



EMES: Efficient Multi-encoding Schemes for HEVC-based Adaptive Bitrate Streaming

VIGNESH V MENON and HADI AMIRPOUR, Christian Doppler Laboratory ATHENA,

Alpen-Adria-Universität Klagenfurt, Austria

MOHAMMAD GHANBARI, School of Computer Science and Electronic Engineering, University of Essex, UK

CHRISTIAN TIMMERER, Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität Klagenfurt, Austria

In *HTTP Adaptive Streaming* (HAS), videos are encoded at multiple bitrates and spatial resolutions (i.e., *representations*) to adapt to the heterogeneity of network conditions, device attributes, and end-user preferences. Encoding the same video segment at multiple representations increases costs for content providers. State-of-the-art multi-encoding schemes improve the encoding process by utilizing encoder analysis information from already encoded representation(s) to reduce the encoding time of the remaining representations. These schemes typically use the highest bitrate representation as the reference to accelerate the encoding of the remaining representations. Nowadays, most streaming services utilize cloud-based encoding techniques, enabling a fully parallel encoding process to reduce the overall encoding time. The highest bitrate representation has a higher encoding time than the other representations. Thus, utilizing it as the reference encoding is unfavorable in a parallel encoding setup as the overall encoding time is bound by its encoding time. This article provides a comprehensive study of various multi-rate and multi-encoding schemes in both serial and parallel encoding scenarios. Furthermore, it introduces novel heuristics to limit the *Rate Distortion Optimization* (RDO) process across various representations. Based on these heuristics, three multi-encoding schemes are proposed, which rely on encoder analysis sharing across different representations: (i) optimized for the *highest compression efficiency*, (ii) optimized for the *best compression efficiency-encoding time savings tradeoff*, and (iii) optimized for the *best encoding time savings*. Experimental results demonstrate that the proposed multi-encoding schemes (i), (ii), and (iii) reduce the overall serial encoding time by 34.71%, 45.27%, and 68.76% with a 2.3%, 3.1%, and 4.5% bitrate increase to maintain the same VMAF, respectively, compared to stand-alone encodings. The overall parallel encoding time is reduced by 22.03%, 20.72%, and 76.82% compared to stand-alone encodings for schemes (i), (ii), and (iii), respectively.

CCS Concepts: • **Information systems** → **Multimedia streaming**;

Additional Key Words and Phrases: HTTP Adaptive Streaming, HEVC, multi-rate encoding, multi-encoding

The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. Christian Doppler Laboratory ATHENA: <https://athena.itec.aau.at/>.

Authors' addresses: V. V Menon, H. Amirpour, and C. Timmerer, Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität Klagenfurt, Austria; emails: {vignesh.menon, hadi.amirpour, christian.timmerer}@aau.at; M. Ghanbari, School of Computer Science and Electronic Engineering, University of Essex, UK; email: ghan@essex.ac.uk.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2023 Copyright held by the owner/author(s).

1551-6857/2023/03-ART129 \$15.00

<https://doi.org/10.1145/3575659>

ACM Reference format:

Vignesh V Menon, Hadi Amirpour, Mohammad Ghanbari, and Christian Timmerer. 2023. EMES: Efficient Multi-encoding Schemes for HEVC-based Adaptive Bitrate Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 3s, Article 129 (March 2023), 20 pages.
<https://doi.org/10.1145/3575659>

1 INTRODUCTION

Motivation: The **Moving Picture Experts Group (MPEG)** has developed a standard called **Dynamic Adaptive Streaming over HTTP (MPEG-DASH)** [24] to meet the high demand for streaming high-quality video content over the Internet and overcome the associated challenges in **HTTP Adaptive Streaming (HAS)**. The main idea behind HAS is to divide the video content into segments and to encode each segment at various bitrates and resolutions, called *representations*, as shown in Figure 1. These representations enable a continuous adaptation of the video delivery to the client's network conditions and device capabilities. The increase in video traffic and improvements in video characteristics such as resolution, framerate [15], and bit-depth raise the need to develop a large-scale, highly efficient video encoding environment [6]. This is even more crucial for DASH-based content provisioning as it requires encoding multiple representations of the same video content.

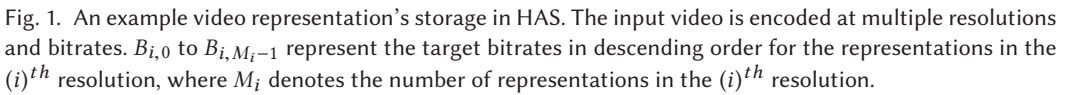
High Efficiency Video Coding (HEVC) [26] is one standard video codec that significantly improves coding efficiency over its predecessor **Advanced Video Coding (AVC)** [27]. This improvement is achieved at the cost of significantly increased encoding time, which is a challenge for content and service providers. As various representations of the same video content are encoded at different bitrates or resolutions, the encoding analysis information from the already encoded representations can be shared to accelerate the encoding of other representations. Several state-of-the-art schemes [7, 9, 12, 17, 23, 28] first encode a single representation, called a *reference representation*. The encoder creates analysis metadata (file) with information such as the slice-type decisions, CU [18], PU, TU partitioning, and the HEVC bitstream itself during this encoding. The remaining representations, called *dependent representations*, analyze the above metadata and then reuse it to skip searching some partitioning, thus reducing the encoding time. With the emergence of cloud-based encoding services,¹ video encoding is accelerated by utilizing an increased number of resources; i.e., with multi-core CPUs, multiple representations can be encoded in parallel.

When encoding multiple representations serially, i.e., one after the other, the overall serial encoding time τ_S is the sum of the encoding times of all representations as shown in Equation (1):

$$\tau_S = \sum_{i=0}^{N-1} \sum_{j=0}^{M_i-1} (\tau_{B_{i,j}}), \quad (1)$$

where N denotes the number of resolutions, M_i denotes the number of bitrate representations in the $(i)^{th}$ resolution, and $\tau_{B_{i,j}}$ represents the time taken for encoding the representations, $B_{i,j} \forall i \in [1, N]$ and $j \in [1, M_i]$. When encoding multiple representations in parallel, the overall parallel encoding time τ_P is bounded by the encoding time of the representation with the highest encoding time. In case of stand-alone encoding, the highest encoding time is for the highest bitrate representation. Although the schemes utilizing the highest bitrate representation as Reference [1] reduce the encoding time for dependent representations, the overall parallel encoding time will

¹<https://bitmovin.com/introducing-cloud-connect-encoding-aws-gcp-azure/> (accessed June 30, 2022).


$$\tau_P = \max_{i,j}(\tau_{B_{i,j}}); j \in [0, M_i - 1], i \in [0, N - 1]. \quad (2)$$

Contributions: In this article, a *double-bound* approach, which is based on our previous work [2], is used for fast CU depth estimation in multi-rate encoding. In this scheme, both the highest and the lowest bitrate representations are used as references to speed up the encoding of the intermediate bitrate representations. This article also introduces prediction mode and motion estimation heuristics to accelerate multi-rate encoding further. Novel multi-resolution encoder analysis sharing methods are presented to accelerate encoding in more than one resolution. Three fast multi-encoding schemes are introduced: (i) optimized for the *highest compression efficiency*, (ii) optimized for the *best compression efficiency-encoding time savings tradeoff*, and (iii) optimized for the *highest encoding time savings*. All the state-of-the-art and proposed schemes in this article are implemented and evaluated using the x265 HEVC open-source encoder² for serial and parallel encoding environments, and the conclusions are presented.

²<http://x265.org/> (accessed June 30, 2022).

Table 1. Overview of the Schemes Presented in This Article

Scheme	Name	Abbreviation	Reference	Section
Multi-rate encoding	SMRES-1	State-of-the-art Multi-Rate Encoding Scheme-1	[23]	2
	SMRES-2	State-of-the-art Multi-Rate Encoding Scheme-2 ²		2
	SMRES-3	State-of-the-art Multi-Rate Encoding Scheme-3	[8, 13]	2
	EMRES-1	Efficient Multi-Rate Encoding Scheme-1	[2]	4.1
	EMRES-2	Efficient Multi-Rate Encoding Scheme-2	[17]	4.2
	EMRES-3	Efficient Multi-Rate Encoding Scheme-3	[17]	4.2
	EMRES-4	Efficient Multi-Rate Encoding Scheme-4	-	4.2
Multi-encoding	SMES	State-of-the-art Multi-Encoding Scheme	[8, 13]	2
	EMES-1	Efficient Multi-Encoding Scheme-1	[17]	5.1
	EMES-2	Efficient Multi-Encoding Scheme-2	[17]	5.2
	EMES-3	Efficient Multi-Encoding Scheme-3	-	5.3

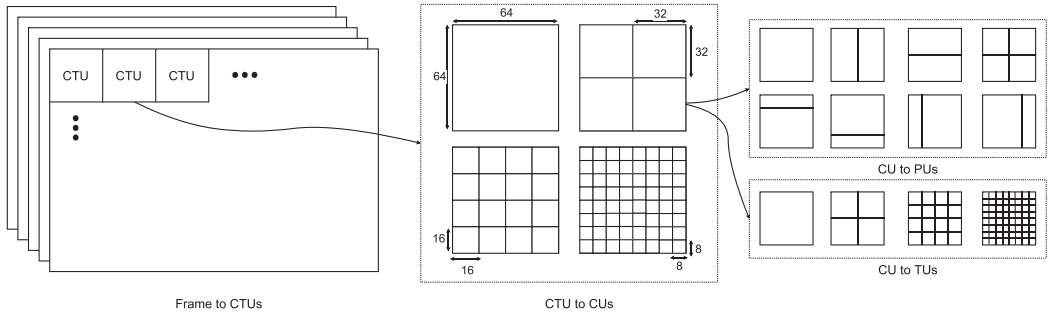


Fig. 2. In HEVC encoding, frames are divided into CTUs, and each CTU is then divided into CUs. Each CU is subdivided into PUs, and the prediction residuals of CUs are partitioned into TUs. The optimal CTU partitioning is found after an exhaustive search process through all CUs, PUs, and TUs.

