Speed Up HEVC Encoder by Precoding with H.264

Fangshun Mu1[*†], Li Song[*†], Zhenyi Luo[†◊], Xiangwen Wang[†◊], Xiaokang Yang[*†],

[*]Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China
[†]Shanghai Key Laboratory of Digital Media Processing and Transmissions, Shanghai, China
[◊]School of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, China
{marcusmu, song_li, xkyang@sjtu.edu.cn, wxw21st@gmail.com, lzy@shiep.edu.cn}

Abstract— **The new video coding standard, High Efficiency Video Coding (HEVC), was developed to replace H.264/AVC and achieves high efficiency by introducing a new coding structure in adoption of coding unit (CU), prediction unit (PU) and transform unit (TU). However, it also imposes great computation burden on the mode decision of encoders. In this paper, a conceptual HEVC encoder architecture with cascaded H.264/AVC encoder is proposed to speed up the CTU splitting process of HEVC encoder. First, the input frame is precoded with an efficient H.264/AVC encoder to gather the partition and prediction information of H.264/AVC macroblocks (MBs). Then based on the gathered information, CU and PU modes are carefully screened during HEVC encoding. Two schemes are proposed to reduce the CU and PU candidate modes to accelerate the Coding Tree Unit (CTU) splitting process. At last, the HEVC encoder tries all combinations of candidate CU and PU mode to determine the optimal CU, PU and TU mode by computing the rate-distortion (R-D) cost. The experimental results on highly-optimized open source codecs - x264 and x265 show that the proposed framework can effectively reduce encoding time with negligible loss in terms of PSNR and bitrate.**

I. INTRODUCTION

The H.265/MPEG-HEVC (High Efficiency Video Coding) is the latest video coding standard of the ITU-T Video Coding Experts Group (ITU-T Q.6/SG 16) and ISO/IEC Moving Picture Experts Group (ISO/IEC JTC 1/SC 29/WG 11) [1]. HEVC aims at achieving significant compression efficiency and roughly doubles the rate-distortion (RD) compression performance of H.264/AVC. The main scheme of HEVC is still based on the well-known block-based hybrid coding scheme (inter-/intra prediction and 2D transform coding) used by H.264/AVC and also employs many new coding tools to improve encoder performance [2]. Among them, the hierarchical quad-tree-structure based block partition is one of the most important changes with dramatic impact on efficiency and complexity. The quad-tree coding block consists of coding unit (CU), partition unit (PU) and transform unit (TU). In order to obtain high coding efficiency, HEVC encoders have to try many CU, PU and TU modes to select the best mode combination in the rate-distortion sense. This "try all and select the best" philosophy is optimal in deciding the CTU mode and also used in H.264 to decide the optimal block size for one Macroblock (MB). Nevertheless, this optimal decision is achieved at the expense of high computational complexity [4]. In HEVC, this problem becomes critically severe because as many as 85 CU and 11 PU modes need to be examined to find the optimal CTU mode. However, the total number of candidate MB types is just 19 for each MB and that just is 1/3 when compared with HEVC.

To overcome this problem, many researches have been devoted to fast HEVC encoding through reducing the candidate CU and PU modes within the HEVC coding structure. Existing algorithms usually predict the best CU and PU modes based on the available information in HEVC encoders themselves. In [5], Liquan Shen et al. tried to predict the maximal and minimal CU depth levels by using spatial neighboring treeblocks (left, upper and top left). In [6], Younhee Kim et al. proposed a fast prediction method based on Rate Distortion cost estimation. In [7], Xiaolin Shen et al. proposed a CU size selection algorithm by trying to predict the CU size based on SVM, which imposed additional computational complexity on the encoder with 5 selected features.

