Live-PSTR: Live Per-Title Encoding for Ultra HD Adaptive Streaming

Vignesh V Menon[1], Hadi Amirpour[1], Christian Feldmann[2], Adithyan Ilangovan[2], Martin Smole[2], Mohammad Ghanbari[1,3], Christian Timmerer[1]
[1]Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität, Klagenfurt, Austria
[2]Bitmovin, Klagenfurt, Austria
[3]School of Computer Science and Electronic Engineering, University of Essex, UK

Abstract – Current per-title encoding schemes encode the same video content at various bitrates and spatial resolutions to find optimal bitrate-resolution pairs (known as bitrate ladder) for each content in Video on Demand (VoD) applications. But in live streaming applications, a fixed bitrate ladder is used for simplicity to avoid the additional latency to find the optimized bitrate-resolution pairs for every video content. However, an optimized bitrate ladder may result in (i) decreased storage or network resources or/and (ii) increased Quality of Experience (QoE). In this paper, a fast and efficient per-title encoding scheme (Live-PSTR) is proposed tailor-made for live Ultra High Definition (UHD) High Framerate (HFR) streaming. It includes a pre-processing step in which Discrete Cosine Transform (DCT)-energy-based low-complexity spatial and temporal features are used to determine the complexity of each video segment, based on which the optimized encoding resolution and framerate for streaming at every target bitrate is determined. Experimental results show that, on average, Live-PSTR yields bitrate savings of 9.46% and 11.99% to maintain the same PSNR and VMAF scores, respectively compared to the HTTP Live Streaming (HLS) bitrate ladder.

Introduction

Motivation

Video on Demand (VoD) and live streaming are widely adopted in video services, and their applications have drawn massive attention in recent years. Since streaming services constantly adapt the video delivery to the end-user's network conditions and device abilities, HTTP Adaptive Streaming (HAS) continues to expand and has become the de-facto standard for providing video over the Internet. In HAS, each video is encoded at a set of bitrate-resolution pairs, referred to as bitrate ladder. Conventionally, a fixed bitrate ladder, e.g., HTTP Live Streaming (HLS) bitrate ladder [1], is used for all video contents. However, due to the extensive variety in video attributes and network conditions, the "one-size-fits-all" can be optimized per title to improve the Quality of Experience (QoE) [2].

Per-title encoding schemes are based on the fact that resolution performs better than others in a specific region for a given bitrate range, and these regions depend on the video content. As shown in Figure 1, for the Bosphorus sequence, the cross-over bitrate between 1080p and 2160p resolutions happens at approx. b_1= 8.0 Mbps, which means at bitrates lower than b_1, 1080p resolution outperforms 2160p, while at bitrates higher than b_1, 2160p resolution outperforms 1080p. For the ShakeNDry sequence, 1080p remains superior at the entire bitrate range, which means 1080p should be selected for the bitrate ladder for the entire bitrate range. This content-dependency to select the optimal bitrate-resolution pairs is the basis of introducing the per-title encoding [2].

____

This paper is excerpted from the Proceedings of the 2022 NAB Broadcast Engineering and Information Technology (BEIT) Conference, © 2022, National Association of Broadcasters, 1 M Street S.E., Washington, DC 20003 USA.

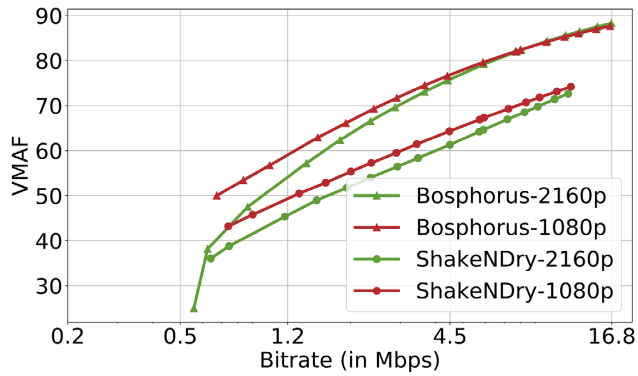


FIGURE 1: RATE-DISTORTION (RD) CURVES OF *BOSPHORUS* AND *SHAKENDRY* SEQUENCES [3] FOR 2160P AND 1080P RESOLUTIONS AT 120FPS.