2 BACKGROUND AND RELATED WORK

In HAS, *the same video content* is encoded at multiple representations, referred as to a bitrate ladder, to continuously adapt the video delivery to the user's needs. Bitrate ladders are usually optimized per content over different dimensions, including bitrate [21], resolution [11, 16], framerate [15], and codec [20]. On the other hand, encoding *the same video content* at multiple representations results in a significant increase in the encoding cost. However, as the same content is encoded multiple times, the encoder analysis information from the already encoded representation(s) can be reused to speed up the encoding process of the remaining representations. In HEVC, frames are first divided into slices, and then they are further divided into square regions called **Coding Tree Units (CTUs)** [26], which are the main building blocks of HEVC. To encode each CTU, it is recursively divided into smaller square regions called **Coding Units (CUs)** (see Figure 2). Depth values from 0 to 3 are assigned to CU sizes from 64×64 to 8×8 pixels. Therefore, to find the best CTU partitioning, 85 CUs including one 64×64 CU, four 32×32 CUs, sixteen 16×16 CUs, and sixty-four 8×8 CUs are searched.

Additionally, for both intra-picture and inter-picture prediction, each CU can be further subdivided into smaller blocks called **Prediction Units (PUs)** along the coding tree boundaries. The inter-prediction modes comprise Merge/Skip $2N \times 2N$, Inter $2N \times 2N$, **Symmetric Motion Partition (SMP)**, including Inter $2N \times N$ and Inter $N \times 2N$, **Asymmetric Motion Partition (AMP)**, including Inter $2N \times nU$, Inter $2N \times nD$, Inter $nL \times 2N$, and Inter $nR \times 2N$, and Inter $N \times N$. In contrast, the intra-prediction modes involve Intra $2N \times 2N$ and Intra $N \times N$. The best PU mode

Table 2. Statistics of CUs in the Bitrate Representations (cf. Figure 1) That Have Depth Values Lower Than the Highest Bitrate Representation ($j = 0$) for JVET Sequences Used in This Article (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p ($i = 0$)			1080p ($i = 1$)			2160p ($i = 2$)		
Bitrate	0.5 Mbps ($j = 3$)	1.0 Mbps ($j = 2$)	1.5 Mbps ($j = 1$)	3.0 Mbps ($j = 3$)	4.5 Mbps ($j = 2$)	5.8 Mbps ($j = 1$)	11.6 Mbps ($j = 3$)	16.8 Mbps ($j = 2$)	20.0 Mbps ($j = 1$)
Average	94.62%	96.31%	95.76%	91.27%	93.29%	94.13%	90.23%	89.68%	90.73%

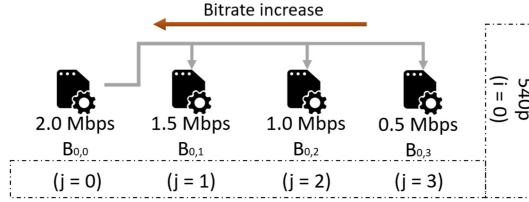


Fig. 3. Encoder analysis flow diagram of SMRES-1 [23]. The gray arrow represents the single bound for CU estimation.

is selected according to all modes' minimum **rate-distortion cost (RD-cost)**. Furthermore, for transform coding of the prediction residuals, each CU can be partitioned into multiple **transform blocks (TBs)**, the size of which can also vary from 4×4 to 32×32 pixels [26]. In general, finding the optimal partitioning for each CTU is time-consuming, given the possible CU, PU, and TU partitionings; multiple reference frame search processes for inter-coded CUs; and the actual motion estimation algorithm.

Schroeder et al. [23] propose a *single bound* approach for CU depth estimation denoted as **State-of-the-art Multi-Rate Encoding Scheme (SMRES)-1** see Figure 3 in this article, in which, first, the highest bitrate representation is encoded. Its CU depth information is then used to encode the other representations. As shown in Table 2, CUs generally have higher depth values in higher bitrate representations. The CU depth search range for the representations, $B_{i,j} \forall i \in [1, M_i - 1]$, is calculated as

$$[d_L, d_U] = \begin{cases} [0, 3], & B_{i,0} \\ [0, d_{i,0}], & B_{i,j} \forall i \in [1, M_i - 1], \end{cases} \quad (3)$$

where M_i represents the number of bitrate representations in the $(i)^{\text{th}}$ resolution, $B_{i,0}$ is the highest bitrate representation of the $(i)^{\text{th}}$ resolution, and $d_{i,0}$ denotes the CU depth of the $B_{i,0}$ representation. In this scheme, larger depths are skipped in the RDO process of the dependent representations, and considerable time is saved. For example, if the optimal CU depth is calculated to be 1 (i.e., 32×32 pixels) for a block in the highest bitrate representation, CUs with depth 2 (i.e., 16×16 pixels) and depth 3 (i.e., 8×8 pixels) are skipped from the RDO process when encoding the co-located CUs in dependent representations. An extension to this scheme was proposed in [22], where the lower-resolution encodes reuse CU depth information from the highest resolution encode. It means that the overall encoding time is bound by the encoding time of the highest-resolution encode. Thus, this approach is not efficient in the parallel encoding environment.

In an analysis reuse scheme of x265³ (SMRES-2), the dependent representations force the optimal decisions made in the highest bitrate representation, such as (i) the slice-type and (ii) scene-cut decisions [14] along with (iii) the quadtree structure, (iv) prediction modes, and (v) reference indices

³analysis-load-reuse-level = 6 in <https://x265.readthedocs.io/en/master/cli.html>.

and skip RDO process for all other possible decisions. In another analysis reuse scheme of x265⁴ (SMRES-3), the dependent representations reuse all CUs and PUs from the reference representation, i.e., the highest bitrate representation without additional RDO. Additionally, the dependent representations also employ static refinement techniques defined in x265 [8]. If the reference depth is d , it is re-evaluated by using the RDO against the cost of splitting the CU (i.e., depth $d + 1$) by computing the optimal PU modes. Additionally, the motion vectors from the reference representation are used as predictors for motion search for the co-located PUs in the current representation.

Amirpour et al. [1] consider all representations from the highest to the lowest bitrate representation as a potential, single reference to speed up the encoding of dependent representations. Using the median bitrate representation as the reference shows the best performance compared to using other representations as the reference encoding. De Praeter et al. [7] first encode the highest-quality representation in the highest resolution as the reference encoding. They predict encoding decisions for dependent representations by a random forest model. The variance of the transform coefficients, motion vector variance, and information about the block structure such as mean, variance, maximum, and minimum values of the co-located CU, PU, and TU block sizes are used as inputs to the random forest model. Grellert et al. [9] use the information from the highest-quality representation to speed up the encoding of other representations based on heuristic and learning-based methods. Random Forests were used as CTU depth upper-bound estimators to skip larger CUs in lower-quality representations. Yang et al. [28] identify CTUs that cover objects of interest in the highest-quality representation and allocate higher bitrates for lower-quality representations. Lindino et al. [12] inherit PU mode information from the highest-quality representation to accelerate the PU mode decision during the encoding of lower-quality representations. Gu et al. [10] utilize the high correlation between low- and high-quality representations along with information on low-quality representations to optimize and accelerate the rate control model of high-quality representations.

In this article, the multi-encoding scheme proposed in [8, 13] is considered as the State-of-the-art Multi-Encoding Scheme (SMES), which is briefly described below:

- (i) The highest bitrate representation of the lowest resolution is encoded.
- (ii) The remaining representations of the lowest resolution are encoded using SMRES-3 with the highest bitrate representation of the lowest resolution as the reference representation.
- (iii) As the multi-resolution approach (referred to as Mode-S in this article⁵), the encoder analysis data from the highest bitrate representation of the lowest resolution is shared with the highest bitrate representation of the next higher resolution scaled by the resolution increase factor. This representation then employs SMRES-3.
- (iv) The remaining bitrate representations of the next higher resolution are encoded using SMRES-3 as used in Step (ii).
- (v) Repeat Step (iii) for the remaining resolution layers in ascending order.

