MACHINE LEARNING BASED EFFICIENT QT-MTT PARTITIONING FOR VVC INTER CODING

A. Tissier**, W. Hamidouche***, J. Vanne*† *and D. Menard**

* Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, Rennes, France.
† Computing Sciences, Tampere University, Tampere, Finland.
Emails: alexandre.tissier2@insa-rennes.fr, whamidou@insa-rennes.fr,
jarno.vanne@tuni.fi and dmenard@insa-rennes.fr

ABSTRACT

The Joint Video Experts Team (JVET) have standardized the Versatile Video Coding (VVC) in 2020 targeting efficient coding of the emerging video services and formats such as 8K and immersive video streaming applications. VVC standard enhances the coding efficiency by 40% at the cost of an encoder computational complexity increase estimated to 859%(x8) compared to the previous standard High Efficiency Video Coding (HEVC). This work aims at reducing the complexity of the VVC encoder under the Random Access (RA) configuration. The proposed method takes advantage of the inter prediction in order to predict the split probabilities through a convolutional neural network. Our solution reaches 31.8% of complexity reduction for a negligible bitrate increase of 1.11% outperforming state-of-the-art methods.

***Index Terms*—** Versatile Video Coding (VVC), complexity reduction, Machine Learning (ML), Convolutional Neural Network (CNN), Decision Tree (DT).

1. INTRODUCTION

The video content visualisation has been revolutionized in a decade with the emergence of new services like video on demand, video streaming or video sharing platforms. Indeed, video represents the majority of data traffic over Internet as mentioned in the Cisco's study [1]. The emerging video formats such as 8K, high-frame rate (HFR), 360° video, multi-view, light field, point clouds increase the quality of experience through the resolution, the frame rate or the depth but at the expense of a significant increase in the amount of data to transmit. The emerging video formats alongside with the explosion of IP video traffic [1] require new video compression techniques that are even more efficient than existing ones. The ISO/IEC, ITU-T Joint Video Experts Team (JVET) designs a new video compression standard named Versatile Video Coding (VVC) ITU-T H.266 | MPEG-I - Part 3 (ISO/IEC 23090-3) [2] in July 2020 as a successor to High Efficiency Video Coding (HEVC).

This new standard brings impressive coding quality and subjective metric [3] compared over HEVC but at the cost of a high encoding complexity increase [4]. The VVC coding performance is enabled by the new tools introduced and especially by the Multi-Type Tree (MTT) process [5] which brings 8.5% coding efficiency gain in Random Access (RA) configuration compared to HEVC. Indeed, compared to HEVC, which is based on a Quad-Tree (QT) block partitioning, VVC integrates a nested MTT partitioning scheme allowing in addition to QT, horizontal and vertical Binary-Tree (BT) and Ternary-Tree (TT) splits [6]. This new partitioning process introduces an important complexity increase and thus offering 97% of complexity reduction opportunity [7]

In this paper, an efficient complexity reduction method is proposed relying on Machine Learning (ML) algorithms under RA configuration. First, a Convolutional Neural Network (CNN) predicts spatial features of the input $128 \times 128 \times 3$ luminance pixel values which contain the current Coding Tree Unit (CTU) and inter prediction information. The predicted vector of features represents the partition of the processed CTU. Then, to leverage the informations provided by the CNN and give more accurate decision, Decision Trees (DTs) are exploited at each depth level. Several DTs are trained to predict the probabilities of each split in a multi-classification task. Finally, the number of split tested or skipped depends on the targeted complexity reduction-Bjøntegaard Delta Bit Rate (BD-BR) trade-off. The splits with the highest probabilities are tested by the VVC Test Model (VTM) encoder to minimize the BD-BR loss and maximize the complexity reduction.