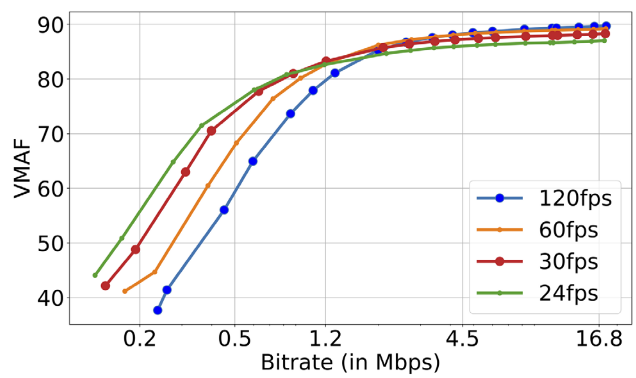


FIGURE 2: RATE-DISTORTION (RD) CURVES OF *HONEYBEE* SEQUENCE [3] AT UHD RESOLUTION AND MULTIPLE FRAMERATES.

In recent years, the spatial resolution of video systems increased to 4K/8K while the temporal resolution mostly remained unchanged [4]. The introduction of HFR videos has stimulated current interest in enriching the viewing experience and visual clarity by increasing the framerate. Several studies show that increasing the framerate reduces temporal artifacts such as flickering, stuttering, and motion blur [5, 6]. However, the main limitation of HFR is a substantial addition in encoding and decoding complexities, which results in increased encoding and decoding times and bitrate overhead compared to low framerate videos [7]. The high bitrate requirement for HFR videos calls for a trade-off between the perceived video quality of *(a)* the compressed video at its original framerate and *(b)* the video compressed at a lower framerate and upscaled in the temporal domain at, *e.g.,* the client [8]. When dropping (some) frames, their bitrate budget is allocated to the remaining frames resulting in a quality improvement of the remaining frames. However, a temporal artifact is added when the video is temporally downscaled. Mackin, et al. [9] showed that higher framerates are advantageous at higher bitrates, particularly in simple sequences with camera motions. However, in terms of encoding efficiency, a lower framerate video performs better at lower bitrates than a higher framerate video. An example UHD (2160p) encoding at various framerates is shown in Figure 2. It reveals that for the *HoneyBee* sequence [3], 24fps is best at 0.2Mbps, 30fps at 1.2Mbps, 120fps at 16.8Mbps. In general, the assumption is that more frames can be dropped in slow-moving videos than fast-moving ones without much noticeable difference in the perceived moving objects' quality [10]. This is the basis for introducing a variable framerate (VFR) coding scheme, depicted in Figure 3. This coding scheme can considerably reduce the bitrate without incurring apparent distortions. Therefore, the bitrate ladder has to be optimized in both spatial and temporal resolutions.

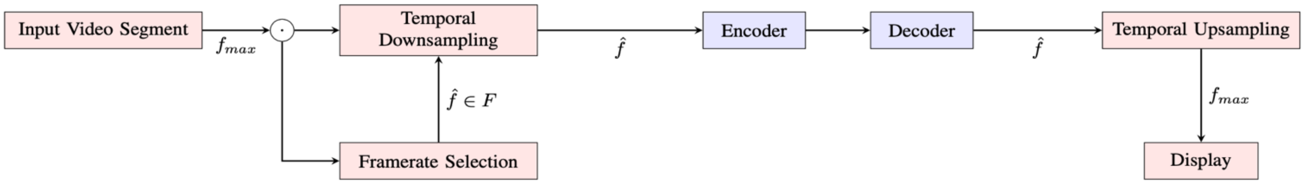


FIGURE 3: BLOCK DIAGRAM OF A VARIABLE FRAME-RATE (VFR) CODING SCHEME [7] IN THE CONTEXT OF VIDEO ENCODING. f_{max} AND $\hat{f}$ DENOTE THE ORIGINAL FRAMERATE OF THE VIDEO AND THE FRAMERATE AT WHICH THE VIDEO IS ENCODED.

.

