Flexible Mode Selection and Complexity Allocation in High Efficiency Video Coding

Tiesong Zhao, *Member, IEEE*, Zhou Wang, *Senior Member, IEEE*, and Sam Kwong, *Senior Member, IEEE*

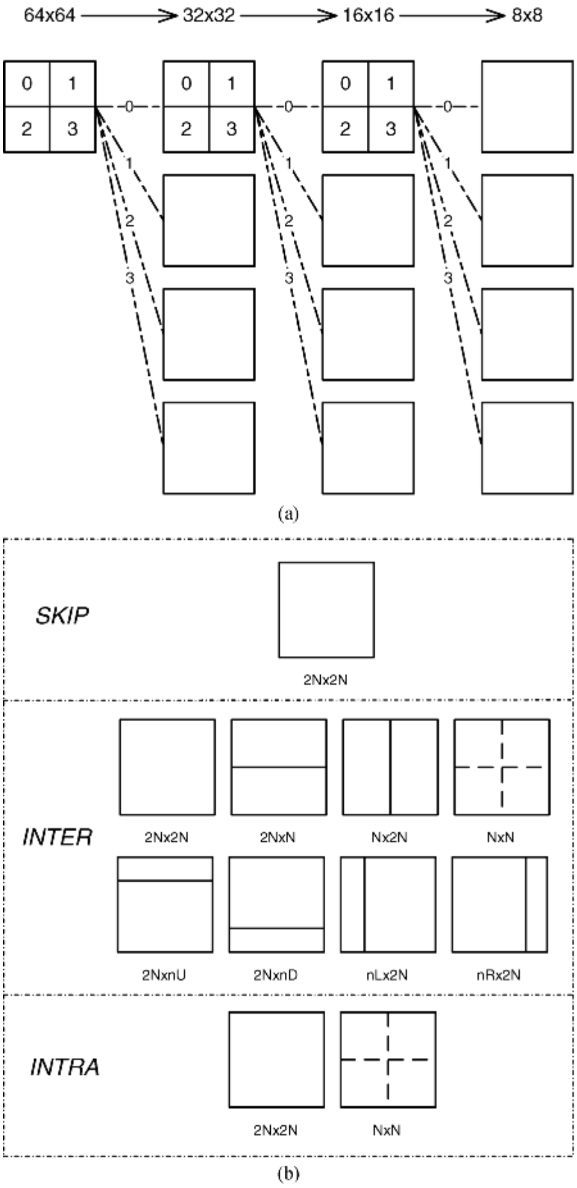
***Abstract*—To improve compression performance, High Efficiency Video Coding (HEVC) employs a quad-tree based block representation, namely Coding Tree Unit (CTU), which can support larger partitions and more coding modes than a traditional macroblock. Despite its high compression efficiency, the number of combinations of coding modes increases dramatically, which results in high computational complexity at the encoder. Here we propose a flexible framework for HEVC coding mode selection, with a user-defined global complexity factor. Based on linear programming, a hierarchical complexity allocation scheme is developed to allocate computational complexities among frames and Coding Units (CUs) to maximize the overall Rate-Distortion (RD) performance. In each CU, with the allocated complexity factor, a mode mapping based approach is employed for coding mode selection. Extensive experiments demonstrate that, with a series of global complexity factors, the proposed model can achieve good trade-offs between computational complexity and RD performance.**

***Index Terms*—Complexity allocation, HEVC, linear programming, mode decision, video coding.**

I. Introduction

THE first edition of High Efficient Video Coding (HEVC) [1], the test model (HM) of the next generation video coding standard, developed by the Joint Collaborative Team on Video Coding (JCT-VC), a joint effort of ITU-T VCEG and ISO/IEC MPEG standardization organizations, has been finalized in January 2013, aiming for a maximum of 50% bit rate reduction for equal perceptual quality to the existing video coding standards [2], [3].

At the core of the coding layer of HEVC is the Coding Tree Unit (CTU), which could be considered as an extension of macroblock in H.264 [4], but with a size that can be larger than a macroblock in [2], [3]. In HEVC, the CTU can be split into smaller Coding Units (CUs) [5] in a quad-tree structure [6], as shown in Fig. 1(a). A CU consists of one luminance Coding Block (CB) and two chrominance CBs and syntax elements. To decide whether to code a CU with inter or intra prediction, the CU and associated CBs are further divided into smaller Prediction Units (PUs) and Prediction Blocks (PBs), as shown in Fig. 1(b). In HEVC, a wide range of PB sizes from 64 × 64 to 4 × 4 are supported, which results in a large number of combinations of mode partitions in CTU. Therefore, it is highly desirable to optimize the encoding procedure for computational complexity reduction while maintaining the compression efficiency.



(a)

(b)

Fig. 1. CTU structure in HEVC. (a) CU splitting with luminance size from 64 × 64 to 8 × 8. (b) Coding mode partitions for splitting a $2N \times 2N$ CB into PBs, where $N \times N$ is supported only when $2N = 8$ $(N = 4)$.

Manuscript received January 07, 2013; revised May 12, 2013; accepted June 20, 2013. Date of publication June 27, 2013; date of current version November 18, 2013. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Nam Ling.

T. Zhao and Z. Wang are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: ztiesong@uwaterloo.ca; zhouwang@ieee.org).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (e-mail: cssamk@cityu.edu.hk).

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/JSTSP.2013.2271421

In recent years, many mode selection algorithms have been proposed for H.264 and its extensions. Some focused on fast intra mode selection [7]–[12] and others on fast inter mode selection [13]–[25]. In these algorithms, to achieve high computational complexity reduction, a variety of techniques are proposed, including mode classification and dependencies based on image and coding mode features [13], [14], neighboring prediction [15], [16], early skip mode determination [17], [18], unnecessary mode skipping based on Rate-Distortion (RD) cost thresholds [19], [20], All Zero Block (AZB) detection for early termination [21], [22], and the other optimization theories [23]–[25]. In some algorithms, more than one of these techniques are employed to develop hybrid approaches; especially for neighboring prediction, which is utilized in almost all mode selection algorithms due to good time reduction performance and low cost in computation. Although the fundamental ideas of these techniques could still be employed to design mode selection algorithms for HEVC, the above H.264 algorithms cannot be directly used in HEVC encoder due to the changes in coding structures and neighboring predictions after CTU is introduced.

Inspired by the aforementioned techniques and fundamental ideas, some efficient mode selection algorithms have been proposed for HEVC [26]–[36]. Among them, a number of schemes are developed to reduce the number of intra candidates in Rough Mode Decision (RMD) [37], and thus to speed up intra prediction mode selection in HEVC [26], [27]–[31]. In [26], intra direction information is predicted from neighboring blocks to select the most probable intra prediction modes. In [27], relationship between block sizes of the current PU and the related Transform Unit (TU) is studied to determine the intra candidates. In [28], the neighboring direction information is also exploited, with a gradient-mode histogram. In [29], spatial angular correlation of intra prediction modes between the current and neighboring blocks is computed, using modulo-N arithmetic operations. In [30], edge direction information of the current PU is also investigated and employed to decide intra candidates. The direction information of co-located neighboring block is also used in [31], together with Sum of Absolute Transformed Differences (SATD) threshold prediction from pre-coded PU.

Because there are more inter partitions in HEVC (see Fig. 1(b)), there is good potential to reduce the computational complexity by developing advanced inter coding mode selection algorithms [32]–[36]. In [32], Gweon *et al.* proposed a Coded Block Flag (CBF) based early termination method, in which if there exist zero CBFs for all luminance and chrominance components, the remaining coding partitions of the current CU could be totally skipped, except CU splitting and N × N mode. In [33], Choi *et al.* proposed an early CU termination method, in which the CU splitting and related sub-CU coding could be skipped when the best prediction mode of the current CU is Skip mode. In [34], Teng *et al.* presented another early termination scheme for CU merging and splitting, in which all-zero block detection and RD cost based early termination are employed to skip unnecessary coding mode partitions. In [35], Tan *et al.* exploited the RD cost thresholds for early terminations of both coding partitions and CU splitting. In [36], Kim *et al.* proposed an early skip mode determination method, using Differential Motion Vectors (DMVs) and CBF of inter 2N × 2N mode.

