FAST CODING UNIT DEPTH DECISION FOR HEVC

Fangshun Mu[1 2], Li Song[1 2], Xiaokang Yang[1 2], Zhenyi Luo[2 3]

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

ABSTRACT

High Efficiency Video Coding (HEVC) 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, we propose a fast CU depth decision scheme to reduce the encoder complexity for HEVC. Firstly, the relationship between rate-distortion (R-D) cost and CU depth is explored carefully with Mean Squared Error (*MSE*) and Number of Encoded Bits (*NEB*) metrics. Then CU splitting is modeled as a binary classification problem and resolved by an offline trained Support Vector Machine (SVM) model. The experimental results show that the proposed algorithm achieves up to 59% running-time reduction with negligible loss in terms of PSNR and bit rate.

***Index Terms*—**Video coding and processing; HEVC; Content adaptation

1. INTRODUCTION

The emerging high efficiency video coding (HEVC) standard is the joint coding standardization project 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 has shown significant advances in compression efficiency and outperforms the existing H.264/AVC high profile by 50% bitrate reduction at the same reconstructed video quality [2]. HEVC inherits the well-known block-based hybrid coding scheme used by H.264/AVC but employs many new coding tools to improve encoder performance, such as the hierarchical quad-tree The quad-tree coding structure block consists of coding unit (CU), partition unit (PU) and transform unit (TU) [3]. Pictures are composed of coding tree unit (CTU). CTU is allowed to be recursively split into four square CUs, and thus a variety of video content can be represented by content adaptive coding trees comprised of CU blocks with different sizes.

Although the quad-tree structure enables each CU to be coded optimally and can greatly improves the compression efficiency significantly, it imposes significant computation burden on the encoder during the exhaustive rate-distortion (R-D) cost calculation of total 85 CUs, where all possible combinations of CU, PU and TU are tested to find the optimal combination. Thus, it is crucial to find a practical implementation of HEVC to reduce the complexity while maintaining its performance. To overcome this problem, a number of algorithms on accelerating the encoder of HEVC have been proposed to reduce the number of candidate CUs. In [4], Seunghyun Cho et al. proposed a fast CU splitting and pruning method according to a Bayes decision rule method based on low-complexity R-D costs and full R-D costs both defined by author themselves. In [5], Liquan Shen et al. try to predict the maximal and minimal values of depth levels using spatial neighboring treeblocks (left, upper and left-upper). In [6], spatial correlation of CU depth was examined to design an adaptive weighting factor, which was used to adjust the threshold in early termination. In [7], Younhee Kim et al. proposed a fast prediction method based on Rate Distortion cost estimation. In [8], Qin Yu investigated *MSE* of the Inter 2N×2N prediction residual block to early terminate the CU splitting process. But *MSE* only reflects the distortion information of the R-D cost. And In [9], Xiaolin Shen 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.

In this paper, we focus on CU size selection for HEVC and propose a new fast CU depth decision algorithm utilizing machine learning to accelerate the CU size

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.

selection process. Firstly, we analyze R-D cost at different CU depths to explore the correlation between *MSE*, *NEB* and the optimal CU splitting since MSE and NEB reflects two aspects of R-D cost. Then, we early terminate the CU splitting process with properly modeling the problem and applying machine learning algorithms.

The reminder of the paper is organized as follows. Section II briefly describes the CU mode decision process in HM. Section III analyzes the two component parts of R-D cost, distortion and rate, and then presents the relationship between the two selected features (*MSE* and *NEB*) and the optimal CU depth. In section IV, our proposed fast CU depth decision method based on SVM is elaborated. We show the experimental results in Section V and conclude this paper in Section VI.

2. CU MODE DECISION IN HM

In HEVC, the basic block is known as the coding tree unit (CTU), which is 64×64 in pixel size. CTU is the basic unit for inter- and intra- coding and can be recursively divided into four equally sized CUs. The size of CUs is always square and can be as large as 64×64, which known as largest coding unit (LCU); or down to 8×8, which known as smallest coding unit (SCU). The PU is the basic unit used for carrying the information related to the prediction process and can be symmetric or asymmetric. TU is the basic unit for the transformation and quantization process. The size of square-shaped TUs ranges from 4×4 or 32×32 and its size must smaller than or equal to the size of CUs but can be larger than the size of PUs. The relationship between CU, PU and TU is demonstrated in Fig. 1 [10].

