Perceptually-aware Live VBR Encoding Scheme for Adaptive Video Streaming

Vignesh V Menon
Christian Doppler Laboratory ATHENA, Alpen-Adria-Universität, Klagenfurt, Austria

13 Dec 2022
Outline

1 Introduction
2 Research Goal
3 Live-VBR scheme
4 Results
5 Summary and Future Directions
Introduction

HTTP Adaptive Streaming (HAS)

Why Adaptive Streaming?
- Adapt for a wide range of devices.
- Adapt for a broad set of Internet speeds.

What HAS does?
- Each source video is split into segments.
- Encoded at multiple bitrates, resolutions, and codecs.
- Delivered to the client based on the device capability, network speed etc.

---

HTTP Adaptive Streaming (HAS) has become the *de-facto* standard in delivering video content for various clients regarding internet speeds and device types.

Traditionally, a fixed bitrate ladder, *e.g.* HTTP Live Streaming (HLS) bitrate ladder\(^2\), is used in live streaming.

---

Introduction
Motivation for Per-title bitrate ladder in Adaptive Streaming

Each resolution performs better than others in a specific region for a given bitrate range. These regions depend on the *video content complexity*.

**Figure:** Rate-Distortion (RD) curves of the Constant Bitrate (CBR) encoding of *RushHour_s000* and *YachtRide_s000* video sequences (segments) of VCD dataset\(^3\) encoded at 1080p and 2160p resolutions using x265 HEVC encoder at *ultrafast* preset. Here, VMAF is used as the quality metric.

---

Introduction

Per-title Encoding

- Though per-title encoding schemes\textsuperscript{4,5,6} enhance the quality of video delivery, determining the \textit{convex-hull} is computationally costly, making it suitable for only VoD streaming applications.
- The plethora of live streaming applications call for low latency approaches to optimize video coding.
- According to the Bitmovin Video Developer Report 2021\textsuperscript{7}, \textit{live (low) latency} is the biggest challenge in video technology today.

\footnotesize


\textsuperscript{7} https://go.bitmovin.com/video-developer-report, last access: Dec 13, 2022.
Introduction
Motivation for perceptually-aware bitrate ladder

Figure: RD curve of the HLS CBR encoding of Characters_s000 video sequence (segment) of VCD dataset using x265 HEVC encoder at ultrafast preset. The points with a bitrate greater than 3.6 Mbps are in the perceptually lossless region. Hence, there is significant storage wastage while storing these representations.

Selecting similar-quality representations for the bitrate ladder does not result in improved QoE, but it will lead to increased storage and bandwidth costs!
Introduction
Motivation for two-pass encoding (CBR versus VBR)

Figure: Constant Bitrate (CBR) versus Variable Bitrate (VBR) encoding.

- In live streaming, Constant Bitrate (CBR) rate-control mode is used to encode video sequences at a fixed bitrate ladder. The consistency of CBR makes it more reliable for time-sensitive data transport.
- In this method, there is no concern about the bitrate exceeding internet speeds. However, this method may result in low compression efficiency.
Introduction

Constrained Variable Bitrate (cVBR) encoding

Figure: cVBR encoding.

- A "rate factor" first-pass to identify the optimized CRF to achieve the target bitrate.
- In the second pass, the segment is encoded with the selected optimized CRF with the maximum bitrate and maximum buffer window constraints.
- The desired target bitrate is achieved with maximum compression efficiency and minimum quality fluctuation.
Joint optimization:

- Perceptual difference of pre-defined $\Delta \text{VMAF}$ between representations.
- Minimize bitrate difference between representations.
- Maximize compression efficiency of representations.
Workflow of Live-VBR

Figure: Live-VBR system envisioned in this work.
Video Complexity Feature Extraction

Compute texture energy per block

A DCT-based energy function is used to determine the block-wise feature of each frame defined as:

\[
H_k = \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} e^{(\frac{ij}{wh})^2-1}|DCT(i,j)| \tag{1}
\]

where \(wxw\) is the size of the block, and \(DCT(i,j)\) is the \((i,j)th\) DCT component when \(i+j > 0\), and 0 otherwise.

