

# 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](mailto:rbultje@twoorioles.com)>  
Founder, Two Orioles



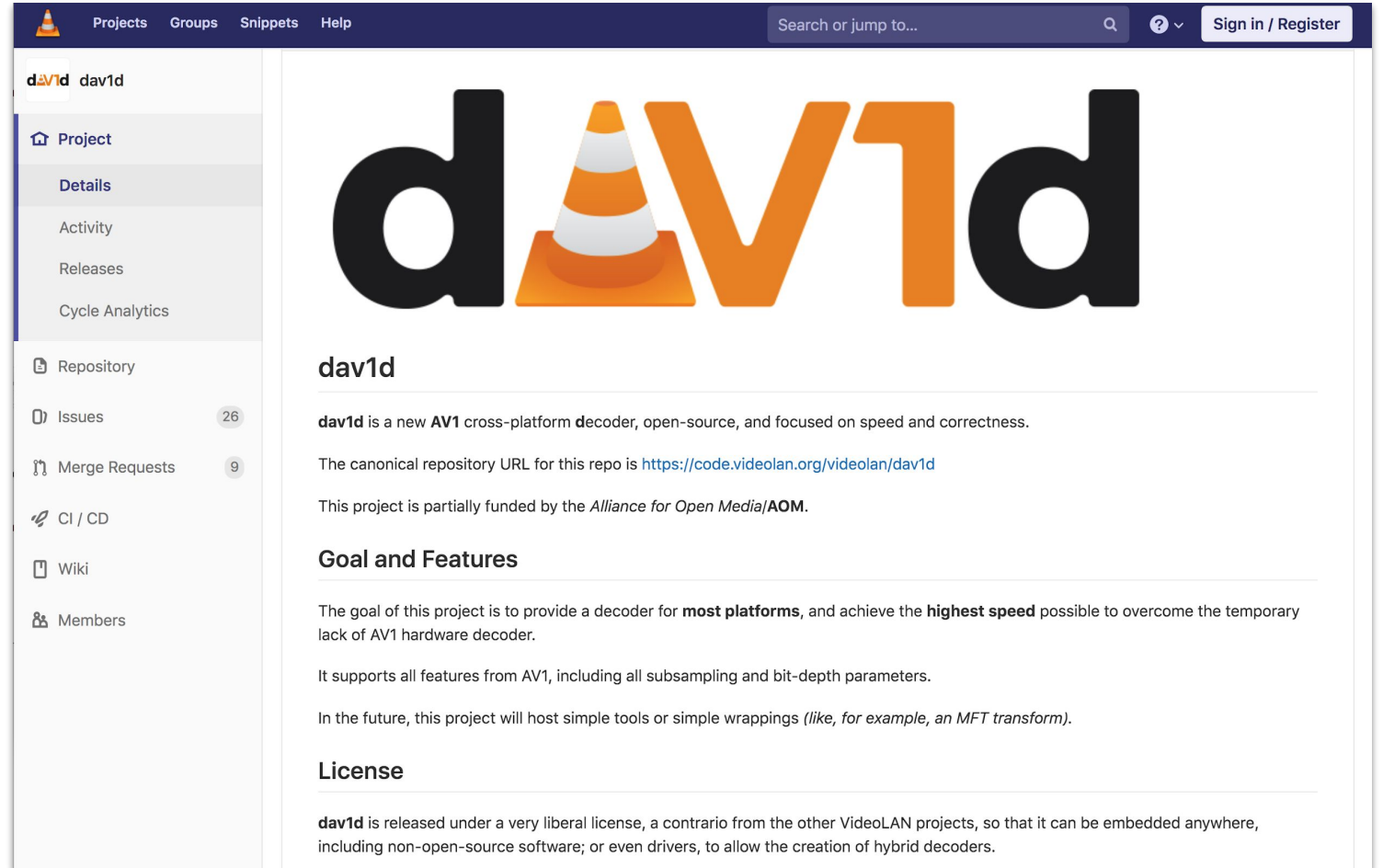
**ALLIANCE FOR  
OPEN MEDIA  
RESEARCH**