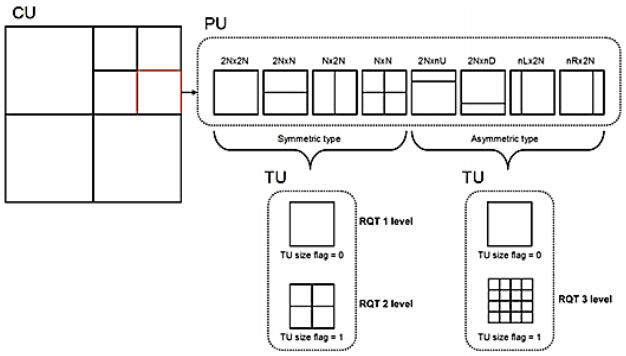


Fig.1 The relationship between CU, PU and TU

The flexible quad-tree coding block partitioning is adopted in HEVC to capture the diversity of video content. CU is similar to the concept of macroblock (MB) in H.264 and extends the concept of MB in H.264/AVC with size varies from 64×64 to 8×8. To decide the best PU mode of a CU, the encoder typically tries all the possible inter PU modes and intra PU modes by computing the rate-distortion (R-D) cost and determines the optimal PU mode with minimum R-D cost. The determination of optimal CU modes is obtained via evaluating the R-D cost of all possible CU, PU and TU sizes.

3. PROBLEM ANALYSIS

In HEVC, the optimal combination of CU, PU and TU size is obtained via full search of all possible combinations of CU, PU and TU sizes. However, it's very complex and time-consuming to decide the optimal coding mode considering many factors such as motion estimation, reference frame selection, etc. To decide the optimal coding mode of LCU, the encoder may computes the R-D costs of all possible coding modes in a brute-force way to find the coding mode with minimum R-D cost. In HM, the R-D cost with mode ***m*** is calculated by [6]:

$$J_m = D_m + \lambda_m(Q_p)R_m$$

where D_m denotes the distortion between the predicted CU and original CU. R_m is the bits to encode the residual in the given mode and prediction. $\lambda_m(Q_p)$ is the Lagrangian multiplier.

Generally, R-D is considered as the criteria of deciding the optimal mode in HEVC and can be calculated for each combination of CU, PU and TU mode. Thus it's necessary for us to investigate the correlations between R-D and CU depth. In the following part, two component parts of R-D, Mean Squared Error *(MSE)* and Number of Encoded Bits *(NEB)*, are analyzed respectively to investigate the relationship between R-D and CU depth.

3.1 Distortion Analysis

When coding a frame, the prediction residuals can reflect the encoding results to some extent. The residuals $R_n^k = B_n^k - P_n^k$ are computed after the CU mode is selected, where B_n^k is the current block and P_n^k is its prediction mode for the block. Then, chosen transformation algorithm is applied to residuals, and transformation coefficients are quantized, entropy encoded and written to the bitstream. Afterwards, the encoder performs inverse transform and inverse quantitation operations, generating the reconstructed residual $R_n^{k\prime}$, which is used to get the reconstructed CU, $R_n^{k\prime} = B_n^{k\prime} - P_n^{k\prime}$.

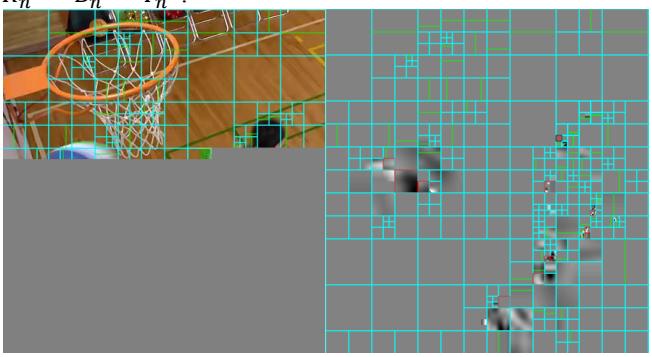


Fig. 2 Original picture (left) and Luma residuals (right) obtained from BasketballDrill (WVGA) and the corresponding optimal CU partition, QP=32

