High-efficiency AV1: dAV1d and Eve-AV1

Getting the most out of AV1; how to make it even better

Ronald S. Bultje <rbultje@twoorioles.com>
Founder, Two Orioles
- **Videolan’s AV1 decoder**
  - Sponsored by AOMedia
  - Released in Sept. 2018
  - 2-clause BSD license
  - by Two Orioles, VideoLabs, MultiCoreWare & many individual contributors

- Fast & multi-threaded
- Low memory usage
- Lean source code
- Small binary size
- Adoption

- AV1 challenges for decoders

---

https://code.videolan.org/videolan/dav1d
- Videolan’s AV1 decoder
- **Fast & multi-threaded**
  - 2-5x as fast as libaom
  - 4-10x as fast as gav1
  - AV1/HEVC decoding have roughly same complexity
  - AV1 decoding is 30% more complex than VP9/H264
- Low memory usage
- Lean source code
- Small binary size
- Adoption

- AV1 challenges for decoders
- Videolan’s AV1 decoder
- **Fast & multi-threaded**
  - 2-5x as fast as libaom
  - 4-10x as fast as gav1
  - AV1/HEVC decoding have roughly same complexity
  - AV1 decoding is 30% more complex than VP9/H264
- Low memory usage
- Lean source code
- Small binary size
- Adoption
- AV1 challenges for decoders
- Videolan’s AV1 decoder
- Fast & multi-threaded
- Low memory usage
  - 30%-50% less than libaom
  - similar to gav1 with 1 thread and 35% more w/ threading
  - 40-50% less than other codecs w/ threading
- Lean source code
- Small binary size
- Adoption

- AV1 challenges for decoders
- Videolan’s AV1 decoder
- Fast & multi-threaded
- **Low memory usage**
  - 30%-50% less than libaom
  - similar to gav1 with 1 thread and 35% more w/ threading
  - 40-50% less than other codecs w/ threading
- Lean source code
- Small binary size
- Adoption

- AV1 challenges for decoders
- Videolan’s AV1 decoder
- Fast & multi-threaded
- Low memory usage
- **Lean source code + SIMD**
  - dav1d: SSSE3-AVX2 (x86), 64bit Neon (arm)
  - libaom: SSSE3-AVX2 (x86), 32+64bit Neon (arm)
  - gav1 has full SSE4.1 (x86), 32+64bit Neon (arm)
- Small binary size
- Adoption

**AV1 challenges for decoders**

<table>
<thead>
<tr>
<th>kLOC, decoder only</th>
<th>dav1d</th>
<th>libaom</th>
<th>gav1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/C++</td>
<td>34.6</td>
<td>87.2</td>
<td>45.5</td>
</tr>
<tr>
<td>x86 asm</td>
<td>43.1</td>
<td>68.5</td>
<td>15.6</td>
</tr>
<tr>
<td>arm asm</td>
<td>18.7</td>
<td>17.2</td>
<td>14.7</td>
</tr>
<tr>
<td>ppc asm</td>
<td>1.0</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>mips asm</td>
<td></td>
<td>15.7</td>
<td></td>
</tr>
</tbody>
</table>
Videolan’s AV1 decoder
Fast & multi-threaded
Low memory usage
Lean source code
Small binary size
Adoption

AV1 challenges for decoders

<table>
<thead>
<tr>
<th></th>
<th>dav1d</th>
<th>libaom</th>
<th>gav1</th>
</tr>
</thead>
<tbody>
<tr>
<td>kB, decoder only</td>
<td>926</td>
<td>2936</td>
<td>1461</td>
</tr>
</tbody>
</table>
Videolan’s AV1 decoder
• Fast & multi-threaded
• Low memory usage
• Lean source code
• Small binary size
• Adoption

AV1 challenges for decoders

Firefox brings you smooth video playback with the world’s fastest AV1 decoder

By Nathan Egge, Christopher Montgomery

Posted on May 23, 2019 in AV1, Featured Article, Firefox, Performance, and Research

Tuesday’s release of Firefox 67 brought a number of performance enhancing features that make this our fastest browser ever. Among these is the high performance, royalty free AV1 video decoder dav1d, now enabled by default on all desktop platforms (Windows, OSX and Linux) for both 32-bit and 64-bit systems.


