Learning based fast H.264 to H.265 transcoding

Qingxiong Huangyuan*†, Li Song*†, Yue Ma*†, Rong Xie*†, Zhengyi Luo◊
*Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China
†Future Medianet Innovation Center, Shanghai, China
◊School of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai, China
{hyqx, song_li, mayue2011, rongxie@sjtu.edu.cn, lzy@shiep.edu.cn}

***Abstract*— The newly proposed video coding standard, High Efficiency Video Coding (HEVC), has been widely accepted and adopted by industry and academia due to its better coding efficiency compared with H.264/AVC. While HEVC achieves an increase of about 40% in coding efficiency, its computational complexity has been increased significantly. Given this, a high performance AVC to HEVC transcoder is needed urgently. This paper introduces a learning based fast transcoding algorithm which can speed up the process of CU decision. The stream is first decoded by JM and then important features are extracted. Those features are used as inputs for a machine learning model and the specific CU depth will be obtained. In x265, we skip depths that are not selected and early pruning is used to terminate splitting in advance. The experimental results show that our proposed transcoding algorithm can save up to 41% coding speed compared with original x265 while the BD-BitRate drop 0.078dB on average. The algorithm achieves a good tradeoff between the performance and transcoding speed.**

I. INTRODUCTION

H.264 / AVC is an industry standard for video compression which introduced in 2004. While it now gradually has been accepted and adopted in online domain for content compression during the past decade. Given the shortage of bandwidth, spectrum and storage, the new generation of video compression standard needed to be proposed urgently. High Efficiency Video Coding (HEVC), a new Standard for video compression developed by the ISO and ITU-T, is the successor standard to H.264.It generates huge optimism for imminent need to take growing UHD content for multi platform delivery. It uses flexible partitioning and introduces coding tree units(CTU) but not macroblock (MB) used in H.264. Empirical studies have shown that HEVC can achieve up to approximately 40 % bit rate savings for similar perceived video quality compared to H.264 with high profile[1]. In intra coding of HEVC, 35 directional modes are used for prediction. Parallel processing architecture is used to accelerate the coding speed and enhance the performance.

The H.264 to H.265 transcoding may meet the challenges of balancing the RD performance and transcoding speed. The tradeoff between the two factors is the key to the transcoding framework. In H.264, the macroblock (MB) is used as the basic unit. Flexible partition of macroblock (MB) and subMB are used for motion estimation. While In HEVC, 64x64 CTUs are basic units to perform further splitting. HEVC encoder performs a recursive traversal on the CU quad-tree. When the encoder examines all candidate modes on current depth, the current depth will be split to sub CU until minimum CU size [2]. The encoder compares all RD cost and choose the best CU depth with the minimum RD cost as the final depth. Transcoding framework first decodes the H.264 stream to extract much information and use appropriate features for re-encoding with H.265.

Many previous works have been done to solve the contradiction between RD performance and speed of transcoding. Some meaningful works have been focused on fast CU decision. Shen Liquan et al propose a fast CU size decision and mode decision algorithm for HEVC intra coding, skipping some specific depth levels rarely used in spatially nearby CUs [3]. Dong Zhang et al proposed a power spectrum based rate-distortion optimization (PS-RDO) model, using residual, modes and motion vectors to estimate the best CU. Through reducing the CU and PU partition candidates, the transcoding complexity can be reduced [4]. Fangshun Mu uses a conceptual HEVC encoder architecture with cascaded H.264/AVC encoder to speed up the CTU splitting process of HEVC encoder. Detail information of H.264/AVC macroblocks (MBs) are gathered in order to reduce the CU and PU candidate modes to accelerate the CTU splitting process [5]. Feiyang Zheng et al presented a fast transcoding algorithm based on residual and motion information extracted from H.264 decoder, which results a relatively effective CU and prediction unit (PU) mode decision strategy.

To make the prediction more precise, machine learning is used to choose accurate CU depth. Some early works have been done with this method. Xiaolin Shen transferred the CU splitting problem to a binary classification with the support vector machine (SVM)[6] Luong Pham Van et al propose a fast algorithm based on the early prediction of the partition split-flags in P pictures with machine learning techniques. At each depth, this method helps to figure out whether to split or not.

