Enhancing SVT-AV1 with LCEVC to improve quality-cycles trade-offs and enhance sustainability of VOD transcoding

Guendalina Cobianchi,
Guido Meardi,
Stergios Poularakis,
Alexei Walisiewicz

V-Nova

Omran Abdelkafi,
Foued Ben Amara,
Faouzi Kossentini

Intel

Ioannis Katsavounidis,
Cosmin Stejerean,
Hassene Tmar

Meta

San Diego, August 23rd 2022

Contents

- Motivation
- Latest improvements in the SVT-AV1 encoder
- Introduction to LCEVC
- Quality-cycles trade-offs evaluation using two Dynamic Optimizer methods:
  - Restricted discrete convex hull approach
  - Fast encoding parameter selection
- LCEVC AV1 decoding power and battery consumption
  - Decoder CPU efficiency results
  - Real-life results on Android devices
- Conclusions

Motivation

Background

- Explosive growth in the generation, storage and transmission of video - trade-offs between compression and complexity are key in software video encoding, even more so with increasing pressure on sustainability
- SPIE 2021 work* described three approaches of evaluating compression efficiency vs cycles trade-offs using the Dynamic Optimizer (DO) framework for VOD applications
- The new video codec enhancer LCEVC (Low Complexity Enhancement Video Coding) standardized in October 2021 as MPEG-5 Part 2

The challenge

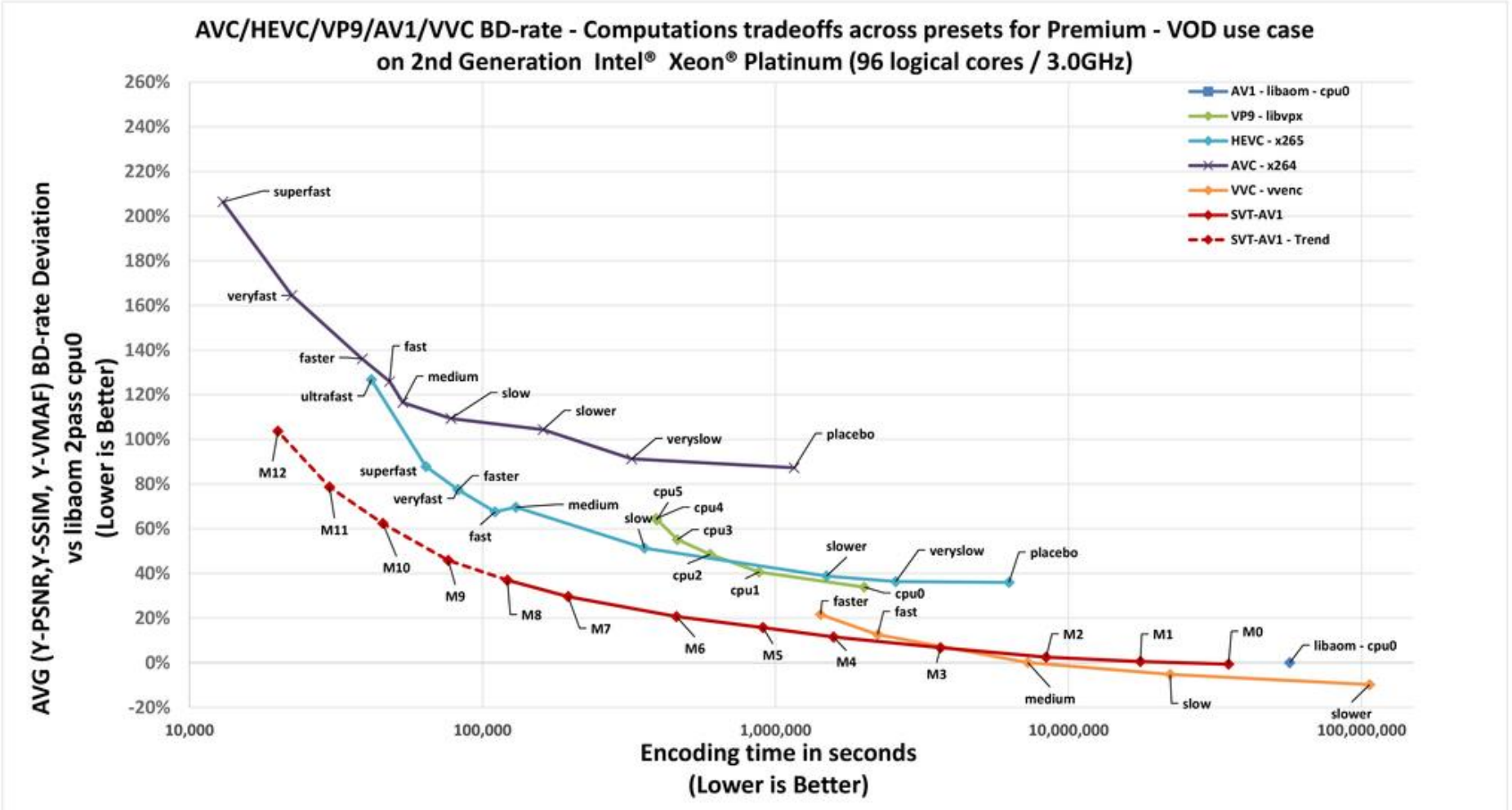
- Bring the cost of VOD processing further down while increasing the compression efficiency
- As of early 2022, only a few devices have HW accelerated AV1 decoding: it could be years until global consumers can decode high-quality HD AV1 content on their mobiles at scale
- Battery power consumption for software AV1 decoders on mobile devices is often unsustainable

Proposed solution

- Combining the Dynamic Optimizer best practices, the latest improvements in the SVT-AV1 encoder and MPEG-5 LCEVC to further improve the compression efficiency vs cycles trade-offs
- Leverage LCEVC+AV1 efficient decoding to expand the range of mobile devices capable of playing HD with AV1 and to extend mobile battery life vs. state-of-the-art AV1 software decoding

* SPIE-118420T (2021), Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications, Ping-Hao Wu, Ioannis Katsavounidis, Zhijun Lei, David Ronca, Hassene Tmar, Omran Abdelkafi, Colton Cheung, Foued Ben Amara, Faouzi Kossentini

SPIE 2021: Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications



Source: SPIE-118420T (2021), Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications, Ping-Hao Wu, Ioannis Katsavounidis, Zhijun Lei, David Ronca, Hassene Tmar, Omran Abdelkafi, Colton Cheung, Foued Ben Amara, Faouzi Kossentini

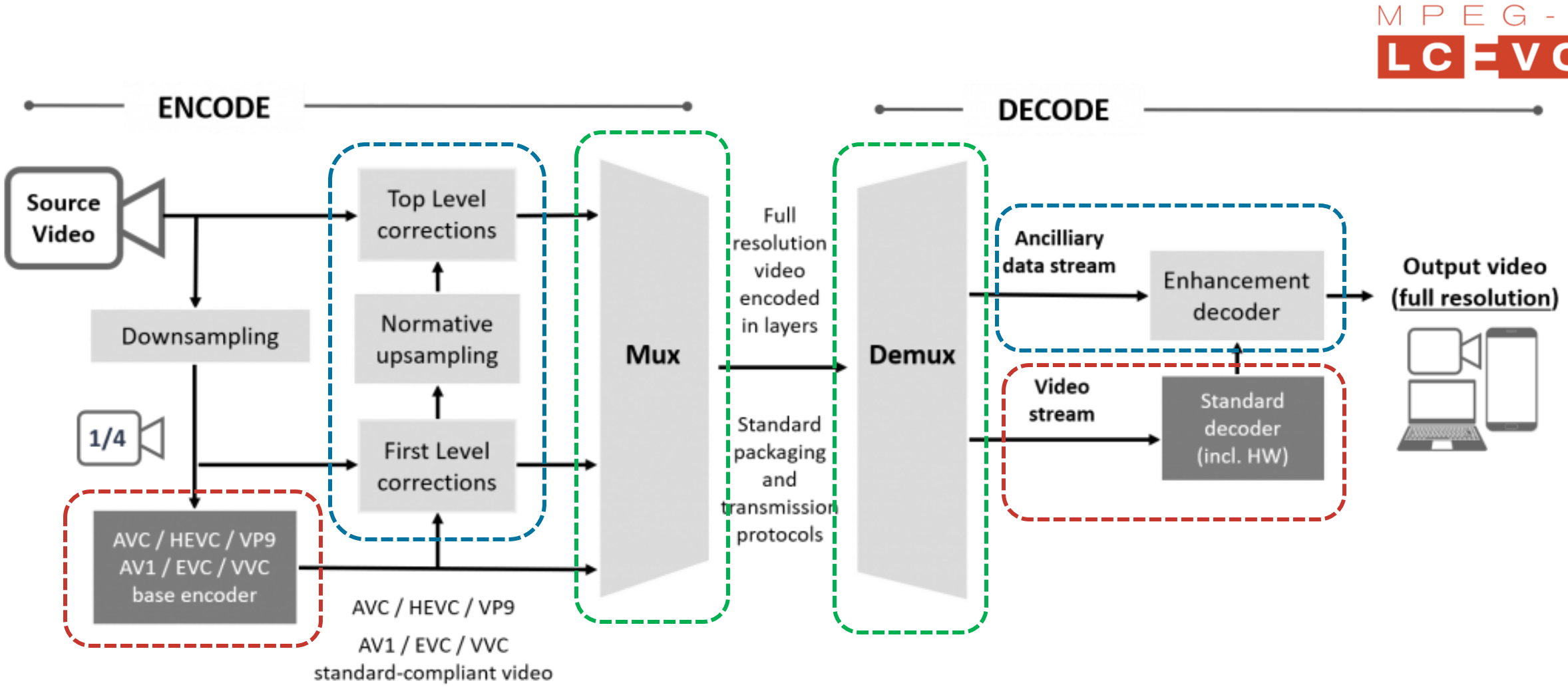
Latest improvements in the SVT-AV1 encoder