Though per-title encoding schemes [2, 8, 11] improve the quality of video delivery, determining the convex-hull is computationally very expensive, which made them suitable only for Video on Demand (VoD) streaming applications. Assuming that the streaming service provider supports r resolutions, f framerates and b bitrates, finding the optimized per-title bitrate ladder requires $r \times f \times b$ encodings [8]. However, some methods pre-analyze the video contents to avoid brute force encoding of all bitrate-resolution pairs [12]. Katsenou et al. [13] introduced a content-gnostic technique that uses machine learning to find the bitrate range for each resolution that transcends the other resolutions. Bhat et al. [14, 15] proposed a Random Forest (RF) classifier to determine which encoding resolution is best suited over different quality ranges and study machine learning based adaptive resolution prediction. Yet, these methods yield latency much higher than the accepted latency in live streaming.

Contributions

This paper presents a low-latency online per-title encoding scheme (Live-PSTR) which improves bitrate ladders for live UHD HFR video streaming applications without any perceptible additional latency. This scheme comprises a resolution prediction algorithm that determines every segment's optimized resolution for a given target bitrate. It also includes a low-latency framerate prediction algorithm which predicts optimized framerate for the predicted resolution and target bitrate for every segment. Content-aware features, i.e., Discrete Cosine Transform (DCT)-energy-based low-complexity spatial and temporal features, are extracted to define video segments' characteristics. Live-PSTR is evaluated using the x265 HEVC open-source encoder [16].

Live-PSTR Scheme

The architecture of the proposed Live-PSTR scheme is shown in Figure 4, according to which the optimized resolution ($\hat{r}$) and the optimized framerate ($\hat{f}$) for every target bitrate for each segment is predicted using the spatial and temporal features (E and h [10, 17, 18]) of the video segment and the set of pre-defined resolutions (R), target bitrates (B) and framerates (F) of the bitrate ladder. The encoding is carried out only for the $(\hat{r}, b, \hat{f})$ tuple for every segment, thereby eliminating the need to encode in all resolutions, bitrates, and framerates to find the optimized $(\hat{r}, b, \hat{f})$ tuples. The convex-hull prediction in Live-PSTR is classified into three steps:

(a) feature extraction,
(b) resolution prediction, and
(c) framerate prediction, respectively.

Feature Extraction

In live streaming applications, selecting low-complexity features is critical to ensure low-latency video streaming without disruptions. In a video segment, two features, *i.e.*, the average texture energy (E) and the average gradient of the texture energy (h), are calculated [10,17–19] as a measure of the spatial and temporal complexity.

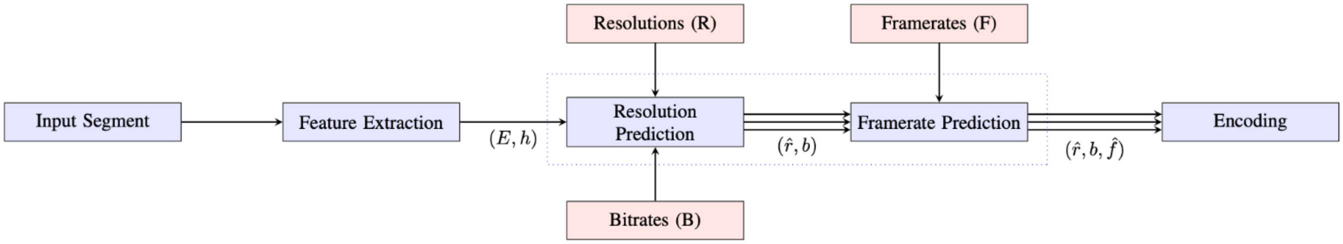


FIGURE 4: *LIVE-PSTR* ARCHITECTURE

Resolution Prediction

Resolution scaling factor (s) is defined as:

$$s = \frac{r}{r_{max}}\, ; r \in R \tag{1}$$

where r_{max} is the maximum resolution in R. In this paper, the optimized resolution scaling factor $\hat{s}$ is determined as a function of the features E and h, and the maximum resolution and framerate. An exponentially decaying (increasing) function is modelled to determine $\hat{s}$ as a function of target bitrate b as shown below:

