IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 May 2024, accepted 17 July 2024, date of publication 29 July 2024, date of current version 7 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434682

RESEARCH ARTICLE

Energy-Aware Spatial and Temporal Resolution Selection for Per-Title Encoding

MOHAMMAD GHASEMPOUR, (Student Member, IEEE), HADI AMIRPOUR, (Member, IEEE), AND CHRISTIAN TIMMERER, (Senior Member, IEEE)
Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria

Corresponding author: Mohammad Ghasempour (mohammad.ghasempour@aau.at)

This work was supported in part by Austrian Federal Ministry for Digital and Economic Affairs; in part by the National Foundation for Research, Technology and Development; in part by the Christian Doppler Research Association; and in part by the Christian Doppler Laboratory ATHENA (https://athena.itec.aau.at/).

ABSTRACT Video streaming has become an integral part of our digital lives, driving the need for efficient video delivery. With the growing demand for seamless video delivery, adaptive video streaming has emerged as a solution to support users with varying device capabilities and network conditions. Traditional adaptive streaming relies on a predetermined set of bitrate-resolution pairs, known as bitrate ladders, for encoding. However, this ‘‘one-size-fits-all’’ approach is suboptimal when dealing with diverse video content. Consequently, per-title encoding approaches dynamically select the bitrate ladder for each content. However, in an era when carbon dioxide emissions have become a paramount concern, it is crucial to consider energy consumption. Therefore, this paper addresses the pressing issue of increasing energy consumption in video streaming by introducing a novel approach, *ESTR*, which goes beyond traditional quality-centric resolution selection approaches. Instead, the *ESTR* considers both video quality and decoding energy consumption to construct an optimal bitrate ladder tailored to the unique characteristics of each video content. To accomplish this, *ESTR* encodes each video content using a range of spatial and temporal resolutions, each paired with specific bitrates. It then establishes a maximum acceptable quality drop threshold (τ), carefully selecting resolutions that not only preserve video quality above this threshold but also minimize decoding energy consumption. Our experimental results, at a fixed τ of 2 VMAF steps, demonstrate a 32.87% to 41.86% reduction in decoding energy demand for HEVC-encoded videos across various software decoder implementations and operating systems, with a maximum bitrate increase of 2.52%. Furthermore, on a hardware-accelerated client device, a 46.37% energy saving was achieved during video playback at the expense of a 2.52% bitrate increase. Remarkably, these gains in energy efficiency are achieved while maintaining consistent video quality.

INDEX TERMS Video streaming, spatial complexity, temporal complexity, per-title encoding, HEVC, VVC.

I. INTRODUCTION

With the ubiquity of video streaming in the digital age, the efficient delivery of high-quality video content is of paramount concern. As more and more aspects of our lives migrate to online platforms, from entertainment and education to business and communication, the demand for seamless, high-resolution video streaming experiences continues to surge. To meet this increasing demand for video content, advanced compression techniques such as *High Efficiency Video Coding* (HEVC) [1] and *Verstile Video Coding* (VVC) [2] have been developed, which efficiently compress video streams to make the transmission of high-quality videos feasible. However, it comes at a significant cost of increased *energy consumption* [3]. The energy-hungry nature of video streaming has raised critical concerns, not only in terms of operational costs but also concerning its environmental impact. Therefore, optimizing the energy

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Gyu Kim.

consumption associated with the video streaming workflow becomes a pressing challenge for researchers and industry experts [4].

Video streaming relies primarily on *HTTP Adaptive Streaming* (HAS) [5], a technique that divides videos into small segments, typically ranging from 2 s to 10 s in duration. Each segment is encoded in various bitrates and resolutions, referred to as bitrate ladder. This approach ensures that each user receives the most appropriate representation based on their device's capabilities, such as screen resolution and processing power, as well as prevailing network conditions. However, it is essential to note that providing multiple versions of the same content to accommodate adaptivity increases the energy demands of the video streaming workflow, which affects both encoding and decoding energy consumption.

Recently, research efforts have been dedicated to enhancing the energy efficiency of the video encoding process, *e.g.*, for HEVC [6], [7], [8], [9] or VVC [10], [11], [12], [13], [14], [15], [16], [17], [18], [19]. Numerous studies have explored ways to accelerate the encoding process by predicting the best coding modes [6], [9] or by introducing early skip [7], [11], or early termination methods [10]. Alternatively, other approaches seek to simplify individual components of the codec, such as intra-mode decision [8], [12], motion estimation [13], or transform [15] component. Amirpour et al. [20] introduced a recommended preset for each encoding, aiming to balance energy-efficient encoding and video quality.

However, it is essential to note that decoding is more prevalent in *Video on Demand* (VOD) scenarios than in encoding. Within VOD platforms, videos are encoded once on the server and then repeatedly decoded on the client side during multiple viewings. Consequently, as the number of views (or impressions) increases, the significance of the decoding process becomes increasingly apparent. As an example, YouTube reported that the amount of videos encoded is only around 65×10^3 every day, while in the same period, there are about 10^8 videos decoded and viewed [21]. Furthermore, it is reported that people, on average, spend about 17 hours per week watching online video content in 2023 [22]. Netflix recently disclosed that over six months, nearly 100 billion hours were viewed across more than 18 000 titles, accounting for 99% of all viewing on the platform [23]. This massive demand for online video content underscores the crucial need to optimize the energy efficiency of decoding.

In addressing the reduced energy consumption of video decoders, the literature has explored a range of approaches dedicated to optimizing video decoding energy consumption. Some of these enhancements focus primarily on simplifying the decoder components. For example, techniques include disabling the deblocking filter for the largest coding units [24] and simplifying motion compensation by reducing Finite Impulse Response (FIR) filter sizes [25]. Furthermore, an approach for implementing approximate computing in HEVC decoding is introduced in [26]. This method adjusts the interpolation filter of luma and chroma blocks based on an approximation level control parameter. The authors have also defined a skip control parameter to bypass deblocking and Sample Adaptive Offset (SAO) filters as needed for energy savings. Another approach addressing motion compensation and deblocking filter operations is proposed in [27], where a complexity control method is proposed for non-salient areas to enhance subjective video quality. In another study [28], the scalable extensions of HEVC are explored, presenting a method to disable a significant portion of deblocking filter and motion compensation operations in the base layer of the video.

Various studies have considered decoding energy consumption as the third variable within the Rate-Distortion (RD) optimization concept [29], [30], [31]. These methods typically involve modeling decoder energy and selecting the coding mode that minimizes decoding energy consumption at the encoder side, with the cost of losing compression efficiency in terms of RD trade-offs. For instance, in [29], the authors proposed a decoder complexity model and modified the cost function used in the RD optimization process. Similarly, Correa et al. [30] estimate the decoding energy consumption based on the encoding process and employ the Running Average Power Limit (RAPL) tool [32] to measure the actual decoding energy. Their approach achieved a reduction of more than 11% in decoding energy with a 3.7% decrease in compression efficiency in terms of Bjøntegaard Delta Rate (BD-Rate). Furthermore, Herglotz et al. [31] introduces a mathematical theory and develops a new optimization function at the encoder, considering the desired maximum bitrate and decoding energy. They also introduce a tunable parameter to control the balance between bitrate and decoder energy consumption. According to their findings, they achieve a reduction of up to 30% in energy consumption when the bitrate increases in the range of 20 % to 50 %, depending on the video content.

The aforementioned approaches primarily focus on optimizing either the encoding or decoding process for a single encoding. However, the optimization of video decoding within the context of video streaming, *i.e.*, where multiple encodings of the same content are involved, has not yet been addressed. For example, when it comes to per-title encoding [33], [34], [35], the impact on energy is not considered, and only the quality is taken into account. Per-title encoding is a dynamic video compression technique that optimizes encoding parameters, such as resolution, for individual videos. This method selects encoding parameters that yield the highest quality at specific bitrates, enhancing the overall viewer experience.

