

MSU Video Codecs Comparison 2020



Graphics & Media Lab
Video Group

Part III: 4K & 10-bit Content, Objective

Evaluation

Video group head Dr. Dmitriy Vatolin

Project head Dr. Dmitriy Kulikov

Measurements & analysis Dr. Mikhail Erofeev,
Anastasia Antsiferova,
Egor Sklyarov,
Alexander Yakovenko,
Nickolay Safonov

Free version

Codecs:

H.265

- BVC1
- MC HEVC/H.265 Encoder
- Reference x265
- SVT-HEVC
- sz265
- x265

AV1

- aom
- SVT-AV1

Other

- SIF Codec
- SVT-VP9
- x264

1. REPORT VERSIONS

	Free version	Enterprise version
Use cases	Offline (1 fps) (partially)	Offline (1 fps), Online (30 fps)
Per-sequence-results	1 of 12 sequences (only Universal use case and only 8-bit content)	All 12 sequences for all use cases (in interactive charts)
Metric: YUV-SSIM, VMAF (overall results), PSNR (overall results)	✓	✓
Other objective metrics (Y-VMAF(0.6.1 for 4K), Y-SSIM, U-SSIM, V-SSIM, YUV-PSNR, Y-PSNR, U-PSNR, V-PSNR)	✗	✓
Per-frame metrics results (in HTML report)	✗	✓
Description of video sequences	✓	✓
Download links for video sequences	✗	✓
Codec info (developer, version number, website link)	✓	✓
Encoders presets description	✗	✓
PDF report	48 pages	70 pages
HTML report	29 interactive charts	15000+ interactive charts

Contents

1	Report Versions	2
2	Acknowledgments	5
3	Overview	6
3.1	Sequences	6
3.2	Codecs	6
4	Objectives and Testing Rules	7
5	4K Universal (1fps)	8
5.1	RD Curves	8
5.2	Encoding Speed	9
5.3	Speed/Quality Trade-Off	11
5.4	4K Universal (1fps) YUV-SSIM	13
5.5	4K Universal (1fps) YUV-PSNR (avg. MSE)	13
5.6	4K Universal (1fps) YUV-PSNR (avg. log)	14
5.7	4K Universal (1fps) Y-VMAF (v0.6.1 for 4K)	15
6	4K Fast (30fps)	16
6.1	RD Curves	16
6.2	Encoding Speed	17
6.3	Speed/Quality Trade-Off	18
6.4	4K Fast (30fps) YUV-SSIM	20
6.5	4K Fast (30fps) YUV-PSNR (avg. MSE)	20
6.6	4K Fast (30fps) YUV-PSNR (avg. log)	21
6.7	4K Fast (30fps) Y-VMAF (v0.6.1 for 4K)	22
7	Conclusion	23
7.1	Overall YUV-SSIM (for all use cases)	23
7.2	Overall YUV-PSNR (avg. MSE) (for all use cases)	24
7.3	Overall YUV-PSNR (avg. log) (for all use cases)	24
7.4	Overall Y-VMAF (v0.6.1 for 4K) (for all use cases)	25
A	Sequences	26
A.1	backgammon	26
A.2	ballerine	27
A.3	california_coast	28
A.4	crowd_run	29
A.5	dron_view	30
A.6	ducks_take_off	31
A.7	news	32

A.8	waterfall	33
B	Codecs	34
C	Video Selection	35
D	Figure Explanation	37
D.1	RD Curves	37
D.2	Relative-Bitrate/Relative-Time Charts	37
D.3	Graph Example	37
D.4	Bitrate Ratio for the Same Quality	37
D.4.1	When RD Curves Fail to Cross the Quality Axis	39
D.4.2	When RD Curves Are Non-monotonic	40
E	Objective-Quality Metric Description	41
E.1	SSIM (Structural Similarity)	41
E.1.1	Brief Description	41
E.1.2	Examples	42
E.1.3	Measurement method	44
E.2	PSNR (Peak Signal-to-Noise Ratio)	45
E.2.1	PSNR (avg. MSE)	45
E.2.2	PSNR (avg. log)	45
F	About the Graphics & Media Lab Video Group	46

2. ACKNOWLEDGMENTS

The Graphics & Media Lab Video Group would like to thank the following companies for providing the codecs and settings used in this report:

- Bytedance Inc.
- MainConcept GmbH
- MulticoreWare, Inc.
- RayShaper
- SIF Codec LLC
- x264 project

We are also grateful to these companies for their help and technical support during the tests.

3. OVERVIEW

3.1. Sequences

	Sequence	Number of frames	Frame rate	Resolution
1.	backgammon	1058	30	3840×1920
2.	ballerine	1310	24	3840×1594
3.	california_coast	1055	30	3840×2160
4.	crowd_run	500	50	3840×2160
5.	dron_view	1117	25	3840×2160
6.	ducks_take_off	500	50	3840×2160
7.	news	1445	30	3840×2160
8.	waterfall	687	25	3840×2160

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.

3.2. Codecs

Codec	Developer	Version	OS
aom	AOMedia	2.0.0-287-g2aa13c436	Windows
BVC1	Bytedance Inc.	V1	Windows
MC HEVC/H.265 Encoder	MainConcept GmbH	HEVC SDK 12.2	Windows
Reference x265	MulticoreWare, Inc.	3.3+21-6bb2d88029c2	Windows
SIF Codec	SIF Codec LLC	1.95	Windows
SVT-AV1	Open Visual Cloud	v0.8.3	Windows
SVT-HEVC	Open Visual Cloud	v1.4.3	Windows
SVT-VP9	Open Visual Cloud	v0.2.0	Windows
sz265	RayShaper	v1.1.0	Linux
x264	x264 project	0.161.3018 db0d417	Windows
x265	MulticoreWare, Inc.	3.4+20-g06c52b0fd	Windows

Table 2: Short codecs' descriptions

Brief descriptions of the codecs used in our comparison appear in Table 2. We used x265 as a good-quality HEVC reference codec. Appendix B provides detailed descriptions of all codecs in our comparison.

4. OBJECTIVES AND TESTING RULES

In this report we use objective assessment methods to compare the encoding quality of recent HEVC encoders as well as encoders implementing other standards. This effort employed 12 video sequences at 4K resolution. 8 video sequences had 8-bit color depth and 4 video sequences had 10-bit color depth. Note: we used technically 10-bit videos for evaluation, but the colours in the videos themselves do not utilize wide luminance range (they are not HDR). A detailed description of the selection process appears in Appendix C.

Our comparison consists of two parts, corresponding to various encoder use cases: fast encoding and universal encoding. For each use case we offered the codec developers the option to provide encoding parameters for our tests. If they declined to provide any, we either used the same parameters from our prior study or, if none were available, did our best to choose good parameters ourselves. Nevertheless, the parameters had to satisfy a minimum speed requirements for their respective use case:

- Fast—30fps
- Universal—1fps

For measurements we used computers 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 YUV-SSIM metric (see Appendix E.1) as a main objective indicator, and other metrics (PSNR, VMAF) as an additional quality metrics. Our team is constantly researching the area of objective video quality metrics to find good solutions for large comparisons.

According to many requests, we also show VMAF results as a subjective quality-oriented indicator. Recently our team investigated tuning for VMAF¹, so the possibility of encoders tuning for increasing VMAF scores need to be taken into account.

As an overall score indication, an approach we called BSQ-rate was used². As it was described in the paper, this method shows more accurate results on complex cases of codecs performance comparison than BD-rate.

¹A. Zvezdakova, S. Zvezdakov, D. Kulikov, D. Vatolin, "Hacking VMAF with Video Color and Contrast Distortion," 2019.

²A. Zvezdakova, D. Kulikov, S. Zvezdakov, D. Vatolin, "BSQ-rate: a new approach for video-codec performance comparison and drawbacks of current solutions," 2020.

5. 4K UNIVERSAL (1FPS)

5.1. RD Curves

Judging from the mean quality scores (computed using the method described in Section D), first place in the quality competition goes to **BVC1**, second place goes to **MC HEVC/H.265 Encoder**, and third place to **sz265**.

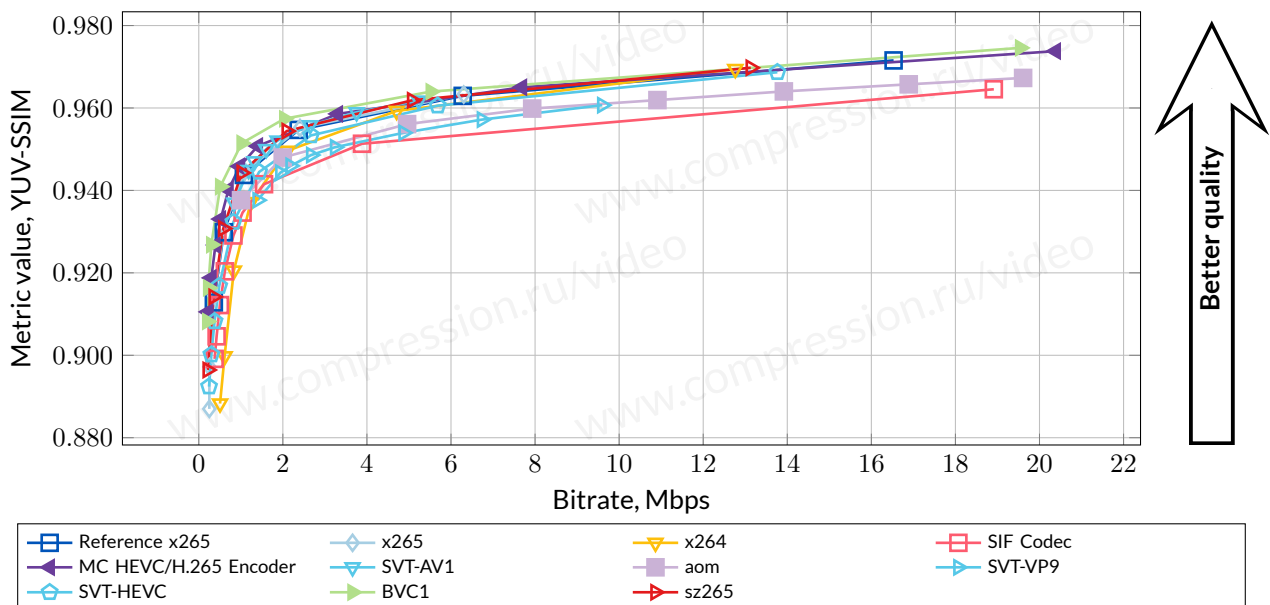
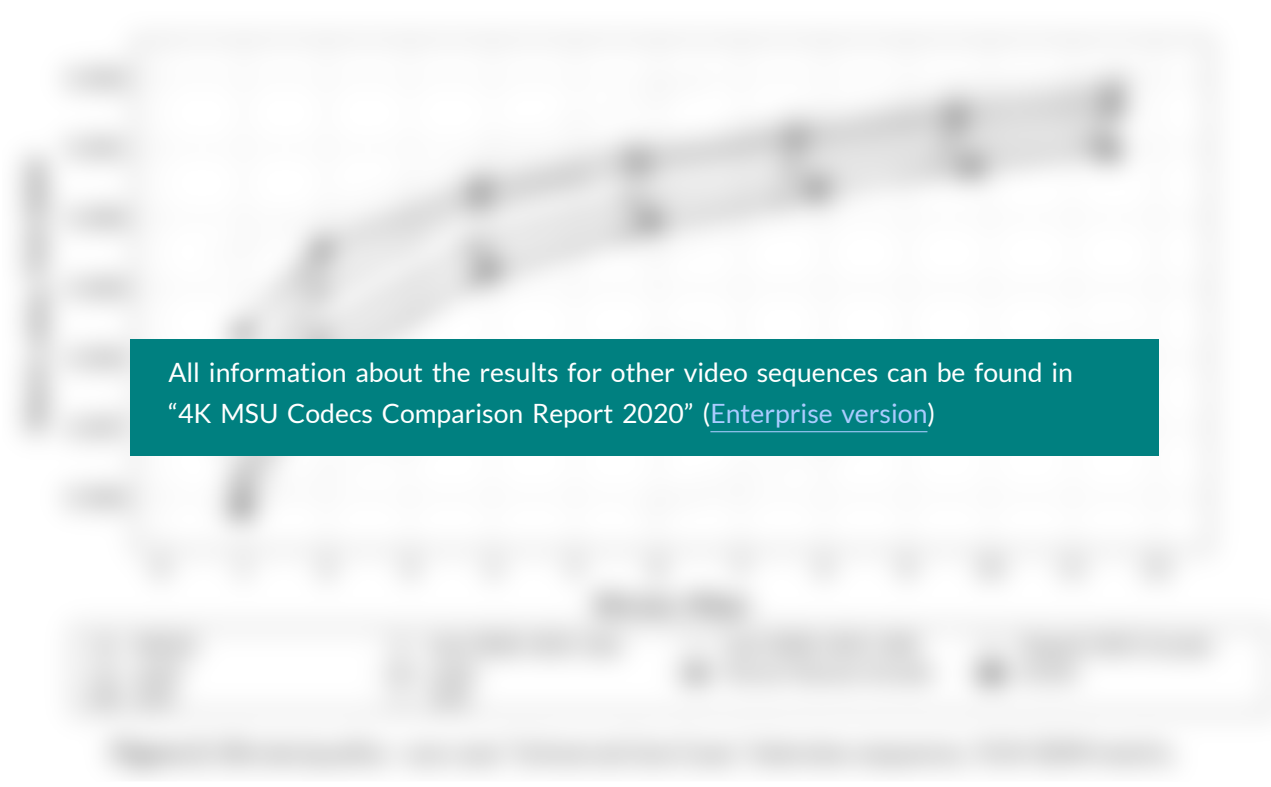


Figure 1: Bitrate/quality—use case “4K Universal (1fps),” *backgammon* sequence, YUV-SSIM metric.