To our knowledge, few fast HEVC encoding algorithms have been proposed by adopting precoders. In fact, the coding correlations between H.264/AVC MBs and HEVC CTUs have been explored first in the H.264/AVC-to-HEVC transcoding field. Tong Shen et al [8] proposed a transcoding algorithm for multi-core processors implementing Wave front Parallel Processing (WPP) and SIMD acceleration, along with expedited motion estimation (ME) and mode decision (MD) by utilizing information extracted from the input H.264/AVC stream. Peiyin Xing et al. [2] present a coding unit (CU) classification based H.264/AVC-to-H.265/HEVC transcoding method which employs decoded motion vectors and the modeled background frame to divide the decoded data into background, foreground and hybrid CUs. Then CUs of different categories were transcoded in different ways to reduce the complexity. Peixoto et al. [9] presented a transcoding solution that uses machine learning techniques in order to map H.264/AVC macroblock into HEVC CUs.

In this paper, a novel fast HEVC encoding framework using cascaded H.264/AVC precoding is proposed. First, the input frame is precoded with a highly optimized H.264/AVC encoder to obtain helpful encoding information. Then based on obtained information CU and PU modes are screened to enable fast HEVC encoding. At last we use H.265/HEVC

This work was supported by National 863 project (2012AA011703), National Key Technology R&D Program of China (2013BAH53F04), NSFC (61221001, 61271221), the 111 Project (B07022) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

encoder to get the optimal combinations of CU and PU mode via evaluating the RD cost of candidate CU and PU modes.

The rest of this paper is organized as follows. In section II, we briefly review the CU and PU selection process in H.265/HEVC and investigate the relationships of partition and prediction modes between H.264/AVC and H.265/HEVC. In section III, the proposed cascaded structure of HEVC encoder is fully described and new CU and PU candidate selection algorithms are introduced. Experimental results are shown in Section IV and Section V concludes this paper.

II. Problem Analysis

In HEVC, some new coding tools, such as Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU), are adopted to capture the diversity of video content and greatly improve the compression performance. But such new tools remarkably increase the coding complexity as well. The CU is similar to the concept of macroblock (MB) in H.264/AVC and extends it with size varies from 8x8 to 64x64. The size of Prediction Unit (PU) and Transform Unit (TU) depends on the size of CU. PU is the basic unit of the prediction process and can be symmetric or asymmetric. TU is the basic unit of the transformation and quantization process. The size of square-shaped TU ranges from 4x4 to 32x32. It must be smaller than or equal to the size of CU but can be larger than the size of PU. The relationship between CU, PU and TU is demonstrated in Fig. 1 [5].

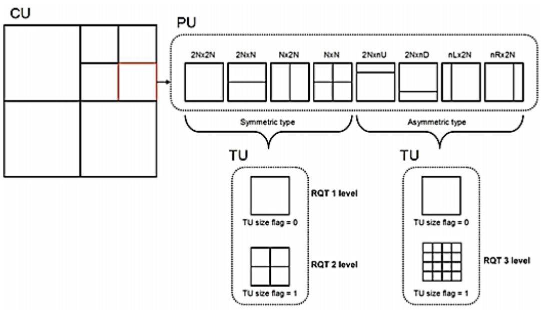


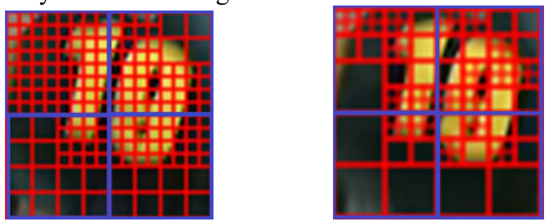
Fig.1 The relationship between CU, PU and TU

To decide the best CU and PU modes of a CTU, the encoder typically tries all the possible combinations of CU and PU modes by computing the rate-distortion (R-D) cost and the modes with minimal R-D costs are selected. In H.264/AVC, similar process is also conducted by the encoder to obtain the optimal partition and prediction mode of each MB.

A. *Analysis of CU Depth and Partition mode of MB*

In H.264/AVC, each picture is partitioned into 16x16 macroblocks, and each macroblock can be further split into smaller blocks (as small as 4x4) for prediction. In H.265/HEVC, a more flexible coding tree structure enables large CU sizes to encode large smooth regions in HD and beyond [10]. The encoding result of the two encoders have strong correlations due to the similar coding structure shared by H.265/HEVC and H.264/AVC encoder. In general, large CU or MB is more likely to work better for smooth regions of a picture since prediction can be more accurate and further splitting into smaller size brings no much prediction improvement but increases side information. Whereas smaller CU or MB partition is preferred for objects with flexible motion since the CU can be hardly predicted accurately for such cases [13].