In this paper, we propose a flexible coding mode selection method, with a user-defined global complexity factor. To achieve efficient computational complexity reduction, a flexible scheme is developed for coding mode selection on CU level, based on mode mapping [16]. To maximize the overall RD performance, a complexity allocation scheme is proposed, which can determine complexity factors for all frames and CUs. Comprehensive experiments demonstrate the effectiveness and robustness of our method, with good trade-offs between computational complexity, bit rates and compressed video quality.

The following of the paper is organized as follows. In Section II, flexible coding mode selection is presented with mode mapping; in Section III, complexity allocation is exploited in a Group-Of-Pictures (GOP). The simulation results are given and discussed in Section IV. Finally, Section V concludes the paper.

II. Flexible Mode Selection in CU Coding

To develop a flexible mode selection scheme in CU coding, an intuitive and effective way is to find a complexity factor, which is positively correlated with the number of coding modes to be tested. In such a case, a lower complexity factor represents fewer candidate mode partitions. Based on the optimal stopping theory [25], to maintain the RD performance with a limited number of modes, these mode candidates should have the highest predicted probability. In this work, the mode mapping assumption [16] is utilized to determine these mode candidates.

A. Flexible Mode Selection by Mode Mapping

By investigating the correlations between Motion Vectors (MVs) of coding mode partitions, a mode mapping method was proposed in [16], which uses Euclidean Points (EPs) to represent different mode partitions, and thus the relevancy between mode partitions can be measured by Euclidean distance. In mode mapping method, the following assumptions are made:

(i) Each coding mode partition can be projected to an EP in a mode space that is Euclidean;

(ii) In the mode space, a smaller Euclidean distance between two EPs represent a higher correlation between the two corresponding mode partitions, and vice versa;

(iii) For each CU to be coded, there exists a corresponding CU Point (CP) in the mode space, which may or may not be an EP;

(iv) The corresponding EP of the best coding mode, is with the minimum Euclidean distance to CP among all mode partitions.

In mode mapping method, there are two related problems yet to be addressed: how to obtain EPs for all coding modes, and how to estimate CPs for all CUs to be coded. The following two sections, II-B, II-C, will focus on the two issues, respectively.

If the predicted CP (denoted as CP^*) is exactly the “real” CP of the CU to be coded, we can simply select the mode partition with the nearest EP and skip all the other modes. However, the “real” CP can only be obtained after the CU is coded. Hence, there exists prediction error between CP^* and the “real” CP. To avoid potential RD loss, usually more than one mode partitions

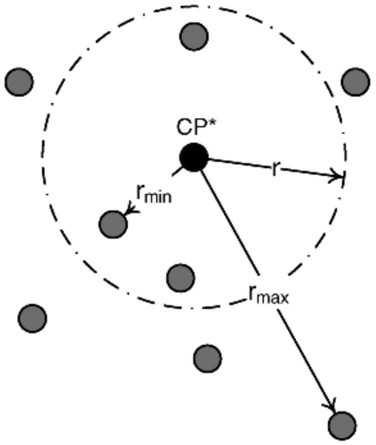


Fig. 2. Flexible mode selection with mode mapping.

should be tested, which is illustrated in Fig. 2. Given CP*, the minimum and maximum distances between CP* and all EPs are first computed as r_{min} and r_{max}, respectively. With a complexity factor $\theta \in [0, 1]$, the search radius r is defined as

$$r = (1 - \theta) \cdot r_{min} + \theta \cdot r_{max}. \tag{1}$$

With the center at CP*, all coding partitions with the corresponding EPs in the search radius are chosen as candidate modes, and the remaining modes are all skipped. In (1), when the complexity factor $\theta = 0$, $r = r_{min}$ indicates only the most probable mode partition is examined. In the other extreme, when $\theta = 1$, all candidate modes are examined with full complexity.

B. Determination of EPs for Coding Modes

In HEVC, a mode partition size consists of two components: the CU depth in quad-tree, which indicates the size of the corresponding CB; and the sizes of PBs, which indicates how the CB is divided for Motion Estimation (ME). Accordingly, both the relative Euclidean coordinates of all CU depths and the relative Euclidean coordinates of all PBs in a CU should be considered to determine EPs for all coding modes. In [16], the above relationships are studied with 16 × 16 and its sub-8 × 8 blocks, which are extended to different CU sizes in this work. From [16], the mode partitions 2N × 2N, 2N × N, N × 2N and N × N (or CU splitting when $N > 4$) constitute a square; and a CU and its sub-CU can be illustrated as two conjunctive squares in a two-dimensional Euclidean space, as shown in Fig. 3, where the ratio between sizes of the two squares is decided to be $R1/R2 = 3/2$.

Ulteriorly, the relative positions between all EPs are determined based on the above observations, with the largest CU size 64 × 64. We set the EP value for 64 × 64 mode (i.e., Skip or 2N × 2N mode when CU size is 64 × 64) as (0, 0), and the EP value for 4 × 4 mode (i.e., N × N mode when CU size is 8 × 8) is (130, 130). The other EPs are consequently determined and shown in Table I. The asymmetric partition (AMP), such as $2\text{N} \times \text{nU}/\text{nD}$, $\text{nL}/\text{nR} \times 2\text{N}$ in Fig. 1(b), are with MV distribution between 2N × 2N and the corresponding symmetric partition, and thus we set the EP as the average of the two. Specifically, the EP of $2\text{N} \times \text{nU}/\text{nD}$ is set to be the average EP of 2N × 2N and 2N × N; the EP of $\text{nL}/\text{nR} \times 2\text{N}$ is set to be the average EP of 2N × 2N and N × 2N.

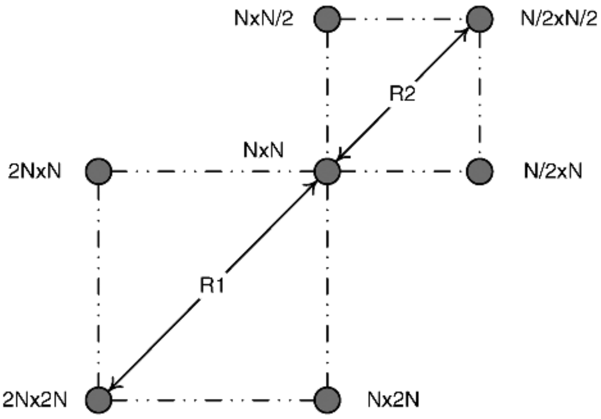


Fig. 3. Illustration of EPs in 2D Euclidean mode space.

TABLE I
EPs OF ALL CODING MODE PARTITIONS

CU depth		0	1	2	3
CU size		64×64	32×32	16×16	8×8
EPs	2N×2N	(0, 0)	(54, 54)	(90, 90)	(114, 114)
	2N×N	(0, 54)	(54, 90)	(90, 114)	(114, 130)
	N×2N	(54, 0)	(90, 54)	(114, 90)	(130, 114)
	N×N	(54, 54)	(90, 90)	(114, 114)	(130, 130)
	2N×nU/nD	(0, 27)	(54,72)	(90, 102)	(114, 122)
	nL/nR×2N	(27, 0)	(72,54)	(102, 90)	(122, 114)

After a CU with depth d is coded, the CP can be obtained by averaging all EPs of its mode partitions, or averaging all CPs of its sub-CUs if this CU is further divided,

$$\overrightarrow{P}_d = \begin{cases} \frac{1}{m} \sum\limits_{i=0}^{m-1} EP_{d,i} & \text{if } d = d_{max}, \\ \frac{1}{4} \sum\limits_{i=0}^{3} \overrightarrow{P}_{d+1,i} & \text{otherwise,} \end{cases} \tag{2}$$

where d and d_{max} represent the current and maximum depth of the CU, respectively; and m denotes the number of PBs.