The explanation of measuring on additional bitrates is presented in Section D.4.



All information about the results for other video sequences can be found in
“4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

5.2. Encoding Speed

Judging from the mean speed scores (computed using the method described in Section D), first place in the speed competition goes to **MC HEVC/H.265 Encoder**, second place goes to **aom** and **sz265**, and third place to **Reference x265**.

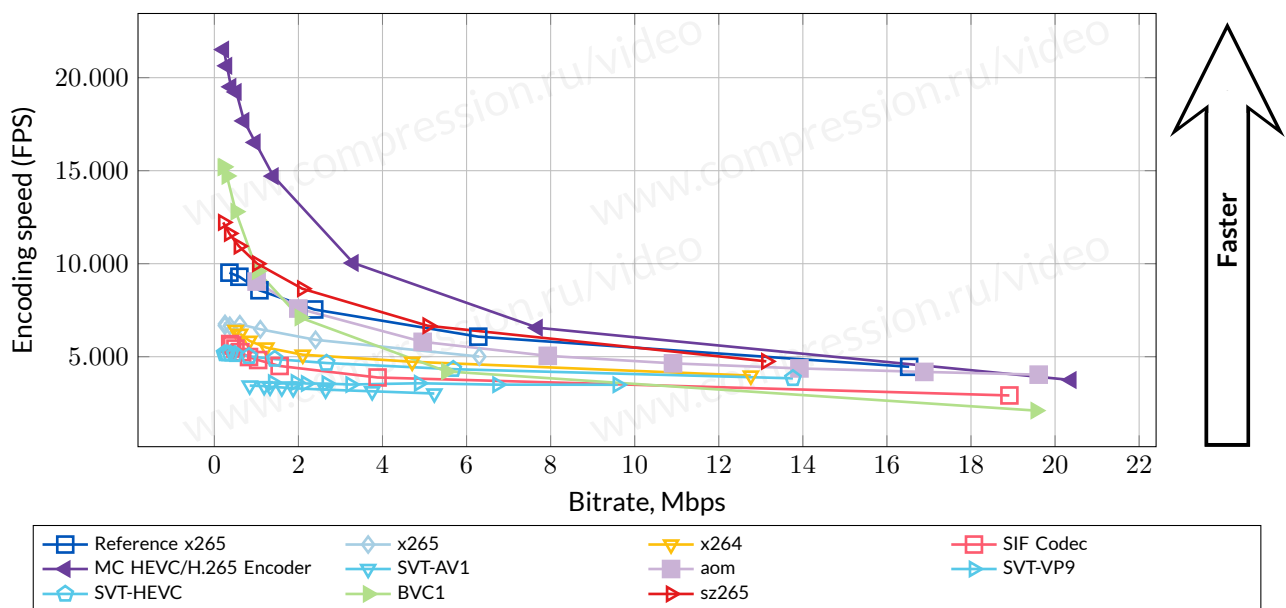


Figure 2: Encoding speed—use case “4K Universal (1fps),” *backgammon* sequence.

The explanation of measuring on additional bitrates is presented in Section [D.4](#).

All information about the results for other video sequences can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

5.3. 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 x265 as the reference codec, we normalized all scores to the x265 scores.

There are two Pareto-optimal encoders: **BVC1** and **MC HEVC/H.265 Encoder**.

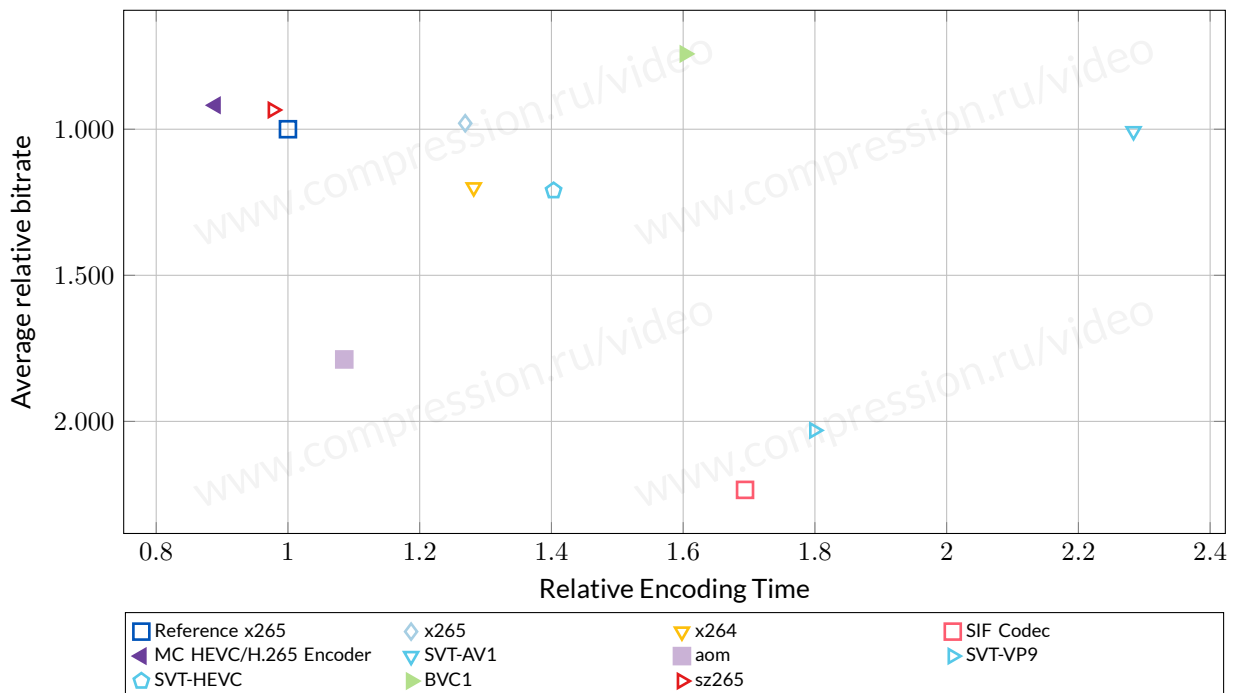


Figure 3: Speed/Quality Trade-Off—use case “4K Universal (1fps),” *backgammon* sequence, YUV-SSIM metric.

Speed-quality chart over all sequences can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

5.4. 4K Universal (1fps) YUV-SSIM

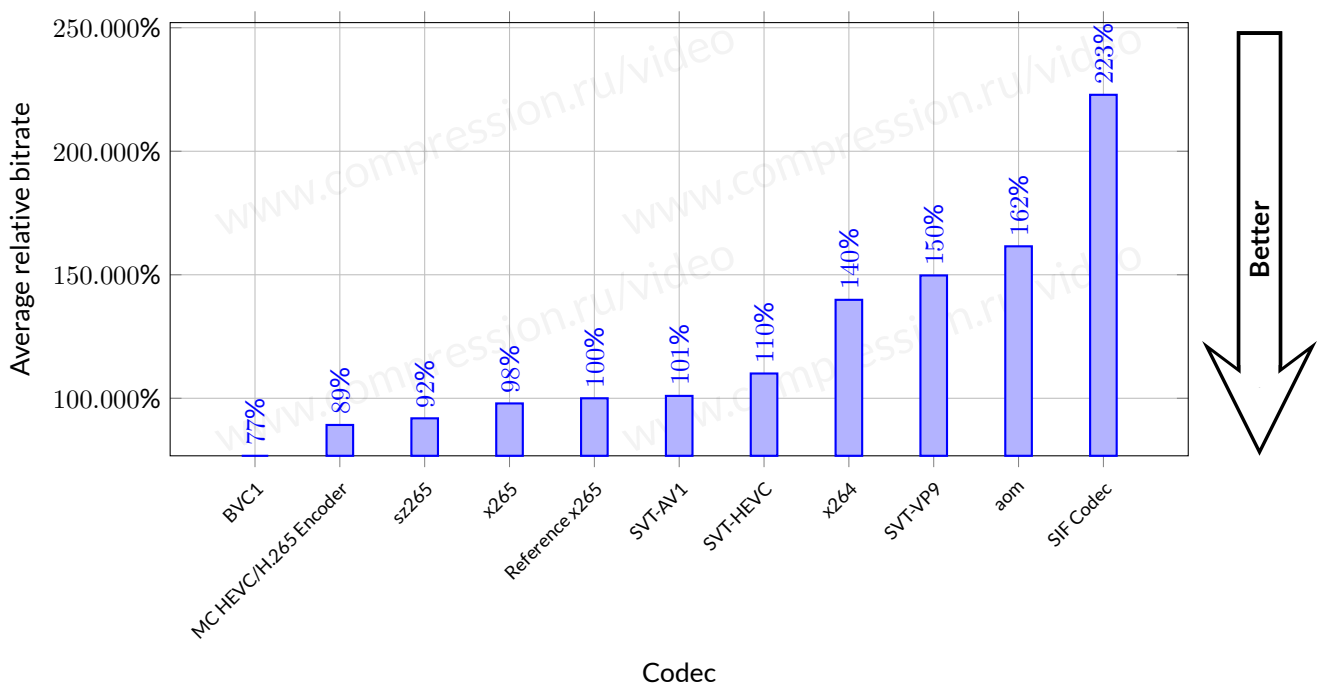


Figure 4: Average bitrate ratio for a fixed quality—use case “4K Universal (1fps),” all sequences, YUV-SSIM metric.

5.5. 4K Universal (1fps) YUV-PSNR (avg. MSE)

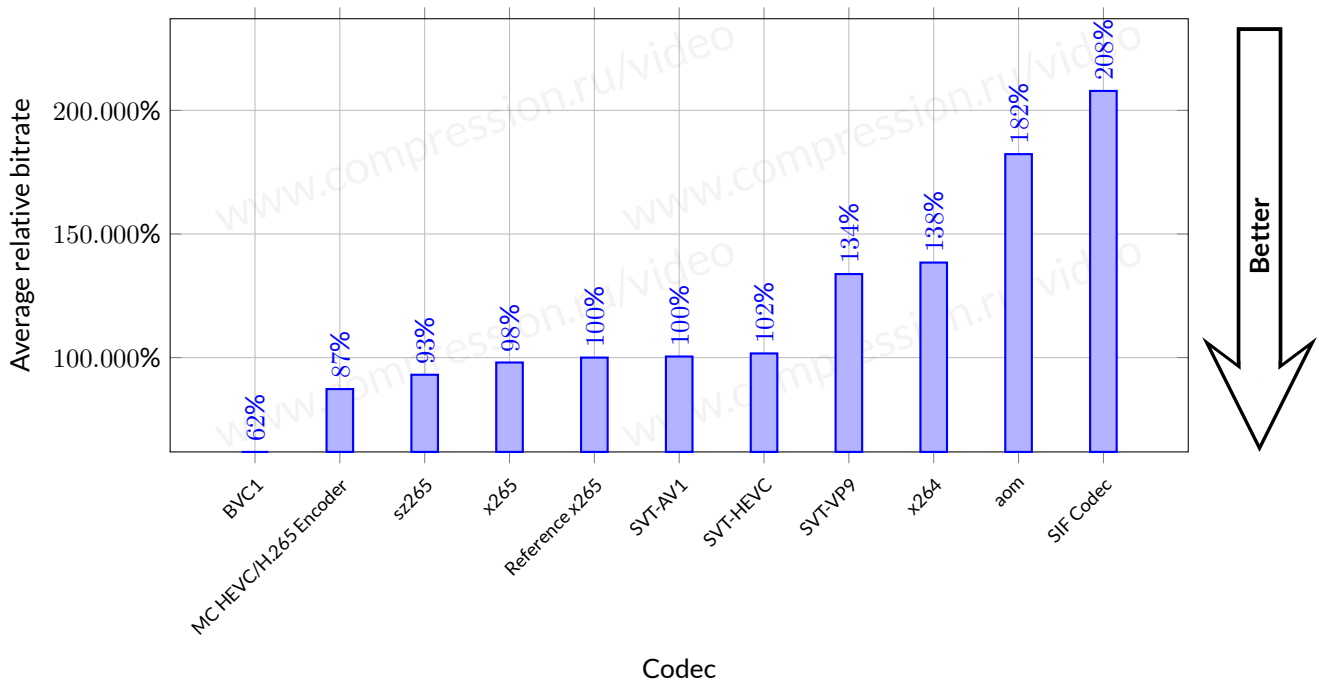


Figure 5: Average bitrate ratio for a fixed quality—use case “4K Universal (1fps),” all sequences, YUV-PSNR (avg. MSE) metric.