Fig. 2 shows the luma residuals obtained from BasketballDrill (832×480) and the corresponding optimal CU partition optimized by HM 9.1 using the configuration of Lowdelay_main. We can find that large CUs are more likely to be chosen as optimal CU for "flat" or "background" regions, where prediction residuals tend to be small and further split usually brings little prediction improvement but increased side information. For regions which contain moving foreground or edges, small CUs are more likely to be chosen as optimal CU size since large CU due to the inaccurate prediction of large CUs [8].

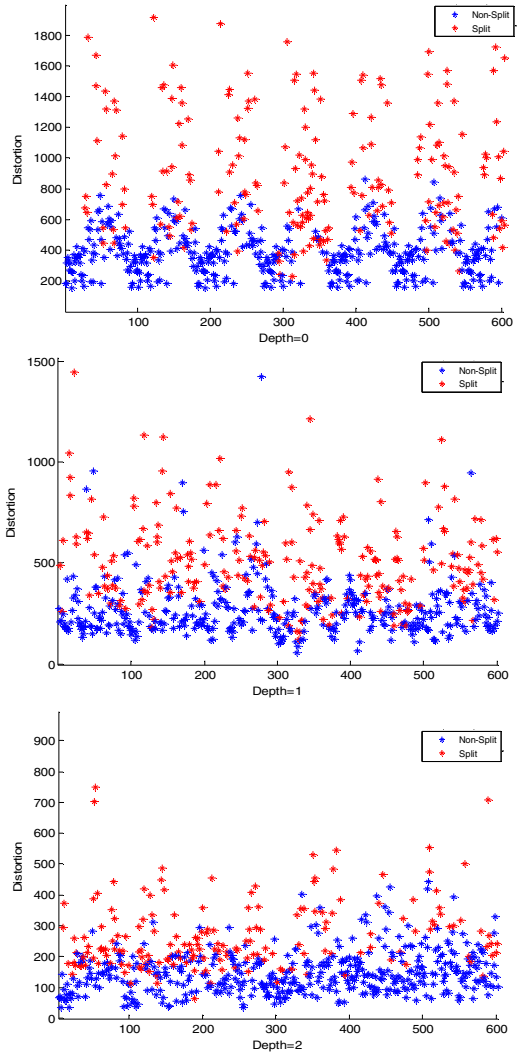


Fig. 3 MSE of CUs at different CU depth under the Inter2Nx2N mode (QP=32)

In order to further explore the relationship between residuals and CU depth, we encode the first 30 frames of BasketballDrill by HM9.1. *MSE* of CUs under the Inter2Nx2N mode along with the CU split information are shown in Fig. 3, where the blue dots represents that current CUs need to be split and the red dots shows that the current CUs are optimal and won't be split further. It can be observed that CUs with small residuals prefer to be coded with large CU size, which those with the large residuals are more likely to be split further for more accurate prediction. This is especially true for CUs at depth 0. Based on the above observation, we propose to predict the CU depth based on residual information to early terminate the CU splitting process.

3.2 Rate Analysis

From the above analysis, we can find that CU depth has a certain relationship with *MSE*. However, R-D consists of two component parts-distortion and rate, and *MSE* only reflects the distortion. So the number of encoded bits also needs to be investigated to further explore the relationship between R-D and CU depth.

In HEVC, *NEB* consisted of two kinds of bits, bits of prediction residuals and bits of CU's mode information. The first part is determined by the distortion of CU. The second part is determined by the CU mode information [7]. As far as a specific CU is concerned, simple modes often leads to large residuals. However, if we use more complicated ME and PU modes, more accurate prediction and less prediction residuals may be obtained. So the encoded bits in R-D can denote the mode information of CU.

Fig. 4 shows the *NEB* for the optimal CUs of BasketballDrill obtained by HM 9.1 under the Lowdelay_main configuration. We can find that large CUs are more likely to be encoded with small amount of bits since they may be predicted accurately with little side information. And the CTUs with large depth are prone to have more bits, one reason is that those CTUs are likely to be foregrounds and have relatively large residuals as demonstrated in 3.1. Another reason is that more side information needs to be encoded.

