EFFICIENT HEVC-TO-VVC TRANSCODER BASED ON A BAYESIAN CLASSIFIER FOR THE FIRST QUADTREE DEPTH LEVEL

D. García-Lucas[1], *G. Cebrián-Márquez*[2], *A. J. Díaz-Honrubia*[3], *T. Mallikarachchi*[4] *and P. Cuenca*[1]

[1] High Performance Networks and Architectures, University of Castilla-La Mancha, Albacete, Spain
[2] Department of Computer Science, University of Oviedo, Oviedo, Spain
[3] ETS de Ingenieros Informáticos, Universidad Politécnica de Madrid, Madrid, Spain
[4] School of Technologies, Cardiff Metropolitan University, Cardiff, United Kingdom

ABSTRACT

In the coming years, the Versatile Video Coding (VVC) standard will be launched to replace the current High Efficiency Video Coding (HEVC) standard, making it necessary to find efficient methods to convert existing multimedia content to the new format. However, transcoding is a complex pipeline composed of a decoding and an encoding process that involves long processing times. On the basis of the existing correlation between the block partitioning structures of both standards, this paper presents an HEVC-to-VVC transcoding scheme. The proposed method consists of a Naïve-Bayes classifier that assists the partitioning decision at the first level of quadtree by using features extracted from the 128×128 pixel blocks of the residual and reconstructed frames in HEVC. The experimental results using random access configuration show an average transcoding time reduction of 13.38% at the cost of a compression efficiency loss of 0.32% in terms of BD-rate.

Index Terms— HEVC, H.265, VVC, Transcoding, MTT.

1. INTRODUCTION

In recent years, the H.264/Advanced Video Coding (AVC) standard has been the predominant codec in the video industry until it has been gradually replaced by the High Efficiency Video Coding (HEVC) since its launch in 2013 [1]. HEVC doubles the compression performance of H.264/AVC, especially for high definition (HD) and ultra-high definition (UHD) content, but at the cost of a great increase in processing times [2]. However, the demand for multimedia content is growing exponentially year after year, exceeding 80% of total Internet traffic, according to predictions for 2022 [3]. In addition, new image formats are emerging, such as 4K, 8K and high dynamic range (HDR). To meet these demands, the Joint Video Experts Team (JVET) was formed in October 2015 to start the development of the Versatile Video Coding (VVC) standard, with a compression capability that significantly surpasses the one achieved by HEVC, albeit introducing high computational costs into the encoding process.

Taking advantage of the compression capability of VVC and the large amount of content encoded using HEVC, a heterogeneous transcoder from HEVC to VVC brings value to many applications and provides interoperability between the two standards. This type of heterogeneous transcoders has been of great relevance in the literature. On the one hand, there are fast transcoders between standards such as the one proposed by J.-F. Franche and S. Coulombe in 2018 [4]. This H.264-to-HEVC transcoder can speed up the coding time by 11.77× with a penalty of 3.82% in terms of the Bjøntegaard delta rate (BD-rate) [5]. To do this, they propose a motion propagation algorithm that reuses information in different partition sizes and a fast mode decision framework to determine whether a CU must be split or not. On the other hand, there are also proposals involving royalty-free video codecs. In 2017, X. Li et al. proposed four models for different depth and quantization parameter (QP) values using Naïve-Bayes classifiers implemented in a machine-learning-based VP9-to-HEVC transcoder [6]. With an average BD-rate penalty of 2.8%, this proposal achieves a 44% time reduction compared with the full VP9-to-HEVC transcoder. Finally, a transcoder based on CU depth inheritance between HEVC and AV1 was presented in 2019 [7]. Aiming to identify possible correlations between coding units (CUs) and block size decisions performed by the two codecs, an average time saving of 35.41% is achieved at the cost of a 4.54% BD-rate penalty.