5.6. 4K Universal (1fps) YUV-PSNR (avg. log)

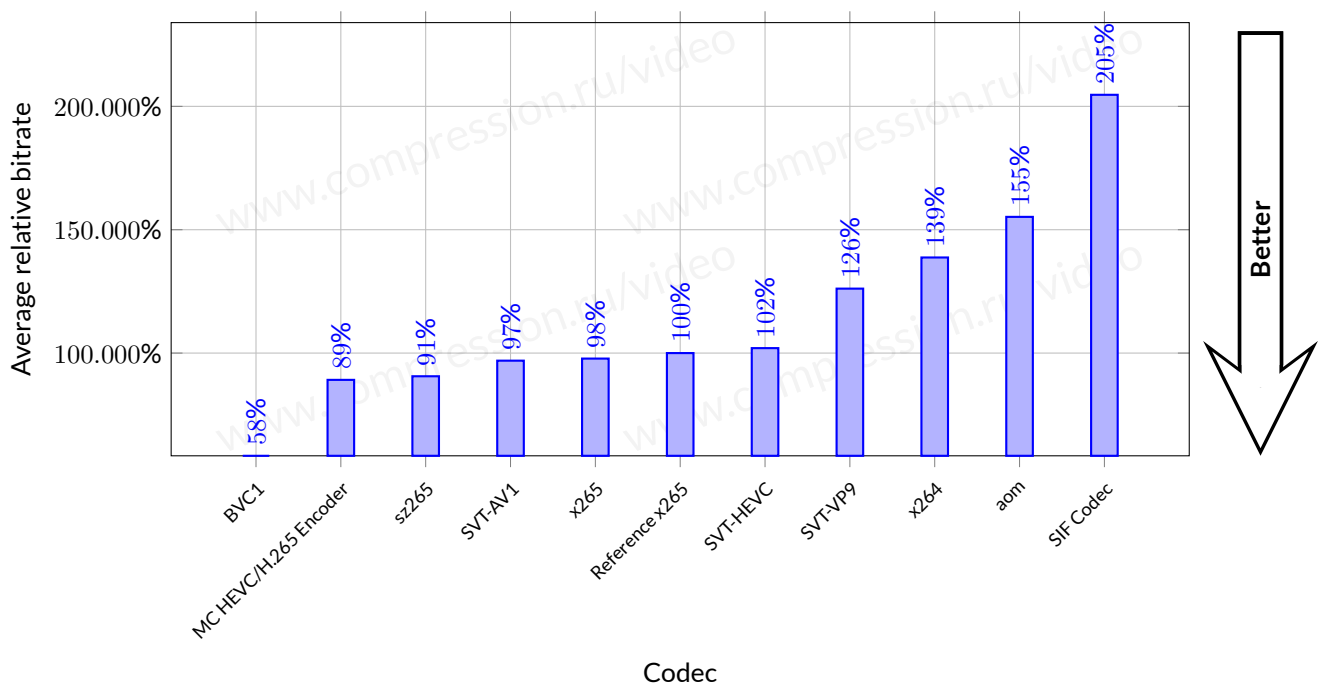


Figure 6: Average bitrate ratio for a fixed quality—use case “4K Universal (1fps),” all sequences, YUV-PSNR (avg. log) metric.

5.7. 4K Universal (1fps) Y-VMAF (v0.6.1 for 4K)

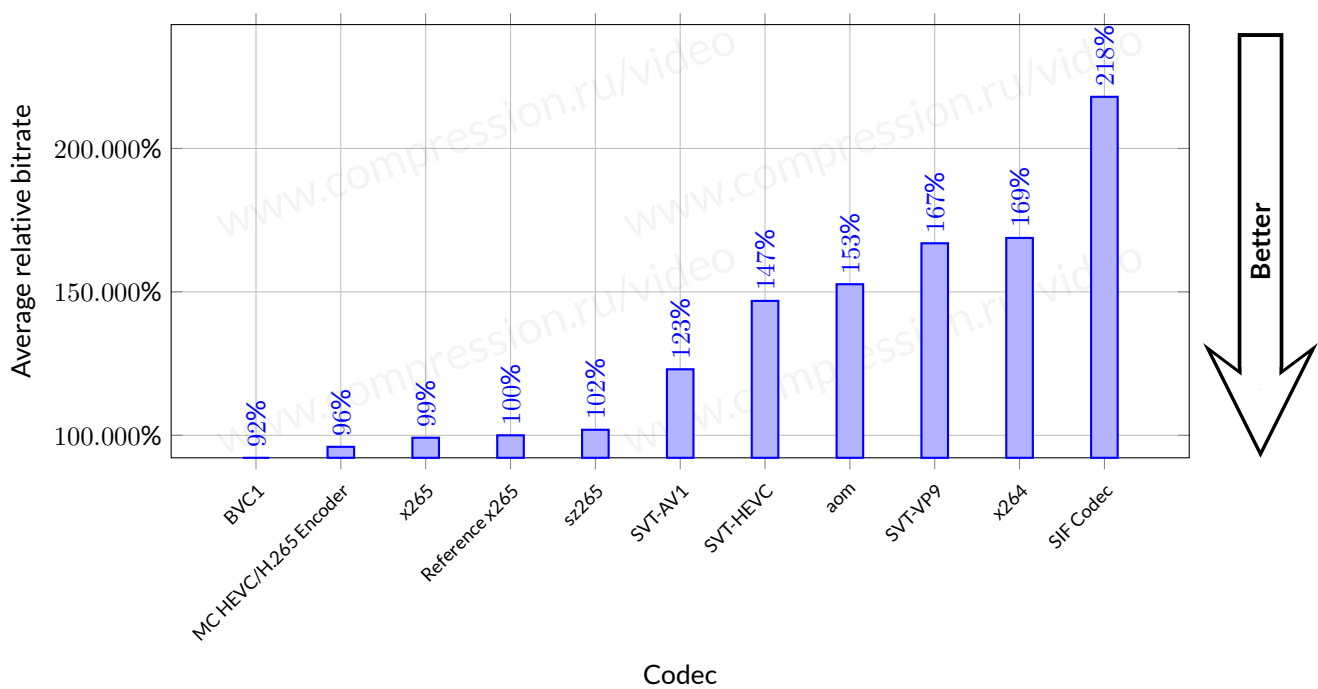


Figure 7: Average bitrate ratio for a fixed quality—use case “4K Universal (1fps),” all sequences, Y-VMAF (v0.6.1 for 4K) metric.

6. 4K FAST (30FPS)

6.1. RD Curves

Judging from the mean quality scores (computed using the method described in Section D), first place in the quality competition goes to **BVC1**, second place goes to **MC HEVC/H.265 Encoder**, and third place to **sz265**.

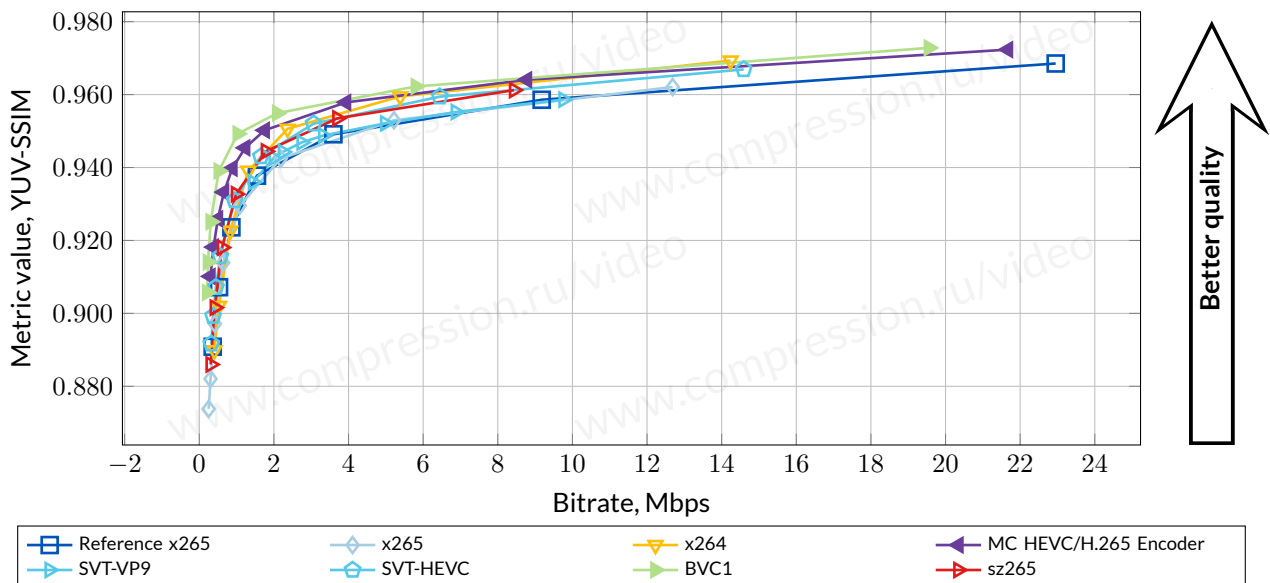


Figure 8: Bitrate/quality—use case “4K Fast (30fps),” *backgammon* sequence, YUV-SSIM metric.

The explanation of measuring on additional bitrates is presented in Section D.4.

All information about the results for other video sequences can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

6.2. Encoding Speed

Judging from the mean speed scores (computed using the method described in Section D), first place in the speed competition goes to **MC HEVC/H.265 Encoder**, second place goes to **SVT-HEVC**, and third place to **Reference x265**.

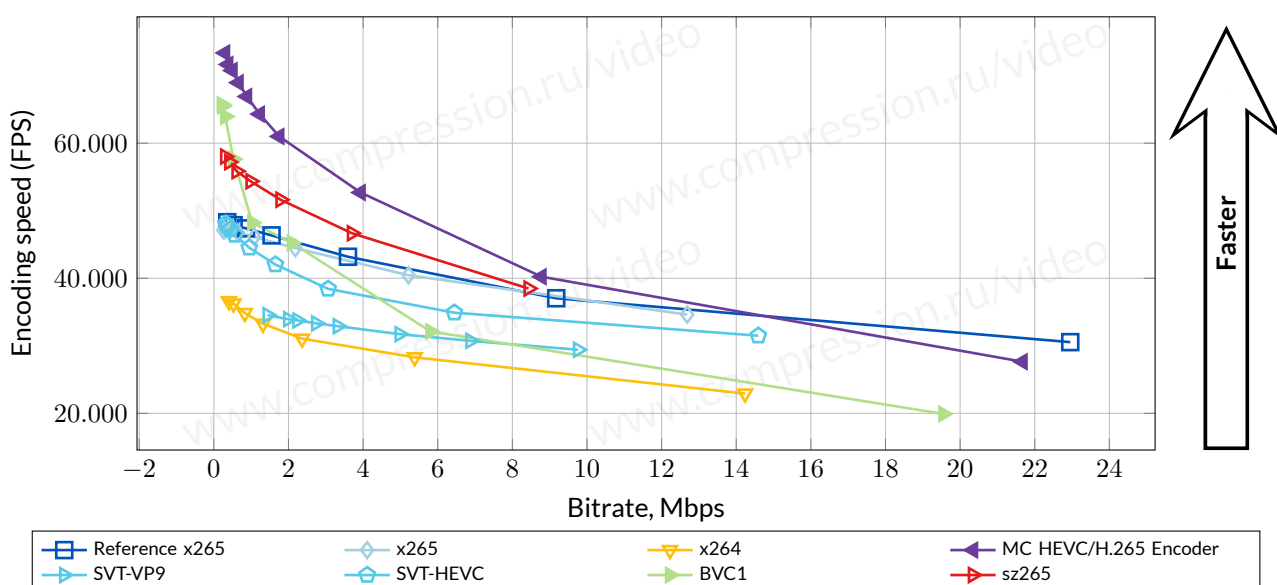


Figure 9: Encoding speed—use case “4K Fast (30fps),” *backgammon* sequence.

The explanation of measuring on additional bitrates is presented in Section [D.4](#).

All information about the results for other video sequences can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

6.3. 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 x265 as the reference codec, we normalized all scores to the x265 scores.

There are two Pareto-optimal encoders: **BVC1** and **MC HEVC/H.265 Encoder**.

