# CODING TREE EARLY TERMINATION FOR FAST HEVC TRANSRATING BASED ON RANDOM FORESTS

†*Thiago Bubolz, Mateus Grellert**, †*Bruno Zatt,* †*Guilherme Correa*

†Video Technology Research Group (ViTech) - Federal University of Pelotas (UFPel), Brazil
*Graduate Program in Elec. and Computer Engineering - Catholic University of Pelotas (UCPel), Brazil

## ABSTRACT

Video transrating has become an essential task in streaming service providers that need to transmit and deliver different versions of the same content for a multitude of users that operate under different network conditions. As the transrating operation is comprised of a decoding and an encoding step in sequence, a huge computational cost is required in such large-scale services, especially when considering the use of complex state-of-the-art codecs, such as the High Efficiency Video Coding (HEVC). This work proposes an early-termination method for complexity reduction of the HEVC transrating based on Random Forests, which use features obtained from the HEVC decoding process to accelerate the coding tree decisions during the re-encoding process. Experimental results show that the proposed method achieves an average transrating time reduction of 47.09% at the cost of a negligible bitrate increase of 0.292%.

***Index Terms*—** HEVC, transrating, transcoding, Random Forests.

## 1. INTRODUCTION

Video compression has become fundamental to allow both storage and transmission of such kind of content, especially with the introduction of high spatial resolution and increased frame rates. With the aim of allowing compatibility among devices, services, and applications that transmit/receive digital videos on the internet, there is a need to convert encoded videos to different standards (called heterogeneous transcoding) or to change their characteristics while keeping the encoding standard (called homogeneous transcoding). Homogeneous transcoding can be employed to modify the video resolution, to insert watermarks in the video, and to change the encoded-video bitrate, for example.

Due to the increasing use of video streaming services such as YouTube and Netflix, transcoding for bitrate adaptation, also called transrating, has become essential, since such services are required to provide several versions of the same video with different bitrates to meet different user requirements and network capabilities. As transrating requires long processing times, the operation is usually performed offline and the several bitstream versions of a video are stored in servers for future requests. Often-accessed contents benefit from this strategy, since they are promptly available for users. However, rarely-accessed videos are also stored in the server in multiple versions, wasting valuable storage resources. Thus, transrating for videos seldom accessed could be performed on-the-fly, i.e., as they are requested.

High Efficiency Video Coding (HEVC) is the current state-of-the-art video coding standard, launched in 2013 by the Joint Collaborative Team on Video Coding (JCT-VC) [1]. HEVC reduces the bitrate of encoded videos by 40% on average [2] while keeping the same image quality of its predecessor, the H.264/AVC [3] standard. However, such compression rates are achieved with a significant increase in terms of computational effort, which can reach up to 500% in comparison to H.264/AVC [4].

Recent solutions found in the literature propose the use of machine learning techniques to reduce the complexity of HEVC, such as [5–8]. However, the proposed solutions lack in some aspect: either they do not achieve significant time savings, or they introduce non-negligible losses in encoding efficiency. In [9], a simple statistical-based heuristic was proposed, outperforming competing solutions in terms of encoding efficiency and time savings.

This paper presents a low-complexity HEVC transrating method focusing on high encoding efficiency and reduced computational cost. The proposed method targets video streaming providers for which transcoding time and energy consumption are relevant, given the enormous amount of information that must be processed. The method is based on a machine learning approach in which the HEVC transrating process inherits a set of features from the reference (High Bitrate – HBR) bitstream and quickly predicts the best frame partitioning during re-encodings of the same sequence with lower bitrate (LBR), speeding up the transrating without causing significant impact on compression efficiency.

## 2. PARTITIONING IN HEVC TRANSRATING

HEVC introduced flexible frame partitioning structures to allow better coding efficiency for various types of content. Each frame is partitioned into square Coding Tree Units (CTUs),

This work was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nvel Superior - Brasil (CAPES) - Finance Code 001 and also by FAPERGS and CNPq Brazilian research support agencies.

**Table 1**. Average CU correlation between HBR and LBR

| HBR CU size | LBR CU size | | | |
|---|---|---|---|---|
| | **64x64** | **32x32** | **16x16** | **8x8** |
| **64x64 (Depth 0)** | **93.09** | 5.73 | 0.86 | 0.29 |
| **32x32 (Depth 1)** | 41.81 | **53.92** | 3.70 | 0.56 |
| **16x16 (Depth 2)** | 19.10 | 22.18 | **56.08** | 2.61 |
| **8x8 (Depth 3)** | 13.99 | 16.64 | 21.55 | **47.79** |

whose size is typically 64x64. Then, each CTU is partitioned into one or more Coding Units (CUs) in a quadtree-based recursive process, in which a 64x64 CU is split into four smaller CUs, until the minimum CU size (8x8) is reached. The best partitioning is the one with smallest rate-distortion (RD) cost. Since RD-cost requires prediction and residual coding to be computed, the computational complexity involved in this decision process is extremely high [4].