The encoding analysis flow diagram of this scheme is shown in Figure 4.

3 PROPOSED HEURISTICS

This section discusses specific proposed heuristics that can be applied coupled with CU depth search optimization-based multi-rate encoding approaches to improve the overall encoding speed without compromising the coding efficiency.

⁴analysis-load-reuse-level = 10, refine-intra = 4, refine-inter = 2, refine-mv = 1 in <https://x265.readthedocs.io/en/master/cli.html>.

⁵scale-factor = 2, analysis-load-reuse-level = 10, refine-intra = 4, refine-inter = 2, refine-mv = 1 in <https://x265.readthedocs.io/en/master/cli.html>.

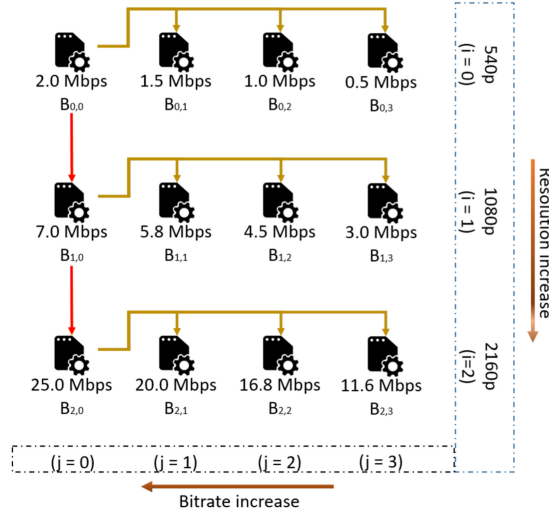


Fig. 4. Encoder analysis flow diagram of SMES. The golden arrow denotes encoder analysis sharing using SMRES-3. The red line denotes the multi-resolution Mode-S.

Table 3. Statistics of PUs in the Bitrate Representations (cf. Figure 1) That Chose Skip $2N \times 2N$ Mode When the Co-located PU in the Highest Bitrate Representation Had Chosen Skip $2N \times 2N$ Mode for JVET Sequences Used in This Article (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)			1080p (i = 1)			2160p (i = 2)		
Bitrate	0.5 Mbps (j = 3)	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	3.0 Mbps (j = 3)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	11.6 Mbps (j = 3)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	89.79%	89.03%	87.95%	87.98%	86.57%	87.01%	86.80%	85.72%	87.22%

Table 4. Statistics of PUs in the Bitrate Representations (cf. Figure 1) That Chose AMP Mode When the Co-located PU in the Highest Bitrate Representation Had Chosen $2N \times 2N$ Mode for JVET Sequences Used in This Article (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)			1080p (i = 1)			2160p (i = 2)		
Bitrate	0.5 Mbps (j = 3)	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	3.0 Mbps (j = 3)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	11.6 Mbps (j = 3)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	1.34%	1.47%	1.42%	1.59%	1.54%	1.61%	2.08%	1.97%	2.13%

Prediction Mode Heuristics: If the CU is split to the CU size chosen in the highest bitrate representation, the prediction mode heuristics is proposed for the remaining representations as follows:

- As shown in Table 3, if Skip $2N \times 2N$ mode was chosen as the best mode in the RDO process of the highest bitrate representation, Skip $2N \times 2N$ mode is likely to be selected in the other representations. Hence, if Skip $2N \times 2N$ mode was selected in the highest bitrate representation, RDO is evaluated for only Merge/Skip $2N \times 2N$ and Inter $2N \times 2N$ modes.
- As observed in Table 4, if $2N \times 2N$ mode was selected as the best mode in the RDO process of the highest bitrate representation, it is less likely that an **asymmetrical motion prediction (AMP)** mode would be selected in the other representations. Hence, if the $2N \times 2N$ mode was chosen in the highest bitrate representation, RDO is skipped for AMP modes.
- It is observed in Table 5 that if any inter-prediction mode was selected as the best mode in the RDO process of the highest bitrate representation, it is less likely that any intra-prediction

Table 5. Statistics of PUs in the Bitrate Representations (cf. Figure 1) That Chose Intra-prediction Mode When the Co-located PU in the Highest Bitrate Representation Had Chosen Intra-Prediction Mode for JVET Sequences Used in This Paper (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)			1080p (i = 1)			2160p (i = 2)		
Bitrate	0.5 Mbps (j = 3)	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	3.0 Mbps (j = 3)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	11.6 Mbps (j = 3)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	3.78%	3.39%	4.06%	3.85%	3.72%	3.97%	4.07%	4.31%	4.23%

Table 6. Statistics of PUs in the Intermediate Bitrate Representations (cf. Figure 1) That Chose Intra-prediction Mode When the Co-located PU in the Highest and Lowest Bitrate Representation Had Chosen Intra-prediction Mode for JVET Sequences Used in This Paper (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)		1080p (i = 1)		2160p (i = 2)	
Bitrate	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	89.76%	87.31%	86.38%	85.32%	87.35%	89.97%

Table 7. Statistics of PUs in the Bitrate Representations (cf. Figure 1) That Chose the Same Reference Frame as That of the Co-located PU in the Highest Bitrate Representation for JVET Sequences Used in This Article (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)			1080p (i = 1)			2160p (i = 2)		
Bitrate	0.5 Mbps (j = 3)	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	3.0 Mbps (j = 3)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	11.6 Mbps (j = 3)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	82.79%	81.36%	79.80%	81.98%	79.96%	79.83%	82.03%	80.17%	80.15%

- mode would be selected in the other representations. Hence, if any inter-prediction mode was chosen in the highest bitrate representation, RDO is skipped for intra-prediction modes.
- (iv) As noticed in Table 6, the probability of PUs in intermediate bitrate representations choosing an intra-prediction mode is very high when the co-located PUs in the highest and lowest bitrate representations have selected an intra-prediction mode. Hence, if an intra-prediction mode was chosen for the highest and the lowest bitrate representations, RDO is evaluated for only Merge/Skip $2N \times 2N$ and intra-prediction modes in the intermediate representations.

Motion Estimation Heuristics: If the CU is split to the CU size chosen in the highest bitrate representation and the PU size is also the same as that of the highest bitrate representation, the motion estimation heuristics is proposed for the remaining representations as follows:

- (i) It is observed in Table 7 that the probability of the dependent representations choosing the same reference frame as that in the highest bitrate representation is very high. Hence, the same reference frame is selected as the highest bitrate representation, and other reference frame searches are skipped in the dependent representations.
- (ii) The **Motion Vector Predictor (MVP)** is set to be the **Motion Vector (MV)** of the highest bitrate representation.
- (iii) The motion search range is decreased to a smaller window size if the MV of the highest bitrate representation and the MV of the lowest bitrate representation are close. The search range is determined to be the maximum difference between the x and y coordinates of the MVs.

Table 8. Statistics of CUs in the Intermediate Bitrate Representations (cf. Figure 1) That Have Depth Values between Highest and Lowest Bitrate Representations for JVET Sequences Used in This Article (cf. Table 11) Encoded Using the *veryslow* Preset of x265 HEVC Encoder

Resolution	540p (i = 0)		1080p (i = 1)		2160p (i = 2)	
Bitrate	1.0 Mbps (j = 2)	1.5 Mbps (j = 1)	4.5 Mbps (j = 2)	5.8 Mbps (j = 1)	16.8 Mbps (j = 2)	20.0 Mbps (j = 1)
Average	88.76%	87.92%	90.21%	89.83%	90.04%	91.39%

4 EFFICIENT MULTI-RATE ENCODING SCHEMES

This section introduces the proposed **Efficient Multi-Rate Encoding Scheme (EMRES)-1**, EMRES-2, EMRES-3, and EMRES-4.

4.1 Efficient Multi-Rate Encoding Scheme (EMRES)-1

It is observed in Table 8 that the CU depth in the intermediate bitrate representations is highly likely to be between the CU depth of the highest and lowest bitrate representations. Thus, in this scheme, which is based on previous work [2], the information from the highest and the lowest bitrate representations is used to reduce the encoding time of the intermediate representations. First, the highest bitrate representation is encoded as a stand-alone encoding (i.e., independent of RDO of other representations). The CU depth information obtained from this encoding is used as the upper bound of the CU depth estimation process of the lowest bitrate representation similar to [23]. The remaining intermediate-quality representations are encoded with a *double-bound* approach for CU depth search and reference frame selection as shown in Figure 5(a). Hence, in the CU depth search of the intermediate bitrate representations, the possible depth-level search window is limited by the depth values of the highest and lowest bitrate representations, as shown in Equation (4):

$$[d_L, d_U] = \begin{cases} [0, 3], & B_{i,0} \\ [0, d_{i,0}], & B_{i,M_i-1} \\ [d_{i,M_i-1}, d_{i,0}], & B_{i,j} \forall j \in [1, M_i - 2], \end{cases} \quad (4)$$

where d_{i,M_i-1} and $d_{i,0}$ denote the CU depths chosen in the lowest and highest bitrate representations of the $(i)^{th}$ resolution, respectively. In addition, other reference frame searches are skipped if both the highest and lowest bitrate representations have the same reference frame [2].