The energy values of blocks in a frame are averaged to determine the energy per frame.\(^8,9\)

\[
E_s = \sum_{k=0}^{K-1} \frac{H_{s,k}}{K \cdot w^2} \tag{2}
\]

---


Video Complexity Feature Extraction

$h_s$: SAD of the block level energy values of frame $s$ to that of the previous frame $s - 1$.

\[
h_s = \sum_{k=0}^{K-1} \left| \frac{H_{s,k}, H_{s-1,k}}{K \cdot w^2} \right|
\]  

(3)

where $K$ denotes the number of blocks in frame $s$.

The luminescence of non-overlapping blocks $k$ of $s^{th}$ frame is defined as:

\[
L_{s,k} = \sqrt{DCT(0,0)}
\]  

(4)

The block-wise luminescence is averaged per frame denoted as $L_s$ as shown below.\(^\text{10}\)

\[
L_s = \sum_{k=0}^{K-1} \frac{L_{s,k}}{K \cdot w^2}
\]  

(5)

Live-VBR

First point of the bitrate ladder

\[ b_0 = b_{\text{min}} \]

Determine \( v_r, b_0 \) \( \forall r \in R \)

\[ v_0 = \max (v_r, b_0) \]

\[ r_0 = \arg \max_{r \in R} (v_r, b_0) \]

\((r_0, b_0)\) is the first point of the bitrate ladder

**Note**

This part of the algorithm needs VMAF prediction for all considered resolutions.

---

Live-VBR

Remaining points of the bitrate ladder

**Figure:** Estimation of the \((t+1)^{th}\) point of the bitrate ladder. \(b_t\) is the minimum value among the \(b_{r,v_t}\) values output from the predicted models trained for resolutions \(r_0, r_1, \ldots, r_{M-1}\). The resolution corresponding to the bitrate \(b_t\) is chosen as \(r_t\).

\[
\begin{align*}
& t = 1 \\
& \textbf{for } t \geq 1 \textbf{ do} \\
& \quad v_t = v_{t-1} + \Delta \text{VMAF} \\
& \quad \text{Determine } b_{r,v_t} \forall r \in R \\
& \quad b_t = \min(b_{r,v_t}) \\
& \quad r_t = \arg \min_{r \in R} (b_{r,v_t}) \\
& \quad \textbf{if } b_t > b_{\max} \text{ or } v_t > v_{\max} \textbf{ then} \\
& \quad \quad \text{End of the algorithm} \\
& \quad \textbf{else} \\
& \quad \quad (r_t, b_t) \text{ is the } (t+1)^{th} \text{ point of the bitrate ladder.} \\
& \quad \quad t = t + 1
\end{align*}
\]

**Note**

This part of the algorithm needs bitrate prediction for all considered resolutions.
Live-VBR

Phase 2: Bitrate ladder Estimation

cVBR encoding of the bitrate ladder

Figure: Estimation of the optimized CRF to achieve the target bitrate $b$ using a prediction model trained for resolution $r$.

- Optimized CRF is determined for the selected $(r, b)$ pairs.
- cVBR encoding for the $(r, b, \text{CRF})$ pairs is performed.