In this paper, a learning based encoder framework is proposed. First, the input stream is decoded by JM and valuable features are extracted. Then we use these features to calculate the specific depth for every 8x8 CU with a learning based model. At last, meaningless depths are skipped with the method of early skipping and early pruning.

The rest of this paper is organized as follows. In Section II, we briefly introduce the CU partition and our encoder frame work. In Section III, we introduce our detailed structure including features selection, the mapping of MBs to CUs and the skipping and pruning of CU depths. Experiment results are shown in the Section IV and Section V concludes this paper.

II. PROBLEM ANALYSIS

The HD and UHD videos promote the development of H.265/HEVC as higher resolution frames needed to be encoded in a relatively larger coding unit. HEVC adopted CU instead of MB in AVC. In H.265, CUs range from 64x64 to 8x8 and they are basically replacement of MBs and blocks in prior standards. HEVC divide the frames into CTUs and in order to facilitate syntax representation of block hierarchy, three block concepts are introduced: coding unit (CU), prediction unit (PU), and transform unit (TU). And the coding structure can be analyzed as calculating different sizes of CU and PU recursively. Fig.1 shows a typical splitting manner of a CTU. The CU is split in a recursive manner from 64x64 to 8x8. For each size of CU, the RD cost is calculated until all sizes of CU are traversed. Then the CU size with minimum RD cost is adopted and other ways of splitting are abandoned. So the encoder has to try all the possible combinations of CUs and PUs which result computation burden largely increasing.

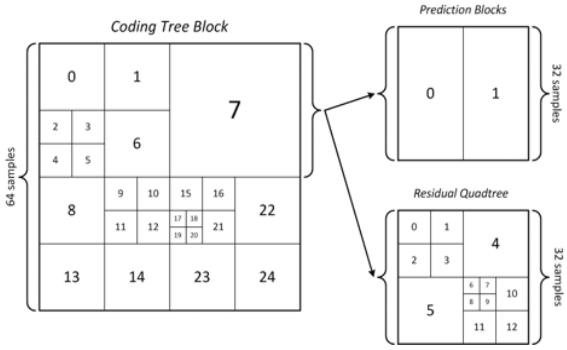
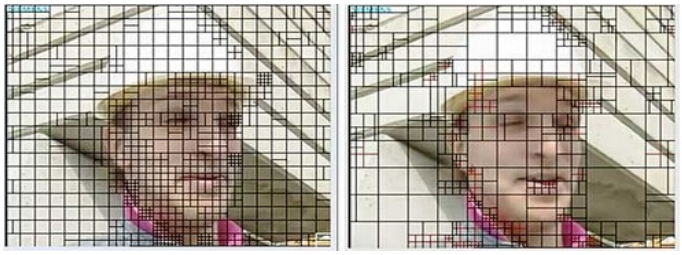


Fig. 1 Splitting of specific CTU

As the coding structure of H.264 is similar to HEVC, detailed information from H.264 stream can be reused to help judge the specific CU and PU modes. Fig. 2 illustrated the similarity between H.264 and HEVC.

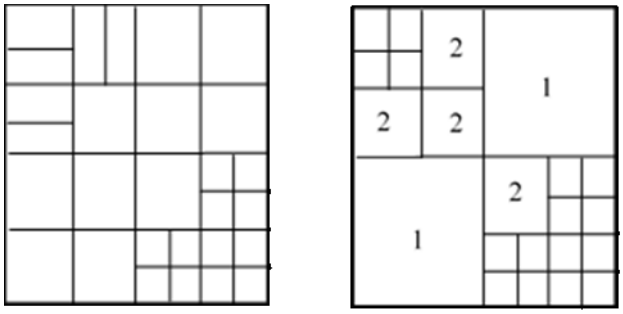
.



