Content Adaptive Fast CU Size Selection for HEVC Intra-Prediction

Buddhiprabha Erabadda, Thanuja Mallikarachchi, Gosala Kulupana and Anil Fernando
Centre for Vision Speech and Signal Processing, University of Surrey, United Kingdom
Email: {e.harshani, d.mallikarachchi, g.kulupana, w.fernando}@surrey.ac.uk

***Abstract*—This paper proposes a content adaptive fast CU size selection algorithm for HEVC intra-prediction using weighted support vector machines. The proposed algorithm demonstrates an average encoding time reduction of 52.38% with 1.19% average BDBR increase compared to HM16.1 reference encoder.**

I. Introduction

High Efficiency Video Coding (HEVC) constitutes an assortment of new coding tools and a quadtree based coding structure, that contribute to its superior coding efficiency improvement over its predecessor, H.264/AVC [1]. However, the Rate-Distortion (RD) optimization based approach used to determine the optimum coding parameters for a given content, substantially increases the complexity of HEVC encoders.

The state-of-the-art methods for encoding complexity reduction can be generally grouped into two categories; statistical knowledge-based and learning-based approaches. For example, the algorithms proposed by Cho *et al.* [2], and Thanuja *et al.* [3] adopt statistical knowledge based approaches to early determine the Coding Unit (CU) size using Bayes decision rules, and texture statistics, respectively. On the other hand, learning based methods use supervised machine learning techniques to generate offline inference models utilizing a large amount of training data. For example, Zhang *et al.* [4] propose Support Vector Machines (SVM) to determine the early termination of CU splitting in the HEVC quadtree structure. In addition, deep learning based neural network models such as Convolutional Neural Networks (CNN) have also been utilized with different architectures and features [5], for similar purposes. However, fixed thresholds, rigid decision trees, and offline trained rigid inference models make these algorithms less flexible to the changing video content. Thus, content adaptive decision making models are crucial to make fast CU size decisions while keeping the coding efficiency performance intact.

To this end, this paper proposes a content adaptive fast CU size selection algorithm for HEVC intra-prediction using weighted SVMs, which, the models are trained using content specific data collected online during the encoding cycle.

II. Proposed Method

The proposed method comprises of weighted SVMs that are generated and applied on two different levels at the encoder.

This work was supported by the CONTENT4ALL project, which is funded under European Commission's H2020 Framework Program (Grant number: 762021).

A. Level-1 (L-1) SVMs: Features and optimal weights

1) Data collection: L-1 has two SVMs for each CU depth. Training data is obtained during the encoding process with the traditional RD optimization, until $N = 2000$ data points are gathered for each CU depth level, $i = 0, 1, 2$. The L-1 SVMs are re-created after a certain number of predictions and the ratio between the number of training and predicted samples is 1:400. This ensures the SVM models are kept relevant to the content and their CU split decisions are content adaptive.

2) Feature selection: Let F_j^i be the set of features extracted for CU depth i in the j^{th} level. Therefore, at L-1, $F_{j=1}^i$ is defined as,

$$F_{j=1}^i := \{\alpha^i, \beta^i, \gamma^i\}, \qquad (1)$$

where, α^i, β^i, and γ^i correspond to the texture complexity, estimated RD cost for the current depth, and context information, respectively for the i^{th} CU depth [4].

3) Weight calculation: The objective function for the two class, weighted SVM quadratic programming problem can be defined as,

$$\min \frac{1}{2}||w||^2 + C \cdot W_{ns} \sum_{i=1}^{N_{ns}} \zeta_i + C \cdot W_s \sum_{i=1}^{N_s} \zeta_i, \qquad (2)$$

s.t. $y_i(w \cdot x_i + b) \geq 1 - \zeta_i$, where $\zeta_i \geq 0$, $\forall x_i$. Here, $x_i = 1, 2, 3..., N$ is the feature vector of the training set, with i^{th} sample being represented as $\{x_i, y_i\}$, where $y_i \in \{+1(split), -1(non-split)\}$ is the class label. Furthermore, w, ζ, C, W_{ns} and W_s are the hyperplane margin, slack variable, and trade-off parameter for hyperplane margin width and misclassification, and weight parameters for CU non-split and split classes, respectively.

