MSU Video Codecs Comparison 2020

Part I: FullHD Content, Objective Evaluation

Graphics & Media Lab
Video Group

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

- BD265
- Reference x265
- S265 (bitrate mode)
- S265 (CRF mode)
- SVT-HEVC
- sz265
- Tencent V265
- x265
- XCCZM265
- xin26x

AV1

- aom
- Aurora AV1 Encoder
- donkey
- QAV1
- rav1e
- SVT-AV1

Other

- BVC2.0
- SIF Codec
- SVT-VP9
- VP9
- x264

CS MSU Graphics & Media Lab, Video Group

December 7, 2020

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

videocodec-testing@graphics.cs.msu.ru

1. REPORT VERSIONS

	Free version	Enterprise version
Use cases	Offline (1 fps) (partially)	Offline (1 fps), Online (30 fps)
Per-sequence-results	1 of 50 sequences (only Universal use case)	All 50 sequences for all use cases (in interactive charts)
Metric: YUV-SSIM, VMAF (overall results), PSNR (overall results)	✔	✔
Other objective metrics (Y-VMAF(0.6.3), 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

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:

- A Father (xin26x)
- Alibaba Group (zhuangshu.wlb, pengkai.pk, yangde.dy, david.wlj, hongkuai.wyh, zuobo.zxg, yilin.zy, shenjie.lrh, xiaobo.lixb, zhengguang.lzg)
- Baidu Inc.
- Bytedance Inc.
- ChangKuoLao
- iQIYI Inc.
- MulticoreWare, Inc.
- RayShaper
- SIF Codec LLC
- Tencent
- The rav1e contributors
- Visionular
- x264 project
- XCCZM Codec Team

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.	ac_origins	900	30	1920×1080
2.	aerial_media	968	30	1920×1080
3.	alt_rock	480	24	1920×1080
4.	apple_tree	338	30	1920×1080
5.	ariadnes_thread	902	30	1920×1080
6.	beach_interview	1001	30	1920×1080
7.	bhutan	967	25	1920×1080
8.	bsu_volleyball	600	30	1920×1080
9.	chili_pepper	3576	60	1920×1080
10.	christmas_cats	1500	25	1920×1080
11.	cineei_mode	971	30	1920×1080
12.	construction_site	1043	30	1920×1080
13.	creek_cooler	1451	30	1920×1080
14.	crowd_run	500	50	1920×1080
15.	dusk_train	1476	24	1920×1080
16.	fishing	952	25	1920×1080
17.	flower_shop	749	25	1920×1080
18.	football	599	30	1920×1080
19.	forest_dog	976	25	1920×1080
20.	forest_eye	1800	25	1920×1080
21.	getawards	1020	25	1920×1080
22.	hard_rock	500	25	1920×1080
23.	hockey	1000	25	1920×1080
24.	humanitarian_day	1714	24	1920×1080
25.	inner_shaq	1569	24	1920×1080
26.	kentucky_orchestra	1107	30	1920×1080
27.	kindergarten_interview	1016	30	1920×1080
28.	kings_park	1618	24	1920×1080
29.	mountain_valley	266	24	1920×1080
30.	music_band	1325	30	1920×1080
31.	okeechobee	1402	24	1920×1080
32.	park_mobile	359	24	1920×1080

33.	pyranha_kayak	235	24	1920×1080
34.	pyranha_rafting	1203	24	1920×1080
35.	restaurant_talk	1047	24	1920×1080
36.	road_timelapse	759	30	1920×1080
37.	saltburn	1103	30	1920×1080
38.	strange_morning	790	24	1920×1080
39.	street_musician	974	24	1920×1080
40.	summer_of_adventure	994	30	1920×1080
41.	surfing	120	30	1920×1080
42.	suriname_reserve	992	24	1920×1080
43.	tennis_vlog	599	30	1440×1080
44.	teton_bros	1003	24	1920×1080
45.	the_refuge	1001	24	1920×1080
46.	underwater_shooting	1170	24	1920×1080
47.	video_lecture	600	30	1920×1080
48.	way_out	931	25	1920×1080
49.	wedding_party	1757	24	1920×1080
50.	wedding_preparations	992	24	1920×1080

Table 1: Summary of video sequences

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

3.2. Codecs

Codec	Developer	Version	OS
aom	AOMedia	2.0.0-287-g2aa13c436	Windows
Aurora AV1 Encoder	Visionular	2.0	Windows
BD265	Baidu Inc.	2.0	Windows
BVC2.0	Bytedance Inc.	V0	Windows
donkey	ChangKuoLao	0.4.2c-a60655ae	Windows
QAV1	iQIYI Inc.	v1.1	Linux
rav1e	The rav1e contributors	0.3.0 (p20200515)	Windows
Reference x265	MulticoreWare, Inc.	3.3+21-6bb2d88029c2	Windows
S265 (bitrate mode)	Alibaba Group	v5.0.2	Windows
S265 (CRF mode)	Alibaba Group	v5.0.2	Windows
SIF Codec	SIF Codec LLC	1.91	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	v1.0.0	Linux
Tencent V265	Tencent	v1.4.5	Windows
VP9	The WebM Project	v1.8.2	Windows
x264	x264 project	0.160.3000 33f9e14	Windows
x265	MulticoreWare, Inc.	3.3+33-3116be008af1	Windows
XCCZM265	XCCZM Codec Team	v3.2_0607	Linux
xin26x	A Father (xin26x)	1.1	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 C provides detailed descriptions of all codecs in our comparison.

4. OBJECTIVES AND TESTING RULES

This report presents the results of video codecs comparison, in which we used objective assessment methods to compare the encoding quality of recent HEVC encoders as well as encoders implementing other standards. This effort employed 50 video sequences at 1080p resolution to evaluate codec performance. The process of video sequences selection involved voting among the participants, organizers and an independent expert. To choose out test set, we analyzed more than 1,500,000 video sequences and selected representative examples (a detailed description of the selection process appears in Appendix D).

Our comparison consists of two parts, corresponding to various encoder use cases: fast encoding and offline 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:

- Online (30 fps)—30fps
- Offline (1 fps)—1fps

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 or Ubuntu.

For objective quality measurements we used YUV-SSIM metric (see Appendix F.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.

In the second part of this report, we will introduce subjective study conducted using Subjectify.us platform, which makes subjective studies much easier but still difficult to compare on 50-100 videos. Most no-reference metrics show inaccurate results, such as our recent investigation on NIQE [1].

According to many requests, we also show VMAF results as a subjective quality-oriented indicator. Recently our team investigated tuning for VMAF [2], 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 used [3]. As it was described in the paper, this method shows more accurate results on complex cases of codecs performance comparison than BD-rate.

[1] A. Zvezdakova, D. Kulikov, D. Kondranin, D. Vatolin, "Barriers Towards No-reference Metrics Application to Compressed Video Quality Analysis: on the Example of No-reference Metric NIQE," 2019.

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

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

5. OFFLINE (1 FPS)

5.1. RD Curves

Judging from the mean quality scores (computed using the method described in Section E), first place in the quality competition goes to **Aurora AV1 Encoder** and **BVC2.0**, second place goes to **QAV1**, and third place to **Tencent V265** and **aom**.

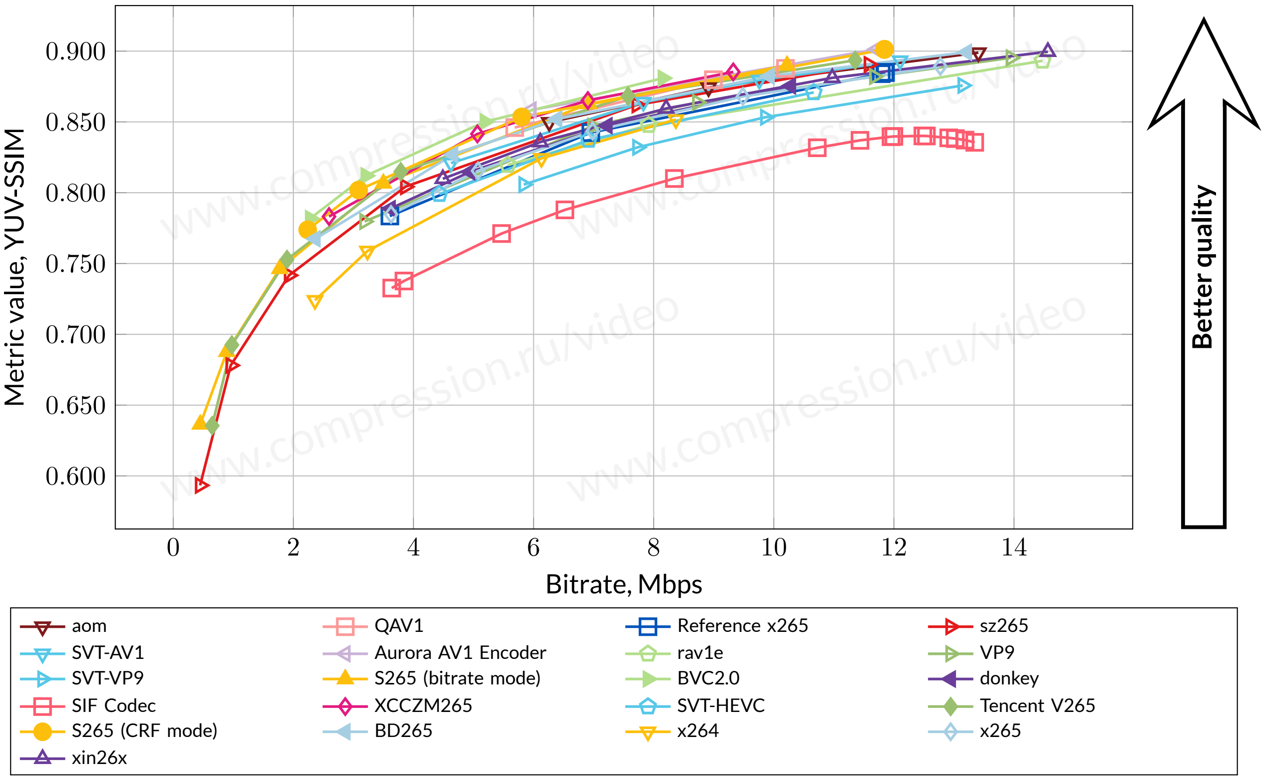
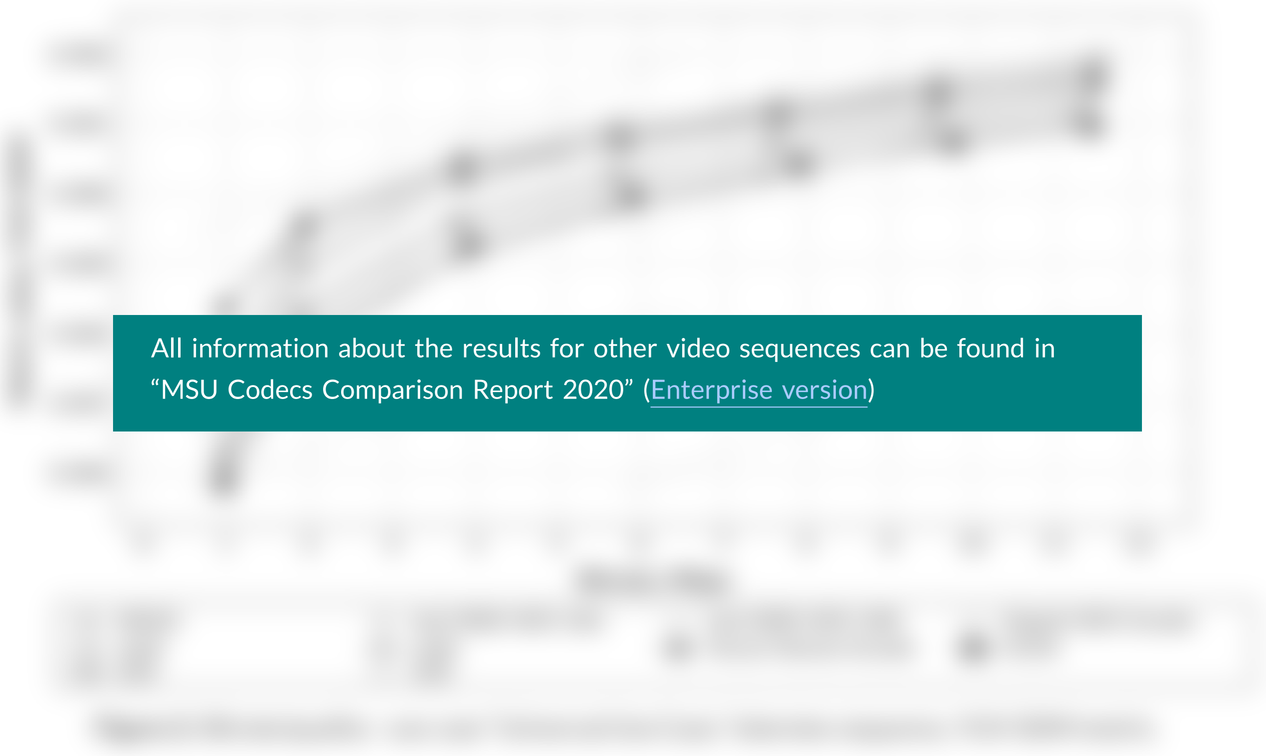


Figure 1: Bitrate/quality—use case "Offline (1 fps)," *crowd_run* sequence, YUV-SSIM metric.