**Symposium 2019**

dawid

# dav1d

- **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

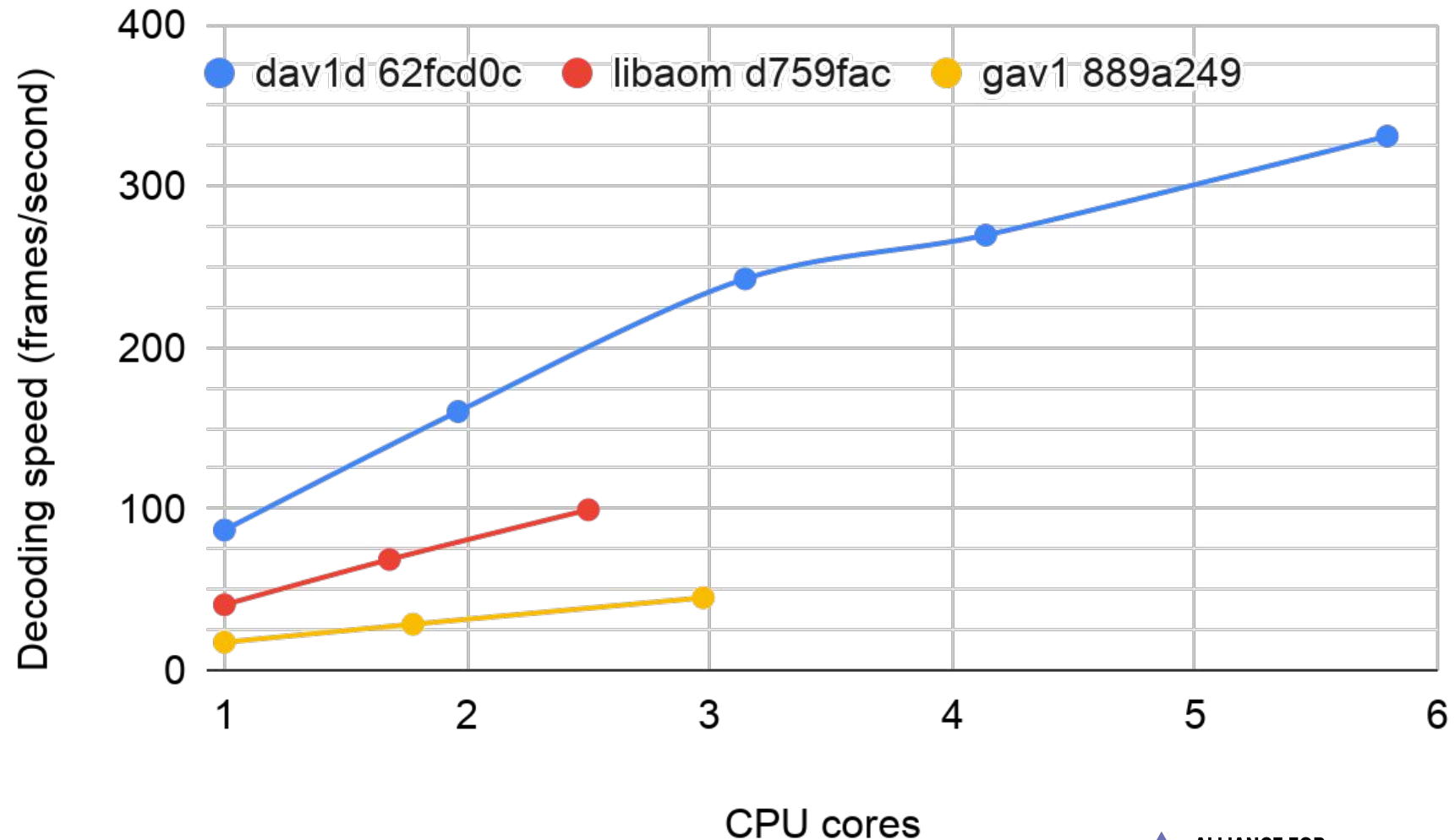


The screenshot shows the GitHub repository page for 'dav1d'. The page features a navigation bar with 'Projects', 'Groups', 'Snippets', and 'Help'. A search bar and 'Sign in / Register' button are also present. The main content area displays the 'dav1d' logo, which consists of the letters 'dav1d' in a bold, black font, with a stylized orange and white traffic cone replacing the letter 'a'. Below the logo, the repository name 'dav1d' is shown, followed by a description: 'dav1d is a new AV1 cross-platform decoder, open-source, and focused on speed and correctness.' The page also includes sections for 'Goal and Features' and 'License'.