To investigate the relationship between CU depth and partition mode of MB, we perform both H.265/HEVC and H.264/AVC encoding on the on the sequence *BasketballDrill*. A representative result of optimal CTU and MB partition can be found in Fig. 2, from which we can find that CTU partitions and MB partitions have strong correlations and we can predict the CU depth range based on the MB partitions obtained by H.264 encoding.



(a) Optimal MB partition (b) Optimal CTU Partition

Fig. 2 A representative result of HEVC and H.264/AVC encoding

B. *Analysis of PU Mode and Prediction mode of MB*

In H.265/HEVC, prediction unit (PU), which is similar to MBs' prediction mode of H.264/AVC in spirit, is the basic unit used for carrying the information related to the prediction process. In general, PU and MBs' prediction mode are not restricted to being square in shape in order to facilitate partitioning that matches the boundaries of real objects in the image [5]. In H.265/HEVC, each CU has eight candidate patterns of PU, including symmetric partitions of 2N×2N, N×N, 2N×N, N×2N and asymmetric motion partitions (AMP) of 2NxnU, 2NxnD, nRx2N and nLx2N.

To analyze the relationship between PU mode (16x16) and prediction modes of MB, we perform H.264/AVC and H.265/HEVC encoding on *BasketballDrill.* The Fig.3 shows the percentage of the PU mode (16x16, HEVC) of each prediction type of MB (H.264/AVC). As can be seen from Fig. 3, 2Nx2N mode accounts for a large percent for MB no matter what type of prediction mode is encoded. However, AMP accounts for less than 5% in MB, especially in the PSKIP and Intra mode. The percentage of 2NxN mode is relatively higher in the 16x8 prediction type of MB and similar results can be found for Nx2N mode since the texture properties of the specific area decides the prediction mode of H.264/AVC and HEVC/H.265.

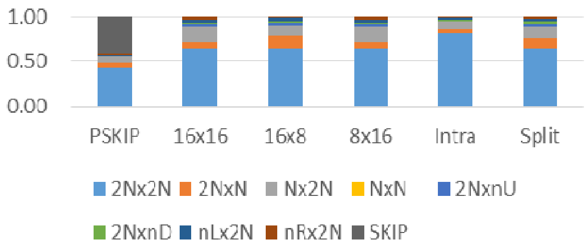


Fig 3. The PU mode distribution of MB's prediction mode

From the above analysis, we can summarize that the partition and prediction modes of CTU and MB have strong correlations and the information obtained from H.264/AVC encoding can be used to screen the CU and PU candidate modes for H.265/HEVC encoding. For CU, we can predict the CU depth range based on the partition information of MBs. For PU, we can map the MBs' prediction modes to PU modes of H.265/HEVC to eliminate the unlikely PU candidates. Both methods aim at reducing the candidate modes of CU and PU to accelerate the CTU splitting process.

III. PROPOSED ENCODER

A. Architecture

In this paper, we propose a conceptual H.265/HEVC encoding framework with cascaded H.264 precoding as shown in Figure 4. During the encoding, we first encode the input frame with a highly optimized H.264/AVC encoder to obtain the partition and prediction results of MBs. Then we screen the CU and PU modes for H.265/HEVC based on the prediction and partition information obtained by H.264/AVC precoding. Two schemes are utilized to reduce the candidate CU and PU modes to accelerate the CTU splitting process. At last, the HEVC encoder tries the remaining combinations of candidate CU and PU modes to determine the optimal CU and PU modes.