(a)H.264(MB partition) (b)H.265(CU partition)

Fig. 2 Representative result of H.264 and HEVC encoding

Both encoders would split in the part with important texture features and stay original size if the part contains less information. In this way, we can use the extracted information from H.264 stream to determine whether a CU needed to be split or not. During this process, we can improve our prediction by means of machine learning dramatically.



(a)H.264 (b)HEVC

Fig. 3 Corresponding partition between H.264 and HEVC for 64x64 size

Homogeneous regions are more likely to be presented by larger blocks in inter prediction. By observation of the final partition of H.264/AVC stream, the coarse partitions are normally applied on the region with smooth motion and result in small residual energy [9]. As illustrated in Fig.3, HEVC has opened up a new method for encoding tree units in four depths range from 0 to 3.The depth value depends on the size of coding unit while 64x64 represents the depth 0.Compared with fixed size of MBs in H.264.This method has better adaptation of interest content. When the depth is 2, the CU size is as big as MB which is 16x16. In this way, a CU depth mode mapping can be built through H.264 and HEVC. Intuitively, if the 16x16 MBs are partitioned in H.264 and the surrounding MBs are partitioned, then this corresponding CU must be split to depth 2 at least. And in turn, the CU may even not be split over depth 1.

III. PROPOSED ALGORITHM

As briefly introduced in Section II, the mode mapping between x264 and x265 is really meaningful. The 16x16 size MBs in H.264 can be mapped as the 16x16 CUs in our x265 encoder in the corresponding position in the same frame. A certain correlation between the two coding regulation must support the depth choice in H.265/HEVC. Thus, the transcoding framework may work well in a relatively faster speed due to the valuable mapping.

A. System Architecture

In this paper, we proposed a H.264/AVC to H.265/HEVC transcoding framework based on JM decoder and x265 encoder and IPPP frame structure is used. As illustrated in Fig.4, we first decode the stream when receiving the stream. During the decoding, all needed information is extracted from stream and stored in x265 encoder. Then the extracted features are calculated in a regular way to use as the input of our proposed learning based model. The learning based model acts as a black box and output the accurate specified depth. The x265 then exactly compute the RD cost of the specified depth rather than calculate the RD cost of all depths recursively .Traditionally, to calculate the best CU depth for a

specific CTU is to complex and tedious. We need to traverse all sizes of a CTU from 8x8 to 64x64, calculating the RD cost and comparing with each other to find the optimal choice.

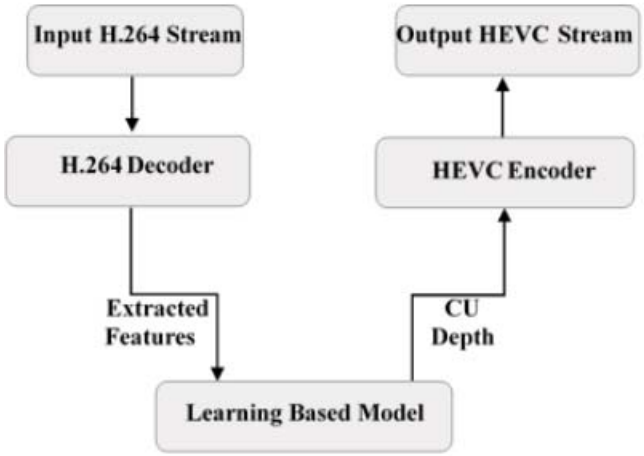


Fig. 4 System Architecture for the proposed transcoding algorithm

B. Depth Prediction using SVM as CU depth mapping

The proposed algorithm is based on the mode mapping, especially the CU depth decision between H.264 and HEVC.

As illustrated in Fig.4, the total framework of our proposed algorithm is separated in two parts: (i) Transcoding from H.264 to HEVC. (ii) Training for a best CU depth mapping model. The input stream first decoded by JM decoder, and we extract features based on early work. Eduardo Peixoto promotes that MV Phase Variance and DCT coefficients may also help to predict the depth [10]. We attempt to add these features in the SVM model for the sake of enhancing accuracy. The results turn out the BDRATE increases. As the correlation between features and videos may be tight or weak, the relatively weak-correlation between videos and MV Phase Variance may result a negative impact for the CU depth decision. To choose the right and suitable features and get a great result appears to be reasonable for fast CU depth decision rather than using all the features. Hence, the partition and MB information are used in this algorithm.

