-J. Han *et al.*, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.

[4] G. Corrêa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Performance and computational complexity assessment of high-efficiency video encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1899–1909, Dec. 2012.

[5] I. R. Ismaeil, A. Docef, F. Kossentini, and R. K. Ward, "A computation-distortion optimized framework for efficient DCT-based video coding," *IEEE Trans. Multimedia*, vol. 3, no. 3, pp. 298–310, Sep. 2001.

[6] C. S. Kannangara, I. E. Richardson, M. Bystrom, and Y. Zhao, "Complexity control of H.264/AVC based on mode-conditional cost probability distributions," *IEEE Trans. Multimedia*, vol. 11, no. 3, pp. 433–442, Apr. 2009.

[7] G. Corrêa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Complexity control of high efficiency video encoders for power-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1866–1874, Nov. 2011.

[8] J.-S. Lee and T. Ebrahimi, "Perceptual video compression: A survey," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 6, pp. 684–697, Oct. 2012.

[9] C. Guo and L. Zhang, "A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression," *IEEE Trans. Image Process.*, vol. 19, no. 1, pp. 185–198, Jan. 2010.

[10] M. Xu, X. Deng, S. Li, and Z. Wang, "Region-of-interest based conversational HEVC coding with hierarchical perception model of face," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 3, pp. 475–489, Jun. 2014.

[11] Y. Liu, Z. G. Li, and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 134–139, Jan. 2008.

[12] X. Xu and Y. He, "Improvements on fast motion estimation strategy for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 285–293, Mar. 2008.

[13] C.-M. Mak, C.-K. Fong, and W.-K. Cham, "Fast motion estimation for H.264/AVC in Walsh–Hadamard domain," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 6, pp. 735–745, Jun. 2008.

[14] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *J. Vis. Commun. Image Represent.*, vol. 17, no. 2, pp. 264–290, 2006.

[15] Y.-W. Huang, B.-Y. Hsieh, S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, Apr. 2006.

[16] X. Li, M. Wien, and J.-R. Ohm, "Rate-complexity-distortion optimization for hybrid video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 7, pp. 957–970, Jul. 2011.

[17] C. S. Kannangara *et al.*, "Low-complexity skip prediction for H.264 through Lagrangian cost estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 2, pp. 202–208, Feb. 2006.

[18] F. Pan *et al.*, "Fast mode decision algorithm for intraprediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.

[19] J.-F. Wang, J.-C. Wang, J.-T. Chen, A.-C. Tsai, and A. Paul, "A novel fast algorithm for intra mode decision in H.264/AVC encoders," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2006, pp. 1–4.

[20] D. Wu *et al.*, "Fast intermode decision in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 953–958, Jul. 2005.

[21] H. Wang, S. Kwong, and C.-W. Kok, "An efficient mode decision algorithm for H.264/AVC encoding optimization," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 882–888, Jun. 2007.

[22] J. Leng, L. Sun, T. Ikenaga, and S. Sakaida, "Content based hierarchical fast coding unit decision algorithm for HEVC," in *Proc. Int. Conf. Multimedia Signal Process. (CMSP)*, vol. 1. May 2011, pp. 56–59.

[23] G. Corrêa, P. Assuncao, L. Agostini, and L. A. da Silva Cruz, "Coding tree depth estimation for complexity reduction of HEVC," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2013, pp. 43–52.

[24] X. Shen, L. Yu, and J. Chen, "Fast coding unit size selection for HEVC based on Bayesian decision rule," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 453–456.

[25] M. U. K. Khan, M. Shafique, and J. Henkel, "An adaptive complexity reduction scheme with fast prediction unit decision for HEVC intra encoding," in *Proc. IEEE ICIP*, Sep. 2013, pp. 1578–1582.

[26] H.-M. Yoo and J.-W. Suh, "Fast coding unit decision algorithm based on inter and intra prediction unit termination for HEVC," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2013, pp. 300–301.

[27] C.-C. Wang, Y.-C. Liao, J.-W. Wang, and C.-W. Tung, "An effective TU size decision method for fast HEVC encoders," in *Proc. Int. Symp. Comput., Consum., Control (IS3C)*, Jun. 2014, pp. 1195–1198.

[28] G. Corrêa, P. A. Assuncao, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC encoding decisions using data mining," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 4, pp. 660–673, Apr. 2015.

[29] H. Zhang and Z. Ma, "Fast intra mode decision for High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 660–668, Apr. 2014.

[30] J. Vanne, M. Viitanen, and T. D. Hamalainen, "Efficient mode decision schemes for HEVC inter prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 9, pp. 1579–1593, Sep. 2014.