In order to compare our results, two best performing state of the art methods are selected. The first method proposed by Tang *et al.* [8] reduces the inter coding complexity through partitioning pruning. A handcrafted metric based on a difference of pixel value between the current frame and its reference frames is defined and compared to a threshold to early terminate the unlikely splits. The second method, proposed by Pan *et al.* [9], reduces the complexity with binary classification deciding the split or the early termination. This

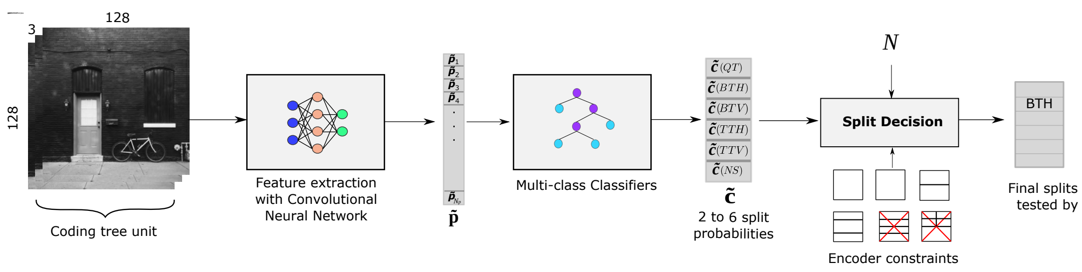

Fig. 1. Workflow diagram of the proposed fast block partitioning scheme for VVC under RA configuration. The current luminance CTU $\mathbf{B}_{cur}$ plus its two CTUs of reference $\mathbf{B}_{MV1}$ and $\mathbf{B}_{MV2}$ which are processed by a CNN to predict $\tilde{\mathbf{p}}$, a vector of N_p probabilities describing all edges at each 4×4 block. The vector $\tilde{\mathbf{p}}$ is then processed by the MCC to predict probabilities of the six partitioning options through the vector $\tilde{\mathbf{c}}$. The top-N splits of the highest probabilities are tested by the RD optimization of the encoder to select the optimal split in terms of RD-cost.

binary classification is the output of a multi-information fusion CNN that takes as input the current block, the residue and the bi-directional motion field.

The remainder of this paper is organized as follow. The proposed method is described in with its ML algorithms in Section 2. The training parameters of the CNN and DT models are presented in Section 3 alongside with the dataset. The performance evaluation of our solution in terms of both complexity reduction and BD-BR loss is proposed in Section 4. Finally, Section 5 concludes the paper.

2. PROPOSED METHOD

To reduce the complexity of the VTM under inter coding, a method based on ML technique is proposed to skip unlikely splits. Our solution is composed of two parts, first a CNN computes spatial features related to the partitioning and then MCCs leverage these spatial features to predict the split probabilities.

2.1. Overall presentation

The proposed method reduces the complexity by skipping unlikely splits through ML techniques inside the VTM encoder process. Fig. 1 details the global scheme of the proposed method to reduce the complexity of the VTM under the RA configuration. First, a CNN predicts a vector of spatial features that represent the CTU partitioning. Second, the MCC determines a probability for each available split based on the features of the processed block. Finally, relying on predefined configuration, the VTM performs the top-N best splits to maximize the trade-off between the complexity and the BD-BR loss.

2.2. Description of the inputs

These ML techniques are taking advantage of inter prediction and the pixels of the current block. Let $\mathbf{B}_{cur}$, the considered CTU, a block of 128×128 pixels. $\mathbf{B}_{MV1}$ and $\mathbf{B}_{MV1}$ are defined to describe the two CTUs of reference that correspond to the two Motion Vectors (MVs) with the lowest RD-costs.

A pre-process is necessary to obtain the three components $\mathbf{B}_{cur}$, $\mathbf{B}_{MV1}$ and $\mathbf{B}_{MV2}$ of the CNN input. Before the partitioning process, the inter prediction method computes different blocks from encoded frames to find the most related block and estimates the RD-cost of the current CTU. Indeed, the two MVs with the lowest RD-costs are used to find the two most related block of size 128×128, $\mathbf{B}_{MV1}$ and $\mathbf{B}_{MV2}$. This is the classical process of the VTM, thus no complexity overhead is added. The pixels of those two blocks are selected to bring information of the inter prediction. Finally, the luminance pixels, which results from blocks $\mathbf{B} = \{\mathbf{B}_{cur}, \mathbf{B}_{MV1}, \mathbf{B}_{MV2}\}$ of dimension $128 \times 128 \times 3$, are given to the CNN as input in order to extract features related to the partition of the current CTU.