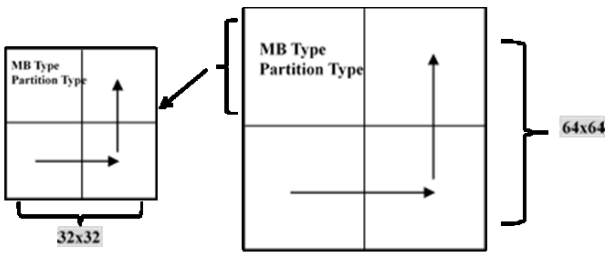


Fig. 5 sum and variance of extracted features for different scales of MBs

Those features can be can be classified into three categories: (i)The QP value of the CTU, the QP value of the CTU ranges from 22 to 37, we classify it into four categories(22,27,32,37). As QP value may influence a lot to the result. (ii)Mb types which indicate the Skip or 16x16 MB size in the stream (iii)Partition Types representing the detailed information 16x8,8x16,8x8 MB size for the specific MB. Considering the spatial correlation of adjacent MBs in a specific region and consisting a CU depth mapping between 16x16 MB to 64x64 CTU need 16 MBs in H.264, the adjacent features of MBs need to be taken into consideration. As shown in Fig.5, we use 32x32 and 64x64 size as ranges for calculating the spatial correlation among different adjacent MBs due to the two sizes representing depth 1 and 0 in CTU of HEVC. The sum and variance of the two features are calculated for 32x32 and 64x64 with 16x16 base units. As the sum represents the partition of depth 0 to 1 and variance reflects a bit more of depth 2 to 3. When we obtain the features, the QP is first predicted and the function enters different model depending on the QP value. For different QP value, the sum and variance calculation function are the same while the specific values are different.

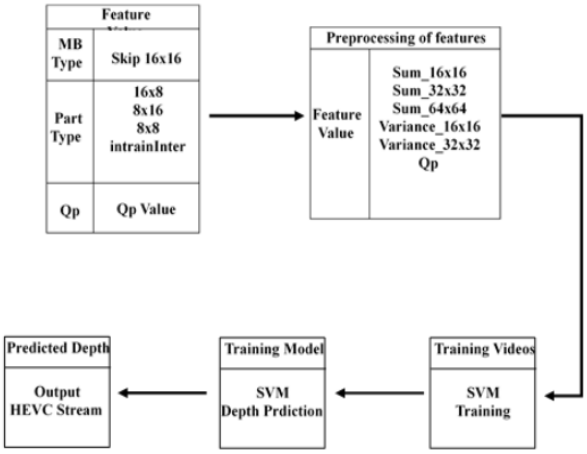


Fig. 6 Detailed process of SVM training and predicting

The Fig.6 details the process of SVM training and predicting. As mentioned above, three kinds of features are extracted and classified. Then we preprocess these features according to the spatial correlation. Sum and variance of different scales of MBs are calculated. We train our model based on supported vector machine (SVM), the supported vector machine (SVM) trains the features to give a CU depth prediction mapping model. This model outputs a specified depth according to the range of different features. During the process of training, a large amount of features for training videos are gathered as the input for the SVM training model. As the second step shown in Fig.6, the features after preprocessing may include sum and variance of features for different scales. We calculate the sum of the value of four smaller blocks in a certain scale and the variance of one smaller block with other three smaller blocks as our preprocessing. There may be tens of thousands combinations of all features as the range of each specific feature may be very large and each feature fluctuate in a large scope. The

SVM Training model first give us all the combinations of features and its corresponding depth.

But we don't calculate all combinations of the features because in many cases the output depth may change even one of these features just change a little. This may result some errors in the judgment. So we combine these features with its corresponding depth when one or two of features change in a certain range with the depth no change.