C. Prediction of CPs for CUs

In the mode selection scheme described in Section II-A, the value of CP* (denoted as $\overrightarrow{P}^*$) should be estimated for each CU depth before the corresponding CU is coded. Due to the fact that there exist strong correlations between neighboring blocks in natural video sequences, we predict CP* with the neighboring CUs, and these CUs can be further classified into two categories. The temporal CU, with the value of its CP denoted as $\overrightarrow{P}_T$, is the co-located CU in the nearest reference frame; and the spatial CUs, are the upper and left CUs in the same frame, with the average of the CPs denoted as $\overrightarrow{P}_S$. To investigate the similarity between the CP value of a CU (denoted as $\overrightarrow{P}$) and those of its neighboring CUs, the average Euclidean distances between $\overrightarrow{P}$, and $\overrightarrow{P}_T/\overrightarrow{P}_S$ are calculated and normalized by the maximum distance in mode map (i.e., $130\sqrt{2}$ from Table I) and shown in Table II, where layer indicates the image temporal layer in a GOP.

TABLE II
AVERAGE CP DISTANCES BETWEEN A CU AND ITS TEMPORAL AND SPATIAL CUs

CU size	Temporal layer	Temporal	Spatial
64×64	0	0.2096	0.1081
	1	0.1223	0.0952
	2	0.1109	0.0797
	3	0.0968	0.0747
32×32	0	0.2286	0.1058
	1	0.1404	0.0868
	2	0.1264	0.0735
	3	0.1096	0.0631
16×16	0	0.2380	0.0751
	1	0.1493	0.0609
	2	0.1360	0.0532
	3	0.1152	0.0425
8×8	0	0.2401	0.0465
	1	0.1501	0.0355
	2	0.1358	0.0307
	3	0.1146	0.0245

In Table II, seven 416 × 240 sequences (*BasketballPass*, *BlowingBubbles*, *BQSquare*, *Flowervase*, *Keiba*, *Mobisode2* and *RaceHorses*), five 832 × 480 sequences (*Flowervase*, *Keiba*, *Mobisode*, *PartyScene* and *RaceHorses*) and five 1280 × 720 sequences (*FourPeople*, *Johnny*, *KristenAndSara*, *SlideEditing* and *SlideShow*) are tested with Quantizing parameters (Qps) from 10 to 40. The results of Table II demonstrate that the neighboring CUs have similar CP values and thus neighboring prediction in mode space is justified. To further reduce the prediction error for various scenes, we use a linear combination of $\overrightarrow{P}_T$ and $\overrightarrow{P}_S$, as

$$\overrightarrow{P}^* = (1 - w_S) \cdot \overrightarrow{P}_T + w_S \cdot \overrightarrow{P}_S, \tag{3}$$

where w_S is a weight factor, which is set to be 0 if spatial CUs are not available; otherwise, w_S is set to be adaptive in consequent frames. Let t be the current frame index and n be a predefined number of frames that are preceding the current frame in coding order, then the current weight, $w_S(t)$, can be derived as

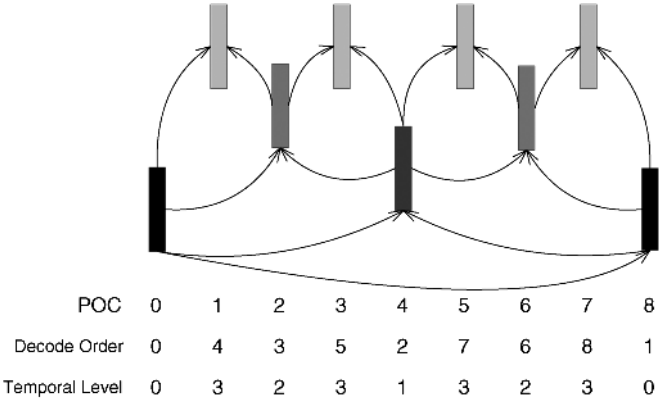
$$w_S(t) = \arg\min_{w_S} \left\{ \frac{1}{n} \sum_{i=t-n}^{t-1} \left\| \overrightarrow{P}(i) - \overrightarrow{P}^*(i) \right\|^2 \right\}. \tag{4}$$

Substitute (3) into (4) and follow the derivations in [16], the current weight can be determined as

$$\begin{aligned} w'_S(t) &= \begin{cases} w_S(t-1) & \text{if } \overrightarrow{P}_S(t-1) = \overrightarrow{P}_T(t-1), \\ (1-\phi)\cdot w_S(t-1) & \\ \quad +\phi\cdot\psi & \text{otherwise,} \end{cases} & (5) \\ w_S(t) &= \min\left\{1, \max\left\{0, w'_S(t)\right\}\right\}, & (6) \end{aligned}$$

where ϕ is an adaptive rate, which can be approximated by a constant with a typical value 0.1, $w_s(0) = 0.8$, and

$$\psi = \frac{\left[\overrightarrow{P}(t-1) - \overrightarrow{P}_T(t-1)\right] \cdot \left[\overrightarrow{P}_S(t-1) - \overrightarrow{P}_T(t-1)\right]}{\left\| \overrightarrow{P}_S(t-1) - \overrightarrow{P}_T(t-1) \right\|^2}. \tag{7}$$



POC	0	1	2	3	4	5	6	7	8
Decode Order	0	4	3	5	2	7	6	8	1
Temporal Level	0	3	2	3	1	3	2	3	0

Fig. 4. Prediction structure of a GOP of size 8.

D. The Overall Mode Selection Scheme on CU Level

Finally, with a complexity factor θ, the flexible coding mode selection algorithm on CU level is summarized as follows. To avoid collision to AMP mode skipping algorithm in HM8.0 [1], these AMP modes are not taken into account in Step 3.

Step 1) Set the initial weight $w_S(0)$ if the current CU belongs to the first frame. If the current frame is an intra frame, check all intra modes and go to the next CU; otherwise go to Step 2.

Step 2) Derive $\overrightarrow{P}_T$ and $\overrightarrow{P}_S$ from the temporal and spatial CUs, respectively, using (2); obtain the value of CP* (i.e., $\overrightarrow{P}^*$) using (3). Go to Step 3.

Step 3) Determine r_{min} and r_{max} with EPs of mode partitions 2N × 2N, 2N × N, N × 2N and N × N, and deduce the search radius r using complexity factor θ and (1). Go to Step 4.

Step 4) Select the candidate mode partitions (including AMP modes) with Euclidean distances to CP* less than r and test all these modes. Go to Step 5.

Step 5) Derive the value of CP (i.e., $\overrightarrow{P}$) using (2) and update w_S using ((5), (6), (7)). If N × N is a candidate mode with CU size larger than 8 × 8, split CU into smaller sizes and repeat Steps 2–5 for all sub-CUs; otherwise go to the next CU.

III. COMPLEXITY ALLOCATION FOR OPTIMAL RD PERFORMANCE

To maximize the overall RD performance with one global complexity factor, we develop a complexity allocation scheme in a GOP. In HEVC, the hierarchical prediction structure is supported, similar to temporal scalability in H.264 Scalable Video Coding (SVC) [38]. An example of a GOP is given in Fig. 4, where the GOP size is 8 and POC denotes the Picture Order Count. In a hierarchical GOP, a frame can only be predicted with frames from the same or lower temporal level.