Faster encoder presets	• Four new fast presets M9-M12 added to extend the range of complexity levels to those of the medium/fast presets of x264 while maintaining a BD-rate advantage
Improved quality complexity trade-offs	• Improved the BD-rates versus cycles trade-offs for all presets vs. those of other open-source encoders in high-latency VOD applications
Subjectively-tuned encoder mode	• Addresses some subjective quality artifacts observed when using the default mode, improvements include higher sharpness and less blockiness
Support for new tools	• Super-resolution frames, overlay frames, S frames and low-delay prediction structures
Rate control improvements	• Improvements in the one-pass VBR rate control and in the CBR rate control algorithm using low-delay prediction structures
Decoder friendly encoder optimizations	• Reduce the computational cost of decoding on M0-M10 presets, while maintaining a minimal impact on encoder's BD-rate. Optimizations include: lighter Motion Field Motion Vector prediction, Overlapped Block Motion Compensation, warped motion prediction, deblocking filter, Constrained Directional Enhancement Filter, and the Self-guided filter with subspace projection

* SPIE-118420T (2021), Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications, Ping-Hao Wu, Ioannis Katsavounidis, Zhijun Lei, David Ronca, Hassene Tmar, Omran Abdelkafi, Colton Cheung, Foued Ben Amara, Faouzi Kossentini

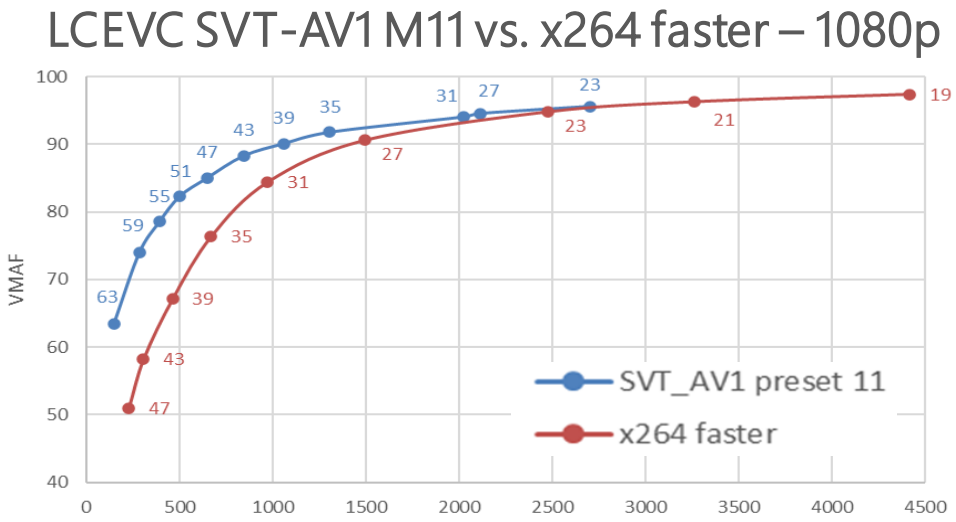
Introduction to MPEG-5 LCEVC

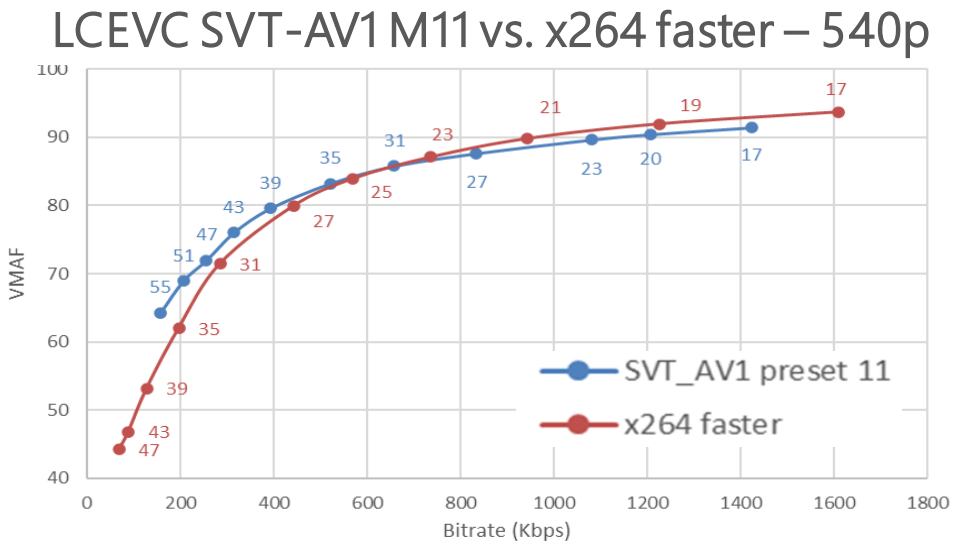
LCEVC (Low Complexity Enhancement Video Coding): Not "one more codec", but a codec-agnostic enhancer



LCEVC-Enhanced SVT-AV1 – optimization opportunities

- LCEVC uses the base codec at low resolution/ low QPs, hence the overall coding efficiency relies on base codec calibration at those points
- Tests on SVT-AV1 (M11) compared against x264 (faster, which has comparable encode time) showed that current SVT-AV1 is not yet optimally calibrated for low-QP/low-resolution operating points
- Intel's SVT-AV1 team developed a patched version* with experimental changes to M11 resulting in 3% BD-rate uplift and reduction in encoding time, resulting in a better BD-rate-computation trade-off
- Similar improvements for presets below M11 are possible, leaving further optimizations of low-QP low-resolution operating points as a potential upside for future releases





Quality-cycles trade-offs evaluation

Encoders used in the comparison

Open-source implementations of different video coding standards

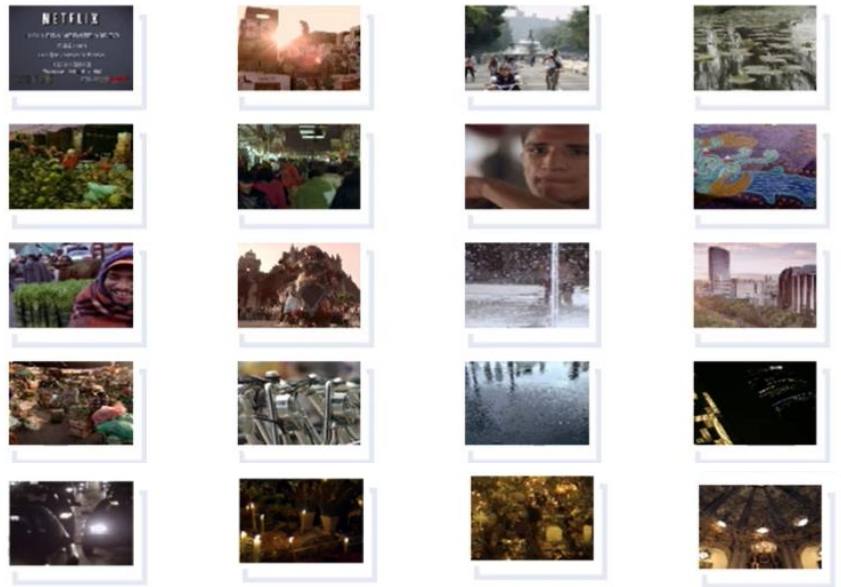
- AVC: x264 - https://code.videolan.org/videolan/x264
- HEVC: x265 - https://www.videolan.org/developers/x265.html
- AV1:
  - SVT-AV1 (SPIE_patch): https://gitlab.com/AOMediaCodec/SVT-AV1/-/tags/v1.1-SPIE2022
  - Libaom (exclusively as a reference for BD-rate): https://aomedia.googlesource.com/aom/

LCEVC-enhanced implementations

- LCEVC AVC: LCEVC x264
- LCEVC HEVC: LCEVC x265
- LCEVC AV1: LCEVC SVT-AV1 (SPIE_patch)

Description of the Test sets

- **Netflix El Fuente* (Premium VOD):** total length: 7min 52sec. / 29.97 fps; 140 shots, 2-5 sec. each
- Resolutions: 1080p to 144p (8 resolutions)
- QPs/CRFs: 11 values, to cover the full range of bitrates/qualities relevant for adaptive streaming
- Presets (native and LCEVC-enhanced):
  - x264/x265: veryfast, faster, fast, medium, slow, slower, veryslow
  - SVT-AV1: M11 to M1
  - Libaom (native AV1): cpu-used=0, as anchor for BD-rate calculation
- Metrics: the open source *vmaf* application was used to collect the quality metrics: VMAF, VMAF_NEG, Y-PSNR, Y-SSIM
- The linux time utility (`/usr/bin/time -v`) was used to collect the total encoding time per executed command line
- Complexity is the sum of the user time and the system time



* Open source El Fuente data set: https://opencontent.netflix.com/#h.goi1q35qkqlw

Restricted Discrete DO approach*

Main idea:

- Combined test sequence: conceptually the concatenation of all the 140 shots
- Generate a convex hull for the combined test sequence to be used in the BD-rate calculation
- The quality range is restricted to reflect common quality levels used in adaptive streaming applications
- The number of points used to compute the BD-rate data is limited to a small number (e.g., 8)