In this paper, our focus is on decoding energy consumption during the construction of the bitrate ladder. We introduce a novel approach referred to as the Energy-aware Spatial and Temporal Resolution (*ESTR*) selection for per-title

TABLE 1. HLS bitrate-resolution pairs for HEVC/H.265.

bitrate (kbps)	Resolution
145	640×360
300	768×432
600	960×540
900	960×540
1600	960×540
2400	1280×720
3400	1280×720
4500	1920×1080
5800	1920×1080
8100	2560×1440
11600	3840×2160
16800	3840×2160

encoding, which is designed to optimize both video quality and decoding energy consumption by **selecting the most appropriate encoding parameters**, such as spatial resolution and temporal resolution (framerate), for each bitrate. Our proposed approach does not employ any changes to the implementations or configurations of the encoder or decoder, making it easily applicable to existing streaming systems. Moreover, we propose a control method to balance video compression efficiency and decoding energy consumption.

The remainder of the paper is organized as follows. Section II provides background information on basic per-title encoding and illustrates the motivation behind our approach through examples. Section III describes the proposed *ESTR* approach in detail, while Section IV presents the experimental results. Finally, Section V concludes the paper.

II. BACKGROUND AND MOTIVATING EXAMPLES

This section provides an overview of the background related to bitrate ladder construction and basic per-title encoding. It performs a comparative analysis of energy consumption across various resolutions and framerates. These insights serve as a basis for the *ESTR* approach presented in this paper.

A. BACKGROUND

In HAS, to accommodate the heterogeneous environment consisting of devices with varying bandwidth and device types, the same video content is made available in multiple representations, often encoded using a fixed bitrate ladder consisting of predetermined bitrate-resolution pairs. For instance, Table 1 provides a fixed bitrate ladder recommended by Apple [36], known as *HTTP Live Streaming* (HLS) bitrate ladder. It comprises 12 different bitrate-resolution pairs, with bitrates ranging from 145 kbps to 16800 kbps and their associated spatial resolution ranging from 360p to 2160p. This "one-size-fits-all" bitrate ladder is easy to implement; however, it is suboptimal as it ignores the characteristics of the video content.

Video content exhibits varying responses to compression, influenced by factors such as complexity. Easy-to-encode videos tend to achieve high-quality output at lower bitrates. However, as videos become more complex in terms of spatial and temporal complexity, the bitrates required to attain high-quality output increase correspondingly. Therefore, the bitrate range to avoid bandwidth wastage for easy-to-encode videos and achieve high quality for hard-to-encode videos strongly depends on the video content.

Another effective approach to enhance the quality of encoding at a given bitrate is to determine the optimal resolution for encoding. In scenarios with constrained bitrates, such as lower bandwidth environments, videos often lack the necessary bitrate allocation to be encoded at higher resolutions like 4K. Consequently, encoding them at lower resolutions ensures that individual video frames receive a sufficient bitrate budget to achieve high quality. However, this approach introduces a potential downside – the upscaling artifact – especially when these lower-resolution videos are displayed on high-resolution screens. The interplay between resolution selection and encoding quality is a critical consideration in the realm of video compression.

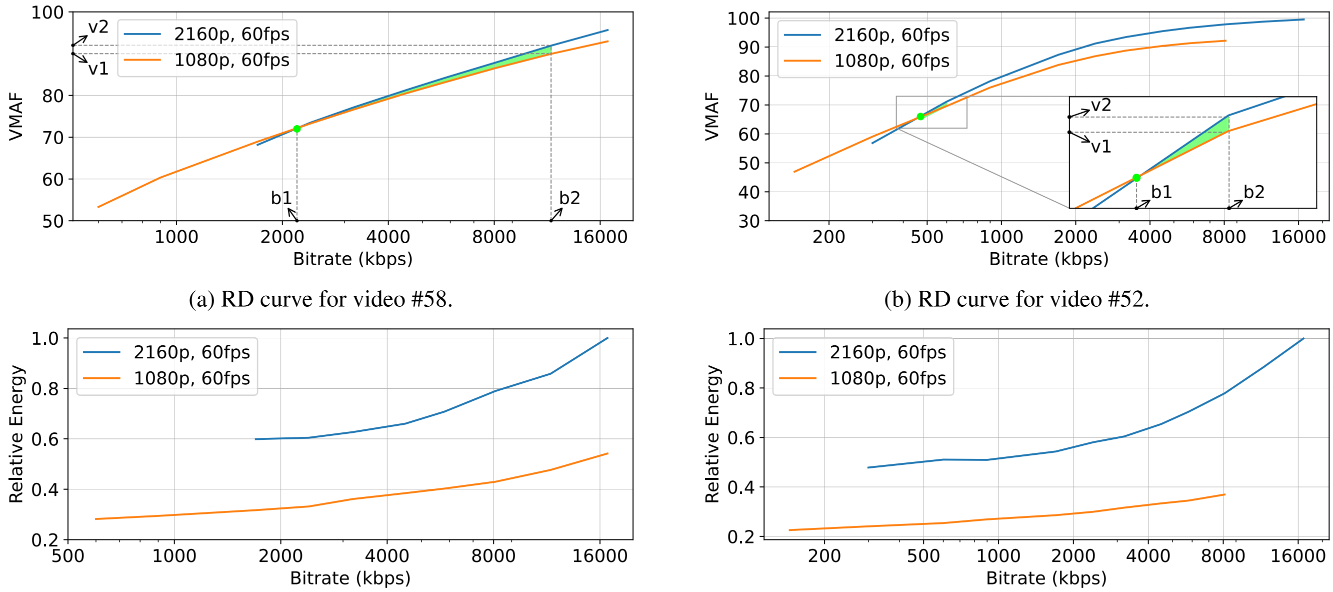
B. MOTIVATING EXAMPLES

Figure 1 shows the rate-distortion (RD) curves for two selected videos, #58 and #52, from the Inter4K dataset [37], as well as their corresponding relative decoding energy consumption. These videos have been encoded with bitrates ranging from 145 kbps to 16800 kbps, under two distinct scenarios: (*i*) at their original spatial resolution of 4K, and (*ii*) at a reduced spatial resolution of 1080p. In the latter case, the videos were upscaled to 4K using the bicubic method, and their quality was evaluated using Video Multi-method Assessment Fusion (VMAF) [38]. VMAF is a video quality assessment metric developed by Netflix to predict the perceived quality score of video content. In Figure 1 (a) and (b), the intersection between the two RD curves occurs at $b1$, denoted by a green dot. Beyond this point, the quality of the 1080p resolution becomes lower than that of the 2160p resolution. This difference remains below a threshold value until the bitrate $b2$, where the quality difference between the two representations (*i.e.*, $v2 - v1$) exceeds the threshold. The specified area, shown in green, is of interest because the quality difference between the two resolutions remains below a threshold (in this case, 2 VMAF points) but the decoding energy consumption of the 1080p resolution is significantly less than that of the 2160p resolution. It should be noted that the decoding energy consumption was measured with the CodeCarbon tool[1] and normalized to the highest energy consumption of each video representation.

This analysis yields three notable observations:

- The intersection bitrate ($b1$) is subject to variation depending on the distinct characteristics of the video content, serving as the foundational principle behind per-title encoding. In this approach, the optimal spatial resolution is

[1] https://codecarbon.io/, last access: Jan 8, 2024



(a) RD curve for video #58.

(b) RD curve for video #52.

(c) relative decoding energy consumption for video #58.

(d) relative decoding energy consumption for video #52.

FIGURE 1. **RD curves for video sequences (a) #58 and (b) #52, as well as the relative decoding energy consumption for video sequences (c) #58 and (d) #52, encoded at two different resolutions.**

chosen for each piece of content within a specified bitrate range. For instance, as shown in Figure 1, the intersection bitrate between the 1080p and 2160p resolutions occurs at $b1 = 2,300\,kbps$ for video #58, while it occurs at $b1 = 470\,kbps$ for video #52.