The weight parameters W_{ns} and W_s are computed with the data collected during the encoding process. Here, weight value pairs ranging from 1:5($\{W_s : W_{ns}\}$) and 5:1($\{W_s : W_{ns}\}$) with 0.5 step sizes are evaluated for each SVM, and the weight pair that achieves the highest precision is selected to build the SVM classification model. In this case, the precisions for split φ_s and non-split φ_{ns} classes are computed as $\varphi_s = {}^{T_p}/_{T_p+F_p}$ and $\varphi_{ns} = {}^{T_n}/_{T_n+F_n}$. Here, T_p, T_n are the true positive, and true negative samples, whereas F_p and F_n are the false positive and false negative samples, respectively.

B. Level-2 (L-2) SVMs: Features and optimal weights

The CUs with split decisions that are not categorized by the L-1 SVMs are evaluated using the L-2 SVMs.

TABLE I
CODING EFFICIENCY AND COMPLEXITY REDUCTION PERFORMANCE (ALL INTRA MAIN).

Sequence	Proposed (δ = 100) vs HM			Proposed (δ = 20) vs HM			Zhang *et al.* [4] vs HM			Liu *et al.* [5] vs HM		
	ΔT(%)	BD-Rate (%)	BD-PSNR(dB)	ΔT (%)	BD-Rate (%)	BD-PSNR(dB)	ΔT (%)	BD-Rate (%)	BD-PSNR(dB)	ΔT (%)	BD-Rate (%)	BD-PSNR(dB)
Kimono	72.60	2.32	-0.08	81.06	5.92	-0.20	80.74	4.13	-0.14	70.50	2.54	-0.08
Basketball Pass	52.68	0.56	-0.03	72.15	5.99	-0.34	51.84	1.21	-0.07	54.55	2.80	-0.16
BQTerrace	56.17	0.91	-0.05	72.94	7.52	-0.33	52.03	0.80	-0.04	56.78	1.95	-0.09
Traffic	61.71	0.56	-0.03	78.59	6.45	-0.30	49.48	0.98	-0.05	59.02	2.35	-0.11
RaceHorses	40.92	0.47	-0.03	62.75	4.82	-0.28	49.07	1.04	-0.06	53.98	2.36	-0.11
BlowingBubbles	31.78	0.28	-0.02	40.21	2.81	-0.18	31.33	0.41	-0.03	31.59	1.93	-0.13
Johnny	56.61	2.22	-0.09	70.46	6.02	-0.25	71.99	2.94	-0.12	71.35	4.28	-0.17
KristenAndSara	59.41	1.68	-0.08	66.74	4.89	-0.24	62.14	2.21	-0.11	68.78	3.18	-0.15
PeopleOnStreet	49.59	2.36	-0.12	75.61	13.65	-0.70	44.42	1.17	-0.06	56.49	2.25	-0.11
PartyScene	42.37	0.58	-0.04	52.91	3.40	-0.24	29.68	0.30	-0.02	44.72	2.23	-0.15
Average	**52.38**	**1.19**	**-0.06**	**67.34**	**6.15**	**-0.31**	**52.27**	**1.52**	**-0.07**	**56.78**	**2.59**	**-0.13**

1) Data collection: In L-2, only one SVM model is used at each CU depth level. The number of training samples collected for L-2 SVM models is maintained at 1000 per depth level. Similar to L-1 models, SVMs are re-trained after a certain number of predictions (i.e., in L-2, 1:200 ratio), allowing the prediction models to be dynamic and content adaptive.

2) Feature selection: Let $F^i_{j=2}$ be the set of features extracted for i^{th} CU depth level in j^{th} SVM level. Thus, for L-2, $F^i_{j=2}$ is defined as,

$$F^i_{j=2} := \{\theta^i, \pi^i, \tau^i\}, \tag{3}$$

where θ^i, π^i, and τ^i refer to texture, context, and coding information, respectively [4].