A. RD Dependency Between Frames

To exploit how to maximize the overall RD performance, the RD dependency between different frames in a GOP is studied first. In our work, the relationship between RD costs of an frame and its nearest reference frame, such as POC2 and POC4 in Fig. 4, is investigated. We change the Qp of lower temporal layer

TABLE III
CODING PARAMETERS USED IN FIG. 5

Index	Sequence	Qp	POCs	Layers
G.1	*BlowingBubbles*(416×240)	30	8→4	0→1
G.2	*BlowingBubbles*(416×240)	30	4→6	1→2
G.3	*BlowingBubbles*(416×240)	30	2→1	2→3
G.4	*BlowingBubbles*(416×240)	20	8→4	0→1
G.5	*BlowingBubbles*(416×240)	30	16→20	0→1
G.6	*Flowervase*(416×240)	30	8→4	0→1
G.7	*Flowervase*(832×480)	30	8→4	0→1
G.8	*FourPeople*(1280×720)	30	8→4	0→1

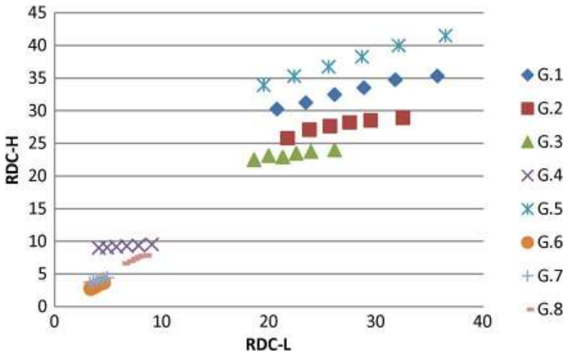


Fig. 5. RD cost dependency in a hierarchical GOP.

and observe how the average RD costs of frames in lower and higher layers are changed. Different sequences with various resolutions, Qps, POCs, and temporal layers are studied, as shown in Table III, where → indicates the prediction order, e.g., POCs 4 → 6 denotes the prediction from POC4 to POC6, and temporal layers 2 → 3 represents the prediction from layer 2 to layer 3.

The RD cost dependencies of G.1 to G.8 in Table III are summarized in Fig. 5, where RDC-L and RDC-H represent the RD costs of the lower and higher layers, respectively. From the figure, there exists an approximately linear relationship between RDC-L and RDC-H, as

$$\Delta C_{l+1} = \gamma \cdot \Delta C_l, \quad l = 0, 1, \ldots, l_{max} - 1, \tag{8}$$

where ΔC is the increase in RD cost; l and l_{max} are the indices of the current and maximum temporal layers, respectively; γ is a linear coefficient, typically between 0.1 and 0.5. In this work, we set $\gamma = 0.3$ based on exhaustive experiments.

The linear relationship illustrated by (8) is consistent with the observation of [39], in which a linear relationship was observed between distortions (D) of a frame and its nearest reference in HBP structure of H.264; besides, it was reported that there is no explicit dependency between the corresponding bit rates (R). Therefore, we can draw the conclusion that there exists an approximately linear relationship between RD costs of a frame and its nearest reference in HBP structure of H.264. In this work, this linear relationship is also observed for HBP structure of HEVC.

In a hierarchical GOP, each frame in layer l can be the nearest reference of two layer $l+1$ frames, as shown in Fig. 4. Follow the derivations in [39], the total RD cost increase in a GOP due to ΔC_l can be expressed as

$$\Delta C_{tot,l} = (1 + 2\gamma)^{l_{max} - l} \cdot \Delta C_l. \tag{9}$$

Especially, frames in temporal layer $l = 0$ can be used as reference frames of the consequent frames in the same layer, and thus,

$$\begin{aligned}\Delta C_{tot,0} &= (1 + 2\gamma)^{l_{max}} \sum_{k=0}^{M} \gamma^k \cdot \Delta C_0 \\ &= \frac{1 - \gamma^{M+1}}{1 - \gamma} (1 + 2\gamma)^{l_{max}} \cdot \Delta C_0, \end{aligned} \tag{10}$$

where M is the number of consequent frames with $l = 0$. Considering M is usually large and γ is a small value, $\gamma^{M+1} \approx 0$ (e.g., $\gamma = 0.3$, $M = 8$, $\gamma^{M+1} = 0.00002 \approx 0$), we set

$$\Delta C_{tot,0} = \frac{1}{1 - \gamma} (1 + 2\gamma)^{l_{max}} \cdot \Delta C_0. \tag{11}$$

Therefore, to maximize the overall RD performance is identical to minimize the total RD cost, as

$$\min \left\{ \sum_{l=0}^{l_{max}} \alpha_l \cdot \Delta C_l \right\}, \tag{12}$$

where

$$\alpha_l = \begin{cases} \frac{1}{1-\gamma} (1 + 2\gamma)^{l_{max}} & \text{if } l = 0, \\ (1 + 2\gamma)^{l_{max} - l} & \text{otherwise} \end{cases}. \tag{13}$$

B. Complexity Allocation in a GOP

Let Θ and θ_l denote the global complexity factor and complexity factor of temporal level l, respectively. Another linear relationship is observed between C_l and θ_l for all frames in a sequence. In Fig. 6, eight frames are tested with different sequences, resolutions, Qp settings and temporal layers. From this figure, C_l can be approximated by a negative linear function of θ_l except when θ_l is very small. Hence, (12) can be rewritten as $\max\{\sum_{l=0}^{l_{max}} \alpha_l \cdot \Delta\theta_l\}$. Considering $\Delta\theta_l = \theta_l - \Theta$ in the complexity allocation problem, (12) can be further simplified as

$$\max \left\{ \sum_{l=0}^{l_{max}} \alpha_l \cdot \theta_l \right\}. \tag{14}$$

The constraint of the complexity allocation problem is that the average value of θ_l for all images equals Θ:

$$\sum_{l=0}^{l_{max}} \beta_l \cdot \theta_l = \Theta, \tag{15}$$

where β_l denotes the number of frames in layer l normalized by the total number of frames in a GOP,

$$\beta_l = \begin{cases} 2^{-l_{max}} & \text{if } l = 0, \\ 2^{-(l_{max} - l + 1)} & \text{otherwise.} \end{cases} \tag{16}$$

Finally, the complexity allocation problem is summarized as a linear programming problem by

$$\begin{aligned} &\max \left\{ \sum_{l=0}^{l_{\max}} \alpha_l \theta_l \right\}, \\ &s.t. \begin{cases} \sum_{l=0}^{l_{\max}} \beta_l \theta_l = \Theta, \\ \theta_l \in \left[\frac{\Theta}{2,1}\right], l = 0, 1, \ldots, l_{\max}, \end{cases} \end{aligned} \tag{17}$$

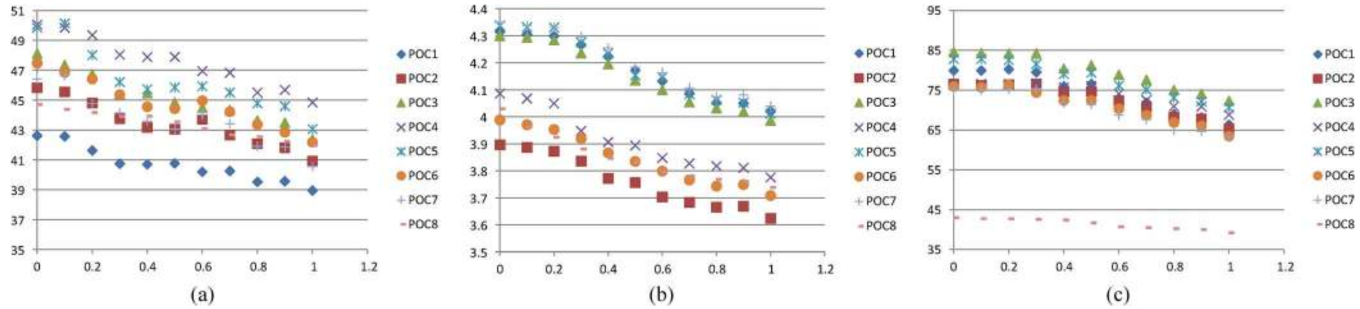


Fig. 6. Relationship between RD costs and complexity factor in a sequence. Horizontal axis: complexity factor. Vertical axis: RD cost. (a) *BlowingBubbles*, 416 × 240, $\mathrm{Qp} = 30$; (b) *FourPeople*, 1280 × 720, $\mathrm{Qp} = 20$; (c) *Tennis*, 1920 × 1080, $\mathrm{Qp} = 30$.