2.3. Feature extraction with CNN

To extract the features required to predict the unlikely splits, a CNN process the 3-component input $\mathbf{B} = \{\mathbf{B}_{cur}, \mathbf{B}_{MV1}, \mathbf{B}_{MV2}\}$ in order to take advantage of the inter prediction. The CNN outputs the N_p-length vector $\tilde{p} = \left[p_0, \ldots, p_i, \ldots, p_{N_p-1}\right]^T$ where each component p_i represents the probability associated to the 4×4 edge i of the CTU. The vector length N_p, is equal to 1984 for a CTU size of 128×128. The features extraction block processes the 3-components input $\mathbf{B}$ to predict the vector of probabilities $\tilde{\mathbf{p}}$:

$$\tilde{\mathbf{p}} = f_\theta(\mathbf{B}_{cur} \oplus \mathbf{B}_{MV1} \oplus \mathbf{B}_{MV2}). \tag{1}$$

where f_θ is a parametric function of the CNN with training parameters θ and $\oplus$ stands for the concatenation operation.

2.4. Split probability determination with MCC

The MCC are fed with the probabilities vector $\tilde{\mathbf{p}}$ derived from the CNN to predict the split decision performed at each depth based on the following expression

$$\tilde{\mathbf{c}} = g_{\omega_i}(\tilde{\mathbf{p}}), \quad \forall i \in \{1, \ldots, M\}, \tag{2}$$

Table 1. Possible splits according to the CU size

Height \ Width	128	64	32	16	8	4
128	QT, BT	BT			-	
64	BT	*All*	BT, TT		BT, TTH	BTH, TTH
32		BT, TT	*All*	BT, TT		
16			BT, TT	*All*		
8	-	BT, TTV			BT	BTH
4		BTV, TTV			BTV	-

where g_{ω_i} is a parametric function of a classifier i, ω_i is its training parameters and M is the number of considered classifiers.

One classifier is available for each block size. Indeed, for a 128×128 block decision, the MCC takes as input the whole vector with the 1984 features. However, with a 8×4 block, the MCC takes as input only the edge inside this block which results in 1 feature. Based on the considered features, the MCC predicts the split probabilities. These split probabilities are gathered inside the vector $\tilde{\mathbf{c}}$. The splits supported in $\tilde{\mathbf{c}}$ are QT, BTH, BTV, TTH, TTV, and no split.

2.5. Split decision

Finally, the VTM encoder selects the top-N best splits predicted in the vector $\tilde{\mathbf{c}}$. However, the encoder has constraints on the available splits which depend of the Coding Unit (CU) size and the previously computed split. Tab. 1 details the splits available depending on the CU size under RA configuration. Based on these constraints, a selection of the top-N splits are then proposed, depending on the selected configuration. For instance, if the top-2 configuration is selected, the two highest probabilities from $\tilde{\mathbf{c}}$ which are available relying on the VTM constraints are tested by the encoder. As the unlikely splits are not performed, the complexity is reduced with the aim to limit the BD-BR loss.

3. TRAINING PROCESS AND EXPERIMENTAL SETUP

This section presents the dataset alongside the CNN and MCC training process with their respective frameworks and parameters. The experimental setup performed to obtain the complexity reduction and BD-BR on the VTM10.2 are also described.

3.1. Dataset

In RA configuration, the temporal relationship between frames has a major impact in the encoding performance. As previously presented, our method exploits the temporal coding by taking advantage of the MV computed for the CTU. Two video datasets were selected with UVG [10] and Xiph [11] to generate the dataset for the learning process.

Table 2. The configurations performed in the VTM depending on the number of split performed.

Configuration	Block size					
	128	64	32	16	8	4
$C1$	top-4	top-4	top-4	top-4	top-4	top-4
$C2$	top-4	top-4	top-3	top-3	top-3	top-3
$C3$	top-3	top-3	top-3	top-3	top-3	top-3

All those sequences brought a wide variety of contents and resolutions. To create the dataset and generate the ground truth to train the CNN and DT models, the sequences are encoded with the VTM10.2 under RA configuration at Quantization Parameter (QP) 22, 27, 32 and 37.

For the learning process of our ML models, the dataset has an important impact on the performance. The disparity of the dataset is one of the key aspect to maximize the performance especially its generalisation capability on unseen data. An optimization on the disparity is conducted on the dataset which leads a dataset equally distributed over both QPs and depth partitioning.

3.2. CNN model

The CNN used in our method is based on the MobileNetV2 architecture [12] which is designed for mobiles with limited computing resources. Indeed, this model decreases the number of operations and the memory resources which is a key aspect for our problem. A modification of the original architecture is proposed in our method to predict the 1984 features related to the partitioning. Moreover, the QP which has an important impact on the partitioning is given as an external input to the fully connected layer.