C. Early Skip and Early pruning

The SVM training model will compute the features and obtain a specified depth according to the input. Then this depth will be set as the depth performed in x265 to proceed with early skip and early pruning.

Depths range from 0 to 3 and depth 0 represents 64x64 size while depth 3 narrow the size to 8x8. Early skip will be performed when the predicted depth is among 1 to 3. Large unnecessary amount of computation can be saved if we skip the unnecessary depth. The RD cost calculation of all PU modes will be skipped when the depth is less than the predicted depth.

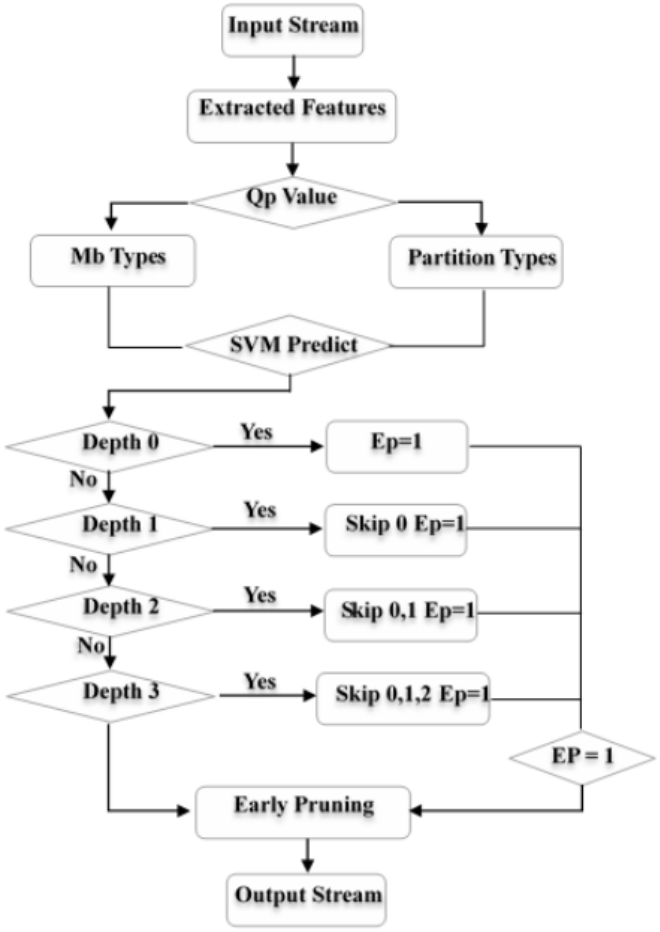


Fig. 7 Proposed algorithm for fast CU decision

The Fig.7 describes the process of the proposed algorithm, the flag EP means early termination of CU splitting. It represents early pruning, the predicted depth first calculates all PU modes and we assume the minimum RD cost modes must exist in this depth. Depth larger than this depth can be abandoned. The x265 partitions its image block using a quad-tree coding strategy recursively. Recursion stops when EP equals to 1 which means pruning is conducted.

Dynamic pruning approach of decision tree is also adopted in this algorithm to reduce error prediction. That means some special situation may need to be taken into consideration. The Fig.6 lists the preprocessing result of the features. And among them, some combinations may result ambiguous prediction. When the features indicate the predicted depth is 2 at least, if the sum and variance of 16x16 CU both exceed a certain threshold. The depth may turn to either 2 or 3. On this occasion, the EP is assigned as 0 and just one more depth will be calculated. That means depth 2 and 3 will be calculated. In this way, we reduce the prediction error dramatically while the coding speed doesn't drop too much.

IV Experimental Results

In this paper, due to the previous work and analysis, the proposed algorithm is not implemented on H.265/HEVC reference Model (HM) as this scientific model is not suitable for the practical application. We adopt x265 instead of HM as our benchmark encoder for its common use in industry. The x265 is an open source implementation of HEVC encoding and targets at real-time coding on generic multicore CPU based platforms. As a result, the encoding speed of x265 is almost hundreds of times of original HM on a modern multicore computer. Now x265 has been used by some famous tools like FFMPEG and VLC [7].