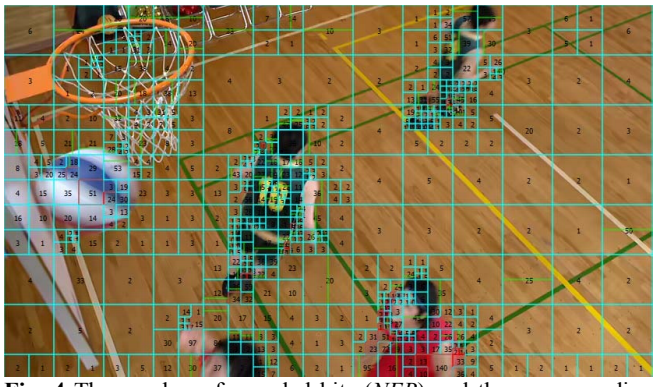


Fig. 4 The number of encoded bits (*NEB*) and the corresponding optimal CU partition of BasketballDrill at QP=32

Similar to the *MSE*, we also probe the relationship between *NEB* and CU depth by encoding the first 30 frames of BasketballDrill using HM9.1. The *NEB* of the optimal CUs at different depths is shown in Fig. 5, where red dots denote split CUs and blue dots denote non-split ones. Similar relationship is also obtained between the *NEB* and the optimal CU depth. From Fig. 5, we can see that t split and non-split CUs at depth 0 (64x64) can be roughly separated by a threshold of *NEB*. Though the thresholds are

not that clear at depth 1 (32x32) and depth 2 (16x16), predicting the non-split CUs to split ones won't influence the performance since we can still find the optimal CU depth through the RDO process.

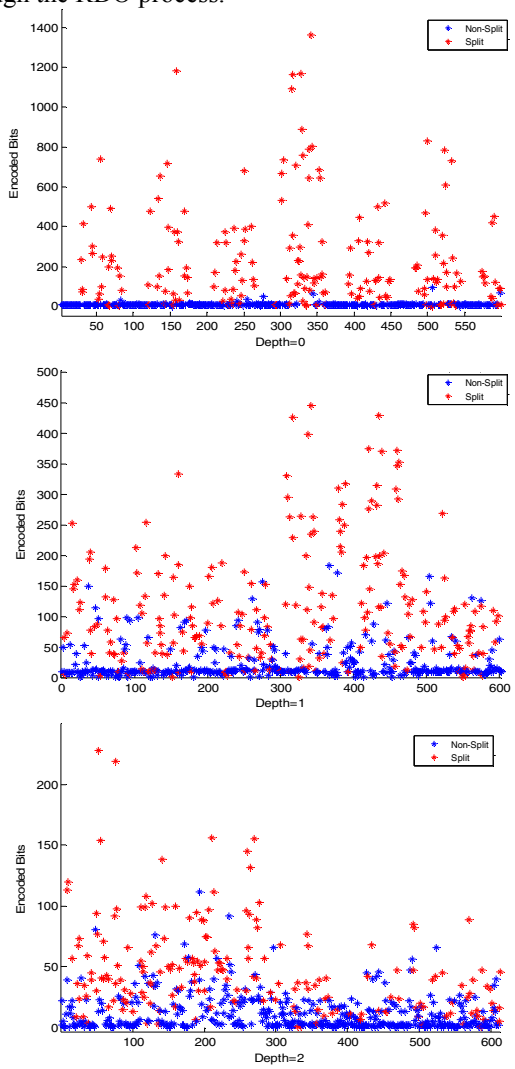


Fig. 5 Number of Encoded Bits (*NEB*) of the optimal CUs at different depth (QP=32)