4.2 Heuristics-based Efficient Multi-rate Encoding Schemes

This section introduces the proposed multi-rate encoding schemes based on the proposed prediction mode and motion estimation heuristics.

Efficient Multi-rate Encoding Scheme (EMRES)-2. This combines the SMRES-1 scheme with the proposed prediction mode and motion estimation heuristics. The encoder analysis information from the highest bitrate representation is used to encode the other representations [17]. Figure 5(b) shows the encoder analysis flowchart for this scheme.

Efficient Multi-rate Encoding Scheme (EMRES)-3. This combines the EMRES-1 scheme with the proposed prediction mode and motion estimation heuristics. This scheme is applied when there are more than two representations per resolution [17]. Figure 5(c) shows the encoder analysis flowchart for this scheme.

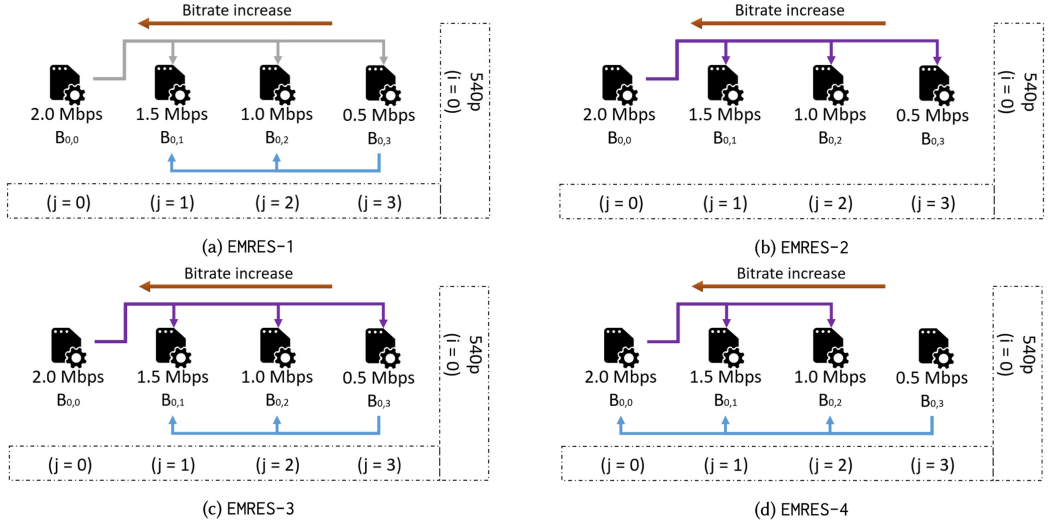


Fig. 5. Encoder analysis flow diagram of (a) EMRES-1, (b) EMRES-2, (c) EMRES-3, and (d) EMRES-4. The gray line denotes the upper bound for CU depth estimation, the blue line denotes the lower bound of CU depth estimation, and the purple line represents the upper bound for CU depth estimation coupled with the proposed heuristics.

Efficient Multi-rate Encoding Scheme (EMRES)-4. This is tailor-made for parallel encoding environments. In this scheme, the lowest bitrate representation is encoded first. The CU depth search in the highest bitrate representation is lower bound by the CU depth values of the lowest bitrate representation. The remaining intermediate-quality representations are encoded with a double-bound approach for CU depth search and the proposed heuristics. This scheme is proposed specifically to encode multiple bitrate representations of a single resolution in parallel. Figure 5(d) shows the encoder analysis flowchart for this scheme.

5 EFFICIENT MULTI-ENCODING SCHEMES

In this section, multi-encoding schemes based on the proposed heuristics and multi-resolution approaches are introduced. EMES-1 is optimized for the highest compression efficiency, EMES-2 is optimized for the best compression efficiency-encoding time savings tradeoff, and EMES-3 is optimized for the highest encoding time savings. The optimizations are skipped for intra-coded CTUs in the following schemes since the contribution to the encoder speedup from those optimizations is insignificant.

5.1 Efficient Multi-encoding Scheme (EMES)-1

In the first proposed multi-encoding scheme, EMRES-3 is used for encoder analysis sharing across representations within a resolution. As the multi-resolution approach (referred to as Mode-1 in this article), the CU depth information from the highest bitrate representation of the $(i-1)^{th}$ resolution, i.e., $d_{i-1,0}$, is shared to the highest bitrate representation of the $(i)^{th}$ resolution. The lower bound for CU depth estimation in the highest bitrate representation of the $(i)^{th}$ resolution representation, i.e., d_L , is determined as

$$d_L = \begin{cases} d_{i-1,0} - 1 & \text{if } d_{i-1,0} \geq 1 \\ 0, & \text{otherwise.} \end{cases}$$

The encoding analysis flow diagram of this scheme is shown in Figure 6(a).

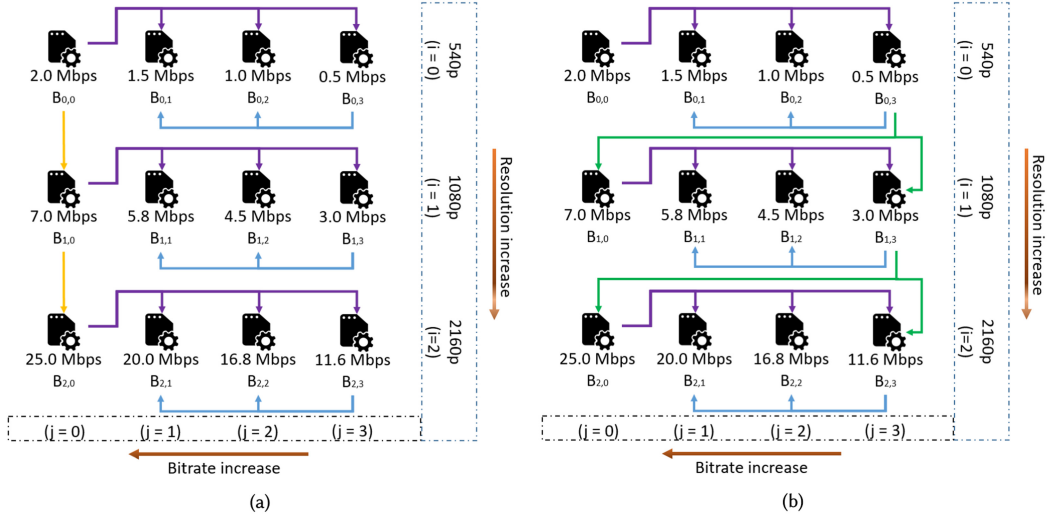


Fig. 6. Encoder analysis flow diagram of (a) EMES-1 and (b) EMES-2. The yellow line denotes the multi-resolution Mode-1 (explained in Section 5.1), while the green line denotes the multi-resolution Mode-2 (explained in Section 5.2).

5.2 Efficient Multi-encoding Scheme (EMES)-2

The second proposed multi-encoding scheme is a small variation of EMES-1, which aims to extend the EMRES-1 scheme across resolution layers. As the multi-resolution approach (referred to as Mode-2 in this article), the CU depth information from the lowest bitrate representation of the $(i-1)^{th}$ resolution, i.e., $d_{i-1, M_{i-1}-1}$, is shared to the highest and lowest bitrate representations of the $(i)^{th}$ resolution. The lower bound for CU depth estimation in the highest and lowest bitrate representations of the $(i)^{th}$ resolution, i.e., d_L , is determined as

$$d_L = \begin{cases} d_{i-1, M_{i-1}-1} - 1, & d_{i-1, M_{i-1}-1} \geq 1 \\ 0, & \text{otherwise.} \end{cases}$$

The encoding analysis flow diagram of this scheme is shown in Figure 6(b).

5.3 Efficient Multi-encoding Scheme (EMES)-3

The third proposed multi-encoding scheme is a small variation of EMES-1, which aims to maximize the encoding time savings in parallel encoding by decreasing the encoding times of the highest bitrate representations of resolution layers. As the multi-resolution approach, d_L is computed for the highest bitrate representation of the $(i)^{th}$ resolution similar to SMES. The PU sizes and MVs are scaled by the resolution increase factor L_i defined as

$$L_i = \frac{W_i \times H_i}{W_{i-1} \times H_{i-1}}, \quad (5)$$

where W_i and H_i represent the width and height of video frame at the $(i)^{th}$ resolution. The CU depth, scaled PU size, and mode decisions are reused in the highest bitrate representation of the $(i)^{th}$ resolution. d_L is re-evaluated by using the RDO against the cost of splitting the CU (i.e., $d_L + 1$) by computing the optimal PU modes. At the same time, the scaled MVs are reused as motion vector predictors for the co-located PUs in the $(i)^{th}$ resolution. The encoding analysis flow diagram of this scheme is shown in Figure 7.