TABLE IV
EXAMPLES OF θ_l IN COMPLEXITY ALLOCATION

Θ	$l_{max} = 1$		$l_{max} = 2$			$l_{max} = 3$				$l_{max} = 4$				
	$l = 0$	$l = 1$	$l = 0$	$l = 1$	$l = 2$	$l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = 4$
0.00	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.05	0.075	0.025	0.125	0.025	0.025	0.225	0.025	0.025	0.025	0.425	0.025	0.025	0.025	0.025
0.10	0.150	0.050	0.250	0.050	0.050	0.450	0.050	0.050	0.050	0.850	0.050	0.050	0.050	0.050
0.15	0.225	0.075	0.375	0.075	0.075	0.675	0.075	0.075	0.075	1.000	0.350	0.075	0.075	0.075
0.20	0.300	0.100	0.500	0.100	0.100	0.900	0.100	0.100	0.100	1.000	0.800	0.100	0.100	0.100
0.25	0.375	0.125	0.625	0.125	0.125	1.000	0.250	0.125	0.125	1.000	1.000	0.250	0.125	0.125
0.30	0.450	0.150	0.750	0.150	0.150	1.000	0.500	0.150	0.150	1.000	1.000	0.500	0.150	0.150
0.35	0.525	0.175	0.875	0.175	0.175	1.000	0.750	0.175	0.175	1.000	1.000	0.750	0.175	0.175
0.40	0.600	0.200	1.000	0.200	0.200	1.000	1.000	0.200	0.200	1.000	1.000	1.000	0.200	0.200
0.45	0.675	0.225	1.000	0.350	0.225	1.000	1.000	0.350	0.225	1.000	1.000	1.000	0.350	0.225
0.50	0.750	0.250	1.000	0.500	0.250	1.000	1.000	0.500	0.250	1.000	1.000	1.000	0.500	0.250
0.55	0.825	0.275	1.000	0.650	0.275	1.000	1.000	0.650	0.275	1.000	1.000	1.000	0.650	0.275
0.60	0.900	0.300	1.000	0.800	0.300	1.000	1.000	0.800	0.300	1.000	1.000	1.000	0.800	0.300
0.65	0.975	0.325	1.000	0.950	0.325	1.000	1.000	0.950	0.325	1.000	1.000	1.000	0.950	0.325
0.70	1.000	0.400	1.000	1.000	0.400	1.000	1.000	1.000	0.400	1.000	1.000	1.000	1.000	0.400
0.75	1.000	0.500	1.000	1.000	0.500	1.000	1.000	1.000	0.500	1.000	1.000	1.000	1.000	0.500
0.80	1.000	0.600	1.000	1.000	0.600	1.000	1.000	1.000	0.600	1.000	1.000	1.000	1.000	0.600
0.85	1.000	0.700	1.000	1.000	0.700	1.000	1.000	1.000	0.700	1.000	1.000	1.000	1.000	0.700
0.90	1.000	0.800	1.000	1.000	0.800	1.000	1.000	1.000	0.800	1.000	1.000	1.000	1.000	0.800
0.95	1.000	0.900	1.000	1.000	0.900	1.000	1.000	1.000	0.900	1.000	1.000	1.000	1.000	0.900
1.00	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000

where α_l and β_l are defined in (13) and (16), respectively. Besides, the lower bound in the constraint is set to be $\Theta/2$ instead of 0 to avoid error propagation due to skipping too many coding partitions.

Eq. (17) can be addressed with Dantzig's simplex method [40], [41], with some examples of θ_ls given in Table IV. In simplex method, the constraint conditions define a polytope as a feasible region; and the algorithm searches all vertices of the polytope to find the optimum solution. Hence, with a limited number of variables (l_{max} is usually small), the number of vertices is also limited. In such a case, the solution of our algorithm is highly robust to the accuracy of RD dependency models. For example, given $l_{max} = 3$, $\Theta = 0.5 \sim 0.9$, we change the linear coefficient γ from 0.3 to 0.4, then the solution of simplex method is also kept the same to the result as given in Table IV. In addition, due to a limited number of variables, (17) consumes a very little computation time and thus its computational complexity is negligible in a video encoder.

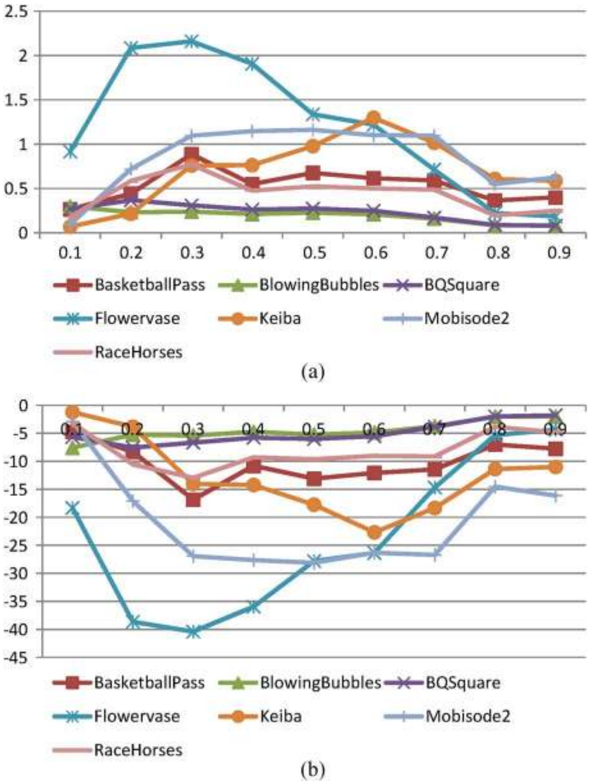
To further justify the RD improvement with complexity allocation, seven 416 × 240 sequences (*BasketballPass*, *BlowingBubbles*, *BQSquare*, *Flowervase*, *Keiba*, *Mobisode2* and *RaceHorses*) are tested with $l_{max} = 3$ and Qps from 10 to 40. The improvement of RD performance is summarized in Fig. 7, in terms of improvement in Bjontegaard average Peak Signal-to-Noise Ratio (BDPSNR) and Bjontegaard average Bit Rate (BDBR) [42]. From the figure, BDPSNR are increased and BDBR are decreased for all sequences, hence, with complexity allocation, the overall RD performance of our mode selection scheme is greatly improved.

TABLE V
SIMULATION ENVIRONMENT

Encoder	HM8.0 [1]			
GOPSize	8 ($l_{max} = 3$)			
MaxCUSize	64×64			
AMP mode	enabled			
FastME	EPZS with HadamardME			
Qps	10 to 40			
Resolution	416×240	832×480	1280×720	1920×1080
SearchRange	32	32	64	96
FramesCoded	97	97	65	65

IV. EXPERIMENTAL RESULTS

To examine the performance of our method, it is implemented on the recent HM reference software, HM8.0 [1], with the configuration parameters summarized in Table V. Five 416 × 240



(a)

(b)

Fig. 7. RD performance improvement by complexity allocation. Horizontal axis: complexity factor Θ. Vertical axis: (a) BDPSNR (dB) increase; (b) BDBR (%) increase.

sequences, five 832 × 480 sequences, three 1280 × 720 sequences and three 1920 × 1080 sequences are tested with the global complexity factor Θ from 0.5 to 0.95. To measure the improvement of our method, three evaluation criteria are used in this section, including Time Saving (TS, %), BDPSNR and BDBR [42].

A. Simulations With Random Access

The coding results for 416 × 240 sequences (*BasketballPass*, *BlowingBubbles*, *BQsquare*, *Flowervase* and *RaceHorses*) are shown in Fig. 8 with random access. From the figure, our method achieves 20% to 60% complexity reduction in terms of TS, with the global complexity factor from 0.5 to 0.95. Nevertheless, the RD performance is less robust for sequences with fast motion or complex texture, such as *BasketballPass* and *RaceHorses*, when a smaller complexity factor Θ is used. Similar conclusions could be drawn from Fig. 9, in which the coding results for 832 × 480 sequences (*BasketballDrill*, *BQMall*, *Keiba*, *Mobisode2* and *PartyScene*) are given with random access, and relatively larger RD losses occur in *BasketballDrill*, *BQMall*, and *Keiba*, when Θ is small. This fact might be due to low correlations between neighboring blocks in fast-motion or complex-texture sequences. In such a case, the candidate mode selection based on neighboring prediction may not work well. In spite of this, our method can still be employed for these sequences, but with a larger global complexity factor.