$$\hat{s}(b) = 1 - s_0 e^{-K_r b} \tag{2}$$

where $s = \frac{r_{min}}{r_{max}}$ (r_{min} denotes the minimum resolution in *R*), K_r determines the rate of decay. The decay rate, K_r, is directly proportional to the temporal characteristics of the segment h and the target bitrate b. At the same time, it is inversely proportional to the spatial characteristics E of the segment. Using this information, K_r is modeled as shown below:

$$K_r = \frac{\Gamma_{MA}(r_{max}, f_{max}) \cdot h}{E} \tag{3}$$

$\Gamma_{MA}(r_{max}, f_{max})$ is the proportion constant named the *Menon Amirpour resolution scaling coefficient*, which depends on the original resolution r_{max} and the original framerate f_{max}. Hence, the final equation for computing the resolution scaling factor is:

$$\hat{s}(b) = 1 - s_0 e^{-\frac{\Gamma_{MA}(r_{max}, f_{max}) \cdot h \cdot b}{E}} \tag{4}$$

To determine $\Gamma_{MA}(r_{max}, f_{max})$, (4) is modified to the form of Y = mX + c as shown below to use linear regression.

$$log\left(\frac{1}{1-\hat{s}}\right) = \frac{\Gamma_{MA}(r_{max}, f_{max}) \cdot h \cdot b}{E} - log(s_0) \tag{5}$$

After determining the optimized resolution scaling factor $\hat{s}$ for any target bitrate, the optimized resolution is determined as:

$$\hat{r} = \hat{s} \cdot r_{max} \tag{6}$$

Framerate Prediction

In this paper, the optimized framerate $\hat{f}$ is determined as a function of the features E and h, the resolution $\hat{r}$ and the maximum framerate f_{max}. An exponential decay (increasing) function is modeled to determine $\hat{f}$ for each segment as shown below:

$$\hat{f} = f_{max}\left(1 - d_0 e^{-K_f \cdot b}\right) \quad (7)$$

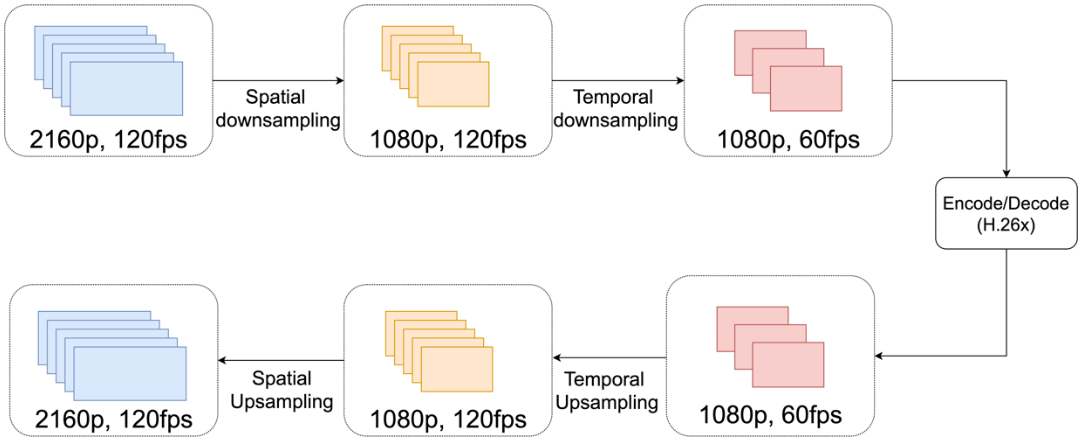


FIGURE 5: THE ORIGINAL VIDEO (2160P, 120FPS) IS DOWNSCALED TO (1080P, 60FPS) AND THEN ENCODED. THE DECODED VIDEO IS UPSCALED BACK TO (2160P, 120FPS) TO MATCH THE ORIGINAL RESOLUTION AND FRAMERATE TO DISPLAY. PLEASE NOTE THAT THE OBJECTIVE METRICS ARE CALCULATED USING THE UPSCALED VIDEO.