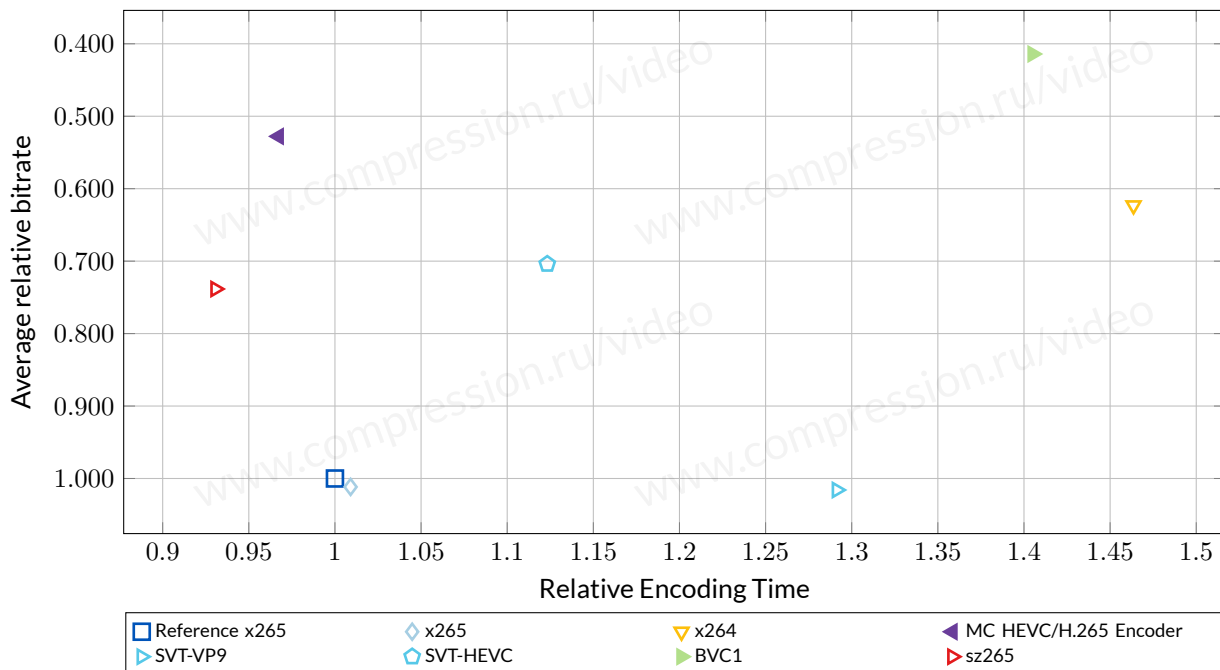


Figure 10: Speed/Quality Trade-Off—use case “4K Fast (30fps),” *backgammon* sequence, YUV-SSIM metric.

Speed-quality chart over all sequences can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

6.4. 4K Fast (30fps) YUV-SSIM

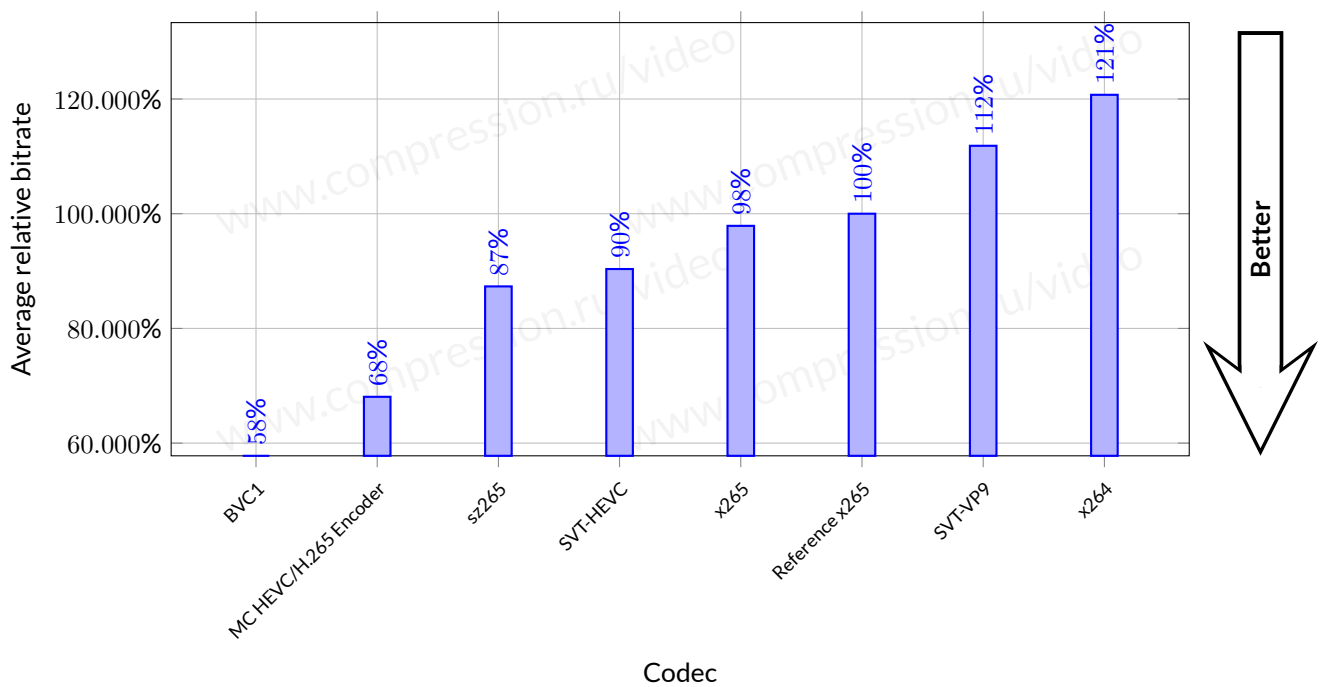


Figure 11: Average bitrate ratio for a fixed quality—use case “4K Fast (30fps),” all sequences, YUV-SSIM metric.

6.5. 4K Fast (30fps) YUV-PSNR (avg. MSE)

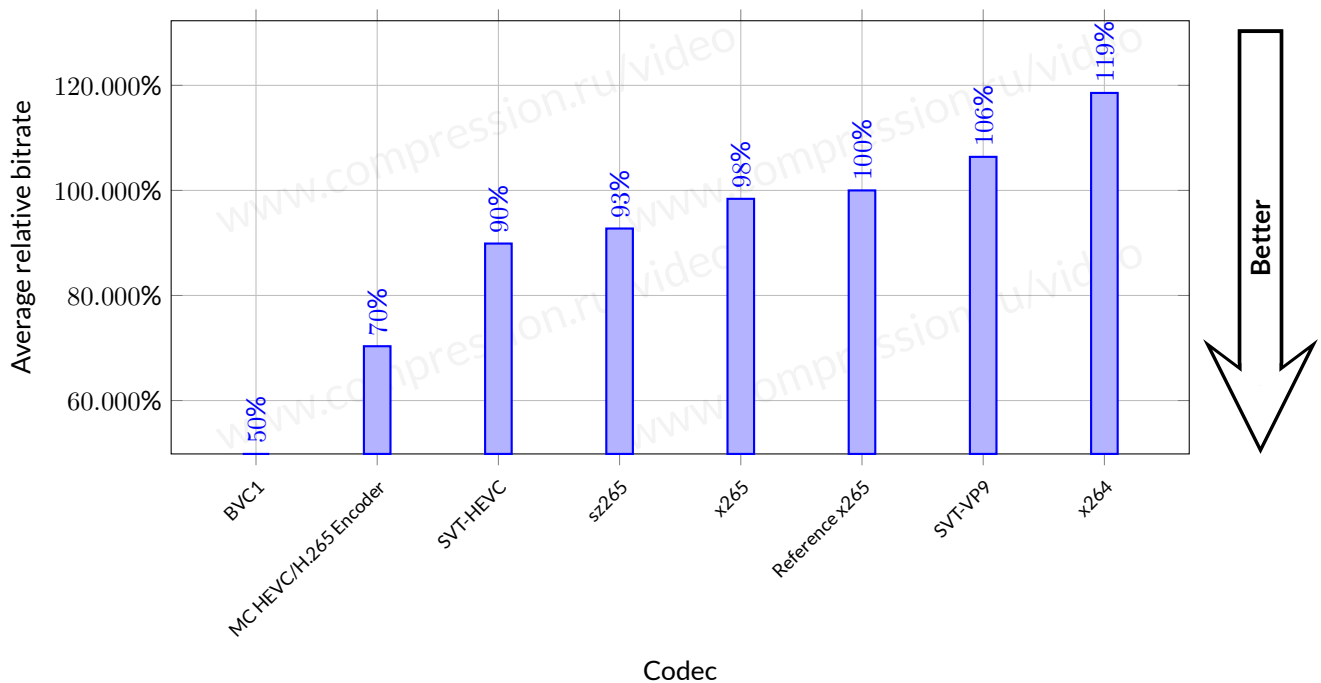


Figure 12: Average bitrate ratio for a fixed quality—use case “4K Fast (30fps),” all sequences, YUV-PSNR (avg. MSE) metric.

6.6. 4K Fast (30fps) YUV-PSNR (avg. log)

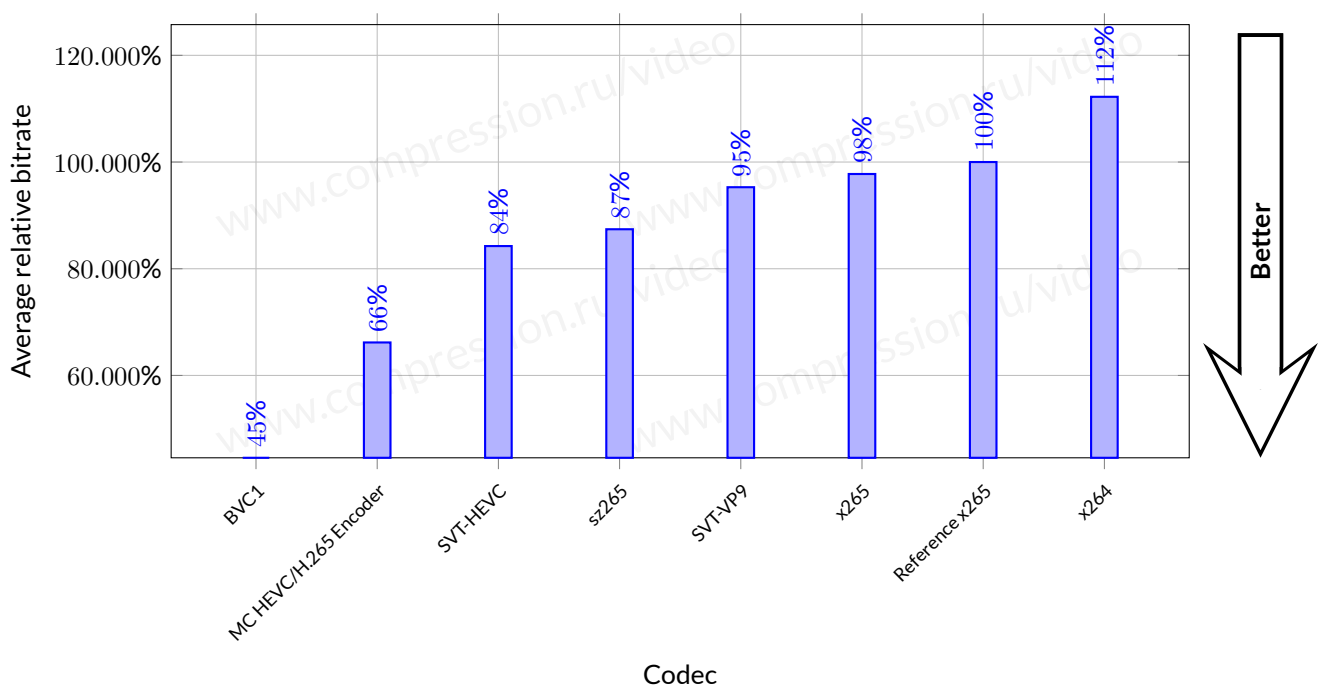


Figure 13: Average bitrate ratio for a fixed quality—use case “4K Fast (30fps),” all sequences, YUV-PSNR (avg. log) metric.

