MSU Codec Comparison 2019

Part IV: FullHD Content, High Quality Use Case

Video group head	Dr. Dmitriy Vatolin
Project head	Dr. Dmitriy Kulikov
Measurements & analysis	Dr. Mikhail Erofeev,
	Egor Sklyarov,
	Anastasia Antsiferova

Codecs:

AV1

- aom
- rav1e
- SVT-AV1

Non AV1

- SVT-HEVC
- SVT-VP9
- x264
- x265

CS MSU Graphics & Media Lab, Video Group

March 27, 2020

http://www.compression.ru/video/codec_comparison/index_en.html

videocodec-testing@graphics.cs.msu.ru

Contents

1. ACKNOWLEDGMENTS

The Graphics & Media Lab Video Group would like to thank **rav1e** for providing the codec and settings used in this report. We are also grateful to them for their help and technical support during the tests.

2. OVERVIEW

2.1. Sequences

	Sequence	Number of frames	Frame rate	Resolution
1.	axebat	354	24	1920×1080
2.	carpets	801	25	1920×1080
3.	foggy_beach	576	24	1920×1080
4.	hera	1023	24	1920×1080
5.	television_studio	970	25	1920×1080
6.	witcher3_700	700	60	1920×1080

Table 1: Summary of video sequences

Brief descriptions of the sequences used in our comparison appear in Table 1. Appendix A provides more-detailed descriptions of these sequences.

2.2. Codecs

Codec	Developer	Version
aom	AOMedia	1.0.0-errata1-avif
rav1e	rav1e	0.2.0 (p20200127)
SVT-AV1	Open Visual Cloud	0.8.1
SVT-HEVC	Open Visual Cloud	1.4.3
SVT-VP9	Open Visual Cloud	0.1.0
x264	x264 Developer Team	core:157 r2969 d4099dd
x265	MulticoreWare, Inc.	3.2+15-04db2bfee5d6

Table 2: Short codecs' descriptions

Brief descriptions of the codecs used in our comparison appear in Table 2. We used x264 as a good-quality AVC reference codec. Appendix C provides detailed descriptions of all codecs in our comparison.

3. OBJECTIVES AND TESTING RULES

In this report we use objective assessment methods to compare the encoding quality of recent AV1 encoders as well as encoders implementing other standards. This effort employed 6 video sequences at FullHD resolution to evaluate codec performance. To choose our test set, we analyzed 384,946 video sequences and selected representative examples (a detailed description of the selection process appears in Appendix B).

Our comparison consists of one use case. For this use case we offered the codec developers the option to provide encoding parameters for our tests. Nevertheless, the parameters had to satisfy a minimum speed requirement:

- High Quality—0.005fps

Our comparison used a computer with the following configuration: based on an Intel Core i7-8700K (Coffee Lake) processor @ 3.7GHz with 32 GB of RAM running Windows 10. For objective quality measurements we used the YUV-SSIM metric (see Appendix E.1).

4. RD CURVES

Judging from the mean quality scores (computed using the method described in Section D), first place in the quality competition goes to **aom**, second place goes to **SVT-AV1**, and third place to **x265**.

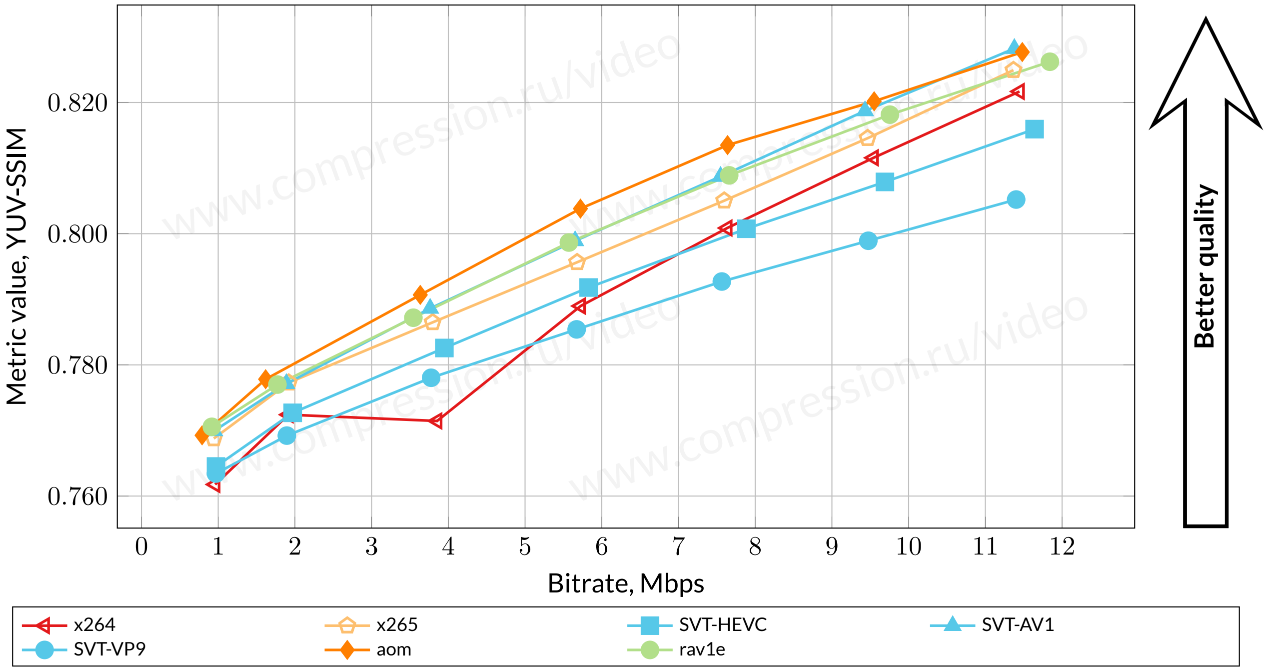


Figure 1: Bitrate/quality—use case "High Quality," *hera* sequence, YUV-SSIM metric.

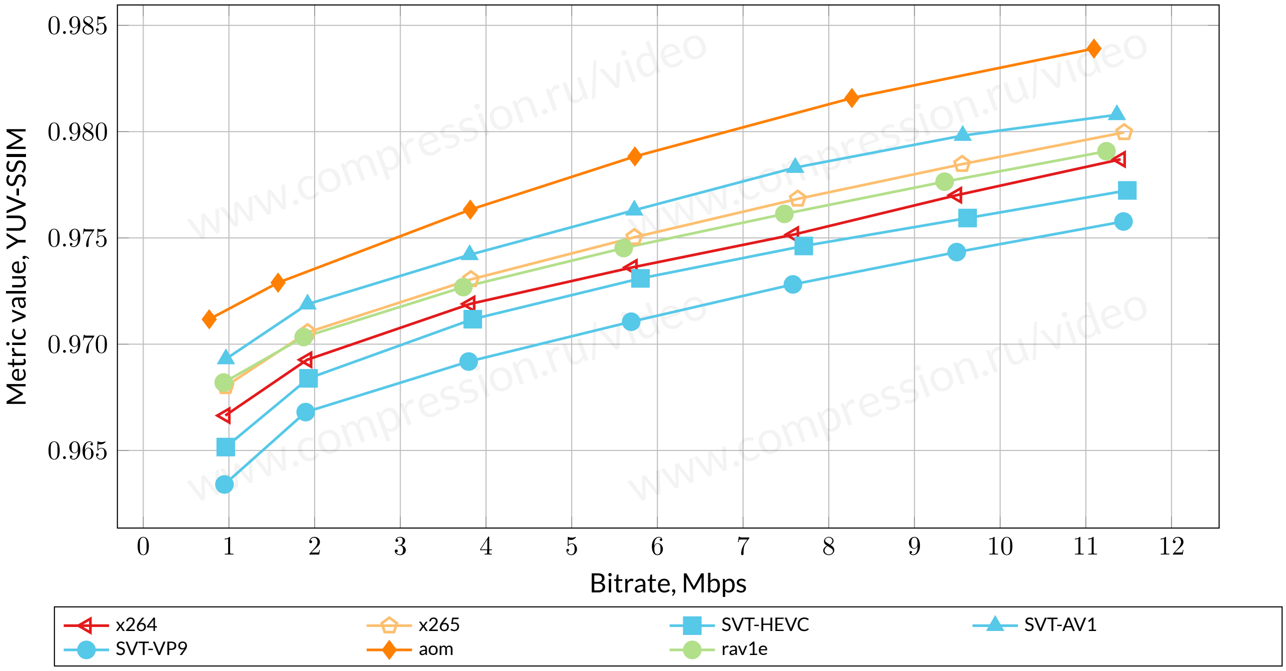


Figure 2: Bitrate/quality—use case "High Quality," *television_studio* sequence, YUV-SSIM metric.

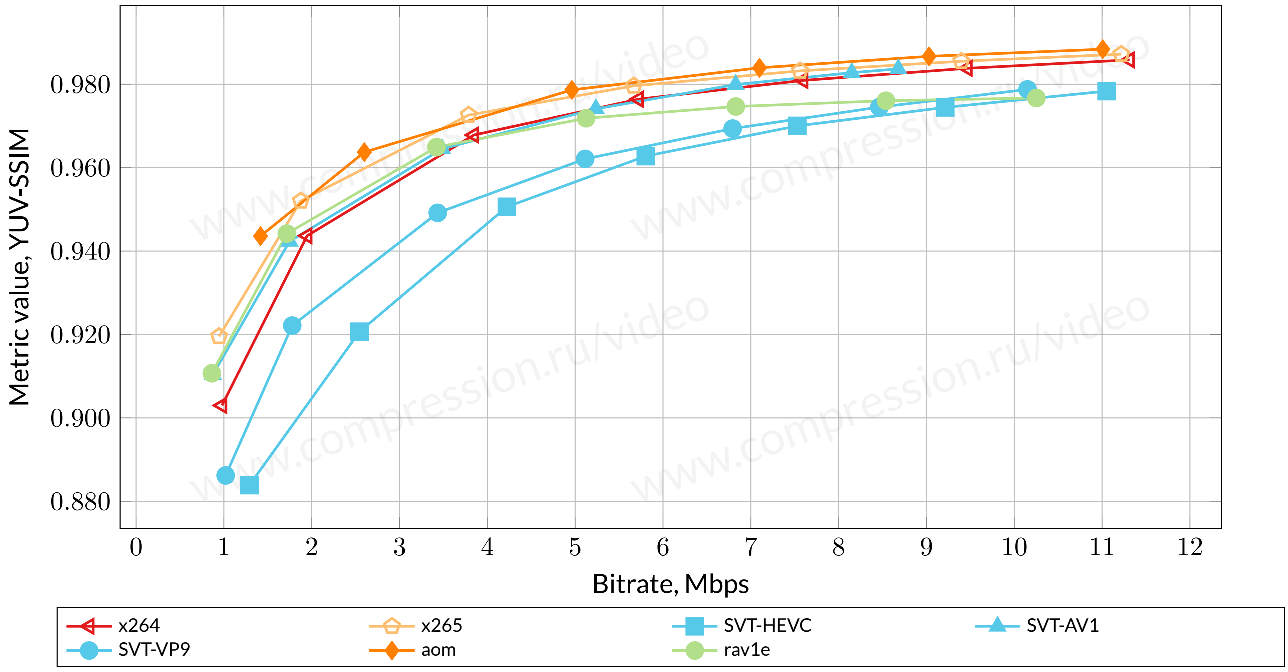


Figure 3: Bitrate/quality—use case "High Quality," *carpets* sequence, YUV-SSIM metric.

5. ENCODING SPEED

Judging from the mean speed scores (computed using the method described in Section D), first place in the speed competition goes to **SVT-VP9**, second place goes to **x264**, and third place to **SVT-HEVC**.