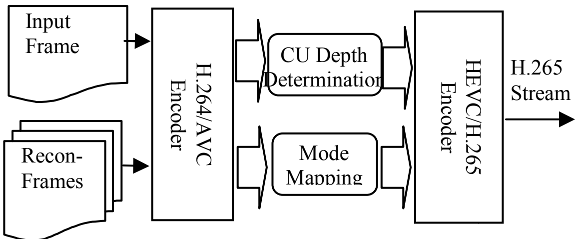
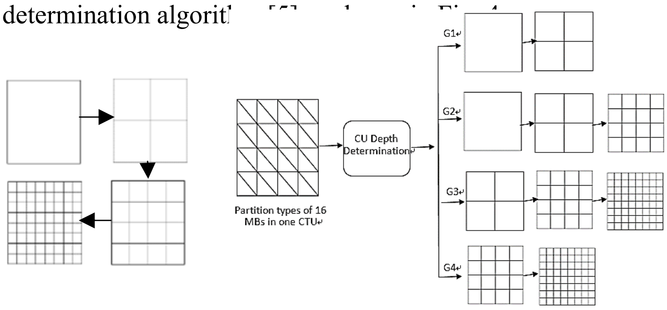


Fig. 4 Proposed Cascaded-encoder Structure

B. CU Depth Range Determination Algorithm

In the standard CTU structure, the maximum CTU depth is fixed and the CTU splitting process always start at depth 0 (64x64) and ends at depth 3(8x8), which means we have to test numerous combinations of CU sizes to get the optimal CU partition. From the analysis in Section II, the partition results of H.265/HEVC and H.264/AVC have strong correlations. If the partition sizes of the majority of MBs lying in a CTU are 16x16 without further split, this area tends to be "flat" or "background" regions and large CUs are more likely to be chosen as optimal CU [11]. And also presumably, small CUs are more likely to be chosen as optimal CU size for the area with MBs split into small blocks which indicates active motion or rich texture in this area. To speed up the CU splitting process, we proposed a CU depth range determination algorithm [5] as shown in Fig. 4.



(a)Original Strategy (b) Proposed Strategy

Fig. 4 CU Depth Range Determination Strategy

To get a rough prediction of whether CTU is complicated or simple, we use the Number of MB partitions (N_{mb}) lying in the CTU to predict the Depth Range (DR) of the CTU. CTU's candidate Depth Range determination algorithm attempts to classify the CU into one of four types (*G0, G1, G2, G3*) [5]. For type *G0* treeblock, all 16 MBs lying the CTU are not split and this area tends to contain motionless or slow-motion content. Thus the current CTU only needs to check the big CU size and no need to further split into small CUs. For type *G4* treeblock, all 16 MBs are split into small blocks, so it would be more efficient if we skip the large CUs and check the small CUs directly. Based on the above analysis, the candidate Depth Range proposed in this paper are summarized in Table 1.

Table 1: Proposed Depth Range for each treeblock type

Type	N_{mb}	Candidate Depth Range	DR [D_{min}, D_{max}]
G0	16	0,1	[0,1]
G1	16~24	0,1,2	[0,2]
G2	24~56	1,2,3	[1,3]
G3	56~64	2,3	[2,3]

C. Mode Mapping Algorithm for PU Selection

The essential tactic of the proposed mode mapping algorithm for PU selection is to reduce the Prediction Unit candidates for H.265/HEVC encoding based on the prediction information of MBs. In this paper, we divide the problem into two parts according to the sizes of the CU, one case is when the size of CU is 32x32 and the other one is when the size of CU is lower or equals to 16x16.

I. PU Mode Mapping in 64x64

For the 64x64 case, we just test Mode SKIP and Mode 2Nx2N because the RD gain in very limited compared with the time-saving according to our experiments.

II. PU Mode Mapping in 32x32

For the 32x32 case, we divide the CU into 4 different types according to the partition types of the 4 MBs lying in the CU and test different kinds of prediction mode to rule out the less-likely PUs to reduce the computational complexity [8].

According to our experiments, the R-D gain for AMP at 32x32 is limited and it's time consuming to calculate their RD cost. So we drop these partitions off to make the problem easy. In this way, we just decide whether to check the remaining six mode partitions or not. They are: Skip, Merge, Inter2Nx2N, InterNx2N, Inter2NxN and Intra2Nx2N. As Skip and Merge are not complex and the cost of not checking them is severe, we will always check these two partitions. Then whether the rest 4 mode partitions will be checked depends on the 4 MBs lying the CU. We depict the four different types of CU which used to decide PU mode mapping method in Fig. 5.