The framework used to train the CNN is Keras [13] running on top of Tensorflow module [14]. The MobileNetV2 CNN weights are pre-trained on ImageNet. The pre-trained weights are then updated at each batch iteration with the ADAM stochastic gradient descent optimizer [15]. The batch size is set to 256 instances of one CTU and the learning rate is equal to 10^{-3}. The loss function optimizes the weights by minimizing the mean squared error. A random shuffle of the dataset is applied at each epoch.

3.3. MCC training parameters

The MCC is based on the LightGBM (LBGM) framework [16] version 2.3.1. The LBGM MCC model is the gradient boosting decision tree method described in [16] which has a fast computation which is essential as the inference is performed for each CU. Indeed at each CU, the MCC determines the top split through the vector of probabilities. The training of the MCC is optimized through the minimization of the cross-entropy loss function. Different parameters are defined with the number of boosting iterations and the early stopping process set at 100000 and 1500, respectively.

Table 3. ΔET and BD-BR performance of the proposed solution in comparison with state-of-the-art techniques.

Class	Tang *et al.* [8]		Pan *et al.* [9]		Ours $C1$		Ours $C2$		Ours $C3$	
	BD-BR	ΔET	BD-BR	ΔET	BD-BR	ΔET	BD-BR	ΔET	BD-BR	ΔET
Class A1	-	-	2.69%	35.7%	1.12%	41.9%	1.81%	51.1%	3.20%	63.8%
Class A2	-	-	3.88%	27%	1%	34.7%	1.86%	44.6%	2.91%	55.9%
Class B	1.80%	34.9%	3.07%	27.9%	1.14%	35.4%	2.21%	46.5%	3.09%	58.2%
Class C	1.72%	32.8%	2%	21%	1.35%	27.5%	3.20%	43.1%	3.79%	53.8%
Class D	0.4%	25.2%	1.73%	17%	1.04%	21.4%	3.02%	36.8%	3.26%	45.2%
Class E	1.47%	25.4%	1.85%	25.5%	0.91%	32.3%	1.45%	38.7%	2.20%	49.6%
Average	1.30%	29.5%	2.52%	25.3%	1.11%	31.8%	2.33%	43.4%	3.12%	54.3%
Class F	1.02%	32.7%	-	-	1.65%	31.3%	2.96%	40.2%	3.79%	51%

3.4. Experimental setup

Our solution is assessed across the Common Test Conditions (CTC) sequences in terms of both BD-BR [17] and computational complexity which is defined as follow

$$\Delta ET = \frac{1}{4} \sum_{QP_i \in \{22,27,32,37\}} \frac{T_R(QP_i) - T_C(QP_i)}{T_R(QP_i)}, \quad (3)$$

with the reference encoding time T_R and the encoding time with the proposed complexity reduction method T_C.

The proposed method is integrated in the VTM10.2 encoder describing the VVC encoding process under RA configuration. All encodings were carried out sequentially on an Intel Xeon E5-2603 v4 processor running at 1.70 GHz under Ubuntu 16.04.5 operating system (OS).

4. EXPERIMENTAL RESULTS

Based on the experimental setup, this section details the results obtained by our proposed method and the state-of-the-art techniques [8], [9] implemented in the VTM 10.2 under RA configuration. Tab. 2 details three configurations used in our solution to reduce the VTM complexity. These configurations allow different complexity reduction-BD-BR trade-offs. Such as presented earlier, the top-N means that the N top split probabilities predicted by the MCC are likely and that those splits will be tested by the encoder.

Tab. 3 depicts our results for configurations $C1$ to $C3$ compared with two state-of-the-art methods [8], [9]. In average, among our configurations, $C1$ reaches the lowest complexity reduction with 31.8% for 1.11% BD-BR loss. $C2$ configuration brings 43.4% complexity reduction for 2.33% BD-BR loss. Finally, the highest complexity reduction is 54.3% for a BD-BR loss of 3.12% is achieved by the $C3$ configuration.

Compared to the two state-of-the-art methods [8], [9], our lowest complexity reduction performance with the $C1$ configuration has higher complexity reduction for lower BD-BR loss. Indeed, the $C1$ configuration enables gains of 0.19% and 1.41% BD-BR compared to [8] and [9], respectively. In terms of complexity reduction, $C1$ enables gains of 2.3% and 6.5% compared to [8] and [9], respectively. With approximately the same BD-BR loss our $C2$ configuration increases the complexity reduction by 20% compared to [9].