For a transcoding framework, JM is used for first decoding the stream and extracting the information as more features can be obtained by JM and its decoding speed is fast.

To test the effect of our algorithm, all the test cases are encoded and compared by four QPs (22,27,32,37) and x265 employs the same QPs. We defined two parameters to compare the performance and analyze the quality degradation. They are $\Delta PSNR$ and $\Delta Bits$, and $\Delta PSNR$ is defined by equation (1). And $\Delta Bits$ is calculated through the similar equation (2)

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

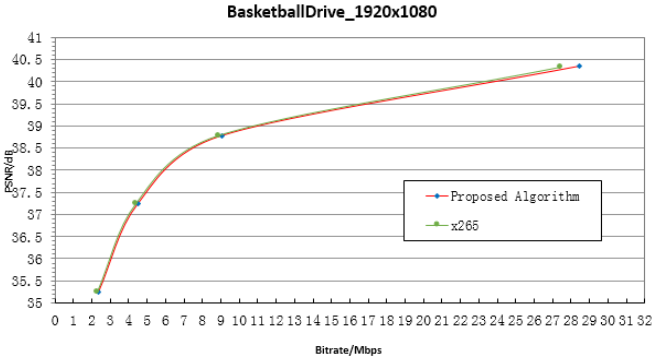
$$\Delta Bits = \Delta Bits_{prop} - \Delta Bits_{x265} \quad (2)$$

To show the result more intuitive and convincing, we calculate the BD-PSNR. The widely-used Bjontegaard Distortion-psnr (BD-PSNR)[8] is adopted and $\Delta PSNR$ and $\Delta Bits$ are used to calculate BD-PSNR. To evaluate the improvement of the coding efficiency, time saving(ΔT) is defined by equation (3) and it represents the time saving between the x265 and our proposed algorithm.

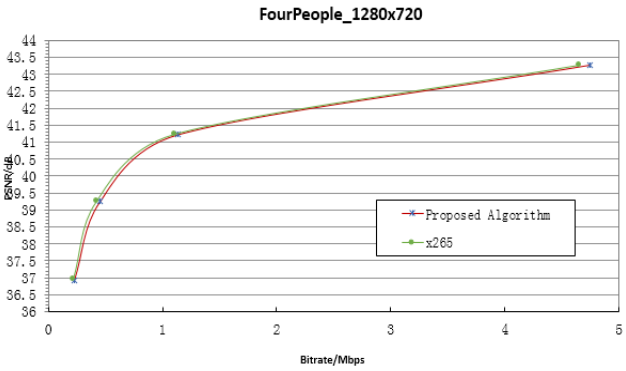
$$\Delta T = \frac{T_{x265orig} - T_{prop}}{T_{x265orig}} \times 100\% \quad (3)$$

We use all standard test sequences of class B and class E. Among videos in each class, just one video is tested and other videos are trained to build the SVM training model. And we cross validate the result in this way. During the process to test and verify our proposed algorithm, we can detect the original depth and the predicted depth to revise our method in order to improve accuracy. We can infer from our method that depth 0 and 1 must hit the real depths because depth 0 and 1 occupy most of depths and will influence the results dramatically. Obviously it's much simple to predict depth 0 and 1 more accurately. And the result indicates us a relatively good mapping table for our transcoding architecture.

The Table I shows the result of our algorithm. Generally, about 41% of encoding speed of x265 is saved on average with only 0.078dB degradation of BD-PSNR. And among all the test sequences, the 720p appears to have lower saving of coding time and higher BD-PSNR increase.



(a) BasketballDrive_1920x1080



(b) FourPeople_1280x720

Fig. 8 RD performance comparison of our proposed transcoding algorithm and the original x265

The resolution influence the performance to some degree and can be considered as a feature for transcoding algorithm, As most past experiments are conducted on HM which has a incredibly slow speed. The result based on x265 is more referenced and practical.