6.7. 4K Fast (30fps) Y-VMAF (v0.6.1 for 4K)

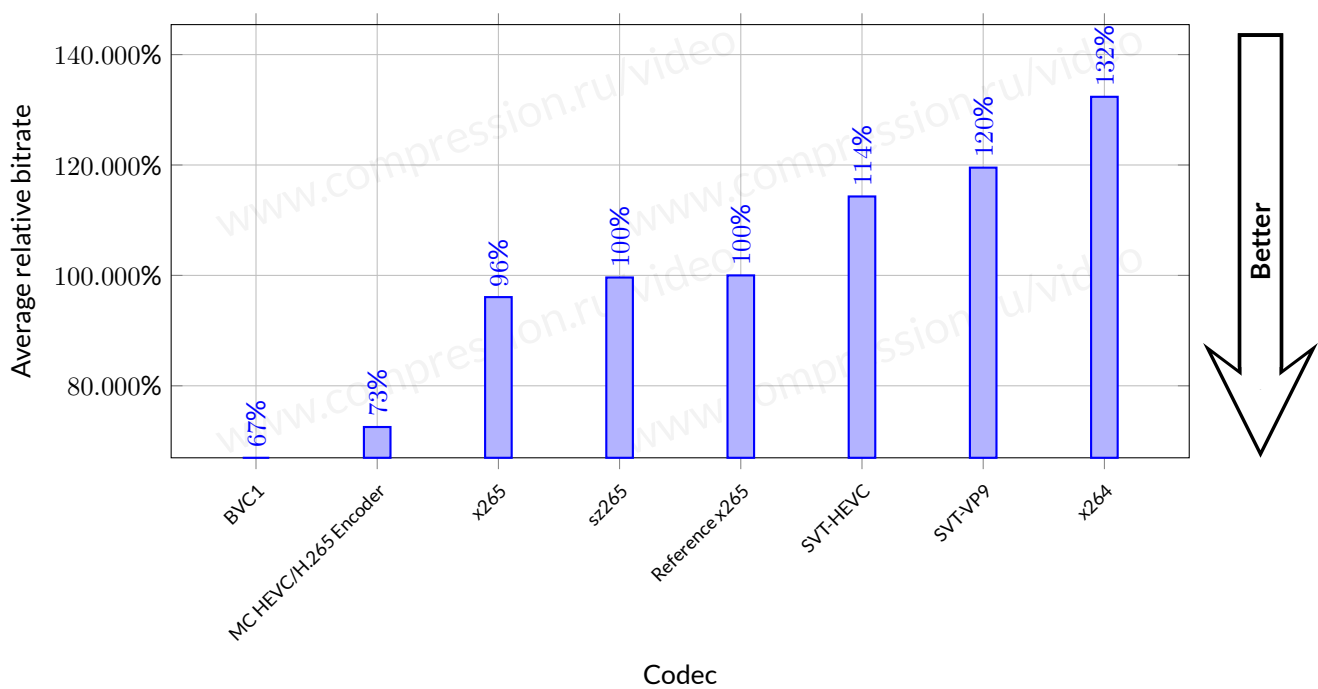


Figure 14: Average bitrate ratio for a fixed quality—use case “4K Fast (30fps),” all sequences, Y-VMAF (v0.6.1 for 4K) metric.

7. CONCLUSION

7.1. Overall YUV-SSIM (for all use cases)

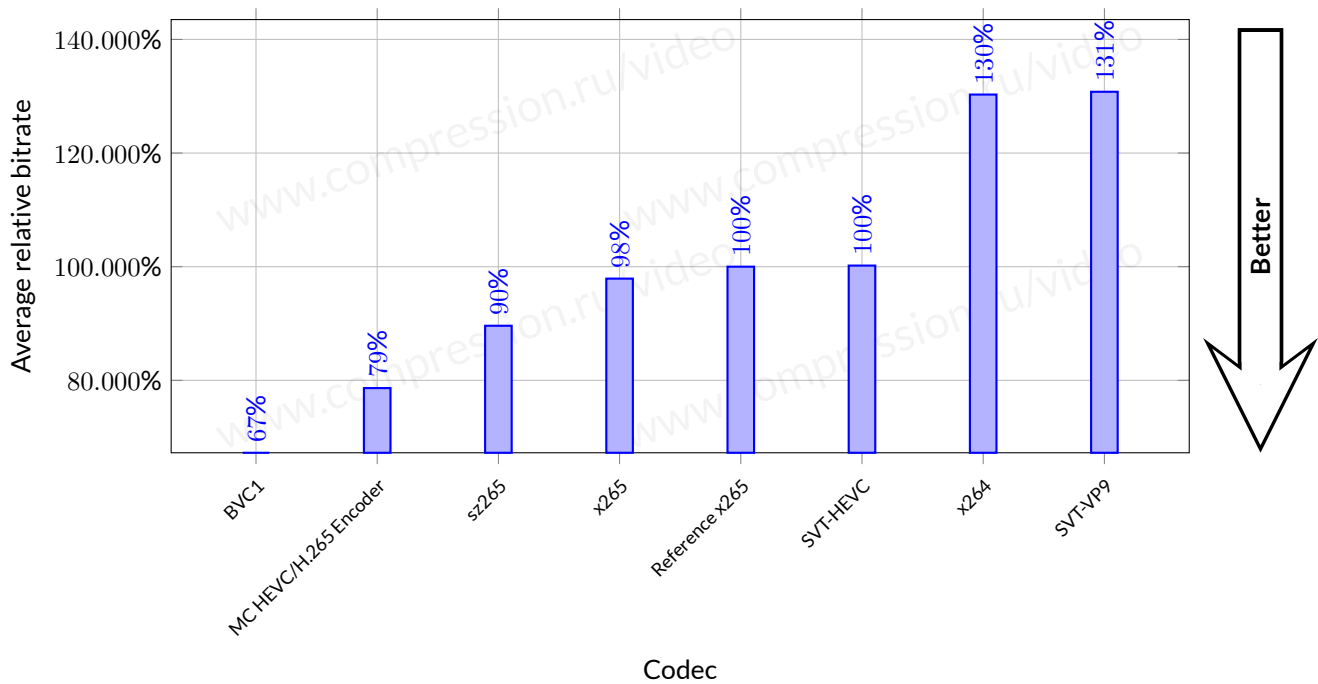


Figure 15: Average bitrate ratio for a fixed quality—all sequences, YUV-SSIM metric.

7.2. Overall YUV-PSNR (avg. MSE) (for all use cases)

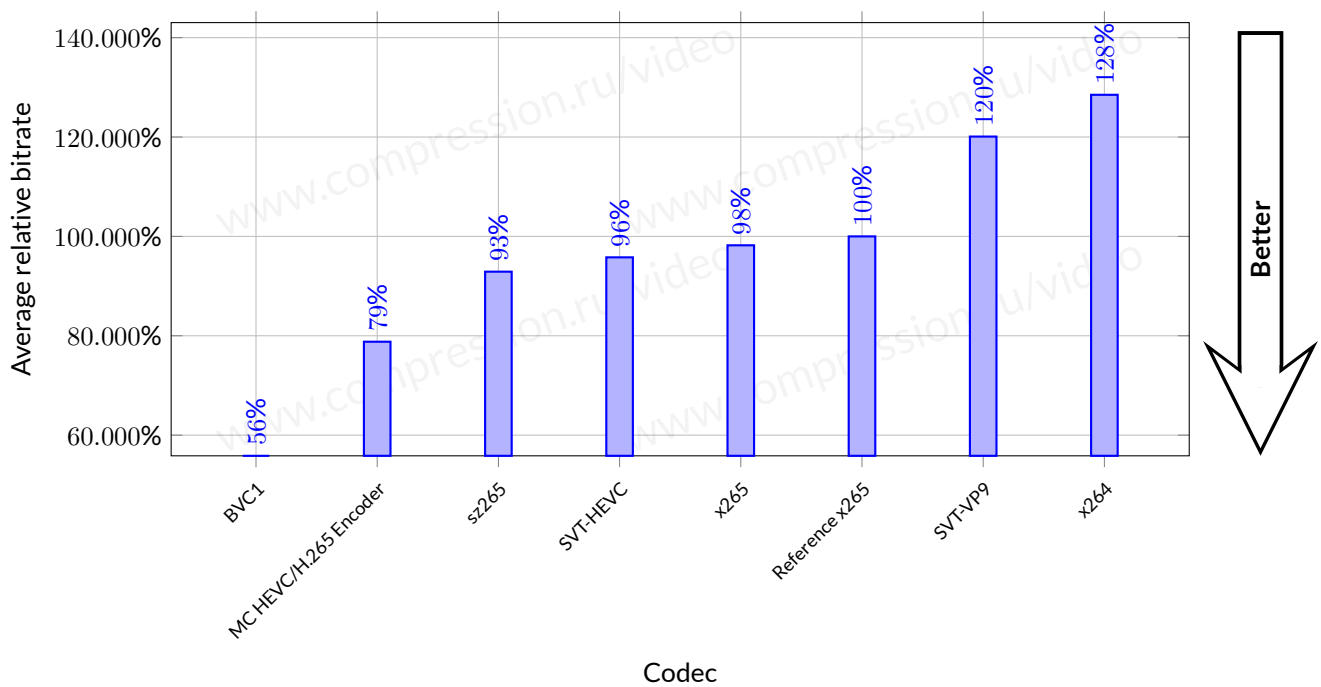


Figure 16: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. MSE) metric.

7.3. Overall YUV-PSNR (avg. log) (for all use cases)

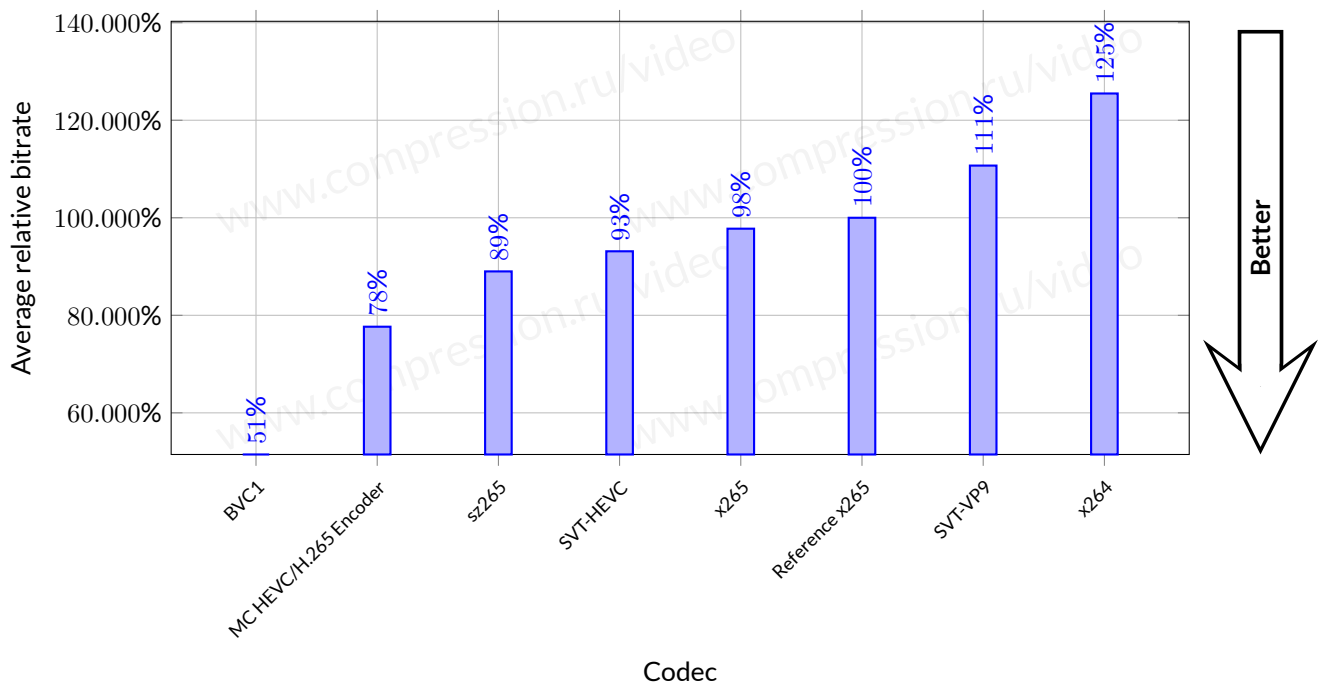


Figure 17: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. log) metric.