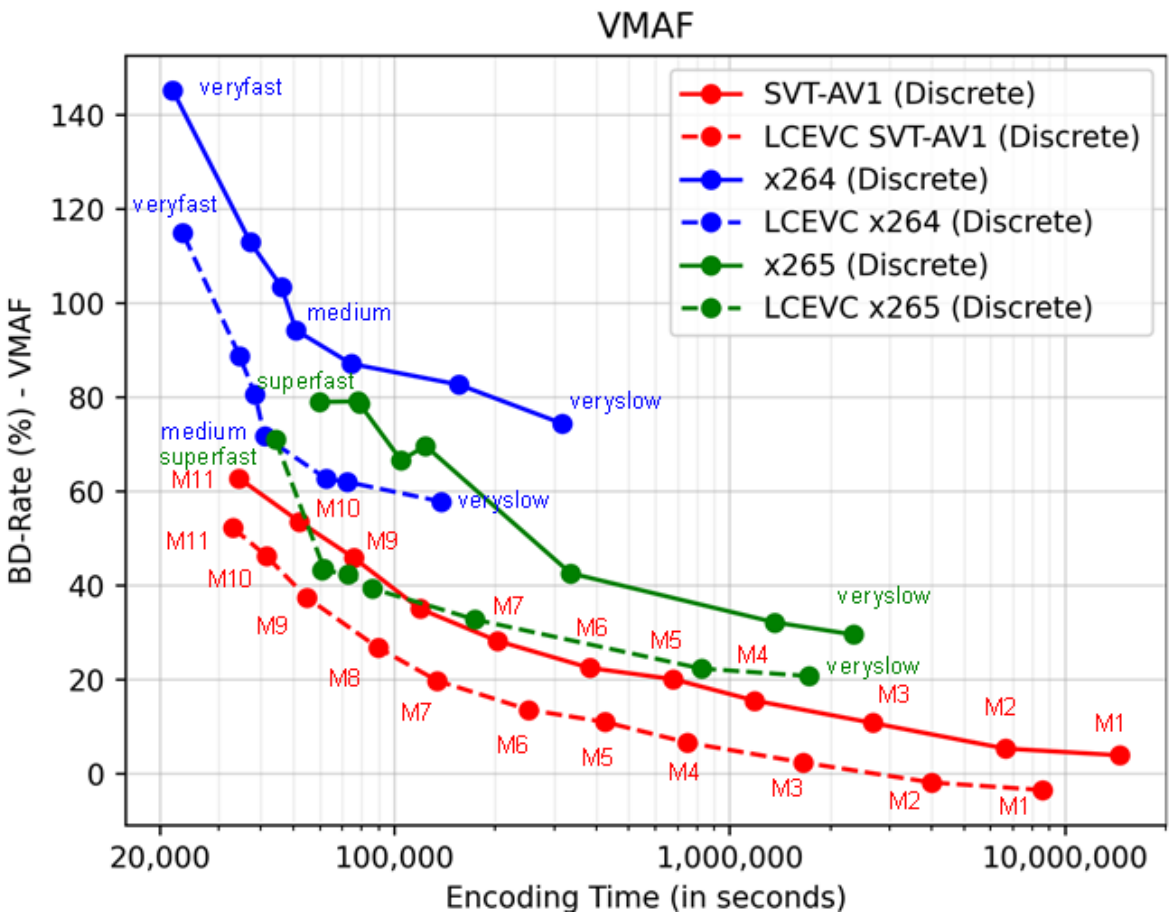
Main steps:

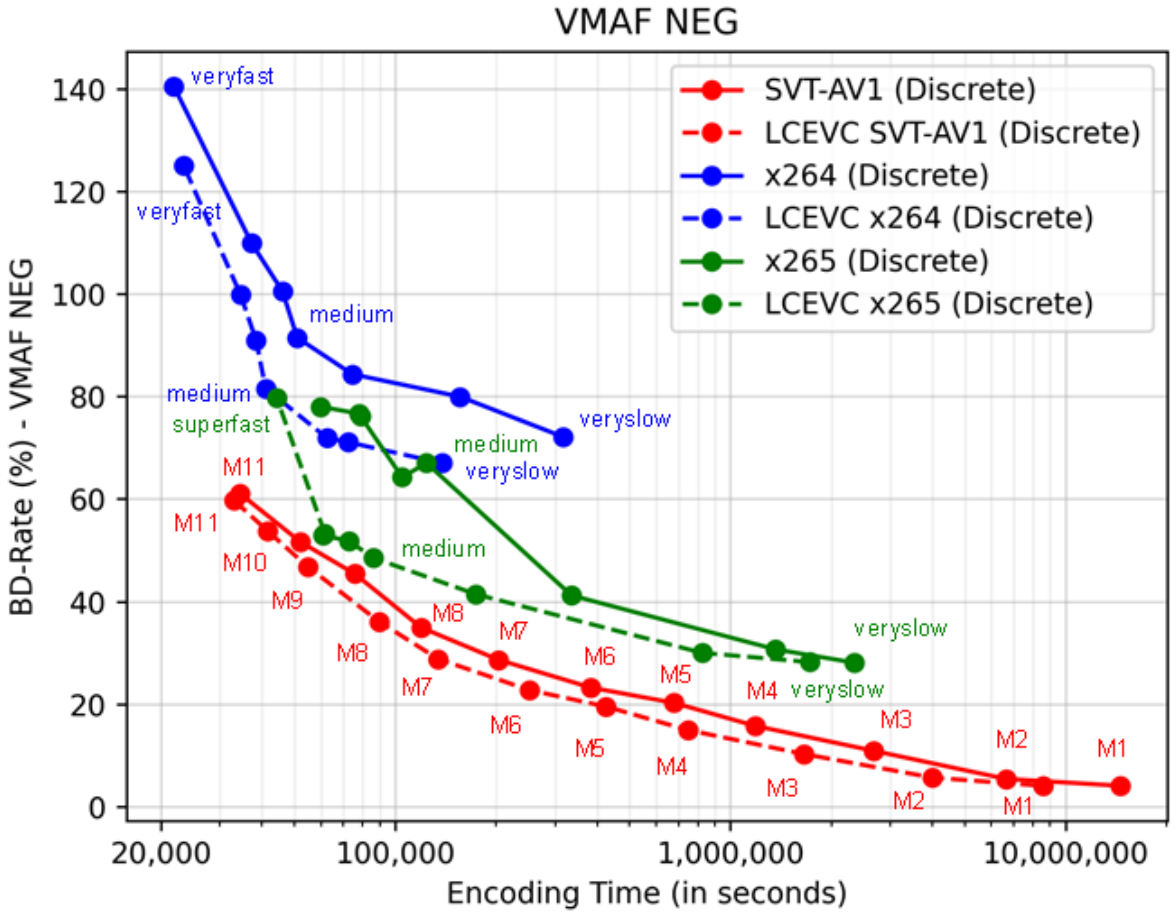
- Use VMAF combined convex hull for the combined sequence to identify bitrates corresponding to VMAF values of 30, 40, 50, 60, 70, 80, 90, 95
- Use the resulting bitrates to identify eight points on each of the VMAF_NEG, PSNR, SSIM combined convex hulls for the combined sequence
- Compute the encoder preset BD-rate based on the selected eight points

* Full methodology description in SPIE-118420T (2021), Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications, Ping-Hao Wu, Ioannis Katsavounidis, Zhijun Lei, David Ronca, Hassene Tmar, Omran Abdelkafi, Colton Cheung, Foued Ben Amara, Faouzi Kossentini

Results: Restricted Discrete DO Approach (1/2)

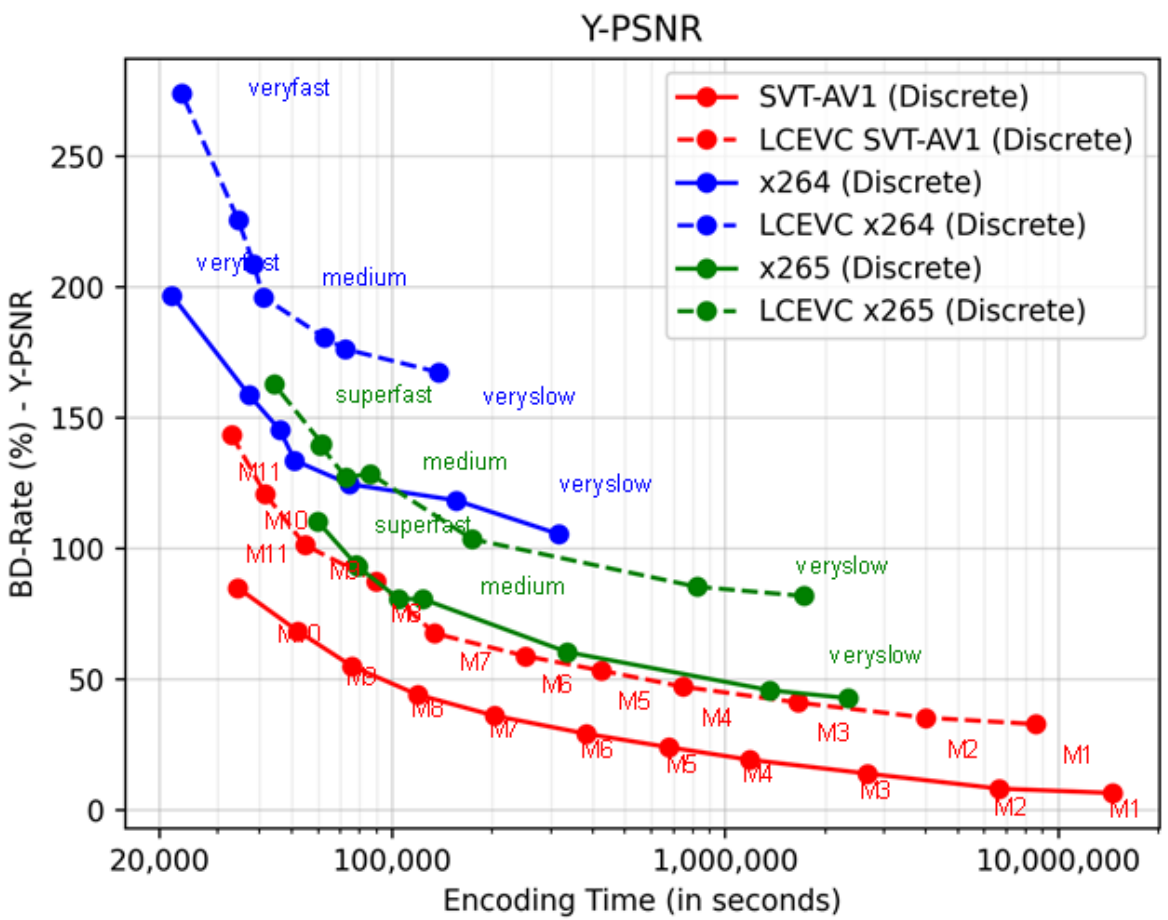
BD-rate – Computations trade-off across presets, restricted discrete DO