<https://code.videolan.org/videolan/dav1d>

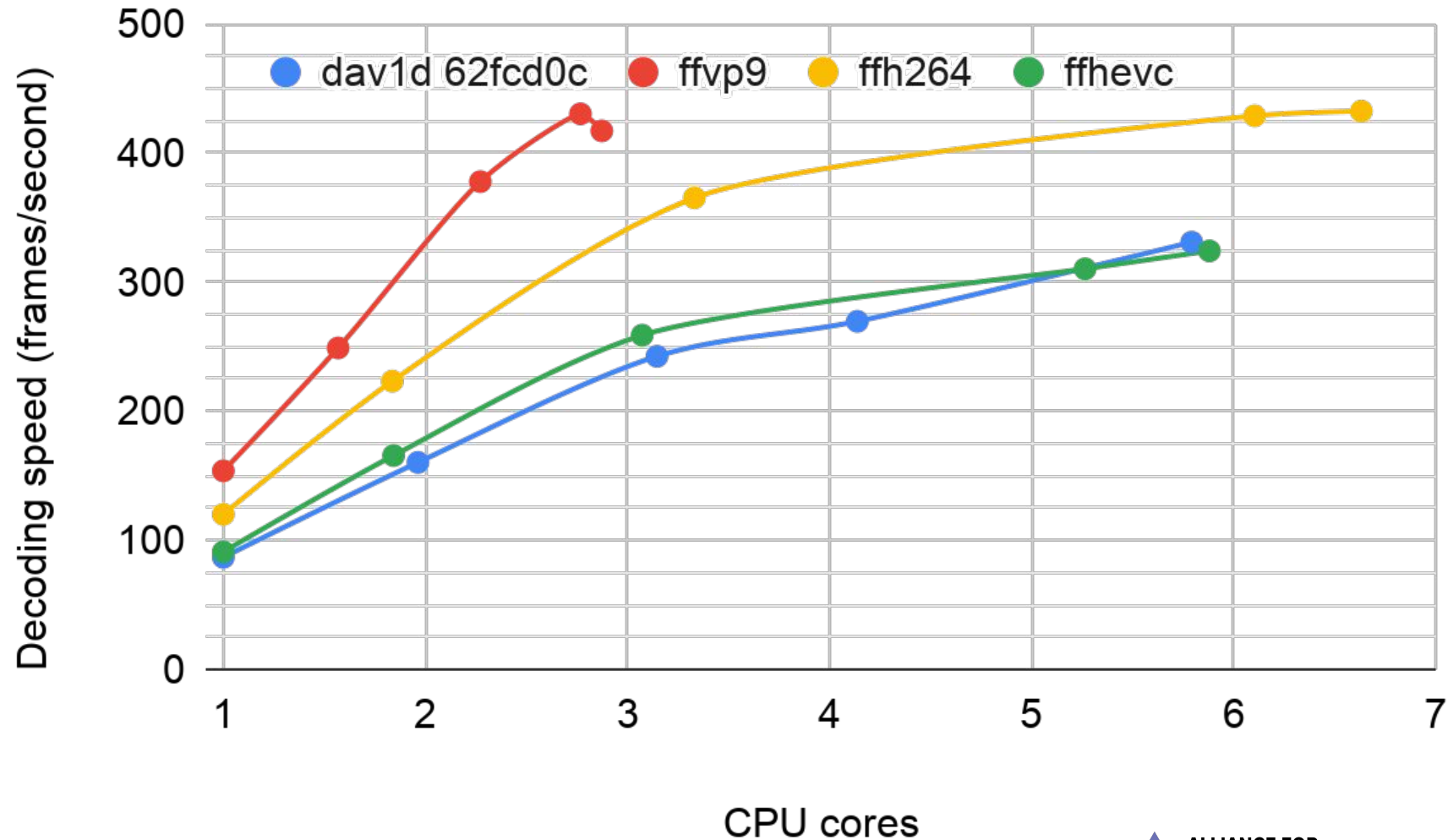
# dAV1d

- 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



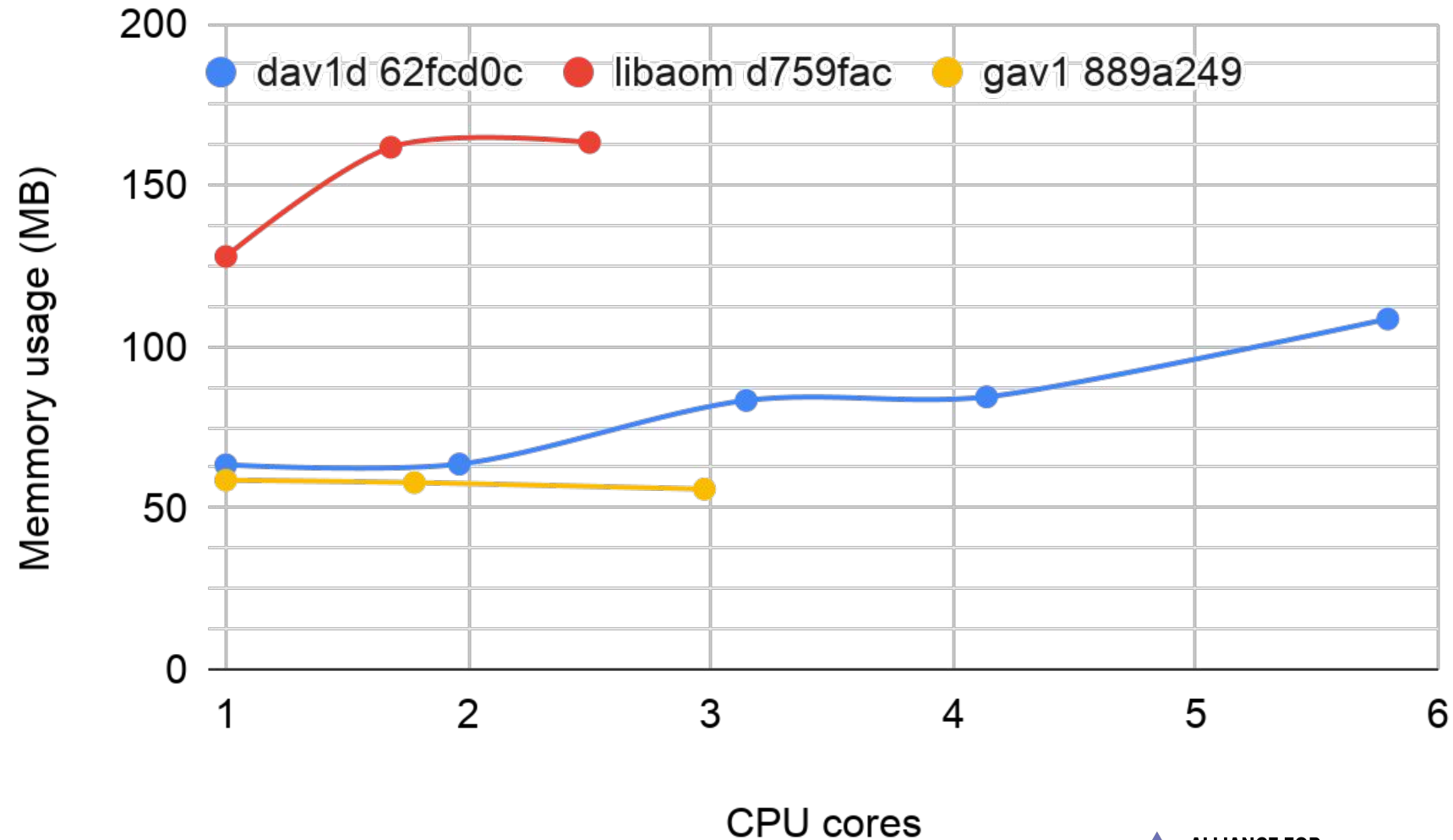
# dAV1d

- Videolan's AV1 decoder
- **Fast & multi-threaded**
  - 2-5x as fast as libaom
  - 4-10x as fast as gavl
  - 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