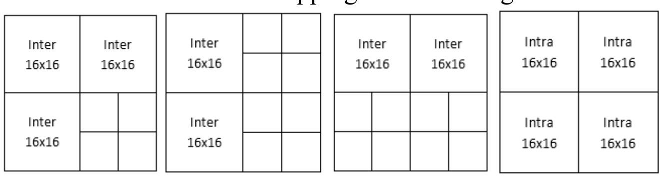


Fig. 5 Explanation of the four different types of CU (32x32) for PU mode mapping

1) Inter2Nx2N is checked when there exists more than 2 MBs using Inter16x16 mode.

2) Inter2Nx2N and InterNx2N are checked when the left two MBs both use Inter16x16 mode or right two MBs both use Inter16x16 mode.

3) Inter2Nx2N and Inter2NxN is checked when the upper two MBs both use Inter16x16 mode or lower two MBs both use Inter16x16 mode.

4) Intra2Nx2N is checked when there exists more than 2 MBs using Intra modes.

5) All 6 modes will be check for all other situations that not listed in above four types.

III. PU Mode Mapping in16x16 and 8x8

Experiments have shown that keeping the exact same prediction modes as the incoming H.264/AVC leads to large rate-distortion loss even if the complexity reduction is large. To balance time reduction and RD performance, a mode mapping algorithm is designed in this paper to predict the Most Probable PU modes for H.265/HEVC. The Most Probable PU modes of H.265/HEVC are mapped from the optimal prediction mode obtained from H.264/AVC encoding. The complete mode mapping relationship [9] between PU modes and MBs' prediction modes for the 16x16 and 8x8 CUs are listed in tables 2 and 3, respectively.

Table 2 PU Mode Mapping in16x16

	CU Size (16x16)	Macroblock Type from H.264/AVC encoder					
		PSKIP	16x16	16x8	8x16	8x8	Intra
PU Candidates for HEVC encoder	SKIP/Merge	√	√	√	√	√	√
	2Nx2N		√	√	√	√	√
	2NxN			√		√	√
	Nx2N				√	√	√
	2NxnU			√		√	√
	2NxND			√		√	√
	nRx2N				√	√	√
	nLx2N				√	√	√
	NxN						√
	Intra						√
	SPLIT					√	√

Table 3 PU Mode Mapping in 8x8

	CU Size (8x8)	Macroblock Type from H.264/AVC encoder					
		PSKIP	16x16	16x8	8x16	8x8	Intra
PU Candidates for HEVC encoder	SKIP/Merge	√	√	√	√	√	√
	2Nx2N		√	√	√	√	√
	2NxN			√		√	√
	Nx2N				√	√	√
	2NxnU			√		√	√
	2NxND			√		√	√
	nRx2N				√	√	√
	nLx2N				√	√	√
	NxN					√	√
	Intra						√

IV. EXPERIMENTAL RESULTS

In this paper, our proposed algorithm isn't verified on H.265/HEVC reference Model (HM) and H.264/AVC reference software Joint Model (JM) since both HM and JM are too slow to be a good benchmark for the practical H.264/AVC and H.265/HEVC encoder respectively. Thus we use x265 instead of HM as our anchor HEVC encoder. x265 is a highly optimized open-source project and free application library which is based on HM cores while doing lots of platform-level optimized for balancing RD performance and encoding speed. Regarding to H.264/AVC encoder, x264 is used instead of JM and x264 is a free software library and application for encoding video streams into the H.264/MPEG-4 AVC compression format and generally accepted as the highest-quality, most efficient H.264/AVC implementation in the world.

To evaluate the performance and quality degradation of the proposed algorithm, we have checked the encoded bits and PSNR. We have defined two parameter, $\Delta PSNR$ and ΔBit, to calculate the quality degradation. The two parameters are calculated as below:

$$\Delta PSNR = PSNR_{pro} - PSNR_{x265} \quad (1)$$

$$\Delta \text{Bit} = \frac{Bit_{x265} - Bit_{pro}}{Bit_{x265}} \times 100\% \quad (2)$$

Where $PSNR_{x265}$, Bit_{x265} and $PSNR_{pro}$, Bit_{pro} denote the PSNR and encoded bits of x265 encoder and our proposed algorithm, respectively.

