Two-Pass Encoding for Live Video Streaming[1]

Mohammad Ghasempour, Hadi Amirpour, and Christian Timmerer
Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität
Klagenfurt, Austria

Abstract – Live streaming has become increasingly important in our daily lives due to the growing demand for real-time content consumption. Traditional live video streaming typically relies on single-pass encoding due to its low latency. However, it lacks video content analysis, often resulting in inefficient compression and quality fluctuations during playback. *Constant Rate Factor* (CRF) encoding, a type of single-pass method, offers more consistent quality but suffers from unpredictable output bitrate, complicating bandwidth management. In contrast, multi-pass encoding improves compression efficiency through multiple passes. However, its added latency makes it unsuitable for live streaming. In this paper, we propose *OTPS*, an online two-pass encoding scheme that overcomes these limitations by employing fast feature extraction on a downscaled video representation and a gradient-boosting regression model to predict the optimal CRF for encoding. This approach provides consistent quality and efficient encoding while avoiding the latency introduced by traditional multi-pass techniques. Experimental results show that *OTPS* offers 3.7% higher compression efficiency than single-pass encoding and achieves up to 28.1% faster encoding than multi-pass modes. Compared to single-pass encoding, encoded videos using *OTPS* exhibit 5% less deviation from the target bitrate while delivering notably more consistent quality.

Introduction

In recent years, the demand for high-quality live streaming has increased due to the popularity of online platforms that offer real-time content, such as gaming, sports, and live events. To meet the quality expectations of the audiences while minimizing bandwidth, processing costs, and latency, efficient encoding strategies are essential. In video streaming or specifically in *HTTP Adaptive Streaming* (HAS) [1], a video is encoded into a set of bitrate and resolution pairs, referred to as bitrate ladder. This allows viewers to receive the most suitable quality representation based on their device capabilities and current network conditions. However, for encoders, the primary challenge lies in achieving the target bitrate for each video while consistently maintaining high visual quality, a task complicated by the need to adapt to varying content and network dynamics.

The rate controller of the video encoder is responsible for dynamically adjusting the encoding parameters to manage the trade-off between bitrate and visual quality. *Constant Bitrate* (CBR) encoding is a widely used approach in live streaming, where maintaining a fixed bitrate throughout the video is crucial to ensure predictable and efficient bandwidth usage. CBR encoding continuously adjusts the encoding parameters to adhere to a specified bitrate, regardless of the complexity of individual frames. This makes it advantageous for live streaming, as it simplifies network management,

[1] 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/

minimizes the risk of buffer underruns, and ensures low latency, which is essential for real-time applications like live sports broadcasts and online game streams. However, since the bitrate remains constant, high-complex frames may suffer from quality degradation due to insufficient bit allocation, while low-complex frames may be over-compressed, leading to inefficient bit usage. This results in a trade-off between achieving consistent video quality and maintaining a steady bitrate, yet CBR remains popular for its predictable nature and suitability for low-latency streaming scenarios.

Constant Rate Factor (CRF) encoding is a quality-based rate control method commonly used in video compression to maintain consistent visual quality across frames. Instead of targeting a specific bitrate, CRF provides a target quality, making it popular for scenarios where quality consistency is more important than precise control over the output bitrate. However, the main drawback of CRF encoding is the unpredictability of the final file size or bitrate, which can pose challenges for applications where bandwidth needs to be carefully managed, such as live streaming. Despite this, CRF is highly effective for offline encoding and scenarios where quality takes precedence over bitrate consistency, as it delivers a more visually pleasing experience compared to traditional single-pass or fixed/constant bitrate methods.

Multi-pass encoding is an advanced method that seeks to combine the advantages of constant bitrate and constant quality encodings by optimizing bitrate allocation to achieve consistent visual quality while adhering to a target bitrate. Unlike single-pass methods, multi-pass encoding processes the video multiple times. In the first pass, the encoder thoroughly analyzes the entire video to collect detailed information about frame complexity and motion characteristics. This data enables the encoder to make informed decisions during subsequent passes, allocating more bits to complex frames that require higher quality and fewer bits to simpler frames. By doing so, multi-pass encoding efficiently balances the trade-off between bitrate and quality, providing better compression efficiency and more consistent visual quality than single-pass CBR method. However, the primary drawback of multi-pass encoding is the increased latency introduced by the additional encoding passes. This added processing time makes it less suitable for live streaming applications where low latency is crucial, but it is highly effective for offline encoding scenarios where achieving optimal quality and bitrate efficiency is more important than encoding speed.

Numerous studies have focused on optimizing video encoding, particularly for *Video on Demand* (VOD) applications, frequently employing multi-pass encoding techniques. For instance, Que et al. [2] introduced a two-pass encoding approach where the first pass employs CBR encoding to collect key encoding statistics from the input video. These statistics are then used to determine the optimal QP for the second pass. Wang et al. [3] proposed a two-pass control algorithm that distributes the bit budget across three different levels: *Group of Pictures* (GOP), frame, and block. At the frame level, the sum of absolute transformed differences is used to enable adaptive QP selection based on the allocated coding bits. Malladi et al. [4] developed a model based on the *Rate-Distortion* (RD) characteristics of each frame, using data from the first pass to identify the best QP for second-pass encoding. Cao et al. [5] introduced a two-pass rate control scheme designed to reduce video quality fluctuations by minimizing distortion variance between GOPs. In first-pass encoding, a fixed-QP configuration is used, and a set of candidate QPs is specified. For each candidate, the corresponding rate and distortion are computed, which enables the selection of the optimal quantizer for the second pass. However, none of these schemes are suitable for live streaming.