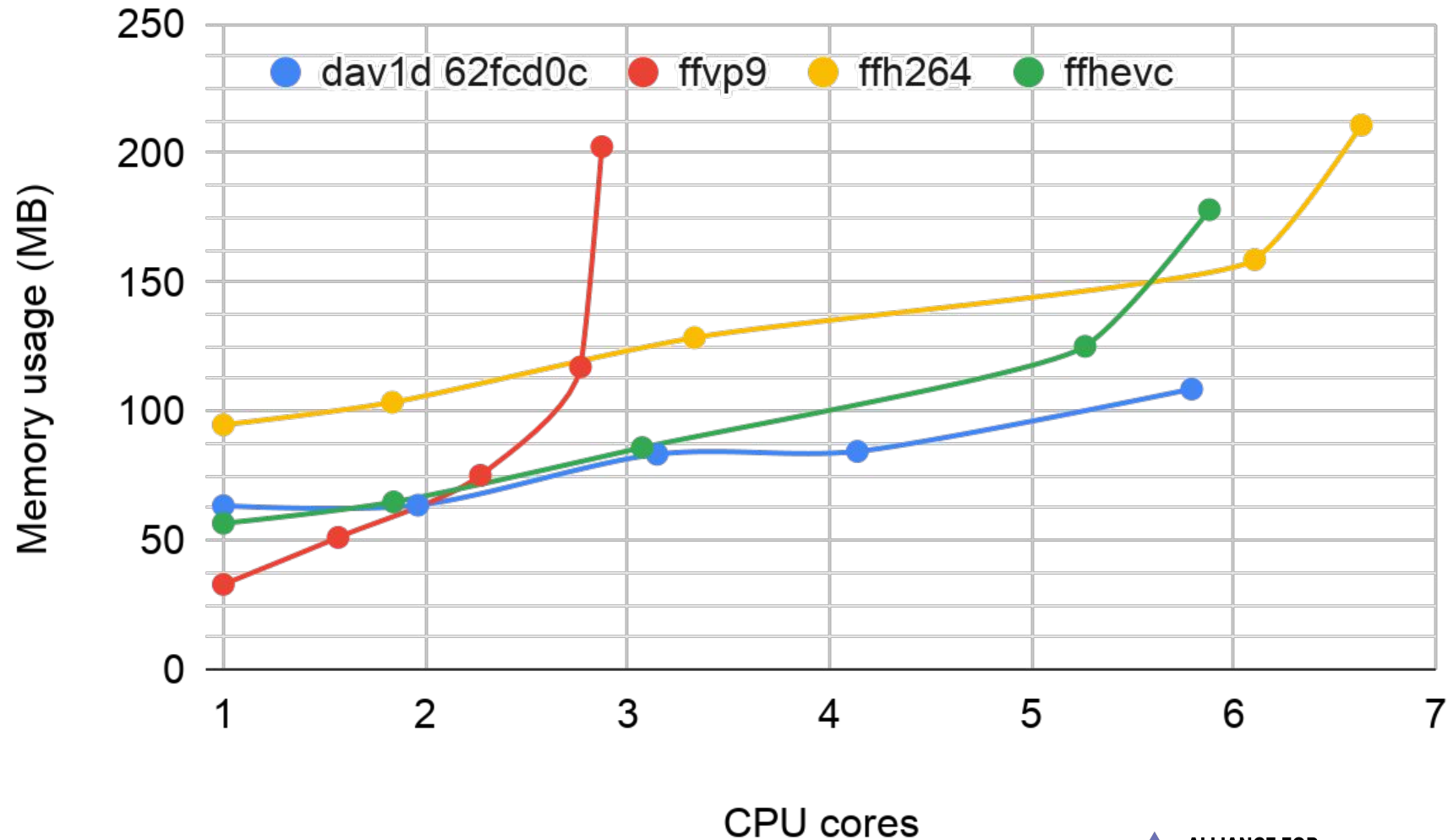
# dav1d

- 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



# dAV1d

- 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)
    - *32bit Neon in progress*
  - 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

*kLOC, decoder only*

	<b>dav1d</b>	<b>libaom</b>	<b>gav1</b>
<b>C/C++</b>	34.6	87.2	45.5
<b>x86 asm</b>	43.1	68.5	15.6
<b>arm asm</b>	18.7	17.2	14.7
<b>ppc asm</b>	1.0	0.3	
<b>mips asm</b>		15.7	



# dAV1d

- Videolan's AV1 decoder
  - Fast & multi-threaded
  - Low memory usage
  - Lean source code
  - **Small binary size**
  - Adoption
- 
- AV1 challenges for decoders

*kB, decoder only*

dav1d	libaom	gav1
926	2936	1461



- 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](#) [♥ Share This](#)

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.

<https://hacks.mozilla.org/2019/05/firefox-brings-you-smooth-video-playback-with-the-worlds-fastest-av1-decoder/>

- VLC 3.1 (April 8)
- Chrome M74 (April 23)
- Firefox 67 (May 14)
- FFmpeg 4.2 (August 5)
- You? (soon!)



- Videolan's AV1 decoder
  - Fast & multi-threaded
  - Low memory usage
  - Lean source code
  - Small binary size
  - Adoption
- 
- **AV1 challenges for decoders**