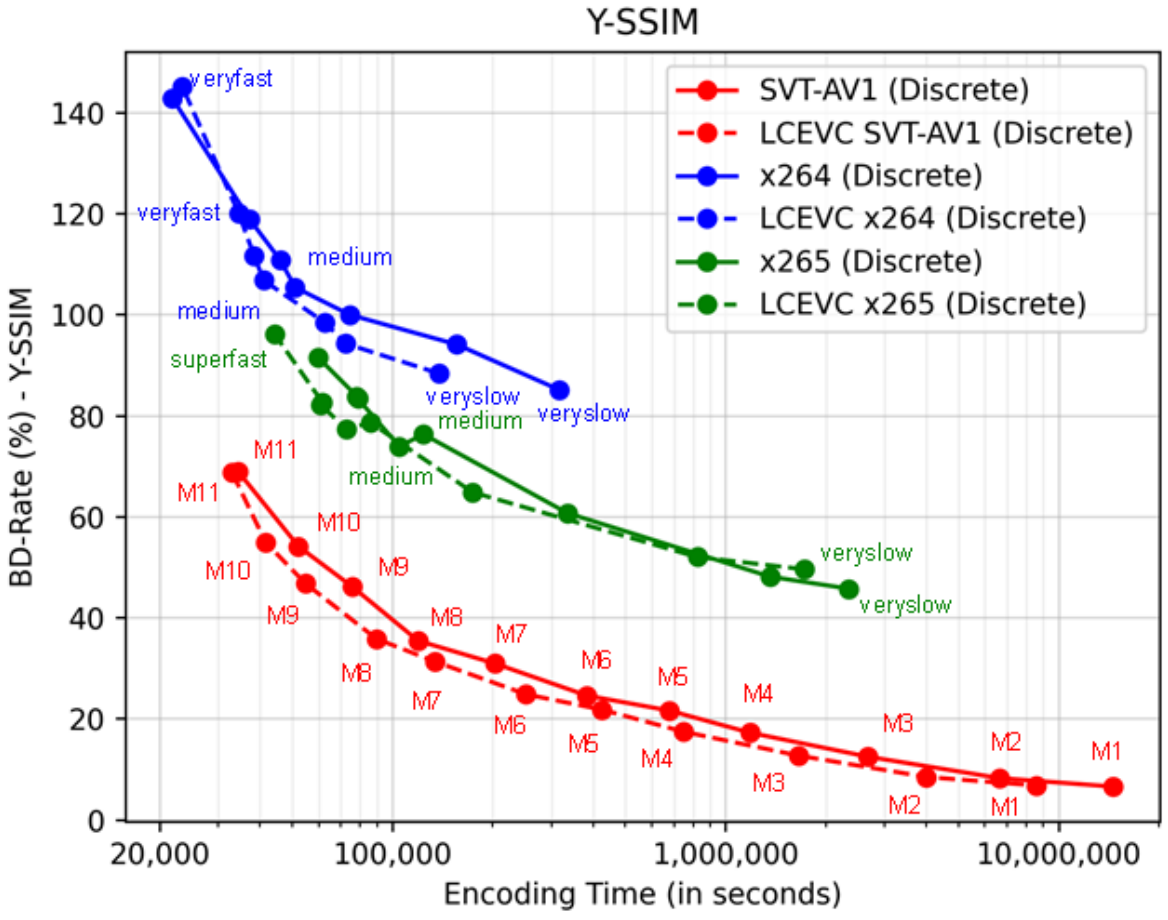




Results: Restricted Discrete DO Approach (2/2)

BD-rate – Computations trade-off across presets, restricted discrete DO





Fast Encoding parameter selection approach*

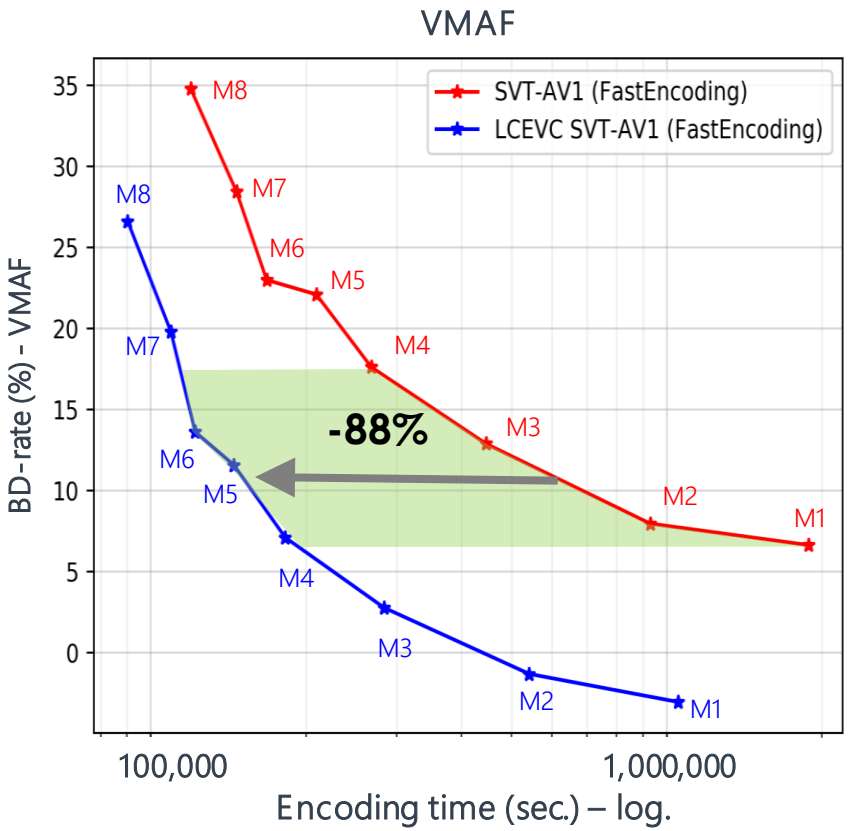
Main idea: same as the restricted discrete convex hull approach, except that:

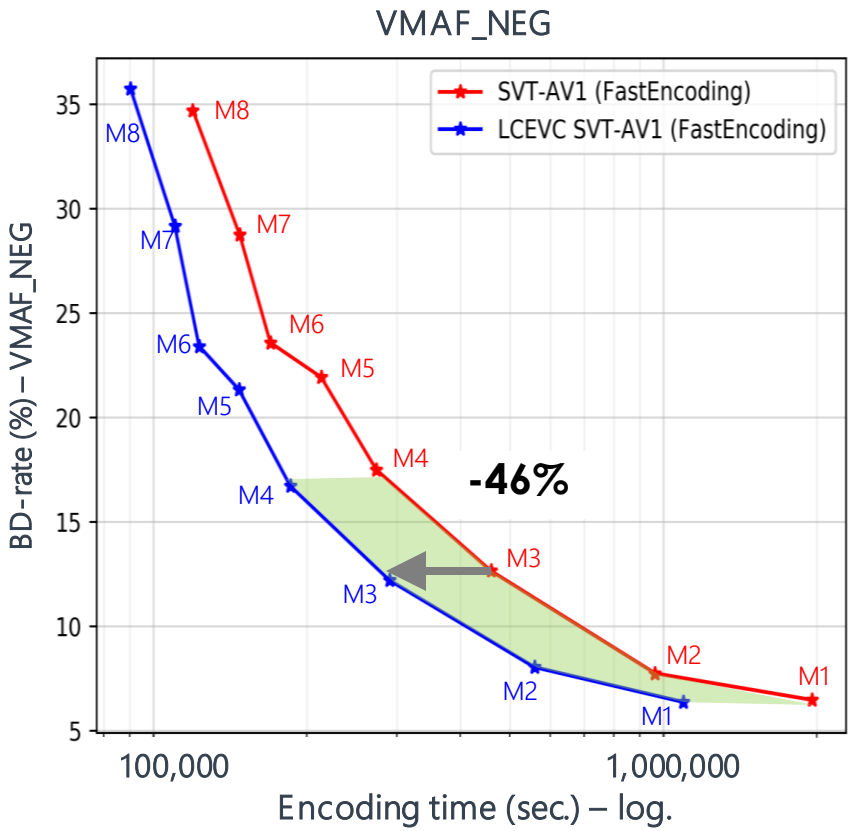
- A fast encoder preset for SVT-AV1 and LCEVC SVT-AV1 (e.g., M8) is used in all steps up to the identification of the eight points on the convex hull
- Final encodings corresponding to each of the eight selected points are regenerated using the desired encoder preset (e.g., M1)
- The generation of the encoder BD-rate data is based on the final encodings
- The complexity is the sum of the complexity measures associated with the two encoding steps above
- To further expand the SPIE_2021 methodology, we also computed the optimal combination of the preset used as a reference and the preset used for the actual encoding