4. PROPOSED CU DEPTH DECISION ALGORITHM

Since R-D plays a critical role in determining the optimal CU mode, it's computed at each RDO process to find the optimal combination of CU, PU and TU size. However, it should be noted that R-D calculation is very time-consuming because the encoder has to perform complete encoding and decoding process. From Section 3, we know that both of the two component parts of R-D, rate and distortion, hold strong relationship with the CU depth. Fig. 6 shows *NEB* with respect to *MSE* at CU depth 0 (64x64), where CUs that won't be split further are denoted by blue dots while red dots represent current CUs are not optimal and need to be split further. From Fig. 6, we can find the correlation is linearly inseparable. It's impossible for us to predict the CU depth with rate and distortion using linear classifier. Therefore, we utilize machine learning to predict the optimal CU by collecting data and analyzing the data distribution. Machine learning provides a powerful tool using Radial Basis Function, polynomial or higher dimensional kernels rather than a liner separable hyperplane or just a threshold [11].

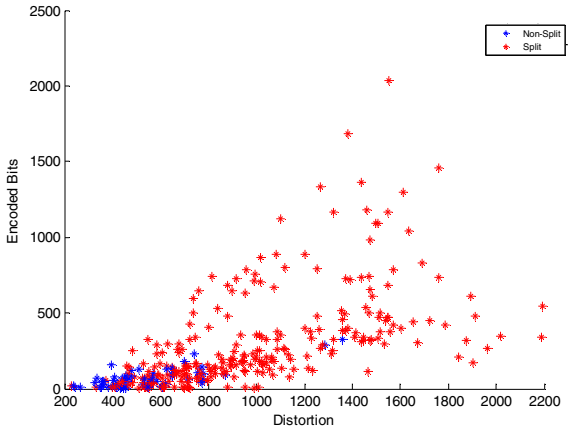


Fig. 6 *NEB* and the *MSE* obtained at CU depth 0 (QP=32)

In this paper, we employ LibSVM developed by Prof. Chih-Jen Lin from National Taiwan University [12]. We adopt LibSVM in our proposed algorithm to find the optimal hyperplane for CU splitting prediction. Since runtime classification of SVM may take much time, to accelerate the encoding process we offline train the SVM model to early terminate the CU splitting process.

We model the CU Depth Decision problem as a binary classification problem and present a learning based solution to resolve the linearly inseparable problem of two selected features, *MSE* and *NEB*. In the proposed algorithm, three models are offline trained on the selected features for different CU sizes {64×64 32×32, 16×16}. The classifier for early termination of the CU splitting process is made by these trained models. The decision comes from the prediction results of the classifier based on *MSE* and *NEB* extracted from CU. The flowchart of the proposed CU depth decision algorithm is shown in Fig. 7.

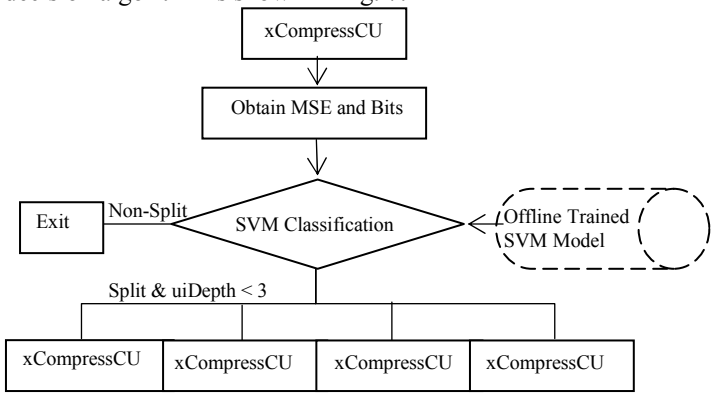


Fig. 7 Flowchart of the proposed fast CU Depth Decision Algorithm

5. EXPERIMENTAL RESULTS

To verify the performance of the proposed algorithm, we implemented the algorithm on HM9.1 under the low delay main configuration. The testing platform used is Intel ®Core(TM) i7-3770@3.40GHz with dual cores, 4 GB RAM. Experiments were performed with the quantization parameters set as 22, 27, 32 and 37. The coding performance is measured by PNSR difference ($\Delta\boldsymbol{PSNR}$), bit rate difference ($\Delta\boldsymbol{Bitrate}$) and time saving ($\Delta\mathbf{T}$) is measured as

$$\Delta\mathbf{T} = \frac{\boldsymbol{T_{HM} - T_{proposed}}}{\boldsymbol{T_{HM}}} \times \mathbf{100\%}$$