In the past decade, machine learning techniques have made video streaming more efficient and have been successfully integrated into video streaming applications. In this study, we aim to take advantage of a lightweight machine learning technique to enable CRF encoding for live video streaming and to achieve the performance of multi-pass in a time close to single-pass encoding. To achieve this, fast feature extraction methods are used to extract information from the video content that can be used to predict the optimal CRF value based on a predefined target bitrate. In this way, the output video is encoded efficiently, minimizing quality fluctuations while closely adhering to the target bitrate.

Online Two-Pass Encoding Scheme

The proposed online two-pass encoding scheme aims to achieve efficient real-time encoding for live video streaming by predicting an optimized CRF for a given target bitrate. This section details the methodology used to predict the optimized CRF.

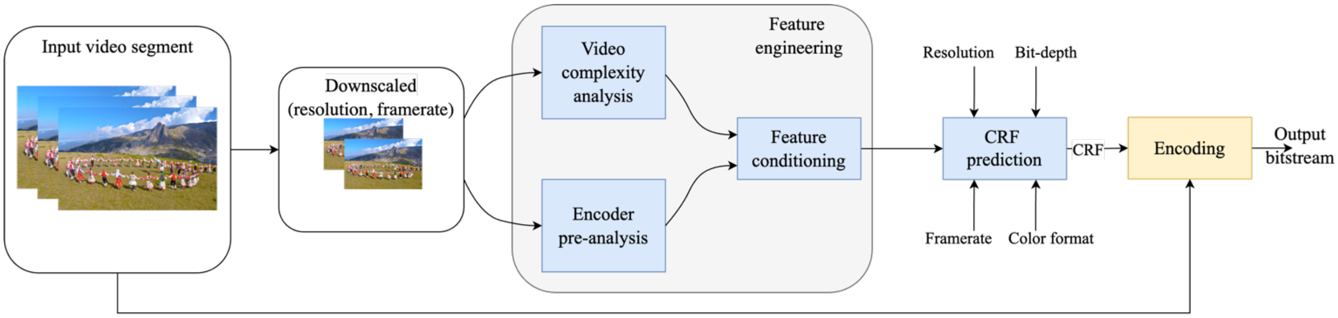


FIGURE 1. THE ARCHITECTURE OF *OTPS*

OTPS Architecture

The architecture of *OTPS* is shown in Figure 1. The process begins with extracting a set of features from a downscaled version of the input video segment. The reason to utilize downscaled video representation for feature extraction is to minimize the computational complexity of the process. Using these features, along with the original video characteristics such as video resolution, framerate, bit-depth, and color format, *OTPS* predicts an optimized CRF to guide the encoding process. The following is the explanation of each block in the *OTPS* workflow.

Feature engineering: For live streaming applications, selecting video segment features that can be extracted with low computational overhead is essential. Two feature extraction approaches are employed to capture video complexity and other relevant information for precise CRF prediction. To extract video complexity, the *Enhanced Video Complexity Analyzer* (EVCA) [6] is used, providing various complexity metrics, such as spatial and temporal complexity metrics, with minimal computational load. To further improve prediction accuracy, a pre-analysis step was added to extract compressed domain features. This step uses the libx264 ultrafast preset to encode the spatially and temporally downscaled version of the video, meeting real-time constraints, please refer to Appendix A1.

For refining the feature set, a feature conditioning block selects and normalizes the most relevant features using three feature selection algorithms: correlation-based [7], variance threshold-based, and importance-based. The correlation-based approach removes features with high inter-correlation to reduce redundancy, while the variance threshold-based eliminates features with minimal variance, assuming they contribute little to prediction. Finally, the importance-based method uses the *Least Absolute Shrinkage and Selection Operator* (Lasso) [8], a lightweight linear model, to identify the most influential features for output prediction. This selection process is performed once during training to define the final input feature set, which is listed in Table 1. The values of these features are normalized using a standard scaler to have a mean $\mu = 0$ and a standard deviation $\sigma = 1$.

CRF Prediction: The optimal CRF value that achieves the target bitrate is influenced by multiple factors beyond just the content of the video. These include the video's original resolution, framerate, bit-depth, and color format. Consequently, alongside video content features, these parameters are included as model inputs. To ensure precise prediction, a dedicated model is trained for each target bitrate. Since CRF encoding allows for floating-point values, the prediction process is structured as a regression problem to minimize the output error. To this end, a gradient-boosting regression model is utilized. This ensemble technique is known for its high accuracy and the ability to capture complex data patterns. The predicted CRF value is then applied to encode videos, enabling efficient, constant-quality encoding that meets the target bitrate requirements.

Feature	Explanation	Feature	Explanation
qpP	Average QP values of P-frames	i8d	Percentage of 8×8 blocks using DC intra prediction
sizeP	Size of P-frames in Bytes	sizeI	Size of I-frames in Bytes
mbPI16	Percentage of intra 16×16 macroblocks in P-frames	yI	Percentage of coded blocks for luma intra blocks
mbPP16	Percentage of inter 16×16 macroblocks in P-frames	SI	Average spatial information
i16v	Percentage of 16×16 blocks using vertical intra prediction	TI	Average temporal information
i16d	Percentage of 16×16 blocks using DC intra prediction	TC	Average temporal complexity (EVCA method)
uvDCP	Percentage of coded blocks for DC chroma inter blocks	E	Spatial complexity (VCA method); Aggregated using average, median, last, and maximum functions
mbskip	Number of skipped macroblocks	h	Temporal complexity (VCA method); Aggregated using average, median, last, and maximum functions
i8p	Percentage of 8×8 blocks using planar intra prediction	l	Brightness (VCA method); Aggregated using average, minimum, maximum, first, and last functions

TABLE 1. THE LIST OF SELECTED FEATURES WITH THEIR EXPLANATION.