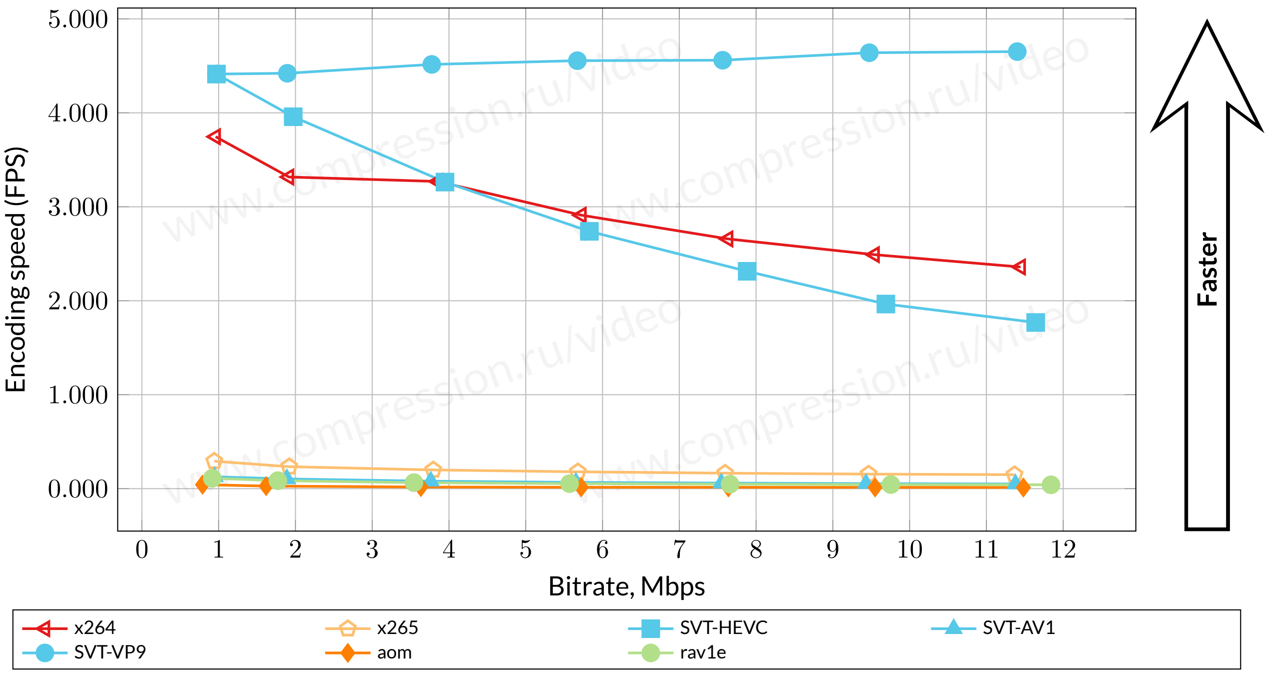


Figure 4: Encoding speed—use case "High Quality," *hera* sequence.

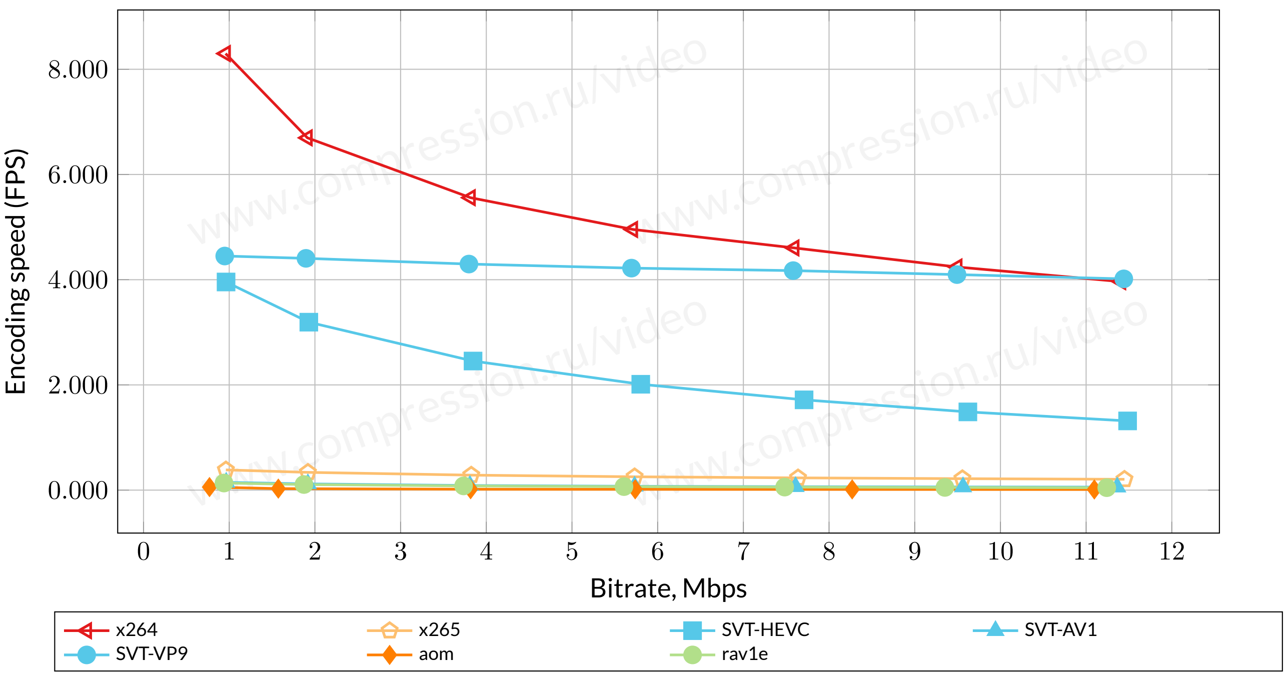


Figure 5: Encoding speed—use case "High Quality," *television_studio* sequence.

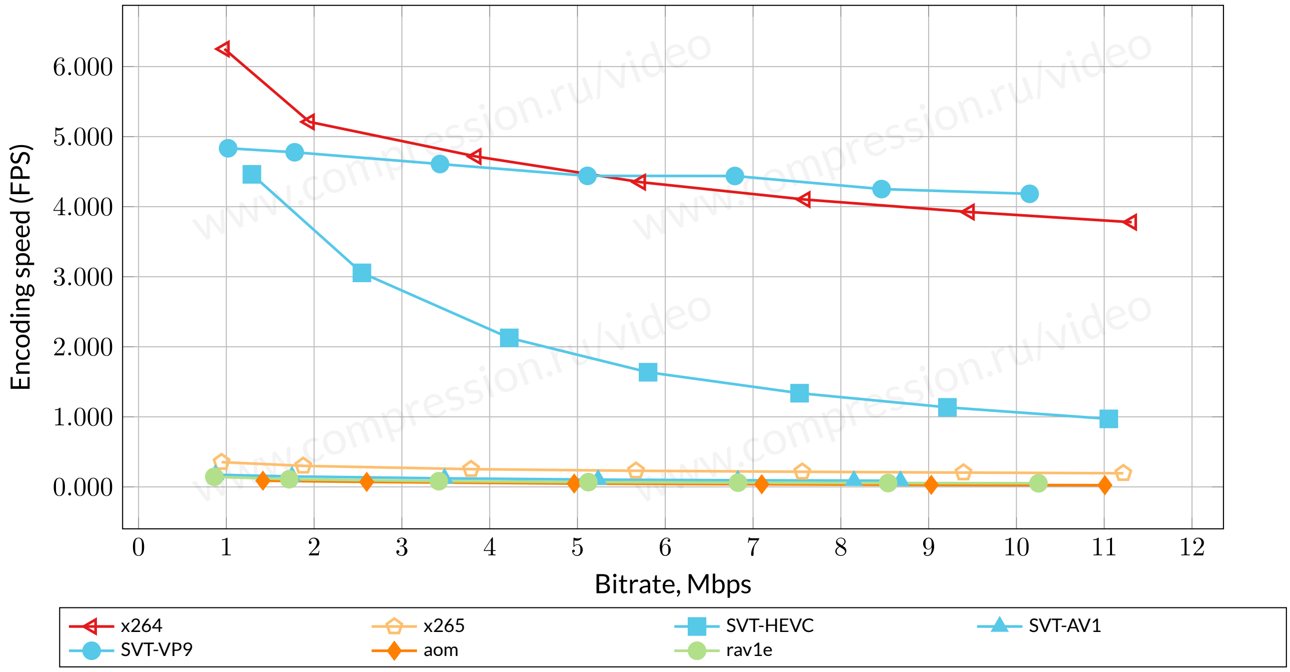


Figure 6: Encoding speed—use case “High Quality,” *carpets* sequence.

6. SPEED/QUALITY TRADE-OFF

Detailed descriptions of the speed/quality trade-off graphs are in Appendix D. Some graphs omit the results for a particular codec owing to that codec's extremely poor performance (i.e., its RD curve fails to intersect with the reference RD curve).
The speed/quality trade-off graphs show both relative quality and speed scores for the encoders under comparison. Since we chose x264 as the reference codec, we normalized all scores to the x264 scores.

There are five Pareto-optimal encoders: **aom**, **SVT-AV1**, **x265**, **x264**, and **SVT-VP9**.

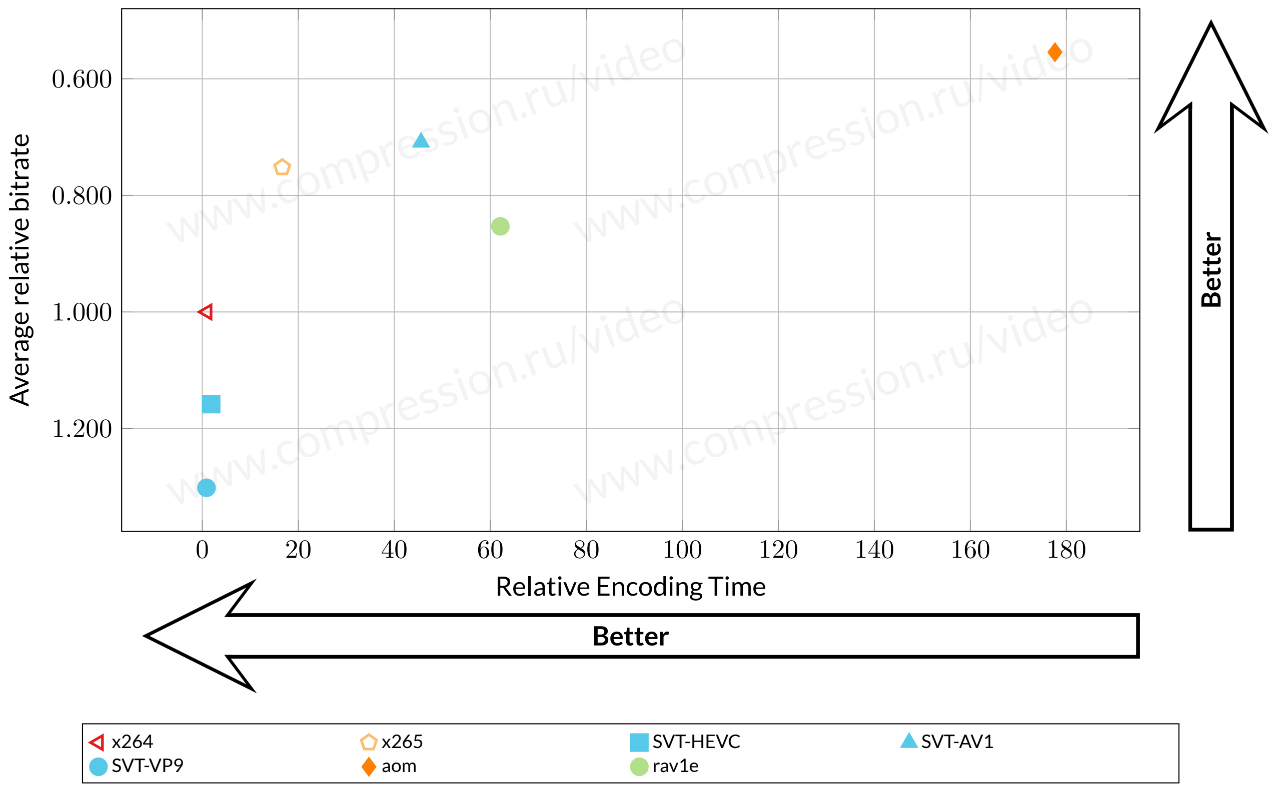


Figure 7: Speed/Quality Trade-Off—use case "High Quality," all sequences, YUV-SSIM metric.