[31] K. Miyazawa, T. Murakami, A. Minezawa, and H. Sakate, "Complexity reduction of in-loop filtering for compressed image restoration in HEVC," in *Proc. Picture Coding Symp. (PCS)*, May 2012, pp. 413–416.

[32] P.-L. Tai, S.-Y. Huang, C.-T. Liu, and J.-S. Wang, "Computation-aware scheme for software-based block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 9, pp. 901–913, Sep. 2003.

[33] Z. Yang, H. Cai, and J. Li, "A framework for fine-granular computational-complexity scalable motion estimation [real-time video coding applications]," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2005, pp. 5473–5476.

[34] X. Gao, L. Zhuo, and L. Shen, "Complexity allocation and control algorithm for H.264 inter coding based on mobile devices," in *Proc. 2nd Int. Congr. Image Signal Process.*, Oct. 2009, pp. 1–5.

[35] L. Su, Y. Lu, F. Wu, S. Li, and W. Gao, "Complexity-constrained H.264 video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 477–490, Apr. 2009.

[36] T. Zhao, Z. Wang, and S. Kwong, "Flexible mode selection and complexity allocation in High Efficiency Video Coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1135–1144, Dec. 2013.

[37] J. Ren, N. Kehtarnavaz, and M. Budagavi, "Computationally efficient mode selection in H.264/AVC video coding," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 877–886, May 2008.

[38] Z. Wang and A. C. Bovik, "Embedded foveation image coding," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1397–1410, Oct. 2001.

[39] J. M. Saragih, S. Lucey, and J. F. Cohn, "Face alignment through subspace constrained mean-shifts," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 1034–1041.

[40] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 FPS via regressing local binary features," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 1685–1692.

[41] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[42] D. Pisinger, "Algorithms for Knapsack problems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Copenhagen, København, Denmark, Feb. 1995.

[43] T. Ibaraki, "Enumerative approaches to combinatorial optimization," in *Annals of Operations Research*, vols. 10–11. Basel, Switzerland: Baltzer, 1987.

[44] P. J. Kolesar, "A branch and bound algorithm for the knapsack problem," *Manage. Sci.*, vol. 13, no. 9, pp. 723–735, 1967.

[45] C. A. Hoare, "Quicksort," *Comput. J.*, vol. 5, no. 1, pp. 10–16, 1962.

[46] Z. Li, S. Qin, and L. Itti, "Visual attention guided bit allocation in video compression," *Image Vis. Comput.*, vol. 29, no. 1, pp. 1–14, 2011.

[47] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1185–1198, May 2011.

[48] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-m33, 2001, pp. 290–294.

Xin Deng (S'14) received the B.S. degree in electronic engineering from Beihang University, Beijing, China, in 2013, where she is currently working toward the master's degree.

Her research interests include perceptual video coding and video quality metrics.

Ms. Deng received the National Scholarship from the Chinese College Students and the Outstanding Graduate Student Award from Beihang University. She also received the 2014 IEEE Circuits and Systems Society Student Travel Award.

Mai Xu (M'10) received the B.S. degree from Beihang University, Beijing, China, in 2003; the M.S. degree from Tsinghua University, Beijing, in 2006; and the Ph.D. degree from Imperial College London, London, U.K., in 2010.

He was a Research Fellow with the Electrical Engineering Department, Tsinghua University, from 2010 to 2012. Since 2013, he has been an Associate Professor with Beihang University. He has authored over 30 technical papers in international journals and conference proceedings. His current research interests include visual communication and image processing.

Lai Jiang received the bachelor's degree from Beihang University, Beijing, China, in 2015, where he is currently working toward the master's degree.

His research interests include saliency detection and video analysis.

Xiaoyan Sun (M'04–SM'10) received the B.S., M.S., and Ph.D. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 1997, 1999, and 2003, respectively.

She has been with Microsoft Research Asia, Beijing, China, since 2004, where she is currently a Lead Researcher with the Internet Media Group. She has authored or co-authored over 60 journal and conference papers and ten proposals to standards. Her research interests include image and video compression, image processing, computer vision, and cloud computing.

Dr. Sun was a recipient of the best paper award of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 2009.

Zulin Wang (M'14) received the B.S. and M.S. degrees in electronic engineering and the Ph.D. degree from Beihang University, Beijing, China, in 1986, 1989, and 2000, respectively.

He is the Dean of the School of Electronic and Information Engineering with Beihang University. He has authored or co-authored over 100 papers and holds six patents, and has authored two books in these fields. He has undertaken approximately 30 projects related to image/video coding and wireless communication. His research interests include image processing, electromagnetic countermeasure, and satellite communication technology.