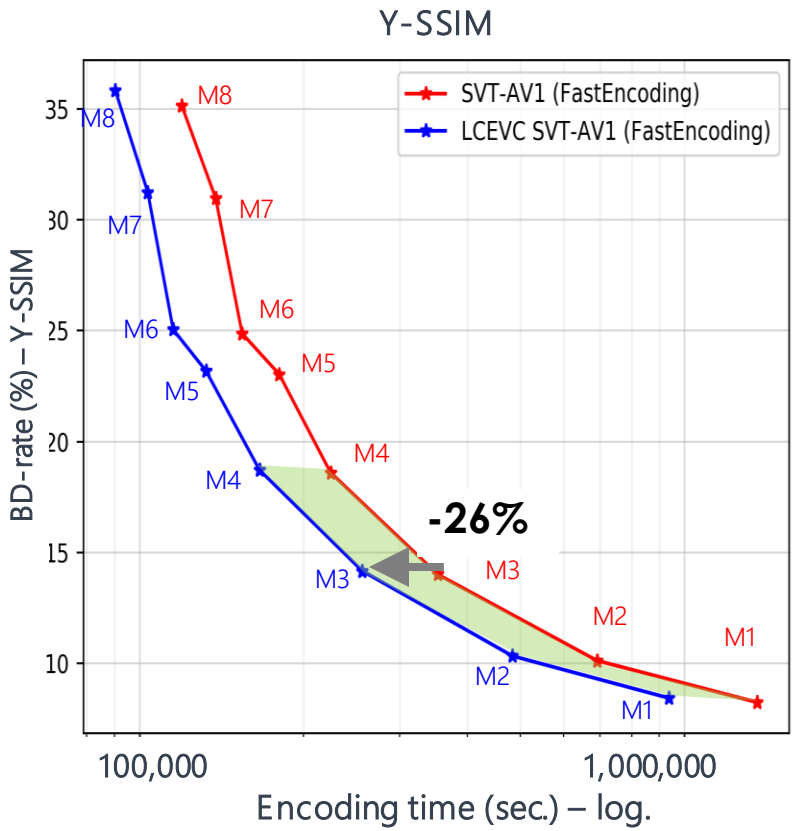
* Full methodology description in SPIE-118420T (2021), Towards much better SVT-AV1 quality-cycles tradeoffs for VOD applications, Ping-Hao Wu, Ioannis Katsavounidis, Zhijun Lei, David Ronca, Hassene Tmar, Omran Abdelkafi, Colton Cheung, Foued Ben Amara, Faouzi Kossentini

Results: Fast Encoding parameter selection (M8 as reference for prediction)

BD-rate – Computations trade-off across presets, Fast Encoding parameter selection

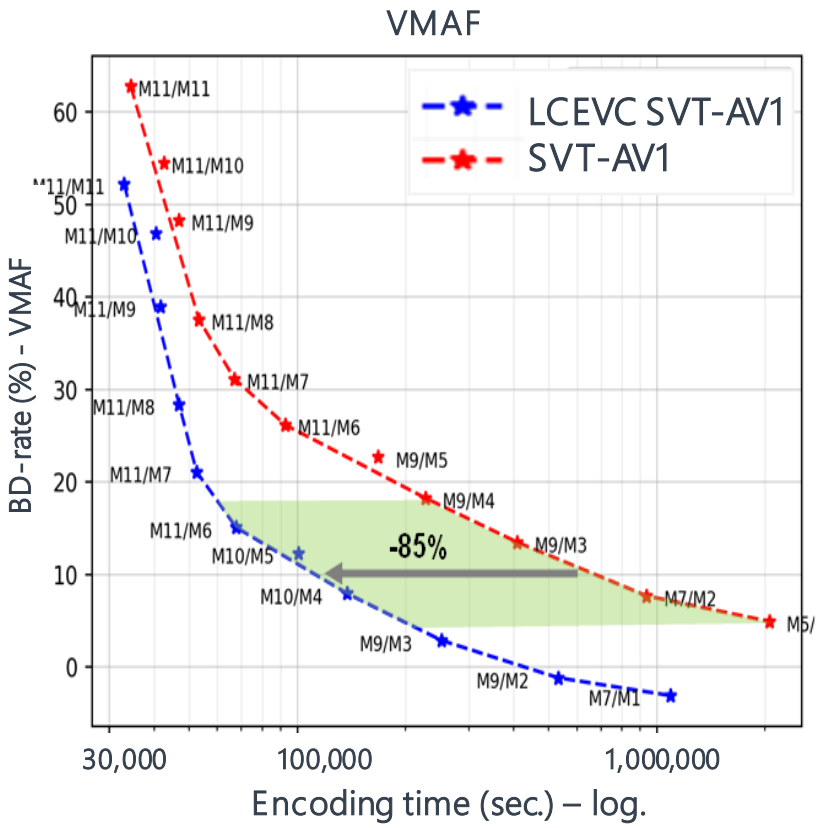


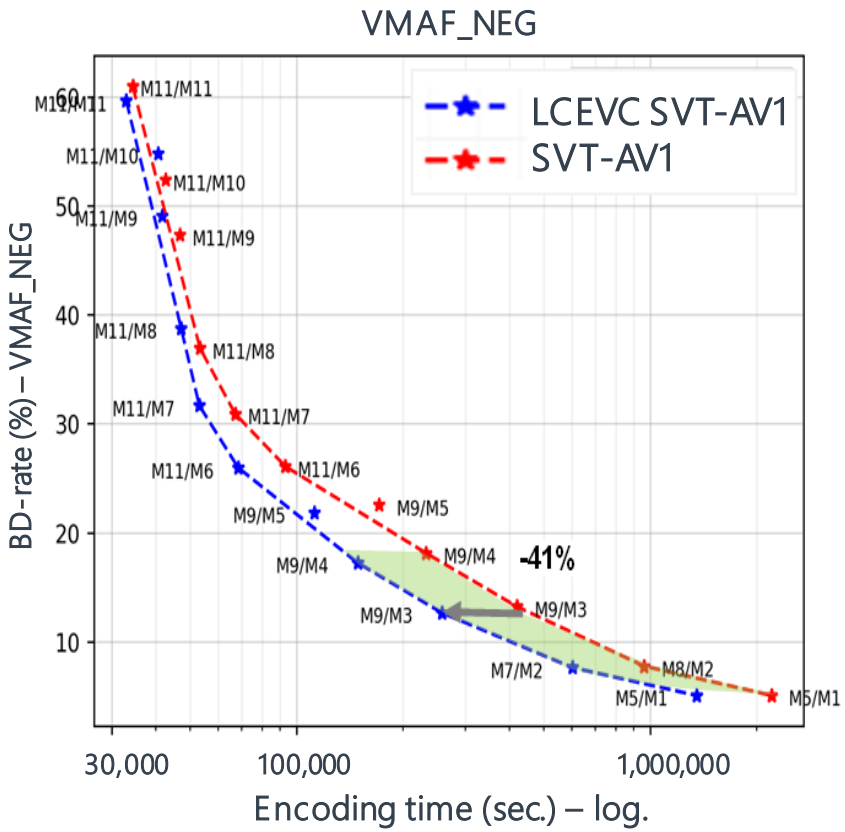


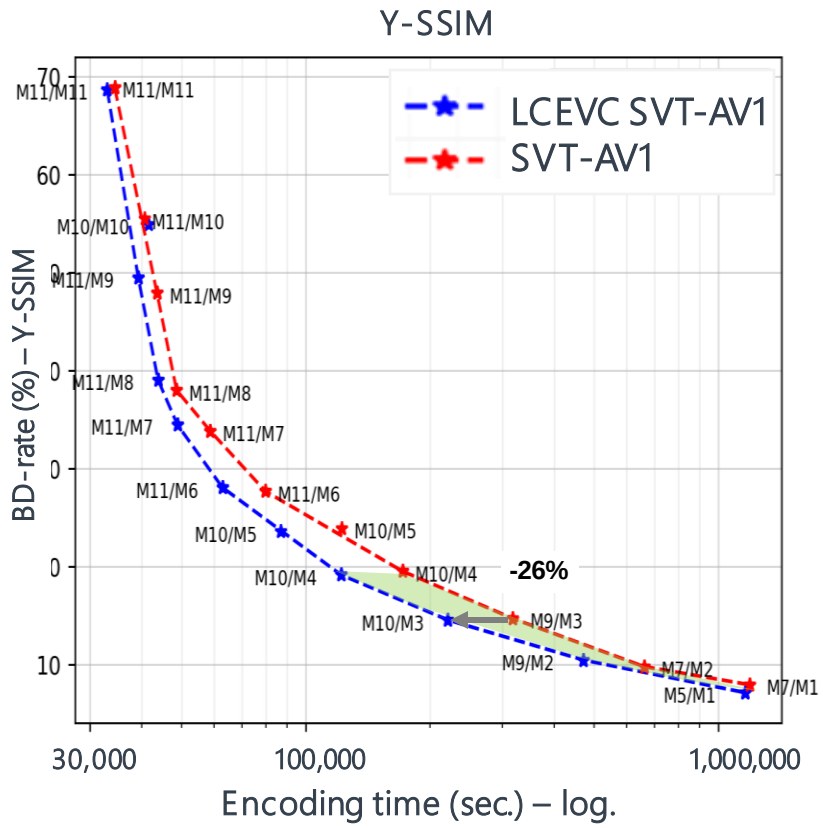


Results: Fast Encoding parameter selection (optimal preset used as a reference for prediction)

BD-rate – Computations trade-off across presets, Fast Encoding parameter selection







LCEVC AV1 decoding power and battery consumption

Decoder CPU efficiency results – 1080p matching

- CPU decoding time as a proxy for decode complexity
- AV1 decoder: Dav1d version 1.0
- LCEVC used CPU-only decoding libraries, which have not undergone significant code-optimization
- `-fast_decode` SVT-AV1 flag active for both SVT-AV1 and LCEVC SVT-AV1
- Concatenation of the El Fuente shots encoded according to the discrete DO
- Test 1) 1080p: selecting matching sections for AV1 and LCEVC AV1 resulting in 1080p resolution, in order to show like-for-like results for same-resolution contents