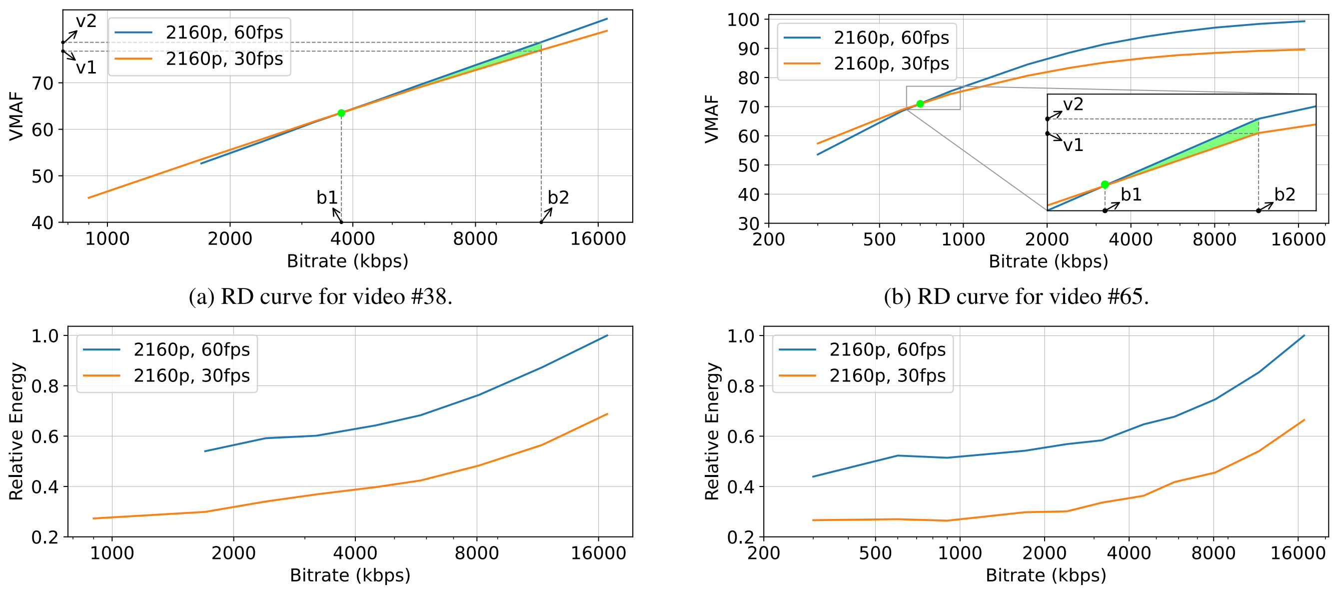
- The observation regarding the relatively insubstantial difference in quality between two representations, particularly within a specific bitrate range centered around the intersection bitrate, underpins the premise of this paper. For example, as shown in Figure 1, the quality between 1080p and 2160p resolutions remains similar over a broad bitrate range (*i.e.*, $b1$ to $b2$) after the intersection bitrate for video #58, highlighted in green. Conversely, for video #52, the quality difference between the two spatial resolutions becomes substantial after the intersection bitrate, and it only remains similar for a very narrow bitrate range.
- The observation regarding the consistent relative decoding energy consumption of these video representations further motivates the selection of a spatial resolution that addresses the trade-off between video quality and decoding energy consumption. As shown in Figure 1, when comparing video #58 and video #52, it becomes evident that the relative difference in decoding energy consumption remains fairly consistent, or at least not significantly divergent, within the bitrate range.

Therefore, choosing the lower spatial resolution within this specific bitrate range centered after the intersection point for both videos, which offers similar quality to 2160p resolution, will result in a noticeable reduction in decoding energy consumption. For example, in the case of video #58, after the intersection bitrate at 2,300 kbps, there is a substantial bitrate range, highlighted in green, where the quality of 1080p remains comparable to that of 2160p. This presents the opportunity to choose 1080p as the spatial resolution rather than 2160p, resulting in a reduction in decoding energy consumption with a negligible drop in quality. However, for video sequence #52, this bitrate range is relatively small.

Similar observations are noted when it comes to the temporal resolution or framerate. Figure 2 shows the RD curves for two selected videos, #38 and #65, from the Inter4K dataset [37], as well as their corresponding relative decoding energy consumption. These videos have been encoded with bitrates ranging from 145 kbps to 16800 kbps, in two distinct scenarios: (*i*) at their original framerate of 60 fps, and (*ii*) at a reduced framerate of 30 fps obtained by sampling every other frame. In the latter case, each removed frame is reconstructed using a frame duplication process based on its preceding frame to calculate the VMAF.

The following observations were made:

- The intersection bitrate ($b1$) is subject to variation depending on the distinct characteristics of the video content. For example, as shown in Figure 2, the intersection bitrate between the framerates 30 fps and 60 fps occurs at ($b1 = 3,900\,kbps$) for video #38, while it occurs at ($b1 = 700\,kbps$) for video #65.
- The minimal quality difference is notable between the framerates 30 fps and 60 fps, especially within a bitrate range (*i.e.*, $b1$ to $b2$). For example, in Figure 2, the quality stays similar over a broad bitrate range after $b1$ for video #38 (highlighted in green). However, for video #65, the quality difference becomes significant beyond the intersection, remaining similar in only a narrow bitrate range.



(a) RD curve for video #38.

(b) RD curve for video #65.

(c) relative decoding energy consumption for video #38.

(d) relative decoding energy consumption for video #65.

FIGURE 2. RD curves for video sequences (a) #38 and (b) #65, as well as the relative decoding energy consumption for video sequences (c) #38 and (d) #65, encoded at two different framerates and 4K resolution.

- The consistent relative decoding energy consumption of these videos motivates selecting a lower framerate within a specific bitrate range. Comparing videos #38 and video #65 in Figure 2 reveals that the decoding energy remains fairly consistent within the bitrate range. For example, after $b1 = 3,900\,kbps$ for video #38, there is a substantial bitrate range (highlighted in green) where the quality of 30 fps remains similar to 60 fps. This offers an opportunity to choose 30 fps for reduced decoding energy consumption with a minimal quality drop. However, for video #65, this bitrate range is smaller.

In summary, it is evident that decoding energy consumption significantly depends on both the spatial and temporal resolution of the video, aligning with findings in multiple studies [39], [40]. This disparity offers an opportunity to optimize spatial and temporal resolution selection, not only by prioritizing quality but also by considering energy efficiency.

III. ENERGY-AWARE SPATIAL-TEMPORAL RESOLUTION SELECTION

In Section II, we presented the RD curves for various representations across multiple videos in Figure 1 and Figure 2. It was shown that each representation provides a different level of quality at each specific bitrate. Nevertheless, encoding a video at every possible bitrate is impractical due to resource constraints. Thus, each video representation is encoded at a set of predefined bitrate values, which are the steps in the bitrate ladder (*e.g.*, as listed in Table 1). Using these bitrate points on each video's RD curve, a Pareto Front (PF), often referred to as the

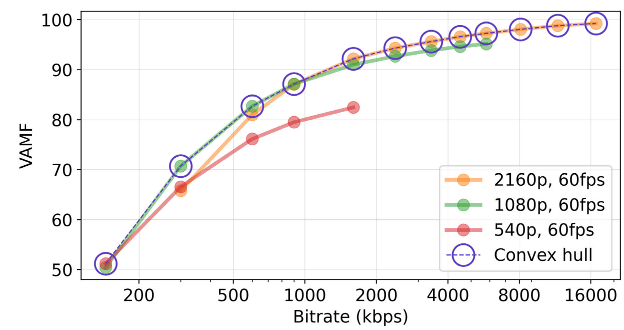


FIGURE 3. Example of RD curves and convex hull construction for the video sequence #32.

convex hull, can be constructed. This convex hull is then employed to determine the bitrate-representation pairs of the bitrate ladder. An example of the convex hull is shown in Figure 3 using empty blue circles. For instance, at bitrates of 145 kbps, 300 kbps, and 1600 kbps, the video representations with resolutions of 540p, 1080p, and 2160p provide the highest quality. The bitrate ladder is built from the set of representations that offer the highest quality at each specific bitrate. This is exactly the approach taken by quality-centric methods like basic per-title encoding [33], where the highest quality representation is selected at predefined bitrate ladder steps.