To evaluate the coding efficiency of our proposed algorithm, Time Saving (ΔT) is calculated and defined as follows:

$$\Delta \text{T} = \frac{T_{x265} - T_{pro}}{T_{x265}} \times 100\% \quad (3)$$

Where T_{x265} and T_{pro} denote the total encoding time of x265 encoder and the proposed encoder, respectively.

We implemented the algorithm under the low delay main configuration and QP is set to 27. X265 encoder with default preset (-medium) is used as HEVC benchmark and x264 encoder with default preset (-medium) as a pre-encoding step to collect MB information. The experiment platform used is Intel ®Core(TM) i7-3770@3.40GHz with dual cores, 4 GB RAM. The experimental results of proposed algorithm conducted on 5 different sets of video clips are shown in Table 4.

In order to evaluate the performance of the proposed algorithm, a large number of experiments have been conducted on HEVC standard test clips. The set of clips contains different levels of motion, texture and different resolutions. It can be seen that the proposed algorithm can achieve up to 16% running-time reduction with an average of 0.008 dB PSNR drop and 0.06% bitrate increase compared with x265 encoder. And we have also performed the proposed algorithm on state-of-the-art HEVC encoder and achieved approximate 20% time reduction. However, the encoding time of H.264/AVC is so heavy that the reduction of HEVC of our proposed method is less than the time added by H.264/AVC. So we choose to use x264 instead of H.264/AVC since x264 is the highest-quality, most efficient H.264/AVC implementation.

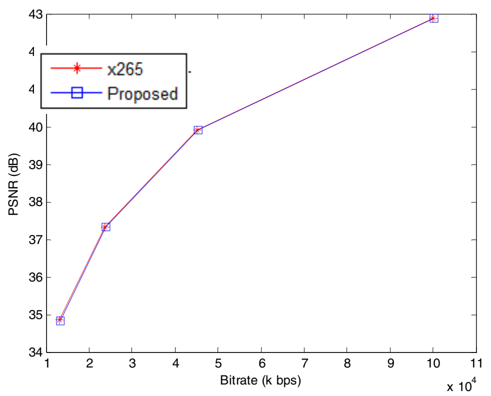
Little research has been conducted on x265 encoder and we don't have many related works for comparison. It is worth to mention that even complexity reduction of our current proposed scheme is not so high, the PSNR loss and bitrate increase is remarkable small compared with other fast Mode Decision algorithm in [5], [6], [7].The outstanding RD performance achieved by proposed algorithm means the

Table 4: PSNR difference, Bitrate difference and Time Saving of the proposed algorithm

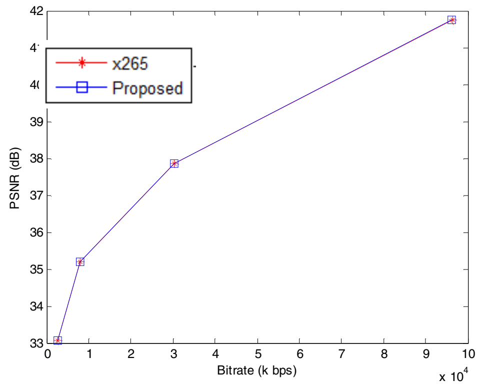
Class	Sequence	TS (%)	PSNR (dB)	Bitrate (%)
A	Traffic	15.80%	-0.011	0.12%
	PeopleOnStreet	9.31%	-0.005	-0.53%
B	ParkScene	13.09%	-0.017	0.03%
	BasketballDrive	15.38%	-0.002	0.82%
	Kimono	9.93%	-0.002	-0.45%
	Cactus	13.17%	-0.007	-0.04%
	BQTerrace	11.65%	-0.010	0.21%
C	PartyScene	9.60%	-0.004	0.09%
	BQMall	14.72%	-0.014	0.46%
	BasketballDrill	12.67%	-0.028	-0.30%
	RaceHorsesC	14.63%	-0.004	-0.29%
D	BQSquare	11.08%	-0.014	0.15%
	BasketballPass	11.31%	-0.004	-0.56%
	BlowinBubbles	8.70%	0.001	-0.02%
	RaceHorses	12.82%	0.000	-0.25%
E	FourPeople	14.36%	-0.009	-0.29%
	Johnny	14.19%	-0.008	-0.13%
	KristenAndSara	19.48%	-0.010	-0.05%
Average		12.88%	-0.008	-0.06%

proposed algorithm is effective in screening the CU and PU mode.