- 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



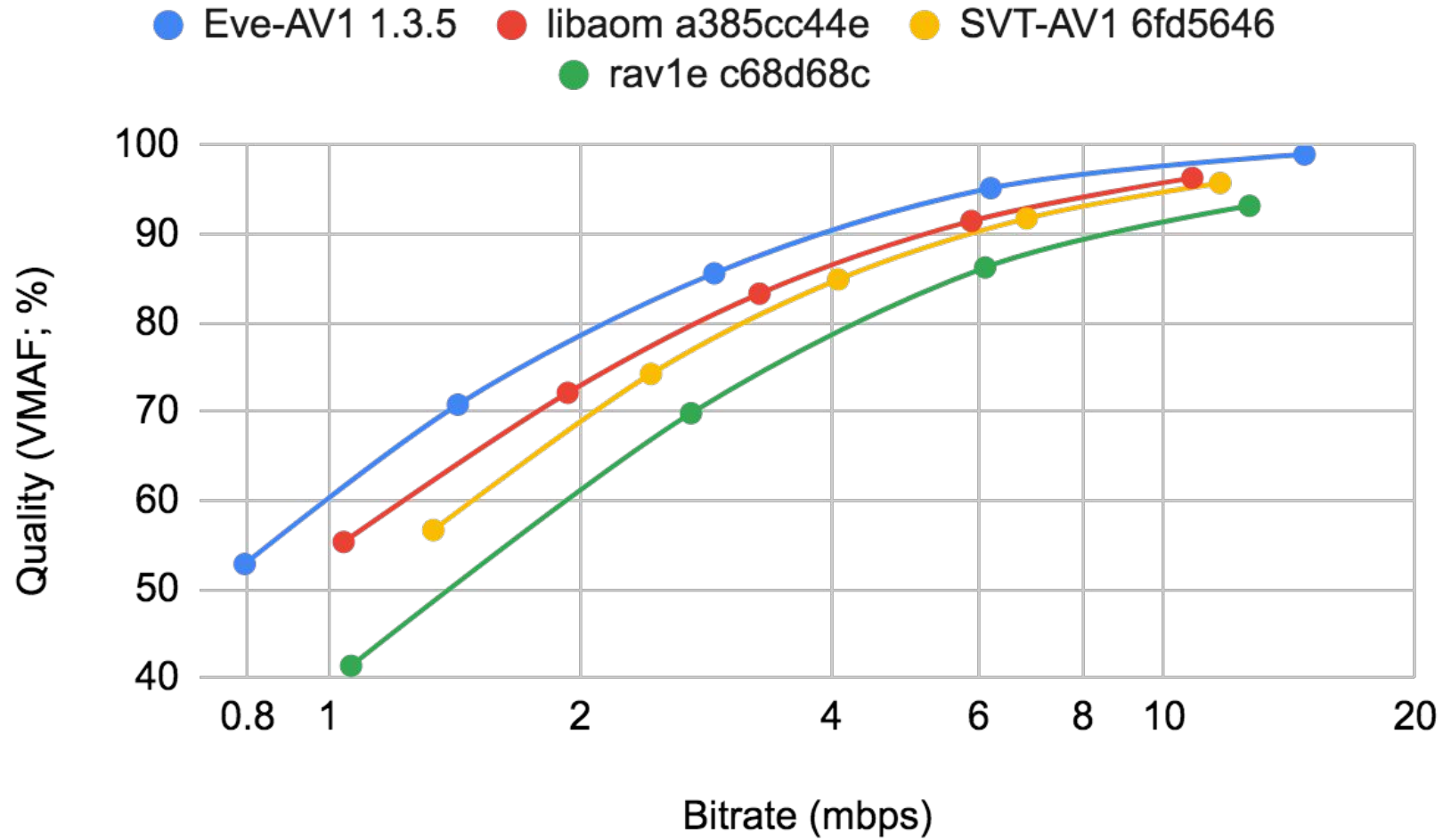
<https://twoorioles.com/>





# Eve-AV1

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





libaom

3mbps

Eve-AV1







## Eve-AV1

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

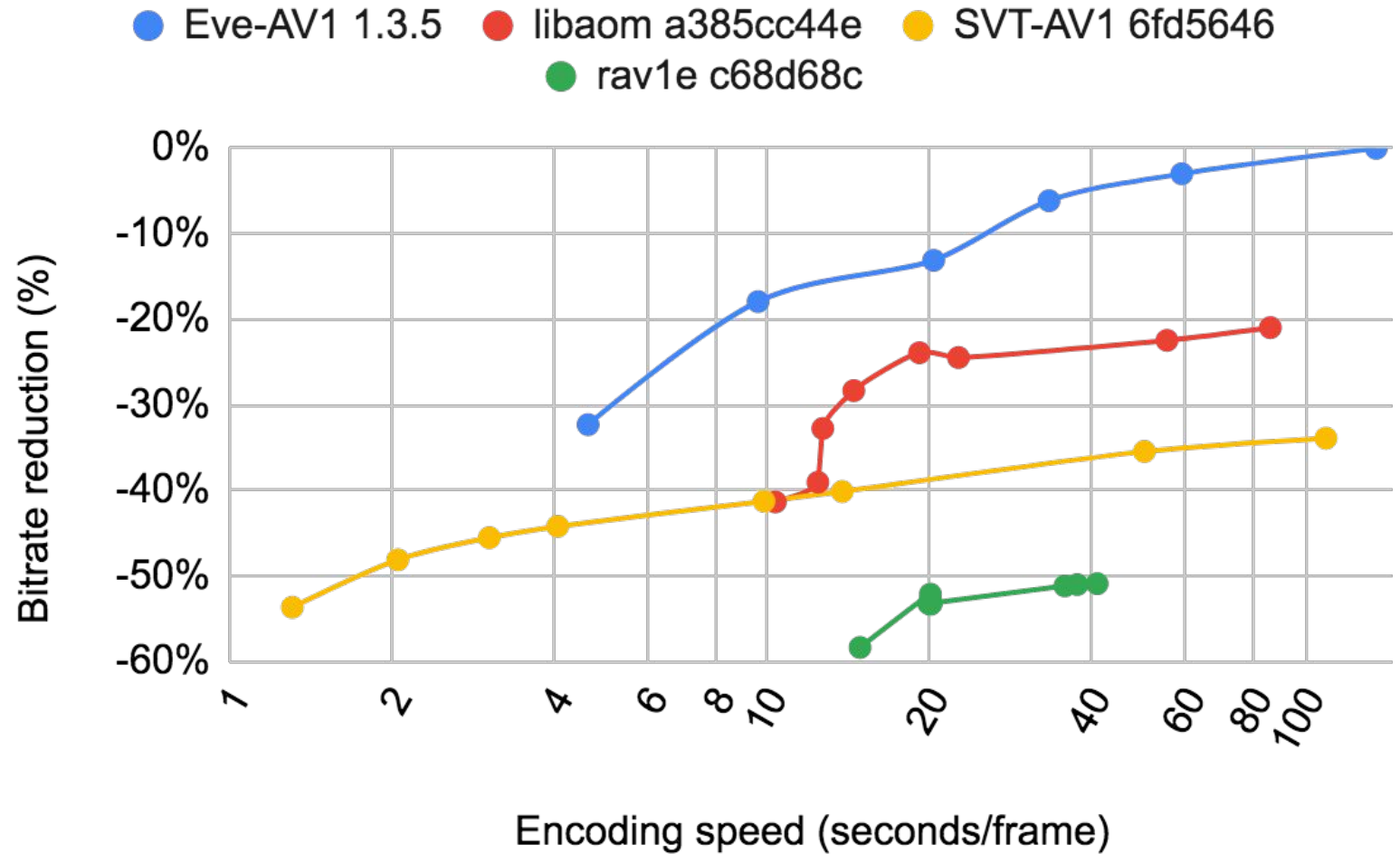
*1080p clips*

	% Bitrate reduction	Runtime (sec/frame)
Eve-AV1 1.3.5	0.00%	135.57
libaom a385cc44e	-20.95%	86.13
rav1e c68d68c	-50.88%	41.01
SVT-AV1 6fd5646	-33.88%	109.29