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



All information about the results for other video sequences can be found in "MSU Codecs Comparison Report 2020" (Enterprise version)

5.2. Encoding Speed

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

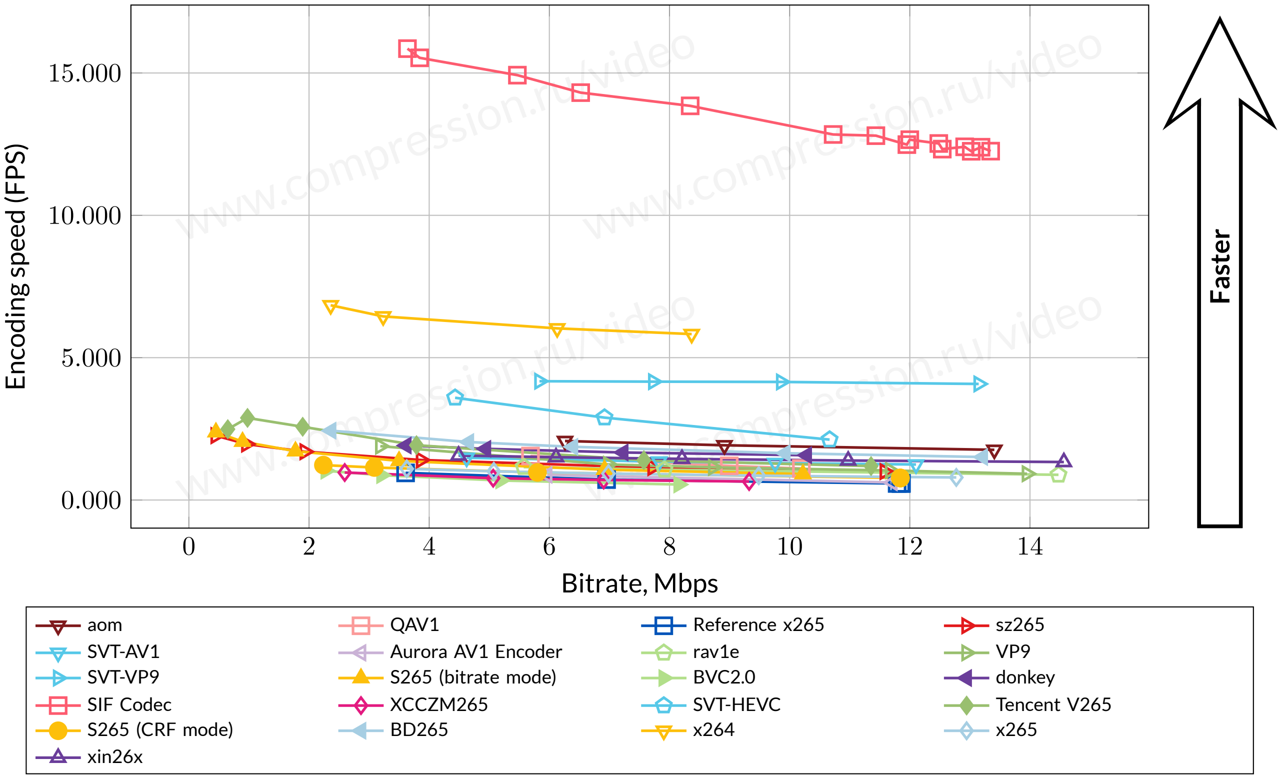


Figure 2: Encoding speed—use case "Offline (1 fps)," *crowd_run* sequence.

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

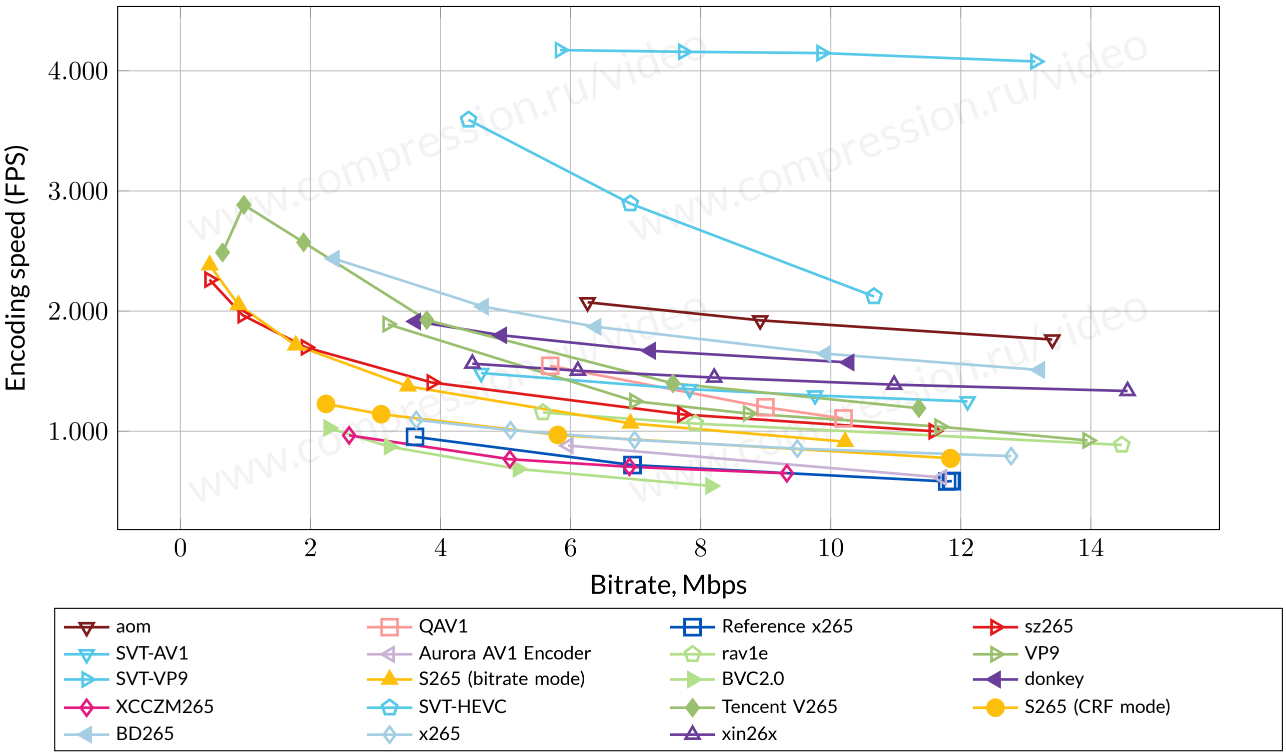
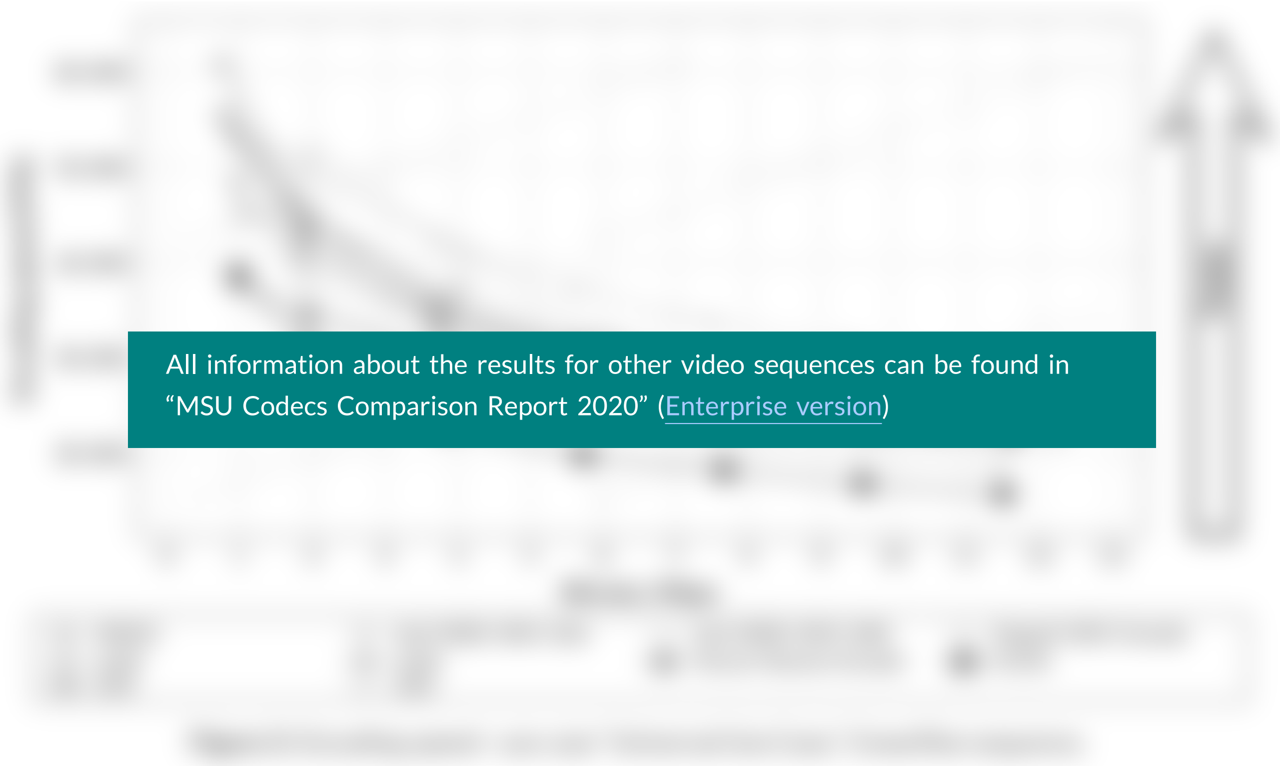


Figure 3: Encoding speed—use case "Offline (1 fps)," *crowd_run* sequence, without: SIF Codec, x264.

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



All information about the results for other video sequences can be found in "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 E. 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 seven Pareto-optimal encoders: **Aurora AV1 Encoder**, **QAV1**, **Tencent V265**, **donkey**, **BD265**, **x264**, and **SIF Codec**.