In an HEVC transrating method, an HBR bitstream encoded under a certain bitrate is decoded, generating a video output that is re-encoded under a different target bitrate. Therefore, transrating is even more complex than encoding due to the extra decoding step. However, information from the HBR bitstream can be used to guide the encoding decisions in the LBR encodings. Table 1 shows the average correlation between CU depths of the HBR and the LBR bitstreams. For HBR, the bitrate was obtained with base QP 22, whereas the LBR bitrate was calculated according to the bitrate ratios for testing, shown in Table 2. In fact, the chances of a CU being re-encoded with the same size or larger is always higher than 93%.

Previous works that propose methods to reduce the complexity of HEVC transrating make use of this correlation. In [6], the authors present a method for fast HEVC spatial re-scaling, which uses the number of CU partitions in the high-resolution video to limit the partitioning decision while encoding the low-resolution video. In [7], the authors trained a machine learning model using the random forest algorithm that can predict whether a CU should be divided or not based on the co-located blocks in the video bitstream with the largest resolution (i.e., differently from this work, [7] focuses on transcoding for resolution adaptation). In [9], a simple strategy that inherits directly the CU partitioning information from the decoding step to speed up the re-encoding step is proposed. The authors in [8] propose a CU early termination solution based on three methods that use features such as motion vectors, average depths and RD costs of co-located CUs to speed up the transcoding process.

These related works achieve significant time savings but incur in non-negligible losses in terms of encoding efficiency. The method proposed in this work achieve significant time saving at the cost of tolerable encoding efficiency losses.

## 3. RANDOM FORESTS FOR CU SIZE DECISION

As most of the HEVC encoding complexity is caused by frame partitioning decisions [4], an efficient approach that allows skipping the exhaustive RD-based tests during the encoding process is essential. In this work, we propose the use of random forests to predict which modes can be skipped without compromising encoding performance.

**Table 2**. Analysis, training and test set configuration

| Codec | HEVC Model 16.4 (HM 16.4) |
|---|---|
| Configuration | Random Access |
| Base QP* | 22 |
| Bitrate Ratios for Training | 5% to 95% in steps of 5% |
| Bitrate Ratios for Testing | 80%, 60%, 40% and 20% |
| Training Sequences | ToddlerFountain, Rollercoaster, BasketballDrive, KristenAndSara, SlideEditing |
| Testing Sequences | Tango, CatRobot, TrafficFlow, DaylightRoad, Kimono, ParkScene, Cactus, BQTerrace, FourPeople, Johnny, SlideShow, ChinaSpeed |

*used to obtain the target bitrate of the HBR bitstream

Random forests are a class of machine learning techniques for general-purpose systems that address classification solutions and are less prone to overfitting compared to decision trees [7]. They consist of multiple decision trees constructed systematically by pseudo-randomly selecting subsets of features [10]. If a relevant set of input variables is used in the training process, random forests models can achieve high prediction accuracy at the cost of a small overhead in terms of computational resource usage.

### 3.1. Model Training and Evaluation

Table 2 shows the training setup of random forests including the video sequences, the base Quantization Parameter (QP), and the target bitrates used to transrate videos and extract the features/labels. The encodings followed the Common Test Conditions (CTC) [11] recommended by JCT-VC.

Before training the random forests, HBR and LBR bitstreams were generated using the reference HEVC encoder [12]. The target bitrate of the HBR encoding was assigned as the average bitrate obtained using a QP of 22. Then, the LBR target bitrates were obtained as a percentage of the HBR one, which will be referred to as bitrate ratio in the remainder of this paper. For training data, LBR bitstreams were encoded with bitrate ratios between 5% and 95% in steps of 5%.

The training data sets were then built using information extracted from the HEVC decoder. Data obtained from the HBR bitstream were used as input features, whereas the LBR CU depths were used to define the labels. Several decoding-domain variables were considered as features, such as prediction mode, partition size, and QP. The labels were assigned to *Split*, when the CU depth of the HBR bitstream is greater than that of the LBR one, and to *Unsplit* otherwise. Using this approach, one data set was built for each CU depth: 0 (64x64), 1 (32x32) and 2 (16x16). Note that depth 3 (8x8) does not require a data set because it is the maximum depth supported

in HEVC (i.e., 8x8 CUs cannot be split).