However, such methods tend to neglect the energy consumption of video representations, which can lead to huge energy consumption in the streaming process. In response to this challenge, we introduce *ESTR*, which is designed to enhance energy efficiency in the streaming ecosystem without requiring modifications to the decoder or encoder implementations, making it compatible with

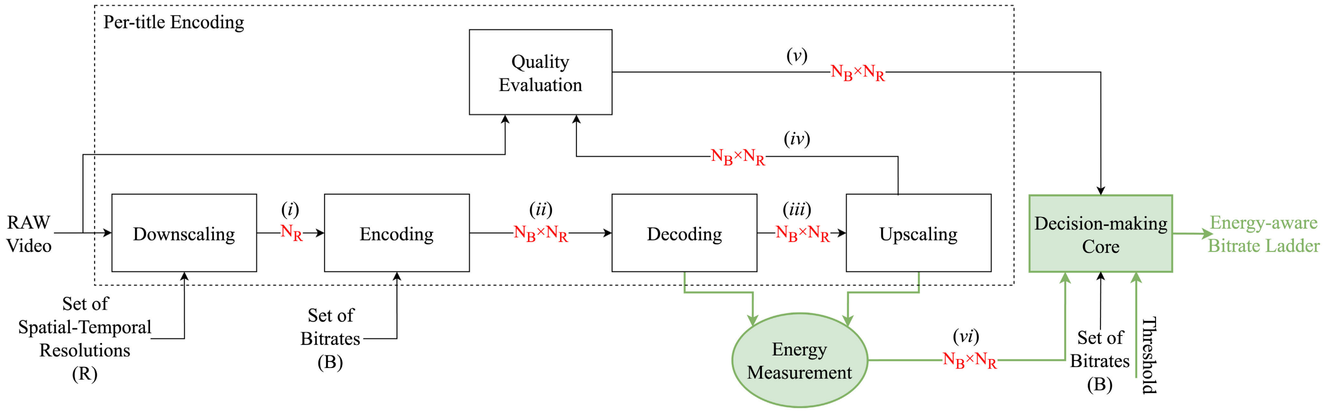


FIGURE 4. **The workflow of *ESTR*.**

existing streaming systems. We first describe the workflow of *ESTR*, and we then explain how the core decision-making module works.

A. THE WORKFLOW

The workflow of the proposed energy-aware per-title encoding approach is shown in Figure 4. The proposed *ESTR* employs the essential elements of basic per-title encoding [33] (enclosed within the dotted box). However, it introduces additional components, including the decoding energy measurement element, a threshold, and the core decision-making block (highlighted in green). To simplify the workflow of *ESTR*, we have assigned numerical labels to the inputs and outputs of intermediate components, which will be elaborated in the following. In addition, the red symbols on the arrows indicate the number of instances passing from one element to the next element.

For a given input video sequence and a set of bitrate values, which are meant to be the steps in the bitrate ladder, *ESTR* constructs a decoding energy-aware bitrate ladder. As shown in Figure 4, the downscaling module acquires the input video sequence at its original resolution and framerate, along with a set of spatial-temporal resolutions, denoted as R. It then generates (*i*) downscaled versions of the input video sequence at the set of various framerates and resolutions (*i.e.*, R). Subsequently, the encoder processes all of these videos using a set of input bitrates, denoted as B, to generate (*ii*) multiple video representations. Each representation is subsequently sent to a standard decoder to produce (*iii*) downscaled raw videos, which are later upscaled by the upscaling module to restore them to (*iv*) their original spatial-temporal resolution. At this point, the quality evaluation module evaluates (*v*) the quality of this encoding by considering both the original video and the processed one. Finally, *ESTR* utilizes (*vi*) decoding energy consumption and quality of representations, the set of bitrates (B), and a tunable threshold to construct the energy-aware bitrate ladder.

To explain the functionality of *ESTR*, let us consider three sets: a set of spatial resolutions, denoted as $S = \{s_i \mid i \in \{0, 1, \ldots, N_S - 1\}\}$, a set of temporal resolution (framerate), denoted as $F = \{f_j \mid j \in \{0, 1, \ldots, N_F - 1\}\}$, and the predefined bitrate values, denoted as $B = \{b_k \mid k \in \{0, 1, \ldots, N_B - 1\}\}$. These bitrate values form the bitrate ladder and depend on the service provider's choice and user requirements. For example, one might adopt HLS bitrate ladder values (provided in Table 1) as the predefined bitrates and encode the videos accordingly. After the downscaling process, the number of raw videos, referred to as N_R, is determined as follows:

$$N_R = N_S \times N_F \tag{1}$$

We now have a set of raw videos denoted as $R = \{r_l \mid l \in \{0, 1, \ldots, N_R - 1\}\}$, each of which is then encoded at various bitrates (B) of the bitrate ladder. Therefore, after the encoding process, we have $N_B \times N_R$ representations for each video sequence.

After decoding, upscaling, and quality measurement, each representation with a bitrate k and spatial-temporal resolution l has a quality $q_{k,l}$ in comparison to its original raw version. In addition, it has a value of $e_{k,l}$, indicating the amount of energy consumption during the decoding process. It is important to note that the quality and energy consumption values are measured after the temporal/spatial upscaling processes. This implies that the downscaled video is first upscaled to the original resolution and framerate before being compared to the original raw version. Additionally, the energy consumption includes both decoding and upscaling processes. Now, we establish two sets, Q and E, which include the quality and energy consumption values for all $q_{k,l}$ and $e_{k,l}$, where $k \in \{0, 1, \ldots, N_B - 1\}$ and $l \in \{0, 1, \ldots, N_R - 1\}$.

Having the sets Q and E at hand, it becomes feasible to identify the highest-quality representation and its quality difference compared to other representations at each specific bitrate. The following explanation details the

Algorithm 1 *ESTR* Bitrate Ladder Construction

Data: Set of qualities (Q), set of decoding energy consumption values (E), set of bitrates (B), set of spatial-temporal resolutions (R), quality threshold (τ)
Result: Energy-aware bitrate ladder (*EBL*)

```
EBL ← ∅
for k=0 to N_B do
    l_max ← arg max(Q[k])
    selected ← l_max
    for l=0 to N_R do
        if ((Q[k][l_max] − Q[k][l]) < τ) then
            if (E[k][l] < E[k][selected]) then
                selected ← l
    EBL.append((B[k], R[selected]))
return EBL
```

decision-making core to *ESTR*, which utilizes these sets to construct the energy-aware bitrate ladder.

B. DECISION-MAKING CORE

Before introducing the decision-making module, we introduce a tunable parameter denoted as (τ). This parameter serves as a tool for fine-tuning the trade-off between video compression efficiency and decoding energy consumption based on the service provider's considerations, offering flexibility in optimizing the energy efficiency of the video streaming workflow. The mechanism of this parameter operates such that if the video quality differences between the highest quality representation and one or more other representations at a specific bitrate fall below the threshold, the representation with the lowest decoding energy consumption is selected to construct the bitrate ladder. Therefore, the higher the value of τ, the more energy is saved per each decoding process. However, this energy saving comes at the cost of reduced compression efficiency. It should be mentioned that τ is a continuous value defined based on the chosen video quality metric. For instance, if VMAF is the quality metric, the unit of τ aligns with that of VMAF.

Given the defined threshold and the measured sets of Q and E, the representations for the energy-aware bitrate ladder construction are determined using the pseudo-code presented in Algorithm 1. At each bitrate, the algorithm tries to identify the representation that offers the highest quality and calculates the quality difference for other video representations. Subsequently, among the candidate representations with quality differences below the threshold, the algorithm aims to select the one with the lowest decoding energy consumption. This iterative process is carried out for every bitrate step, reaching to the creation of the set of representations, which constitutes our energy-aware bitrate ladder. Within the algorithm, the index *selected* specifies the index of the chosen representation for the specific bitrate. If none of the representations meets the threshold condition or their energy consumption is not lower than that of the highest quality representation, the *selected* index remains unchanged, retaining the index of the highest quality representation, which is initialized for each bitrate in the algorithm.

IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results for the *ESTR* approach, designed to select the most suitable spatial and temporal resolutions of each video content while considering decoding energy consumption. We evaluate the performance of the proposed approach compared to basic per-title and fixed-resolution streaming. For this purpose, we first explain the dataset and codec configurations utilized in IV-A. Subsequently, in Section IV-B, we describe the metrics employed for quality assessment and energy consumption. In Section IV-C, we conduct a performance evaluation of the *ESTR* comparing it against two different video streaming approaches. The first approach involves maintaining a constant spatial and temporal resolution for all bitrates, where we consider two different spatial resolutions and two temporal resolutions (framerates) to enhance the depth of this comparative analysis. The second approach involves the utilization of two "one-size-fits-all" bitrate ladders. Following that, in Section IV-D, we analyze how adjusting the threshold value (τ) affects both the decoder energy consumption and compression efficiency. Furthermore, in Section IV-E, we explore the impact of various video decoder implementations and end-device operating systems. Finally, in Section IV-F, we apply our method to a range of video codecs, including HEVC, VVC, and AVC [41], and evaluate its performance.