The coding efficiency gain achieved by VVC is due to the integration of new coding tools that considerably increase the computational cost. Among them, the new block partitioning scheme known as the multi-type tree (MTT) replaces the quadtree (QT) structure of HEVC. MTT is based on two splitting stages: firstly, a QT is used to split the initial CU into four sub-CUs of equal size recursively, and secondly, the leaf nodes of the QT are split horizontally or vertically by the use of binary trees (BTs) and ternary trees (TTs), respectively. The MTT partitioning scheme of VVC features a maximum block size of 128×128 pixels, while in HEVC the maximum size is 64×64 pixels. Therefore, there is no direct relationship between a 128×128 block in VVC and any of the blocks in HEVC.

To address this, the first HEVC-to-VVC transcoding scheme is proposed in this paper, introducing a prediction model using Naïve-Bayes classifier to assist the QT partitioning at the first level depth, i.e., whether to split the 128×128 block into 4 sub-blocks of 64×64 pixels or not. To build the model, different statistical information from the HEVC bitstream has been used from both the residual and the reconstructed images of the HEVC Test Model (HM) 16.16 decoder [8]. Since the most demanded services are on-demand video and live streaming, the model was built from information of sequences encoded using random access (RA) configuration. The proposed algorithm, which has been implemented in the VVC Test Model (VTM) 2.0.1 encoder [9], achieves an average transcoding time reduction of 13.38% with a compression penalty of only 0.32% in terms of the BD-rate, compared to the anchor transcoder.

This work was supported by the Spanish Ministry of Science, Innovation and Universities, and the European Commission (FEDER funds) under project RTI2018-098156-B-C52, by the Regional Government of Castilla-La Mancha under project SBPLY/17/180501/000353, and by the Spanish Ministry of Education, Culture and Sports under Grant FPU16/05692.

2. HEVC-TO-VVC TRANSCODING ALGORITHM

Our proposal consists of a probabilistic prediction model that assists the partitioning of 128×128 pixel blocks. This section includes the details of the algorithm design process, including the analysis of the variables, the generation of the dataset, the training of the prediction model and the integration of the algorithm in the transcoder.

2.1. Data understanding

A large set of features and statistical information numbered from V_1 to V_{16} has been extracted from the HEVC bitstream and from the transcoding process itself of 128×128 pixel blocks. This information can potentially describe the characteristics of the block and texture of the image to make an accurate decision, based on the residual and the reconstructed images [10]. The initial set of features contains the following information for each block:

- Average of the block: calculated for the samples in the 128×128 residual block (V_1), which can describe the complexity of the prediction obtained for the current block.
- Variance of the block: variance of the samples in the 128×128 block, both in the residual frame (V_2) and in the reconstructed image (V_9).
- Variance of the means in sub-blocks: the 128×128 residual block is divided into four blocks of size 64×64. The mean of the residual values of each 64×64 is calculated, and then the variance of these means (V_3).
- Variance of the variances in sub-blocks: similar to the previous statistic, the variance of the variances of each 64×64 residual block is calculated (V_4).
- Fisher coefficient of skewness: measure the lack of symmetry of a set of values based on their distribution around the average. It has been calculated for the 128×128 block in both the residual frame (V_5) and the reconstructed image (V_7).
- Mean absolute deviation: the amount of deviation that occurs around the mean in the samples of a block has been calculated for the 128×128 block in both the residual frame (V_6) and the reconstructed image (V_8).
- Number of zero values: the complexity of the prediction for a block can be estimated with the amount of zero values in the residual of the 128×128 block (V_{10}).
- Coefficient of Kurtosis: measure the concentration of values close to the average, which has been calculated for the 128×128 block in both the residual frame (V_{11}) and the reconstructed image (V_{12}).
- Spatial index (SI) of the 128×128 block: the SI feature defines the level of detail in the block, i.e., whether it is a complex region of the frame or a homogeneous zone, so it has been calculated only in the reconstructed image (V_{13}), using the Sobel filter.
- Cost in bits of the block in the HEVC stream (V_{14}).
- Number of pixels in the frame (width × height) of the sequence to which the 128×128 block belongs (V_{15}).
- Lambda value used to encode the frame (V_{16}). This depends on the QP and the temporal layer of the frame in the hierarchy established by the group of pictures (GOP) configuration.

