Multi-Level Machine Learning-based Early Termination in VP9 Partition Search

Yang Xian; The City University of New York; New York, NY. Yunqing Wang; Google Inc.; Mountain View, CA. Yingli Tian; The City University of New York; New York, NY. Yaowu Xu; Google Inc.; Mountain View, CA. Jim Bankoski; Google Inc.; Mountain View, CA.

Abstract

In VP9, a 64×64 superblock can be recursively decomposed all the way to blocks of size 4×4. The encoder performs the encoding process for each possible partitioning and the optimal one is selected by minimizing the rate and distortion cost. This scheme ensures the encoding quality, but also brings in large computational complexity and substantial CPU resources. In this paper, to speed up the partition search without sacrificing the quality, we propose a multi-level machine learning-based early termination scheme. One weighted Support Vector Machine classifier is trained for each block size. The binary classifiers are used to determine that provided a block, whether it is necessary to continue the search down to smaller blocks, or to perform the early termination and take the current block size as the final one. Moreover, the classifiers are trained with varying error-tolerance for different block sizes, i.e., a stricter error-tolerance is adopted for larger block size compared with the smaller ones to control the encoder performance drop. Extensive experimental results demonstrate that for HD and 4K videos, the proposed framework accomplishes remarkable speed-up (20-25%) with less than 0.03% performance drop measured in the Bjøntegaard delta bit rate (BDBR) compared with current VP9 codebase[1].

Introduction

VP9, an open-source video codec released by Google [16], adopts a flexible quad-tree structure during encoding. For applications targeting at high performance video coding, an exhaustive search has to be performed in a recursive manner in order to find the optimal partitioning of an encoding unit. The encoder performs the encoding process for each possible partitioning and the optimal one is chosen by the rate and distortion (RD) cost, i.e., the partitioning that gives the least RD error is selected.

In VP9, superblocks with size 64×64 can be recursively decomposed all the way down to blocks of size 4×4. As shown in Fig. 1, blocks with size $2N \times 2N$ ($N = 32, 16, 8, 4$) could either be the final encoding block (i.e., non-partition mode), or they can be further decomposed into smaller sub-blocks in three different modes (i.e., horizontal, vertical, and split). Each sub-block after 'split' mode could be further decomposed in a similar manner. The use of a broad range of the partition sizes in a quad-tree structure improves the coding efficiency, but on the other hand, increases the computational complexity and consumes substantial CPU resources.

To speed up the encoding process without sacrificing the final encoding performance, in current VP9 codebase, a thresholding-based strategy [1] is utilized by setting a combination of three termination criteria (i.e., SKIP flag, Rate_THRESHOLD, Distortion_THRESHOLD) to early terminate the partition search. These criteria are used to evaluate the partition node to see if the current partition size is good enough to be the final encoding unit, and if so, all its child nodes are not visited and the search is terminated for this branch. While this simple thresholding-based scheme provides a significant speed-up, it is still computationally expensive for the whole encoding process, especially for HD and 4K videos which are increasingly popular recently.

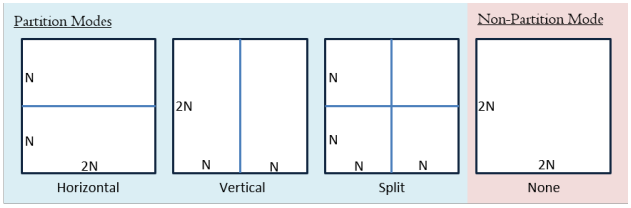

Figure 1. *Encoding modes for blocks with size $2N \times 2N$ ($N = 32, 16, 8, 4$). Figure is better viewed in color.*

Numerous research has been done to reduce the heavy computational burden on encoder for HEVC based on fast coding unit (CU) decisions. In [5], the depth information of the neighboring and the co-located CUs are employed to accelerate the RD optimization in both the frame level and the CU level. A Bayesian decision rule based framework was proposed in [9] for fast CU size decisions. Xiong *et al.* presented a pyramid motion divergence method to early skip the specific inter CUs [4]. In [10], a fast CU decision algorithm was proposed based on the spatio-temporal encoding parameters.