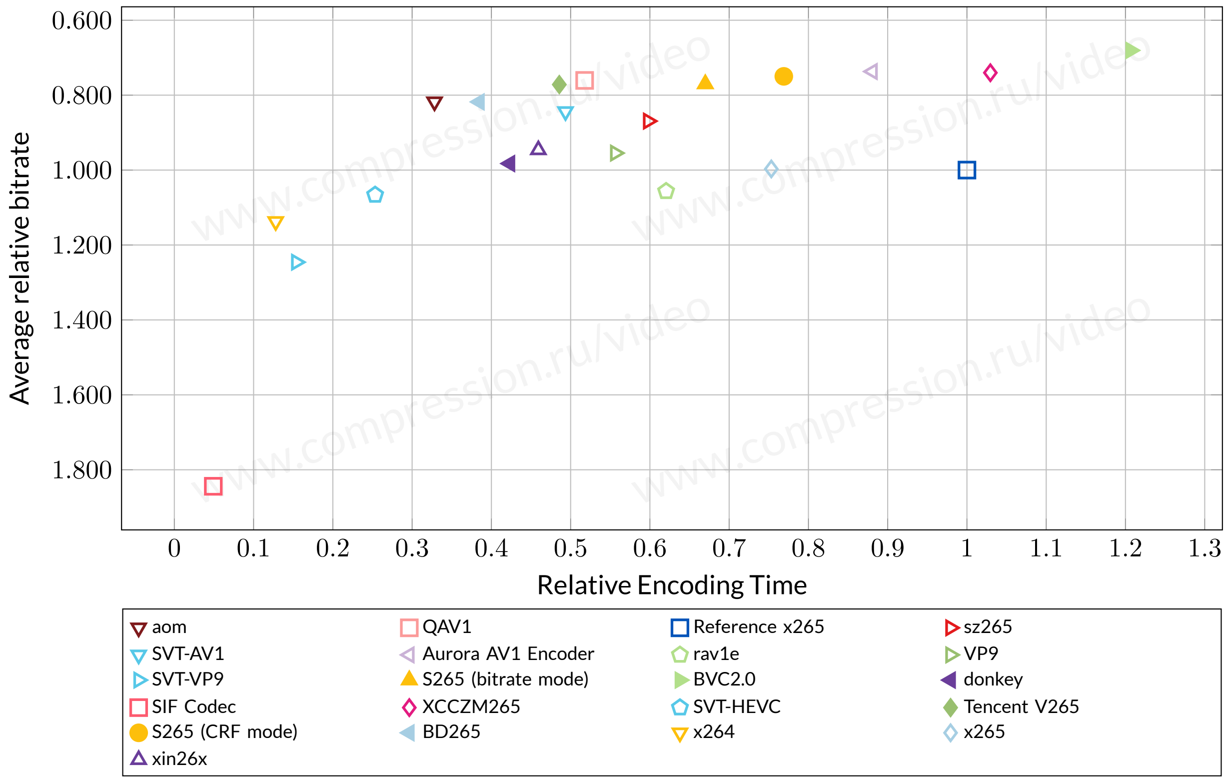


Figure 4: Speed/Quality Trade-Off—use case "Offline (1 fps)," *crowd_run* sequence, YUV-SSIM metric.

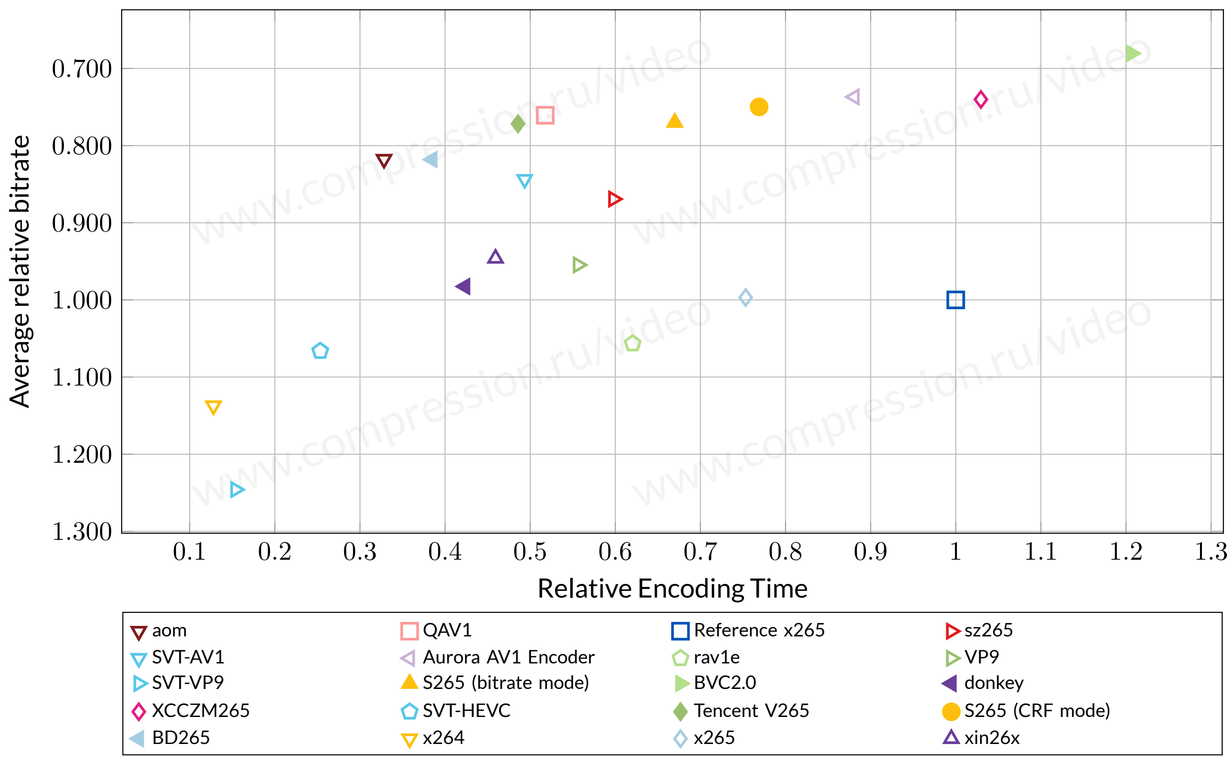
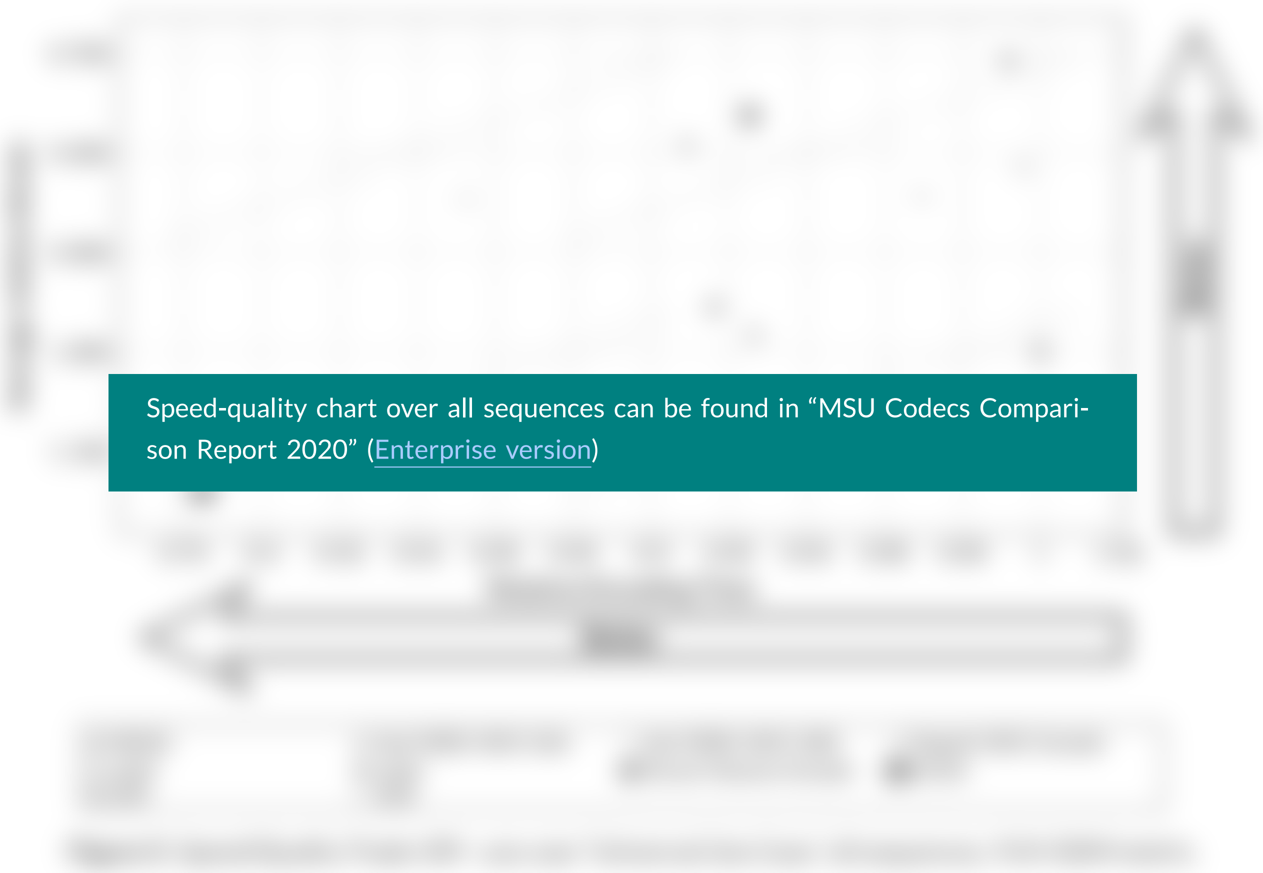


Figure 5: Speed/Quality Trade-Off—use case "Offline (1 fps)," *crowd_run* sequence, YUV-SSIM metric, without SIF Codec.



Speed-quality chart over all sequences can be found in "MSU Codecs Comparison Report 2020" (Enterprise version)

5.4. Offline (1 fps) YUV-SSIM

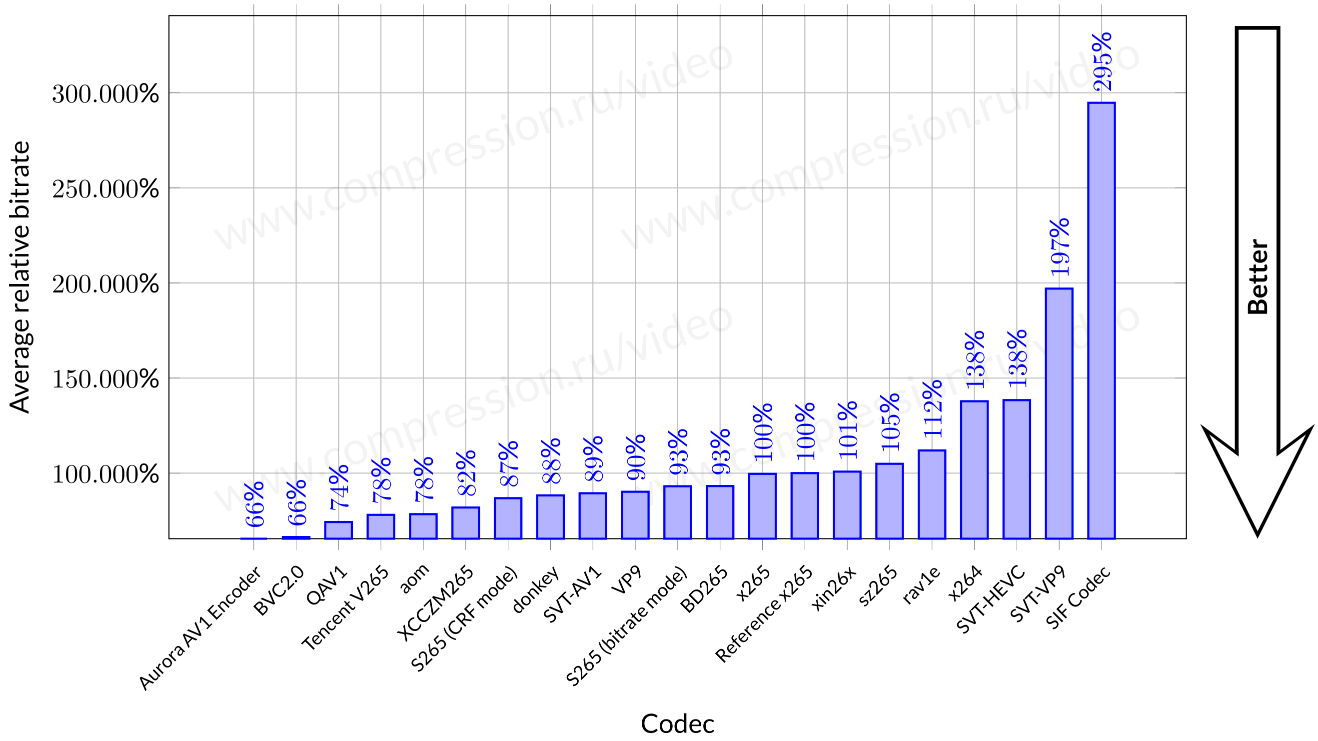


Figure 6: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-SSIM metric.