Training Process

To train the gradient boosting model, the first step is to encode videos in the training dataset at various CRF values within a specified range from C_{min} to C_{max}. This range varies by codec; for instance, with *Advanced Video Coding* (AVC) [9], the CRF values span from 0 to 51. After encoding, the “desired CRF,” is calculated, which is the CRF value that would produce an output video bitrate matching the target bitrate. Since the target bitrate often does not precisely match the bitrates of the encoded outputs, a linear interpolation is used to estimate the desired CRF. Linear interpolation is effective for cases where the CRF difference between two encodings is small, yielding a close approximation of the desired CRF. Specifically, to determine the desired CRF, two records in the dataset that target bitrate falls in between them should be identified. The bitrate and CRF of these records are denoted as B_l, B_h, C_l, and C_h. B_l and C_l are for the lower bitrate index, B_h and C_h are for the higher bitrate index. The desired CRF is then calculated as in (1), (2), and (3):

$$S = \frac{C_h - C_l}{B_h - B_l} \tag{1}$$

$$I = C_l - (S \times B_l) \tag{2}$$

$$C_{desired} = \left(S \times B_{target}\right) + I \tag{3}$$

where S and l represent the slope and intercept, respectively, simplifying the formula, and B_{target} is the target bitrate.

Experimental Setup

To assess the performance of *OTPS*, we selected videos from the Inter4K dataset [10], originally in 3840×2160 resolution at 60 fps. It should be noted that the videos contain scene-cuts. The evaluation spans four key areas: compression efficiency, encoding time, bitrate difference, and quality consistency. For compression efficiency, we used the *Bjøntegaard Delta Rate* (BD-Rate) metric [11], which quantifies bitrate savings at a consistent quality level. Results were validated with *Video Multi-method Assessment Fusion* (VMAF) [12] quality metric. Furthermore, *OTPS*'s time consumption was analyzed in comparison to existing encoding techniques as in (4):

$$T_{saving} = \frac{T_{ref} - T_{OTPS}}{T_{ref}} \times 100 \quad (4)$$

where T_{ref} is the time consumed by the existing single-pass and multi-pass encoding modes.

The bitrate difference is also calculated using the target bitrate as its reference for each video sequence as in (5):

$$B_{deviation} = \frac{|B_{target} - B_{OTPS}|}{B_{target}} \times 100 \quad (5)$$

where B_{target} and B_{OTPS} are the target bitrate and the output bitrate of encoded video, respectively.

Finally, the quality deviation is defined as the difference between the maximum and minimum provided quality for each video sequence as in (6):

$$Q_{deviation} = \frac{max(f_1, f_2, \ldots, f_n) - min(f_1, f_2, \ldots, f_n)}{max(f_1, f_2, \ldots, f_n)} \times 100 \quad (6)$$

where n is the number of frames and f_n specifies the VMAF quality of the n-th frame.

Dataset

To train the model, a diverse dataset of videos encoded with a range of CRF values is required. For this purpose, all videos from the Inter4K dataset are encoded at CRF values ranging from 16 to 51, capturing a broad spectrum of bitrate and quality variations. Additionally, the above-mentioned features are extracted from each video to provide comprehensive data points. Figure 2 shows the spatial and temporal complexity of the videos in this dataset. This extensive dataset, which has 35000 records, varied content, and a CRF range, is well-suited for tasks such as rate and quality control and cost optimization [13].

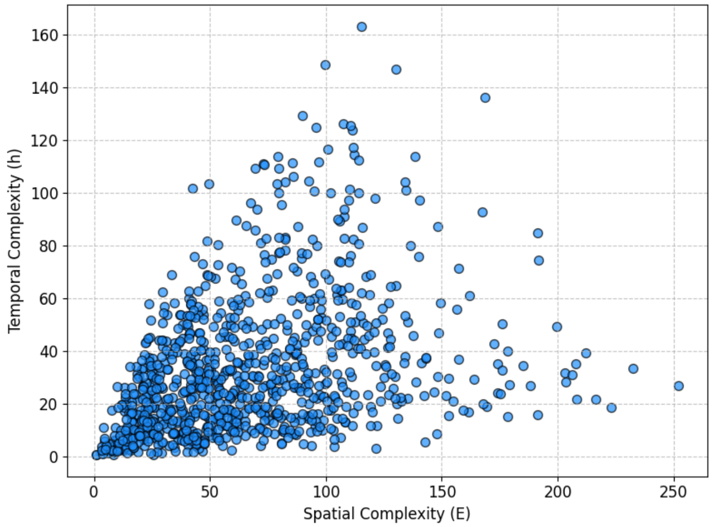


FIGURE 2. DISTRIBUTION OF SPATIAL (E) AND TEMPORAL (H) COMPLEXITIES USING EVCA FOR THE VIDEOS USED IN THIS STUDY.