Recently, machine learning and data mining techniques have been applied to assist the encoder speed-up. Shen and Yu employed weighted support vector machines (SVM) [20] to perform CU early termination for HEVC [13]. In [17], three sets of decision trees obtained through data mining are built to avoid running the RD optimization algorithm to its full extent. Another machine learning-based fast CU depth decision framework for HEVC was proposed in [11] where the problem is modeled as a hierarchical binary classification problem. Zhu *et al.* constructed three-level binary classifiers to predict CU partition in HEVC in which a feature selection algorithm and a probability threshold determination scheme are incorporated [12]. Han *et al.* applied weighted SVMs to early terminate partition search in VP9 [2].

In this paper, we propose a multi-level machine learning-based early termination framework for VP9. The major contributions are: 1) Varying error-tolerance (measured in RD cost in-

[1]This work was done while Yang Xian was an intern under the supervision of Yunqing Wang at Google Inc, Mountain View, CA.

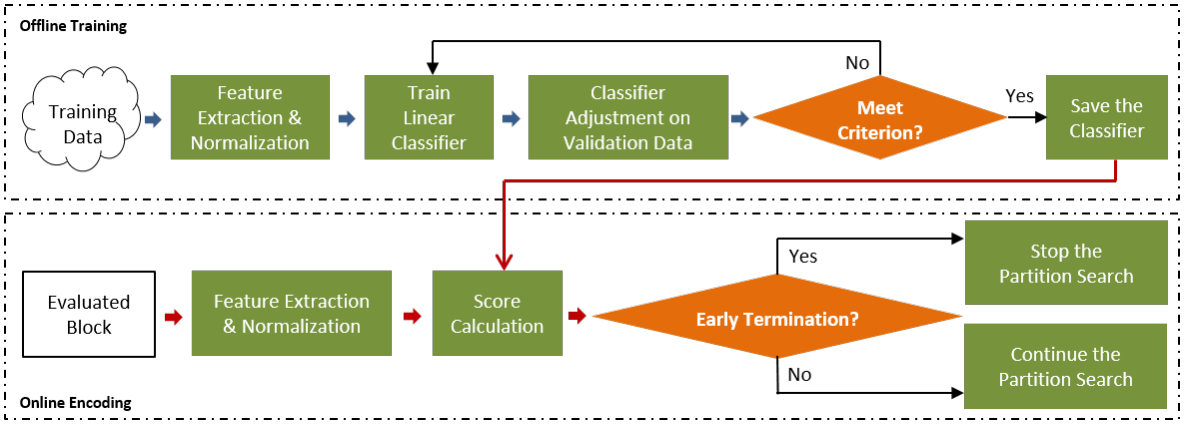

Figure 2. *Overview of the proposed multi-level machine learning-based early termination framework.*

crease) is adopted for different block sizes. Early termination related calculation overhead is minimized. Therefore, the encoder speed-up is maximized with the minimal performance loss. 2) Remarkable performance on videos of different resolutions are observed even though the classifiers are trained with four HD clips. For example, compared with the VP9 codebase[2], 20% speed-up is achieved for HD videos with less than 0.03% performance drop measured in Bjøntegaard delta bit rate (BDBR) [18]. 25% speed-up is observed without any encoder performance loss for 4K videos. 3) The trained classifiers are robust to ensure a stable encoder performance when the VP9 codebase is updated. Therefore, it will not be necessary to retrain the classifiers each time with the update.

Multi-Level Machine Learning-based Early Termination in Partition Search

In this section, we present the multi-level machine learning-based early termination framework to avoid unnecessary trials during the partition search. The block partition is modeled as a binary classification problem which separates the partition modes from non-partition mode (as illustrated in Fig. 1). The early termination decision is made at every block size level with varying error-tolerance.