7.4. Overall Y-VMAF (v0.6.1 for 4K) (for all use cases)

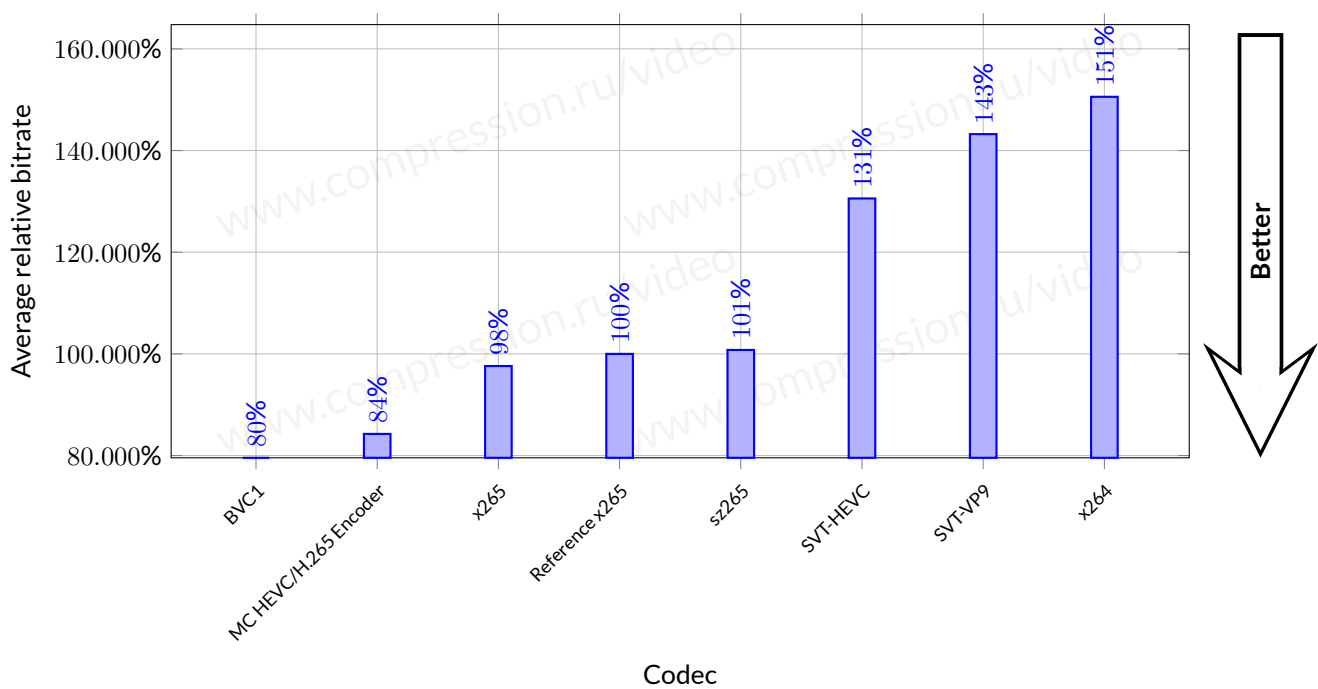


Figure 18: Average bitrate ratio for a fixed quality—all sequences, Y-VMAF (v0.6.1 for 4K) metric.

A. SEQUENCES

Direct download links to video sequences used in this comparison can be found in “MSU Codec Comparison Report ” ([Enterprise version](#))

A.1. backgammon

Sequence title	backgammon
Resolution	3840×1920
Number of frames	1058
Color space	IYUV
Frames per second	30
Source resolution	4K
Bitrate	79.13

360-degree video of people playing boardgame.



Figure 19: backgammon sequence, frame 388

A.2. ballerine

Sequence title	ballerine
Resolution	3840×1594
Number of frames	1310
Color space	IYUV
Frames per second	24
Source resolution	4K
Bitrate	120.10

A woman dancing in the city.

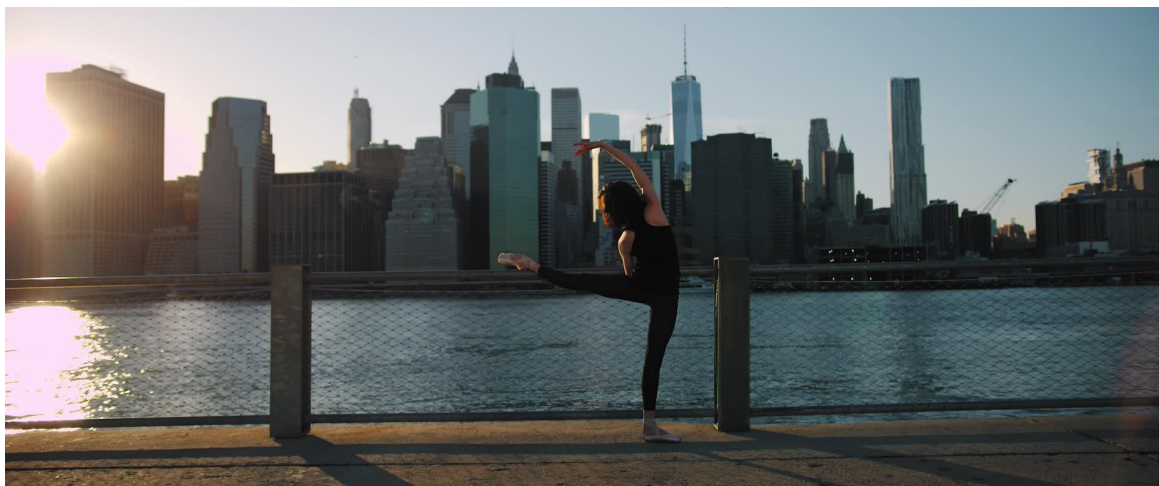


Figure 20: ballerine sequence, frame 1058

A.3. california_coast

Sequence title	california_coast
Resolution	3840×2160
Number of frames	1055
Color space	IYUV
Frames per second	30
Source resolution	4K
Bitrate	26.94

Handheld shooting of the sea and the landscape around.

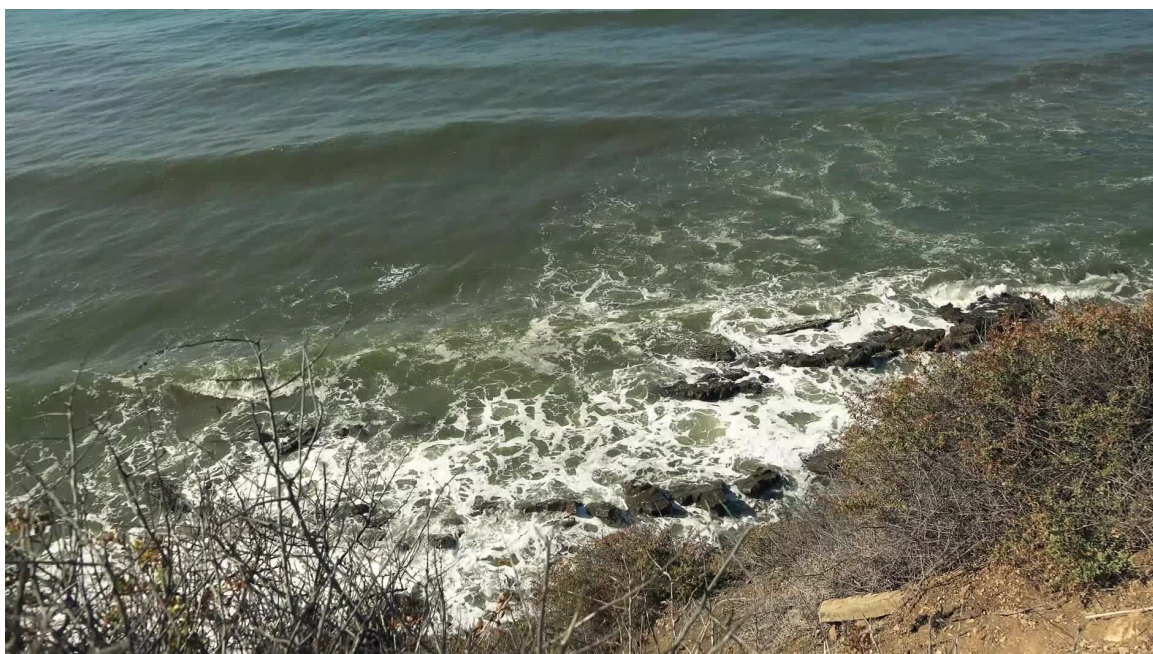


Figure 21: california_coast sequence, frame 952

A.4. crowd_run

Sequence title	crowd_run
Resolution	3840×2160
Number of frames	500
Color space	YV12
Frames per second	50
Source resolution	4K
Bitrate	4976.64

A crowd of sportsmen runs while the camera slowly moves left and right.



Figure 22: crowd_run sequence, frame 418

A.5. dron_view

Sequence title	dron_view
Resolution	3840×2160
Number of frames	1117
Color space	IYUV
Frames per second	25
Source resolution	4K
Bitrate	50.20

Aerial shooting of different landscapes.



Figure 23: dron_view sequence, frame 785

A.6. ducks_take_off

Sequence title	ducks_take_off
Resolution	3840×2160
Number of frames	500
Color space	YV12
Frames per second	50
Source resolution	4K
Bitrate	4976.64

The flock of ducks takes off the pond.



Figure 24: ducks_take_off sequence, frame 117

A.7. news

Sequence title	news
Resolution	3840×2160
Number of frames	1445
Color space	IYUV
Frames per second	30
Source resolution	4K
Bitrate	200.00

News reportage on a crowded street.



Figure 25: news sequence, frame 339

A.8. waterfall

Sequence title	waterfall
Resolution	3840×2160
Number of frames	687
Color space	IYUV
Frames per second	25
Source resolution	4K
Bitrate	57.76

Views of a waterfall.



Figure 26: waterfall sequence, frame 598

B. CODECS

All tested encoders presets can be found in “4K MSU Codecs Comparison Report 2020” ([Enterprise version](#))

C. VIDEO SELECTION

In “MSU Video Codecs Comparison 2016” we introduced a technique for selecting test video sequences. This technique allows for creating a set containing representative sequences. For this report, we used the same method and updated the video database from which we sample videos.

Figure 27 shows the bit rate distributions for our video data set by years. Table 3 shows the number of videos in our video collection.

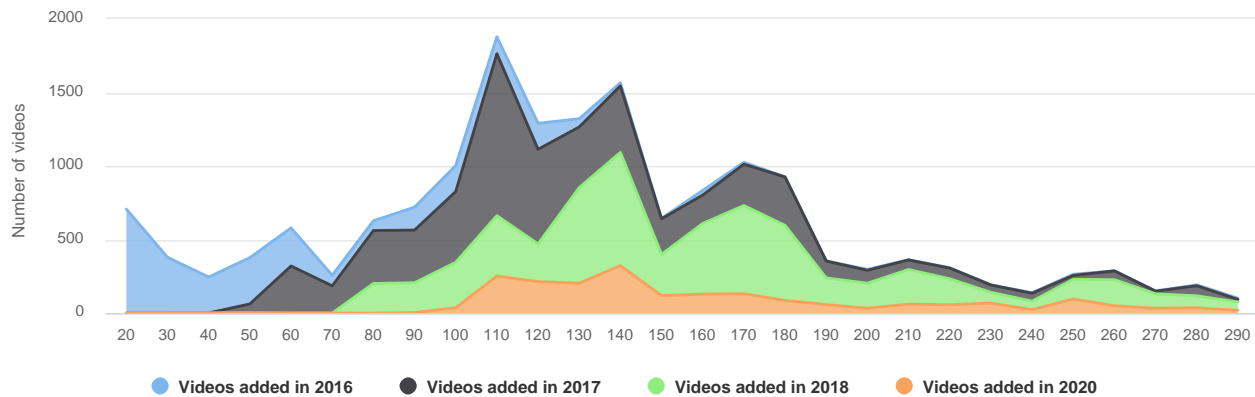


Figure 27: Bit rate distributions for comparison video set.

Year	FullHD videos	FullHD samples	4K videos	4K samples	Total (videos)	Total (samples)
2016	3	7	882	2902	885	2909
2017	1996	4638	1544	4561	3540	9299
2018	4342	10330	1946	5503	6288	15833
2020	4945	12402	2091	6016	7036	18418

Table 3: Number of videos in MSU video collection.

In order to avoid compression artifacts, and at scene changes, we cut all videos to samples using an approximate length of 1,000 frames. 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 the average size of I-frame.³ Also, an additional preprocessing step was added to unify chroma subsampling of videos which affects evaluating complexity. All videos were converted to YUV 4:2:0 chroma subsample.

This year, we conducted a voting to choose final set of 12 videos for the comparison. We divided the video collection into 12 clusters. For each cluster, we randomly selected from 1 to 6 candidate videos that were close to

³C. Chen et. al., “A Subjective Study for the Design of Multi-resolution ABR Video Streams with the VP9 Codec,” 2016.

the cluster centre and that had a license enabling derivatives and commercial use. Figure 28 shows the cluster boundaries and constituent sequences.

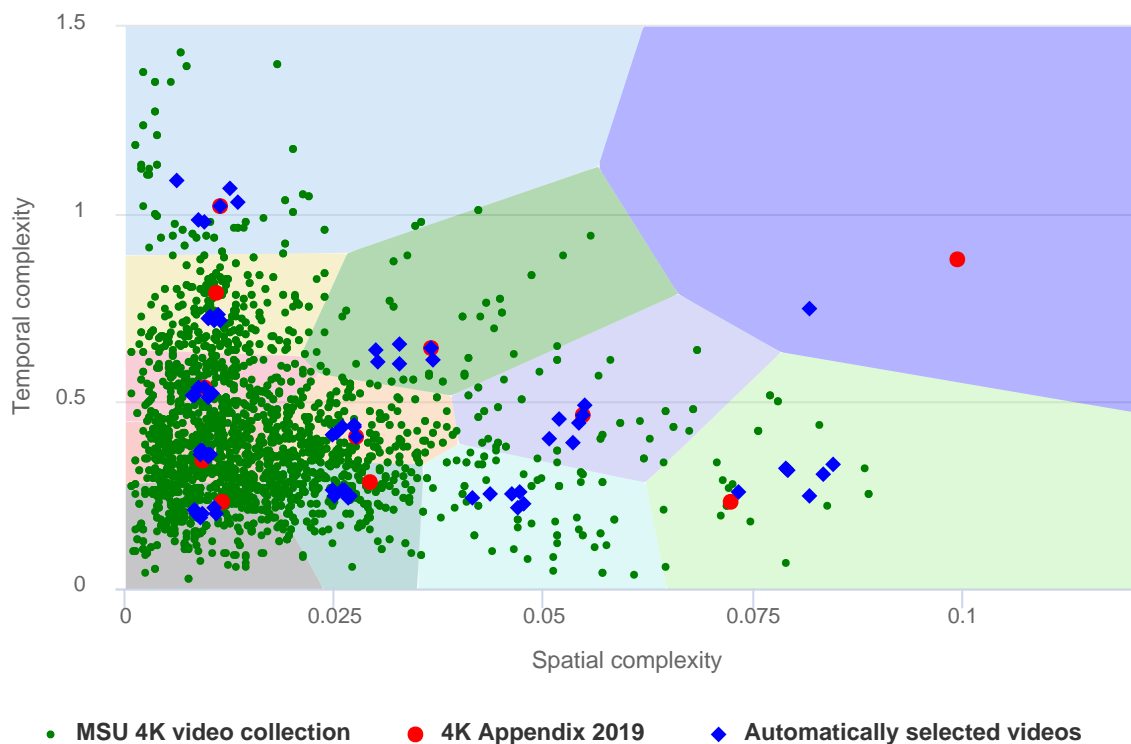


Figure 28: Segmentation of samples.