CPU time gain for LCEVC AV1 vs. AV1 decoding (1080p matching shots)

Preset	M3	M4	M5	M6	M7	M8	M9	M10	M11
VMAF90	-19%	-20%	-21%	-19%	-20%	-24%	-26%	-25%	-28%
VMAF95	-25%	-25%	-29%	-26%	-24%	-29%	-30%	-28%	-27%

Decoder CPU efficiency results – DO concatenation

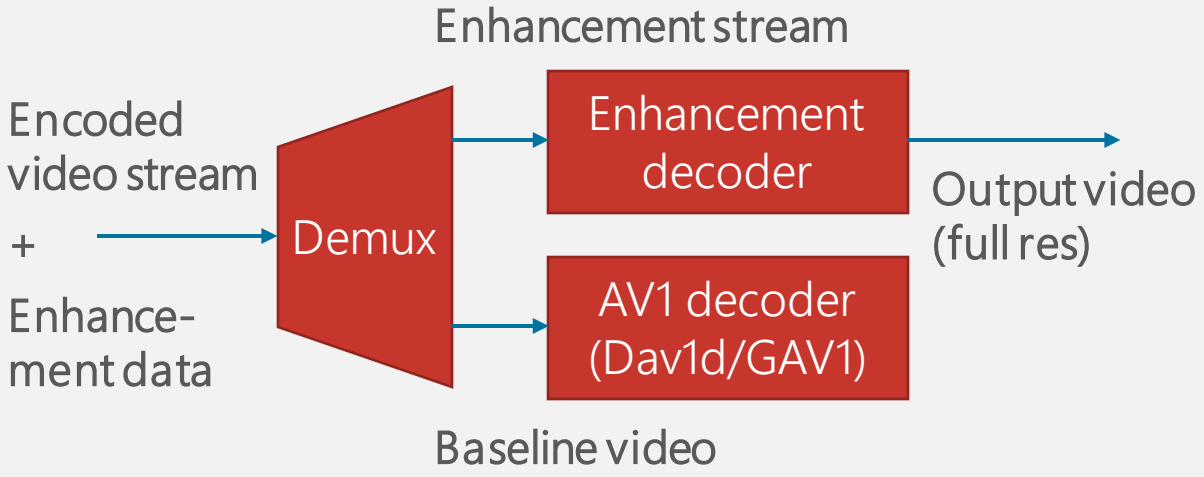
Same as before, except that:

- Test run on the concatenation of all 140 shots, at the resolution determined by the discrete DO
- Using bilinear upsampler to upscale to 1080p
- Focus on higher qualities/resolutions: VMAF quality levels between 95 and 60

CPU time gain for LCEVC AV1 vs. AV1 decoding (full concatenation, bilinear upsampling)

Preset	M3	M4	M5	M6	M7	M8	M9	M10	M11
VMAF60	-12%	-12%	-16%	-27%	-31%	-36%	-41%	-43%	-47%
VMAF70	-34%	-38%	-38%	-41%	-45%	-47%	-51%	-47%	-44%
VMAF80	-47%	-44%	-44%	-45%	-44%	-41%	-38%	-39%	-32%
VMAF90	-38%	-37%	-34%	-34%	-34%	-30%	-30%	-30%	-29%
VMAF95	-33%	-32%	-29%	-27%	-27%	-26%	-28%	-30%	-31%
Average VMAF	-33%	-33%	-32%	-35%	-36%	-36%	-38%	-38%	-37%

Real-life power tests on Android mobile devices

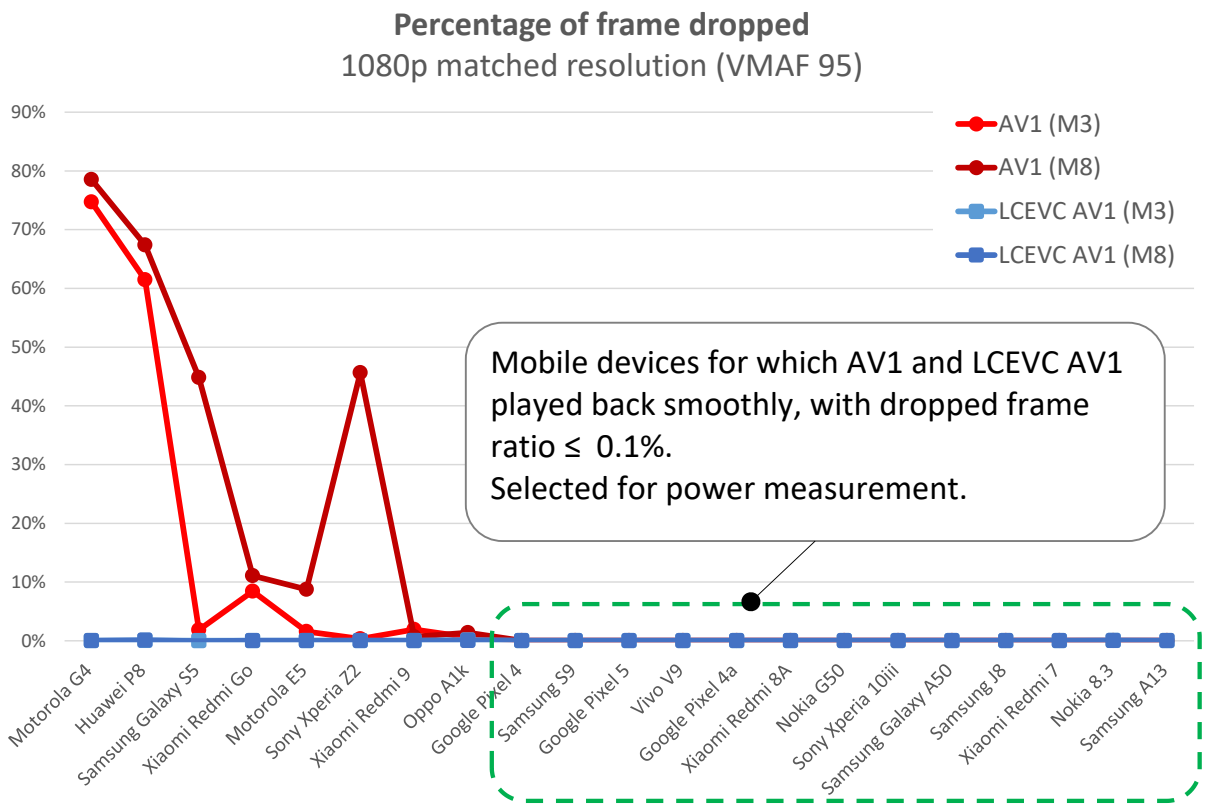


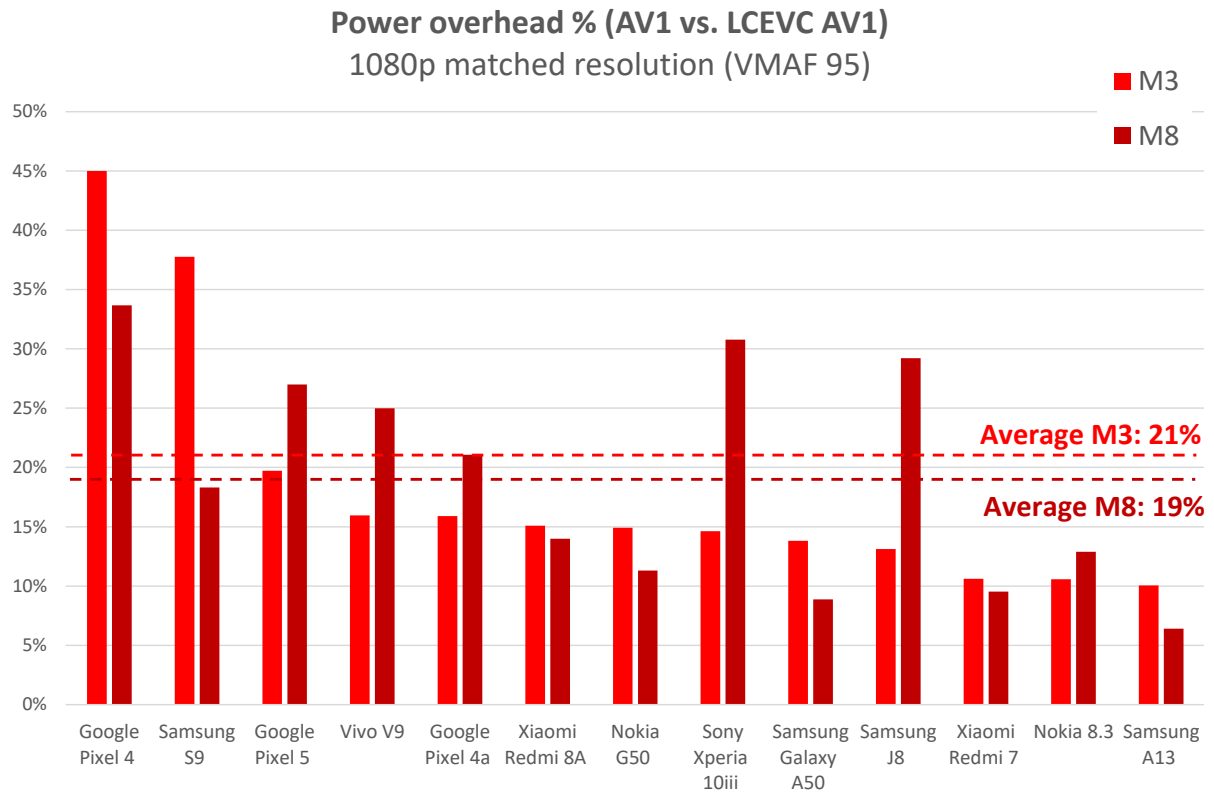
- Due to its hierarchical structure, the LCEVC decoder decodes the AV1 base layer at a quarter resolution (e.g., 540p for a 1080p stream)
- The LCEVC enhancement is efficiently decoded using existing hardware blocks in the device (e.g., GPU shaders and scalers)