• VLC 3.1 (April 8)
• Chrome M74 (April 23)
• Firefox 67 (May 14)
• FFmpeg 4.2 (August 5)
• You? (soon!)
• Tools
  • So many (~ implementation complexity)
  • Confusing rules for which tools are available at which block sizes
    • e.g. why are compound inter/inter wedges allowed at all block sizes between 8x8 and 32x32, but inter/intra wedges only at 2:1, 1:1 and 1:2 block sizes between 8x8 and 32x32?

• Symbol coding
  • Compound inter/inter type or intra prediction mode is only partially multi-symbol’ed
  • Coef high token coding is loopy, which hurts SIMD implementations
  • Grain scaling points are not using quniform
  • Motion vector range limits (2k pixels)

• Overall, things look pretty good 🙂
Eve-AV1

- **Two Orioles’ AV1 encoder**
  - Closed-source / proprietary
  - VoD, offline encoding
  - High-value content
    - *high-speed presets in progress*
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders

https://twoorioles.com/
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders
Eve-AV1

- Two Orioles' AV1 encoder
  - Closed-source / proprietary
  - VoD, offline encoding
  - High-value content
  - High-speed presets in progress

Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

AV1 challenges for encoders

3mbps
## Eve-AV1

- Two Orioles’ AV1 encoder
- **Quality vs. Bitrate**
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders

### 1080p clips

<table>
<thead>
<tr>
<th>Encoder</th>
<th>% Bitrate Reduction</th>
<th>Runtime (sec/frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eve-AV1 1.3.5</td>
<td>0.00%</td>
<td>135.57</td>
</tr>
<tr>
<td>libaom a385cc44e</td>
<td>-20.95%</td>
<td>86.13</td>
</tr>
<tr>
<td>rav1e c68d68c</td>
<td>-50.88%</td>
<td>41.01</td>
</tr>
<tr>
<td>SVT-AV1 6fd5646</td>
<td>-33.88%</td>
<td>109.29</td>
</tr>
</tbody>
</table>
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders

Graph showing Bitrate reduction (%) vs. Encoding speed (seconds/frame) with different encoders: Eve-AV1 1.3.5, libaom a385cc44e, SVT-AV1 6fd5646, rav1e c68d68c.
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

- AV1 challenges for encoders

![Graph showing bitrate reduction vs. encoding speed for Eve-AV1 1.3.5, libaom_a385cc44e, SVT-AV1 6fd5646, and rav1e_c68d68c. The graph illustrates how different encoders perform under varying encoding speeds.]
Tools
- So many (coding & code complexity)
- $O(x^n)$ vs. $O(x*n)$ tools
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

Multi-threading
- Limit top/right edge access at SB corners
  - increasing LRU size gives significant coding gains, but increases delay
- Allow rectangular LRUs ($w > h$)?
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)

MT encoder models for AV2?

AV1 challenges for encoders
- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

• Two Orioles’ AV1 encoder
  - Quality vs. Bitrate
  - Quality-per-bit vs. Speed
  - Multi-threading

• AV1 challenges for encoders
  - Two Orioles’ AV1 encoder
    - Quality vs. Bitrate
    - Quality-per-bit vs. Speed
    - Multi-threading
**Tools**
- So many (coding & code complexity)
- $O(x^n)$ vs. $O(x*n)$ tools
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

**Multi-threading**
- **Limit top/right edge access at SB corners**
- increasing LRU size gives significant coding gains, but increases delay
  - Allow rectangular LRUs ($w > h$)?
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?
Two Orioles’ AV1 encoder
Quality vs. Bitrate
Quality-per-bit vs. Speed
Multi-threading

AV1 challenges for encoders

Tools
- So many (coding & code complexity)
- $O(x^n)$ vs. $O(x*n)$ tools
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

Multi-threading
- Limit top/right edge access at SB corners
- increasing LRU size gives significant coding gains, but increases delay
  - Allow rectangular LRUs ($w > h$)?
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?
**Tools**
- So many (coding & code complexity)
- $O(x^n)$ vs. $O(x*n)$ tools
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