It can be noticed that, from $\Theta = 0.65$ to $\Theta = 0.7$, there exists a large TS change, compared with the other Θs. The reason is, there is at least one more mode tested when Θ increases from 0.65 to 0.7. As shown in Table IV, with $l_{max} = 3$, the $\theta_{l=2}$ increases from 0.95 to 1, which will result in one more mode tested (i.e., the mode with $r = r_{max}$), based on (1). Another fact is, there exists RD performance loss in the sequence *BQMall* when the global complexity factor Θ increases from 0.55 to 0.6. The reason might be, due to error propagation, to check more coding modes will not surely increase the overall RD performance. In our method, the coding modes of a CU are predicted from neighboring CUs. Hence, there exists a probability that the mode selection error in a CU may cause RD improvement in the consequent CUs. Despite these facts, our method still works well because we can still use Θ to control the global complexity and thus result in different complexity reductions and RD performances to satisfy different user requirements.

In Figs. 10 and 11, the coding results for 1280 × 720 sequences (*FourPeople*, *Johnny* and *KristenAndSara*) and 1920 × 1080 sequences (*BasketballDrive*, *BQTerrace* and *Cactus*) are shown respectively, with random access. Compared with results of low resolution sequences in Figs. 8 and 9, more computational complexity is saved and the coding RD performance is more robust, even for fast-motion sequence *BasketballDrive* and complex-textured sequence *Cactus*. The reason may be that, in sequences with larger video resolutions, the neighboring CUs are more likely to have the same motion or texture and thus neighboring prediction works better. As a result, our method is more preferable to High Definition (HD) sequences, which is the main target of state-of-the-art video coding standard.

Finally, the average coding results for sequences with different resolutions are summarized in Table VI. From the table and Figs. 8, 9, 10 and 11, our method can achieve significantly more complexity reduction, on top of the afore-mentioned fast mode selection algorithms enabled in the HM reference software. For fast-motion or complex-texture sequences with low resolution, the global complexity factor Θ is suggested to be 0.7 to 0.8, which can result in about 30% complexity reduction in terms of TS, with negligible loss in RD performance; and for the other sequences, especially for sequences with HD or higher resolutions, Θ can be set to be 0.5 or less, which can achieve 50% or more computational complexity reduction, and meanwhile the RD performance is almost intact as in the original encoder.

B. Comparison With Low Delay and HM Integrated Algorithms

To justify its efficiency, our flexible method is also tested with HEVC low delay configuration. Three sequences of different resolutions, including *BlowingBubbles* (416 × 240), *Mobisode2* (832 × 480) and *Johnny* (1280 × 720), are tested with the global complexity factor $\Theta = 0.5 \sim 0.95$. The coding results are summarized in Fig. 12, in terms of TS, BDPSNR and BR. Compare with random access results in Figs. 8, 9 and 10, the simulation results of low delay case can also achieve similar coding performances with a flexible mode selection, which justifies the effectiveness and robustness of our flexible method.

In the HEVC encoder, our flexible method is implemented on top of several mode selection algorithms, including CBF based early termination (CbfFastMode), early CU termination (EarlyCU) and early Skip mode decision (EarlySkip), which have

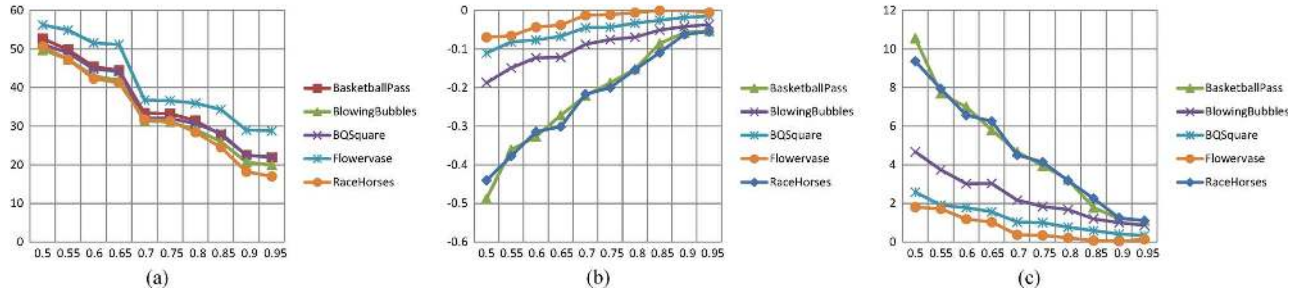


Fig. 8. Summary of coding results with 416 × 240 sequences and random access. Horizontal axis: complexity factor Θ. Vertical axis: (a) TS (%); (b) BDPSNR (dB); (c) BDBR (%).

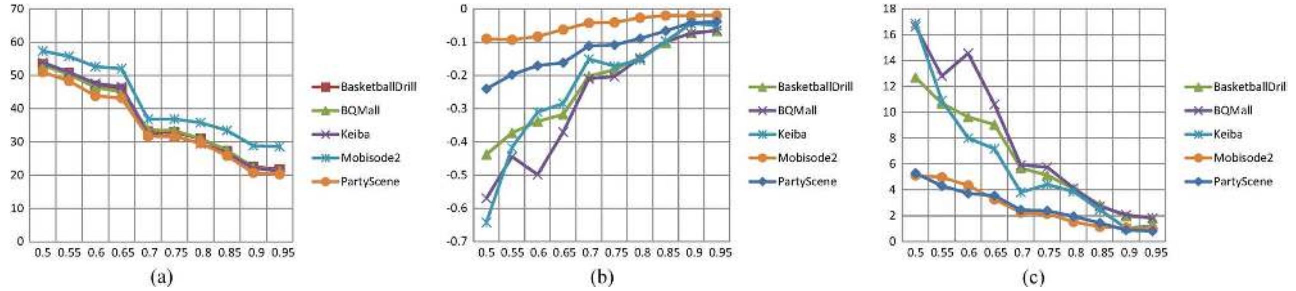


Fig. 9. Summary of coding results with 832 × 480 sequences and random access. Horizontal axis: complexity factor Θ. Vertical axis: (a) TS (%); (b) BDPSNR (dB); (c) BDBR (%).

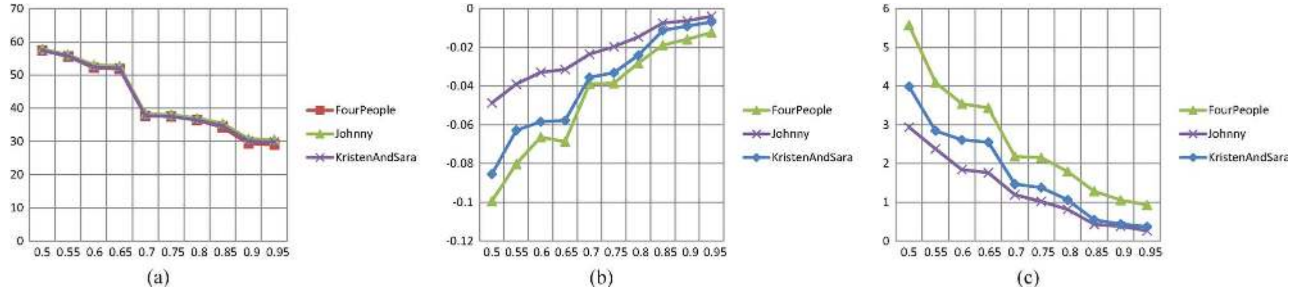


Fig. 10. Summary of coding results with 1280 × 720 sequences and random access. Horizontal axis: complexity factor Θ. Vertical axis: (a) TS (%); (b) BDPSNR (dB); (c) BDBR (%).