where $d_0 = \frac{f_{max} - f_{min}}{f_{max}}$, K_f determines the rate of decay. The decay rate, K_f, is directly proportional to the temporal characteristics of the segment h and the target bitrate b. At the same time, it is inversely proportional to the spatial characteristics E. K_f is thus modeled as:

$$K_f = \frac{\beta_{MA}(\hat{r}, f_{max}) \cdot h}{E} \quad (8)$$

β_{MA} is the proportion constant named the *Menon Amirpour framerate coefficient*, which depends on the video resolution ($\hat{r}$) and the original framerate (f_{max}). Hence, the final equation for computing $\hat{f}$ is:

$$\hat{f} = f_{max}\left(1 - d_0 e^{-\frac{\beta_{MA}(\hat{r}, f_{max}) \cdot h \cdot b}{E}}\right) \quad (9)$$

To determine $\beta_{MA}(\hat{r}, f_{max})$, (9) is modified to the form of Y = mX + c as shown below to use linear regression.

$$log\left(\frac{f_{max}}{f_{max} - \hat{f}}\right) = \frac{\beta_{MA}(\hat{r}, f_{max}) \cdot h \cdot b}{E} - log(d_0) \quad (10)$$

After determining $\beta_{MA}(\hat{r}, f_{max})$, the model predicts $\hat{f}$ for every segment for any given target bitrate and resolution.

On the encoding side, frames are dropped based on the optimized framerate. The video segment is decoded and upscaled in the temporal and spatial domain such that it is displayed at the original resolution (r_{max}) and framerate (f_{max}) as shown in Figure 5. This is because many of the displays do not support VFR coding [7].

Evaluation

Test Methodology

The test sequences are encoded using x265 v3.5 [16] with the ultrafast preset and Video Buffering Verifier (VBV) rate control mode. All experiments are run on a dual-processor server with Intel Xeon Gold 5218R (80 cores, frequency at 2.10GHz). The resolutions specified in Apple HLS authoring specifications [1] are considered in the evaluation, *i.e.*, R = {360, 432, 540, 720, 1080, 1440, 2160}. The lower resolution sources are generated from the original video source by applying bi-cubic scaling using FFmpeg [20]. The set of framerates considered in this paper is F = {20, 24, 30, 45, 60, 90, 120}. The resulting quality in PSNR and VMAF [21], and the achieved bitrate are compared for each test sequence. Since the content is assumed to be displayed on the highest resolution, *i.e.*, 2160p, the encoded content is scaled (bi-cubic) to 2160p resolution, and VMAF and PSNR are calculated. Bjøntegaard delta rates [22] BDR_P and BDR_V refer to the average increase in bitrate of the representations compared with that of the fixed bitrate ladder encoding to maintain the same PSNR and VMAF, respectively. A negative *BDR* indicates a gain in coding efficiency of *Live-PSTR* compared to the fixed bitrate ladder encoding. ΔT represents the cumulative difference in time taken for encoding representations selected using *Live-PSTR* compared to the brute-force approach which encodes in all resolutions, bitrates, and framerates to choose the optimized representations.

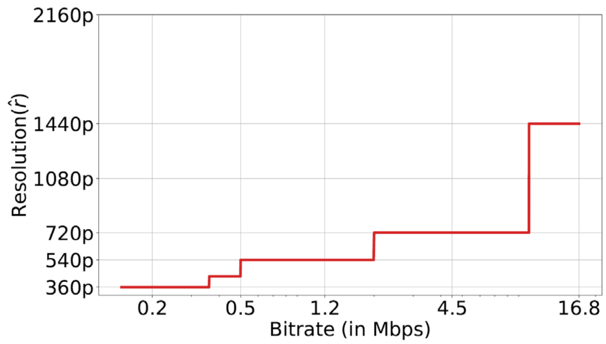
Video Properties				Results				
Dataset	Video	E	h	$\|\| s_G - \hat{s} \|\|_2$	$\left\|\left\| \frac{f_G - \hat{f}}{f_{max}} \right\|\right\|_2$	BDR_P	BDR_V	ΔT
JVET [23]	CatRobot	66.99	12.86	0.01	0.02	-8.95%	-10.43%	-77.62%
	DaylightRoad2	54.78	20.35	0.02	0.03	-9.35%	-13.52%	-76.97%
	FoodMarket4	60.61	22.67	0.02	0.02	-8.74%	-12.11%	-78.01%
UVG [3]	Beauty	37.72	19.34	0.02	0.01	-12.14%	-12.29%	-76.89%
	Bosphorus	23.35	6.33	0.03	0.01	-10.47%	-11.78%	-77.71%
	HoneyBee	40.71	5.29	0.01	0.01	-13.51%	-14.96%	-79.76%
	Jockey	33.79	53.08	0.02	0.01	-7.36%	-11.95%	-76.65%
	Lips	36.96	10.01	0.02	0.01	-6.93%	-9.67%	-76.16%
	ReadySteadyGo	55.40	46.06	0.01	0.03	-5.76%	-10.02%	-76.74%
	ShakeNDry	40.83	10.13	0.03	0.03	-11.31%	-16.02%	-77.67%
	YachtRide	37.72	19.34	0.02	0.03	-6.56%	-9.14%	-77.34%
Average				**0.02**	**0.02**	**-9.46%**	**-11.99%**	**-77.72%**

