Optimal Video Encoding @ Scale: Dynamic Optimizer

Ioannis Katsavounidis
Video Algorithms,
Netflix
R: 1Mbps  
R: 1Mbps  
R: 1Mbps  
...  
R: 1Mbps  

R: 100kbps  
R: 500kbps  
...  
R: 2000kbps
Motivation

1. Why use same encoding parameters throughout a long and diverse video sequence?
2. Why impose a fixed Intra-frame interval?
3. Why consider (only) compression artifacts in video quality?
4. Why use MSE (PSNR)?
5. How to choose optimal combination of encoding parameters for a long and diverse video sequence?
6. How can we obtain the entire convex hull of optimal encodes?
Framework: Single shot processing

Input

- High Quality
- Scaled VMAF, PSNR

Encoding parameters

- Encoding parameters
- Non scaled VMAF, PSNR

Encode

Decode

Bitrate

- R, D_s, D_ns point
Framework: Convex hull of optimal shot encodes
Framework: Trellis

Number of pre-encodes = number of QPs (n) x number of shots (m)

VP9, 63 QPs
Framework: Trellis optimal path

- Lowest (average) bitrate encode, with quality y
- Highest (average) quality encode, with bitrate x kbps
- Lowest (average) bitrate encode, with quality y

Fixed QP encode

QP

Shot
Framework: Resolutions

Number of pre-encodes = number of QPs (n) x number of shots (m) x number of resolutions (r)

VP9, 63 QPs, 7 resolutions
Results: Video Content

- 10 titles
  - 8 representative titles from NETFLIX catalog
  - 2 publicly available ("El Fuente" and "Meridian")
Results: DO VP9 vs. Per-Title Optimal QP
## Results: DO vs. Fixed Q BD-rate

<table>
<thead>
<tr>
<th>Title</th>
<th>VMAF</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bloodline</td>
<td>-18.88%</td>
<td>-31.26%</td>
</tr>
<tr>
<td>BoJack</td>
<td>-48.39%</td>
<td>-60.18%</td>
</tr>
<tr>
<td>BreakingBad</td>
<td>-22.81%</td>
<td>-38.10%</td>
</tr>
<tr>
<td>Daredevil</td>
<td>-20.07%</td>
<td>-34.42%</td>
</tr>
<tr>
<td>ElFuente</td>
<td>-25.93%</td>
<td>-45.58%</td>
</tr>
<tr>
<td>HoC</td>
<td>-25.85%</td>
<td>-34.52%</td>
</tr>
<tr>
<td>Meridian</td>
<td>-38.82%</td>
<td>-70.11%</td>
</tr>
<tr>
<td>OINB</td>
<td>-19.22%</td>
<td>-31.92%</td>
</tr>
<tr>
<td>TheAvengers</td>
<td>-25.61%</td>
<td>-37.32%</td>
</tr>
<tr>
<td>WHAS</td>
<td>-14.45%</td>
<td>-38.13%</td>
</tr>
<tr>
<td><strong>AVERAGE</strong></td>
<td><strong>-25.72%</strong></td>
<td><strong>-41.96%</strong></td>
</tr>
</tbody>
</table>
Results: Applied to AVC-High, VP9 and HEVC

- Run at scale (production pipeline)
- Multiple titles (30)
- About 10 min. each
- Reasonably slow speed settings

<table>
<thead>
<tr>
<th>Codec</th>
<th>VMAF</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVC-High (x264)</td>
<td>-28.04%</td>
<td>-27.99%</td>
</tr>
<tr>
<td>VP9 (libvpx)</td>
<td>-37.61%</td>
<td>-36.97%</td>
</tr>
<tr>
<td>HEVC (x265)</td>
<td>-33.51%</td>
<td>-30.52%</td>
</tr>
</tbody>
</table>
Parallel Encoding

Input → N Tasks → Chunk Encode → N chunks → Assembly → Bitrate Ladder

Encode
Encode_1
Encode_2
Encode_M

Tasks = chunks * encodes
## Implementation challenges

### VP9, 63 QPs, 7 resolutions

<table>
<thead>
<tr>
<th></th>
<th>Shots</th>
<th>Encodes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avengers</td>
<td>2915</td>
<td>441</td>
<td>1,285,515</td>
</tr>
<tr>
<td>El Fuente</td>
<td>96</td>
<td>441</td>
<td>42,336</td>
</tr>
</tbody>
</table>