Encoder Configuration

For the evaluation, we selected the cloud-based Bitmovin encoder because it provides three encoding modes: single-pass, two-pass, and three-pass. This capability allows for a comprehensive comparison with *OTPS*, ensuring a thorough evaluation of the results. These encoding modes operate by taking the target bitrate as input, while *OTPS* employs CRF encoding. In all cases, live encoding configurations are applied as shown in Table 2. The evaluation is conducted at four target bitrates: 2 Mbps, 5 Mbps, 8 Mbps, and 12 Mbps.

Parameter	Value
adaptiveQuantizationMode	VARIANCE
adaptiveQuantizationStrength	1
adaptiveSpatialTransform	TRUE
bAdaptiveStrategy	FAST
bPyramid	NORMAL
bframes	3
cabac	TRUE
encodingMode	SINGLE_PASS
fastSkipDetectionPFrames	TRUE
macroblockTreeRatecontrol	TRUE
mixedReferences	FALSE
motionEstimationMethod	HEX
mvPredictionMode	SPATIAL
mvSearchRangeMax	16
nalHrd	NONE
partitions	"I4X4,I8X8,P8X8,B8X8"
profile	HIGH
rcLookahead	20
refFrames	2
sceneCutThreshold	0
subMe	QPEL4
trellis	ENABLED_FINAL_MB

weightedPredictionBFrames	TRUE
weightedPredictionPFrames	SIMPLE

TABLE 2. LIVE ENCODING CONFIGURATION.

Experimental Results

In this section, we first evaluate the prediction accuracy of the gradient boosting regression model used in *OTPS* by applying three different metrics: *Mean Absolute Error* (MAE), *Mean Squared Error* (MSE), and the R^2 score. Next, we examine the quality deviation of each encoding mode compared to the consistency achieved by *OTPS*. Finally, we compare the compression efficiency and time savings of the proposed *OTPS* against the three encoding modes discussed.

Prediction Performance

To evaluate the gradient boosting model's performance, the video sequences in the dataset are divided into training and testing sets. Figure 3 shows the distribution of the training and testing video sequences based on spatial and temporal complexity metrics. A total of 80% of the video sequences are used for training, while the remaining 20% are used for testing to ensure that the test video sequence is not seen in the training phase. Four separate models are trained, each tailored to a specific target bitrate.

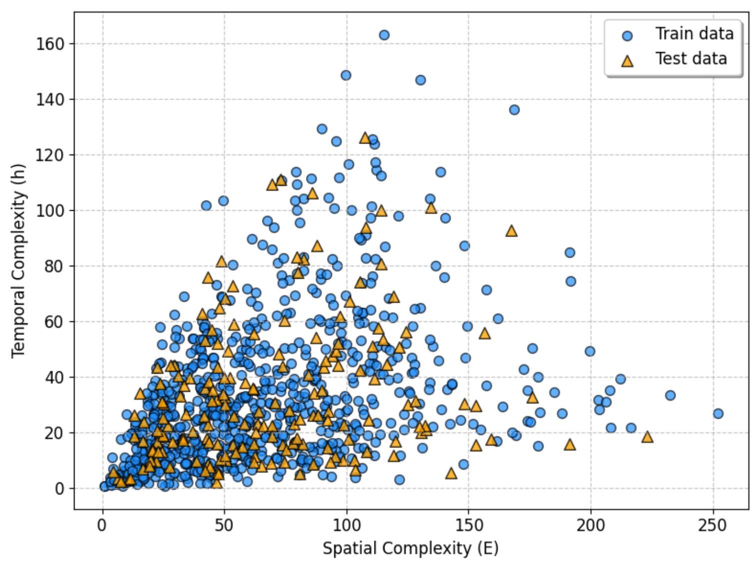


FIGURE 3. THE DISTRIBUTION OF TRAINING AND TESTING VIDEO SEQUENCES.

To assess the impact of each input feature on CRF prediction, Figure 4 displays the correlation values of individual features with the target CRF. Features such as *qpP*, *sizeP*, and *h_avg* show the highest positive correlations, indicating that temporal complexity and inter-frame information significantly impact the CRF output. In contrast, the number of skipped macroblocks (*mbskip*) and *i16v* show the strongest negative correlations, suggesting that these features inversely influence CRF prediction.