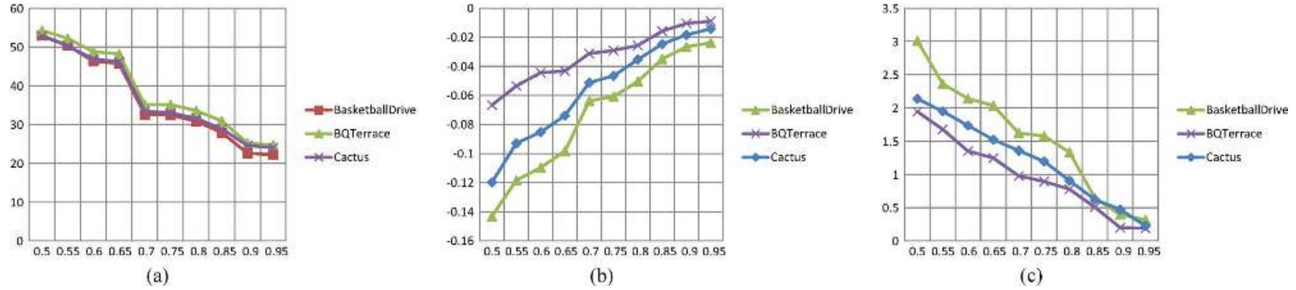


Fig. 11. Summary of coding results with 1920 × 1080 sequences and random access. Horizontal axis: complexity factor Θ. Vertical axis: (a) TS (%); (b) BDPSNR (dB); (c) BDBR (%).

been integrated in HM 8.0. The flexible method is designed in the sense that it can be either used individually or combined with these integrated algorithms for further more complexity reduction. To compare our flexible method with these aforementioned algorithms (as well as the combination of the three algorithms, Combine3), we give the RD performances of these approaches with a similar TS, in Table VII. From the table, our flexible method can achieve similar or at least not much worse coding performances than these integrated algorithms, but meanwhile support the flexibility in computational complexity reduction, so that it can satisfy different user requirements.

In our work, the user requirement is expressed as a global complexity factor Θ. On the other hand, the TS behaves as a monotonically decreasing function of Θ, from Figs. 8, 9, 10, 11 and 12. Therefore, if the user requirement is represented by TS, we can still find a function mapping from TS to Θ, and then employ our flexible method to reduce the computational complexity to satisfy the user requirement.

TABLE VI
AVERAGE CODING RESULTS FOR RANDOM ACCESS

Θ	416×240			832×480			1280×720			1920×1080		
	TS	BDPSNR	BDBR	TS	BDPSNR	BDBR	TS	BDPSNR	BDBR	TS	BDPSNR	BDBR
0.50	52.14	-0.259	5.80	53.72	-0.396	11.33	57.56	-0.078	4.17	53.35	-0.110	2.36
0.55	49.67	-0.207	4.60	51.21	-0.305	8.74	55.83	-0.061	3.11	50.98	-0.088	2.00
0.60	45.45	-0.177	3.91	47.49	-0.28	8.06	52.56	-0.053	2.67	47.33	-0.080	1.74
0.65	44.61	-0.160	3.54	46.58	-0.239	6.73	52.21	-0.053	2.59	46.73	-0.072	1.60
0.70	33.08	-0.116	2.55	33.49	-0.143	4.02	37.95	-0.033	1.62	33.75	-0.049	1.32
0.75	32.88	-0.103	2.26	33.28	-0.142	3.98	37.72	-0.030	1.52	33.54	-0.045	1.22
0.80	31.15	-0.083	1.82	31.39	-0.113	3.12	36.62	-0.022	1.22	32.03	-0.037	1.01
0.85	28.18	-0.054	1.18	28.09	-0.077	2.13	34.80	-0.013	0.76	29.24	-0.025	0.59
0.90	22.57	-0.036	0.78	23.30	-0.050	1.41	30.10	-0.010	0.63	24.14	-0.018	0.36
0.95	21.95	-0.033	0.71	22.64	-0.048	1.34	29.81	-0.008	0.52	23.68	-0.016	0.24

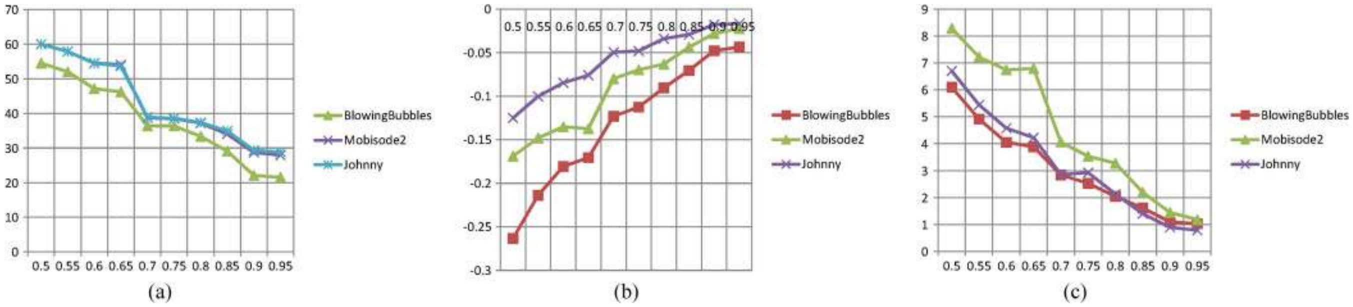


Fig. 12. Summary of coding results with low delay. Horizontal axis: complexity factor Θ. Vertical axis: (a) TS (%); (b) BDPSNR (dB); (c) BDBR (%).

TABLE VII
COMPARISON OF HM INTEGRATED MODE SELECTION ALGORITHMS AND OUR FLEXIBLE METHOD

Algorithms	*BlowingBubbles*			*Mobisode2*			*Johnny*		
	TS	BDPSNR	BDBR	TS	BDPSNR	BDBR	TS	BDPSNR	BDBR
CbfFastMode	25.75	-0.023	0.54	37.95	-0.010	0.57	40.70	-0.006	0.09
FlexibleMethod	26.04	-0.051	1.20	36.85	-0.041	1.84	38.38	-0.024	1.20
EarlyCU	20.88	-0.014	0.34	41.54	-0.008	0.33	45.63	-0.005	0.11
FlexibleMethod	20.67	-0.042	1.01	36.85	-0.041	1.84	52.73	-0.031	1.76
EarlySkip	19.89	-0.013	0.30	34.64	0.001	0.03	37.30	-0.001	-0.04
FlexibleMethod	19.94	-0.037	0.87	33.42	-0.020	1.15	37.06	-0.015	0.82
Combine3	30.45	-0.044	1.04	51.97	-0.020	0.87	56.40	-0.012	0.22
FlexibleMethod	29.15	-0.069	1.68	52.04	-0.062	3.27	56.13	-0.039	2.38

V. CONCLUSION

In this paper, a novel mode selection and complexity allocation method is presented for HEVC. Based on mode mapping, a flexible coding mode selection scheme is designed and implemented on CU level; and based on linear programming, a complexity allocation scheme is developed to maximize the overall RD performance given a limited number of candidate modes. The experiments demonstrate the effectiveness and robustness of our method, which can satisfy different requirements with a user defined global complexity factor.

REFERENCES

[1] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-J1003, ITU-T/ISO/IEC, Jul. 2012.

[2] G. J. Sullivan, J. R. 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.

[3] 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.

[4] *Advanced Video Coding for Generic Audiovisual Services*, ISO/IEC 14496-10:2005(E) ITU-T Rec. H.264 (E), Mar. 2005.

[5] W. J. Han, J. Min, and I.-K. Kim *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.

[6] H. Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 187–260, Jun. 1984.

[7] F. Pan, X. Lin, S. Rahardja, and K. P. Lim *et al.*, "Fast mode decision algorithm for intra prediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 813–822, Jul. 2005.

[8] C.-H. Tseng, H.-M. Wang, and J.-F. Yang, "Enhanced intra-4 × 4 mode decision for H.264/AVC coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 8, pp. 1027–1032, Aug. 2006.

[9] J.-C. Wang, J.-F. Wang, J.-F. Yang, and J.-T. Chen, "A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1414–1422, Oct. 2007.

[10] Y.-M. Lee, Y.-T. Sun, and Y. Lin, "SATD-based intra mode decision for H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 3, pp. 463–469, Mar. 2010.