5.5. Offline (1 fps) YUV-PSNR (avg. MSE)

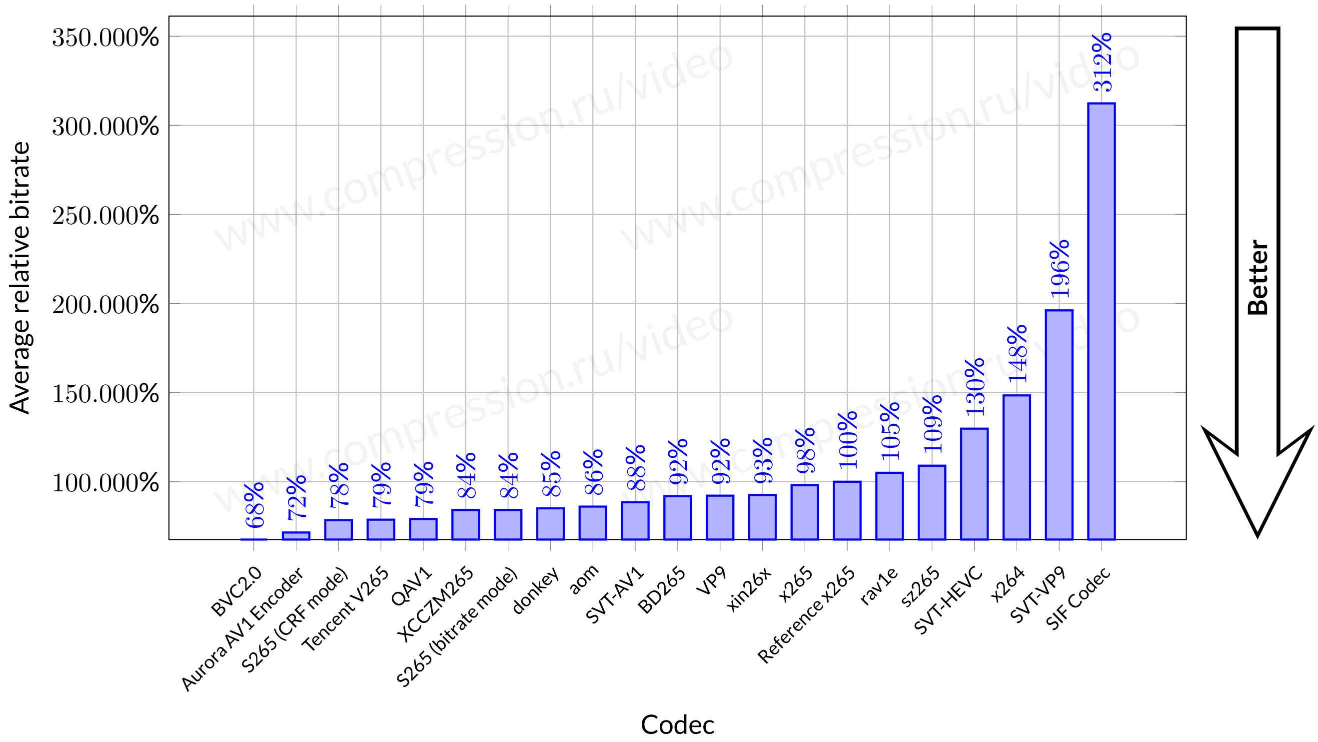


Figure 7: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-PSNR (avg. MSE) metric.

5.6. Offline (1 fps) YUV-PSNR (avg. log)

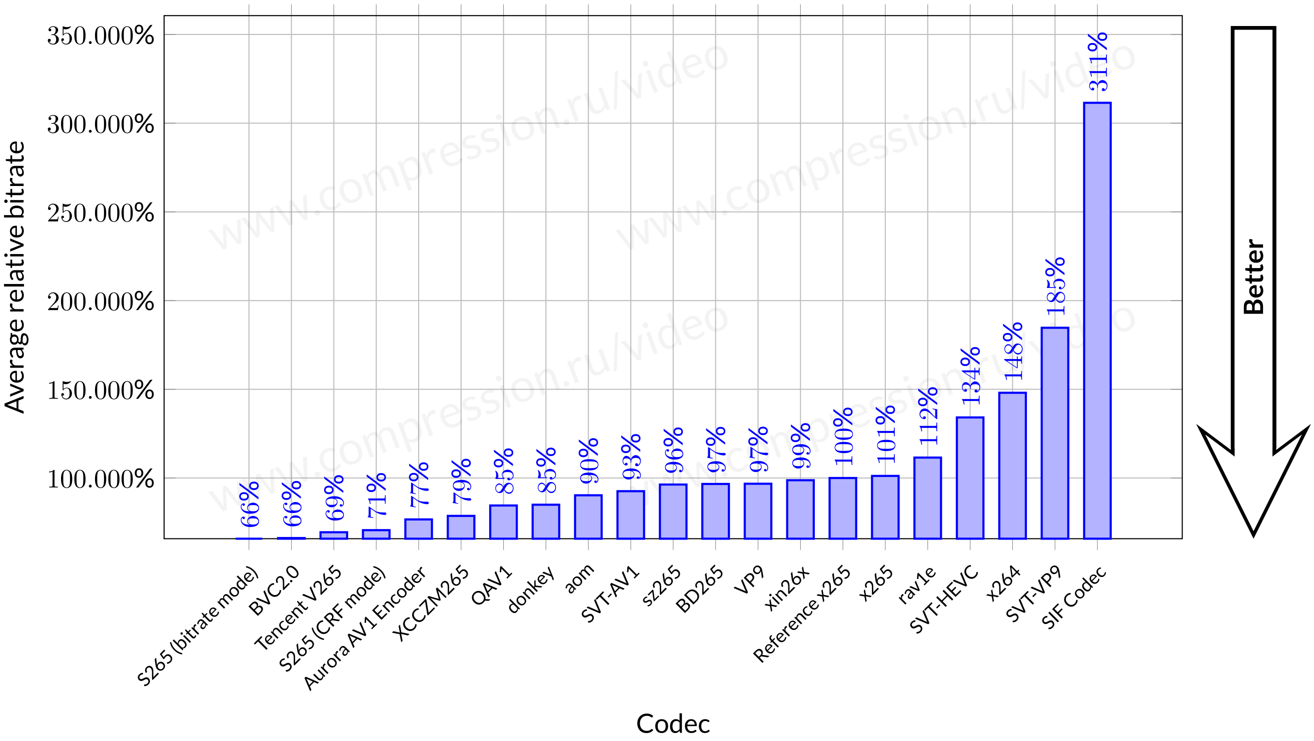


Figure 8: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-PSNR (avg. log) metric.

5.7. Offline (1 fps) Y-VMAF (v0.6.3)

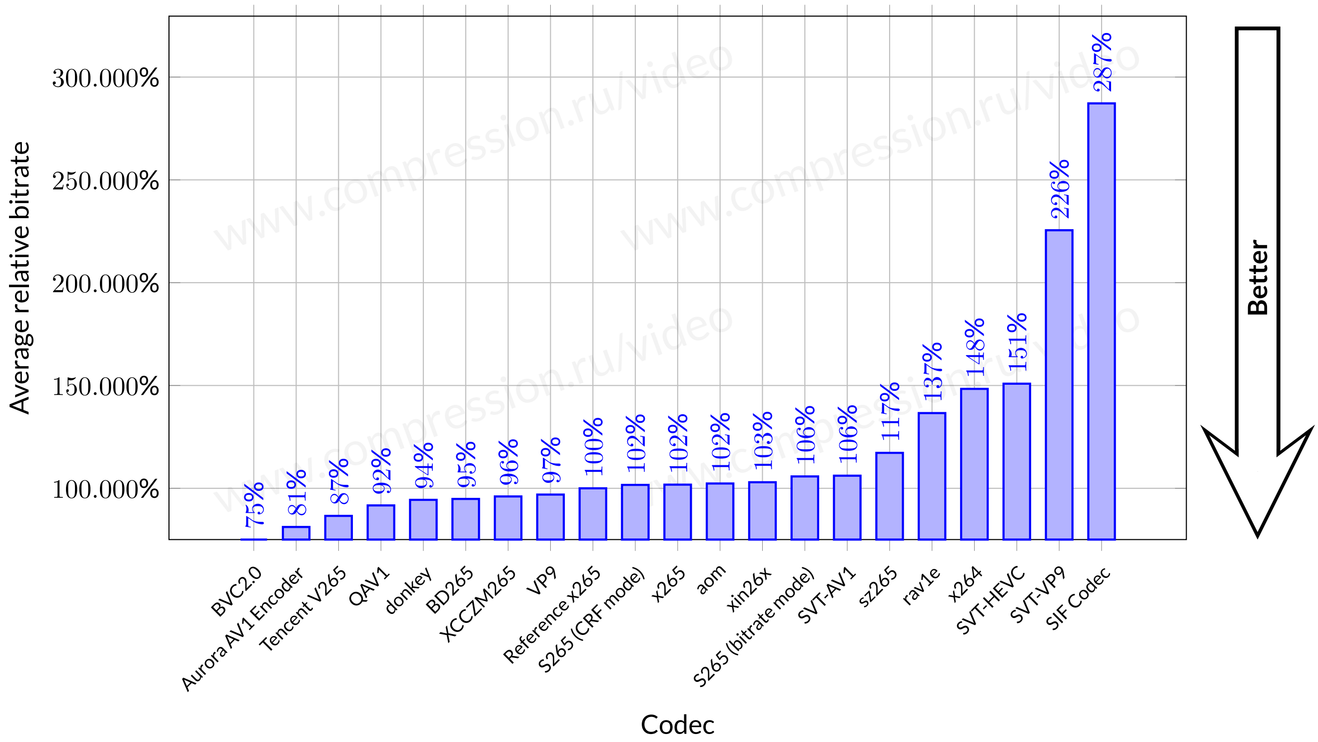
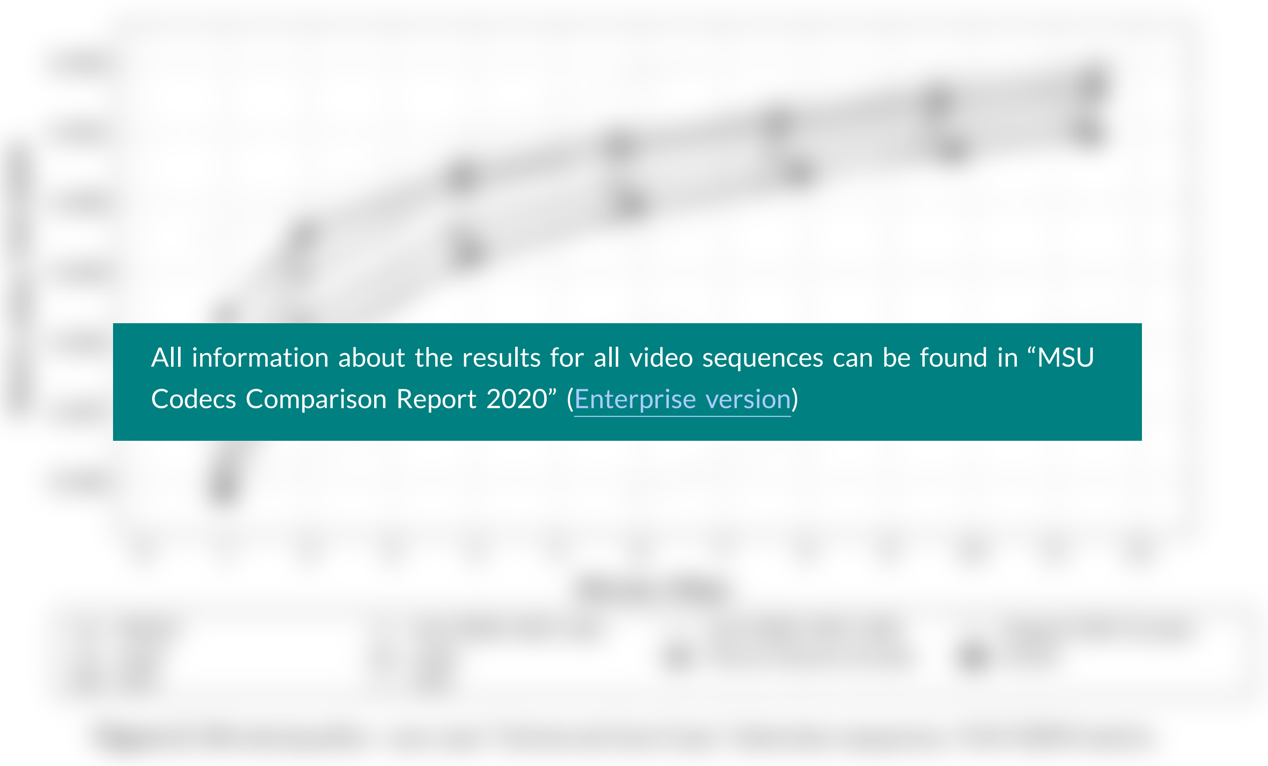


Figure 9: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, Y-VMAF (v0.6.3) metric.