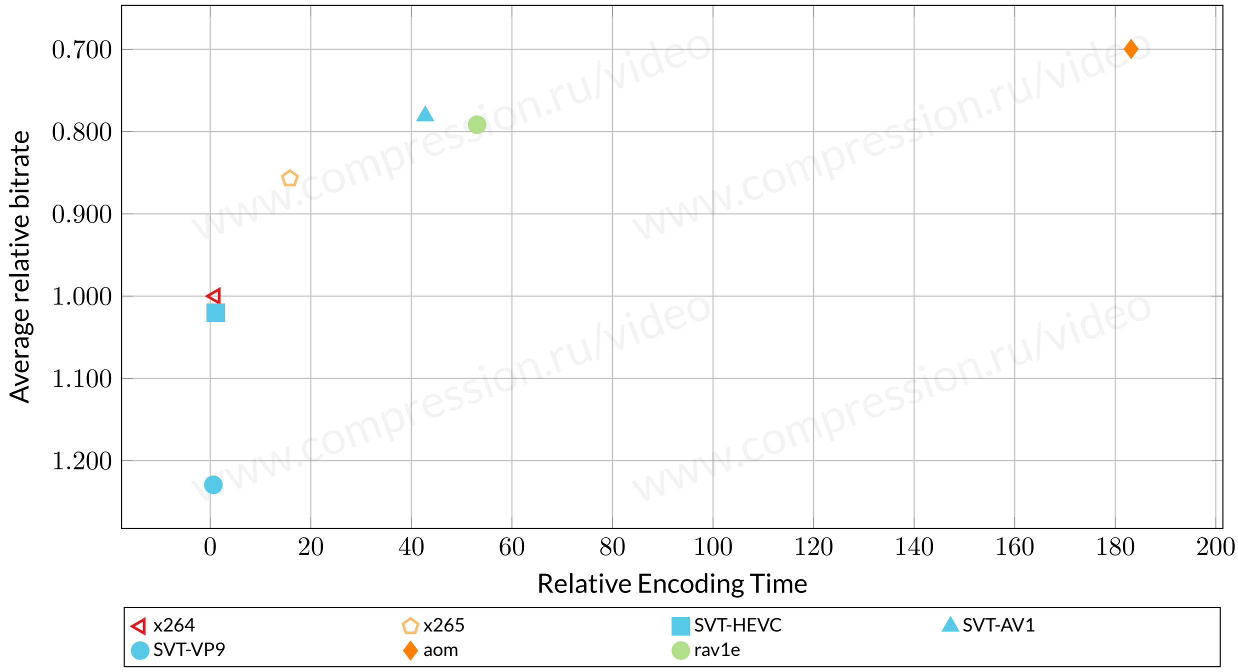


Figure 8: Speed/Quality Trade-Off—use case "High Quality," *hera* sequence, YUV-SSIM metric.

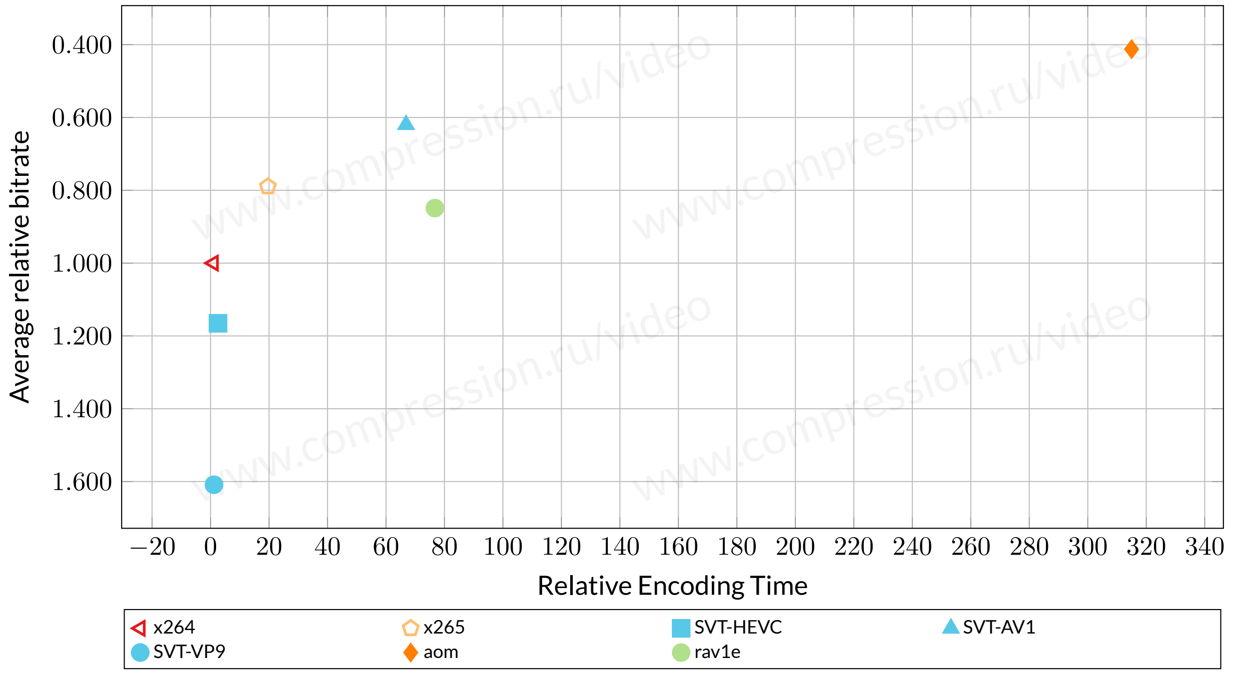


Figure 9: Speed/Quality Trade-Off—use case "High Quality," *television_studio* sequence, YUV-SSIM metric.

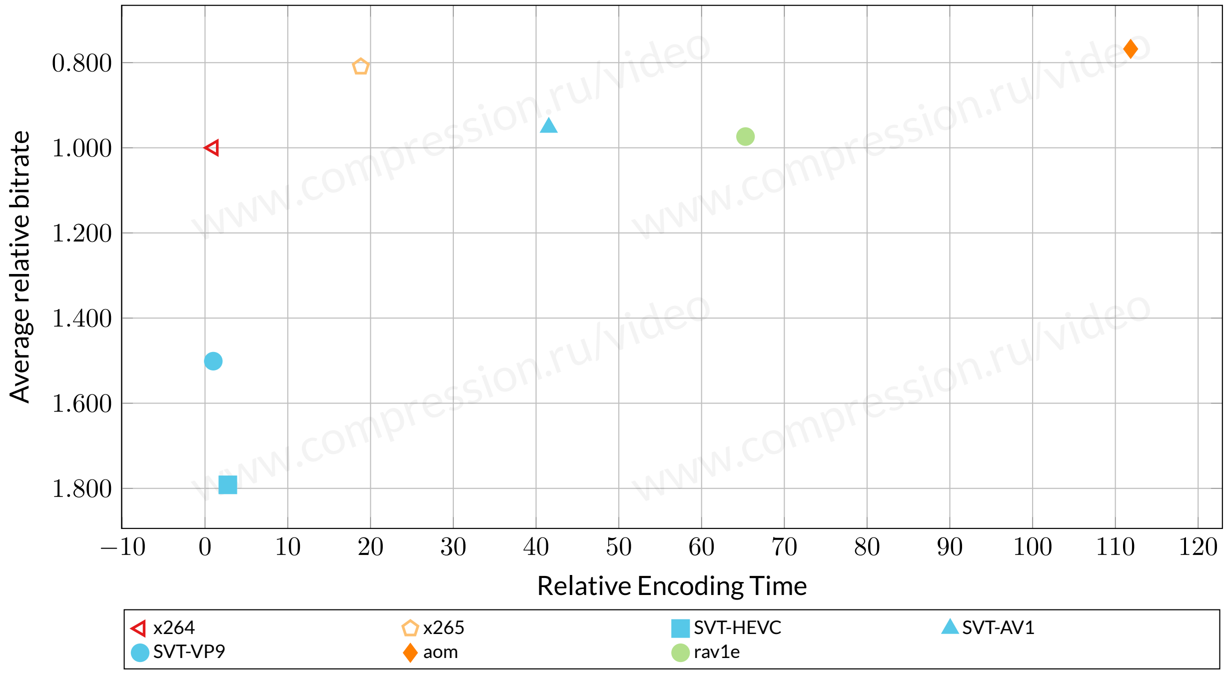


Figure 10: Speed/Quality Trade-Off—use case "High Quality," *carpets* sequence, YUV-SSIM metric.

7. BITRATE HANDLING

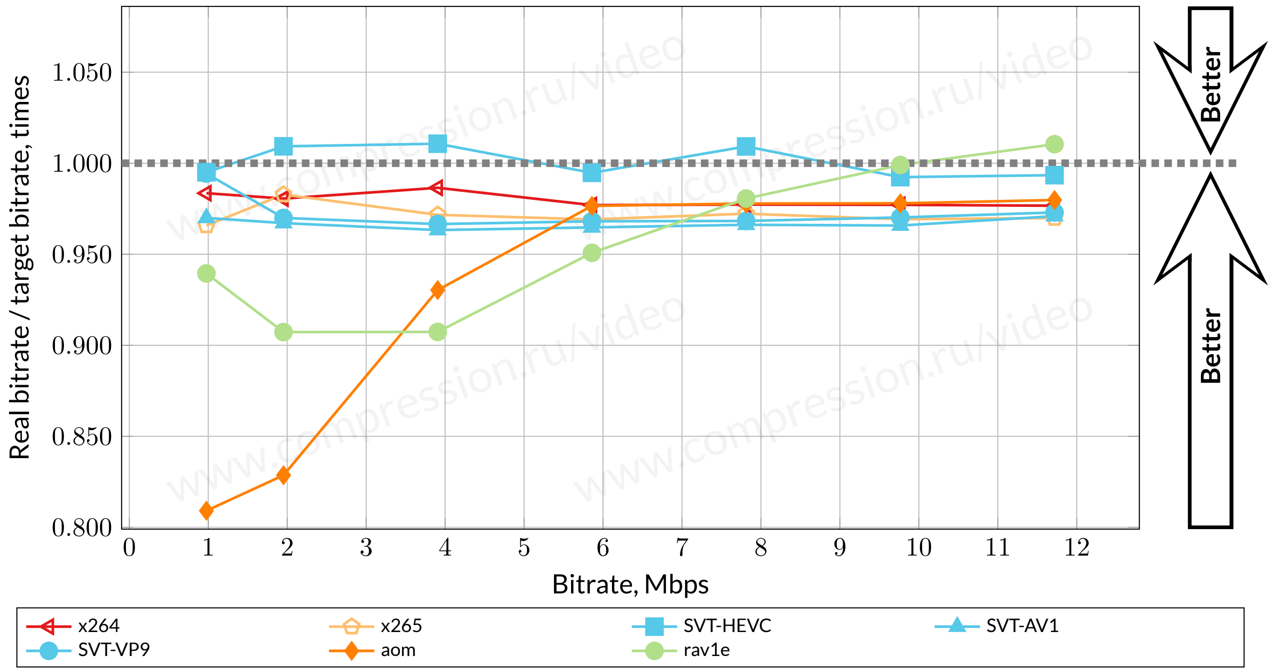


Figure 11: Bitrate handling—use case "High Quality," *hera* sequence.