After building the training data sets, the importance of each decoding feature to predict the CU partitioning in the LBR re-encodings was assessed. In this work, the Gini Importance (GI) was used, which is the default metric of the *Scikit-Learn* toolkit [13]. The Gini Importance, calculated using (1), measures how much a given feature reduces the likelihood of an incorrect classification for a new instance.

$$GI(feat) = Imp(Y) - R(feat) \quad (1)$$

$$R(feat) = \sum_{v \in Vals} P(feat = v) * Imp(Y|feat = v) \quad (2)$$

$$Imp(Y) = \sum_{y \in Y} P(Y = y) * (1 - P(Y = y)) \quad (3)$$

In (1), the GI is computed as the impurity in the output variable *Y* (in our case, the *Split/Unsplit* variable) that remains when a given feature (*feat*) is known. Features with high GI are the ones that minimize the remainder *R* in equation (2), which is interpreted as a weighted sum of the impurity of the output variable for each value of the attribute. The obtained GI values of the extracted features for each data set is depicted in Fig. 1. Note that the GIs are similar across all classifiers. The bitrate ratio achieved the highest GI among all features, which is expected, as the encoder tends to favor less CU partitionings as the target bitrate becomes smaller.

Using the data sets as input, random forests were finally trained using the *Scikit-Learn* library [13]. The training parameters were left in their default value, except for the number of estimators (i.e., the number of decision trees in the forest) in each model. Forests with 5, 20, 50, 100, 200 and 1000 trees were trained, but cross-validation accuracy did not improve significantly with more than 20 estimators. Therefore, we defined a maximum of 20 trees to reduce prediction complexity while keeping efficient model performance. The number of estimators used in each data set as well as the 5-fold cross-validation accuracy, are shown in Table 3. The models achieved accuracy values between 81.5% and 92.2%.

### 3.2. Proposed Early Termination Algorithm

An overview of the proposed method is shown in Fig. 2, where each forest is used to decide if the CU evaluation should stop at the same depth of the input HBR bitstream (*unsplit* CU), or be further partitioned (*split*) for evaluation.

The input features are obtained when decoding the video and the classifier decisions are then used to build CU depth maps. These maps are then used to limit the CU partitioning decisions in the LBR re-encodings. For example, if a CU that was encoded at depth 1 in the HBR bitstream is classified as *Split*, the maximum depth allowed for it in the LBR encoding will be 2. Therefore, encoding time is reduced whenever the maximum depth is smaller than 3, since it means that the CU splitting process will be early terminated and the RD cost calculation for its lower levels will no longer be necessary.

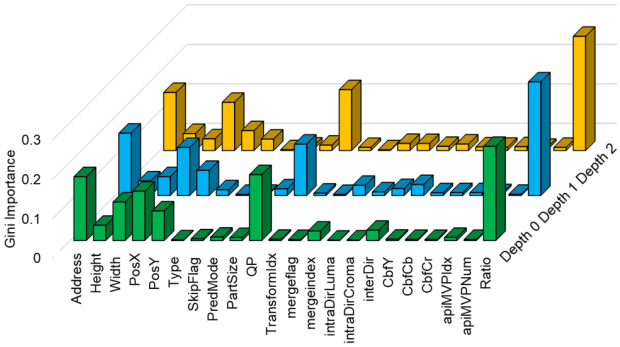


**Fig. 1**. Gini importance of features used in the RF models

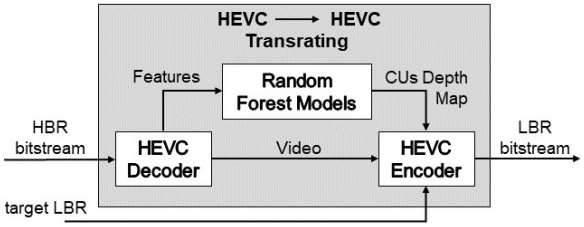


**Fig. 2**. Proposed early termination transrating scheme

## 4. EXPERIMENTAL RESULTS

The same setup presented in section 3 was employed to obtain the experimental results, except for the video sequences that are different from those to guarantee unbiased results. The testing sequences are recommended in the CTC [11] and are listed in the last row of Table 2. Notice that the 12 sequences belong to three spatial resolution classes: HD (1280x720 and 1024x768 pixels), Full HD (1920x1080 pixels) and 4K (4096x2160 and 3840x2160 pixels).