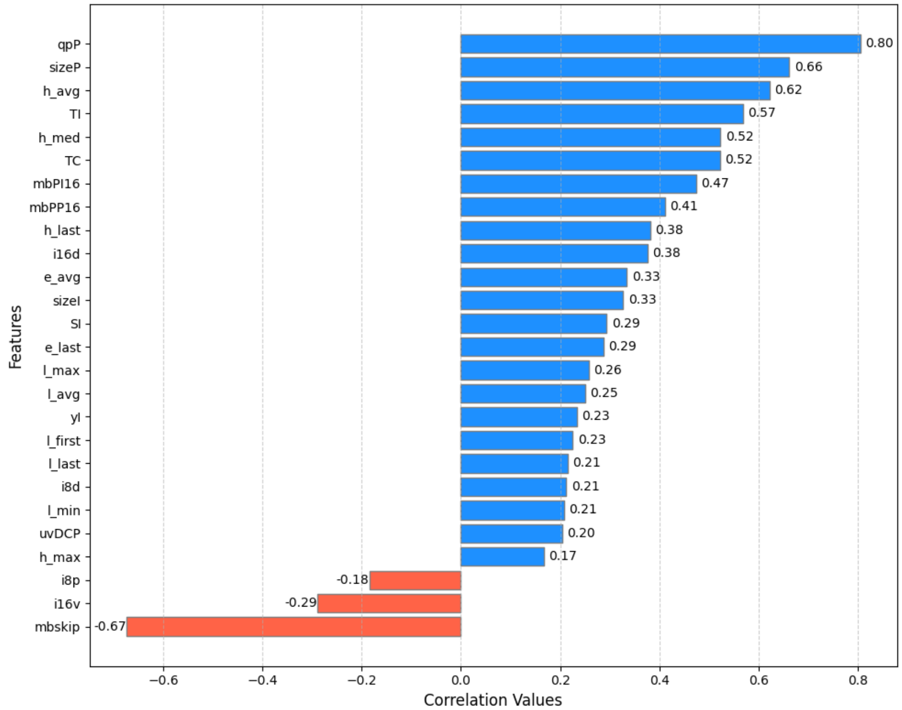


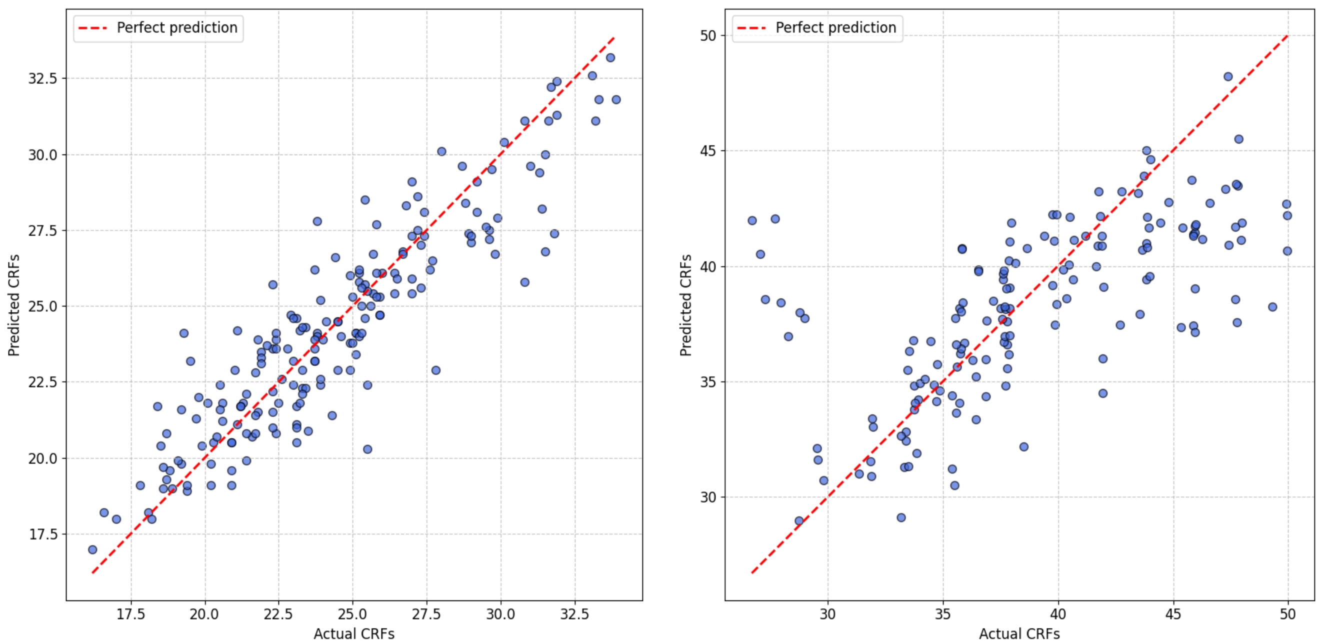
FIGURE 4. THE CORRELATION VALUES OF INPUT FEATURES TO THE TARGET VALUE (CRF).

The prediction performance results for each target bitrate are reported in Table 3. For target bitrates other than 2 Mbps, the MAE remains below or equal to 0.82 for the training set and 1.4 for the test set, with the R^2 score close to 0.8. However, for 2Mbps, the MAE for the test set increases to 3.04. This is due to the limitation of the CRF range up to 51, which restricts the bitrate to values that some test videos cannot achieve at 3840×2160 resolution and 60 fps, resulting in higher prediction errors.

Target Bitrate	Train data			Test data		
	MAE	MSE	R^2 score	MAE	MSE	R^2 score
2 Mbps	2.22	9.57	0.81	3.04	18.71	0.40
5 Mbps	0.82	1.09	0.95	1.40	3.66	0.80
8 Mbps	0.79	1.01	0.94	1.32	3.10	0.81
12 Mbps	0.75	0.92	0.93	1.22	2.65	0.82

TABLE 3. THE ACCURACY METRICS FOR TRAINING AND TESTING DATA FOR DIFFERENT BITRATE VALUES FOR CRF PREDICTION.

To illustrate the prediction error more clearly, Figure 5 presents a scatter plot of actual versus predicted CRF values for target bitrates of 12 Mbps and 2 Mbps. The dashed line represents a perfect prediction; deviations from this line indicate prediction errors. For 12 Mbps, most points cluster close to the perfect prediction line, with prediction errors generally below 1 CRF value. In contrast, for 2 Mbps, the model struggles to track increases in CRF values beyond 45 and shows similar errors for CRF values below 30, indicating limitations in predicting extreme values in either range.



(a) (b)

FIGURE 5. PREDICTION PERFORMANCE OF THE TRAINED MODEL FOR (A) 12 MBPS AND (B) 2 MBPS.