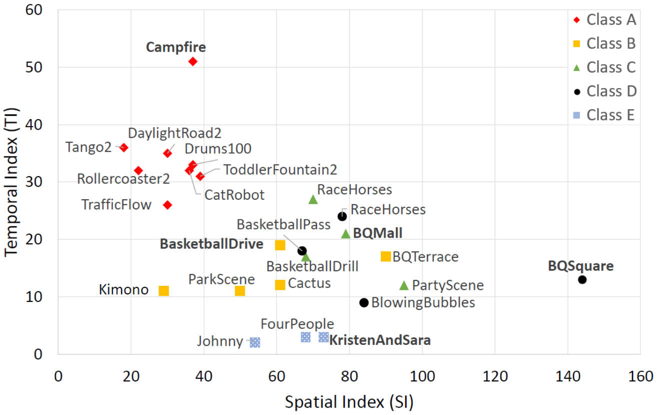

Fig. 1. SI and TI of the test sequences.

2.2. Generation of the dataset

A dataset must contain different information sources so that the model can adapt to different scenarios; in our case, video sequences with different resolutions and content. For this reason, the training set has been elaborated by choosing five sequences (one per class) from those specified in the framework of common testing conditions of the JVET group [11].

Having defined the features used in the construction of the proposed decision model, it is necessary to describe the process followed to generate the instances of the dataset. Ideally, a dataset should contain instances from as many different scenarios as possible to ensure the adaptivity of the model to any context and input sequence. The criterion used in the selection consisted of taking one sequence per class on the basis of their SI and temporal index (TI) [12]. On the basis of Fig. 1, which shows the distribution of these indices for all sequences, we selected Campfire (Class A), BasketballDrive (Class B), BQMall (Class C), BQSquare (Class D), and KristenAndSara (Class E).

With the aim of homogenizing the number of instances per class and avoiding overfitting due to the significant difference in resolution, only 1,000 instances per temporal layer belonging to hierarchical B frames and encoded sequence were selected. The instances not used for training and those corresponding to the remaining sequences were left for evaluation. In addition to the type of content being transcoded, it is also important to consider configuration parameters that may affect the prediction. In particular, on the basis of the JVET document, we considered four QP values to generate the dataset, namely 22, 27, 32 and 37, which cover a wide range of rate-distortion scenarios. The class attribute, which represents the value to be predicted by the model, was obtained from each 128×128 block in VTM. If the block was split in four CUs using a QT, its value would be 1, and 0 otherwise.

Based on the above considerations, the total number of instances in the dataset was almost 6 million instances, of which only 1.40% were used for training the model, and the remaining 98.60% for validation and evaluation.

2.3. Model generation and training

The decision model used in the first partitioning level of the QT was generated using the WEKA software [13] following a knowledge discovery from data (KDD) approach [14]. This tool, which was developed in Java, supports well-known data mining algorithms and

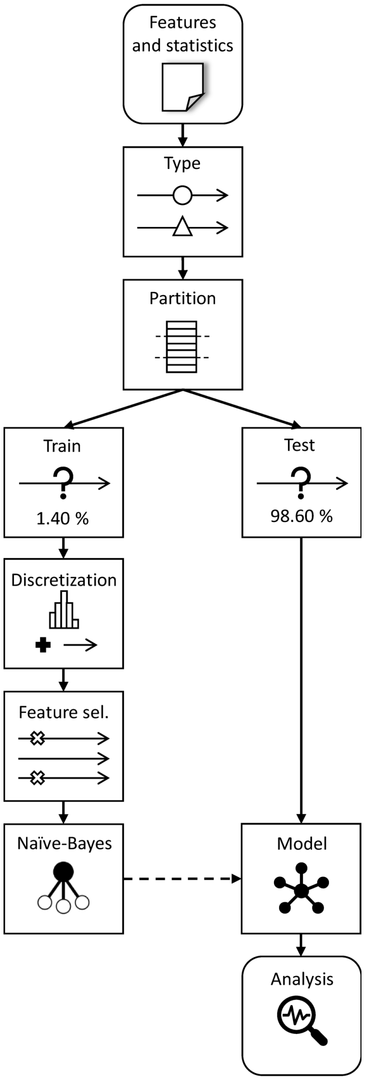

Fig. 2. Data processing and model generation flowchart.

operations such as clustering, regression and visualization. With this aim, we used the information obtained from the $128{\times}128$ blocks to generate the model.