We chose 12 videos from the candidated trying to include videos of different semantic in a final dataset. The new data set consists of 12 sequences (8 sequences with 8-bit color depth and 4 sequences with 10-bit color depth), the complete list of sequences appears in Appendix A.

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 29 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.

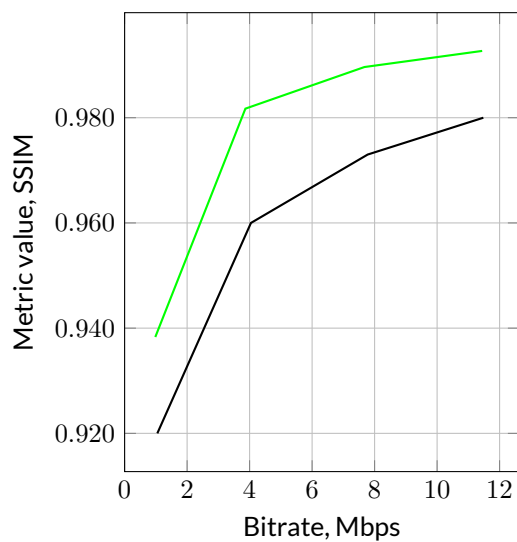
Owing to these advantages, we frequently use relative-bitrate/relative-time graphs in this report because they assist in evaluating the codecs in the test set, especially when the number of codecs is large.

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

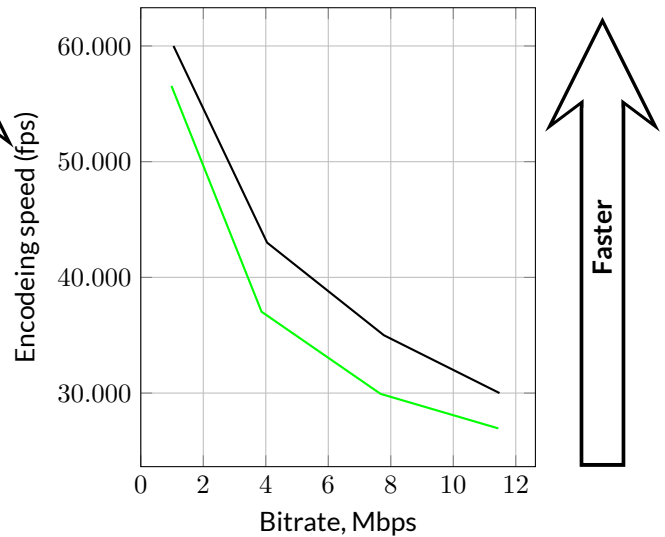
D.4. Bitrate Ratio for the Same Quality

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 30b). All further computations use the inverted graph.

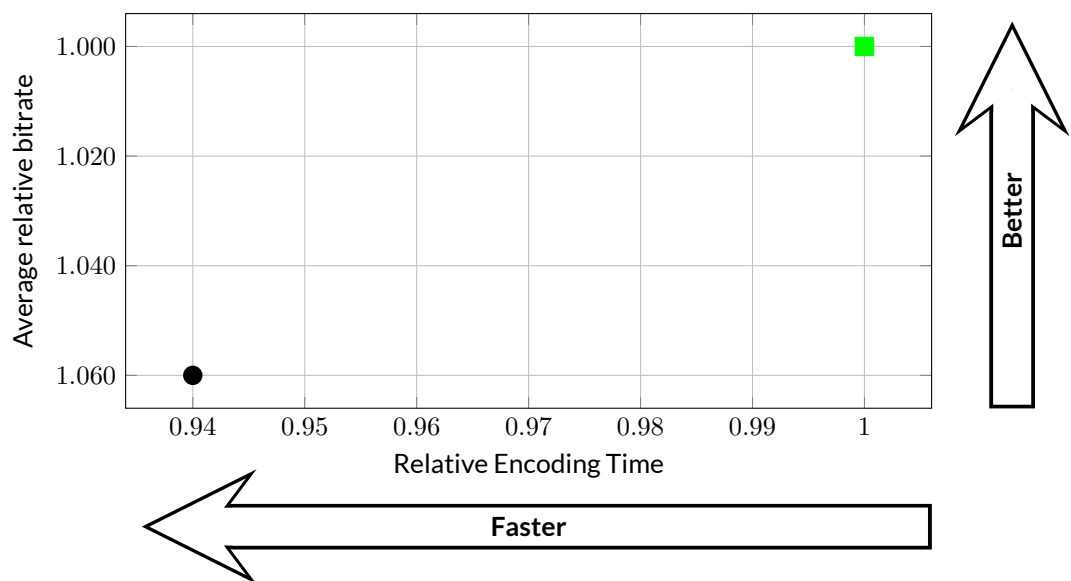
The second step involves averaging the interval over which the quality axis is chosen. The averaging is only over those segments for which both codecs yield results. This limitation is due to the difficulty of developing extrap-



(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 29: Speed/Quality trade-off example

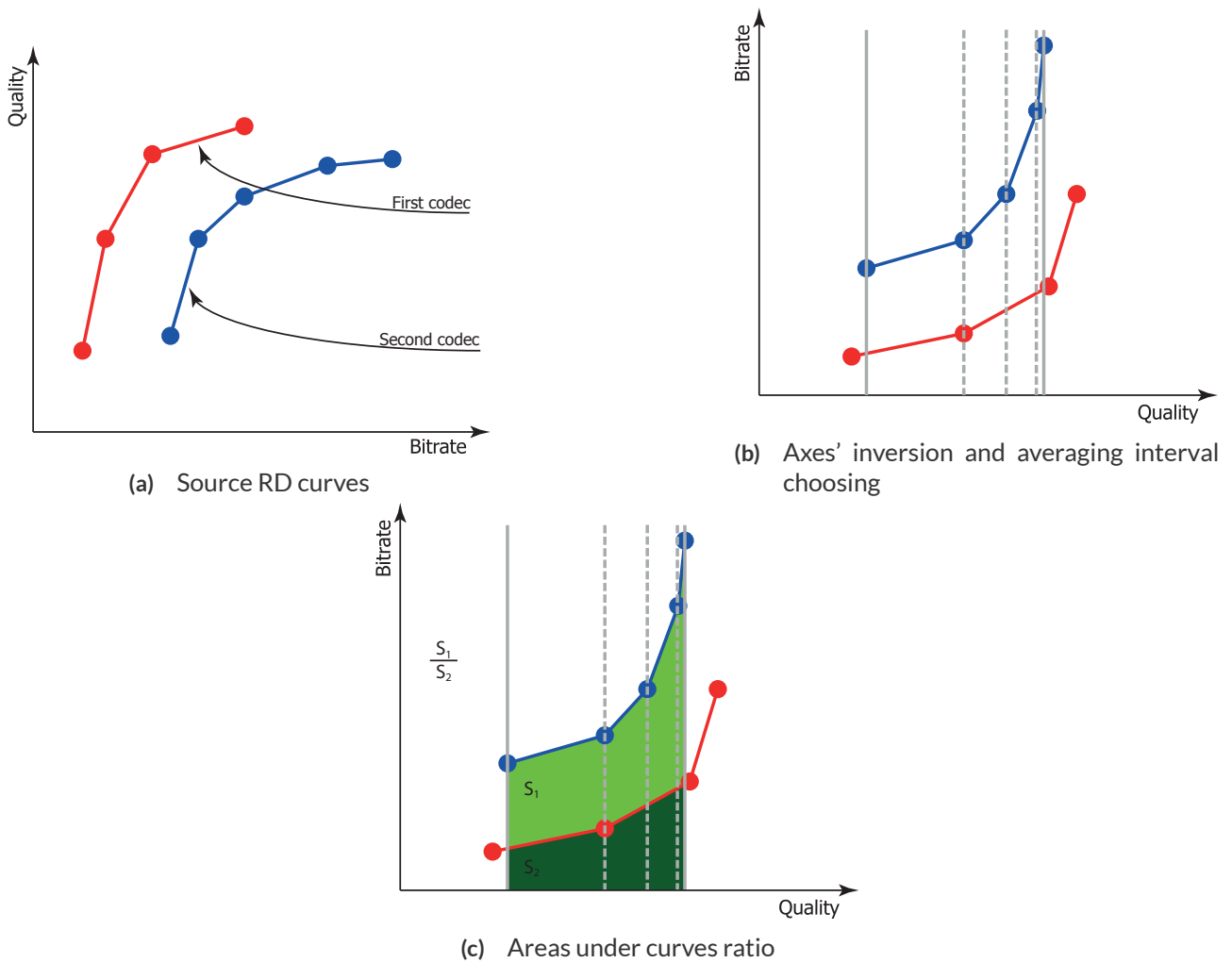


Figure 30: Average bitrate ratio computation

olation 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 30c). This result is an average bitrate ratio at a fixed quality for the two codecs. When 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 31) shows that the results for these extra bitrates (purple) cross with codec two and enable a comparison with codec one.

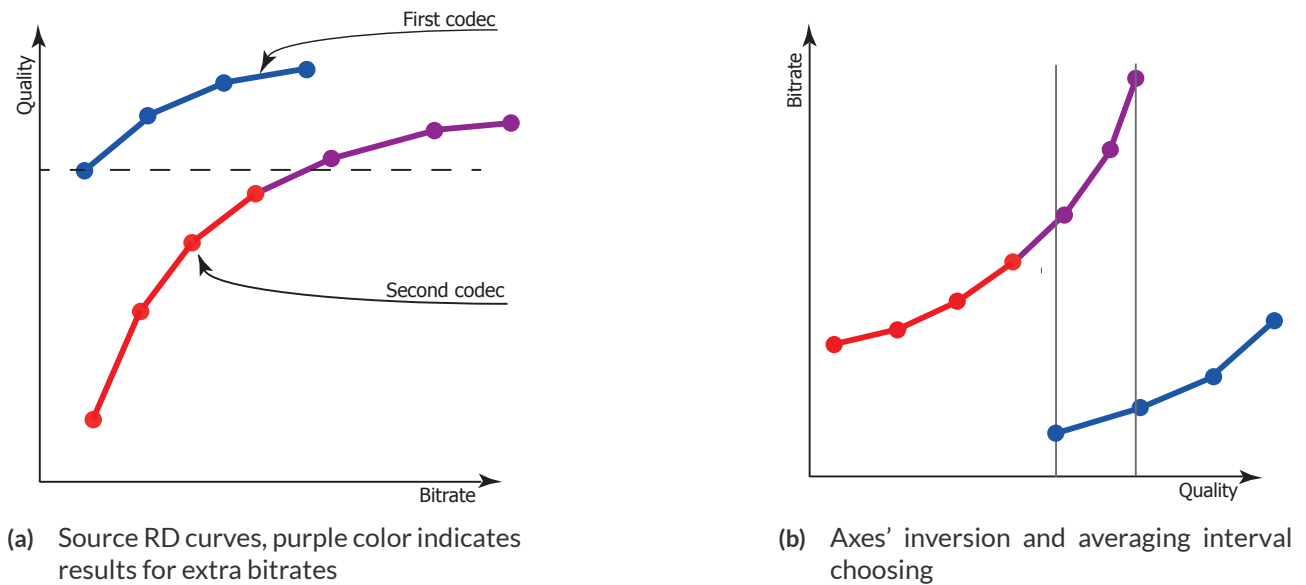


Figure 31: Measuring codec on additional bitrates to make it cross with other codecs over the quality axis.

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 32.