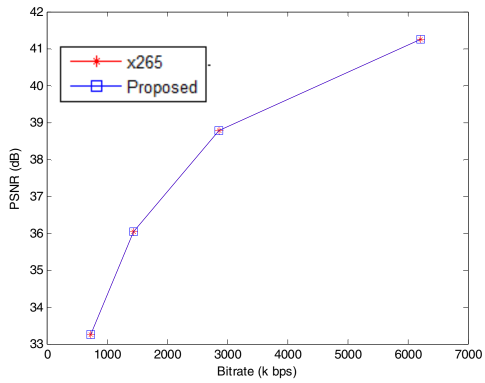
In Fig. 6 (a), (b), (c), (d), (e), the R-D curves of 5 different classes of video with different resolutions are shown. In Fig. 6, the red curves denote the benchmark R-D performance of x265 and blue curves denote the performance achieved by our proposed algorithm. We can find that the quality loss produced by our algorithm is extremely small and nearly can be neglected. From the aspects of R-D performance, it's easier and applicable for our proposed algorithm combined with other fast CU mode selection algorithms.



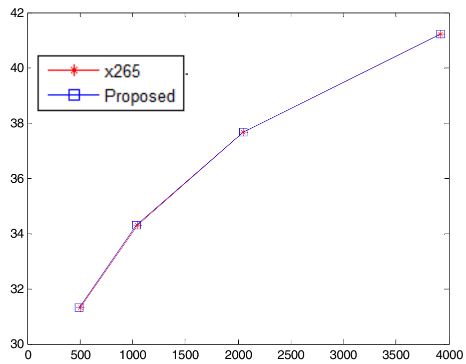
(A) Class A PeopleOnStreet (2560x1600)



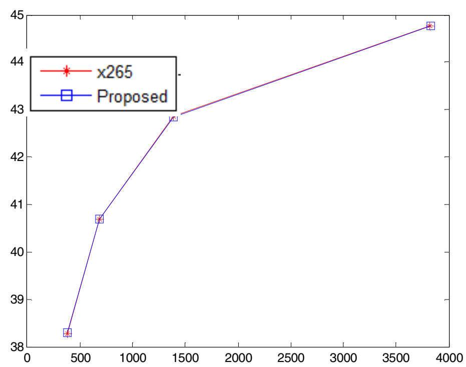
(B) Class B BQTerrace (1920x1080)



(C) Class C BQMall (832x480)



(D) Class E RaceHorces (416x240)



(E) Class E KristenAndSara (1280x720)

Fig. 6 RD Curves of x265 and our proposed algorithm

A. *Additional overhead of x264 encoder*

In this paper, we utilize x264 encoder to preliminary screen the CU and PU modes for HEVC to accelerate the CTU splitting process. Obviously, x264 encoder will impose additional computational complexity on encoder and it's critical for us to investigate the overhead of x264 encoder. Some experiments are conducted to investigate the overhead of x264 encoder and the experimental results shows in Table 5. The total encoding time of x264 is shown in column "Total 264" and the total time to encode sequences with x265 encoder is shown in column "Total 265". As it shown in column "percentage", the computational overheads are not critical and the time of x264 encoder is just less than 1% of that of x265 encoder.