6. ONLINE (30 FPS)

6.1. RD Curves

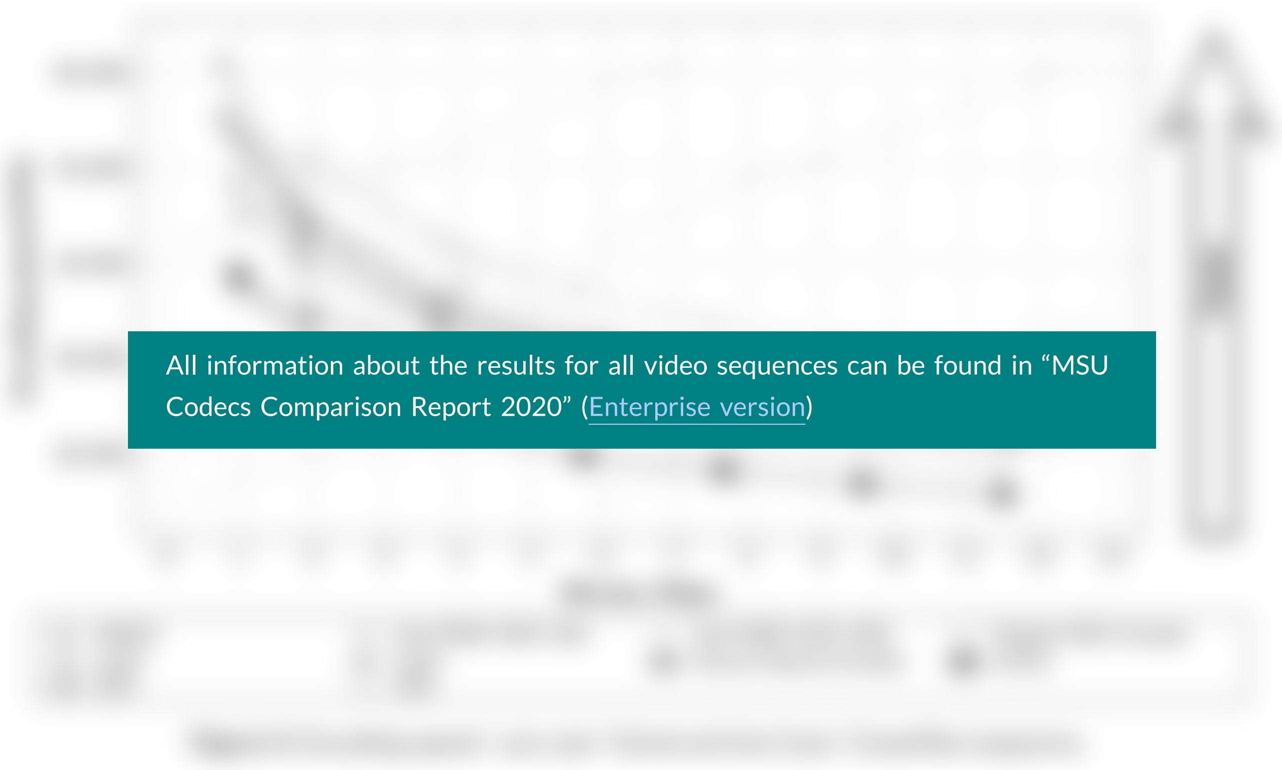
Judging from the mean quality scores (computed using the method described in Section E), first place in the quality competition goes to **Tencent V265**, second place goes to **XCCZM265**, and third place to **S265 (CRF mode)**.



All information about the results for all video sequences can be found in “MSU Codecs Comparison Report 2020” (Enterprise version)

6.2. Encoding Speed

Judging from the mean speed scores (computed using the method described in Section E), first place in the speed competition goes to **SVT-VP9** and **x264**, second place goes to **xin26x**, and third place to **SVT-HEVC** and **sz265**.

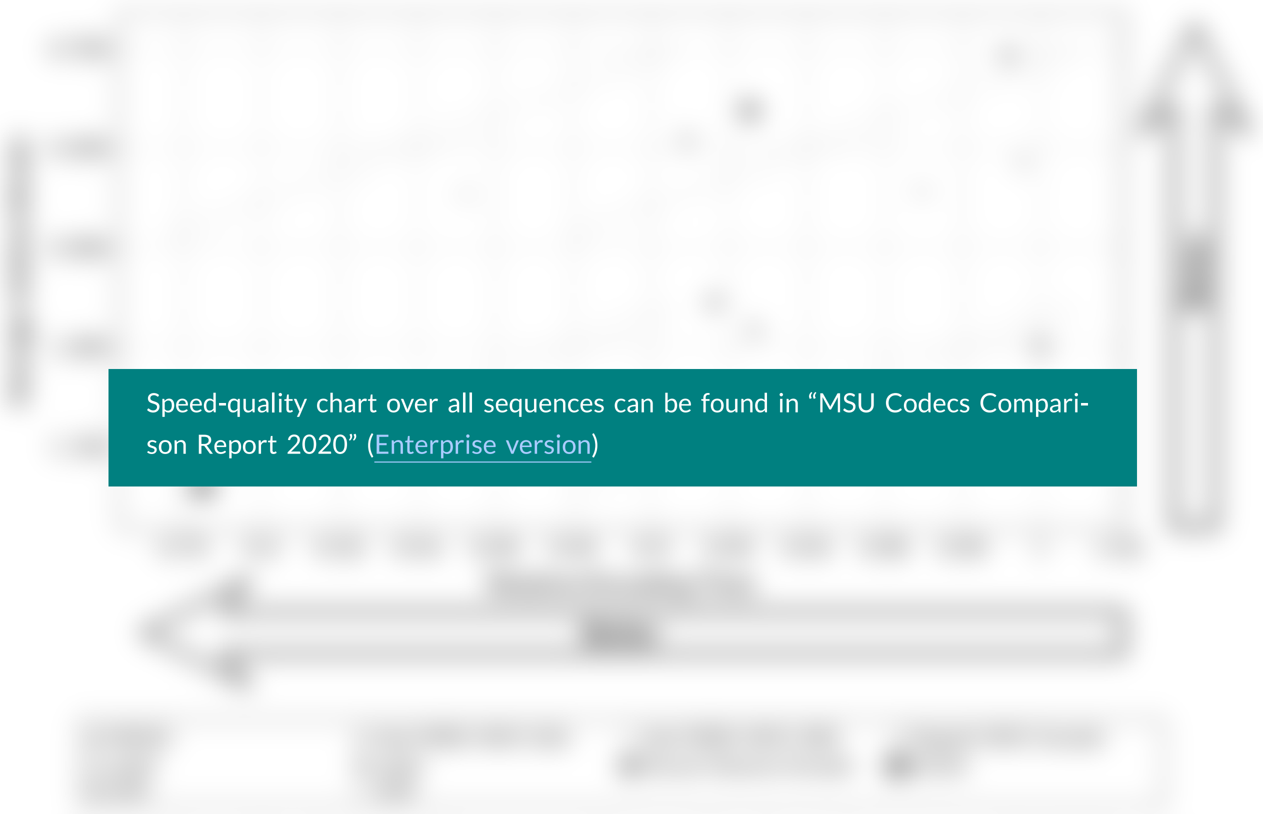


All information about the results for all video sequences can be found in "MSU Codecs Comparison Report 2020" (Enterprise version)

6.3. Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs are in Appendix E. 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 four Pareto-optimal encoders: **Tencent V265**, **xin26x**, **x264**, and **SVT-VP9**.



Speed-quality chart over all sequences can be found in "MSU Codecs Comparison Report 2020" (Enterprise version)

6.4. Online (30 fps) YUV-SSIM

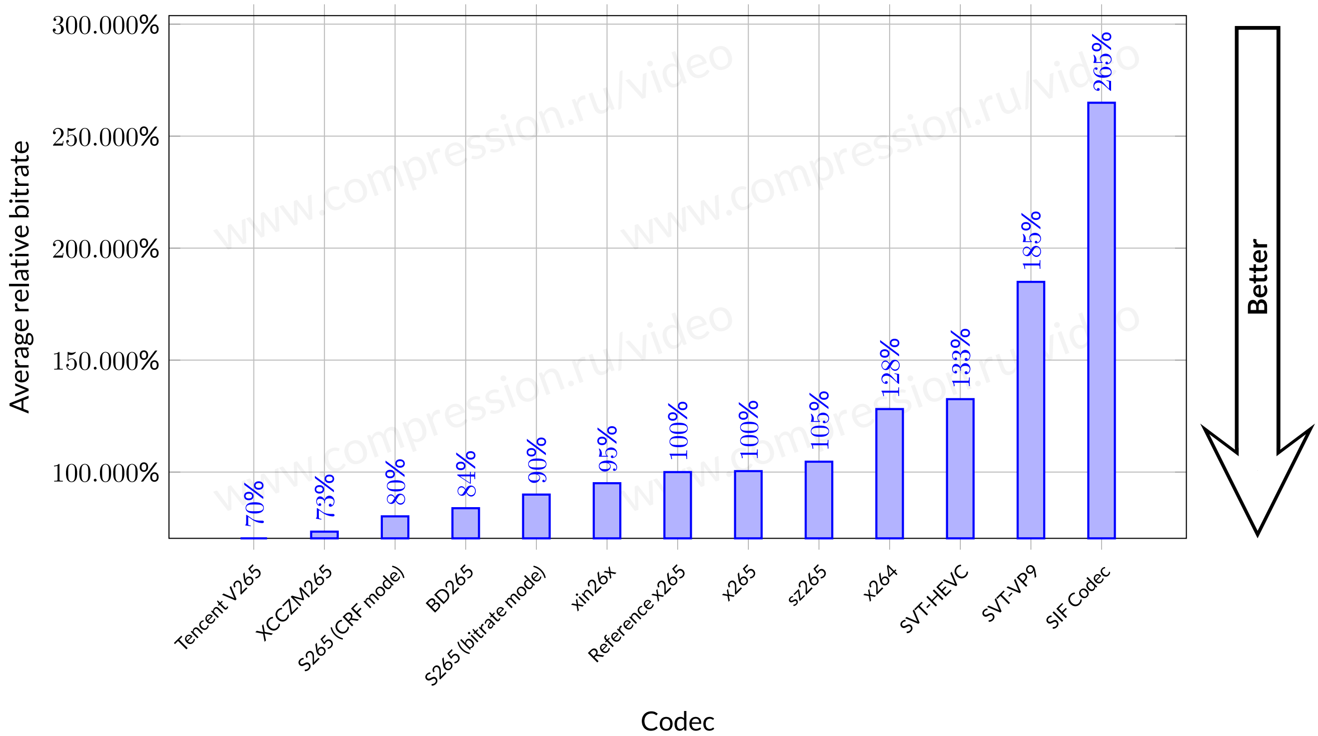


Figure 10: Average bitrate ratio for a fixed quality—use case “Online (30 fps),” all sequences, YUV-SSIM metric.