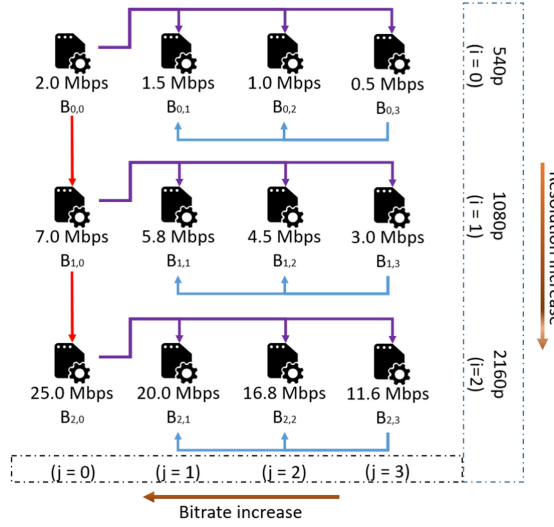


Fig. 7. Encoder analysis flow diagram of EMES-3. The red line denotes the multi-resolution Mode-S (explained in Section 2).

6 EVALUATION

This section first introduces the test methodology used in this article and then presents the experimental results.

6.1 Test Methodology

All schemes presented in this article are implemented using x265 v3.5⁶ with the *veryslow* preset and the **Video Buffering Verifier (VBV)** rate control mode. Psycho-visual optimizations and adaptive quantization are not used in the evaluation. For encoder analysis sharing in the considered schemes, per-segment encoding analysis metadata (file) is generated by the reference representations along with the HEVC bitstream and shared to the dependent representations. All experiments are run on a dual-processor server with Intel Xeon Gold 5218R (80 cores, frequency at 2.10 GHz), which utilizes multi-threading optimizations [19] (i.e., **Wavefront Parallel Processing (WPP)** and frame-threading) of x265. ABR ladder encoding of four test sequences was run in parallel. In this configuration, the full multi-threading capabilities of x265 are utilized.

Video sequences from JVET [4], MCML [5], and SJTU [25] datasets are used, representing various types of contents (cf. Table 11). The discussion of the multi-encoding schemes in this article is restricted to the resolutions that are integer power-of-2 multiples of each other. As mentioned in Table 9, 12 representations are considered in the ABR ladder: three resolutions with four bitrates for each resolution ($M_1 = M_2 = M_3 = 4$). The bitrates are chosen in compliance to HTTP Live Streaming (HLS) specification⁷. The lowest resolution is 960×540 , intermediate resolution is 1920×1080 , and highest resolution is 3840×2160 pixels. The lower-resolution sources were generated from the original video source by applying bi-cubic scaling using FFmpeg.⁸

⁶<http://x265.org/> (accessed June 30, 2022).

⁷https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices (accessed June 30, 2022).

⁸<https://ffmpeg.org/ffmpeg.html> (Accessed June 30, 2022).

Table 9. Bitrate Ladder

Bitrate	540p (i = 0)	1080p (i = 1)	2160p (i = 2)
$B_{i,0}$	2.0 Mbps	7.0 Mbps	25.0 Mbps
$B_{i,1}$	1.5 Mbps	5.8 Mbps	20.0 Mbps
$B_{i,2}$	1.0 Mbps	4.5 Mbps	16.8 Mbps
$B_{i,3}$	0.5 Mbps	3.0 Mbps	11.6 Mbps

Table 10. Results of the Multi-rate Encoding Schemes

Scheme	ΔT_S	ΔT_P	BDR_P	BDR_V
SMRES-1	15.65%	0%	0.30%	1.05%
SMRES-2	18.33%	0%	2.69%	2.72%
SMRES-3	54.88%	0%	9.88%	10.55%
EMRES-1	25.92%	0%	1.86%	2.22%
EMRES-2	26.85%	0%	0.90%	1.73%
EMRES-3	37.97%	0%	2.43%	2.88%
EMRES-4	37.89%	20.74%	4.50%	4.42%

The resulting encoding time, quality in terms of PSNR and VMAF,⁹ and achieved bitrate are compared for each test sequence. Since it is assumed that representations are displayed on the highest resolution, i.e., 2160p, all representations are scaled (bi-cubic) to 2160p to calculate VMAF and PSNR [6]. In the experimental results, ΔT_S and ΔT_P represent the cumulative encoding time savings for all bitrate representations compared to the stand-alone encoding in serial and parallel encoding scenarios, respectively; Bjøntegaard delta rates [3] BDR_P and BDR_V refer to the average increase in bitrate of the representations to that of the stand-alone encoding to maintain the same PSNR and VMAF, respectively. A positive BDR indicates a drop in coding efficiency of the proposed method compared to the stand-alone encoding, while a negative BDR represents a coding gain.

6.2 Experimental Results

In this section, the results of multi-rate encoding schemes and multi-encoding schemes are presented. The analysis for both serial and parallel encoding scenarios are discussed.

6.2.1 Multi-rate Encoding.

Serial Encoding: Table 10 provides the summary, and Figure 8 presents a graphical analysis of the encoding time savings (ΔT_S) and BDR , respectively. SMRES-1 is observed to be a BDR conservative scheme with a negligible increase in BDR while reducing the overall encoding time only by 15.65%. The highest encoding time savings is observed with SMRES-3, but it has a significant BDR_P and BDR_V of 9.88% and 10.55%, respectively. Hence, SMRES-1 and SMRES-3 are extreme cases with the least BDR and the highest BDR . SMRES-1 is not suitable for HAS implementations because of its low ΔT_S . SMRES-3 is also not practical, owing to its very high BDR . The proposed scheme, EMRES-1, yields an encoding time savings of 25.92% with a BDR_P and BDR_V of 1.86% and 2.22%, respectively. EMRES-2 shows an encoding time savings of 26.85% with a BDR_P and BDR_V of 0.90% and 1.73%, respectively. Hence, EMRES-1 and EMRES-2 yield better results than SMRES-2 with higher ΔT_S and lower BDR . EMRES-3 shows an encoding time savings of 37.97% with a BDR_P and BDR_V of 2.43% and 2.88%, respectively.

⁹<https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652> (accessed June 30, 2022).

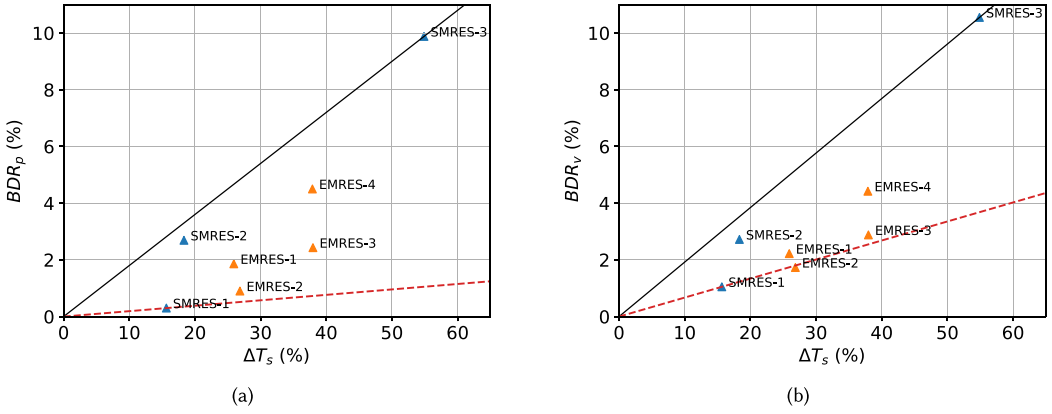


Fig. 8. Comparison of (a) BDR_P and (b) BDR_V and ΔT of multi-rate encoding schemes in serial encoding. The blue and orange points represent the state-of-the-art and proposed multi-rate encoding schemes, respectively. The red dashed line in each figure has a slope of $\frac{BDR}{\Delta T_S}$ of SMRES-1, which shows the least BDR , while the black line in each figure has a slope of $\frac{BDR}{\Delta T_S}$ of SMRES-3, which shows the highest BDR .

Table 11. Results for SMES

Dataset	Video	ΔT_S	ΔT_P	BDR_P	BDR_V
JVET	CatRobot	76.88%	79.36%	19.11%	12.57%
JVET	DaylightRoad2	77.68%	76.22%	18.90%	13.45%
JVET	FoodMarket4	80.76%	78.12%	11.78%	8.61%
MCML	Basketball	74.89%	70.34%	10.27%	12.73%
MCML	Bunny	75.06%	69.44%	4.53%	5.06%
MCML	Construction	76.76%	76.69%	9.58%	13.72%
SJTU	BundNightScape	76.24%	78.64%	1.55%	2.40%
SJTU	RushHour	77.16%	76.28%	6.82%	7.36%
SJTU	TreeShade	72.57%	72.59%	12.65%	9.20%
Average		76.44%	76.56%	9.45%	9.53%

Parallel Encoding: As shown in Table 10, the encoding time savings (ΔT_P) is 0% for SMRES-1, SMRES-2, SMRES-3, EMRES-1, EMRES-2, and EMRES-3. This is because the encoding time is bound by the encoding time of the highest bitrate representation. However, EMRES-4 decreases the encoding time of the highest bitrate representation. Hence, an overall encoding time savings of 20.74% is observed for EMRES-4.