The results comparing the classes shows that the best trade-off between complexity reduction and BD-BR loss is reached on class A1 with 51.1% complexity reduction for 1.81% BD-BR loss, which compared to [9] has 15.4% more complexity reduction with 0.88% less BD-BR loss. As shown in the table, the high resolution classes has better results than the lowest one. For instance, the class D which is the lowest resolution has 21.4% complexity reduction for 1.04% BD-BR loss. This is due to the dataset which is not equally distributed in terms of resolutions. Class F has specific contents such as screen content which does not appear in our dataset, which contains mainly natural scene sequences.

5. CONCLUSION

In this paper, a two-stage ML method is proposed to reduce the complexity of the VTM encoder under RA configuration while limiting the BD-BR loss. This solution takes advantage of the MV computed before the partitioning process. First, the CNN takes as input the current CTU $\mathbf{B}_{cur}$ plus two reference CTUs $\mathbf{B}_{MV1}$ and $\mathbf{B}_{MV2}$ obtained by the two MVs with the lowest RD-cost. Then, for each CU, a MCC predicts the split probabilities based on the features defined by the CNN. Finally, a top-N predicted splits are tested by the encoder to reach different trade-off between complexity reduction and BD-BR loss.

Our $C1$ solution brings 31.8% complexity reduction for 1.11% BD-BR loss on the CTC. This result outperforms in both complexity reduction and BD-BR loss the two state-of-the-art techniques. The results on the CTC classes and on the Full HD and 4K sequences reach different trade-offs of complexity reduction and BD-BR loss. Indeed, higher complexity reduction for a lower BD-BR loss is enabled for high resolution sequences. To limit this gap between the resolutions, the dataset heterogeneity is an important factor as the lowest resolution contains a negligible number of CTUs.

6. REFERENCES

[1] Cisco, "Cisco visual networking index : Forecast and trends, 2017-2022," .

[2] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[3] Naty Sidaty, Wassim Hamidouche, Olivier Deforges, Pierrick Philippe, and Jerome Fournier, "Compression performance of the versatile video coding: HD and UHD visual quality monitoring," in *2019 Picture Coding Symposium (PCS)*. pp. 1–5, IEEE.

[4] Frank Bossen, Xiang Li, and Karsten Suehring, "AHG report: Test model software development (AHG3)," in *Document JVET-T0003 Twentieth meeting*.

[5] E François, M Kerdranvat, R Julian, C Chevance, P DeLagrange, F Urban, T Poirier, and Y Chen, "Vvc per-tool performance evaluation compared to hevc," in *IBC*, 2020.

[6] A. Wieckowski, J. Ma, H. Schwarz, D. Marpe, and T. Wiegand, "Fast partitioning decision strategies for the upcoming versatile video coding (VVC) standard," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 4130–4134.

[7] A. Tissier, A. Mercat, T. Amestoy, W. Hamidouche, J. Vanne, and D. Menard, "Complexity reduction opportunities in the future VVC intra encoder," in *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. pp. 1–6, IEEE.

[8] Na Tang, Jian Cao, Fan Liang, Jun Wang, Hongmei Liu, Xiaoyang Wang, and Xiaorong Du, "Fast CTU partition decision algorithm for VVC intra and inter coding," in *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. pp. 361–364, IEEE.

[9] Zhaoqing Pan, Peihan Zhang, Bo Peng, Nam Ling, and Jianjun Lei, "A cnn-based fast inter coding method for vvc," *IEEE Signal Processing Letters*, vol. 28, pp. 1260–1264, 2021.

[10] Alexandre Mercat, Marko Viitanen, and Jarno Vanne, "UVG dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*. pp. 297–302, ACM.

[11] Chris Montgomery and others, "Xiph. org video test media (derf's collection), the xiph open source community, 1994," *Online, https://media. xiph. org/video/derf*, vol. 3.

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[13] F. Chollet et al., *Keras*.

[14] F. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," p. 19.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980 [cs]*.

[16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems 30*, p. 9.

[17] Gisle Bjøntegaard, "Calcuation of Average PSNR Differences Between RD-curves," in *Document VCEG-M33 Thirteenth Meeting: Austin, Texas, USA*, April 2001.