For a better understanding of the model creation process, Fig. 2 depicts a flowchart of the different stages in the data processing, from the extraction of input information from the HEVC stream to the generation of the model. As can be seen, the first step is the characterization of each variable and attribute by its type. Different data types include numeric, nominal, string or dates. In our case, all the information extracted from the blocks is numeric in all cases except for the class attribute. The second step consists in splitting the instances in training and testing datasets as previously specified.

Once the datasets have been created, the next step is the generation of the model from the training set. Among all the possible classifiers, we selected Naïve-Bayes, which is based on the idea that an event occurs after other events that may have an influence on the former, but that are independent of each other once the class is known. Mathematically this is expressed as the factorization by the probability of the class multiplied by the probability of each variable given the class, i.e. given a class Y and a set of variables $\{X_1, \ldots, X_N\}$, the following expression is satisfied:

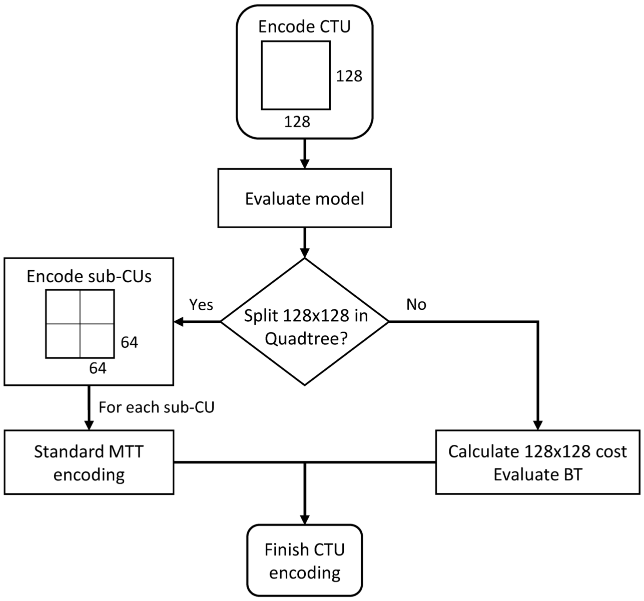

Fig. 3. Encoding process of the HEVC-VVC transcoder.

$$P(Y|X_1, \ldots, X_N) \propto P(Y) \cdot P(X_1|Y) \ldots P(X_N|Y)$$

This set of frequencies is computed in only one reading, and thus the computational complexity of building a Naïve-Bayes classifier is $\mathcal{O}(Nn)$, where N is the number of instances and n the number of features [15]. In addition, Naïve-Bayes is linear in its classification phase, i.e. $\mathcal{O}(n)$, becoming one of the fastest classifiers available.

To ensure a correct classification of the model, the training dataset requires prior preprocessing, including feature discretization and selection. Without any prior preprocessing, the accuracy of the model employing a 5-fold cross validation on the training set using a Naïve-Bayes classifier is only 79.01%. Given that the input attributes are continuous quantitative variables, which forces us to assume that they follow a specific distribution (e.g., a normal distribution), and given that Naïve-Bayes results in better accuracy with categorical variables, all attributes are discretized into intervals whose range depends on their contribution to the class attribute [16]. After this discretization step, the accuracy of the model has been increased to 84.42%.