A. TEST SEQUENCES AND ENCODER CONFIGURATIONS

We collected the first 100 video sequences from the Inter4K dataset [37], each of which was trimmed to contain 64 frames. These sequences have a native spatial resolution of 3840×2160, utilize a 4:2:0 chroma format, and maintain a temporal resolution (framerate) of 60 fps with an 8-bit pixel depth. We considered five spatial resolutions, specifically S = {2160p, 1440p, 1080p, 720p, 540p} and applied downscaling to each original video sequence using the bicubic method. In addition to the 60 fps video sequences, we generated their 30 fps versions by selecting every other frame of each spatial resolution. Consequently, we have 10 different downscaled versions (N_R = 10) for each video sequence. Following this, we encoded all downscaled videos using the HEVC encoder in the *random_access* configuration, utilizing a Group of Picture (GOP) size of 32 frames, and employed the constant bitrate method with the required target bitrates for experiments, which was the bitrates in the HLS authoring specification [36] (*cf.* Table 1) and the *Multi-codec Dynamic Adaptive Streaming over HTTP* (MC-DASH) bitrate ladder [42]. Subsequently, these video representations were decoded and upscaled to the original

TABLE 2. The performance comparison of *ESTR* approach with fixed bitrate ladders.

	Fixed spatial-temporal resolution				One-size-fits-all	
Spatial resolution / Temporal resolution	2160p 60 fps	540p 60 fps	2160p 30 fps	540p 30 fps	HLS	MC-DASH
BD-Rate %	-1.37	-48.70	-49.12	-57.39	-36.14	-32.42
BD-VMAF	0.17	13.00	11.72	17.54	7.08	6.32
BDDE %	-29.88	12.80	-10.17	78.18	17.35	26.98

resolution (*i.e.*, 2160p at 60 fps), utilizing frame duplication and bicubic filter. The entire HEVC encoding and decoding process was performed using FFmpeg v6.1.1.[2] We used an Apple Mac mini with an Octa-Core Apple M1 processor, and 16 GB of RAM, running macOS Ventura (version 13.3.1) for conducting these experiments.

B. METRICS

The performance of the proposed approach is evaluated in terms of video quality, compression efficiency, and energy consumption. Video quality is assessed using two commonly used metrics, PSNR and VMAF, with reference to the original YUV sequence (*i.e.*, 2160p at 60 fps). For compression efficiency, we employed Bjøntegaard Delta (BD) metrics [43]. The BD-Rate metric quantifies bitrate savings at the same video quality, and BD-VMAF metric specifies the amount of quality degradation at the same bitrate. To evaluate the energy efficiency of the decoder, we used Bjøntegaard Delta Decoding Energy (BDDE), introduced in [31], which measures energy savings as a percentage for the same video quality. A negative BDDE indicates that the same quality is achieved with reduced energy consumption. Additionally, we quantified the energy consumption associated with decoding each video representation using the CodeCarbon tool, a lightweight and open-source software package for estimating carbon dioxide emissions from computing resources.

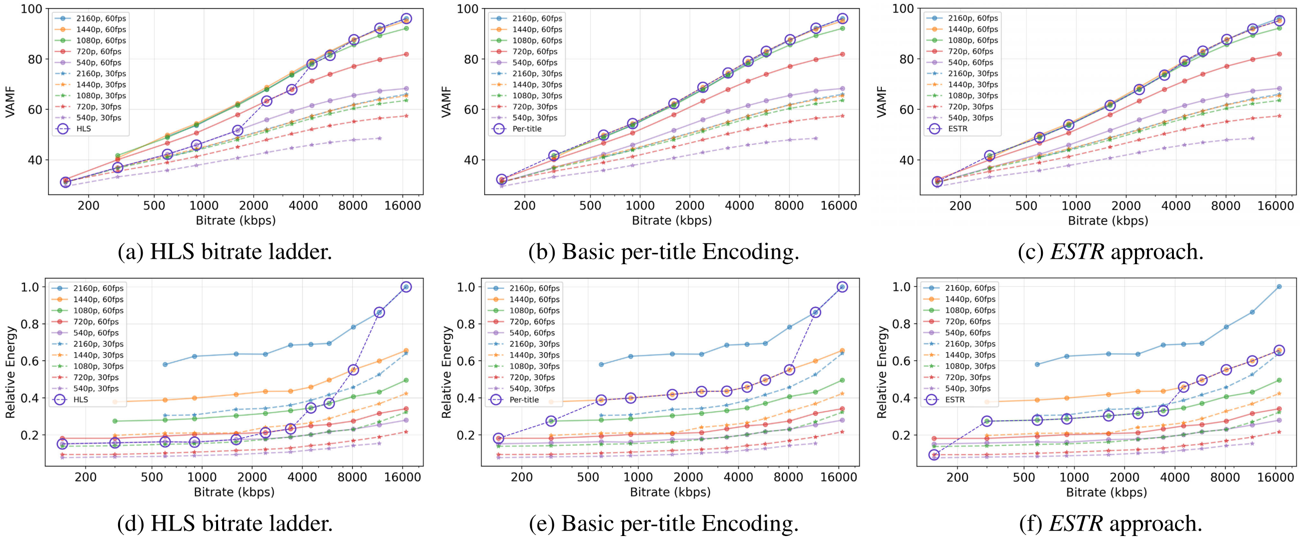
C. COMPARISON WITH FIXED BITRATE LADDERS

This section aims to compare the performance of the proposed approach at fixed spatial and temporal resolutions, as well as the "one-size-fits-all" bitrate ladders for HEVC-encoded videos. Fixed spatial and temporal resolution ladder refers to a scenario where all video sequences at different bitrates are encoded at the same spatial and temporal resolution. Furthermore, we included two different predefined bitrate ladders for our study, *i.e.*, the HLS bitrate ladder for HEVC and the MC-DASH bitrate ladder for HEVC. The results are summarized in Table 2.

According to the results, the proposed approach demonstrates an approximate 30% reduction in decoding energy consumption compared to streaming at the original resolution (3840 × 2160 at 60 fps), while achieving a bitrate saving of around 1%. This means that streaming at the original resolution achieves near-optimal compression efficiency, indicated by the RD curve of encoding at the original resolution closely resembling the convex hull. However, it is important to note that the energy consumption remains relatively high, and there may be compatibility issues with devices that do not support 4K decoding. On the other hand, when compared to streaming at the lowest resolution (960 × 540 at 30 fps), *ESTR* achieves a superior compression efficiency of over 57%, measured in terms of BD-Rate, with an accompanying increase in energy consumption of 78%. Furthermore, the proposed approach provides 36.14% and 32.42% higher compression efficiency and requires 17.35% and 26.98% more decoding energy consumption compared to HLS and MC-DASH bitrate ladders, respectively. It is because, the proposed approach dynamically chooses the spatial-temporal resolution for each bitrate considering both quality and decoding energy consumption for each video content, as opposed to relying on a fixed set of spatial-temporal resolutions for all types of content.

To provide a more detailed explanation, Figure 5 and Figure 6 show examples of selected representation by the HLS bitrate ladder, basic per-title encoding, and *ESTR* with $\tau = 1.00$, along with their corresponding quality and decoding energy consumption for videos #14 and #40. Please note that the basic per-title encoding method only selects the spatially downscaled representation. However, we have displayed all representations for all approaches in these figures for consistency reasons. The energy consumption values are normalized against the maximum energy consumption, which in this case is the energy consumed energy for decoding 3840 × 2160 at 60 fps, measured by the CodeCarbon tool. As can be seen, the HLS bitrate ladder opts for lower spatial-temporal resolutions in lower bitrates, resulting in lower energy consumption. However, the compression efficiency of these selections is lesser than that of basic per-title encoding and the *ESTR* approaches. On the contrary, the basic per-title encoding approach selects the highest-quality representation, usually from higher spatial-temporal resolutions, leading to greater energy consumption. The proposed *ESTR* approach achieves comparable quality to basic per-title encoding, but by selecting representations with lower energy consumption, it saves energy in comparison.