Fig. 2 provides the schematic pipeline for the proposed early termination scheme. It consists of two parts−the offline training phase and the online encoding phase. For each block size, a weighted SVM classifier is trained over the features extracted from the training videos, i.e., four HD clips. In order to maintain the RD performance when a misclassification occurs, error control (measured in RD cost increase in the validation dataset) is performed. During the online encoding process, the saved binary classifiers are then applied to determine whether it is necessary to early terminate the partition search of the evaluated block. Details are presented in the following subsections.

[2]VP9 is being updated regularly. In this paper, the comparison is performed with the codebase as of June. 15, 2016 which utilizes the thresholding-based early termination strategy [1]. In the 'Experimental Results' section, we evaluate the robustness of the proposed framework with the codebase update.

Feature Selection and Normalization

Adopting representative and relevant features is crucial for classification tasks, and in our case, to be able to accurately separate non-partition case from the partition modes. The same set of features are employed in the proposed framework as presented in [2] which are: 1) rate cost of the non-partition mode in the current block; 2) distortion cost of the non-partition mode in the current block; 3) magnitude of the motion vector of the non-partition mode in the current block; 4) the partitioning modes of the co-located block in the previous frame, the above block and the left block in the current frame; 5) the number of nonzero coefficients to encode the non-partition mode in the current block; 6) quantizer Q value of the current frame. These features are selected using a similar scheme as introduced in [13] based on F-score.

The extracted features need to be normalized per dimension before feeding to the classifier training due to the fact that the absolute values vary dramatically for each feature dimension. Since this normalization procedure needs to be performed for each evaluated block during the online encoding process, we adopt a simple yet effective normalization method (i.e., standardization) to minimize the calculation overhead.

We denote x_i as the original feature vector of the i-th dimension. The normalized version $x_i^{'}$ is calculated as $x_i^{'} = (x_i - \mu_i)/\sigma_i$ where μ_i and σ_i represent the mean and standard deviation of dimension i in the training set. As observed in our experiments, compared with other sophisticated normalization strategies (e.g., softmax with sigmoid function), the classification performance is comparable. However, the standardization scheme has minimal calculation overhead and therefore, is more suitable for the fast encoding applications.

Classifier Training and Optimization

The early termination decision in each block size level is modeled as a binary classification problem, where a SVM classifier is applied. The key idea of SVM is to find an optimal classification hyper-plane with the maximum margin between the two classes [20]. During the training, blocks that require further partitioning (i.e., horizontal, vertical, split) are assigned with label -1 (negative samples), and label $+1$ for non-partition blocks (positive samples).

To maintain the RD performance when a misclassification occurs and to reduce the impact of outliers, RD loss due to mis-

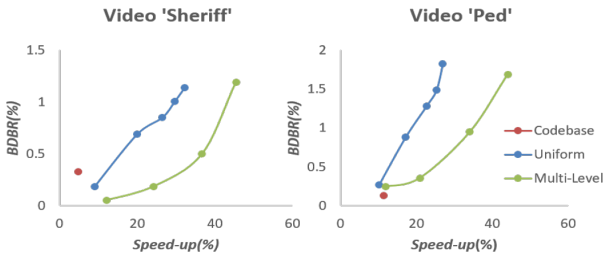

Figure 3. *Performance comparisons of VP9 codebase, uniform machine learning-based early termination, and the proposed multi-level early termination scheme on videos 'sheriff' and 'ped'. Under each early termination setting, multiple error thresholds are tested. Figure is best viewed in color.*

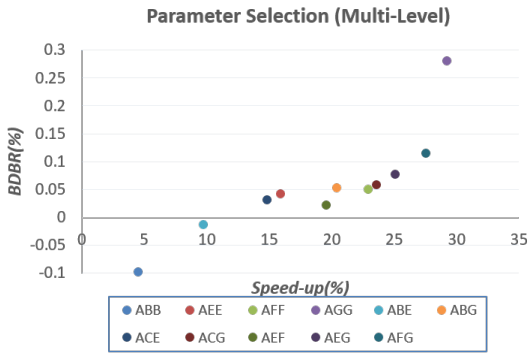

Figure 4. *Error threshold θ selection for the proposed early termination framework. Figure is best viewed in color.*

classification is introduced as weights in SVM training. Misclassifying non-partition blocks as partition blocks will not cause any RD cost increase since there is no early termination performed, but not vice versa. Moreover, misclassifying different negative samples brings in different RD cost increase. Therefore, we adopt the same weighting scheme as introduced in [2], i.e., unit weight for all positive samples and for negative samples, the weights are proportional to the ratio of the RD cost increase caused by misclassifying this block to the average RD cost increase in this video. This proportion is controlled by factor λ. In general, a larger cost is raised when misclassifying a negative sample under a larger λ.