Building on the preprocessing of the training set, Naïve-Bayes classifiers, like other probabilistic classifiers, are sensitive to the feature sets used to induce them. Therefore, it is necessary to discern the attributes that actually contribute to the prediction of the class variable from those that are irrelevant or redundant. To this end, we selected Wrapper with forward selection to generate the corresponding training subset [17]. As a result, only two of the variables were selected, namely V_{14} and V_{12}, omitting the remaining ones and obtaining an accuracy of 92.34%. In this regard, it is worth noting the difference in terms of accuracy of the model before and after the preprocessing of the dataset.

2.4. Integration of the model in the transcoder

The coding flow of the proposed transcoder is depicted in Fig. 3. When a coding tree unit (CTU) is going to be encoded in the

Table 1. Accuracy of the proposed model.

Class	Accuracy (%)			
	QP 22	QP 27	QP 32	QP 37
A1	96.24	91.33	90.06	89.76
A2	91.90	85.30	83.83	85.73
B	94.30	88.66	85.85	86.24
C	97.64	95.23	92.96	90.05
D	99.13	95.44	89.92	87.69
E	88.47	90.33	92.32	94.37
Average	94.86	90.98	88.89	88.64

transcoder, the model is evaluated for the current 128×128 block. If the model decides to split the block, it is divided into 4 CUs of 64×64 pixels each, thus reducing the total computation time, since the evaluation of QT, BT and TT is skipped for this level. On the contrary, if the model decides not to split the 128×128 block, the evaluation of lower levels in the partitioning tree is completely omitted, resulting in significant time savings.

3. EVALUATION RESULTS

Regarding the experimental setup, the hardware platform used in the tests was composed of an Intel® Xeon® E5-2630L v3 CPU running at 1.80 GHz and 16 GB of main memory. The encoders were compiled with GCC 5.4.0-6 and executed on Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-143). Turbo Boost was disabled to achieve the reproducibility of the results. Sequences were encoded according to the common coding conditions issued by the JVET [11], RA configuration, QP values 22, 27, 32, and 37, 10-bit encoding, and 4:2:0 chroma subsampling.

On the one hand, Table 1 shows the accuracy of the testing sets for each class and QP used for the validation of the model. The results show that the accuracy achieved is significantly higher for lower QPs, while still near 90% in the case of higher QPs. Moreover, it can be seen that there is no overfitting with respect to any QP or class, since high accuracy is achieved in all cases. When considering all the instances of the testing set, the average accuracy achieved by the proposed model is 89.42%. While it is slightly lower than the accuracy obtained from the training set, it is still significantly high, and thus confirms the validity of the model.

On the other hand, Table 2 shows the results of the proposed first depth level transcoding algorithm in terms of BD-Rate and time reduction (TR). As can be observed, it achieves an average time saving of 13.38% with a negligible impact in coding efficiency of 0.32% BD-Rate, which measures the increment in bitrate while maintaining the same objective video quality [5]. In addition, the results related to the encoding time show that the implemented model performs better in high-resolution classes, that is, in classes A1, A2, B and E. This is because a 128×128 block represents a smaller part of the image compared to class C and D resolutions, and therefore the possibilities of splitting the block in QT are lower, saving longer encoding time by avoiding the evaluation of lower partitioning levels.

4. CONCLUSIONS

Table 2. Results of the proposal using RA configuration.

Class	Sequence	BD-rate (%)	TR (%)
A1	Tango2	1.07	30.38
	Drums100	0.75	16.87
	Campfire	0.06	9.83
	ToddlerFountain2	−0.05	6.50
A2	CatRobot	0.75	20.11
	TrafficFlow	0.44	24.55
	DaylightRoad2	1.35	25.52
	Rollercoaster2	0.95	30.00
B	Kimono	0.23	17.46
	ParkScene	0.17	12.50
	Cactus	0.08	12.33
	BasketballDrive	0.27	11.86
	BQTerrace	0.02	12.97
C	BasketballDrill	0.20	6.16
	BQMall	−0.06	6.92
	PartyScene	−0.11	4.78
	RaceHorsesC	−0.01	4.41
D	BasketballPass	0.66	3.40
	BQSquare	−0.19	6.50
	BlowingBubbles	0.04	4.27
	RaceHorses	0.23	3.30
E	FourPeople	0.06	15.15
	Johnny	0.33	16.36
	KristenAndSara	0.35	18.88
Per-class average	Class A1	0.46	15.90
	Class A2	0.87	25.01
	Class B	0.15	13.42
	Class C	0.01	5.57
	Class D	0.19	4.37
	Class E	0.25	16.80
Total average		0.32	13.38