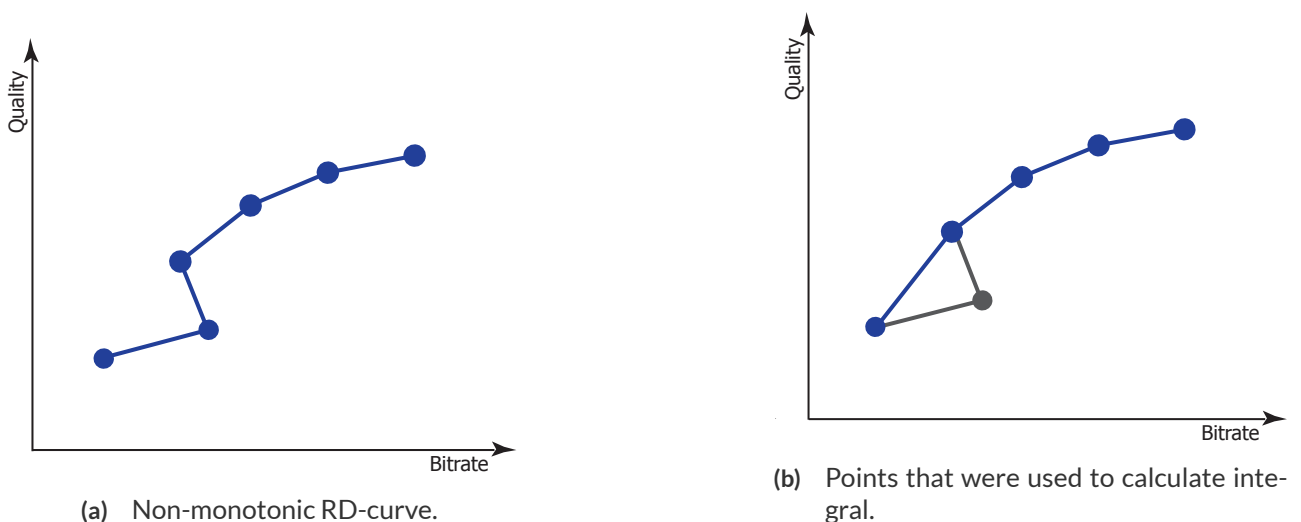


Figure 32: Processing non-monotonic RD-curves.

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):

$$\text{YUV-SSIM} = \frac{4 \text{Y-SSIM} + \text{U-SSIM} + \text{V-SSIM}}{6}. \quad (1)$$

Below is a brief description of SSIM computation.

E.1.1. Brief Description

Wang, et al.⁴ 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

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

where

$$\mu_x = \sum_{i=1}^N \omega_i x_i, \quad (3)$$

$$\sigma_x = \sqrt{\sum_{i=1}^N \omega_i (x_i - \mu_x)^2}, \quad (4)$$

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

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 $K_1, K_2 \ll 1$.

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.

⁴Zhou 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.

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 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 33 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.

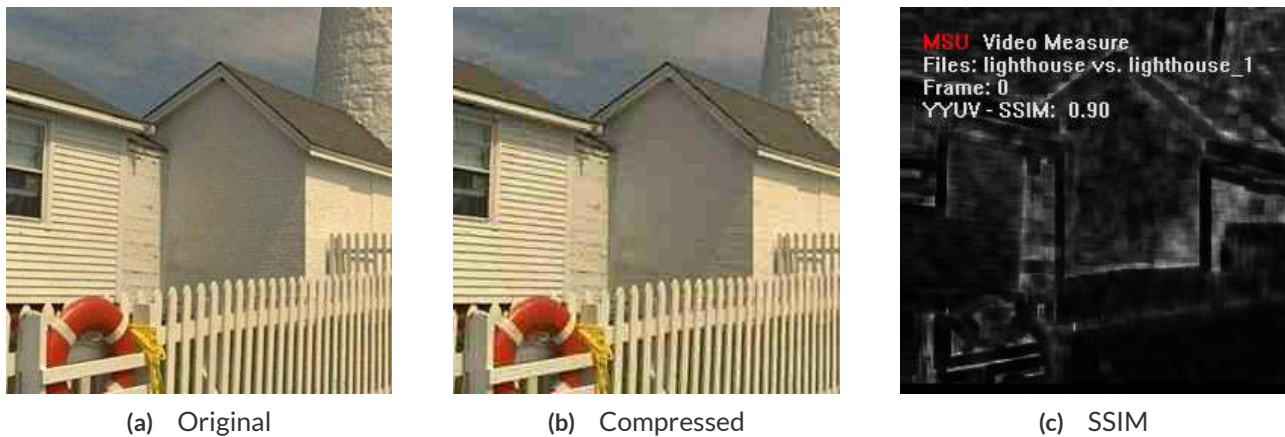


Figure 33: SSIM example for compressed image

Figure 34 depicts various distortions applied to the original image, and Figure 35 shows SSIM values for these distortions.



(a) Original image



(b) Image with added noise



(c) Blurred image



(d) Sharpen image

Figure 34: 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 sharpened image,
SSIM = 0.958917

Figure 35: SSIM values for original and processed images

E.1.3. Measurement method

We used the [MSU Video Quality Measurement Tool \(VQMT\)](http://compression.ru/video/quality_measure/vqmt_download.html#start) 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}x{height} -in "decoded_yuv" IYUV
{width}x{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.

E.2. PSNR (Peak Signal-to-Noise Ratio)

PSNR correlates poorly with subjective scores compared to VMAF, however it is still widely used to assess video quality.

For images I and \hat{I} with resolution $n \times m$:

$$MSE(I, \hat{I}) = \frac{1}{nm} \sum_{i=1}^n \sum_{j=1}^m (I_{ij} - \hat{I}_{ij})^2 \quad (6)$$

$$PSNR(I, \hat{I}) = 10 \log_{10} \frac{MAX_I^2}{MSE(I, \hat{I})} \quad (7)$$

There are two averaging strategies, both are used for codec development.

E.2.1. PSNR (avg. MSE)

For two videos V and \hat{V} :

$$PSNR_{avg. MSE}(V, \hat{V}) = 10 \log_{10} \frac{MAX_I^2}{\frac{1}{n} \sum_{i=1}^n MSE(V_{(i)}, \hat{V}_{(i)})} \quad (8)$$

E.2.2. PSNR (avg. log)

For two videos V and \hat{V} :

$$PSNR_{avg. log}(V, \hat{V}) = \frac{1}{n} \sum_{i=1}^n 10 \log_{10} \frac{MAX_I^2}{MSE(V_{(i)}, \hat{V}_{(i)})} \quad (9)$$

F. ABOUT THE 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



MSU Video Quality Measurement Tool



Speedup of your video quality measurement up to 12 times

3 reasons to use VQMT:

- Fastest implementation of VMAF
- Fastest SSIM/MS-SSIM speed on 4K/8K video
- Professional analysis with NIQE and artifact metrics video-measure@compression.ru



Widest Range of Metrics & Formats

20+ Objective Metrics

PSNR several versions	Spatio-Temporal SSIM
MSAD	MSU Blurring Metric
Delta	MSU Brightness Flicking Metric
MSE	MSU Brightness Independent
VQM	PSNR
SSIM	MSU Drop Frame Metric
MS-SSIM	MSU Noise Estimation Metric
3-SSIM	MSU Scene Change Detector
VMAF	MSU Blocking Metric
	NIQE (no-reference comparison)

HDR support

Hundreds Video and 30+ Image Formats

All popular video codecs, including H264 and HEVC.
Special support for: RAW, Y4M, AviSynth, PXM.
All popular image formats: PNG, JPEG, TIFF (with HDR support), EXR, BMP, PSD, and others

2k, 4k, 8k support

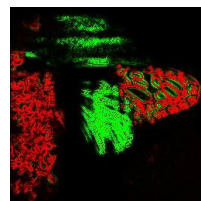
Fastest Video Quality Measurement

Up to 11.7x faster calculation of metrics with GPU (CUDA & OpenGL support)

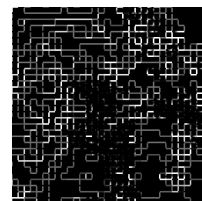
Multi-core Processors Support

Visualization Examples

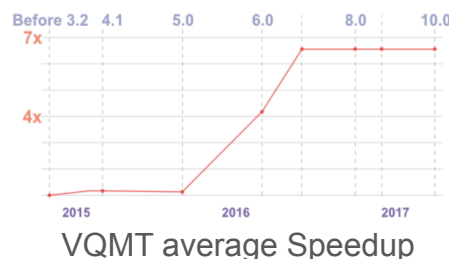
Allows easily detect where codec/filter fails



MSU Blurring Metric



MSU Blocking Metric



Easy Integration

Linux Support

DEB & RPM packages

Batch Processing with JSON and CSV output

Plugins SDK

Professional Analysis

Comparative Analysis

Metric Visualization

MSU VQMT Official Page

compression.ru/video/quality_measure/video_measurement_tool.html

Tool was downloaded more than 200 000 times!

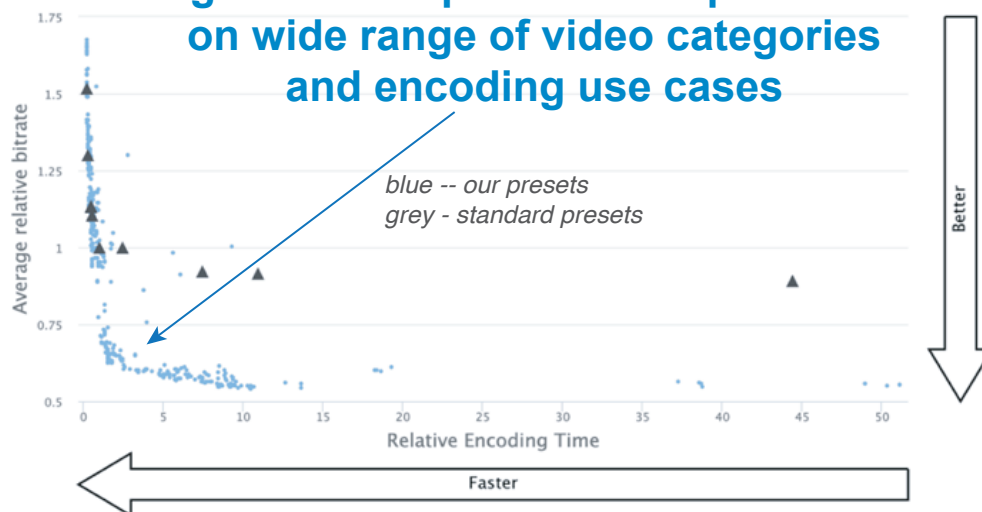
Free and Professional versions are available

Big thanks to our contributors:



Video Group of Lomonosov MSU Graphics&Media Lab has **15-years experience** in video codecs analysis and optimization. We know that almost always it is possible to find efficient encoding options for every video which increase encoding performance

Our goal is to improve codec performance on wide range of video categories and encoding use cases



Why is codec tuning difficult?

Example of x264 tuning for one 20-second video:

- 49 encoding options
- many options make unexpected influence on encoding performance
- exhaustive search for 500-frames video sequence will last $\sim 2.2 \cdot 10^{13}$ computing centuries ($\sim 488\,000$ Earth ages)

15% bitrate savings in average

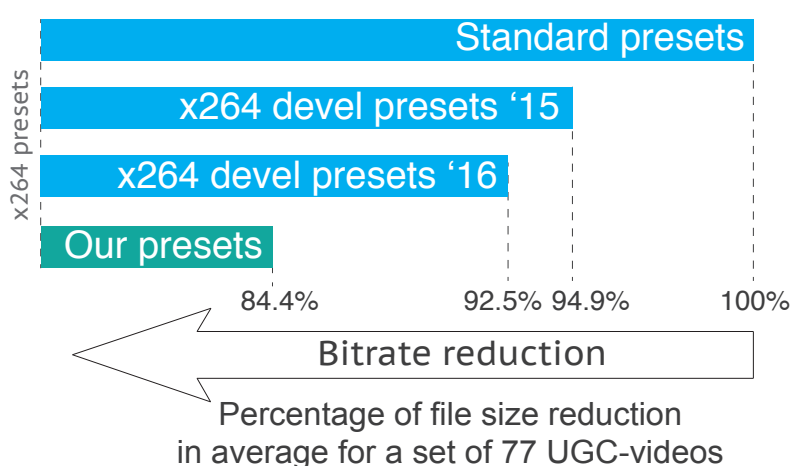
Encoding presets determined by our method beats x264 developers' presets with keeping encoding time and encoded video quality

We find presets that **do not reduce encoding speed and objective quality of encoded video** compared to your given reference

You give limitations, and we guarantee the same or higher objective quality and encoding speed

You use standard presets and don't believe it will work for your videos?

Give us a chance — request a free demo!



We can find best encoding presets for your videos

- Your video**
send us uncompressed video and your preset
- Report**
get a report with optimal presets for your video and their gain
- Choose and pay**
we offer additional options for better compression and analysis
- Get preset** or **Get video**
and encode similar videos with it / compressed with chosen preset

Our project page: compression.ru/video/video_codec_optimization/