TABLE 1: RESULTS OF *LIVE-PSTR* AGAINST FIXED BITRATE LADDER APPROACH.

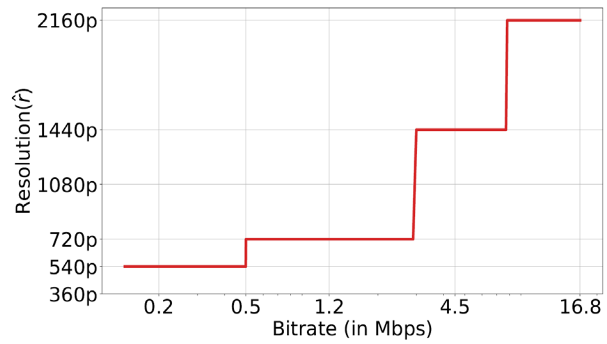
Experimental Results

Table 1 summarizes the results of *Live-PSTR*-based encodings using UHD (2160p) resolution and high framerate (60 and 120fps) test sequences from JVET [23] and UVG [3] datasets. The spatio-temporal characteristics of the video are measured using the E and h features [18], which is later used to predict the optimized resolution and framerate for the target encoding bitrates. The optimized resolution scaling factor s_G is determined manually for each target bitrate for every segment which is used as the ground truth. The accuracy of the resolution prediction algorithm is determined by the L_2 norm of the s_G and the selected resolution scaling factor ($\hat{s}$), *i.e.*, $\|| s_G - \hat{s} \||_2$. The average $\|| s_G - \hat{s} \||_2$ is observed as 0.02. Similarly, the optimized framerate f_G is determined manually for each resolution and target bitrate for every segment which is used as the ground truth. The accuracy of the framerate prediction algorithm is determined by the normalized L_2 norm of f_G. The average speed of the convex-hull prediction for each segment is 370 fps. Hence, *Live-PSTR* introduces no additional latency in streaming. On average, BDR_P and BDR_V of *Live-PSTR* compared to the fixed bitrate ladder approach are -9.46% and -11.99%, respectively. Compared to the brute-force approach to select the optimized resolution and framerate, *Live-PSTR* yields an average time saving of 77.72%.

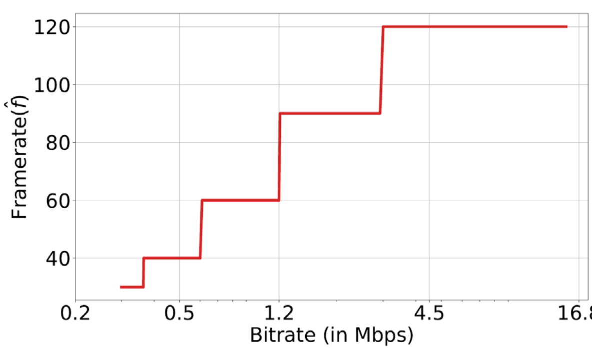
The improvement in compression efficiency of the encodings of *Beauty* and *HoneyBee* sequences, the predicted optimized resolution and framerates using *Live-PSTR* scheme is depicted in Figure 6.