In this paper, a CU partitioning decision for the first depth level based on a prediction model using the Naïve-Bayes classifier for a HEVC-to-VVC transcoder is presented. To this end, the model predicts the QT splitting decision of 128×128 blocks from information extracted from the HEVC bitstream. On the one hand, if the model decides to split the block, it is divided into 4 CUs of 64×64 pixels each, continuing with the normal coding flow of the MTT structure for each block. On the other hand, if the model decides not to split the 128×128 block, the QT partitioning ends at the first level, resulting in significant time savings since the lower QT levels are skipped. The evaluation results evince the high accuracy of the model, which reaches 89.42% for the testing set. The performance analysis of the transcoder shows the efficiency of the Naïve-Bayes classifier, given that a TR of 13.38% is achieved with a negligible penalty in terms of BD-rate, compared with the anchor transcoder.

As future work, new techniques will be implemented in our HEVC-to-VVC transcoder to achieve greater time savings. Since this proposal has been integrated into the first partitioning level of the MTT structure, new approaches will focus on accelerating the remaining depth levels. In addition to square blocks, the prediction units (PUs) in HEVC may provide meaningful information to assist BT and TT splitting decisions. Finally, proposals that focus on other encoder modules can also achieve time savings, which are compatible with the one presented in this paper, such as the intra and inter prediction modules.

5. REFERENCES

[1] ISO/IEC and ITU-T, "High Efficiency Video Coding (HEVC). ITU-T Recommendation H.265 and ISO/IEC 23008-2," Apr. 2013.

[2] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[3] CISCO, "Cisco Visual Networking Index: Forecast and Trends, 2017 - 2022," Feb. 2019.

[4] J. Franche and S. Coulombe, "Efficient H.264-to-HEVC Transcoding Based on Motion Propagation and Post-Order Traversal of Coding Tree Units," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3452–3466, Dec. 2018.

[5] G. Bjøntegaard, "Improvements of the BD-PSNR Model," Tech. Rep. VCEG-AI11, ITU-T SG16 Q6, July 2008.

[6] X. Li, R. Xie, L. Song, and L. Zhang, "Machine learning based VP9-to-HEVC video transcoding," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Jun. 2017, pp. 1–6.

[7] A. Borges, B. Zatt, M. Porto, and G. Correa, "Fast HEVC-to-AV1 Transcoding Based On Coding Unit Depth Inheritance," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 3571–3575.

[8] "JCT-VC HEVC Test Model Version - 16.16," "`https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.16/`".

[9] "JVET VVC Test Model Version - 2.0.1," "`https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM/-/tags/VTM-2.0.1`".

[10] J. M. Ha, J. H. Bae, and M. H. Sunwoo, "Texture-based fast CU size decision algorithm for HEVC intra coding," in *2016 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Oct. 2016, pp. 702–705.

[11] X. Li and K. Suehring, "Common Test Conditions and Software Reference Configurations," Tech. Rep. JVET-H1010, Joint Video Experts Team (JVET), Oct. 2017.

[12] ITU-T, "P.910 - Subjective Video Quality Assessment Methods for Multimedia Applications," Apr. 2008.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[14] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Advances in Knowledge Discovery and Data Mining," chapter From Data Mining to Knowledge Discovery: An Overview, pp. 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[15] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011.

[16] U. M. Fayyad and K. B. Irani, "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning," in *Proceedings of the International Joint Conference on Uncertainty in AI*, Aug. 1993.

[17] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.