6.5. Online (30 fps) YUV-PSNR (avg. MSE)

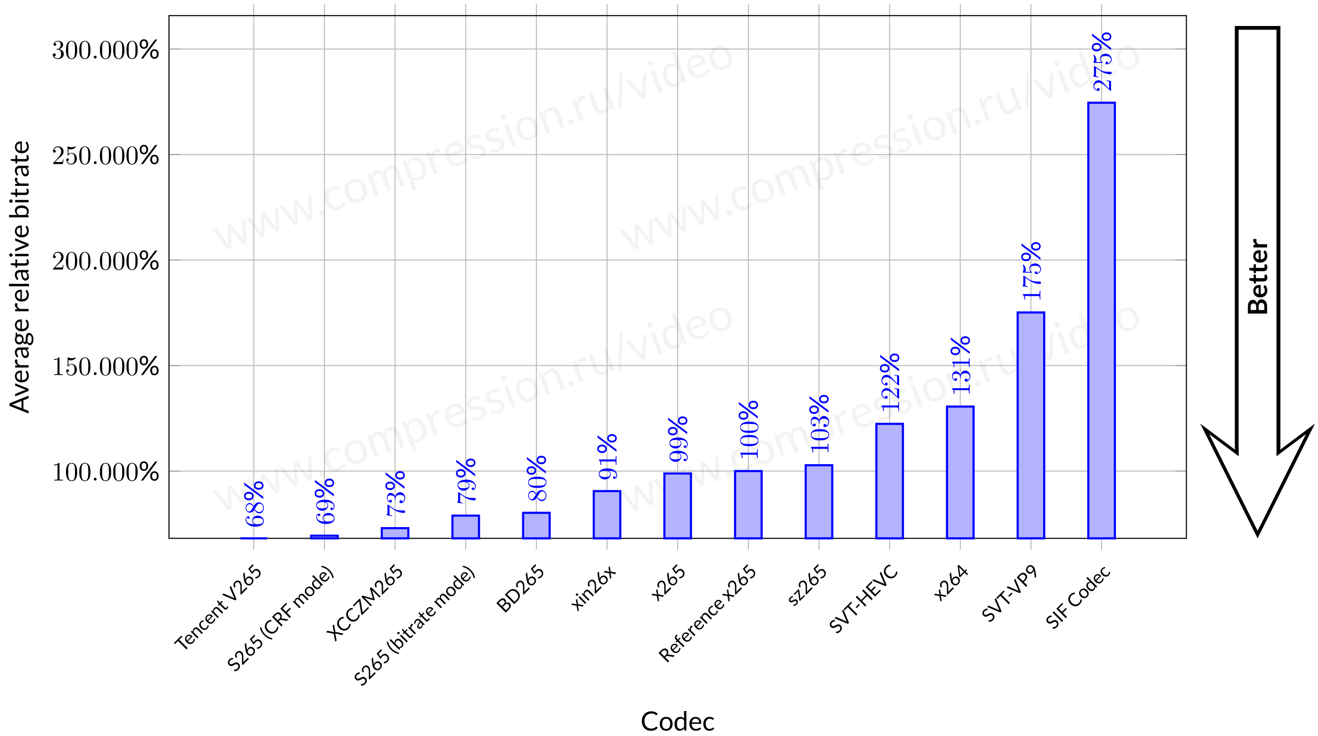


Figure 11: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-PSNR (avg. MSE) metric.

6.6. Online (30 fps) YUV-PSNR (avg. log)

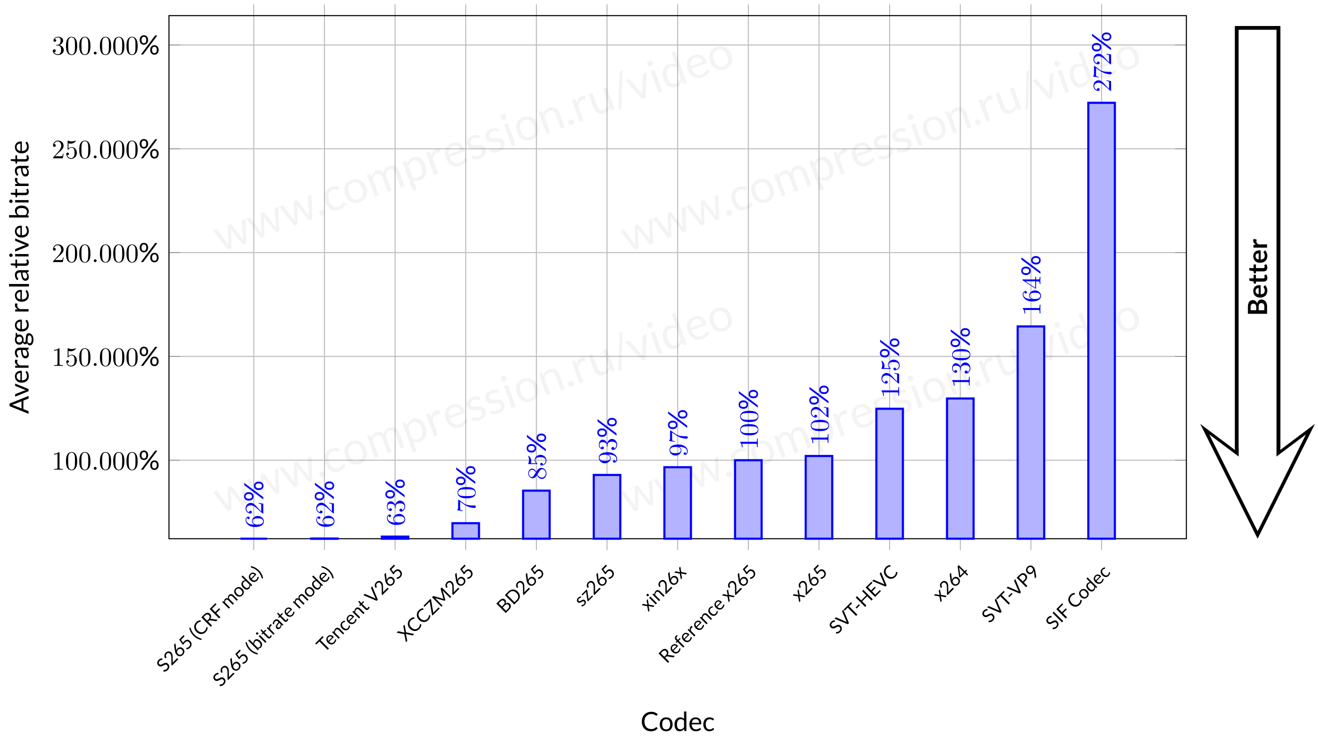


Figure 12: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-PSNR (avg. log) metric.

6.7. Online (30 fps) Y-VMAF (v0.6.3)

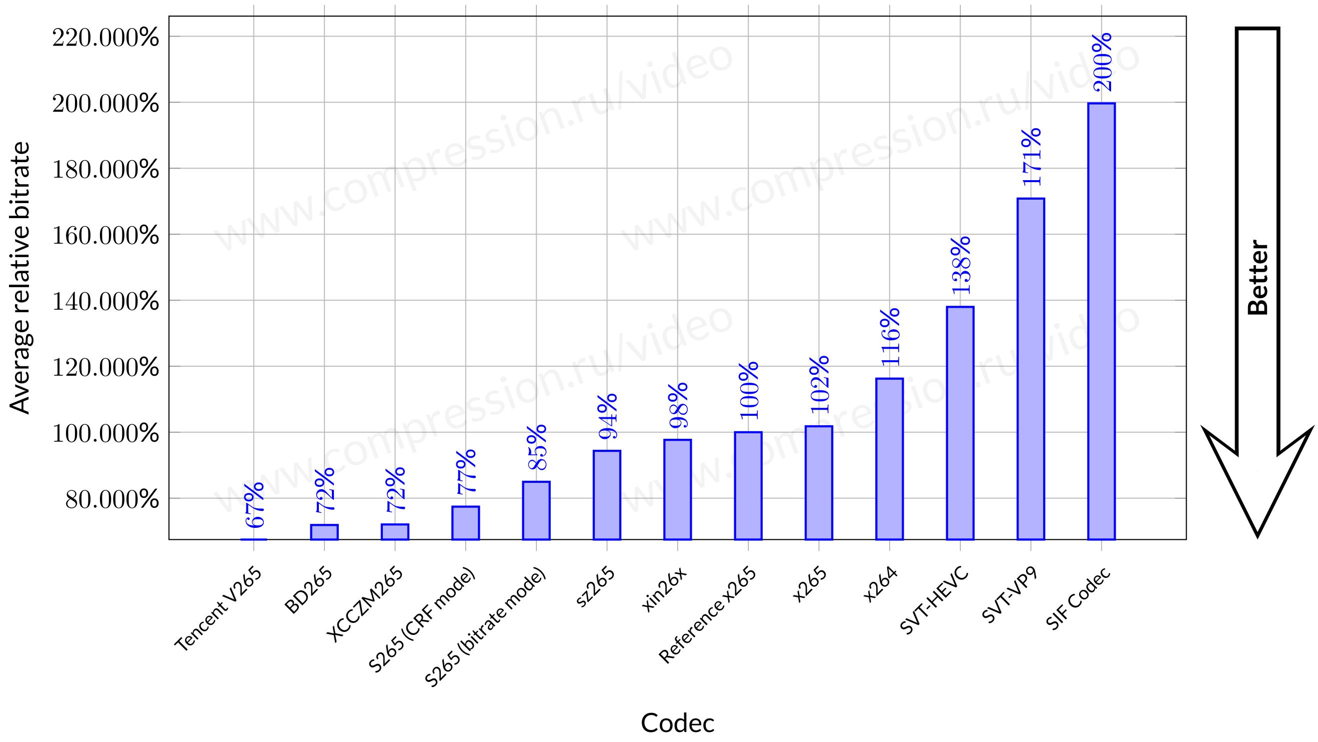


Figure 13: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, Y-VMAF (v0.6.3) metric.

7. CONCLUSION

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

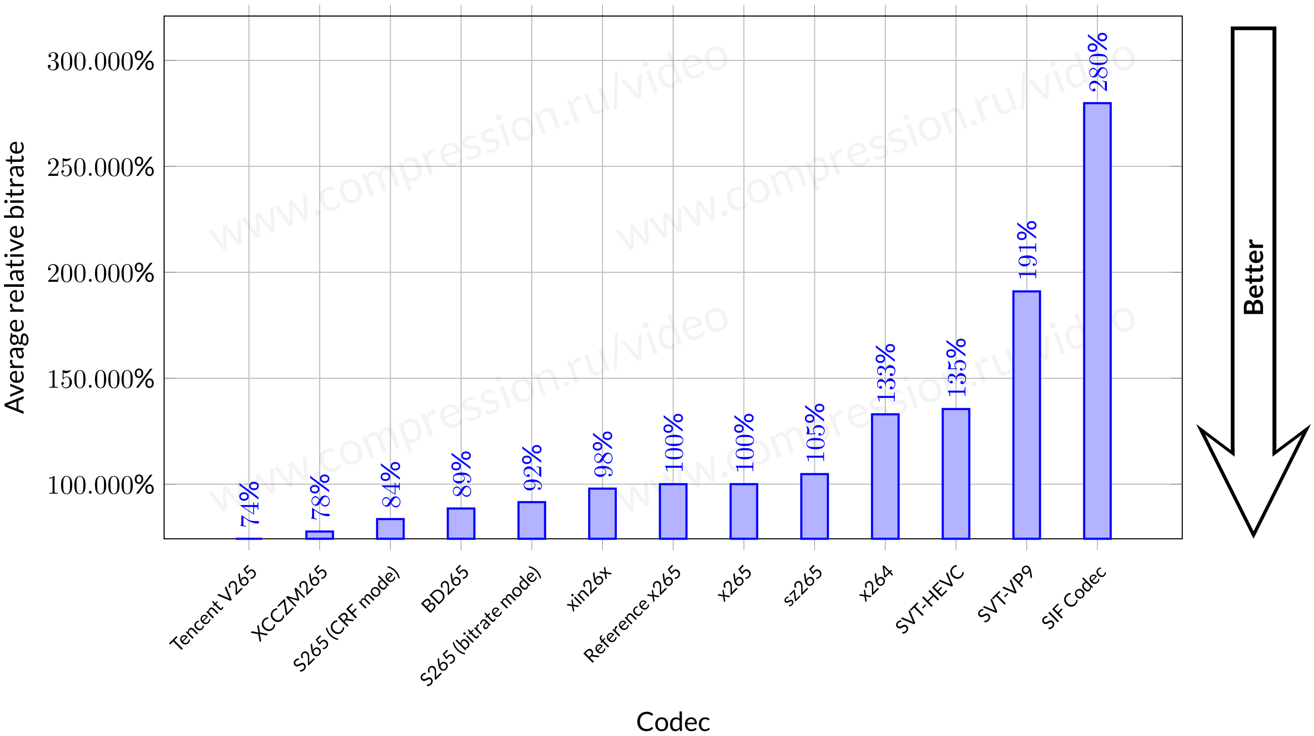


Figure 14: Average bitrate ratio for a fixed quality—all sequences, YUV-SSIM metric.