# Eve-AV1

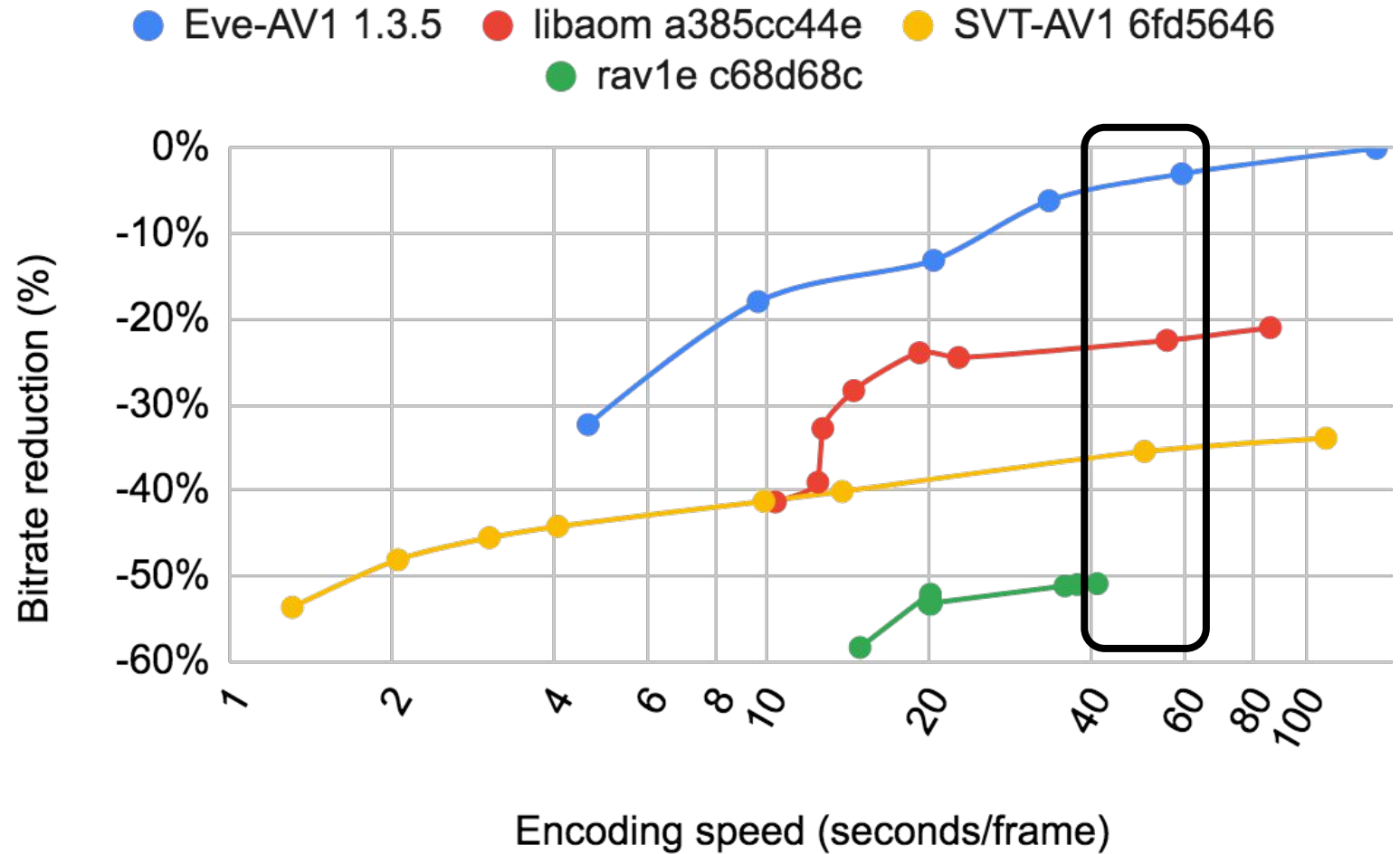
- Two Orioles' AV1 encoder
- Quality vs. Bitrate
- **Quality-per-bit vs. Speed**
- Multi-threading
  
- AV1 challenges for encoders





# Eve-AV1

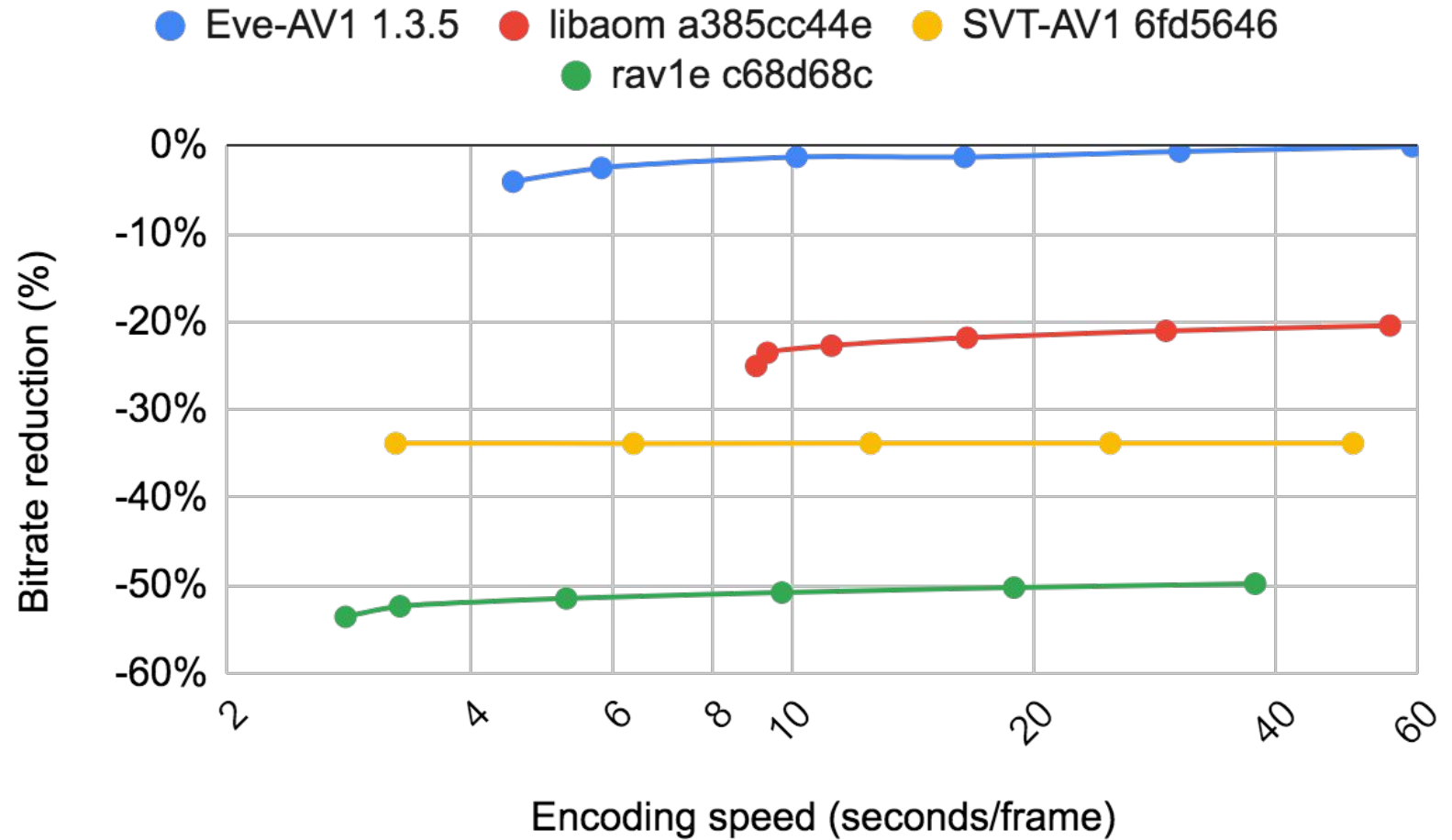
- Two Orioles' AV1 encoder
- Quality vs. Bitrate
- **Quality-per-bit vs. Speed**
- Multi-threading
  
- AV1 challenges for encoders





# 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**

- **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?)