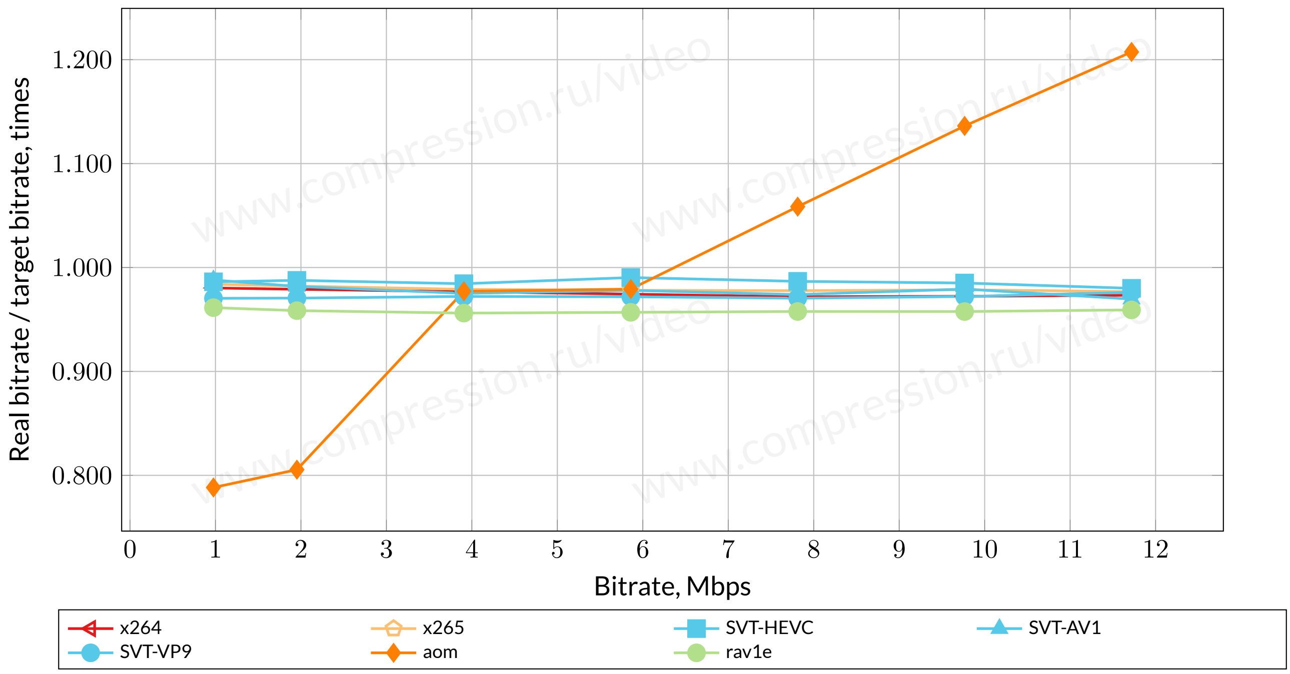


Figure 12: Bitrate handling—use case "High Quality," *television_studio* sequence.

aom and **rav1e** poorly keep low target bitrates, greatly lowering the real bitrate.

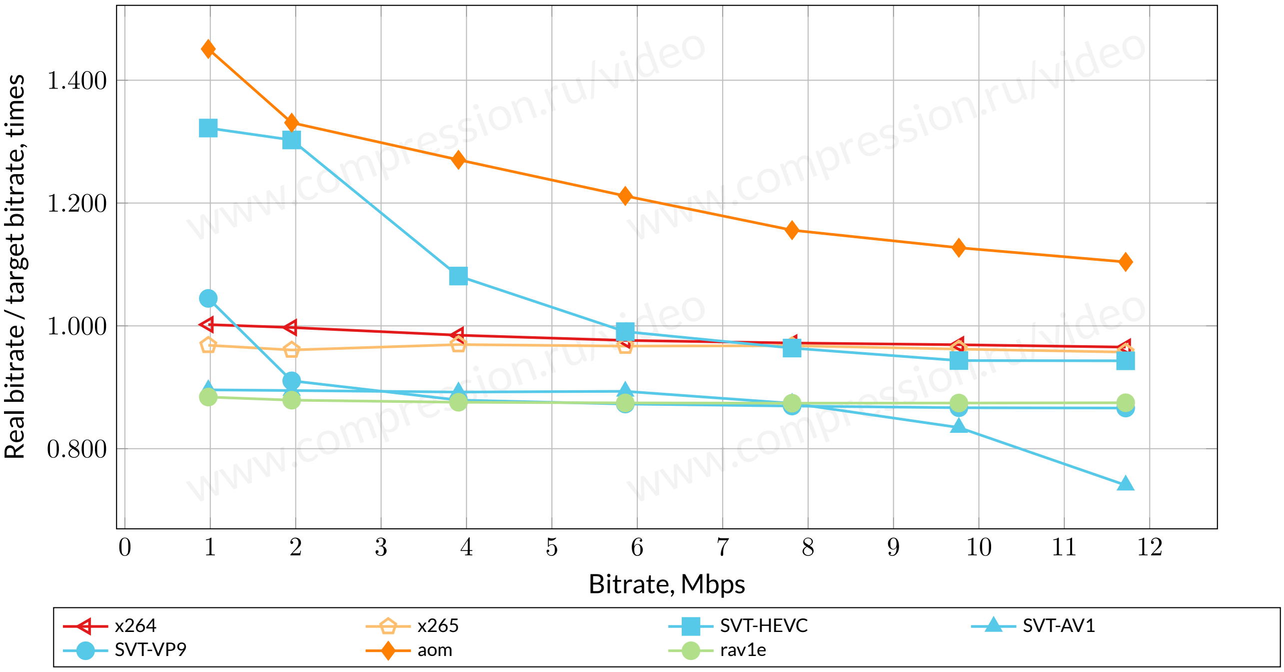


Figure 13: Bitrate handling—use case "High Quality," *carpets* sequence.

aom very much exceed the bitrate, while most codecs lower it.

8. RELATIVE QUALITY ANALYSIS

Note that each number in the tables below corresponds to some range of bitrates (see Appendix D.5). Unfortunately, these ranges can differ significantly because of differences in the quality of compared encoders. This situation can lead to some inadequate results when three or more codecs are compared.

	x264	x265	SVT-HEVC	SVT-AV1	SVT-VP9	aom	rav1e
x264	100.0%	134.0%	91.0%	146.0%	80.0%	195.0%	118.0%
x265	75.0%	100.0%	66.0%	110.0%	56.0%	143.0%	87.0%
SVT-HEVC	116.0%	156.0%	100.0%	168.0%	90.0%	224.0%	136.0%
SVT-AV1	71.0%	93.0%	60.0%	100.0%	55.0%	120.0%	81.0%
SVT-VP9	130.0%	185.0%	115.0%	191.0%	100.0%	241.0%	160.0%
aom	55.0%	73.0%	48.0%	84.0%	46.0%	100.0%	63.0%
rav1e	85.0%	117.0%	76.0%	127.0%	66.0%	169.0%	100.0%

Confidence ☹ 0% — 😐 50% — ☺ 100%

Table 3: Average bitrate ratio for a fixed quality—use case "High Quality," all sequences, YUV-SSIM metric.

Table explanation is presented in Section D.5.

Figure below depicts the data from the table above. Each line in the figure corresponds to one codec. Values on the vertical axis are the average relative bitrates compared with the codecs along the horizontal axis. A lower bitrate indicates better relative results.

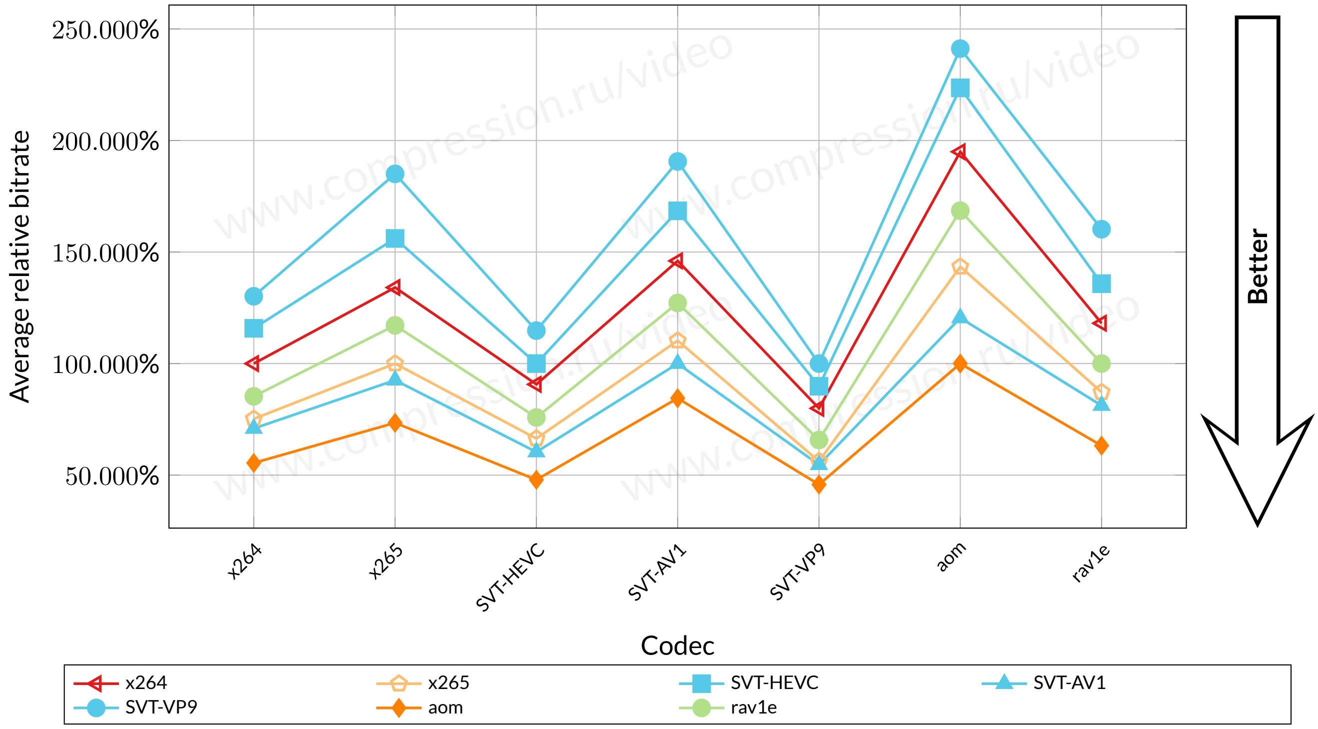


Figure 14: Average bitrate ratio for a fixed quality—use case "High Quality," all sequences, YUV-SSIM metric.

9. CONCLUSION

SVT-AV1 failed to encode foggy_beach video sequence correctly (it was impossible to calculate metrics). The chart below includes **SVT-AV1** results averaged only for 5 videos sequences. Other encoders scores are averaged for all 6 videos.

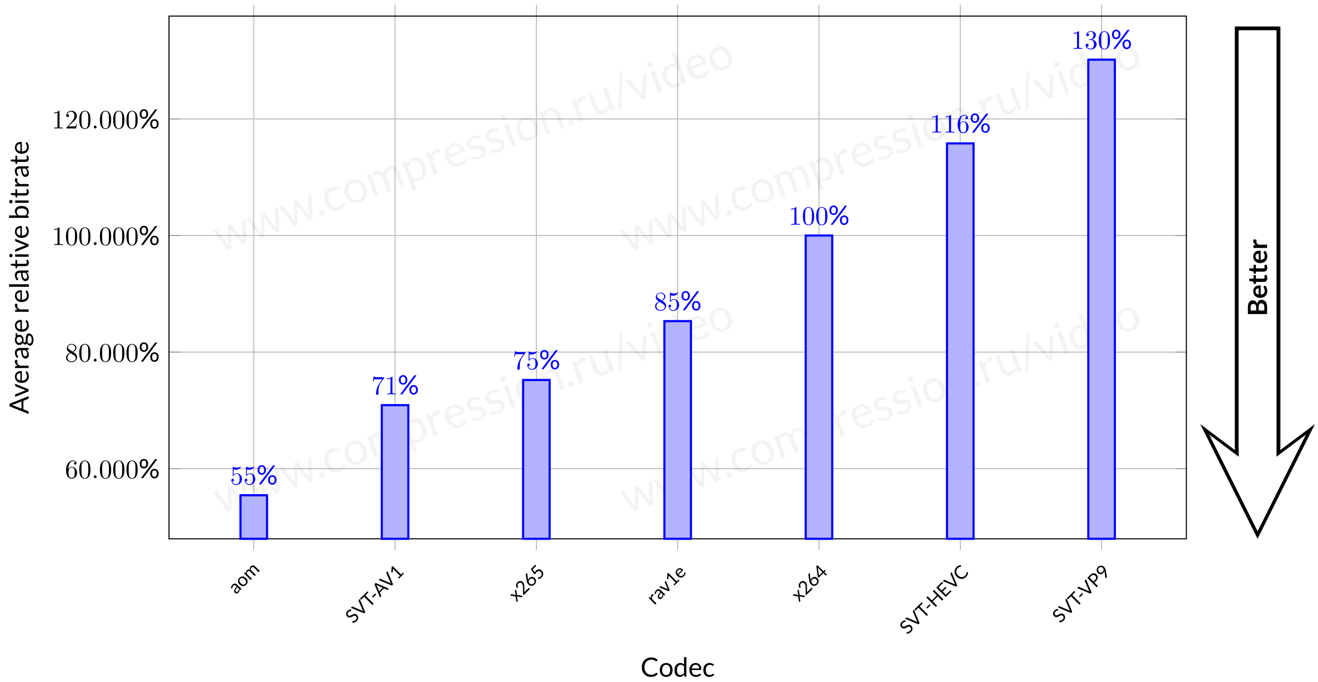


Figure 15: Average bitrate ratio for a fixed quality—use case "High Quality," all sequences, YUV-SSIM metric.