3) Weight calculation: L-2 only uses one SVM per CU depth level, hence, the CUs are classified either split or non-split. The weight parameters are calculated using a F-score which is given by, $\mathrm{F-score} = {}^{2\times(\chi * \omega)}/_{(\chi+\omega)}$, where χ and ω are precision and recall, respectively. Here, recall ω is calculated using $\omega = {}^{T_p}/_{T_p+F_n}$.

4) Complexity control parameter (δ): The proposed method introduces a complexity control parameter to allow $\delta\%$ number of CUs that reach L-2 models, to go through traditional RD-optimization, to determine their CU split decisions. Controlling δ in the proposed algorithm, facilitates the flexibility to trade-off coding complexity to the coding efficiency.

III. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed algorithm has been implemented in HM16.1 reference encoder [6] using *libSVM* [7] to handle the SVM functionalities. In addition, Radial Basis Function has been used as the kernel function due to its higher accuracy when dealing with small number of features, and its ability to handle non-linear decision boundaries. The parameter C is maintained at $C = 100$ to achieve better generalization and two coding complexity trade-off parameter values $\delta\% \in \{100, 20\}$ have been used to evaluate the proposed algorithm.

The performance of the proposed method under *all intra main* encoding configuration has been compared against two state-of-the-art methods in the literature, i.e., [4], [5] and is illustrated in the Table I. The encoding time performance $\Delta T(\%)$ is evaluated using,

$$\Delta\mathrm{T}(\%) = \frac{T_{HM} - T_\rho}{T_{HM}} \times 100, \tag{4}$$

where T_{HM}, and T_ρ are the encoding times of the HM reference encoder and the evaluating algorithm, respectively, for $\mathrm{QP} \in \{22, 27, 32, 37\}$. The impact on coding efficiency for the proposed and state-of-the-art methods is measured using the Bjøntegaard Delta Bit Rate (BDBR) increase [8].

It can be observed that the proposed algorithm with δ=100% achieves an average encoding time saving of 52.38% with a negligible BDBR loss, which outperform the state-of-the-art methods. The generation of SVM models online during the encoding, maintains the BDBR increase in proposed algorithm at 1.19%, compared to BDBR increases in [4] and [5], which use offline trained rigid inference models. In addition, the experimental results with δ=100% and δ=20% demonstrate the proposed algorithm's capability to trade-off the coding efficiency to the encoding complexity.

IV. CONCLUSION

The experimental results of the proposed algorithm demonstrate an average encoding time saving of up to 52.38%, with only 1.19% average BDBR increase compared to the HM16.1. The future work will focus on extending the framework for fast content adaptive HEVC inter coding.

REFERENCES

[1] G.J.Sullivan, J. Ohm, W-J Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] S. Cho and M. Kim, "Fast CU splitting and pruning for suboptimal cu partitioning in hevc intra coding," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 9, pp. 1555–1564, 2013.

[3] T. Mallikarachchi, A. Fernando, and H. Kodikara Arachchi, "Efficient coding unit size selection based on texture analysis for HEVC intra prediction," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2014, pp. 1–6.

[4] Y. Zhang, Z. Pan, N. Li, X. Wang, G. Jiang, and S. Kwong, "Effective Data Driven Coding Unit Size Decision Approaches for HEVC INTRA Coding," *IEEE Trans. on Circuits and Syst. for Video Technol.*, 2017.

[5] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "CU Partition Mode Decision for HEVC Hardwired Intra Encoder Using Convolution Neural Network," *IEEE Trans. on Image Proc.*, vol. 25, no. 11, pp. 5088–5103, 2016.

[6] "HM software manual [online] Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware," .

[7] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[8] G. Bjontegarrd, "Calculation of average PSNR differences between RD-curves," *ITU - Telecommunications Standardization Sector STUDY GROUP 16 Video Coding Experts Group (VCEG)*, 2001.