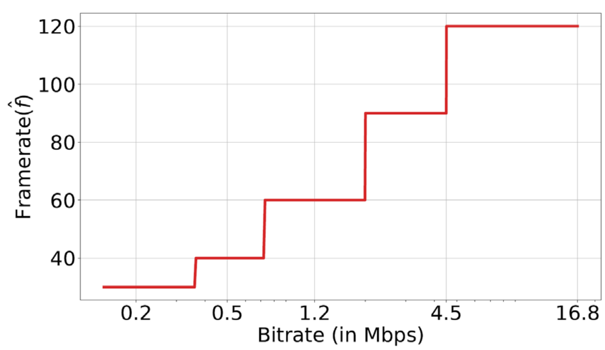
(A) OPTIMIZED RESOLUTION



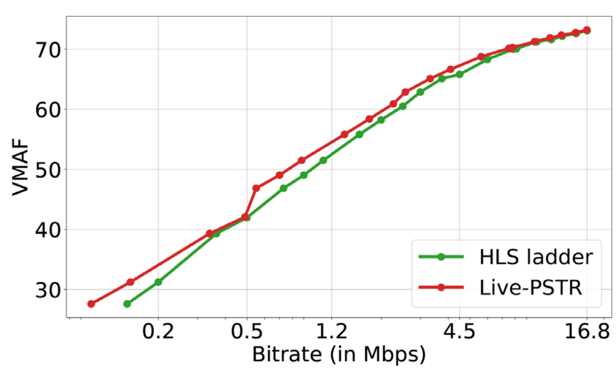
(B) OPTIMIZED RESOLUTION



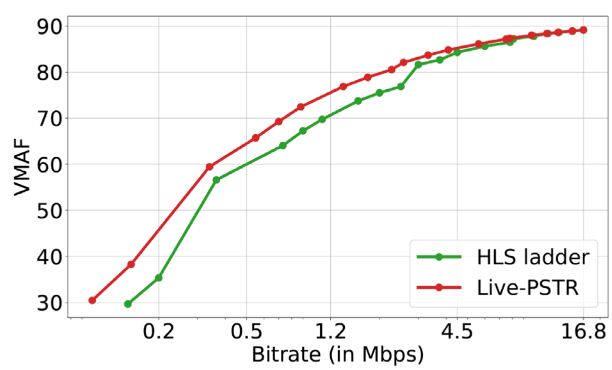
(C) OPTIMIZED FRAMERATE



(D) OPTIMIZED FRAMERATE



(E) RD CURVES



(F) RD CURVES

FIGURE 6: THE PREDICTED OPTIMIZED RESOLUTION AND OPTIMIZED FRAMERATE, AND COMPARISON OF RD CURVES OF *BEAUTY* (A,C,E) AND *HONEYBEE* (B,D,F) SEQUENCES USING THE HLS BITRATE LADDER AND *LIVE-PSTR*.

Conclusions

This paper proposed an online per-title encoding scheme with spatial and temporal resolutions (*Live-PSTR*), especially for live streaming of UHD HFR videos. *Live-PSTR* includes a convex-hull prediction algorithm that predicts the optimized resolution and framerate for a given target bitrate for each segment, which helps improve the quality of live streaming. DCT-energy-based features are calculated to determine segments' spatial and temporal complexities, which is fast and effective. It is observed that live streaming using *Live-PSTR* requires 9.46% fewer bits to maintain the same PSNR and 11.99% fewer bits to maintain the same VMAF compared to the HLS bitrate ladder encoding using x265 HEVC encoder.

Acknowledgment

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/.

References

[1] Apple Inc., "HLS Authoring Specification for Apple Devices," [Online]. Available: https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices.

[2] J. De Cock, Z. Li, M. Manohara, and A. Aaron, "Complexity-based consistent-quality encoding in the cloud," in 2016 IEEE International Conference on Image Processing (ICIP), 2016.