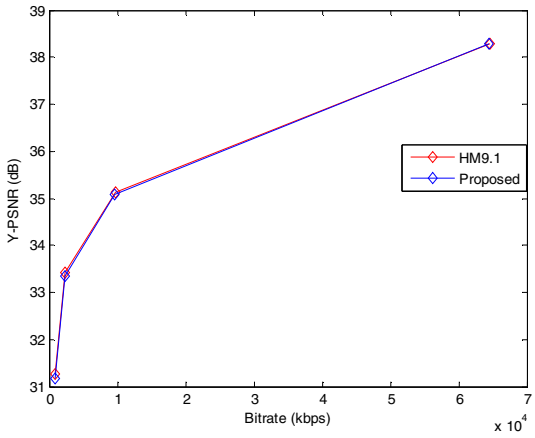
Where T_{HM} and $T_{proposed}$ denote the total encoding time of HM9.1 encoder and the proposed encoder, respectively. In the experiments, the trained set was obtained via the 1920 × 1080 Kimono sequence. The experimental results are presented in Table I and LowDelay configuration is applied.

Table 1: PSNR difference, Bitrate difference and Time Saving of the proposed algorithm

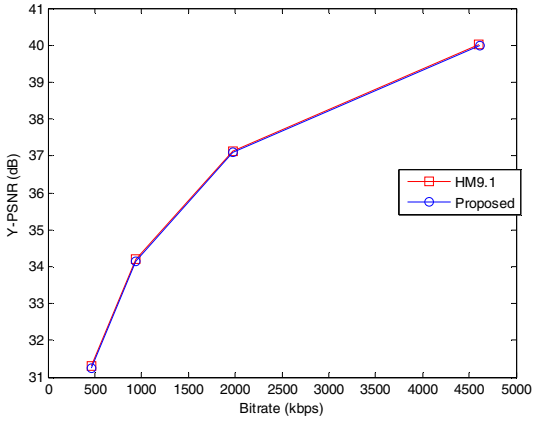
Sequences	QP	ΔTS (%)	ΔPSNR (dB)	ΔBitrate(%)
Basketball Drive	22	7.12	0.00	-0.23
	27	19.23	0.01	-0.08
	32	30.41	0.02	-0.12
	37	30.98	0.02	0.00
BQ Terrance	22	1.15	0.01	-0.08
	27	26.82	0.05	-1.85
	32	47.64	0.07	-2.08
	37	59.07	0.09	-1.90
Cactus	22	9.62	0.02	-0.67
	27	23.07	0.03	0.17
	32	32.76	0.09	0.28
	37	41.84	0.13	0.28
Park Scene	22	7.66	0.03	-0.27
	27	25.77	0.04	-0.47
	32	36.01	0.06	-0.47
	37	47.85	0.07	-0.75
Average		25.44%	0.05	-0.52

The experimental results show that our proposed algorithm can achieve up to 59% running-time reduction with almost negligible decrease in PSNR and increase in bit rate. Furthermore, we observe that QP influences the performance effectively; the reason why our method saves more time at high QPs or low bit rates is that QP is closely related to CU sizes and large CU are more likely to be chosen when QP is large.

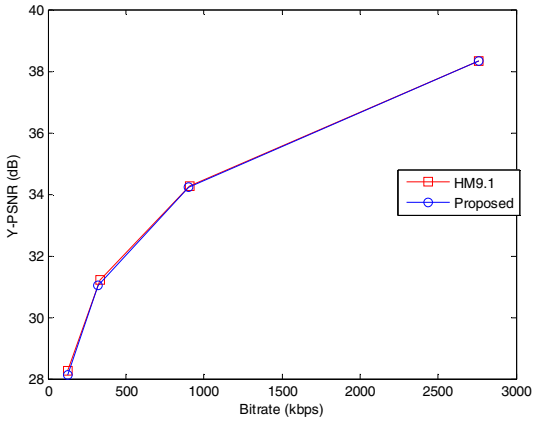
In Fig. 8 (a), (b), (c), (d), the R-D curves of BQTerrance, BQMall, BQ Square, FourPeople are shown. The R-D curves of the test sequences are shown in Fig. 8, where red curves denote the R-D performance of ... and blue curves denote the performance of the proposed algorithm. We can find that our proposed algorithm has similar R-D performance with the reference software regardless of the video sizes.