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

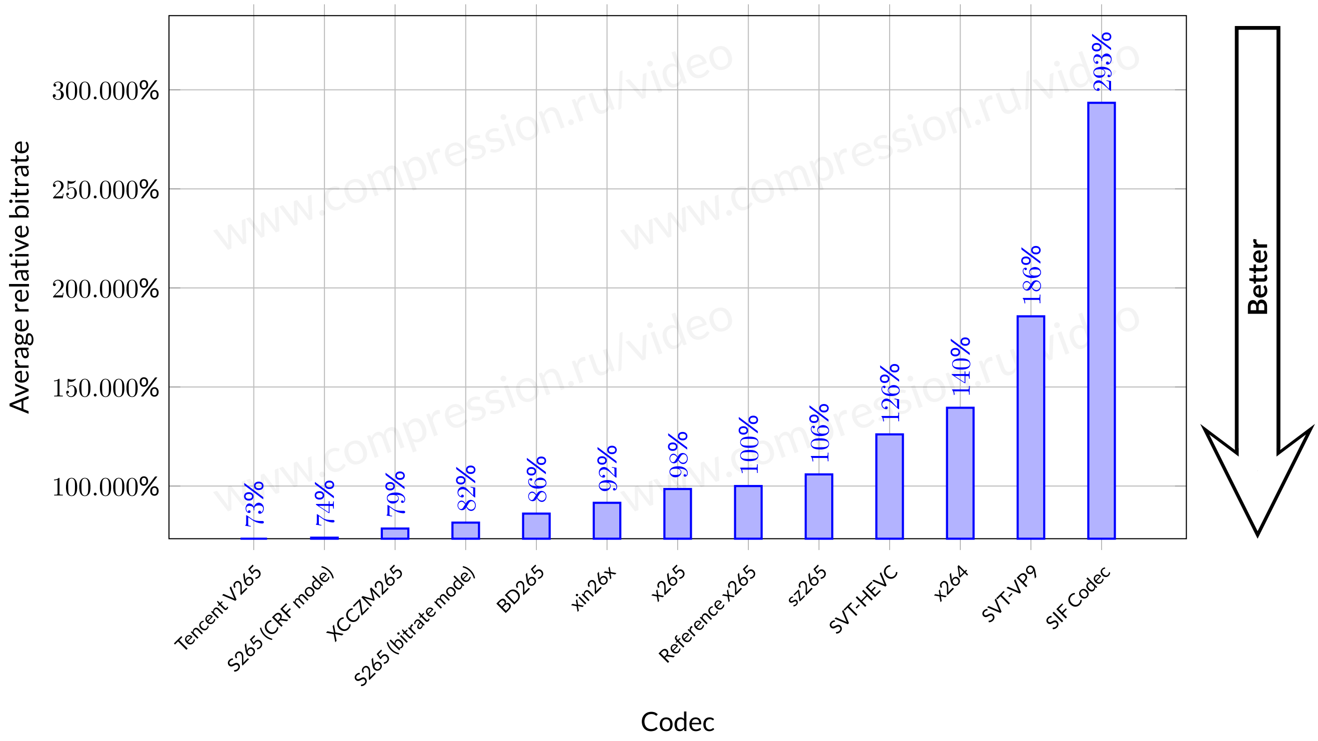


Figure 15: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. MSE) metric.

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

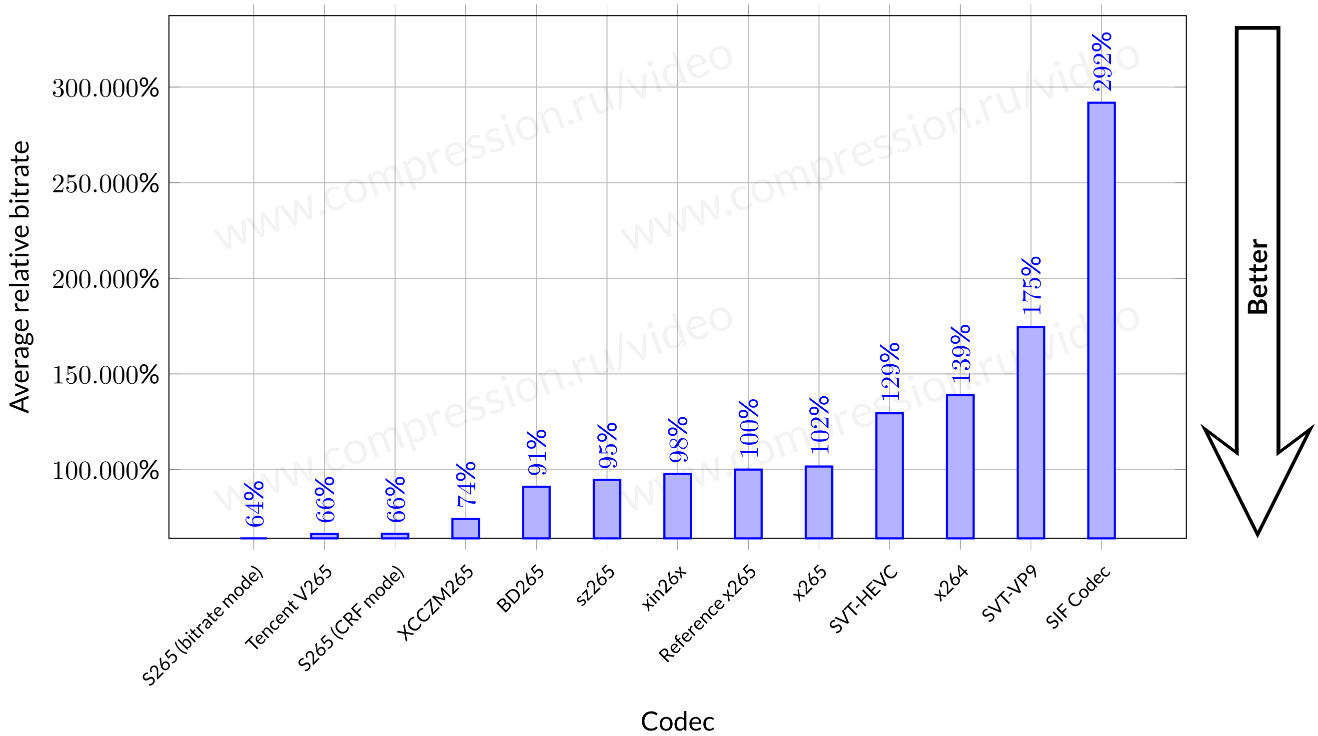


Figure 16: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. log) metric.

7.4. Overall Y-VMAF (v0.6.3) (for all use cases)

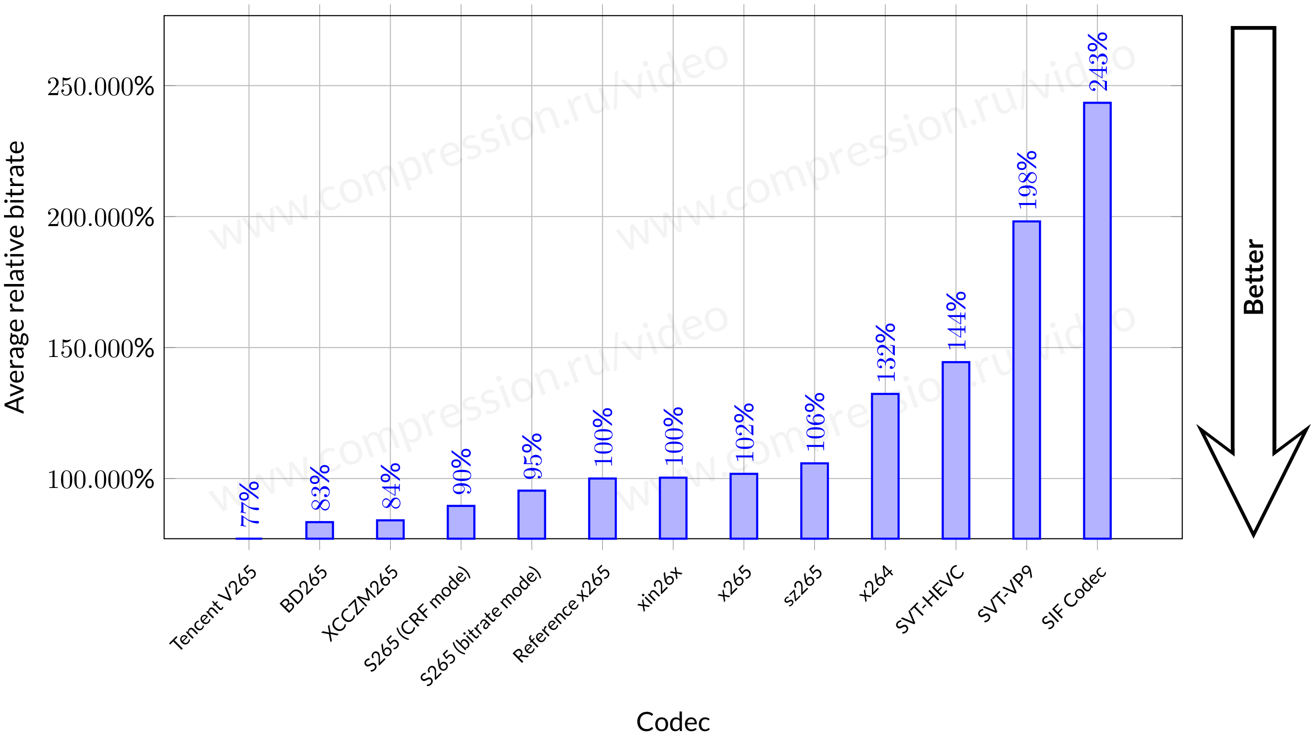


Figure 17: Average bitrate ratio for a fixed quality—all sequences, Y-VMAF (v0.6.3) metric.

A. PARTICIPANTS' COMMENTS

A.1. ChangKuoLao (Donkey codec)

Donkey codec achieves a much better performance in the PSNR of frame average with a significant leads over the open source codecs.

A.2. SIF Codec LLC

Due to a pre-compressed status of the absolute majority of the testing sequences, SIF Codec has certain disadvantages compared with other codec which are based on traditional architecture. We believe that the most objective observation is done based on following raw video sequences:
Ariadnes_thread_1920x1080_30.yuv
Apple_tree_1920x1080_30.yuv
Crowd_run_1920x1080_50.yuv

A.3. xin26x

Xin26x is originally a personal RTC video encoder. In the future, more offline coding tools will be added to make PSNR better. In addition, AV1 and VVC are supported in Xin26x. Check https://github.com/pigpeppa/xin26x for more information.

A.4. Bytedance Inc. (BVC2.0 codec)

BVC2.0 was developed and optimized (only for the PSNR metric during this year's comparison event) by a small group of ByteDancers in a short period. Thanks to their continued dedication, the encoder has been further improved for all assessment metrics, and it is expected to be applied to a variety of video applications. Finally, we'd also like to express our gratitude for the generous assistance by the MSU Video Group.

A.5. Alibaba Group (S265 codec)

S265 developed by Alibaba Taobao and Aliyun joint team, referenced some rate-control technique from Prof. Zhenyu Liu and Prof. Xi-angyang Ji of Tsinghua University.
PSNR is a traditional criterion for codec's quality, It's simple, accurate and widely used in signal compression domain. Alibaba's S265 codec achieves best performance in the avg log PSNR with a significant leads over all codecs, including AV1, HEVC, VP9, AVC. It's the best PSNR codec.

A.6. rav1e

The low memory footprint of rav1e is suitable for parallel independent encodes on the same hardware. The balance between compression efficiency and encoder complexity is comparable with other popular open source encoders.

B. SEQUENCES

Full descriptions of all videos used in this comparison are presented on a project page and in separate PDF, provided with this report.

C. CODECS

All tested encoders presets can be found in "MSU Codecs Comparison Report 2020" (Enterprise version)

D. 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 18 shows the bit rate distributions for our video data set by years. Table Table 3 shows the number of videos in our video collection.

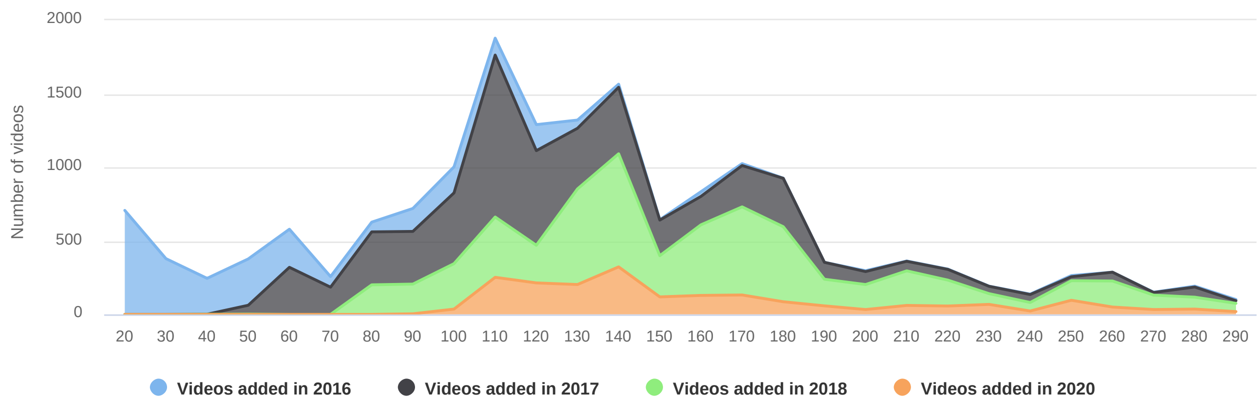


Figure 18: Bit rate distributions for comparison video set.