The reason of **SVT-HEVC** and **SVT-VP9** lower quality might be in one-pass encoding.

A. SEQUENCES

A.1. axebat

Sequence title	axebat
Resolution	1920×1080
Number of frames	354
Color space	YV12
Frames per second	24
Source	https://vimeo.com/173811221
Source resolution	FullHD
Bitrate	106.717

Baseball team celebrates the victory. Slow motion scenes, a lot of water drops.

Figure 16: axebat sequence, frame 0

A.2. carpets

Sequence title	carpets
Resolution	1920×1080
Number of frames	801
Color space	YV12
Frames per second	25
Source	https://vimeo.com/193845705#t=0
Source resolution	FullHD
Bitrate	105.48

Shortchanging scenes of people, children and animals acting on different carpets. Filmed from one angle.

Figure 17: carpets sequence, frame 25

A.3. foggy_beach

Sequence title	foggy_beach
Resolution	1920×1080
Number of frames	576
Color space	YV12
Frames per second	24
Source	https://vimeo.com/169766399
Source resolution	4K
Bitrate	93.707

Sandy beach with a green hill, approaching fog and grey clouds filmed using a handheld camera.

Figure 18: foggy_beach sequence, frame 25

A.4. hera

Sequence title	hera
Resolution	1920×1080
Number of frames	1023
Color space	YV12
Frames per second	24
Source	https://vimeo.com/184240706#t=83
Source resolution	FullHD
Bitrate	90.874

Part of a music clip containing different scenes on a seashore and in a backyard; shows grain and double-exposure effects.

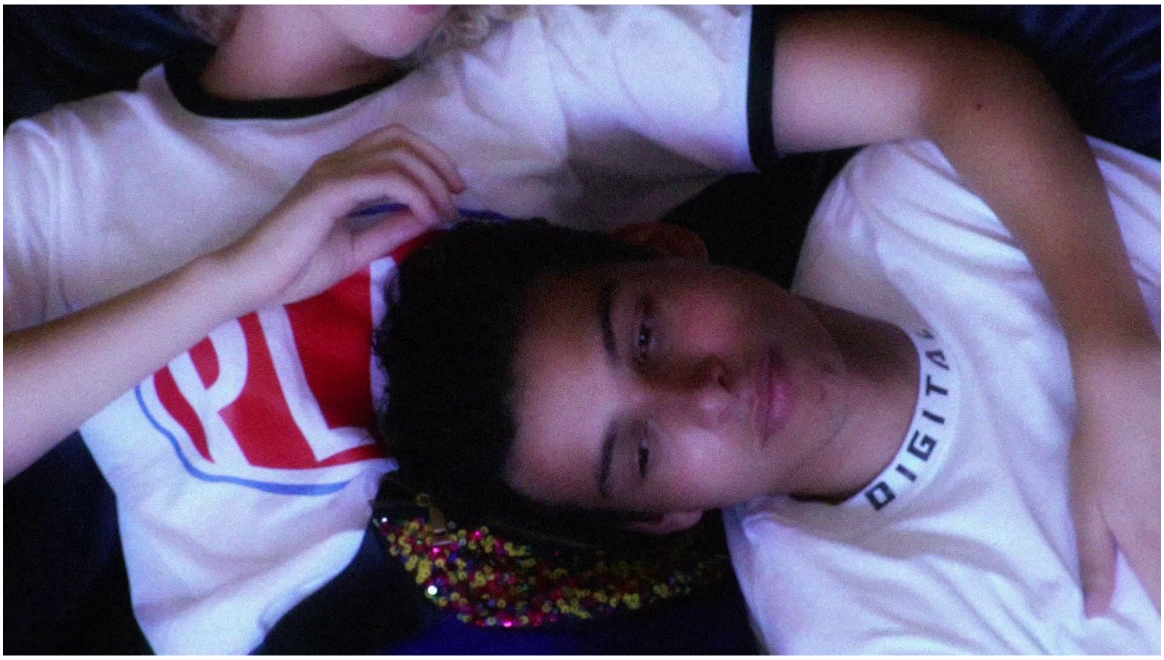


Figure 19: hera sequence, frame 25

A.5. television_studio

Sequence title	television_studio
Resolution	1920×1080
Number of frames	970
Color space	YV12
Frames per second	25
Source	https://vimeo.com/232795528#t=200
Source resolution	FullHD
Bitrate	92.161

Shot of a TV show in studio. Close-ups of participants change with a long-shot.

Figure 20: television_studio sequence, frame 0

A.6. witcher3_700

Sequence title	witcher3_700
Resolution	1920×1080
Number of frames	700
Color space	YV12
Frames per second	60
Source	https://media.xiph.org/video/derf/
Source resolution	FullHD
Bitrate	1492.992

Screen capture of a videogame quest in which the hero is fighting wolves in a forest.

Figure 21: witcher3_700 sequence, frame 100

B. SEQUENCE SELECTION

In “MSU Video Codecs Comparison 2016” we introduced a new technique for selecting test sequences. This technique create a data set containing representative sequences that encoders face in everyday situations. For this report we use the same method, but we updated the video database from which we sample videos.

We analyzed 384,946 videos at Vimeo, looking for 4K and FullHD examples with high bitrates (we chose 50 Mbps as our minimum). Figure 22 shows the bitrate distributions for last year’s data set and for the updated data set.

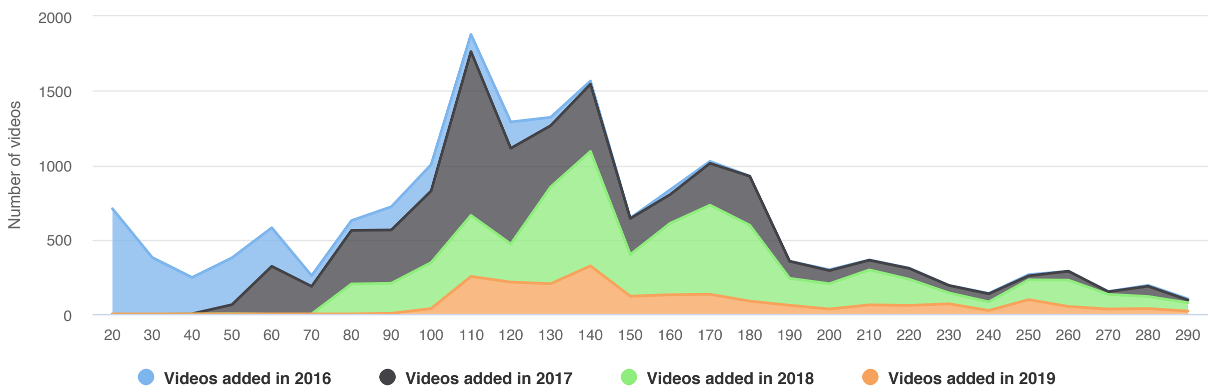


Figure 22: Bitrate distributions for video data set.

All videos were cut at scene change points to samples, with 1000 frames approximate length. Besides 2,585 samples from newly downloaded videos, we also used 6,628 samples from “MSU Video Codecs Comparison 2018”, 6,390 samples from “MSU Video Codecs Comparison 2017” and 2,913 samples from “MSU Video Codecs Comparison 2016”. Thus, our sample database for this year consisted of 18,516 items.

To evaluate spatial and temporal complexity, we encoded all samples using x264 with a constant quantization parameter (QP). We calculated the temporal and spatial complexity for each scene, defining spatial complexity as the average size of the I-frame normalized to the sample’s uncompressed frame size. Temporal complexity in our definition is the average size of the P-frame divided by average size of I-frame. [1]

In this year we slightly changed the temporal and spatial complexity calculation process by adding an additional preprocessing step. We use source videos from Vimeo, that was uploaded by users, so they all have different chroma subsampling which affects the results of videos evaluated complexity. Therefore to unificate the spatial and temporal complexity results of analysed videos, they all were converted to YUV 4:2:0 chroma subsample. Distribution of obtained samples compared to samples from previous codec comparisons is shown in Figure 23.

[1] C. Chen et. al., “A Subjective Study for the Design of Multi-resolution ABR Video Streams with the VP9 Codec,” 2016.

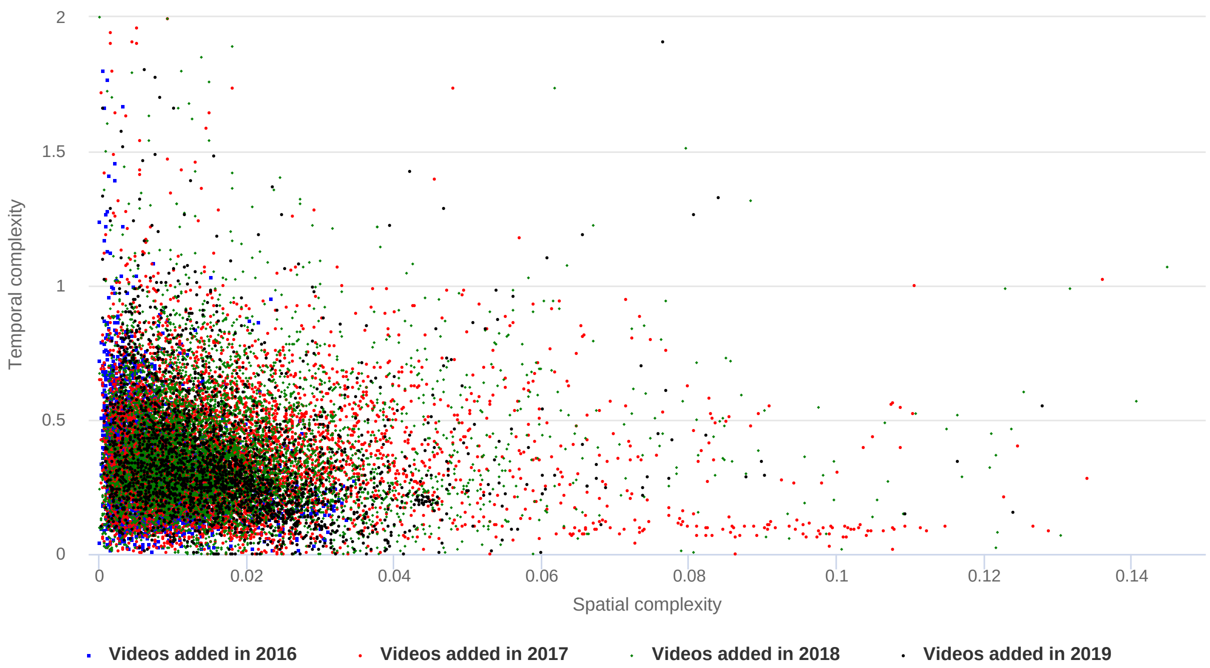


Figure 23: Distribution of obtained samples.