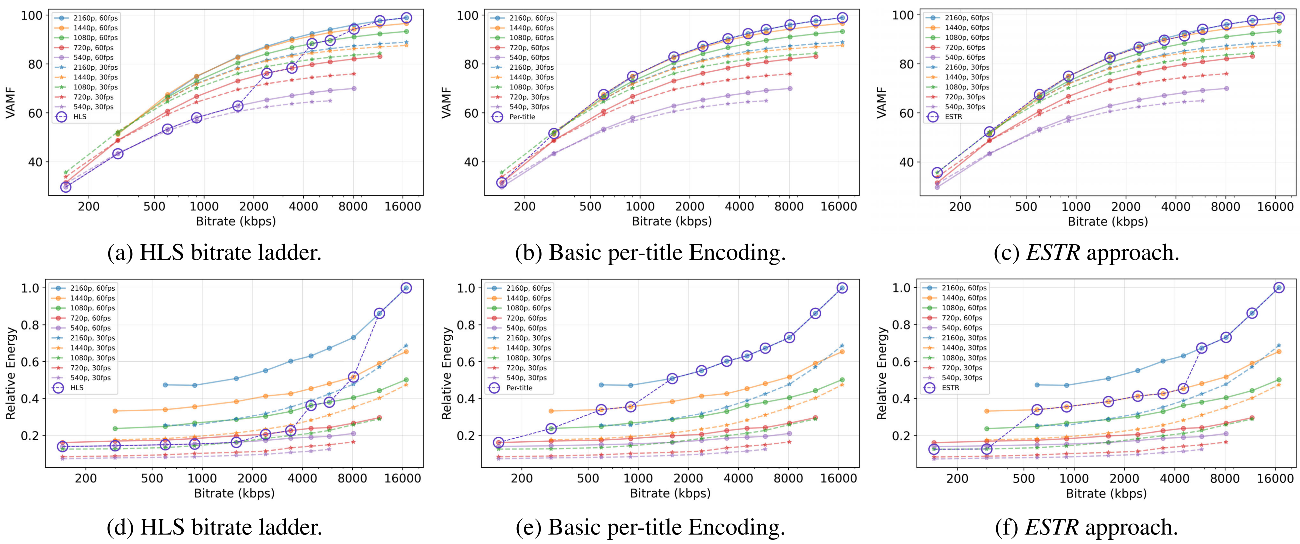
[2]https://www.ffmpeg.org/, last access: Jan 8, 2024



(a) HLS bitrate ladder. (b) Basic per-title Encoding. (c) *ESTR* approach.

(d) HLS bitrate ladder. (e) Basic per-title Encoding. (f) *ESTR* approach.

FIGURE 5. The rate-quality curves for video sequence #14 when utilizing (a) HLS bitrate ladder, (b) basic per-title encoding, (c) *ESTR* approach and their relative decoding energy consumption when employing (d) HLS bitrate ladder, (e) basic per-title encoding, (f) *ESTR* approach.



(a) HLS bitrate ladder. (b) Basic per-title Encoding. (c) *ESTR* approach.

(d) HLS bitrate ladder. (e) Basic per-title Encoding. (f) *ESTR* approach.

FIGURE 6. The rate-quality curves for video sequence #40 when utilizing (a) HLS bitrate ladder, (b) basic per-title encoding, (c) *ESTR* approach and their relative decoding energy consumption when employing (d) HLS bitrate ladder, (e) basic per-title encoding, (f) *ESTR* approach.

The amount of energy savings also depends on the threshold value τ, which is thoroughly investigated in the following section.

D. IMPACT OF THRESHOLD (τ)

The proposed approach enables video streaming providers to manage decoding energy consumption on the client side by adjusting the threshold τ provided by *ESTR*. To achieve higher energy saving at the decoder, the video service provider can increase τ, keeping compression efficiency within an acceptable range. This range can be customized by the provider and may vary across different streaming services. This range is determined by the threshold value τ, which can be interpreted as the maximum allowable quality degradation for selecting a lower energy consumption representation. It means that if the quality difference of several representations, compared to the highest quality option, is within this threshold, they become candidates for selection. Among these candidates, the one with the lowest energy consumption will be chosen. It is worth noting that the threshold is based on video quality metrics; for example, if VMAF is used to measure video quality, the threshold will be expressed in VMAF steps.

To evaluate the impact of threshold values, we compared the *ESTR* to basic per-title encoding [33] with varying threshold values ranging from 0.5 to 5.0, increasing by

TABLE 3. The impact of changing the threshold value on the *ESTR* performance in terms of BD-Rate, BD-VMAF, and BDDE.

τ	BD-Rate %	BD-VMAF	BDDE %						
			Device 1 (Apple Mac)			Device 2 (Lenovo Linux)			Device 3 (Lenovo Windows)
			FFmpeg	HM	FFplay	FFmpeg	HM	FFplay	VLC (Hardware Accelerated)
0.50	-2.44	0.27	-21.37	-22.50	-19.58	-20.54	-21.56	-18.03	-22.35
0.75	-2.12	0.21	-24.94	-26.80	-23.79	-23.30	-25.76	-21.21	-27.14
1.00	-1.54	0.13	-27.69	-30.10	-27.04	-25.26	-29.01	-23.40	-31.93
1.25	-0.73	0.00	-31.18	-34.54	-31.15	-28.11	-33.27	-26.60	-36.75
1.50	0.09	-0.15	-33.84	-37.55	-34.25	-30.07	-36.20	-28.80	-40.00
1.75	1.09	-0.30	-36.44	-40.31	-37.15	-31.93	-38.90	-30.72	-43.35
2.00	2.52	-0.49	-39.20	-43.25	-40.22	-33.96	-41.86	-32.87	-46.37

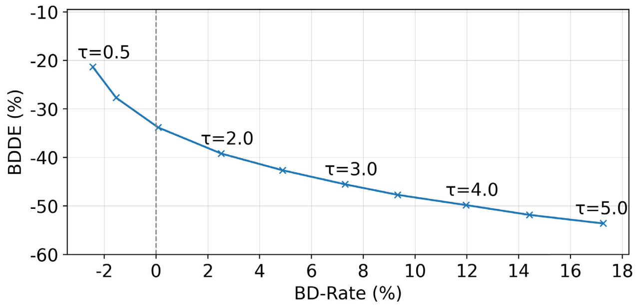


FIGURE 7. The evaluation of the *ESTR* at different threshold values in terms of BDDE (vertical axis) and BD-Rate (horizontal axis) compared to the basic per-title encoding.

0.5 each step. This comparison is shown in Figure 7. For simplicity, specific threshold values are annotated, and each marker represents a 0.5-step progression in the figure. It is shown that increasing the threshold to higher values leads to increased energy savings and BD-Rate. For instance, with $\tau = 2.0$, *ESTR* could save approximately 40% of decoding energy at the cost of 2.5% compression efficiency compared to basic per-title encoding. In addition, for $\tau < 1.5$ (Gray dashed line), *ESTR* outperformed basic per-title encoding in terms of compression efficiency and decoding energy consumption.

It is important to note that the enhanced performance is achieved at the expense of increased server-side energy consumption for encoding, as it considers both spatial and temporal resolutions. In contrast, the basic per-title encoding approach [33] optimizes only based on spatial resolution. To assess this, we initially encoded all 100 test videos at 60 fps and measured the associated energy consumption. We then compared this to the energy consumed when encoding the complete set of spatial-temporal resolutions specified in IV-A. On average, the encoding process required by *ESTR* consumed 1.51 times more energy compared to basic per-title encoding.

E. THE IMPACT OF VIDEO DECODER AND OPERATING SYSTEM

In this section, we assess how different decoders and client devices (including operating systems) influence the performance of the *ESTR* in comparison to the basic per-title encoding for various thresholds τ. The test setup involved three computing devices: device 1, an Apple Mac mini (as mentioned IV-A); device 2, a Lenovo ThinkPad P1 Gen2 laptop running Linux Ubuntu 22.04.3 LTS with an Intel Core i7-9750H CPU at 2.60 GHz, 16 GB of RAM, and Nvidia Quadro T1000 Mobile GPU; and device 3, the same Lenovo laptop running Windows 11. For the evaluation, two different HEVC video decoders (*i.e.*, FFmpeg v6.1.1, HM-18.0 (HEVC's reference software) [44]) and two players (*i.e.*, FFplay v6.1.1, and VLC v3.0.20[3]) were used. We also explored the influence of hardware acceleration on video decoding by testing GPU decoding with VLC, specifically on device 3. The decoding energy consumption of HM and FFmpeg software tools was measured after upscaling to the original spatial-temporal resolution, and for FFplay and VLC, the full-screen option was enabled, with the video displayed on an external 4K monitor. The results are shown in Table 3.