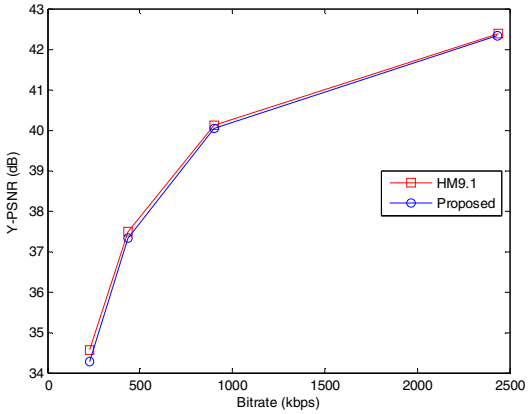
(A) BQTerrance (1920×1080)



(B) BQMall (832×480)



(C) BQ Square (416×240)



(D) Class E FourPeople (1270×720)

Fig. 8 RD Curves of HM9.1 and our proposed algorithm

6. CONCULSION

In this paper, we proposed a fast CU depth decision algorithm based on SVM for HEVC. We utilize the Mean Square Error (*MSE*) and Number of Encoded Bits (*NEB*) as distortion and rate metrics to explore the relationship between R-D and CU depth. Then we incorporates Support Vector Machine (SVM) to model the CU splitting problem as a binary classification problem to resolve the non-linear prediction problem with offline trained SVM model. Experimental results show that our proposed method can significantly reduce the encoding time. And in practice, our method can also be combined with other methods, such as early SKIP mode decision, to further reduce the encoding time.

7. REFERENCES

[1] B. Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, T. Wiegand, “High Efficiency Video Coding (HEVC) text specification draft 9”, JCTVC-K1003_v13, Shanghai, CN, Oct. 2012.

[2] Ugur, Kemal, et al. "High performance, low complexity video coding and the emerging HEVC standard." *Circuits and Systems for Video Technology,* IEEE Transactions on 20.12 (2010): 1688-1697.

[3] Mshsa T. Pourazad, Colin Doutre, Maryam Azimi, Panos Nasiopoulos, “HEVC: The New Gold Standard for Video Compression-How does HEVC compare with H.264/AVC?”, *IEEE Consumer Electronics Magazine*, 2012, pp. 36-46.

[4] Seunghyun Cho; Munchurl Kim, "Fast CU Splitting and Pruning for Suboptimal CU Partitioning in HEVC Intra Coding," *Circuits and Systems for Video Technology,* IEEE Transactions on , vol.23, no.9, pp.1555,1564, Sept. 2013.

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

[6] K Jongho, J Seynoon, C Sukhee, C Jin Soo, “Adaptive Coding Unit Early Termination Algorithm for HEVC, *2012 IEEE International Conference on Consumer Electronics (ICCE),* Las Vegas, pp. 261-262

[7] Younhee Kim; Jongho Kim; et al. "A Rate-Distortion cost estimation approach to fast intra prediction in Video Coding for Ultra High Definition TV applications," *Consumer Electronics (ICCE), 2012 IEEE International Conference on* , vol., no., pp.164-165.

[8] Qin Yu, Xinfeng Zhang, Shiqi Wang, Siwei Ma, “Early termination of coding unit splitting for HEVC”, *Signal a& Information Processing Association Annual Summit and Conference (APSIPA ASC),* 2012,pp.1-4.

[9] 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.

[10] Ken McCann, et al. “Samsung’s Response to the Call for Proposals on Video Compression Technology”; Proposal to JCT-VC ; JCTVC-A124

[11] Chiang Chen-Kuo, et al. "Fast H. 264 encoding based on statistical learning." *Circuits and Systems for Video Technology,* IEEE Transactions on 21.9 (2011): 1304-1315.

[12] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A Library for Support Vector Machines.