6.2.2 Multi-encoding.

Serial Encoding: As shown in Table 11, using the SMES scheme, the overall encoding time decreases by 76.44% with a BDR_P and BDR_V of 9.45% and 9.53%, respectively. Though the encoding time savings is significant, this scheme is expensive in terms of bitrate increase, which is nearly 10%. It is observed in Table 12 that EMES-1 is a BDR conservative approach with the least increase in BDR , with an overall encoding time savings of 34.71%. EMES-2 shows an overall encoding time savings of 45.27% with a BDR_P and BDR_V of 3.32% and 3.17%, respectively. EMES-3 shows an overall encoding time savings of 68.76% with a BDR_P and BDR_V of 4.51% and 4.48%, respectively. Figure 9 represents the ΔT_S and BDR results for the schemes graphically. The points for EMES-1, EMES-2,

Table 12. Results for the Proposed Multi-encoding Schemes Using *veryslow* Preset of x265

Dataset	Video	EMES-1				EMES-2				EMES-3			
		ΔT_S	ΔT_P	BDR_P	BDR_V	ΔT_S	ΔT_P	BDR_P	BDR_V	ΔT_S	ΔT_P	BDR_P	BDR_V
JVET	CatRobot	35.32%	27.81%	3.06%	2.27%	46.62%	16.76%	3.27%	2.54%	69.22%	79.16%	7.47%	5.26%
JVET	DaylightRoad2	34.54%	30.58%	6.30%	5.10%	43.10%	18.59%	4.47%	3.75%	69.07%	79.16%	6.73%	6.34%
JVET	FoodMarket4	35.96%	16.57%	2.52%	1.99%	46.91%	5.92%	3.29%	2.53%	74.67%	78.55%	4.67%	4.05%
MCML	Basketball	36.85%	18.91%	2.74%	2.94%	48.96%	16.84%	5.14%	4.63%	66.67%	70.10%	4.10%	5.79%
MCML	Bunny	36.97%	10.59%	0.93%	1.90%	46.13%	10.15%	2.47%	2.81%	67.88%	69.13%	2.63%	3.12%
MCML	Construction	25.26%	17.04%	1.70%	3.27%	35.10%	15.41%	2.84%	4.73%	69.64%	76.48%	3.29%	5.87%
SJTU	BundNightScape	35.39%	18.52%	1.99%	0.71%	45.42%	18.90%	3.26%	1.73%	69.82%	78.28%	0.12%	0.76%
SJTU	RushHour	38.24%	29.03%	0.64%	1.67%	53.65%	19.66%	2.83%	3.66%	68.44%	75.63%	2.93%	3.72%
SJTU	TreeShade	33.87%	17.52%	1.45%	1.53%	41.57%	15.66%	2.33%	2.15%	63.41%	70.28%	8.66%	5.44%
Average		34.71%	22.03%	2.37%	2.38%	45.27%	20.72%	3.32%	3.17%	68.76%	76.82%	4.51%	4.48%

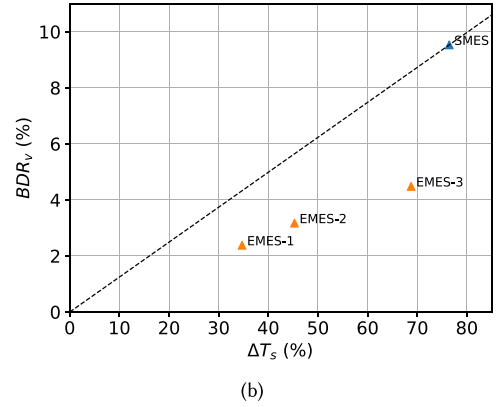
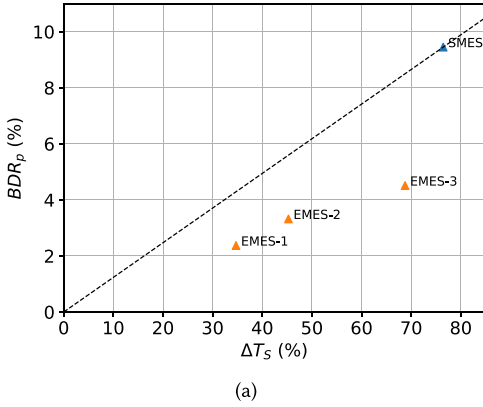


Fig. 9. Comparison of (a) BDR_P and (b) BDR_V and ΔT of multi-encoding schemes in serial encoding. The blue and orange points represent the state-of-the-art and proposed multi-encoding schemes, respectively. The gray dashed line in each figure has a slope of $\frac{BDR}{\Delta T_S}$ of SMES, which shows the highest BDR .

and EMES-3 are under the line with slope $\frac{BDR}{\Delta T_S}$ of SMES. Also, EMES-3 shows ΔT_S close to SMES with much lower BDR .

Parallel Encoding: As shown in Table 11, using the SMES scheme, the encoding time decreases by 76.56% in parallel encoding. As discussed earlier, this scheme is not practical owing to its high BDR . It is observed in Table 12 that EMES-1, EMES-2, and EMES-3 yield encoding time savings of 22.03%, 20.72%, and 76.82%, respectively. Figure 10 represents the ΔT_P and BDR results for the schemes graphically. We observe that ΔT_P is the highest for EMES-3 with a much lower BDR increase compared to SMES. Hence, EMES-3 is the best scheme for a parallel encoding scenario.

Figure 11 summarizes the BD-PSNR and BD-VMAF results of the proposed multi-encoding schemes. The maximum-quality degradation in terms of PSNR is observed for the EMES-2 scheme (0.13 dB), while the maximum-quality degradation in terms of VMAF is observed for the EMES-3 scheme (0.59 VMAF). Figure 12 shows the relative encoding time (in percentage) of all bitrate representations for the considered multi-encoding schemes. The encoding times are normalized by the stand-alone encoding time of the 25 Mbps representation. As discussed earlier, the representation with the maximum encoding time is the bottleneck in the parallel encoding scenario for each scheme. For stand-alone encoding, the maximum encoding time is of the 25 Mbps representation. During parallel encoding, encoding other representations shall be completed before 25 Mbps representation encoding is completed. Thus, the overall encoding time is bound by the encoding time of 25 Mbps representation. For SMES, the maximum encoding time is of the 2 Mbps representation

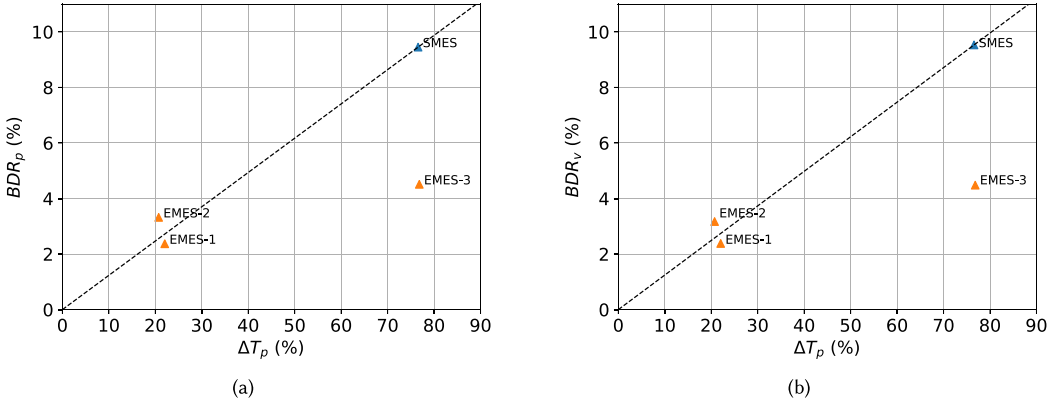


Fig. 10. Comparison of (a) BDR_P and (b) BDR_V and ΔT of multi-encoding schemes in parallel encoding. The black dashed line in each figure has a slope of $\frac{BDR}{\Delta T_P}$ of SMES, which shows the highest BDR.

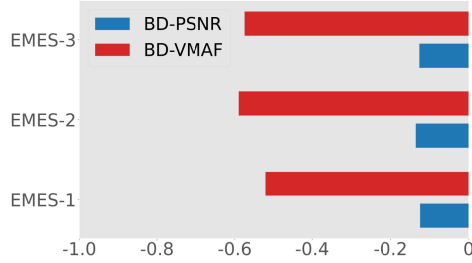


Fig. 11. BD-PSNR and BD-VMAF results of the multi-encoding schemes.