Quality Deviation

A key limitation of single-pass encoding in live video streaming is its inability to adjust encoding parameters effectively based on frame complexity, which results in noticeable quality fluctuations. This section evaluates quality deviation across three encoding modes, emphasizing the benefits of using CRF-based encoding in *OTPS*. In Figure 6, the per-frame quality of video sequence #453 is shown, comparing single-pass, two-pass, three-pass, and *OTPS* CRF encoding. The single-pass encoding, shown in blue, initially exhibits a sharp quality drop of around 30 VMAF points as it adjusts encoding parameters on the fly to compensate. In contrast, multi-pass encoding methods analyze the video content during the first pass, enabling them to provide more stable quality across frames. *OTPS*'s use of CRF encoding further enhances stability, maintaining the most consistent quality among all modes. Quantitatively, the quality deviation ($Q_{deviation}$) for single-, two-, three-pass, and *OTPS* are 33.7%, 11.2%, 10.3%, and 4.9%, respectively. This highlights the superior consistency and suitability of *OTPS* for live streaming applications.

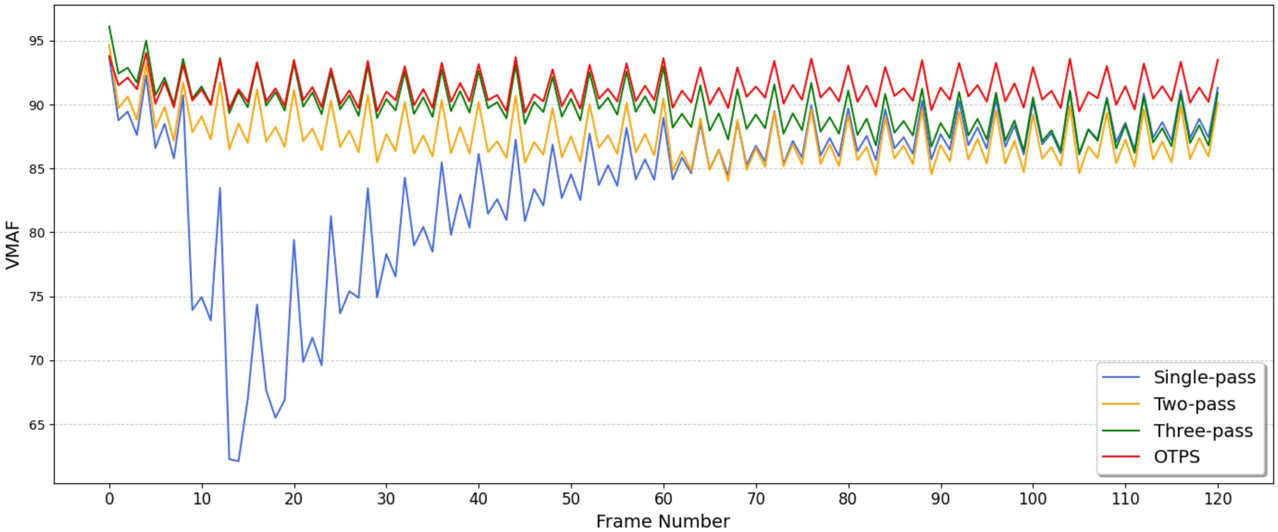


FIGURE 6. PER FRAME QUALITY OF ENCODED VIDEO WITH DIFFERENT ENCODING MODES FOR VIDEO #453.

OTPS Performance

To demonstrate the impact of prediction error on encoding results, the test videos were encoded using the predicted CRF values for all four target bitrates. Table 4 reports the compression efficiency, measured by BD-Rate, and time savings. Compared to single-pass encoding, *OTPS* achieves a 3.7% improvement in compression efficiency with nearly identical encoding time. When compared to two-pass and three-pass encoding modes, *OTPS* reduces encoding time by 18.8% and 28.1%, respectively, while maintaining comparable compression efficiency.

Encoding Mode	BD-Rate (%)	T_{saving} (%)
Single-pass	-3.70	1.84
Two-pass	-0.92	18.80
Three-pass	-0.54	28.14

TABLE 4. THE PERFORMANCE OF *OTPS* COMPARED TO SINGLE- AND MULTI-PASS ENCODING.

To have a more detailed analysis, Figure 7 presents the compression efficiency of *OTPS* compared to single-pass encoding across all test video sequences. In this scatter plot, red points indicate videos where *OTPS* achieves better compression efficiency (negative BD-Rate values), while blue points denote videos where single-pass encoding performs slightly better. The size of each point reflects the magnitude of the efficiency difference: larger red points represent significantly improved compression efficiency achieved by *OTPS*. This visualization indicates that *OTPS* performs particularly well even for complex videos with limited data available in the training set. Additionally, most blue points are quite small, suggesting that *OTPS* closely matches single-pass mode performance for those videos.

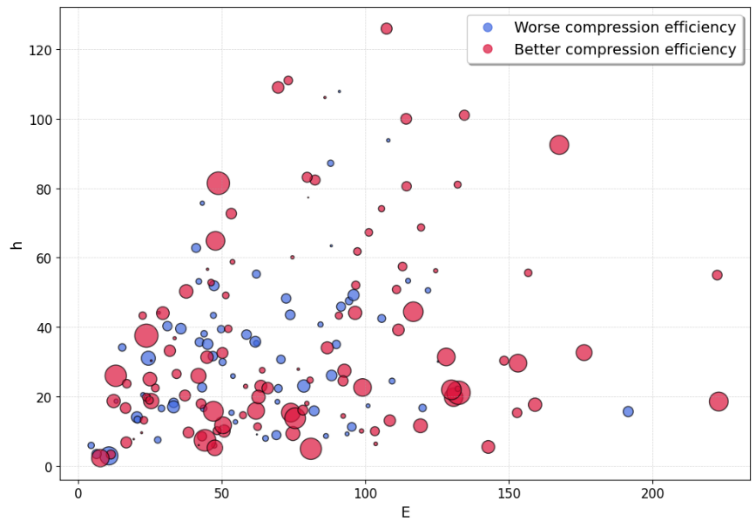


FIGURE 7. THE COMPRESSION EFFICIENCY OF *OTPS* COMPARED TO SINGLE-PASS ENCODING.