The optimal λ is then determined by measuring the RD cost increase utilizing a separate validation dataset. To be more specific, an error threshold θ (e.g., 0.1%) is first pre-defined which measures the ratio of the RD cost increase caused by the machine learning-based early termination to the best RD cost without any early termination scheme. This threshold controls the maximum performance degradation allowed. As observed, in general, larger speed-up resulted from early-termination is often accompanied by more RD cost increase. Therefore, we aim to generate a classifier that will maximize the time saved by early termination scheme while keeping the resulted RD cost increase under this pre-set threshold. In practice, the classifier is adjusted by tuning λ aiming to maximize ΔT subject to the condition: $\Delta\theta \leq \theta$ (ΔT and $\Delta\theta$ measure the corresponding time saving and RD cost increase). This step is performed recursively as shown in Fig. 2.

Multi-level Early Termination Scheme

In the training phase, one classifier is trained for each block size $N \times N$ ($N = 64, 32, 16$ in our implementation). During the online encoding, provided an evaluated block, the corresponding offline trained SVM classifier is loaded, which predicts the class label. The label is then utilized to decide whether it is necessary to continue the search, or to perform the early termination and take the current block as the final encoding unit.

As shown in the previous subsection, the error threshold θ is crucial in controlling the trade-off between the encoder speed-up and the final encoding performance drop. In general, the larger the threshold is, the higher the speed-up gain, but we will also have a larger performance loss. If a global θ is adopted for the training of all three classifiers (denoted as uniform-setting machine learning-based early termination), as observed in the experiments, for some videos (e.g., video 'sheriff' in Fig. 3-left), the uniform setting outperforms the codebase. However, for some other video clips, the performance is not that satisfying (e.g., video 'ped' Fig. 3-right).

To address this issue, a multi-level error control scheme is proposed to control the quality loss and to maximize the speed-up from early-termination. θ is adjusted adaptively for each classifier. We place a stricter error-tolerance control for classifiers in higher level of the search tree (i.e., blocks with larger sizes), but keep a relatively looser control in the lower levels. The search process follows a pre-order depth-first traversal in which the parent node is evaluated before the child nodes. In other words, a 64×64 block has to be evaluated before possibly going down the tree to further evaluate blocks with smaller sizes. If a 64×64 block is misclassified to be a non-partition block but it actually should be further partitioned, it will cause more quality loss than misclassifying a 32×32 or 16×16 block. Therefore, to ensure the encoding quality and maximize the speed-up, we adopt the multi-level control scheme so that blocks with smaller sizes are 'more encouraged' to be early terminated. As demonstrated in Fig. 3, by performing error control in different levels, a better performance is gained for both videos (and for all the videos tested in our experiments). Multiple data points are reported in each curve which correspond to different choices of θ. More details of the parameter selections are presented in the 'Experimental Results' section.

Experimental Results

In this section, the proposed multi-level machine learning-based early termination framework is evaluated with multiple videos in different resolutions. Experimental results demonstrate that the proposed method speeds up the encoding process significantly with negligible loss of encoding performance measured in BDBR.

Implementation Details:

LIBSVM [19] is employed to train the linear classifiers. As recommended in [2], 2-20 frames of four HD videos (i.e., old_town_cross_420_720p50, blue_sky_1080p25, city, crowd_run_1080p50) are utilized for training. These videos are encoded with multiple bit rates to cover the possible Q values during real encoding. 21-100 frames of these videos serve as the validation dataset to find the optimal θ.