TABLE I
BD-PSNR RESULT OF CU CU MAPPING ALGORITHM

Sequences	ΔTime(%)	ΔPSNR(dB)	ΔBitrate(%)	BD-PSNR(dB)
Cactus (1920x1080)	43.07	0.0163	2.07	-0.0572
BQTerrace (1920x1080)	44.47	0.0175	1.79	-0.06
BasketDrive (1920x1080)	45.90	-0.005	3.26	-0.0569
Jonny (1080x720)	36.73	0.0083	3.22	-0.068
FourPoeple (1080x720)	38.94	-0.0075	2.60	-0.078
Aveg	41.16	0.003	2.79	-0.064

Fig.8 illustrates the RD curves of our proposed algorithm and the x265 full-rdo preset. The performance of our proposed CU mapping algorithm based on transcoding is the same as x265. The RD performance seems to decline a bit when the bitrate is low. We obtain a good tradeoff between the saving of coding speed and the RD performance with no more than 0.08dB drop of BD-PSNR on average.

IV. CONCLUSIONS

In this paper, a learning based fast CU decision algorithm is proposed for H.264 to H.265 transcoder. This new method provides a new way for fast transcoding based on the machine learning. The overall algorithm consists of three important aspects in the transcoding framework (1) The bit stream first decoded by JM. Several significant features are extracted for every specific 16x16 MB in x264.Machine learning model is then used to calculate the specific depth mapping in x265 CU as mentioned above.(2)The x265 skip the unnecessary CU depths, that means depths less than the determined depths mentioned above. The calculation of these depths is dispensable. We directly jump to the determined depth without redundant calculation. (3)If the determined depth is not the last depth, the subsequent depth calculation can be abandoned. In other words, only the chosen depth is calculated. Skipping cu and early pruning may reduce the complexity of x265 encoding while the performance is not bad. The experiment result shows more than 40% coding time is saved with negligible BD-BitRate loss. Further work will be focused on fast PU decision mapping based on 4k and even 8k videos to meet the demands of times.

ACKNOWLEDGMENT

This work was supported by National Key Technology R&D Program of China (2013BAH53F04), NSFC (61221001, 61271221), the 111 Project (B07022 and Sheitc No.150633) and the Shanghai Key Laboratory of Digital Media Processing and Transmissions.

REFERENCES

[1] Haskell, Barry G. Digital Video: An Introduction to MPEG-2: An Introduction to MPEG-2. Springer Science & Business Media, 1997.

[2] Zong-Yi Chen, Tseng Chi-Teng, and Chang Pao-Chi, "Fast Inter Prediction for H. 264 to HEVC Transcoding." 3rd International Conference on Multimedia Technology (ICMT-13). Atlantis Press, 2013.

[3] 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.

[4] Dong Zhang, Bin Li, Jizheng Xu, Houqiang Li, "Fast Transcoding from H.264 AVC to High Efficiency Video Coding" Multimedia and Expo (ICME), 2012 IEEE International Conference on , vol., no., pp.651,656, 9-13 July 2012.

[5] Fangshun Mu, Li Song, "Speed Up HEVC Encoder by Precoding with H.264." Asia-Pacific Signal and Information Processing Association Annual Submit (APSIPA ASC 2014).

[6] Shen, Xiaolin, and Lu Yu, "CU splitting early termination based on weighted SVM." EURASIP Journal on Image and Video Processing 2013.1 (2013): 1-11.

[7] Qingxiong Huangyuan, Li Song, "Performance Evaluation of H. 265/MPEG-HEVC Encoders for 4K Video Sequences." Asia-Pacific Signal and Information Processing Association Annual Submit (APSIPA ASC 2014).

[8] G. Bjontegaard, "Improvements of the BD-PSNR model" VCEG-AI11, July 2008.

[9] Feiyang, Zheng, "Effective H. 264/AVC to HEVC transcoder based on prediction homogeneity." Visual Communications and Image Processing Conference, 2014 IEEE 2014 IEEE (pp. 233-236).

[10] Peixoto, Eduardo, "Fast H. 264/AVC to HEVC transcoding based on machine learning." Telecommunications Symposium (ITS), 2014.