Test methodology

- LCEVC AV1 decoding with optimized CPU-GPU libraries
- `-fast_decode` SVT-AV1 flag active
- Player capable of playing back both AV1 and LCEVC AV1, based on ExoPlayer with added support for Dav1d (v 1.0) and GAV1 software decoders, as provided by Android's MediaCodec interface
- El Fuente concatenation (1080p and mixed resolution according to DO), 30 fps and 60 fps
- Measuring power consumption (mW), dropped frames and battery drain

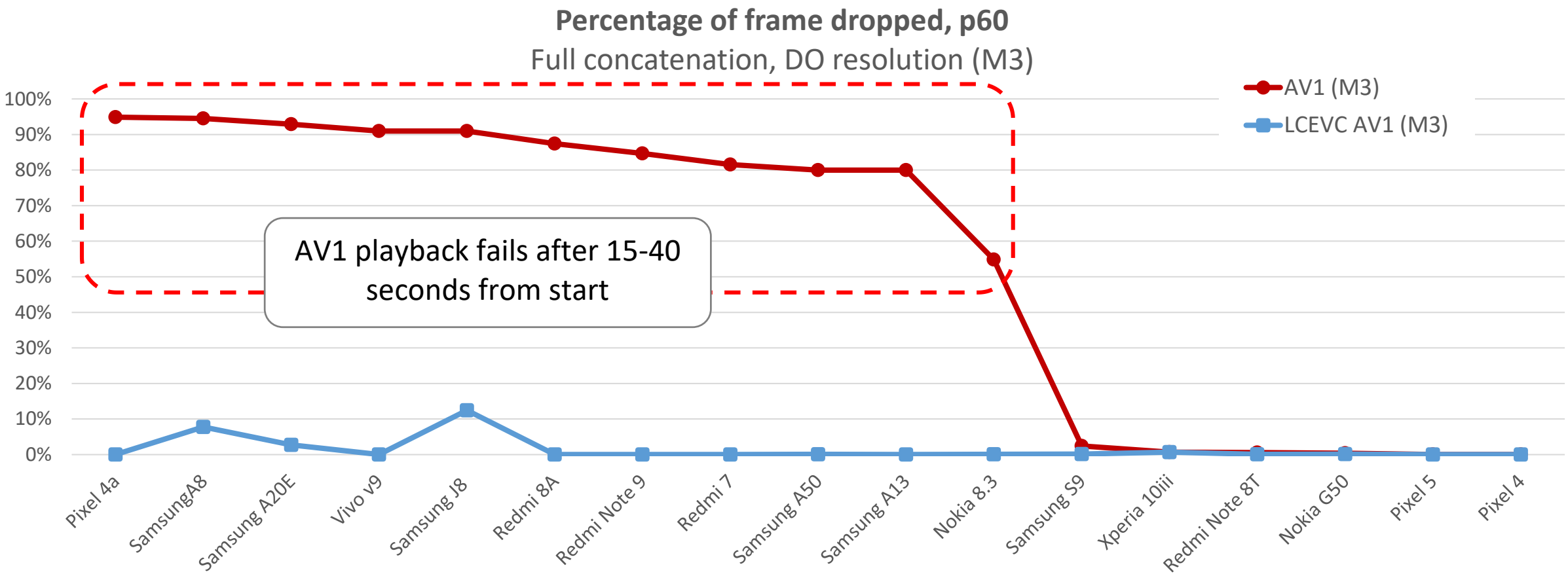
Real-life device results, 1080p30 (Dav1d decoder)





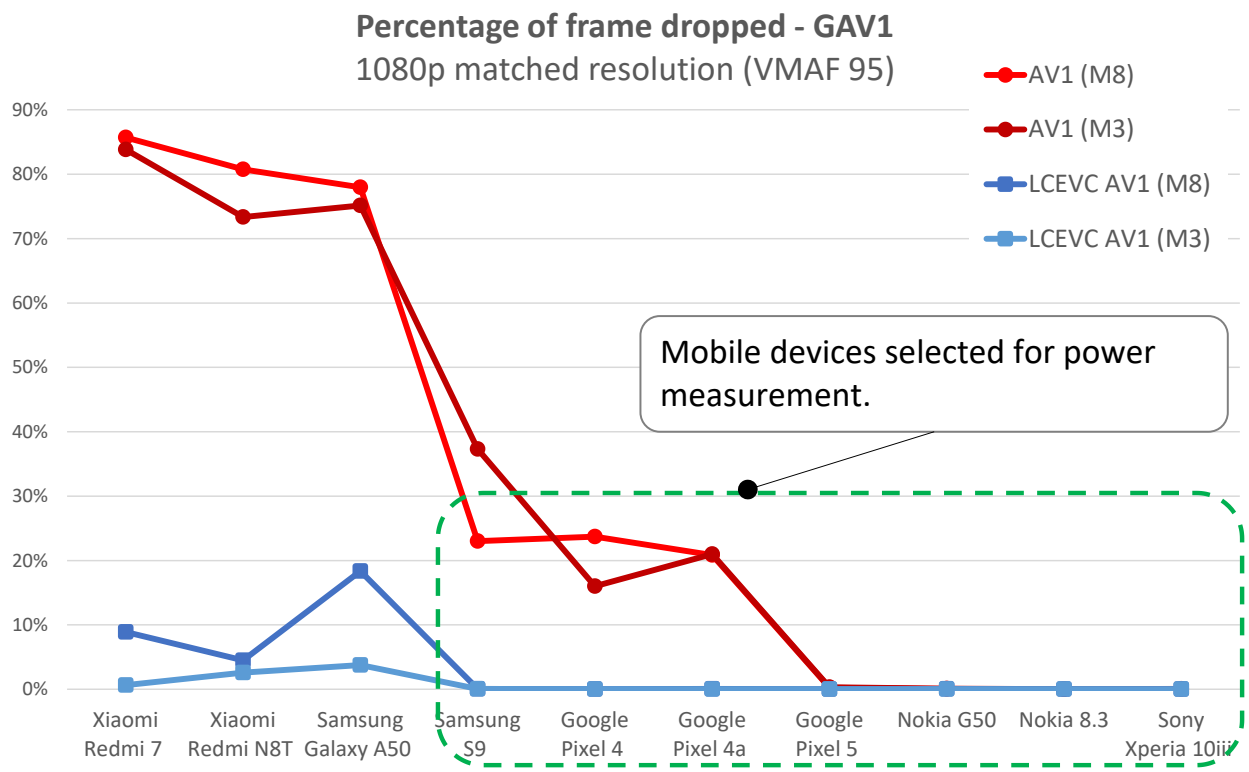
- On less powerful devices, AV1 struggled or failed to playback, while LCEVC AV1 played back smoothly
- When playback was smooth, AV1 consistently **consumed ~20% more power** than LCEVC AV1

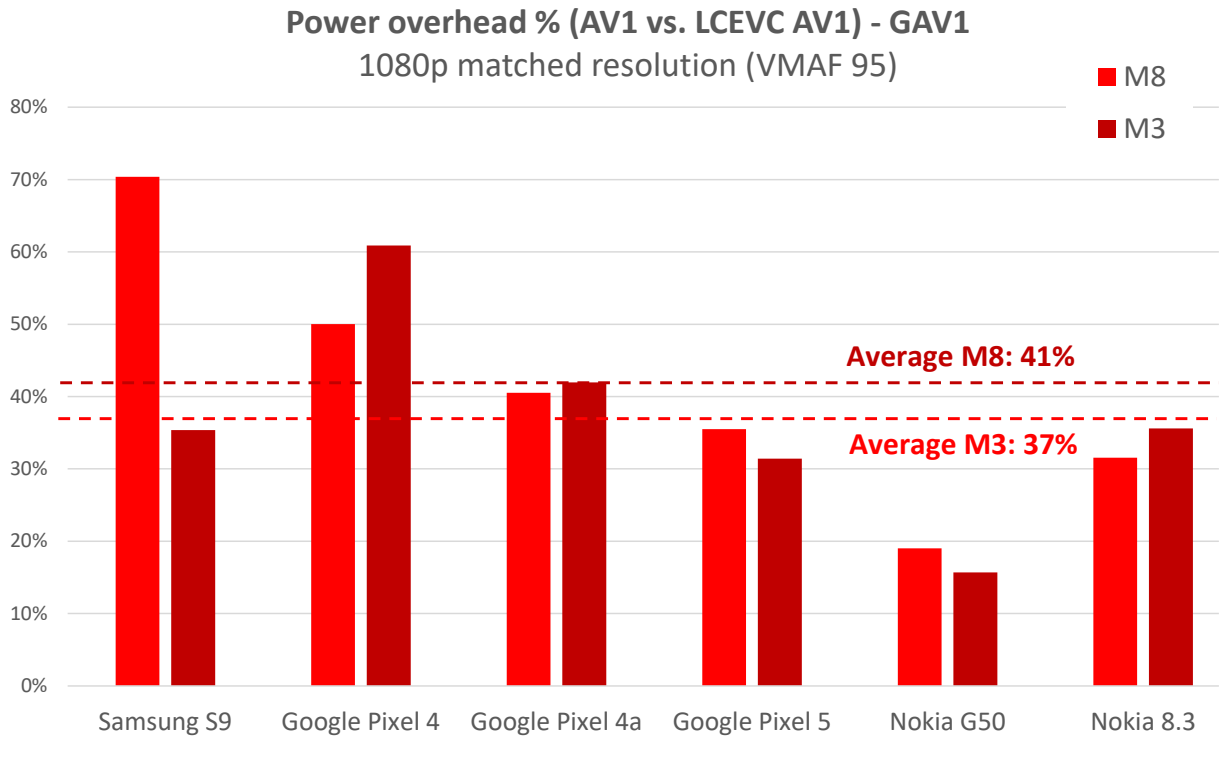
Real-life device results, 1080p60 (Dav1d decoder)