Furthermore, four videos with varying levels of spatial and temporal complexity were selected to illustrate RD performance, as shown in Figure 8. In this figure, vertical dashed lines indicate the target bitrates. Notably, *OTPS* achieves RD-curves that are very similar to those of three-pass encoding, demonstrating comparable quality across the bitrate spectrum. By accurately predicting CRF values to meet target bitrates, *OTPS* achieves higher video quality than single-pass and two-pass modes. Only in video #997 does three-pass encoding achieve slightly closer alignment with the target bitrate, though this comes with increased encoding time.

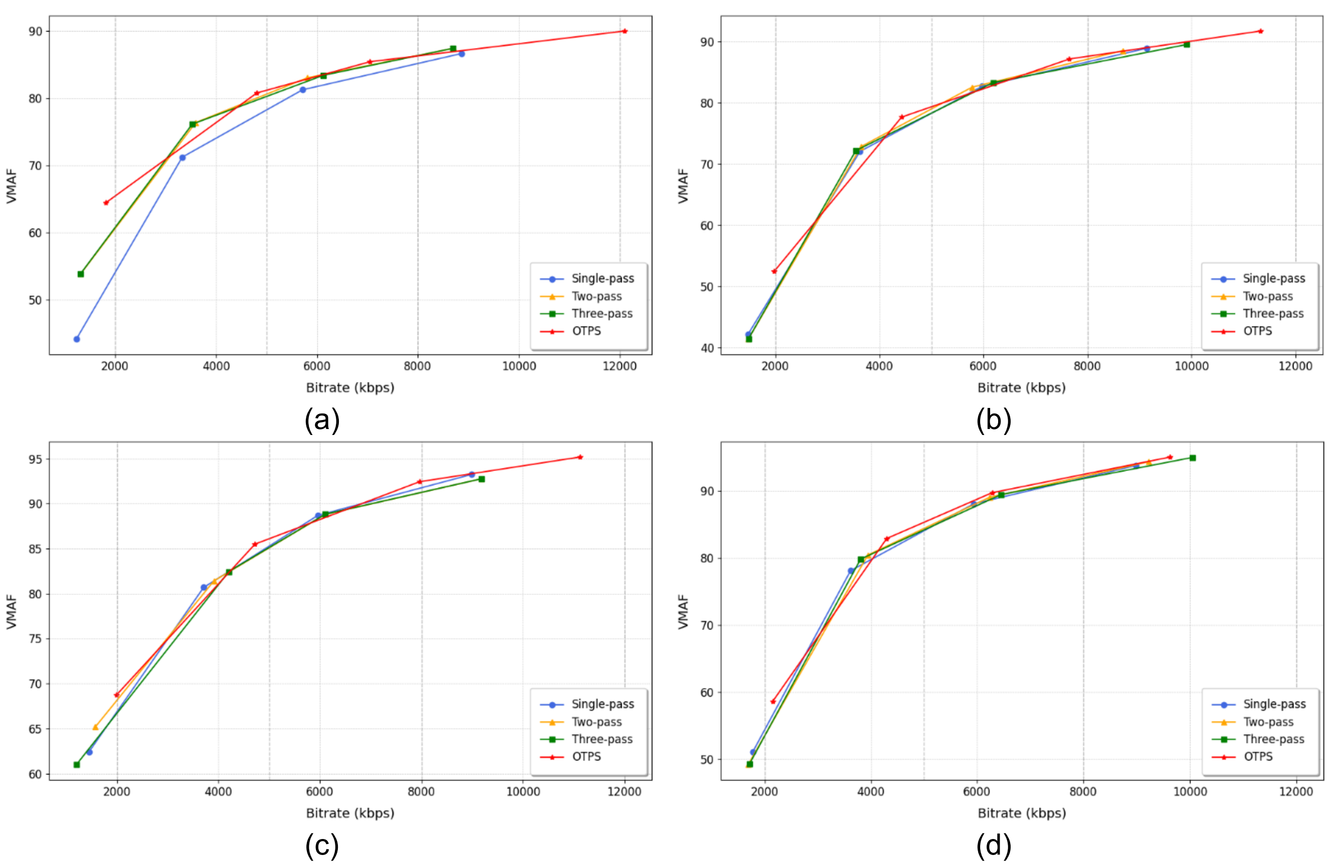


FIGURE 8. RD-CURVES FOR VIDEOS (A) #193, (B) #551, (C) #901, AND (D) #997.

Achieving a bitrate close to the target is crucial. If the output bitrate is below the target, video quality is lower; if it significantly exceeds the target, buffering issues may arise. To assess how each encoding mode performs in this regard, we calculate the bitrate deviation ($B_{deviation}$) for each mode across all four target bitrates, as shown in Table 5. The results demonstrate that *OTPS* consistently achieves a bitrate closer to the target compared to the single-pass, two-pass, and three-pass modes across all targets. On average, *OTPS* produces approximately 5% less deviation from the target bitrate, underscoring its effectiveness in balancing quality and streaming stability.