As shown in Table 2, the HBR bitstreams were encoded with the average bitrate obtained in a prior encoding using QP 22, and the LBR cases were defined as 80%, 60%, 40%, and 20% of the HBR bitrate. To evaluate the proposed method in terms of compression efficiency and transrating time, the original tandem transrating process (i.e., with no modifications on both decoder and encoder) was first executed for the 12 test sequences and the four LBR cases, yielding 48 transcodings. The proposed low-complexity transrating scheme was also executed for the 12 video sequences, taking the same four LBR target bitrates. Thus, the obtained results are comparisons between the proposed transrating strategy and the original tandem transcoder.

### 4.1. Time Savings

Table 4 shows average time savings (TS) and compression efficiency (BD-rate) results achieved by the proposed solution,

**Table 3**. Characteristics of each trained model

| Data set | Number of Estimators | Precision (%) |
|---|---|---|
| Depth 0 (64x64) | 20 | 81.50 |
| Depth 1 (32x32) | 10 | 85.49 |
| Depth 2 (16x16) | 10 | 92.19 |

**Table 4**. Experimental results

| Resolution | Sequence | BD-rate (%) | TS (%) |
|---|---|---|---|
| 4096x2160 | Tango | -0.681 | 50.7 |
| 3840x2160 | CatRobot | -2.423 | 31.5 |
| 3840x2160 | TrafficFlow | -1.426 | 32.8 |
| 3840x2160 | DaylightRoad | 0.499 | 34.1 |
| 1920x1080 | Kimono | 0.764 | 55.2 |
| 1920x1080 | ParkScene | 0.601 | 43.9 |
| 1920x1080 | Cactus | 1.219 | 38.1 |
| 1920x1080 | BQTerrace | 0.979 | 37.9 |
| 1280x720 | FourPeople | 0.730 | 64.5 |
| 1280x720 | Johnny | 0.634 | 69.1 |
| 1280x720 | SlideShow | 1.844 | 67.8 |
| 1024x768 | ChinaSpeed | 0.763 | 37.7 |
| Average | | 0.292 | 47.09 |

as well as a metric correlating BD-rate with TS (BD/TS). TS values show that the strategy is capable of reducing transrating time significantly, with an average reduction of 47.1% in comparison to the original tandem transcoder. Average results per spatial resolution show that HD sequences achieved the largest TS results, reducing transrating time by 59.77%. The sequence that achieved the greatest reduction in transrating time is Johnny, which reached an average TS of 69.1%. This sequence comprises a large static background area composed mainly of 64x64 CUs in the HBR bitstream. This allows the transrating process to be significantly simplified in CU spitting decisions, which leads to the large TS results observed. The worst results in TS occur for the CatRobot video, which still managed to achieve an average TS of 31.5%.

### 4.2. Compression Efficiency

Bjontegaard Delta (BD) metrics [14] were used to evaluate the encoding efficiency of the proposed method for all video sequences. BD-rate is usually calculated based on the bitrate and the Peak Signal-to-Noise Ratio (PSNR) obtained when encoding video sequences under four different QPs. In this work, however, QP cannot be fixed during the full encoding process, since it is adjusted by the rate control algorithm to achieve the target bitrates. Thus, the bitrate and PSNR values obtained when transrating to the four LBR are used in this work to calculate BD-rate.

The obtained BD-rate results are presented in Table 4 and they show that the employment of the proposed method resulted in an average compression efficiency loss of 0.292% in comparison to the original tandem transcoder. This is a very small drawback considering the achieved time savings, as represented by the ratio between BD-rate and TS values (BD/TS). For clarity purposes, the ratio is scaled by a constant factor of 100 in Table 4. For each percentage point in TS, a BD-rate increase of only 0.00266% is noticed. The satisfactory compression efficiency results are explained by the accuracy of the random forest models previously presented.

It is important to note that three video sequences presented negative BD-rate values, which means that the compression efficiency of the proposed solution was better than the original transcoder in some cases. This happens because the encoding decisions performed by the original transcoder - based on the Rate-Distortion Optimization (RDO) algorithm - are locally optimal, i.e., they consider only the effect on the CU being encoded. However, the encoding of the next CUs is also affected by such decisions, since they change reference pixels used in predictions. This way, even though the proposed strategy sometimes incurs in non-optimal local decisions for a CU, such decisions end up improving the global encoding efficiency in subsequent encodings steps.