(about 23%), as shown in Figure 12. Hence, the overall encoding time is bound by the encoding time of 2 Mbps representation (i.e., overall speedup in parallel encoding is $(100-23)\% = 77\%$). For EMES-1, the maximum encoding time is of the 25 Mbps representation (about 78%). It implies that the overall encoding time is bound by the encoding time of 25 Mbps representation (i.e., overall speedup in parallel encoding is $(100-78)\% = 22\%$). For EMES-2, the maximum encoding time is of the 25 Mbps representation (about 80%). Hence, the overall encoding time is bound by the encoding time of 25 Mbps representation (i.e., overall speedup in parallel encoding is $(100-80)\% = 20\%$). We also observe that the reduction in the encoding time complexities of each resolution's highest bitrate representations is less than EMES-1. Hence, better encoding time savings is observed using EMES-1 compared to EMES-2 in parallel encoding. Also, the encoding times of the intermediate bitrate representations of each resolution are reduced more using EMES-2 than EMES-1. Hence, we observe better time savings using EMES-2 compared to EMES-1 in serial encoding. Similar to SMES, for EMES-3, the maximum encoding time is of the 2 Mbps representation (about 23%). Thus, the overall encoding time savings is similar to SMES for parallel encoding. We observe that the encoding time savings is the same as that of SMES for the highest bitrate representations of each resolution. In contrast, the encoding time savings is lower than that of SMES for the other bitrate representations. Hence, the overall encoding time savings for serial encoding is less than SMES.

Table 13 provides a comprehensive analysis of the encoding time savings and BDR results using the *ultrafast* preset of x265. The encoding time savings decreases as we use faster presets. Since the amount of RDO computations is already very low in the *ultrafast* preset, the scope of significant

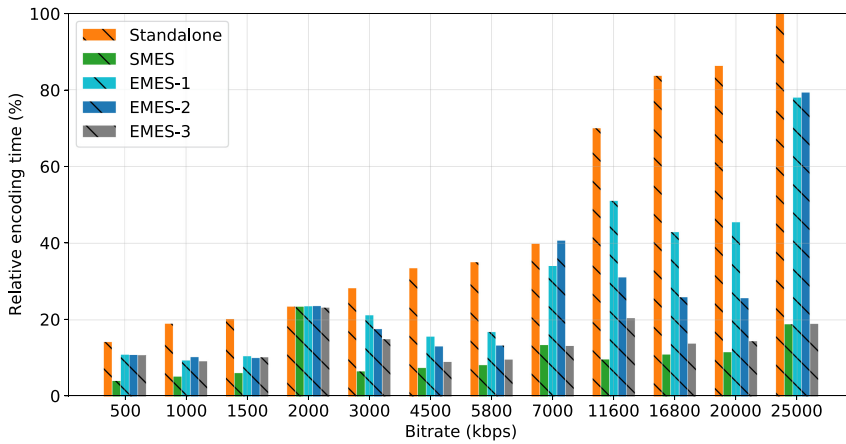


Fig. 12. Relative encoding time (in percentage) of all bitrate representations for the considered multi-encoding schemes. The encoding times are normalized by the stand-alone encoding time of the 25 Mbps representation.

Table 13. Results for the Proposed Multi-encoding Schemes Using *ultrafast* Preset of x265

Dataset	Video	EMES-1				EMES-2				EMES-3			
		ΔT_S	ΔT_P	BDR_P	BDR_V	ΔT_S	ΔT_P	BDR_P	BDR_V	ΔT_S	ΔT_P	BDR_P	BDR_V
JVET	CatRobot	29.57%	22.65%	3.06%	2.27%	41.93%	11.97%	4.29%	4.45%	66.13%	71.14%	4.11%	4.59%
JVET	DaylightRoad2	28.39%	26.39%	2.79%	2.53%	37.97%	13.86%	3.94%	3.49%	65.97%	72.43%	3.67%	3.49%
JVET	FoodMarket4	31.02%	13.39%	3.09%	3.27%	44.34%	7.93%	4.47%	4.30%	72.86%	70.30%	3.00%	3.10%
MCML	Basketball	31.72%	17.73%	2.44%	2.98%	42.86%	11.76%	2.12%	3.11%	62.79%	63.06%	3.19%	4.03%
MCML	Bunny	31.83%	9.55%	8.60%	7.98%	43.93%	9.25%	8.57%	7.92%	64.67%	62.67%	7.35%	7.45%
MCML	Construction	21.98%	13.25%	1.62%	6.57%	31.08%	11.08%	1.44%	6.93%	68.12%	71.98%	1.43%	6.54%
SJTU	BundNightScape	31.64%	13.21%	3.22%	4.64%	41.35%	12.34%	3.36%	4.85%	62.95%	72.54%	3.25%	4.83%
SJTU	RushHour	32.76%	22.90%	4.44%	2.83%	49.06%	13.27%	5.31%	4.44%	61.05%	70.73%	4.71%	3.57%
SJTU	TreeShade	29.09%	13.76%	4.35%	4.60%	37.85%	11.07%	4.63%	4.78%	58.97%	64.87%	4.39%	4.35%
Average		30.02%	17.84%	3.78%	4.36%	40.84%	16.52%	4.24%	4.81%	65.06%	70.95%	3.90%	4.66%

encoding time savings is also low. The difference between the BDR values compared to the results using the *veryslow* preset is negligible. Figure 13 shows an example frame from the *Bunny* test sequence encoded with SMES and EMES-3 schemes in 1080p resolution at 4.5 Mbps. It is observed that SMES representation used 10% more bits to yield a similar VMAF.

Since EMES-1 yields the least BDR with a substantial encoding time savings, it is an ideal choice for the encoding applications, which are very strict in terms of coding efficiency, and when the encoding time savings is not so crucial. It is observed that EMES-2 is faster than EMES-1 in serial encoding but slower in parallel encoding. Hence, EMES-2 is ideal in serial encoding, where the constraint on the coding efficiency is relaxed. EMES-3 is an ideal choice when the encoding time is constrained, and BDR of about 5% is acceptable.

7 CONCLUSIONS

This article presented efficient multi-encoding schemes for HTTP Adaptive Streaming deployments. The schemes are analyzed by integrating them into the open-source x265 HEVC encoder. The article first explores existing state-of-the-art schemes for multi-rate encoding and multi-encoding. Prediction mode heuristics and motion estimation heuristics are proposed for multi-rate encoding. The experimental results demonstrate that, on average, the proposed heuristics improve the overall encoding time savings of CU depth-bound-based schemes by 12%, with negligible

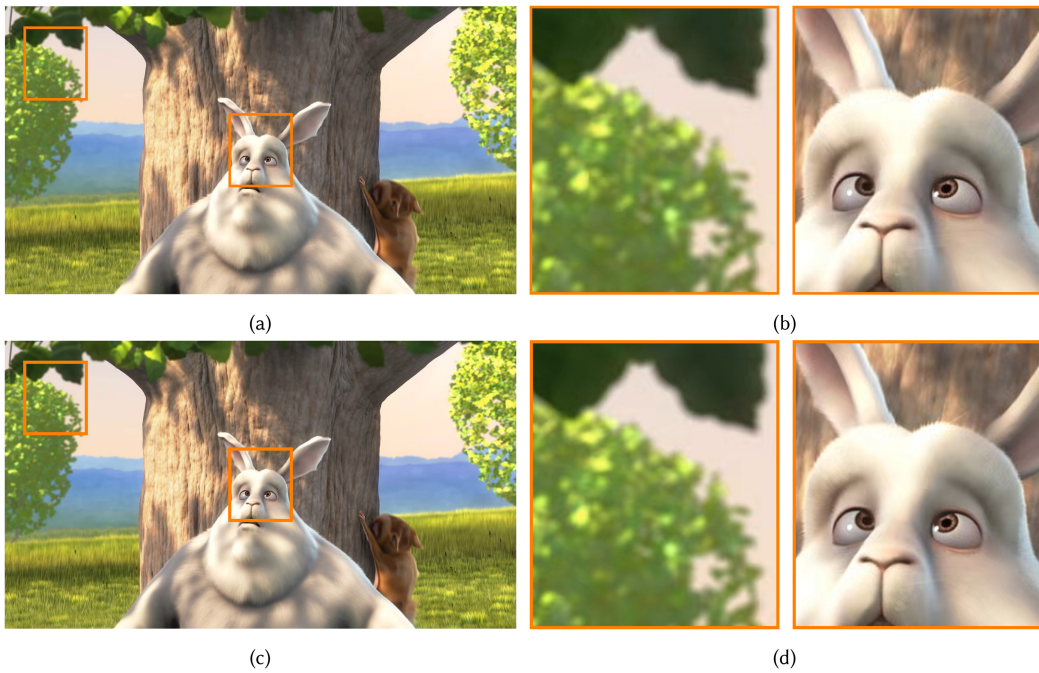


Fig. 13. An example frame from the *Bunny* test sequence encoded at 4.5 Mbps (1080p) using (a) SMES, (b) its zoomed patches, (c) EMES-3, and (d) its zoomed patches. SMES yields a bitrate of 4.955 Mbps for VMAF 90, while EMES-3 yields a bitrate of 4.400 Mbps for VMAF 89.

reduction in compression efficiency. Another efficient scheme for multi-rate encoding is proposed, saving 20.74% of the overall encoding time in parallel encoding with a negligible increase in bitrate. The article then proposes novel multi-encoding schemes that extend the proposed multi-rate encoding schemes across resolutions, emphasizing the best compression efficiency, best compression efficiency-encoding time savings tradeoff, and best encoding time savings, respectively. The experimental results suggest that the multi-encoding schemes (i) optimized for the highest compression efficiency, (ii) optimized for the best compression efficiency-encoding time savings tradeoff, and (iii) optimized for the best encoding time savings reduce the serial encoding time by 34.71%, 45.27%, and 68.76%, respectively, with a 2.3%, 3.1%, and 4.5% respective increase in bitrate to maintain the same VMAF. In parallel encoding, the overall encoding time is reduced by 22.03%, 20.72%, and 76.82%, respectively. The proposed multi-encoding scheme for the best encoding time savings also yields the best compression efficiency-encoding time savings tradeoff.