Figure 23 reveals that the new samples have a distribution similar to that of samples from "MSU Video Codecs Comparison 2017". We used the following process to prepare the data set.

We divided the video database into 6 clusters. To avoid completely changing the sequence list, we gave sequences from last year's FullHD data set 35 times greater weight than other sequences. For each cluster we selected the video sequence that's closest to the cluster's center and that has a license enabling derivatives and commercial use. Figure 24 shows the cluster boundaries and constituent sequences.

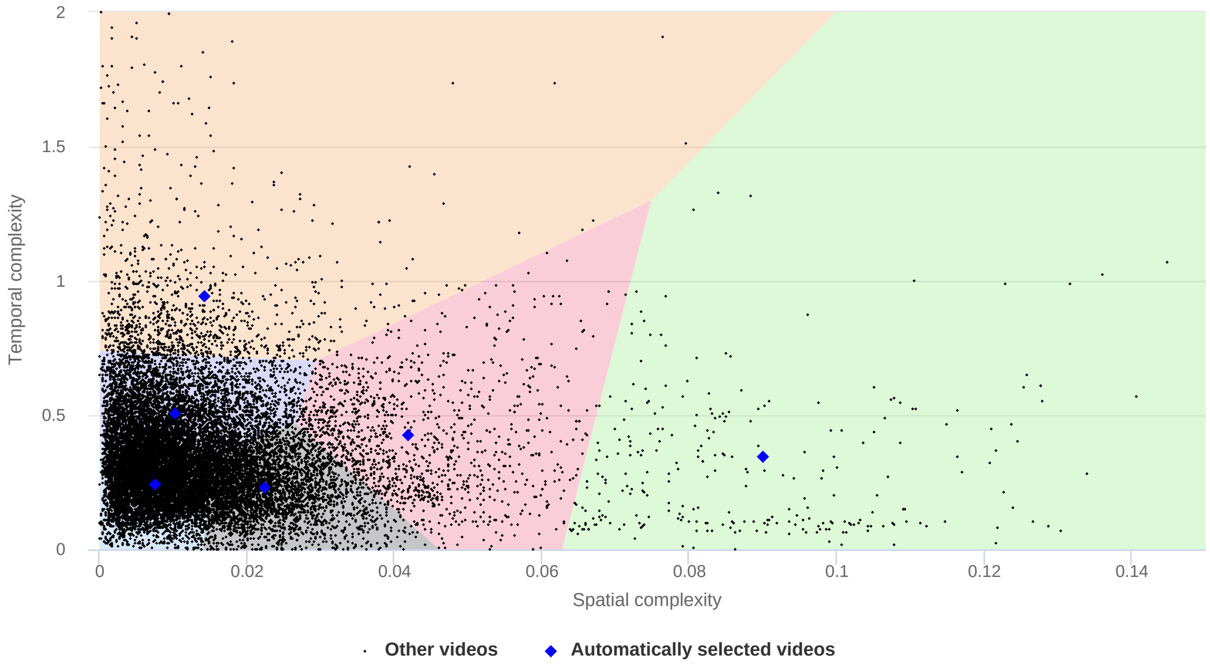


Figure 24: Segmentation of samples.

Some automatically chosen samples contain company names or have other copyright issues, so we removed them from their respective clusters and replaced them with other samples having a suitable license. Figure 25 illustrates these adjustments.

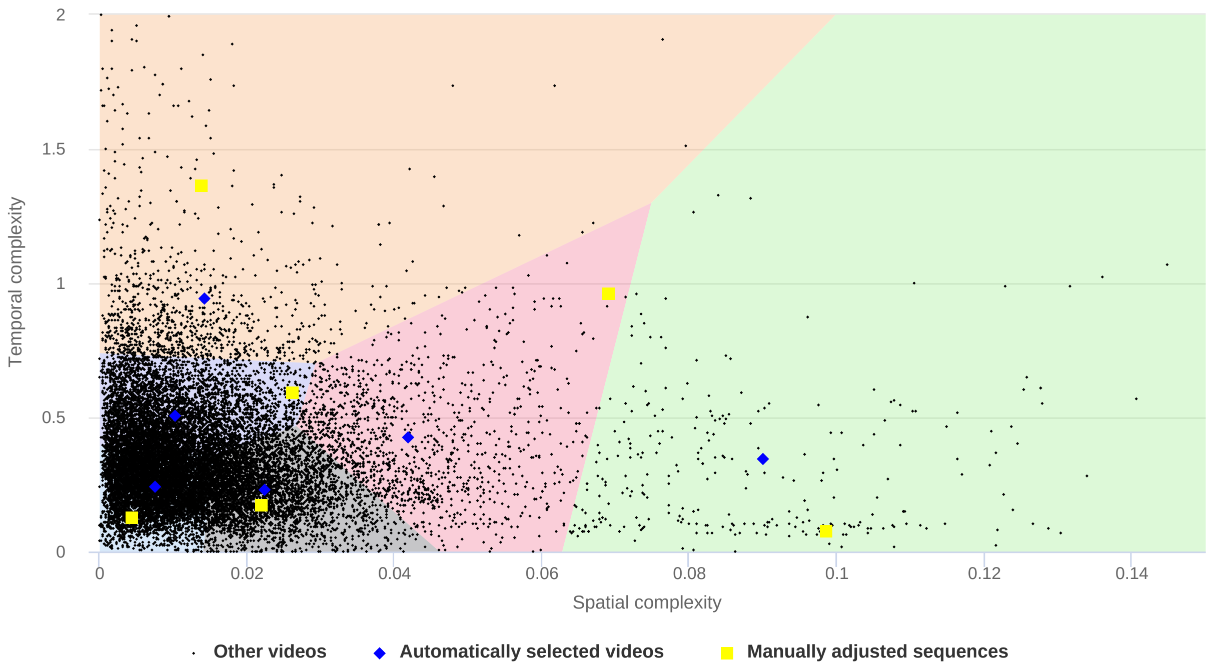


Figure 25: Adjustments to test data set.

Figure 26 shows the final distribution of sequences in the data set.

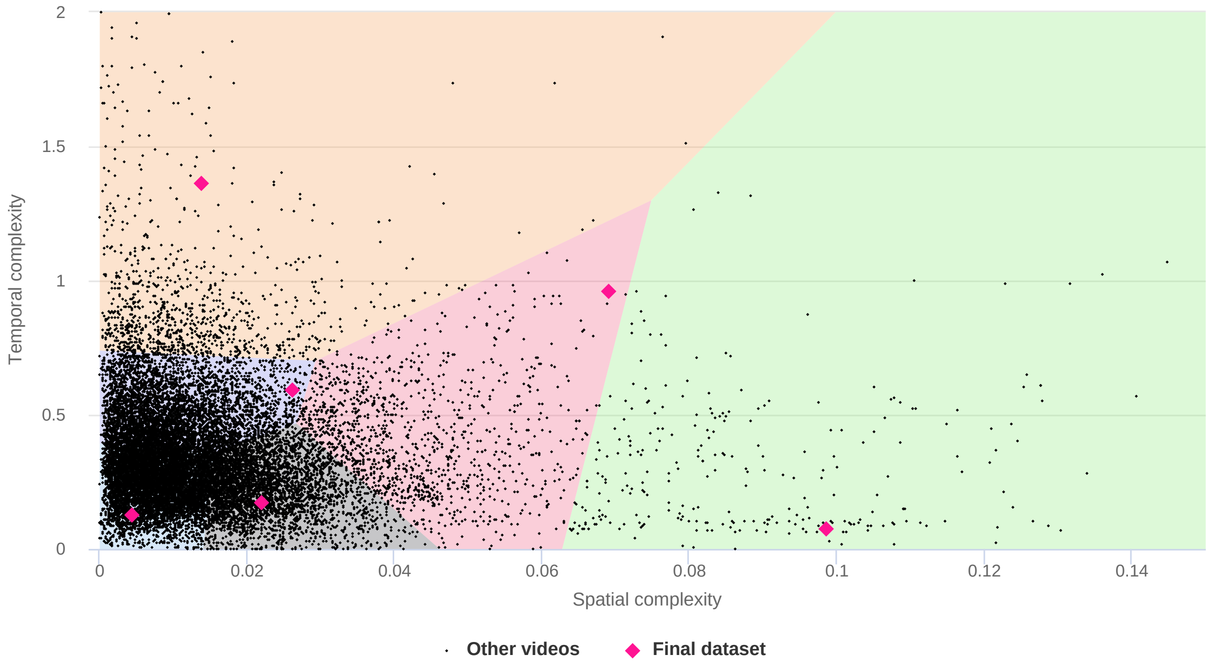


Figure 26: Distribution of sequences in final data set.

The new data set consists of 6 sequences: 5 new ones from Vimeo and 1 from xiph.org. The average bitrate for all sequences in the final set is 331.18 Mbps, median – 100.21 Mbps. The complete list of sequences for new data set appears in Appendix A.

C. CODECS

C.1. aom

Encoder title	aom
Version	1.0.0-errata1-avif
Developed by	AOMedia

Preset name	Encoder parameters
High Quality Encoding'	`aomenc.exe --width=%WIDTH% --height=%HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% --bit-depth=8 --end-usage=vbr --cpu-used=0 --target-bitrate=%BITRATE_KBPS% --ivf --threads=32 --tune=ssim -o %TARGET_FILE% %SOURCE_FILE%`

C.2. rav1e

Encoder title	rav1e
Version	0.2.0 (p20200127)
Developed by	rav1e

Preset name	Encoder parameters

High Quality Encoding'	`rav1e.exe --output %TARGET_FILE%.ivf --speed 10 --first-pass 2pass.bin -b %BITRATE_KBPS% %SOURCE_FILE%`
	`rav1e.exe --output %TARGET_FILE%.ivf --speed 1 --tiles 2 --second-pass 2pass.bin -b %BITRATE_KBPS% %SOURCE_FILE%`
	`move %TARGET_FILE%.ivf %TARGET_FILE%`

C.3. SVT-AV1

Encoder title	SVT-AV1
Version	0.8.1
Developed by	Open Visual Cloud

Preset name	Encoder parameters
High Quality Encoding'	`SvtAv1EncApp.exe -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -fps %FPS% -rc 1 -tbr %BITRATE_KBPS% -enc-mode 0 -b %TARGET_FILE%`

C.4. SVT-HEVC

Encoder title	SVT-HEVC
Version	1.4.3
Developed by	Open Visual Cloud

Preset name	Encoder parameters
High Quality Encoding'	`SvtHevcEncApp.exe -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -fps %FPS% -rc 1 -tbr %BITRATE_KBPS%000 -encMode 0 -b %TARGET_FILE%`

C.5. SVT-VP9

Encoder title	SVT-VP9
Version	0.1.0
Developed by	Open Visual Cloud

Preset name	Encoder parameters
High Quality Encoding'	`SvtVp9EncApp.exe -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -fps %FPS% -rc 1 -tbr %BITRATE_KBPS%000 -enc-mode 0 -b %TARGET_FILE%`

C.6. x264

Encoder title	x264
Version	core:157 r2969 d4099dd
Developed by	x264 Developer Team

Preset name	Encoder parameters
Reference	`x264-r2969-d4099dd.exe --tune ssim --preset veryslow --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
High Quality Encoding'	`x264-r2969-d4099dd.exe --preset placebo --me umh --merange 32 --keyint infinite --tune ssim --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o NUL x264-r2969-d4099dd.exe --preset placebo --me umh --merange 32 --keyint infinite --tune ssim --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o %TARGET_FILE%`

C.7. x265

Encoder title	x265
Version	3.2+15-04db2bfee5d6
Developed by	MulticoreWare, Inc.

Preset name	Encoder parameters
High Quality Encoding'	`x265_64_8bit.exe --preset placebo --tune ssim --keyint -1 --tune ssim --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%` `x265_64_8bit.exe --preset placebo --tune ssim --keyint -1 --tune ssim --pass 3 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%` `x265_64_8bit.exe --preset placebo --tune ssim --keyint -1 --tune ssim --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`

D. FIGURE EXPLANATION

The main charts in this comparison are classic RD curves (quality/bitrate graphs) and relative-bitrate/relative-time charts. Additionally, we also used bitrate-handling charts (the ratio of real to target bitrates) and per-frame quality charts.