[11] H. Zeng, K.-K. Ma, and C. Cai, "Hierarchical intra mode decision for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 907–912, Jun. 2010.

[12] Y.-H. Huang, T.-S. Ou, and H.-H. Chen, "Fast decision of block size, prediction mode and intra block for H.264 intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 8, pp. 1122–1132, Aug. 2010.

[13] H. Li, Z.-G. Li, and C. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 7, pp. 889–895, Jul. 2006.

[14] Z. Liu, L. Shen, and Z. Zhang, "An efficient intermode decision algorithm based on motion homogeneity for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 128–132, Jan. 2009.

[15] D. Wu, F. Pan, K. P. Lim, and S. 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.

[16] T. Zhao, H. Wang, S. Kwong, and C.-C. J. Kuo, "Fast mode decision based on mode adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 5, pp. 697–705, May 2010.

[17] I. Choi, J. Lee, and B. Jeon, "Fast coding mode selection with rate-distortion optimization for MPEG-4 part-10 AVC/H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 12, pp. 1557–1561, Dec. 2006.

[18] 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.

[19] C. Grecos and M. Y. Yang, "Fast mode prediction for the baseline and main profiles in the H.264 video coding standard," *IEEE Trans. Multimedia*, vol. 8, no. 6, pp. 1125–1134, Dec. 2006.

[20] E. Martinez-Enriquez, A. Jimenez-Moreno, M. Angel-Pellon, and F. Diaz-de-Maria, "A two level classification-based approach to inter mode decision in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1719–1732, Nov. 2011.

[21] H. Wang, S. Kwong, and C. 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] S.-W. Jung, S.-J. Beak, C.-S. Park, and S.-J. Ko, "Fast mode decision using all-zero block detection for fidelity and spatial scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 201–206, Feb. 2010.

[23] C. Kim and C.-C. J. Kuo, "Feature-based intra/inter coding mode selection for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 4, pp. 441–453, Apr. 2007.

[24] C.-S. Park, B.-K. Dan, H. Choi, and S.-J. Ko, "A statistical approach for fast mode decision in scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1915–1920, Dec. 2009.

[25] T. Zhao, S. Kwong, H. Wang, and C.-C. J. Kuo, "H.264/SVC mode decision based on optimal stopping theory," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2607–2618, May 2012.

[26] L. Zhao, L. Zhang, S. Ma, and D. Zhao, "Fast mode decision algorithm for intra prediction in HEVC," in *Proc. Visual Commun. Image Process. (VCIP)'11*, Nov. 2011, pp. 1–4.

[27] J. Kim, J. Yang, H. Lee, and B. Jeon, "Fast intra mode decision of HEVC based on hierarchical structure," in *Proc. Inf., Commun. Signal Process. (ICICIS)'11*, Dec. 2011, pp. 1–4.

[28] W. Jiang, H. Ma, and Y. Chen, "Gradient based fast mode decision algorithm for intra prediction in HEVC," in *Proc. Consumer Electron., Commun. Netw. (CECNet)'12*, Apr. 2012, pp. 1836–1840.

[29] X. Xu, R. Cohen, A. Vetro, and H. Sun, "Predictive coding of intra prediction modes for high efficiency video coding," in *Proc. Picture Coding Symp. (PCS)'12*, May 2012, pp. 457–460.

[30] T. L. da Silva, L. V. Agostini, and L. A. da Silva Cruz, "Fast HEVC intra prediction mode decision based on EDGE direction information," in *Proc. Signal Process. Conf. (EUSIPCO'12)*, Aug. 2012, pp. 1214–1218.

[31] A. S. Motra, A. Gupta, M. Shukla, P. Bansal, and V. Bansal, "Fast intra mode decision for HEVC video encoder," in *Proc. Software, Telecomm. Comput. Netw. (SoftCOM'12)*, Sep. 2012, pp. 1–5.

[32] R. H. Gweon, Y.-L. Lee, and J. Lim, "Early termination of CU encoding to reduce HEVC complexity," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) Document JVTVC-F045, Jul. 2011.

[33] K. Choi, S.-H. Park, and E. S. Jang, "Coding tree pruning based CU early termination," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) Document JVTVC-F092, Jul. 2011.

[34] S.-W. Teng, H.-M. Hang, and Y.-F. Chen, "Fast mode decision algorithm for residual quadtree coding in HEVC," in *Proc. Vis. Commun. Image Process. (VCIP'11)*, Nov. 2011, pp. 1–4.

[35] H. L. Tan, F. Liu, Y. H. Tan, and C. Yeo, "On fast coding tree block and mode decision for high-efficiency video coding (HEVC)," in *Proc. Acoust., Speech, Signal Process. (ICASSP'12)*, Mar. 2012, pp. 825–828.

[36] J. Kim, J. Yang, K. Won, and B. Jeon, "Early determination of mode decision for HEVC," in *Proc. Picture Coding Symp. (PCS)'12*, May 2012, pp. 449–452.

[37] Y. Piao, J. Min, and J. Chen, "Encoder improvement of unified intra prediction," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) Document JCTVC-C207, Oct. 2010.

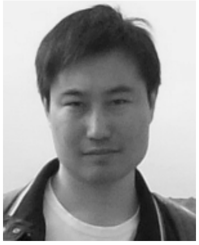
[38] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H. 264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[39] S. Hu, H. Wang, S. Kwong, T. Zhao, and C.-C. J. Kuo, "Rate control optimization for temporal-layer scalable video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 8, pp. 1152–1162, Aug. 2011.

[40] G. B. Dantzig and M. N. Thapa, *Linear Programming 1: Introduction*. New York, NY, USA: Springer-Verlag, 1997.

[41] G. B. Dantzig and M. N. Thapa, *Linear Programming 2: Theory and Extensions*. New York, NY, USA: Springer-Verlag, 2003.

[42] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," Austin, TX, USA, Doc. VCEG-M33, Apr. 2001.



Tiesong Zhao (M'12) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 2006, and the Ph.D. degree in computer science from City University of Hong Kong, Hong Kong SAR, China, in 2011. He is currently working as a Postdoctoral Fellow at the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests include video coding, perceptual video coding and image/video quality assessment.

Zhou Wang (S'97–A'01–M'02) received the Ph.D. degree in Electrical and Computer Engineering from The University of Texas at Austin in 2001. He is currently an Associate Professor in the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include image processing, coding, and quality assessment; computational vision and pattern analysis; multimedia communications; and biomedical signal processing. He has more than 100 publications in these fields with over 15,000 citations (Google Scholar). Dr. Wang has served as a member of IEEE Multimedia Signal Processing Technical Committee (2013–2015), an Associate Editor of IEEE Transactions on Image Processing (2009–present), IEEE Signal Processing Letters (2006–2010), and Pattern Recognition (2006–present), and a Guest Editor of IEEE Journal of Selected Topics in Signal Processing (2007–2009 and 2013–present), EURASIP Journal of Image and Video Processing (2009–2010), and Signal, Image and Video Processing (2011–2013). He was a recipient of 2009 IEEE Signal Processing Society Best Paper Award, ICIP 2008 IBM Best Student Paper Award (as senior author), and 2009 Ontario Early Researcher Award.

Sam Kwong (SM'04) received his B.Sc. and M.Sc. degrees in electrical engineering from the State University of New York at Buffalo, USA and University of Waterloo, Canada, in 1983 and 1985, respectively. He later obtained his Ph.D. degree from the University of Hagen, Germany in 1996. From 1985 to 1987, he was a diagnostic engineer with the Control Data Canada. He later joined the Bell Northern Research Canada, Oattawa, Canada as a member of scientific staff and he joined the City University of Hong Kong as a lecturer at the Department of Electronic Engineering in 1990. He also serve as the Associate Editor for the IEEE Transactions on Industrial Electronics and IEEE Transactions on Industrial Informatics. He is currently a Professor at the Department of Computer Science, City University of Hong Kong, Hong Kong, China. His research interests are video coding, evolutionary algorithms, speech processing and recognition, and digital watermarking.