REFERENCES

- [1] Hadi Amirpour, Ekrem Çetinkaya, Christian Timmerer, and Mohammad Ghanbari. 2021. Towards optimal multirate encoding for HTTP adaptive streaming. In *MultiMedia Modeling: 27th International Conference (MMM'21), Proceedings, Part I*. Springer-Verlag, Berlin, 469–480. https://doi.org/10.1007/978-3-030-67832-6_38
- [2] Hadi Amirpour, Ekrem Çetinkaya, Christian Timmerer, and Mohammad Ghanbari. 2020. Fast multi-rate encoding for adaptive HTTP streaming. In *2020 Data Compression Conference (DCC'20)*. 358–358. <https://doi.org/10.1109/DCC47342.2020.00080>
- [3] Gisle Bjontegaard. 2001. Calculation of average PSNR differences between RD-curves. In *VCEG-M33*.
- [4] Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin. 2018. JVET-J1010: JVET Common Test Conditions and Software Reference Configurations.

- [5] Manri Cheon and Jong-Seok Lee. 2018. Subjective and objective quality assessment of compressed 4K UHD videos for immersive experience. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 7 (2018), 1467–1480. <https://doi.org/10.1109/TCSVT.2017.2683504>
- [6] Jan De Cock, Zhi Li, Megha Manohara, and Anne Aaron. 2016. Complexity-based consistent-quality encoding in the cloud. In *2016 IEEE International Conference on Image Processing (ICIP'16)*. 1484–1488. <https://doi.org/10.1109/ICIP.2016.7532605>
- [7] Johan De Praeter, Antonio Jesús Díaz-Honrubia, Niels Van Kets, Glenn Van Wallendael, Jan De Cock, Peter Lambert, and Rik Van de Walle. 2015. Fast simultaneous video encoder for adaptive streaming. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP'15)*. 1–6. <https://doi.org/10.1109/MMSP.2015.7340802>
- [8] Kalyan Goswami, Bhavna Hariharan, Pradeep Ramachandran, Alex Giladi, Dan Grois, Kavitha Sampath, Aruna Matheswaran, Ashok Kumar Mishra, and Kevin Pikus. 2018. Adaptive multi-resolution encoding for ABR streaming. In *2018 25th IEEE International Conference on Image Processing (ICIP'18)*. 1008–1012. <https://doi.org/10.1109/ICIP.2018.8451485>
- [9] Mateus Grellert, Luis A. da Silva Cruz, Bruno Zatt, and Sergio Bampi. 2021. Coding mode decision algorithm for fast HEVC transrating using heuristics and machine learning. *J. Real-Time Image Process.* 18, 6 (Dec. 2021), 1881–1896. <https://doi.org/10.1007/s11554-020-01063-x>
- [10] Jiawen Gu, Jiangtao Wen, Bichuan Guo, and Yuxing Han. 2018. Multi-representations encoding framework for adaptive http streaming. In *2018 25th IEEE International Conference on Image Processing (ICIP'18)*. 988–992. <https://doi.org/10.1109/ICIP.2018.8451251>
- [11] Angeliki V. Katsenou, Joel Sole, and David R. Bull. 2021. Efficient bitrate ladder construction for content-optimized adaptive video streaming. *IEEE Open Journal of Signal Processing* 2 (2021), 496–511. <https://doi.org/10.1109/OJSP.2021.3086691> Conference Name: IEEE Open Journal of Signal Processing.
- [12] Matheus Lindino, Thiago Bubolz, Bruno Zatt, Daniel Palomino, and Guilherme Correa. 2021. Low-complexity HEVC transrating based on prediction unit mode inheritance. In *2020 28th European Signal Processing Conference (EUSIPCO'21)*. 550–554. <https://doi.org/10.23919/Eusipco47968.2020.9287789>
- [13] Aruna Matheswaran, Praveen Kumar Karadugattu, Pradeep Ramachandran, Alex Giladi, Dan Grois, Pooja Venkatesan, and Alex Balk. 2020. Open source framework for reduced-complexity multi-rate HEVC encoding. In *Applications of Digital Image Processing XLIII*, Andrew G. Tescher and Touradj Ebrahimi (Eds.), Vol. 11510. International Society for Optics and Photonics, SPIE, 461–471. <https://doi.org/10.1117/12.2567877>
- [14] Vignesh V. Menon, Hadi Amirpour, Mohammad Ghanbari, and Christian Timmerer. 2021. Efficient content-adaptive feature-based shot detection for HTTP adaptive streaming. In *2021 IEEE International Conference on Image Processing (ICIP'21)*. 2174–2178. <https://doi.org/10.1109/ICIP42928.2021.9506092>
- [15] Vignesh V. Menon, Hadi Amirpour, Mohammad Ghanbari, and Christian Timmerer. 2022. CODA: Content-aware frame dropping algorithm for high frame-rate video streaming. In *2022 Data Compression Conference (DCC'22)*. 475–475. <https://doi.org/10.1109/DCC52660.2022.00086>
- [16] Vignesh V. Menon, Hadi Amirpour, Mohammad Ghanbari, and Christian Timmerer. 2022. OPTE: Online per-title encoding for live video streaming. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'22)*. 1865–1869. <https://doi.org/10.1109/ICASSP43922.2022.9746745> ISSN: 2379-190X.
- [17] Vignesh V. Menon, Hadi Amirpour, Christian Timmerer, and Mohammad Ghanbari. 2021. Efficient multi-encoding algorithms for HTTP adaptive bitrate streaming. (2021), 1–5. <https://doi.org/10.1109/PCS50896.2021.9477499>
- [18] Vignesh V. Menon, Hadi Amirpour, Christian Timmerer, and Mohammad Ghanbari. 2021. INCEPT: Intra CU depth prediction for HEVC. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP'21)*. 1–6. <https://doi.org/10.1109/MMSP53017.2021.9733517>
- [19] Stefan Radicke, Jens-Uwe Hahn, Christos Grecos, and Qi Wang. 2014. A multi-threaded full-feature HEVC encoder based on wavefront parallel processing. In *2014 International Conference on Signal Processing and Multimedia Applications (SIGMAP'14)*. 90–98. <https://doi.org/10.5220/0005018300900098>
- [20] Yuriy A. Reznik, Xiangbo Li, Karl O. Lillevold, Abhijith Jagannath, and Justin Greer. 2019. Optimal multi-codec adaptive bitrate streaming. In *2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW'19)*. 348–353. <https://doi.org/10.1109/ICMEW.2019.00066>
- [21] Daniel J. Ringis, Francois Pitie, and Anil Kokaram. 2021. Per-clip and per-bitrate adaptation of the Lagrangian multiplier in video coding. In *Applications of Digital Image Processing XLIV*, Vol. 11842. SPIE, 185–194. <https://doi.org/10.1117/12.2593238>
- [22] Damien Schroeder, Adithyan Ilangoan, Martin Reisslein, and Eckehard Steinbach. 2018. Efficient multi-rate video encoding for HEVC-based adaptive HTTP streaming. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 1 (Jan. 2018), 143–157. <https://doi.org/10.1109/TCSVT.2016.2599028>
- [23] Damien Schroeder, Patrick Rehm, and Eckehard Steinbach. 2015. Block structure reuse for multi-rate high efficiency video coding. In *2015 IEEE International Conference on Image Processing (ICIP'15)*. 3972–3976. <https://doi.org/10.1109/ICIP.2015.7351551>

- [24] Iraj Sodagar. 2011. The MPEG-DASH standard for multimedia streaming over the Internet. *IEEE MultiMedia* 18, 4 (April 2011), 62–67. <https://doi.org/10.1109/MMUL.2011.71>
- [25] Li Song, Xun Tang, Wei Zhang, Xiaokang Yang, and Pingjian Xia. 2013. The SJTU 4K video sequence dataset. In *2013 5th International Workshop on Quality of Multimedia Experience (QoMEX'13)*. 34–35. <https://doi.org/10.1109/QoMEX.2013.6603201>
- [26] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (2012), 1649–1668. <https://doi.org/10.1109/TCSVT.2012.2221191>
- [27] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H.264/AVC video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (2003), 560–576. <https://doi.org/10.1109/TCSVT.2003.815165>
- [28] Shih-Hsuan Yang, Chih-Hung Lin, and Hung-Xin Chen. 2019. Object-based rate adjustment for HEVC transrating. In *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW'19)*. 1–2. <https://doi.org/10.1109/ICCE-TW46550.2019.8991793>

Received 26 January 2022; revised 10 October 2022; accepted 20 October 2022