The results lead to several key conclusions. Firstly, while the decoding energy was measured solely with Device 1 using FFmpeg, *ESTR* demonstrates relatively consistent energy savings across various hardware and software implementations and operating systems. Secondly, when it comes to video decoders, *ESTR* showed greater energy savings when HM is used for video decoding, compared to when FFmpeg is used for both devices. This outcome is consistent with HM being the reference software for HEVC decoding, whereas FFmpeg represents a more optimized version of the HEVC decoder. Thirdly, regardless of end-device hardware and operating system, at $\tau = 1.50$, where the compression efficiency of both *ESTR* and basic per-title encoding is nearly identical (*i.e.*, BD-Rate ≈ 0), *ESTR* provides energy savings of 28.8% to 40%, depending on the device and operating system. Furthermore, according to the results, when the VLC player was utilized at the client side with enabled hardware acceleration, *ESTR* delivers energy saving ranging from 22.35% to 46.37%, with a maximum of 2.52% increase in BD-Rate. When FFplay is employed for video playback, *ESTR* provides energy savings of 32.87% and 40.22% for

[3]https://www.videolan.org/vlc/, last access: Jan 8, 2024

TABLE 4. The performance of *ESTR* compared to the basic per-title encoding across various codecs.

τ	AVC (FFmpeg libx264)			HEVC (FFmpeg libx265)			VVC (VVdeC)		
	BD-Rate %	BD-VMAF	BDDE %	BD-Rate %	BD-VMAF	BDDE %	BD-Rate %	BD-VMAF	BDDE %
0.50	-5.01	0.83	-14.46	-2.44	0.27	-21.37	-0.70	0.08	-17.24
0.75	-4.83	0.79	-15.29	-2.12	0.21	-24.94	-0.01	0.00	-20.97
1.00	-4.46	0.74	-15.73	-1.54	0.13	-27.69	0.93	-0.11	-24.40
1.25	-4.17	0.64	-16.23	-0.73	0.00	-31.18	1.86	-0.23	-26.89
1.50	-4.02	0.57	-16.52	0.09	-0.15	-33.84	3.48	-0.39	-29.53
1.75	-3.65	0.47	-17.08	1.09	-0.30	-36.44	5.36	-0.56	-31.69
2.00	-2.99	0.37	-17.47	2.52	-0.49	-39.20	7.02	-0.73	-33.59

Device 1 (Apple Mac) and Device 2 (Lenovo Linux), respectively. Moreover, across all end-device configurations, the streaming platform using *ESTR* with a $\tau = 0.50$, achieved a minimum of 18.03% more energy savings compared to basic per-title encoding.

F. THE IMPACT OF VIDEO CODECS

Video streaming providers utilize various video coding standards and their implementations, each with its unique decoding energy demand and compression efficiency levels. Therefore, to evaluate the performance of our approach in diverse streaming environments, we explored three video coding standards: AVC/H.264 [41], HEVC/H.265 [1], and VVC/H.266 [2]. For decoding, FFmpeg libx264 for AVC, FFmpeg libx265 for HEVC, and VVdeC [45] for VVC implementations were used. We compared the proposed approach with basic per-title encoding in terms of compression efficiency and decoding energy consumption for different thresholds τ. The results of this analysis are presented in Table 4.

According to the results, the utilization of HEVC demonstrates the highest energy savings compared to AVC and VVC at the same threshold value when streaming platforms employ the *ESTR* approach. Specifically, within the specified range of τ (0.50 to 2.00), *ESTR* yields a significant reduction in decoder energy consumption, ranging from 21.37% to 39.20% when compared to basic per-title encoding, all while maintaining the same video quality. For AVC, *ESTR* achieves decoding energy savings in the range of 14.46% to 17.47% with the threshold value applying a comparatively lesser impact on energy saving. In the case of VVC, *ESTR* demonstrates decoding energy savings ranging from 17.24% and 33.59%.

In terms of compression efficiency, *ESTR* provides a significant improvement of approximately 5% over basic per-title encoding when $\tau = 0.5$ and AVC is employed for streaming. For HEVC and VVC, known for their efficient compression capabilities, this improvement is 2.44% and 0.70%, respectively. Increasing the threshold value, the energy savings rise at the expense of compression efficiency. At $\tau = 2.00$ for AVC, the improvement decreases to 2.99%, while for HEVC and VVC, basic per-title encoding surpasses *ESTR* in compression efficiency, with *ESTR* requiring 2.52% and 7.02% more bitrate for the same quality. However, at $\tau = 1.00$ for VVC, adopting the proposed approach for streaming empowers clients to save 24.4% in energy during the decoding of the same content compared to basic per-title encoding with only a 0.93% increase in bitrate.

V. CONCLUSION

This paper introduced an energy-aware approach for HTTP adaptive streaming, empowering clients to reduce energy consumption during video playback while maintaining video quality. The proposed approach achieves this by selecting spatial and temporal resolutions of each video that provide video quality approximately identical to the highest quality while minimizing decoding energy consumption at each bitrate. This process is applied to each video content, and the resulting spatial-temporal resolution at each bitrate is combined to create bitrate-resolution pairs, forming the energy-aware bitrate ladder. This ensures that users with varying bandwidths can watch videos with an acceptable quality loss and lower energy consumption. The amount of maximum acceptable quality degradation can be controlled by each service provider through the tunable threshold value (τ) provided by *ESTR*. Experimental results demonstrate that in a video streaming platform using HEVC, regardless of software decoder implementations and operating systems, at a fixed τ of 2, *ESTR* provides a 32.87% to 41.86% reduction in decoding energy consumption. Additionally, it is shown that on a hardware-accelerated client device running the Windows operating system, a 46.37% energy saving is achievable during video playback at the cost of 2.52% compression efficiency decrease. Moreover, using VVdeC for decoding VVC-encoded streams and adjusting the threshold value, *ESTR* can reduce energy consumption by 24.4% with a 0.93% bitrate increase compared to basic per-title encoding.

REFERENCES

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

[2] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.

[3] T. Chachou, W. Hamidouche, S. A. Fezza, and G. Belalem, "Energy consumption and carbon emissions of modern software video encoders," *IEEE Consum. Electron. Mag.*, vol. 1, no. 1, pp. 1–16, Sep. 2023.

[4] S. Afzal, N. Mehran, Z. A. Ourimi, F. Tashtarian, H. Amirpour, R. Prodan, and C. Timmerer, "A survey on energy consumption and environmental impact of video streaming," 2024, *arXiv:2401.09854*.

[5] I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the Internet," *IEEE MultimediaMag.*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[6] L. Zhu, Y. Zhang, Z. Pan, R. Wang, S. Kwong, and Z. Peng, "Binary and multi-class learning based low complexity optimization for HEVC encoding," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 547–561, Sep. 2017.

[7] K.-H. Tai, M.-Y. Hsieh, M.-J. Chen, C.-Y. Chen, and C.-H. Yeh, "A fast HEVC encoding method using depth information of collocated CUs and RD cost characteristics of PU modes," *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 680–692, Dec. 2017.

[8] M. Jamali and S. Coulombe, "Fast HEVC intra mode decision based on RDO cost prediction," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 109–122, Mar. 2019.

[9] E. Çetinkaya, H. Amirpour, M. Ghanbari, and C. Timmerer, "CTU depth decision algorithms for HEVC: A survey," *Signal Process., Image Commun.*, vol. 99, Nov. 2021, Art. no. 116442.

[10] S. Wu, J. Shi, and Z. Chen, "HG-FCN: Hierarchical grid fully convolutional network for fast VVC intra coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5638–5649, Aug. 2022.

[11] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Configurable fast block partitioning for VVC intra coding using light gradient boosting machine," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3947–3960, Jun. 2022.

[12] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1668–1682, Jun. 2020.

[13] S.-H. Park and J.-W. Kang, "Fast affine motion estimation for versatile video coding (VVC) encoding," *IEEE Access*, vol. 7, pp. 158075–158084, 2019.

[14] Y. Fan, J. Chen, H. Sun, J. Katto, and M. Jing, "A fast QTMT partition decision strategy for VVC intra prediction," *IEEE Access*, vol. 8, pp. 107900–107911, 2020.