D.1. RD Curves

The RD charts show variation in codec quality by bitrate or file size. For this metric, a higher value presumably indicates better quality.

D.2. Relative-Bitrate/Relative-Time Charts

Relative-bitrate/relative-time charts show the average bitrate's dependence on relative encoding time for a fixed-quality output. The y-axis shows the ratio of a codec's bitrate under test to the reference codec's bitrate for a fixed quality. A lower value (that is, a higher the value on the graph) indicates a better-performing codec. For example, a value of 0.7 means the codec can encode the sequence in a file that's 30% smaller what the reference codec produces.

The x-axis shows the relative encoding time. Larger values indicate a slower codec. For example, a value of 2.5 means the codec works 2.5 times slower, on average, than the reference codec.

D.3. Graph Example

Figure 27 shows a situation where these graphs can be useful. In the top-left graph, the "Green" codec clearly produces better quality than the "Black" codec. On the other hand, the top-right graph shows that the "Green" codec is slightly slower. Relative-bitrate/relative-time graphs can be useful in precisely these situations: the bottom graph clearly shows that one codec is slower but yields higher visual quality, whereas the other codec is faster but yields lower visual quality.

Owing to these advantages, we frequently use relative-bitrate/relative-time graphs in this report because they assist in evaluating the codecs in the test set, especially when the number of codecs is large.

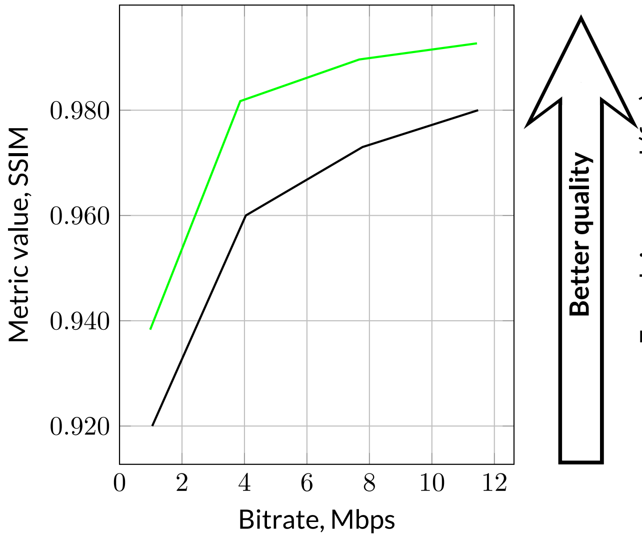
A more detailed description of how we prepared these graphs appears below.

D.4. Bitrate Ratio for the Same Quality

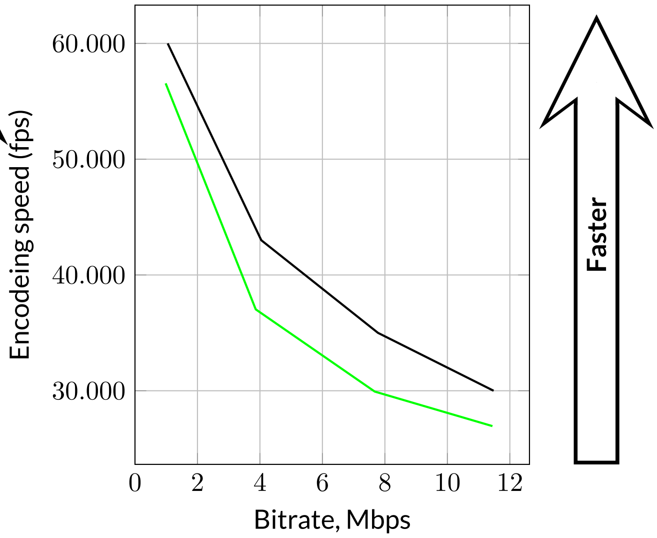
The first step in computing the average bitrate ratio for a fixed quality is to invert the axes of the bitrate/quality graph (see Figure 28b). All further computations use the inverted graph.

The second step involves averaging the interval over which the quality axis is chosen. The averaging is only over those segments for which both codecs yield results. This limitation is due to the difficulty of developing extrapolation methods for classic RD curves; nevertheless, even linear methods are acceptable when interpolating RD curves.

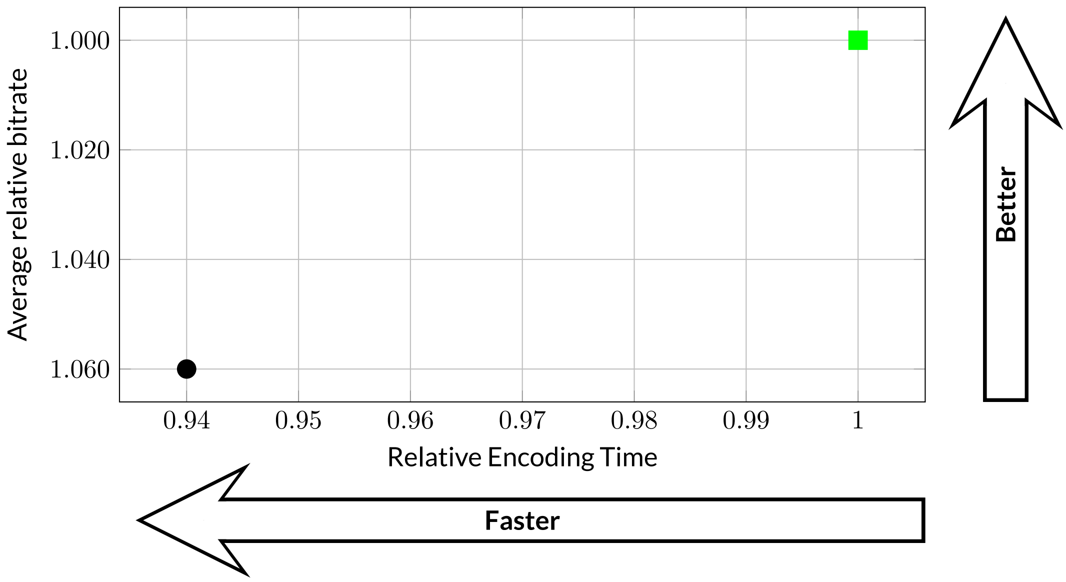
The final step is calculation of the area under the curves in the chosen interpolation segment and determination of their ratio (see Figure 28c). This result is an average bitrate ratio at a fixed quality for the two codecs. When



(a) RD curve. “Green” codec is better!

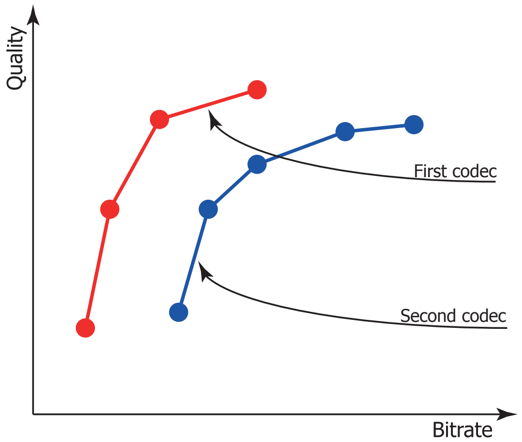


(b) Encoding speed (frames per second). “Green” codec is slower!

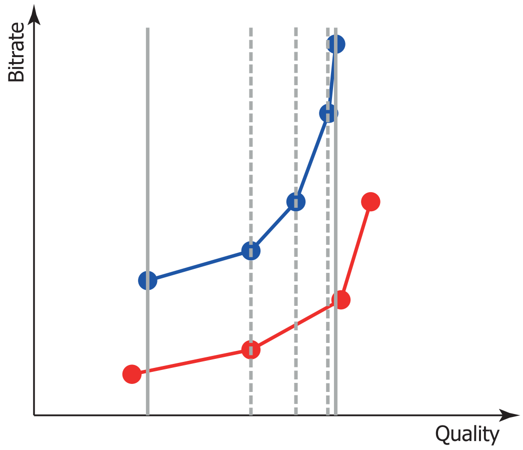


(c) Integral situation with codecs. This plot shows the situation more clearly

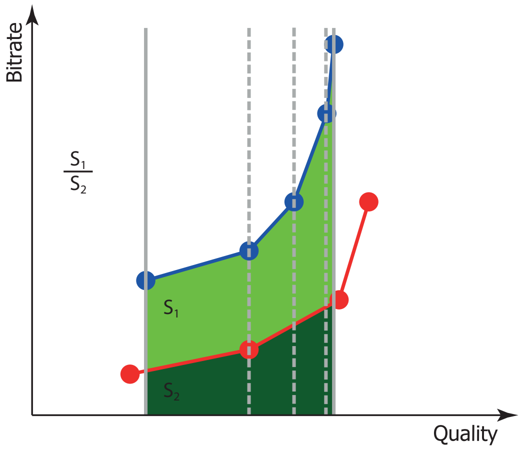
Figure 27: Speed/Quality trade-off example



(a) Source RD curves



(b) Axes' inversion and averaging interval choosing



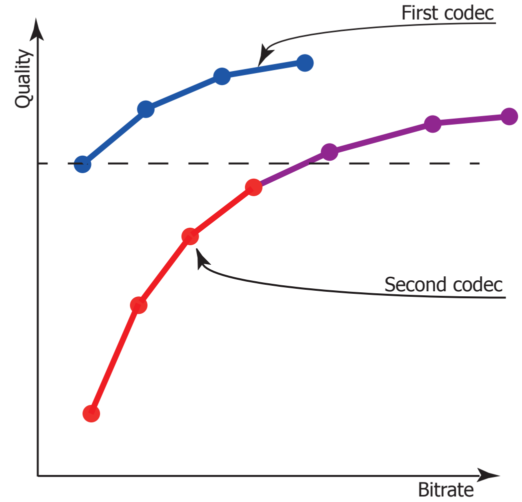
(c) Areas under curves ratio

Figure 28: Average bitrate ratio computation

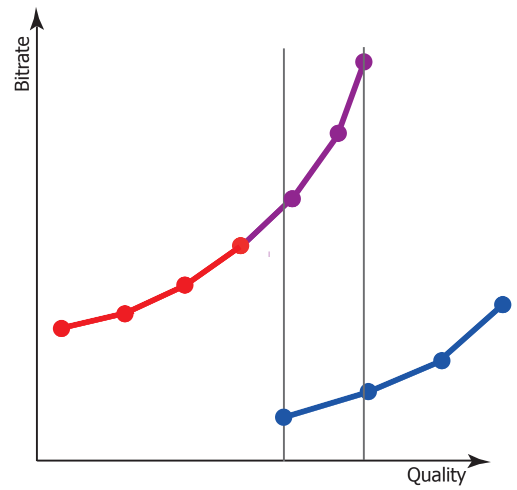
considering more than two codecs, one of is defined as a reference codec, and the quality of the others is compared with that of the reference.

D.4.1. When RD Curves Fail to Cross the Quality Axis

If no segment exists for which two codecs both produce encoding results, we measured the results for additional higher and/or lower bitrates. The schematic example (Figure 29) shows that the results for these extra bitrates (purple) cross with codec two and enable a comparison with codec one.