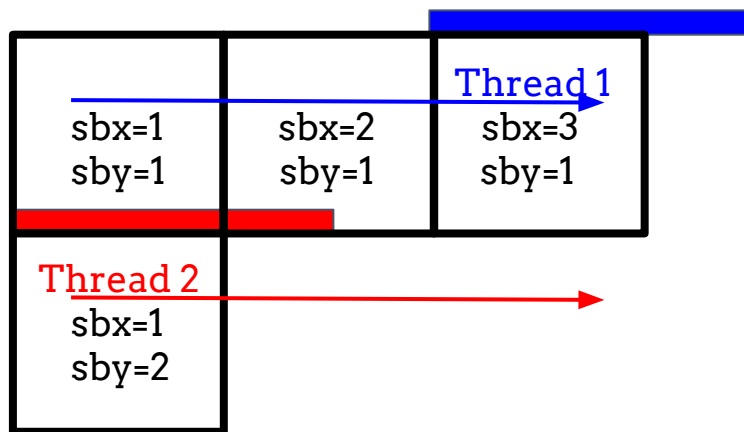
- **MT encoder models for AV2?**



## 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**

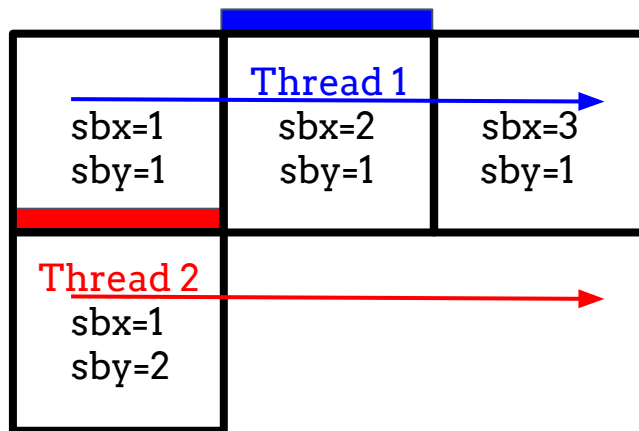
- **Limit top/right edge access at SB corners**
- increasing LRU size gives significant coding gains, but increases delay
  - Allow rectangular LRUs ( $w > h$ )?
- CDEF Us overhang deblocked SB row boundaries (but LRUs do not?)
- MT encoder models for AV2?



## 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?**



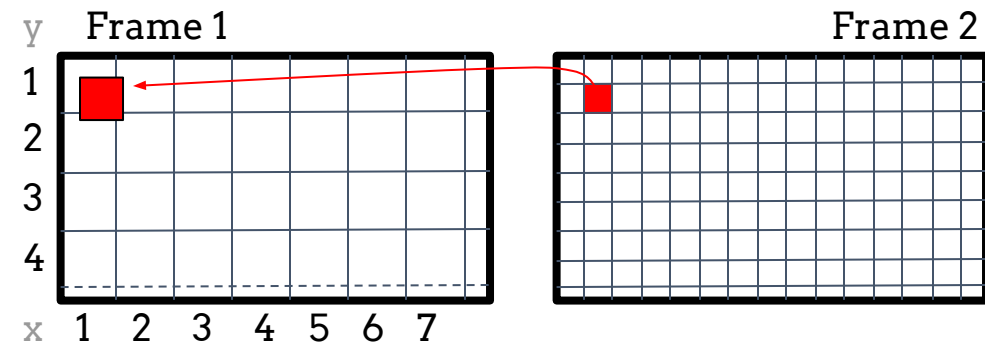


## Eve-AV1

- 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?

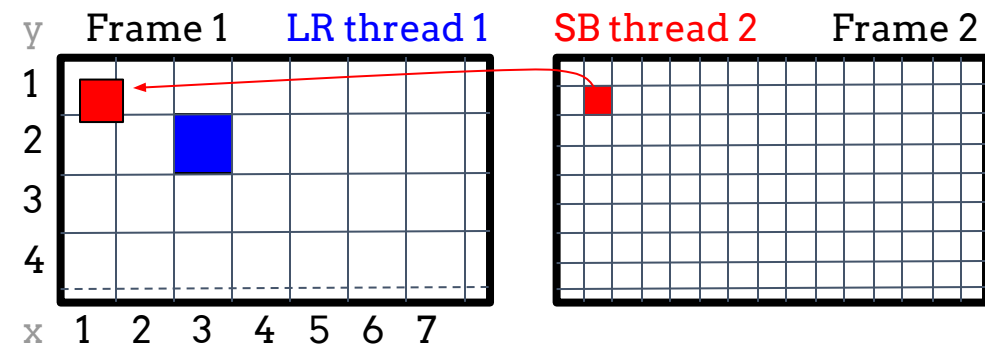


## Eve-AV1

- 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?