Number of actual encodes

Number of tasks

Tasks = chunks * encodes

- ~3 min chunks
- Encodes
- Tasks

<table>
<thead>
<tr>
<th></th>
<th>~3 min chunks</th>
<th>Encodes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avengers</td>
<td>48</td>
<td>10</td>
<td>480</td>
</tr>
<tr>
<td>El Fuente</td>
<td>2</td>
<td>10</td>
<td>20</td>
</tr>
</tbody>
</table>
## Number of encodes

<table>
<thead>
<tr>
<th></th>
<th>Shots</th>
<th>Encodes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avengers</td>
<td>2915</td>
<td>441</td>
<td>1,285,515</td>
</tr>
<tr>
<td>El Fuente</td>
<td>96</td>
<td>441</td>
<td>42,336</td>
</tr>
</tbody>
</table>

- **Avengers**
  - Shots: 2915
  - Encodes: 441
  - Tasks: 1,285,515

- **El Fuente**
  - Shots: 96
  - Encodes: 441
  - Tasks: 42,336

### The point of diminishing returns

Find subset of operating points that produce “equivalent” performance to the “full” optimizer

*Constrained* dynamic optimizer

\[
\text{Tasks} = \text{chunks} \times \text{encodes}
\]
### Number of tasks

<table>
<thead>
<tr>
<th></th>
<th>Shots</th>
<th>Encodes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avengers</td>
<td>2915</td>
<td>35</td>
<td>102,025</td>
</tr>
<tr>
<td>El Fuente</td>
<td>96</td>
<td>35</td>
<td>3,360</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Chunks</th>
<th>Encodes</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avengers</td>
<td>46</td>
<td>35</td>
<td>1610</td>
</tr>
<tr>
<td>El Fuente</td>
<td>2</td>
<td>35</td>
<td>70</td>
</tr>
</tbody>
</table>

#### Collation

- Combine multiple shots into chunks
- Checkpoints after every shots

Tasks = chunks * encodes
Internal Spot market

- Borrow unused instances
- Daily peak of ~12,000 instances

115 movies, runtime between 2 - 3 hrs, numshots range from 725 to 3973, total shots 235,017

<table>
<thead>
<tr>
<th>Codec</th>
<th>Total CPU time</th>
<th>Avg CPU time</th>
<th>Avg wall clock time</th>
</tr>
</thead>
<tbody>
<tr>
<td>H264</td>
<td>15,281 days</td>
<td>132 days</td>
<td>7 days</td>
</tr>
<tr>
<td>AVC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VP9</td>
<td>38,284 days</td>
<td>332 days</td>
<td>9 days</td>
</tr>
</tbody>
</table>
So far...

<table>
<thead>
<tr>
<th>Shots</th>
<th>CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>18,436,049</td>
<td>1,466,311 days</td>
</tr>
</tbody>
</table>

“Compute complexity is the currency we use to buy video quality”
David Ronca, Director of Encoding Technologies, VQEG meeting @NETFLIX, May 2017
Demo
Summary

- Joint optimization of shots
- Codec agnostic and object metric agnostic framework
- Orthogonal to I/P/B quality optimization by codecs
- Upper bound to compare rate control mechanisms within and between codecs
- Provides ~25% bitrate savings at same quality
- Streams are 100% compliant; ready to be consumed by existing clients
Academic research partners

University of Texas Austin
  Prof. Al Bovik
  Todd Goodall
  Christos Bampis
  Zeina Sinno

Université de Nantes
  Prof. Patrick Le Callet
  Lukáš Krasula

University of Southern California
  Prof. C.-C. Jay Kuo
  Joe Yuchieh Lin
  Haiqiang Wang

University of Bristol
  Prof. David Bull
  Felix Mercer Moss
  Mariana Afonso
Open problems

Perceptual quality metrics
- VMAF for HDR and 4K
- Temporal features
- Temporal pooling

Dynamic optimizer complexity reduction

Next-generation video codecs
- AV1 speed-up
- Integrating perceptual metrics

Next-generation image compression
Picture Coding Symposium 2018
June 24-27, San Francisco

Paper deadline: January 10, 2018
Special session: “System-level perceptual video coding optimization”