To find the optimal set of error tolerance parameters for different block sizes, we conduct an experiment with a separate set

of HD clips. For ease of notation, we label the selected θ values as: $A(0.01\%)$, $B(0.02\%)$, $C(0.03\%)$, $D(0.04\%)$, $E(0.05\%)$, $F(0.1\%)$, and $G(0.2\%)$. The notation '*ABC*' indicates that a threshold of 0.01% is adopted for 64×64 level while 0.02% and 0.03% for sizes 32×32 and 16×16, respectively. Different combinations are tested as shown in Fig. 4. In general, the encoding performance decreases as we target at a higher speed-up. Compared with the VP9 codebase, we manage to achieve 10% speed-up with even better encoding performance. Moreover, the encoding performance loss is negligible (less than 0.03%) when a 20% speed-up is accomplished. To ensure the robustness of the trained classifiers, we adopt '*AEF*' as our final setting.

Table 1. Comparison of the proposed multi-level machine learning-based early termination scheme with the codebase measured in BDBR and corresponding speed-up.

	midres	STDHD	HD	4k
Speed-up	10.71%	19.58%	20.00%	24.57%
BDBR	+0.027%	+0.022%	+0.000%	−0.003%

Table 2. Comparison of the proposed multi-level machine learning-based early termination scheme (trained over two different codebase) with the codebase in Aug. 1, 2016 measured in BDBR.

	midres videos	HD videos	4k videos
C_1	+0.089%	+0.063%	+0.112%
C_2	+0.062%	+0.097%	+0.160%

Quantitative Evaluation:

In Table 1, we present the encoder time saving and BDBR for videos in multiple datasets compared with the codebase. Four datasets are evaluated: 1) 'midres'—with 30 480p videos; 2) 'STDHD'—standard HD set with 15 720p & 1080p videos; 3) 'HD'—extended HD set with 41 720p & 1080p videos; 4) '4K'—with 15 4K videos. As observed, the classifiers trained from four HD videos achieve remarkable performance on videos in different resolutions, i.e., 10%–25% speed-up is accomplished with less than 0.03% performance drop for all datasets. In general, better performance is obtained as the resolution gets higher. For the recent emerging 4K videos, compared with the codebase, we achieve better encoder performance while accomplishing around 25% speed-up. The comparisons are performed with the same codebase between our previous work [2] and the one proposed in this paper. The experimental results indicate that, in order to achieve a similar speed-up for HD videos, our current framework substantially reduces the encoder performance loss from over 1.5% to less than 0.03% in BDBR.

Robustness Evaluation:

VP9 is the current generation video codec and is being improved and updated actively. It is infeasible to retrain a set of classifiers each time the codebase is updated. To investigate the robustness of the trained classifiers in adjusting the codebase update, the following experiment is conducted.

Compared with the codebase of June. 15, 2016, in Aug. 1, 2016, the encoder performance in the codebase gets improved significantly (i.e., -1.5% in BDBR for midres videos; -2.0% in BDBR for HD videos). The original set of classifiers trained via the codebase in June is denoted as C_1. We retrain the classifiers (represented as C_2) based on the codebase in August utilizing the features re-extracted following the same pipeline (details can be found in the previous section). Table 2[3] provides the comparison over the August codebase for these two classifiers on 'midres', 'HD', and '4K'. The encoder performance is comparable measured in speed-up (not shown in this table) and BDBR which demonstrates the robustness of the trained classifiers in adaption to the codebase update.

Conclusions

In this paper, we have proposed a multi-level machine learning-based framework to early terminate the partition search in VP9. For HD and 4K videos, a 20%–25% speed-up is obtained with less than 0.03% performance drop in BDBR. The trained classifiers are general enough to handle videos in various resolutions and are robust in handling the VP9 codebase update.

References

[1] Yunqing Wang, Early Termination In Partition Search for Encoding, Technical Disclosure Commons. (2015).

[2] Xintong Han, Yunqing Wang, Yaowu Xu, and Jim Bankoski, Machine Learning-based Early Termination in Prediction Block Decomposition for VP9, Proc. IS&T/SPIE Electronic Imaging. (2016).