[3] A. Mercat, M. Viitanen, and J. Vanne, UVG Dataset: 50/120fps 4K Sequences for Video Codec Analysis and Development. New York, NY, USA: Association for Computing Machinery, 2020, pp. 297–302. [Online]. Available: https://doi.org/10.1145/3339825.3394937

[4] C. Ge, N. Wang, G. Foster, and M. Wilson, "Toward QoE-Assured 4K Video-on-Demand Delivery Through Mobile Edge Virtualization With Adaptive Prefetching," IEEE Transactions on Multimedia, vol. 19, no. 10, pp. 2222–2237, 2017.

[5] P. C. Madhusudana, X. Yu, N. Birkbeck, Y. Wang, B. Adsumilli, and A. C. Bovik, "Subjective and Objec tive Quality Assessment of High Frame Rate Videos," IEEE Access, vol. 9, pp. 108 069–108 082, 2021.

[6] J. Y. C. Chen and J. E. Thropp, "Review of Low Frame Rate Effects on Human Performance," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 37, no. 6, 2007.

[7] G. Herrou, C. Bonnineau, W. Hamidouche, P. Dumenil, J. Fournier, and L. Morin, "Quality- driven Variable Frame-Rate for Green Video Coding in Broadcast Applications," IEEE Transactions on Circuits and Systems for Video Technology, pp. 1–1, 2020.

[8] H. Amirpour, C. Timmerer, and M. Ghanbari, "PSTR: Per-Title Encoding Using Spatio-Temporal Resolutions," in 2021 IEEE International Conference on Multimedia and Expo (ICME), 2021, pp. 1–6.

[9] A. Mackin, F. Zhang, M. A. Papadopoulos, and D. Bull, "Investigating the impact of high frame rates on video compression," in 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 295–299.

[10] V. V. Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, “CODA: Content-aware Frame Dropping Algorithm for High Frame-rate Video Streaming,” in 2022 Data Compression Conference (DCC), 2022, pp. 475-475.

[11] V. P. K. Malladi, C. Timmerer, and H. Hellwagner, “MiPSO: Multi-Period Per-Scene Optimization For HTTP Adaptive Streaming,” in 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1–6.

[12] Bitmovin, “White paper: Per title encoding,” 2018. [Online]. Available: https://bitmovin.com/whitepapers/Bitmovin-Per-Title.pdf

[13] A. V. Katsenou, J. Sole, and D. R. Bull, “Content-gnostic bitrate ladder prediction for adaptive video streaming,” in 2019 Picture Coding Symposium (PCS), 2019.

[14] M. Bhat, J.-M. Thiesse, and P. L. Callet, “Combining Video Quality Metrics To Select Perceptually Accurate Resolution In A Wide Quality Range: A Case Study,” in 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2164–2168.

[15] M. Bhat, J.-M. Thiesse, and P. L. Callet, “A Case Study of Machine Learning Classifiers for Real-Time Adaptive Resolution Prediction in Video Coding,” in 2020 IEEE International Conference on Multimedia and Expo (ICME), 2020, pp. 1–6.

[16] MulticoreWare Inc., “x265 HEVC Encoder/H.265 Video Codec.” [Online]. Available: http://x265.org/

[17] V. V. Menon, H. Amirpour, M. Ghanbari, and C. Timmerer, “Efficient Content-Adaptive Feature-Based Shot Detection for HTTP Adaptive Streaming,” in 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 2174–2178.

[18] V. V. Menon, C. Feldmann, H. Amirpour, M. Ghanbari, and C. Timmerer, “VCA,” Feb. 2022. [Online]. Available: https://github.com/cd-athena/VCA

[19] M. King, Z. Tauber, and Z.-N. Li, “A New Energy Function for Segmentation and Compression,” in 2007 IEEE International Conference on Multimedia and Expo, 2007, pp. 1647–1650.

[20] FFmpeg, “Ffmpeg documentation.” [Online]. Available: https://ffmpeg.org/ffmpeg.html

[21] Z. Li, C. Bampis, J. Novak, A. Aaron, K. Swanson, A. Moorthy, and J. Cock, “VMAF: The journey continues,” Netflix Technology Blog, vol. 25, 2018.

[22] G. Bjontegaard, “Calculation of average PSNR differences between RD-curves,” VCEG-M33, 2001.

[23] J. Boyce, K. Suehring, X. Li, and V. Seregin, “JVET- J1010: JVET common test conditions and software reference configurations,” Jul. 2018.