Table 5 Additional Overload of x264 encoder

Sequence	Resolution	Total 264	Total 265	Percentage
PeopleOnStreet	2560x1600	4.285	649.15	0.66%
BQTerrace	1920x1080	1.927	354.62	0.54%
BQMall	832x480	0.331	46.27	0.72%
RaceHorses	416x240	0.201	22.44	0.90%
KristenAndSara	1280x720	0.638	45.90	1.39%

V. Conclusions

In this paper, a conceptual HEVC encoding framework with cascaded H.264/AVC encoder is proposed to speed up the CTU splitting process of H.265/HEVC encoding. The new H.265/HEVC encoding framework provides a new direction for fast CU or PU decision problems in H.265/HEVC. During the encoding process, we first encode the input frame with a highly optimized H.264/AVC encoder to obtain the partition and prediction result of MBs. Then we screen the CU and PU mode of HEVC based on the prediction and partition information obtained by H.264/AVC encoder. Two algorithms utilized to eliminate the CU and PU candidates to accelerate the CTU splitting process. We proposed a CU depth range decision algorithm to classify CU into four different types based on the partition of MBs. Besides, a mode mapping algorithm is proposed to predict the PU mode of HEVC based on the prediction mode of MBs. At last we use H.265/HEVC encoder to get the optimal combinations of CU and PU mode via evaluating the RD cost of candidate CU and PU modes. In particular, the experimental results show that our algorithm can speed up H.265/HEVC encoding with negligible loss in terms of PSNR and bit rate.

As a preliminary but promising result, the works presented in this paper opens up several interesting directions for further work. For instance, we can reuse more information obtained from H.264/AVC encoder, such as motion vectors and DCT coefficients to accelerate the motion estimation process or other time-consuming processes in HEVC. Also, some machine learning techniques could be used to predict the CU and PU mode more accurately. By integrating these techniques into current framework in future, we can expect much gains of speeding up H.265/HEVC encoding.

VI. References

[1] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan and Thomas Wiegand, "High Effiency Video Coding (HEVC) text specification draft 9," JCTVC-K1003_v13, Shangha,CN, Oct. 2012.

[2] Peiyin Xing, Yonghong Tian, Xianguo Zhang, Yaowei Wang and Tiejun Huang, "A coding unit classification based AVC-to-HEVC transcoding with background modeling for surveillance videos," *Visual Communications and Image Processing (VCIP)*, pp.1-6, Nov. 2013.

[3] Joint Video Team (JVT) of ITU-T and ISO/EC JTC 1, Advanced Video Coding: ITU-T Rec. H.264 and ISO/EC14496-10, version 4, January 2005.

[4] Xiangwen Wang, Jun Sun, Yunqiang Liu and Renjie Li, "Fast Mode Decision for H.264 Video Encoder Based on MB Motion Characteristic," *2007 IEEE Int. Conf. on Multimedia and Expo*, pp.372-375, July 2007.

[5] Liquan Shen, Zhi Liu, Xinpeng Zhang, Wenqiang Zhao and Zhaoyang Zhang, "An Effective CU Size Decision Method for HEVC Encoders," *IEEE Trans. On Multimedia*, vol.15, no.2, pp.465-470, Feb. 2013.

[6] Younhee Kim, Jongho Kim, DongSan Jun, Soonheung Jung and Jinsoo Choi, "A Rate-Distortion cost estimation approach to fast intra prediction in Video Coding for Ultra High Definition TV applications," *2012 IEEE Int. Conf. on Consumer Electronics (ICCE)*, pp.164-165, Jan. 2012.

[7] Xiaolin Shen and Lu Yu. "CU splitting early termination based on weighted SVM," *EURASIP Journal on Image and Video Processing*, pp. 1-11, 2013.

[8] Tong Shen, Yao Lu, Ziyu Wen, Linxi Zou, Yucong Chen and Jiangtao Wen, "Ultra-Fast H.264/AVC to HEVC Transcoder," *Data Compression Conf. (DCC)*, 2013 , pp.241-250, March 2013

[9] E. Peixoto, B. Macchiavello, E.M. Hung, A. Zaghetto, T. Shanableh and E. Izquierdo, "An H.264/AVC to HEVC video transcoder based on mode mapping,", *2013 20th IEEE Int. Conf. on Image Processing (ICIP)*, pp.1972-1976, Sept. 2013.

[10] 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?" *Consumer Electronics Magazine, IEEE*, vol.1, no.3, pp.36-46, July 2012.

[11] Fangshun Mu, Li Song, Xiaokang Yang and Zhenyi Luo, "Fast coding unit depth decision for HEVC," *2014 IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, pp.1-6, July 2014.