[3] Chenggang Yan, Yongdong Zhang, Jizheng Xu, Feng Dai, Liang Li, Qionghai Dai, and Feng Wu, A Highly Parallel Framework for HEVC Coding Unit Partitioning Tree Decision on Many-core Processors, Signal Processing Letters, 21, pg. 573-576. (2014).

[4] Jian Xiong, Hongliang Li, Qingbo Wu, and Fanman Meng, A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence, IEEE Transactions on Multimedia, 16, pg. 559-564. (2014).

[5] Jie Leng, Lei Sun, Takeshi Ikenaga, and Shinichi Sakaida, Content Based Hierarchical Fast Coding Unit Decision Algorithm for HEVC, Proc. International Conference on Multimedia and Signal Processing. (2011).

[6] Liquan Shen, Zhi Liu, Xinpeng Zhang, Wenqiang Zhao, and Zhaoyang Zhang, An effective CU size decision method for HEVC encoders, IEEE Transactions on Multimedia, 15, pg. 465-470. (2013).

[7] Jaehwan Kim, Jungyoup Yang, Kwanghyun Won, and Byeungwoo Jeon, Early determination of mode decision for HEVC, Proc. Picture Coding Symposium. (2012).

[8] Michele Belotti Cassa, Matteo Naccari, and Fernando Pereira, Fast rate distortion optimization for the emerging HEVC standard, Proc. Picture Coding Symposium. (2012).

[9] Xiaolin Shen, Lu Yu, and Jie Chen, Fast coding unit size selection for HEVC based on bayesian decision rule, Proc. Picture Coding Symposium. (2012).

[10] Sangsoo Ahn, Bumshik Lee, and Munchurl Kim, A Novel Fast CU Encoding Scheme Based on Spatiotemporal Encoding Parameters for HEVC Inter Coding, IEEE Transactions on Circuits and Systems for Video Technology, 25, pg. 422-435. (2015).

[11] Yun Zhang, Sam Kwong, Xu Wang, Hui Yuan, Zhaoqing Pan, and Long Xu, Machine Learning-Based Coding Unit Depth Decisions

[3] Please note that Table 1 is based on comparisons with the codebase in June. 15, 2016. Table 2 is compared with codebase in Aug. 1, 2016.

for Flexible Complexity Allocation in High Efficiency Video Coding, IEEE Transactions on Image Processing, 24, pg. 2225-2238. (2015).

[12] Linwei Zhu, Yun Zhang, Na Li, Gangyi Jiang, and Sam Kwong, Machine learning based fast H.264/AVC to HEVC transcoding exploiting block partition similarity, Journal of Visual Communication and Image Representation, 38, pg. 824-837. (2016).

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

[14] Jian Xiong, Hongliang Li, Fanman Meng, Qingbo Wu, and King Ngi Ngan, Fast HEVC Inter CU Decision based on Latent SAD Estimation, IEEE Transactions on Multimedia, 17, pg. 2147-2159. (2015).

[15] I. Ahmad, Xiaohui Wei, Yu Sun, and Ya-Qin Zhang, Video transcoding: an overview of various technique and research issues, IEEE Transactions on Multimedia, 5, pg. 793-804. (2005).

[16] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald Bultje, The latest open-source video codec VP9-an overview and preliminary results, Proc. Picture Coding Symposium. (2013).

[17] Guilherme Corrêa, Pedro A. Assuncao, Luciano Volcan Agostini, and Luis A. da Silva Cruz, Fast HEVC Encoding Decisions Using Data Mining, IEEE Transactions on Circuits and Systems for Video Technology, 25, pg. 660-673. (2015).

[18] Gisle Bjøntegaard, Calculation of Average PSNR Differences between RD-Curves, ITU-T Video Coding Experts Group (VCEG). (2011).

[19] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology, pg. 1-27. (2011).

[20] Corinna Cortes and Vladimir Vapnik, Support-Vector Networks, Machine Learning, 20, pg. 273-297. (1995).