**Multi-threading**
- Limit top/right edge access at SB corners
- increasing LRU size gives significant coding gains, but increases delay
  - **Allow rectangular LRUs ($w > h$)?**
  - CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?
Two Orioles’ AV1 encoder
• Quality vs. Bitrate
• Quality-per-bit vs. Speed
• Multi-threading

**Eve-AV1**

- Tools
  - So many (coding & code complexity)
  - $O(x^n)$ vs. $O(x*n)$ tools
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

**AV1 challenges for encoders**

- 1920x1080 frame | 128x128 SBs | 256x256 LRUs

**Multi-threading**

- Limit top/right edge access at SB corners
- increasing LRU size gives significant coding gains, but increases delay
  - **Allow rectangular LRUs ($w > h$)?**
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

AV1 challenges for encoders
- 1920x1080 frame | 128x128 SBs | 256x256 LRUs

Multi-threading
- Limit top/right edge access at SB corners
- Increasing LRU size gives significant coding gains, but increases delay
  - **Allow rectangular LRUs (w > h)?**
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?

- Tools
  - So many (coding & code complexity)
  - $O(x^n)$ vs. $O(x*n)$ tools
    - subpel filters, wedge index, inter/intra mode, reference frame, transform type
    - global motion, deblock, CDEF, loop restoration, film grain
Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

AV1 challenges for encoders

1920x1080 frame | 128x128 SBs | 256x256 LRUs

- Tools
  - So many (coding & code complexity)
  - O(x^n) vs. O(x*n) tools
    - subpel filters, wedge index, inter/intra mode, reference frame, transform type
    - global motion, deblock, CDEF, loop restoration, film grain

Multi-threading

- Limit top/right edge access at SB corners
- increasing LRU size gives significant coding gains, but increases delay
  - Allow rectangular LRUs (w > h)?
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?
Two Orioles’ AV1 encoder
Quality vs. Bitrate
Quality-per-bit vs. Speed
Multi-threading

Eve-AV1

- Tools
  - So many (coding & code complexity)
  - \( O(x^n) \) vs. \( O(x^n) \) tools
    - subpel filters, wedge index, inter/intra mode, reference frame, transform type
    - global motion, deblock, CDEF, loop restoration, film grain

- AV1 challenges for encoders
  - 1920x1080 frame | 128x128 SBs | 128x128 LRUs
  - \( O(x^2) \) vs. \( O(x^2) \)
  - subpel filters, wedge index, inter/intra mode, reference frame, transform type
  - global motion, deblock, CDEF, loop restoration, film grain

- Multi-threading
  - Limit top/right edge access at SB corners
  - increasing LRU size gives significant coding gains, but increases delay
    - \textbf{Allow rectangular LRUs (w > h)?}
  - CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
  - MT encoder models for AV2?
### Eve-AV1

- Two Orioles’ AV1 encoder
- Quality vs. Bitrate
- Quality-per-bit vs. Speed
- Multi-threading

### AV1 challenges for encoders

- Tools
  - So many (coding & code complexity)
  - $O(x^n)$ vs. $O(x*n)$ tools
    - subpel filters, wedge index, inter/intra mode, reference frame, transform type
    - global motion, deblock, CDEF, loop restoration, film grain

### Multi-threading

- Limit top/right edge access at SB corners
- increasing LRU size gives significant coding gains, but increases delay
  - **Allow rectangular LRUs ($w > h$)?**
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?

---

**1920x1080 frame | 128x128 SBs | 256x128 LRUs**

- SB thread 3
  - Frame 1
- LR thread 1
  - Frame 1
- SB thread 2
  - Frame 2

**SB distance (thread 2-3): 20, nSBs: 15*8.5**

Max. concurrency: $15*8.5/20 = 6.4$

**Quality loss: 0.5%**
• **Tools**
  • So many (coding & code complexity)
  • $O(x^n)$ vs. $O(x*n)$ tools
    • subpel filters, wedge index, inter/intra mode, reference frame, transform type
    • global motion, deblock, CDEF, loop restoration, film grain

• **Multi-threading**
  • Limit top/right edge access at SB corners
  • increasing LRU size gives significant coding gains, but increases delay
    • Allow rectangular LRUs ($w > h$)?
  • CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)

• **MT encoder models for AV2?**
Questions?

Ronald S. Bultje <rbultje@twoorioles.com>
Founder, Two Orioles