[15] Z. Zhang, X. Zhao, X. Li, L. Li, Y. Luo, S. Liu, and Z. Li, "Fast DST-VII/DCT-VIII with dual implementation support for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 355–371, Jan. 2021.

[16] S.-H. Park and J.-W. Kang, "Context-based ternary tree decision method in versatile video coding for fast intra coding," *IEEE Access*, vol. 7, pp. 172597–172605, 2019.

[17] H. Li, P. Zhang, B. Jin, and Q. Zhang, "Fast CU decision algorithm based on texture complexity and CNN for VVC," *IEEE Access*, vol. 11, pp. 35808–35817, 2023.

[18] L. Si, W. Zhu, and Q. Zhang, "Fast adaptive CU partition decision algorithm for VVC intra coding," *IEEE Access*, vol. 11, pp. 119766–119778, 2023.

[19] W. Song, C. LI, and Q. Zhang, "Fast CU partitioning algorithm for VVC based on CNN and FSVM," *IEEE Access*, vol. 12, pp. 52150–52160, 2024.

[20] H. Amirpour, V. V Menon, S. Afzal, R. Prodan, and C. Timmerer, "Optimizing video streaming for sustainability and quality: The role of preset selection in per-title encoding," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2023, pp. 1679–1684.

[21] D. Dubey, A. Amritphale, A. Sawhney, D. Dubey, and N. Srivastav, "Analysis of Youtube as a source of information for west Nile virus infection," *Clin. Med. Res.*, vol. 12, nos. 3–4, pp. 129–132, Dec. 2014.

[22] (2023). *Video Marketing Statistics*. [Online]. Available: https://www.wyzowl.com/video-marketing-statistics/

[23] *What We Watched: A Netflix Engagement Report*. Accessed: May 27, 2024. [Online]. Available: https://about.netflix.com/en/news/what-we-watched-a-netflix-engagement-report

[24] R. Yang, M. Xu, L. Jiang, and Z. Wang, "Subjective-quality-optimized complexity control for HEVC decoding," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2016, pp. 1–6.

[25] E. Nogues, E. Raffin, M. Pelcat, and D. Menard, "A modified HEVC decoder for low power decoding," in *Proc. 12th ACM Int. Conf. Comput. Frontiers*, May 2015, pp. 1–6.

[26] E. Nogues, D. Menard, and M. Pelcat, "Algorithmic-level approximate computing applied to energy efficient HEVC decoding," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 1, pp. 5–17, Jan. 2019.

[27] R. Yang, M. Xu, Z. Wang, Y. Duan, and X. Tao, "Saliency-guided complexity control for HEVC decoding," *IEEE Trans. Broadcast.*, vol. 64, no. 4, pp. 865–882, Dec. 2018.

[28] C. Feldmann, F. Jäger, and M. Wien, "Decoder complexity reduction for the scalable extension of HEVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 3729–3733.

[29] T. Mallikarachchi, D. S. Talagala, H. K. Arachchi, and A. Fernando, "A feature based complexity model for decoder complexity optimized HEVC video encoding," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2017, pp. 366–369.

[30] D. Corrua, G. Correa, D. Palomino, and B. Zatt, "OTED: Encoding optimization technique targeting energy-efficient HEVC decoding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–5.

[31] C. Herglotz, A. Heindel, and A. Kaup, "Decoding-energy-rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 171–182, Jan. 2019.

[32] A. Katsenou, X. Wang, D. Schien, and D. Bull, "Comparative study of hardware and software power measurements in video compression," 2023, *arXiv:2312.12150*.

[33] J. De Cock, Z. Li, M. Manohara, and A. Aaron, "Complexity-based consistent-quality encoding in the cloud," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 1484–1488.

[34] H. Amirpour, M. Ghanbari, and C. Timmerer, "DeepStream: Video streaming enhancements using compressed deep neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 1–18, Sep. 2022.

[35] H. Amirpour, C. Timmerer, and M. Ghanbari, "PSTR: Per-title encoding using spatio-temporal resolutions," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2021, pp. 1–6.

[36] (2015). *HTTP Live Streaming (HLS) Authoring Specification for Apple Devices Apple Developer Documentation*. [Online]. Available: https://developer.apple.com/documentation/httplivestreaming/httplivestreaminghlsauthoringspecificationforappledevices

[37] A. Stergiou and R. Poppe, "AdaPool: Exponential adaptive pooling for information-retaining downsampling," *IEEE Trans. Image Process.*, vol. 32, pp. 251–266, 2023.

[38] Z. Li, K. Swanson, C. Bampis, L. Krasula, and A. Aaron. (2020). *Toward a Better Quality Metric for the Video Community*. [Online]. Available: https://netflixtechblog.com/toward-a-better-quality-metric-for-the-video-community-7ed94e752a30

[39] M. Bey Ahmed Khernache, Y. Benmoussa, J. Boukhobza, and D. Menard, "HEVC hardware vs software decoding: An objective energy consumption analysis and comparison," *J. Syst. Archit.*, vol. 115, May 2021, Art. no. 102004.

[40] C. Herglotz, S. Coulombe, C. Vazquez, A. Vakili, A. Kaup, and J.-C. Grenier, "Power modeling for video streaming applications on mobile devices," *IEEE Access*, vol. 8, pp. 70234–70244, 2020.

[41] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[42] B. Taraghi, H. Amirpour, and C. Timmerer, "Multi-codec ultra high definition 8K MPEG-DASH dataset," in *Proc. 13th ACM Multimedia Syst. Conf.*, Jun. 2022, pp. 216–220.

[43] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-curves*, document ITU SG16, 2001.

[44] *HEVC Reference Software HM-18.0*. Accessed: May 27, 2024. [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/HM/-/tags/HM-18.0

[45] A. Wieckowski, G. Hege, C. Bartnik, C. Lehmann, C. Stoffers, B. Bross, and D. Marpe, "Towards a live software decoder implementation for the upcoming versatile video coding (VVC) codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3124–3128.

MOHAMMAD GHASEMPOUR (Student Member, IEEE) received the bachelor's degree from the University of Semnan, Iran, in 2016, and the master's degree in computer architecture from the University of Tehran, in 2018. He is currently pursuing the Ph.D. degree with the University of Klagenfurt. He is also actively contributing to the ATHENA Project. His research interests include compressed-domain video processing drive his specialization in video compression, streaming, and transcoding solutions.

HADI AMIRPOUR (Member, IEEE) received the B.Sc. degree in electrical and biomedical engineering and the M.Sc. degree in electrical engineering from the K. N. Toosi University of Technology and the Ph.D. degree in computer science from the University of Klagenfurt, in 2022. He is currently a Postdoctoral Research Fellow with the Christian Doppler (CD) Laboratory ATHENA, University of Klagenfurt. He was involved in the project EmergIMG, a Portuguese consortium on emerging imaging technologies, funded by Portuguese funding agency and H2020. His research interests include video streaming, image and video compression, quality of experience, emerging 3D imaging technology, and medical image analysis. He has been involved in standardization committees, such as JPEG Pleno and MPEG. He has been co-chairing Qualinet TF7, since 2021. He has been involved in organizing special sessions, workshops, in international conferences, such as IEEE ICME 2023, ACM Multimedia 2022, IEEE EUVIP 2022, and ACM MobiSys 2022. For more information visit the link (https://hadiamirpour.github.io).

CHRISTIAN TIMMERER (Senior Member, IEEE) is currently a Full Professor in computer science with the Institute of Information Technology (ITEC) and the Director of the Christian Doppler (CD) Laboratory ATHENA (https://athena.itec.aau.at/). His research interests include immersive multimedia communication, streaming, adaptation, and quality of experience, where he has co-authored seven patents and more than 300 articles. He was the General Chair of WIAMIS 2008, QoMEX 2013, MMSys 2016, and PV 2018; and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, COST IC1003 QUALINET, ICoSOLE, and SPIRIT. He also participated in ISO/MPEG work for several years, notably in the area of MPEG-21, MPEG-M, MPEG-V, and MPEG-DASH, where he also served as a Standard Editor. In 2013, he cofounded Bitmovin (http://www.bitmovin.com/) to provide professional services around MPEG-DASH, where he holds the position of the Chief Innovation Officer (CIO)—Head of Research and Standardization. For more information visit the link (http://timmerer.com).