---

Results

Prediction accuracy of the models

**Table:** $R^2$ score and MAE of the prediction models for various resolutions.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VMAF</td>
<td>0.821</td>
<td>0.852</td>
<td>0.882</td>
<td>0.906</td>
<td>0.910</td>
<td>0.906</td>
<td>0.930</td>
<td>4.860</td>
<td>4.899</td>
<td>4.832</td>
<td>4.393</td>
<td>3.838</td>
<td>3.490</td>
<td>2.941</td>
</tr>
<tr>
<td>log(b)</td>
<td>0.859</td>
<td>0.864</td>
<td>0.888</td>
<td>0.915</td>
<td>0.932</td>
<td>0.937</td>
<td>0.943</td>
<td>0.765</td>
<td>0.751</td>
<td>0.737</td>
<td>0.709</td>
<td>0.711</td>
<td>0.706</td>
<td>0.681</td>
</tr>
<tr>
<td>CRF</td>
<td>0.969</td>
<td>0.969</td>
<td>0.970</td>
<td>0.969</td>
<td>0.968</td>
<td>0.967</td>
<td>0.965</td>
<td>1.924</td>
<td>1.920</td>
<td>1.914</td>
<td>1.942</td>
<td>1.940</td>
<td>1.972</td>
<td>1.990</td>
</tr>
</tbody>
</table>

Note

Just three values ($E, h, L$) are used as the measure of video complexity. If we increase the information measure, e.g., block-wise features), the accuracy can be improved further.
Results

RD plots of Live-VBR using x265

Figure: Bunny_s000 ($E = 22.40$, $h = 4.70$)

Figure: Characters_s000 ($E = 45.42$, $h = 36.88$)
Results

RD plots of Live-VBR using x265

Figure: Eldorado_s005 ($E = 100.37, h = 9.23$)

Figure: Wood_s000 ($E = 124.72, h = 47.03$)
Results

Summary

Table: Average results of the encoding schemes compared to the HLS CBR encoding using x265 HEVC encoder.

<table>
<thead>
<tr>
<th>Method</th>
<th>$BDR_p$</th>
<th>$BDR_V$</th>
<th>BD-PSNR</th>
<th>BD-VMAF</th>
<th>$\Delta S$</th>
<th>$\Delta E$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground truth ($\Delta VMAF=2$)</td>
<td>-23.09%</td>
<td>-43.23%</td>
<td>1.34 dB</td>
<td>10.61</td>
<td>-25.99%</td>
<td>89.54%</td>
</tr>
<tr>
<td>Ground truth ($\Delta VMAF=4$)</td>
<td>-28.15%</td>
<td>-42.75%</td>
<td>1.70 dB</td>
<td>10.08</td>
<td>-59.07%</td>
<td>-0.54%</td>
</tr>
<tr>
<td>Ground truth ($\Delta VMAF=6$)</td>
<td>-25.36%</td>
<td>-40.73%</td>
<td>1.67 dB</td>
<td>9.19</td>
<td>-70.50%</td>
<td>-31.24%</td>
</tr>
<tr>
<td>Live-VBR ($\Delta VMAF=2$)</td>
<td>-14.25%</td>
<td>-29.14%</td>
<td>1.36 dB</td>
<td>7.82</td>
<td>23.57%</td>
<td>90.19%</td>
</tr>
<tr>
<td>Live-VBR ($\Delta VMAF=4$)</td>
<td>-18.41%</td>
<td>-32.48%</td>
<td>1.41 dB</td>
<td>8.31</td>
<td>-56.38%</td>
<td>0.34%</td>
</tr>
<tr>
<td>Live-VBR ($\Delta VMAF=6$)</td>
<td>-18.80%</td>
<td>-32.59%</td>
<td>1.34 dB</td>
<td>8.34</td>
<td>-68.96%</td>
<td>-28.25%</td>
</tr>
</tbody>
</table>

Relative storage difference

$$\Delta S = \frac{\sum b_{opt}}{\sum b_{ref}} - 1$$

Relative energy utilization difference

$$\Delta E = \frac{\sum E(b_{opt})}{\sum E(b_{ref})} - 1$$
Summary and Future Directions

- Presented an application of video complexity analysis, where VMAF, target bitrate, and CRF are predicted using video complexity features.

In the future, we shall include the following:
- Optimized encoding framerate
- Optimized encoding preset and number of CPU threads
Thank you for your attention!

Vignesh V Menon (vignesh.menon@aau.at)