**Table 5**. Comparison with related works

| Reference | BD-rate (%) | Time Savings (%) | BD/TS |
|---|---|---|---|
| Praeter [7] | 5.60 | 61.0 | 9.180 |
| Yang [8] | 2.26 | 55.0 | 4.109 |
| Shroeder [6] | 0.76 | 38.4 | 1.979 |
| Bubolz [9] | 0.88 | 45.4 | 1.938 |
| **Proposed** | **0.29** | **47.1** | **0.266** |

### 4.3. Comparison with Related Works

Four state-of-the-art solutions [6–9] were selected for comparison with the proposed method. Table 5 shows a comparison between them in terms of BD-rate, TS and BD/TS (x100). The solution proposed by [7] reaches the best results in terms of time savings (61%), but incurs in large compression efficiency losses, reaching an average BD-rate increase of 5.6%. This leads to the worst scenario in terms of BD/TS (9.18). On the other hand, the best result in terms of compression efficiency is achieved by [6], with a BD-rate increase of only 0.76%. However, the solution leads to the smallest TS results, reducing transcoding time in only 38.4%.

Among all compared works, [9] presents the best trade-off between compression efficiency and time savings. The strategy reaches an average BD/TS of 1.938, which means that a BD-rate increase of only 0.01938% is noticed for each percentage point in TS. However, as shown in Table 5, the solution proposed in this paper also surpasses such results.

## 5. CONCLUSIONS

This paper presented a method for complexity reduction of the HEVC transrating process based on random forests. The proposed models use information gathered from the HEVC decoding process to accelerate the recursive CU decision process during the re-encoding and is capable of reducing processing time with small pr negligible effects in encoding efficiency. An average transrating time reduction of 47.9% was achieved in comparison to the original tandem transcoder, at the cost of a small BD-rate increase of only 0.2923%. The proposed strategy is especially useful for large-scale video streaming services that employ online transrating, thus requiring multiple transcodings for bitrate adaptation upon user request.

## 6. REFERENCES


[1] High Efficiency Video Coding, "Recommendation itu-t h. 265," *International Standard ISO/IEC*, pp. 23008–2, 2013.

[2] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, Thomas Wiegand, et al., "Overview of the high efficiency video coding(hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] Thomas Wiegand, "Draft itu-t recommendation and final draft international standard of joint video specification (itu-t rec. h. 264— iso/iec 14496-10 avc)," *JVT-G050*, 2003.

[4] Guilherme Correa, Pedro Assuncao, Luciano Agostini, and Luis A da Silva Cruz, "Performance and computational complexity assessment of high-efficiency video encoders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1899–1909, 2012.

[5] Guilherme Correa, 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*, vol. 25, no. 4, pp. 660–673, 2015.

[6] Damien Schroeder, Adithyan Ilangovan, Martin Reisslein, and Eckehard Steinbach, "Efficient multi-rate video encoding for hevc-based adaptive http streaming," *IEEE Transactions on Circuits and systems for Video Technology*, vol. 28, no. 1, pp. 143–157, 2018.

[7] Johan De Praeter, Antonio Jesús Díaz-Honrubia, Niels Van Kets, Glenn Van Wallendael, Jan De Cock, Peter Lambert, and Rik Van de Walle, "Fast simultaneous video encoder for adaptive streaming," in *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*. IEEE, 2015, pp. 1–6.

[8] Shih-Hsuan Yang and Chong-Cheng Zhong, "Fast coding-unit mode decision for hevc transrating," in *Computer and Information Technology (CIT), 2017 IEEE International Conference on*. IEEE, 2017, pp. 93–100.

[9] Thiago Bubolz, Ruhan Conceição, Mateus Grellert, Bruno Zatt, Luciano Agostini, and Guilherme Correa, "Fast and energy-efficient hevc transrating based on frame partitioning inheritance," in *2018 IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS)*. IEEE, 2018, pp. 1–4.

[10] Tin Kam Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, Aug 1998.

[11] K Sharman and C Rosewarne, "Common test conditions and software reference configurations for hevc," in *Proceedings of the Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ISO/IEC Z1100*, 2017.

[12] Ken McCann, C Rosewarne, B Bross, M Naccari, K Sharman, and GJ Sullivan, "High efficiency video coding (hevc) test model 16 (hm 16) improved encoder description," *Jonit Collaborative Team on Video Coding, JCTVC-S1002, Strasbourg, FR*, 2014.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[14] Gisle Bjontegaard, "Calculation of average psnr differences between rd-curves," *VCEG-M33*, 2001.