(a) Source RD curves, purple color indicates results for extra bitrates

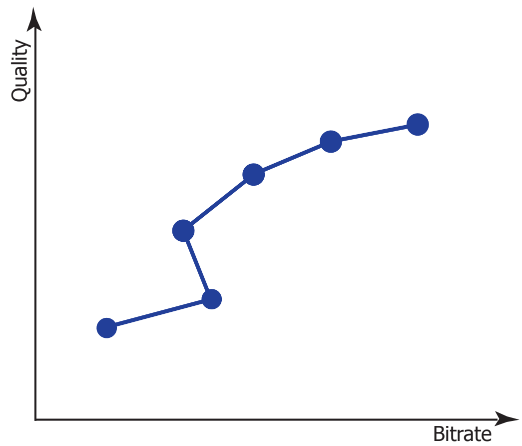


(b) Axes' inversion and averaging interval choosing

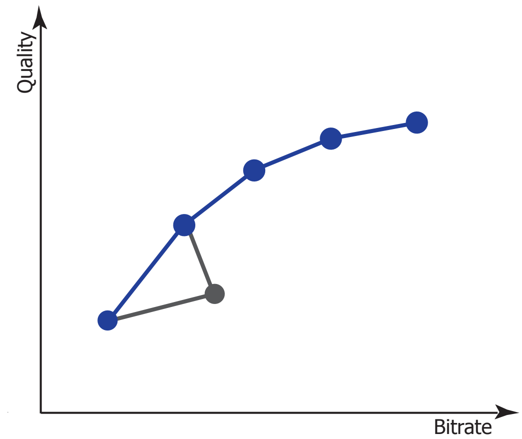
Figure 29: Measuring codec on additional bitrates to make it cross with other codecs over the quality axis.

D.4.2. When RD Curves Are Non-monotonic

Sometimes, especially on complex videos, the encoding results for neighboring bitrates vary greatly owing to the codec's operating characteristics. This situation leads to a non-monotone RD curve, which we process as follows: for each point, use the next point at the target bitrate that has greater or equal quality. This technique yields the reduced monotonic curve, which appears in the example of Figure 30.



(a) Non-monotonic RD-curve.



(b) Points that were used to calculate integral.

Figure 30: Processing non-monotonic RD-curves.

D.5. Relative Quality Analysis

Although most figures in this report provide codec scores relative to a reference encoder (i.e., x264), the "Relative Quality Analysis" sections provide the bitrate ratio at a fixed quality score (see Section D.4) for each codec pair. This approach may be useful when comparing codec A relative with codec B only.

Below is a simplified example table showing the average bitrate ratio, given a fixed quality, for just two codecs.

	A	B
A	100% ☺	75% ☹
B	134% ☹	100% ☺

Confidence ☹ 0% — 😐 50% — ☺ 100%

Table 4: Example of average bitrate ratio for a fixed quality table

Consider column “B”, row “A” of the table, which contains the value 75%. This number should be interpreted in the following way: the average bitrate for Codec B at a fixed quality is 75% less than that for codec A. The icon in the cell depicts the confidence of this estimate. If projections of RD curves on the quality axis (see Figure 28) have large common areas, the cell contains a happy icon. If this overlapping area is small, and thus the bitrate-score calculation is unreliable, the cell contains a sad icon.

Plots of the average bitrate ratio for a fixed quality are visualizations of these tables. Each line in the plot depicts values from one column of the corresponding table.

E. OBJECTIVE-QUALITY METRIC DESCRIPTION

E.1. SSIM (Structural Similarity)

We used the YUV-SSIM objective-quality metric in this report to assess the quality of encoded video sequences. We compute YUV-SSIM as the weighted average of SSIM values for each channel individually (Y-SSIM, U-SSIM and V-SSIM):

$$\text{YUV-SSIM} = \frac{4\,\text{Y-SSIM} + \text{U-SSIM} + \text{V-SSIM}}{6}. \tag{1}$$

Below is a brief description of SSIM computation.

E.1.1. Brief Description

Wang, et al.[2] published the original paper on SSIM. This paper available at http://ieeexplore.ieee.org/iel5/83/28667/01284395.pdf. The SSIM author homepage is http://www.cns.nyu.edu/~lcv/ssim/

The main idea that underlies the structural-similarity (SSIM) index is comparison of the distortion of three image components:

- Luminance
- Contrast
- Structure

The final formula, after combining these comparisons, is

$$\mathbf{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x + \mu_y + C_1)(\sigma_x + \sigma_y + C_2)}, \tag{2}$$

where

$$\mu_x = \sum_{i=1}^{N} \omega_i x_i, \tag{3}$$

$$\sigma_x = \sqrt{\sum_{i=1}^{N} \omega_i (x_i - \mu_x)}, \tag{4}$$

$$\sigma_{xy} = \sum_{i=1}^{N} \omega_i (x_i - \mu_x)(y_i - \mu_y). \tag{5}$$

Finally, $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$, where L is the dynamic range of the pixel values (e.g. 255 for 8-bit greyscale images), and $K1, K2 \ll 1$.

We used $K_1 = 0.01$ and $K_2 = 0.03$ were used for the comparison presented in this report, and we filled the matrix with a value "1" in each position to form a filter for the results map.

For our implementation, one SSIM value corresponds to two sequences. The value is in the range $[-1, 1]$, with higher values being more desirable (a value of 1 corresponds to identical frames). One advantage of the SSIM

[2] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh and Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, No. 4, April 2004.

metric is that it better represents human visual perception than does PSNR. SSIM is more complex, however, and takes longer to calculate.

E.1.2. Examples

Figure 31 shows an example SSIM result for an original and processed (compressed with lossy compression) image. The value of 0.9 demonstrates that the two images are very similar.

(a) Original (b) Compressed (c) SSIM

Figure 31: SSIM example for compressed image

Figure 32 depicts various distortions applied to the original image, and Figure 33 shows SSIM values for these distortions.

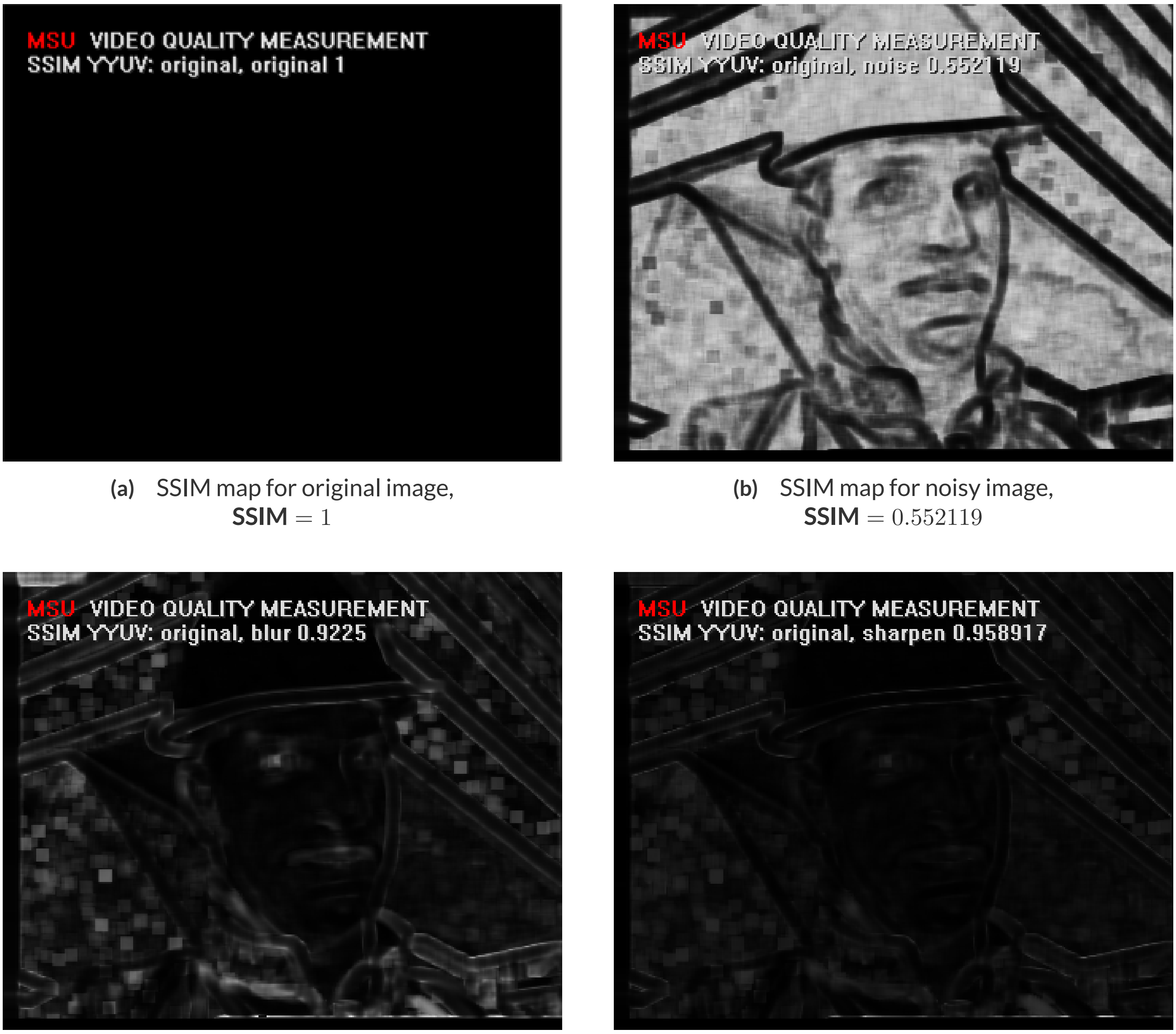
(a) Original image

(b) Image with added noise

(c) Blurred image

(d) Sharpen image

Figure 32: Examples of processed images



(a) SSIM map for original image, SSIM $= 1$

(b) SSIM map for noisy image, SSIM $= 0.552119$

(c) SSIM map for blurred image, SSIM $= 0.9225$

(d) SSIM map for sharpen image, SSIM $= 0.958917$

Figure 33: SSIM values for original and processed images

E.1.3. Measurement method

We used the MSU Video Quality Measurement Tool (VQMT) to calculate objective metrics for the encoded streams. The tool can be downloaded or purchased at `http://compression.ru/video/quality_measure/vqmt_download.html#start`.

Run the command

```
vqmt -in "{original_yuv}" IYUV {width}×{height} -in "decoded_yuv" IYUV
{width}×{height} metrics_list -subsampling -json -json_file "{json_filename}" -threads
3
```

where input_yuv is the encoded stream name, width and height are the size of encoded stream in pixels, metrics_list is a list of metrics to measure (e.g., "-metr ssim_precise YYUV -metr ssim_precise UYUV -metr ssim_precise VYUV"), and json_filename is the name of the output file containing the metric results.

F. ABOUT THE GRAPHICS & MEDIA LAB VIDEO GROUP

Graphics & Media Lab
Video Group

The Graphics & Media Lab Video Group is part of the Computer Science Department of Lomonosov Moscow State University. The Graphics Group began at the end of 1980's, and the Graphics & Media Lab was officially founded in 1998. The main research avenues of the lab include areas of computer graphics, computer vision and media processing (audio, image and video). A number of patents have been acquired based on the lab's research, and other results have been presented in various publications.

The main research avenues of the Graphics & Media Lab Video Group are video processing (pre- and post-, as well as video analysis filters) and video compression (codec testing and tuning, quality metric research and codec development).

The main achievements of the Video Group in the area of video processing include:

- High-quality industrial filters for format conversion, including high-quality deinterlacing, high-quality frame rate conversion, new, fast practical super resolution and other processing tools.
- Methods for modern television sets, such as a large family of up-sampling methods, smart brightness and contrast control, smart sharpening and more.
- Artifact removal methods, including a family of denoising methods, flicking removal, video stabilization with frame edge restoration, and scratch, spot and drop-out removal.
- Application-specific methods such as subtitle removal, construction of panorama images from video, video to high-quality photo conversion, video watermarking, video segmentation and practical fast video deblur.

The main achievements of the Video Group in the area of video compression include:

- Well-known public comparisons of JPEG, JPEG-2000 and MPEG-2 decoders, as well as MPEG-4 and annual H.264 codec testing; codec testing for weak and strong points, along with bug reports and codec tuning recommendations.
- Video quality metric research; the MSU Video Quality Measurement Tool and MSU Perceptual Video Quality Tool are publicly available.
- Internal research and contracts for modern video compression and publication of MSU Lossless Video Codec and MSU Screen Capture Video Codec; these codecs have one of the highest available compression ratios.

The Video Group has also worked for many years with companies like Intel, Samsung and RealNetworks.

In addition, the Video Group is continually seeking collaboration with other companies in the areas of video processing and video compression.

E-mail: video@graphics.cs.msu.ru