Target Bitrate	$B_{deviation}$ (%)			
	Single-pass	Two-pass	Three-pass	*OTPS*
2 Mbps	22.86	21.80	22.59	20.43
5 Mbps	23.14	21.98	23.61	16.16
8 Mbps	22.01	22.37	21.59	16.75
12 Mbps	21.89	22.51	21.28	15.72
Average	22.47	22.16	22.26	17.26

TABLE 5. THE PERCENTAGE OF DEVIATION FROM THE TARGET

Conclusions

In this paper, we proposed *OTPS*, an online two-pass encoding scheme, designed specifically for live video streaming. Traditionally, single-pass encoding is used for live-streaming applications. Live streaming relies on single-pass encoding, which, while fast, is less efficient due to its lack of video content analysis before encoding. This could also lead to quality fluctuations during playback. Multi-pass encoding, though more efficient, introduces latency that is impractical for live applications. *OTPS* addresses these limitations by leveraging efficient feature extraction techniques on a downscaled version of the video. A well-selected set of features, combined with a gradient boosting regression model, enables *OTPS* to predict the optimal CRF that aligns closely with the target bitrate in the output, providing high encoding efficiency and consistent quality. To improve prediction accuracy, a separate model is trained for each target bitrate. Given that only a limited number of bitrates are typically used in streaming, it is feasible to manage multiple lightweight models. *OTPS*'s performance is evaluated across four areas: compression efficiency, encoding time, bitrate accuracy, and quality consistency. Experimental results demonstrate that *OTPS* achieves 3.7% greater compression efficiency than single-pass encoding with comparable time consumption, thus adding no significant latency to live streaming workflows. Furthermore, *OTPS* offers time savings of 18.8% and 28.1% compared to two-pass and three-pass encoding modes, respectively, while providing similar compression efficiency. In terms of output bitrate accuracy to the target bitrate, *OTPS* maintains a deviation of around 5% from the target bitrate, outperforming other encoding modes tested. Quality consistency is also improved, with *OTPS* showing only a 4.9% quality deviation, significantly lower than the 33.7% observed with single-pass encoding. These results position *OTPS* as a fast, efficient, and suitable solution for live video streaming applications.

References

[1] Sodagar, I., "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," IEEE Multimedia, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[2] Que, C., Chen, G., and Liu, J., "An Efficient Two-Pass VBR Encoding Algorithm for H.264," in 2006 International Conference on Communications, Circuits and Systems, vol. 1, 2006, pp. 118–122.

[3] Wang, S., Rehman, A., Zeng, K., Wang, J., and Wang, Z., "SSIM-Motivated Two-Pass VBR Coding for HEVC," IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 10, pp. 2189–2203, 2017.

[4] Kumar, M. V. P., Ravi, K., and Mahapatra, S., "A Novel Two Pass Rate Control Scheme for Variable Bit Rate Video Streaming," in 2015 IEEE International Symposium on Multimedia (ISM), 2015, pp. 140–143.

[5] Cao, G., Pan, X., Zhou, Y., Li, Y., and Chen, Z., "Two-pass rate control for constant quality in high efficiency video coding," in 2018 IEEE Visual Communications and Image Processing (VCIP), 2018, pp. 1–4.

[6] Amirpour, H., Ghasempour, M., Qu, L., Hamidouche, W., and Timmerer, C., "EVCA: Enhanced Video Complexity Analyzer," in Proceedings of the 15th ACM Multimedia Systems Conference, ser. MMSys '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 285–291.

[7] Hall, M., "Correlation-Based Feature Selection for Machine Learning," Department of Computer Science, vol. 19, 06 2000.

[8] Tibshirani, R., (1996). "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.

[9] Wiegand, T., Sullivan, G., Bjontegaard, G., and Luthra, A., "Overview of the H.264/AVC Video Coding Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, pp. 560–576, 2003.

[10] Stergiou, A., and Poppe R., "AdaPool: Exponential Adaptive Pooling for Information-Retaining Downsampling," IEEE Transactions on Image Processing, vol. 32, pp. 251–266, 2023.

[11] Bjontegaard, G., "Calculation of average PSNR differences between RD-curves," ITU SG16 Doc. VCEG-M33, 2001.

[12] Li, Z., Swanson, K., Bampis, C., Krasula, L., and Aaron, A., "Toward a Better Quality Metric for the Video Community," Dec. 2020.

[13] Menon, V. V., Amirpour, H., Feldmann, C., Ilangovan, A., Smole, M., Ghanbari, M., Timmerer, C., "Live-PSTR: Live Per-title Encoding for Ultra HD Adaptive Streaming," in Proc. 2022 NAB Broadcast Engineering and Information Technology Conference, Las Vegas, USA, 2022.

Appendix A1: Command-line Options

Feature extraction using X264

- ffmpeg -hide_banner -i $input.mp4 -vf "scale=640x360,select=not(mod(n\,6))" -preset ultrafast -c:v libx264 -crf 32 -vsync 0 -f null –

Bitrate calculation

- ffprobe -v info -select_streams v:0 -show_entries stream=bit_rate -of default=noprint_wrappers=1:nokey=1 $output.mp4

Quality assessment

- vmaf -r $reference.yuv -d $distorted.yuv -w $w -h $h -p $p -b $b -o quality.json --json --feature psnr -m path=$model_path