## Eve-AV1

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

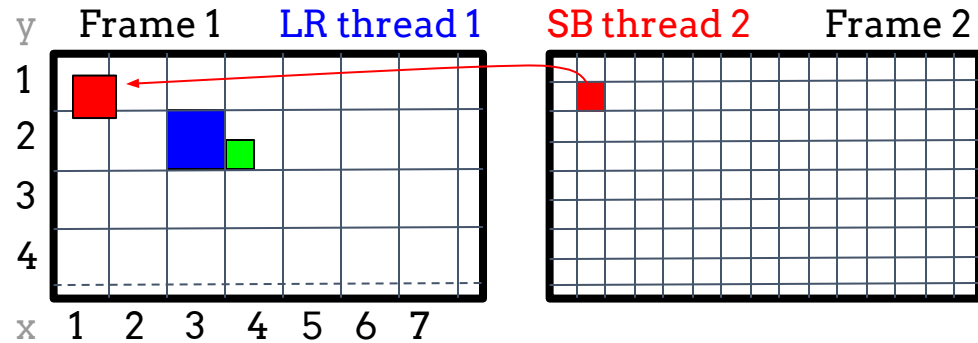
- **AV1 challenges for encoders**

1920x1080 frame | 128x128 SBs | 256x256 LRUs

SB thread 3

LR thread 1

SB thread 2



- 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?



## Eve-AV1

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

- **AV1 challenges for encoders**

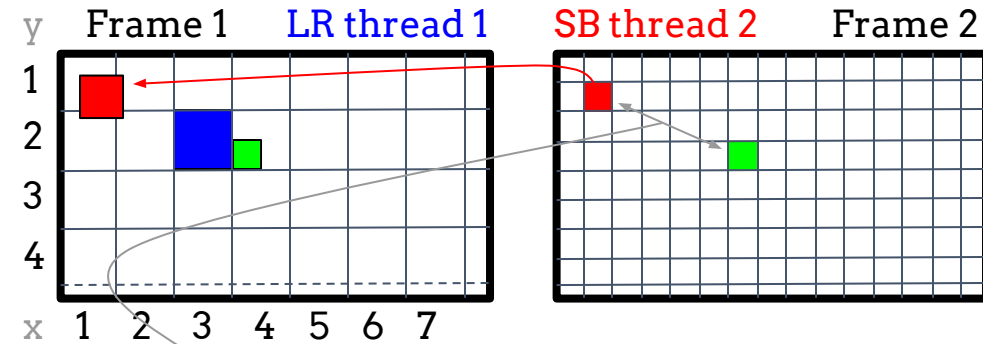
1920x1080 frame | 128x128 SBs | 256x256 LRUs

SB thread 3

LR thread 1

SB thread 2

Frame 2



SB distance (thread 2-3): 35, nSBs: 15\*8.5

Max. concurrency:  $15 \cdot 8.5 / 35 = 3.6$

- Tools

- So many (coding & code complexity)
- $O(x^n)$  vs.  $O(x \cdot n)$  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?



# Eve-AV1

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

- **AV1 challenges for encoders**

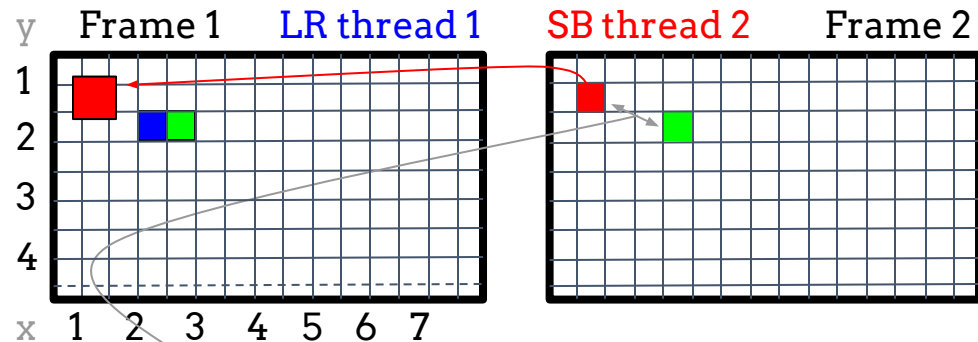
1920x1080 frame | 128x128 SBs | 128x128 LRUs

SB thread 3

LR thread 1

SB thread 2

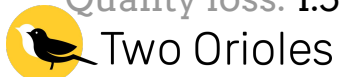
Frame 2



SB distance (thread 2-3): 18, nSBs: 15\*8.5

Max. concurrency: 15\*8.5/18 = 7.1

Quality loss: 1.5%



- 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?



# Eve-AV1

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

- **AV1 challenges for encoders**

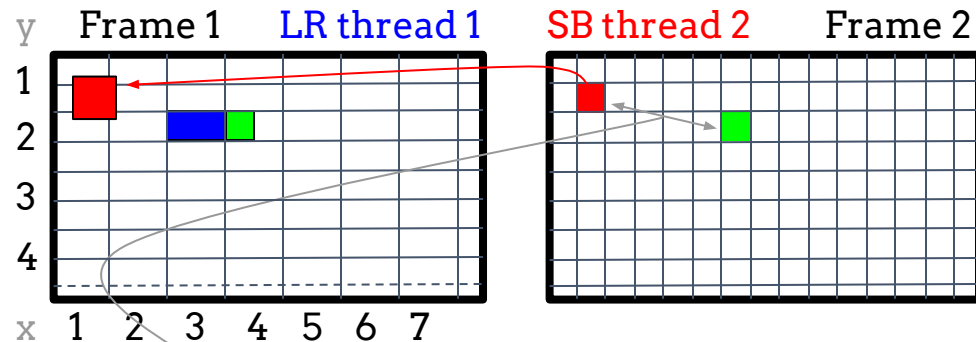
1920x1080 frame | 128x128 SBs | 256x128 LRUs

SB thread 3

LR thread 1

SB thread 2

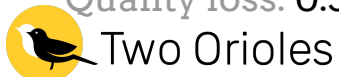
Frame 2



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

Max. concurrency:  $15 \cdot 8.5 / 20 = 6.4$

Quality loss: 0.5%



- Tools

- So many (coding & code complexity)
- $O(x^n)$  vs.  $O(x \cdot n)$  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?



## 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?)

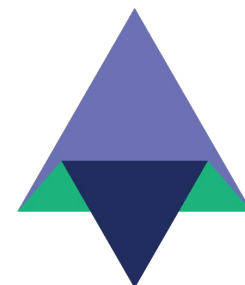
- **MT encoder models for AV2?**



# Questions?



Ronald S. Bultje <[rbultje@twoorioles.com](mailto:rbultje@twoorioles.com)>  
Founder, Two Orioles



**ALLIANCE FOR  
OPEN MEDIA  
RESEARCH**

**Symposium 2019**