- Most tested devices could not playback 1080p60 AV1 content, resulting in frequent freezes or failed playback
- With LCEVC AV1 video playback was smooth, with occasional stuttering only on older devices

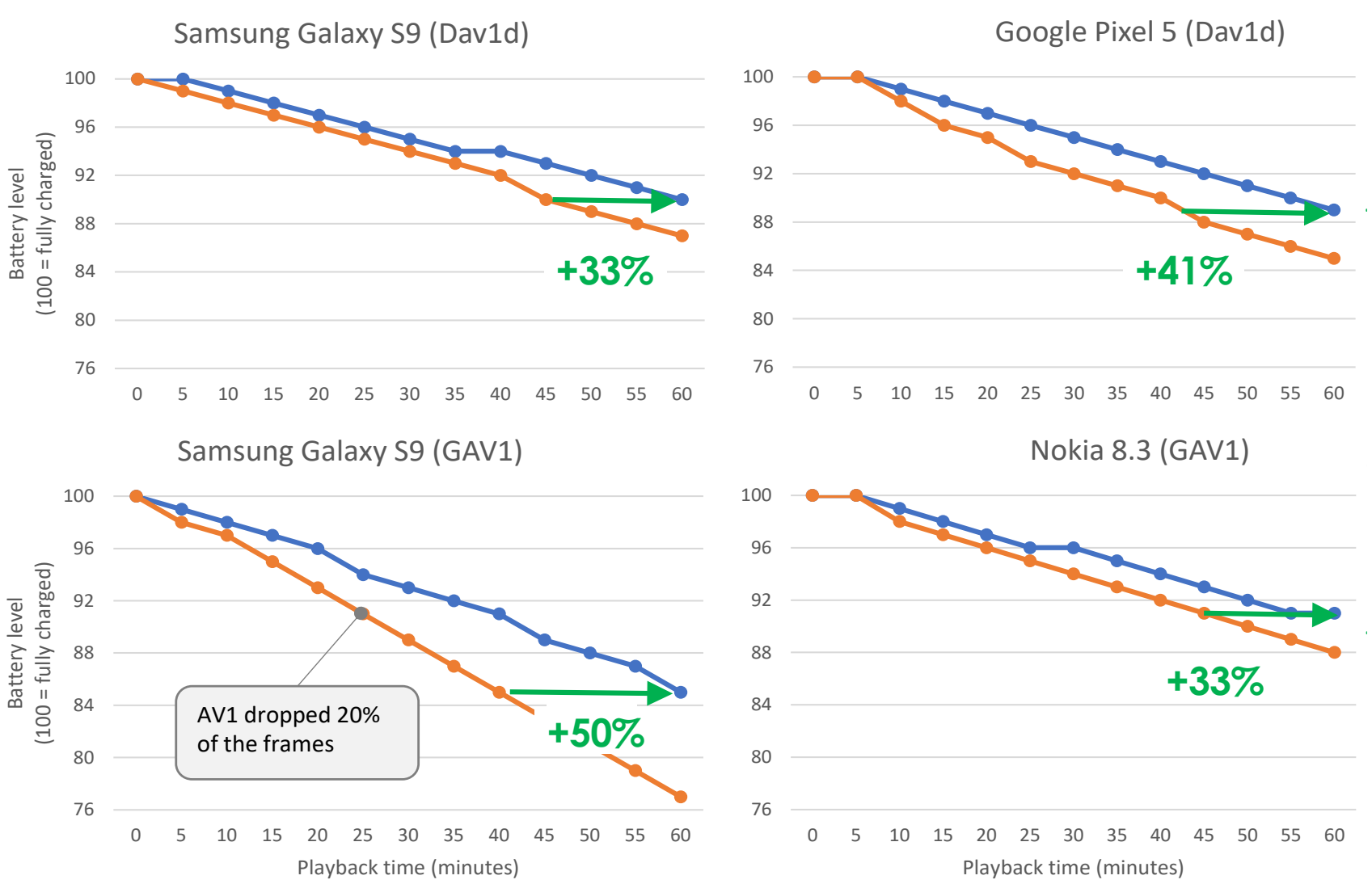
Real-life device results, 1080p30 (GAV1 decoder)





- With GAV1 decoder, AV1 playback failed on most tested devices, while LCEVC AV1 significantly improved playback
- AV1 consistently **consumed ~40% more power** than LCEVC AV1

Mobile battery life extension



- 1h playback measuring battery drain
- LCEVC extends mobile battery life by up to 50% with respect to state-of-the-art AV1 software decoding (Dav1d and GAV1)

Conclusions

Conclusions

This work evaluated MPEG-5 LCEVC to assess its impact in improving the compression efficiency vs cycles trade-offs vs. native SVT-AV1, x264 and x265 for VOD applications, as well as its capability to improve playback and battery life of AV1 software decoding:

1. LCEVC is a valuable tool to improve the quality-cycles trade-offs across the full complexity range for the three tested standards – AVC, HEVC and AV1
2. On decoding, LCEVC enlarges the set of mobile devices capable of playing back high quality and high frame-rate AV1 content, and extends mobile battery life by up to 50% vs. state-of-the-art AV1 software decoding

The combination of LCEVC + SVT-AV1 builds on notable improvements of SVT-AV1 (improved speed-quality trade-off & fast decoding) and showcases the concrete possibility for widespread and energy efficient deployments

Additional information

Software and hardware configuration for encoding tests

HW Configuration:

- AWS c6i.32xlarge instance, Ubuntu Server 20.04 Linux OS on a Third Generation Intel® Xeon® Platinum 8375C.
- Clips chosen: Open source El Fuente clip 7m52sec, 140 shots, https://opencontent.netflix.com/#h.goi1q35qkalw
- Clips downscaled using ffmpeg to the following resolutions:
  - 1920x1080 ->1920x1080, 1280x720, 960x540, 768x432, 640x360, 512x288, 384x216, 256x144
  - Ffmpeg (n5) cli : ffmpeg -y -i input.y4m -sws_flags lanczos+accurate_rnd+full_chroma_int -sws_dither none -param0 5 -strict -1 -s:v 1280x720 output.y4m
- All clips at all resolutions are then encoded at 11 crf (or pCRF for LCEVC) values:
  - x264/x265 [19,21,23,25,27,29,31,33,35,37,41]
  - SVT-AV1 [23,27,31,35,39,43,47,51,55,59,63]
  - LCEVC x264/x265 [21, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42], aimed to cover bitrates/qualities similar to native
  - LCEVC SVT-AV1 [21, 22, 24, 26, 28, 30, 32, 34, 36, 38, 41, 44], aimed to cover bitrates/qualities similar to native
- Used the parallel tool to schedule the tasks: **parallel -j 128** and **/usr/bin/time –v** command to measure the **user time + system time**
- Sample encoding command lines:
  - **x264 (0.163.3060):** ./ffmpeg -y -i input.y4m -g <clip_length+1> -keyint_min <clip_length+1> -crf <crf> -c:v libx264 -threads 1 -preset <preset> -tune psnr -x264-params scenecut=0 -f h264 output.bin
  - **X265 (3.5):** ./ffmpeg -y -i input.y4m -g <clip_length+1> -keyint_min <clip_length+1> -crf <crf> -threads 1 -c:v libx265 -preset <preset> -tune psnr -x265-params pools=1:wpp=0:scenecut=0:no-info=1 -f hevc output.bin
  - **SVT-AV1 (v1.1):** ./ffmpeg -y -i input.y4m -crf <QP> -preset <preset> -g <clip_length+1> -svtav1-params
  - fast-decode=1:lp=1 -threads 1 -c:v libsvtav1 -f webm output.webm
  - **LCEVC x264:** ffmpeg -y -i input.y4m -g <clip_length+1> -b:v 0k -c:v lcevc_h264 -base_encoder x264
  - -eil_params "lcevc_tune=vmaf_neg;accel_num_worker_threads=0;min-keyint=<clip_length+1>;rc_pcrf=<pCRF>;threads=1;preset=<preset>;lcevc_preset=<lcevc_preset>;scenecut=0" -f h264 output.bin
  - **LCEVC x265:** ffmpeg -y -i input.y4m -g <clip_length+1> -b:v 0k -c:v lcevc_hevc -base_encoder x265
  - -eil_params "lcevc_tune=vmaf_neg;accel_num_worker_threads=0;min-keyint=<clip_length+1>;rc_pcrf=<pCRF>;numa-pools=1;wpp=0;preset=<preset>;lcevc_preset=<lcevc_preset>;scenecut=0" -f hevc output.bin
  - **LCEVC SVT-AV1:** ffmpeg -y -i input.y4m -b:v 0k -g <clip_length+1> -threads 1 -c:v lcevc_av1
  - -base_encoder svt_av1 -eil_params "lcevc_tune=vmaf_neg;accel_num_worker_threads=0;rc_pcrf=<pCRF>;preset=<preset>;lcevc_preset=<lcevc_preset>;fast-decode=1;logical-processors=1" -f webm output.webm
- Result encodings are then decoded and scaled back to the initial resolution (1080p) and metrics VMAF / VMAF_NEG / PSNR / SSIM (libvmaf v2.3.0) are computed based on the initial resolution
- Bitrates and Distortion metrics for all crf/pCRF and all resolutions for every clip are passed to the convex hull measurement tool: github.com/ikatsavounidisFB/convex_hull
- BD rates are then measured over only the convex-hull points and plotted vs the user time of all the encodings per preset.
- Performance results are based on testing as of July 2022 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.
- For details on EC2 instance protections for various vulnerabilities including side-channel, please refer to Amazon security bulletins: https://aws.amazon.com/security/security-bulletins/