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.

We resized and cropped 4K videos to FullHD resolution 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. [4] 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. Distribution of obtained samples compared to samples from previous codec comparisons is shown in Figure 19.

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

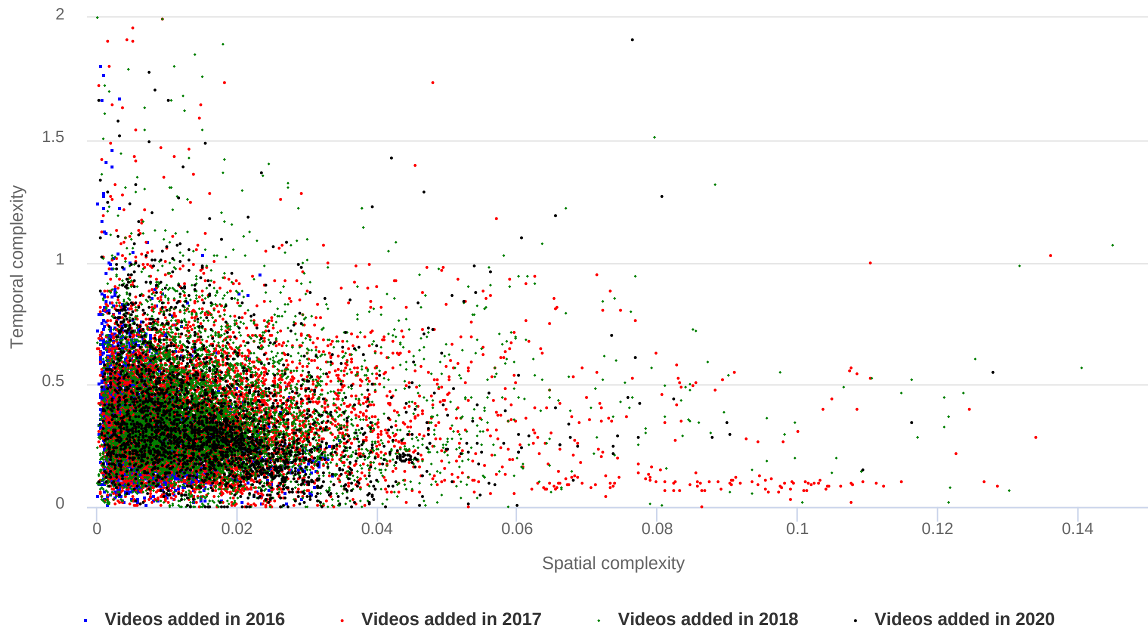


Figure 19: Distribution of obtained samples.

This year, we conducted a voting to choose final set of 50 videos for the comparison. Participation in video selection was optional. We divided the video collection into 50 clusters. For each cluster, we randomly selected from 2 to 6 candidate videos that were close to the cluster centre and that had a license enabling derivatives and commercial use. Figure 20 shows the cluster boundaries and constituent sequences.

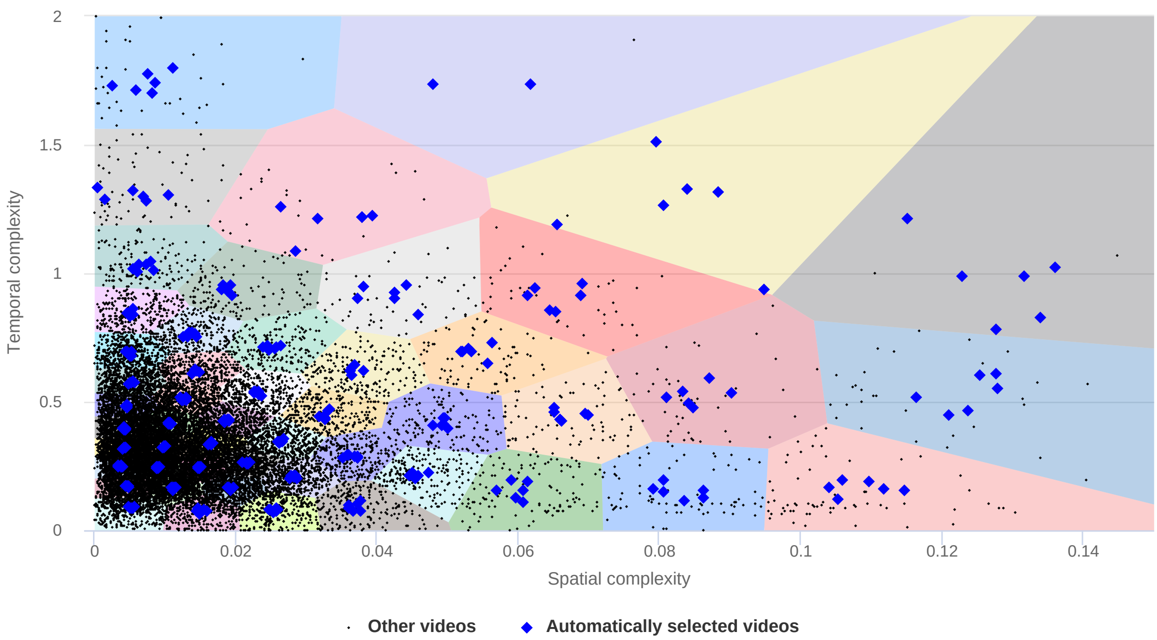


Figure 20: Segmentation of samples.

All comparison participants were invited to participate in video selection, and seven took part in it. Also, four members of codecs comparison group (Dr Dmitriy Kulikov, Anastasia Antsiferova, Egor Sklyarov and Nickolay Safonov) and independent industry expert (Jan Ozer `https://streaminglearningcenter.com/about-jan-ozer`) took part in voting for final video set. Table 4 contains information about video selection participants.

Voter	Number of clusters to vote	Number of received votes
Dr D. Kulikov	50	50
J. Ozer	50	50
Egor Sklyarov	50	50
Anastasia Antsiferova	30	29
Nickolay Safonov	30	28
Participant #0	15	15
Participant #1	15	15
Participant #2	15	15
Participant #3	15	12
Participant #4	15	15
Participant #5	15	15
Participant #6	15	15
Participant #7	15	15

Table 4: Voted members of video selection.

For every participant, only a subset of clusters is available for voting. Each participant was suggested to choose one video in each of 15 given clusters. These clusters were chosen randomly, overlapped for different voters and equally covered all 50 clusters. A participant was able to change a vote until the end of voting. Fig. 21 shows the interface of video selection platform.

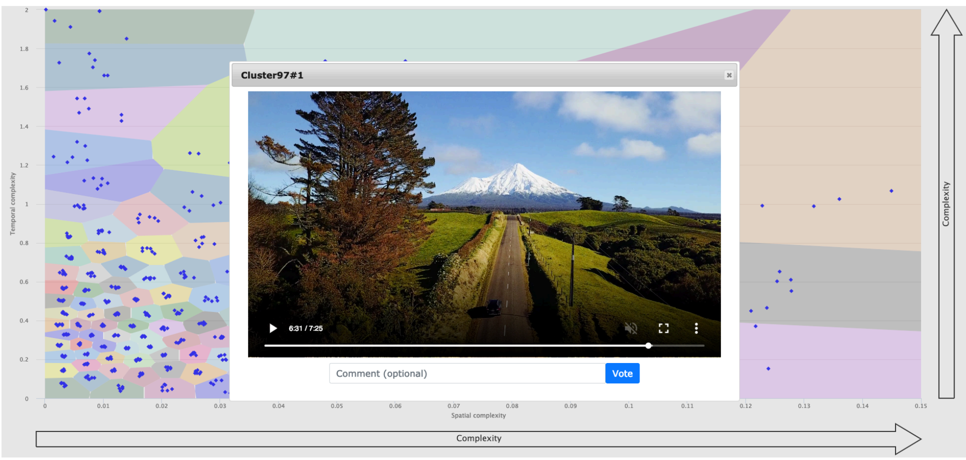


Figure 21: Video selection platform interface.

At the end of voting, videos with the highest number of votes were selected for the final comparison set. List of final videos and votes for them is presented in separate PDF with videos descriptions, and their distribution in SI/TI space among all videos from collection is shown in Fig. 22.

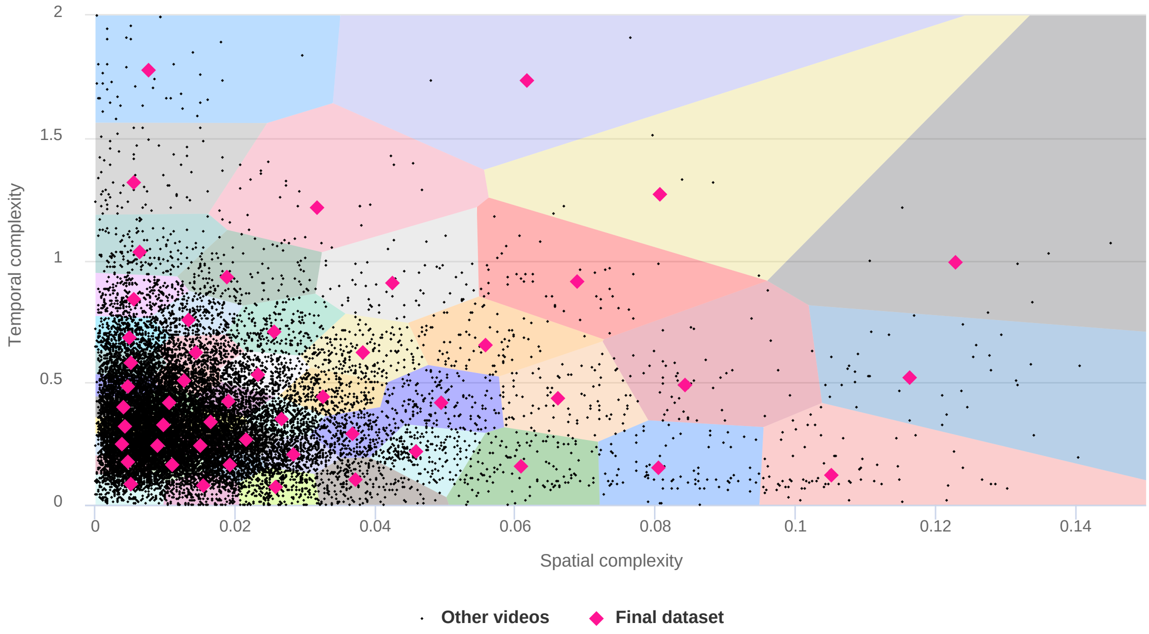


Figure 22: Distribution of sequences in final set.

The new data set consists of 50 sequences, the complete list of sequences appears in Appendix B.

E. 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.

E.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.

E.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.

E.3. Graph Example

Figure 23 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.

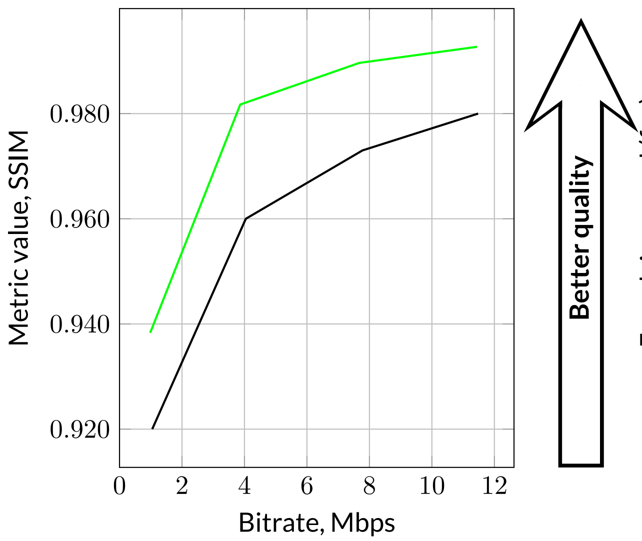
Owing to these advantages, we frequently use relative-bitrate/relative-time graphs in this report because they assist in evaluating the codecs in the test set, especially when the number of codecs is large.

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

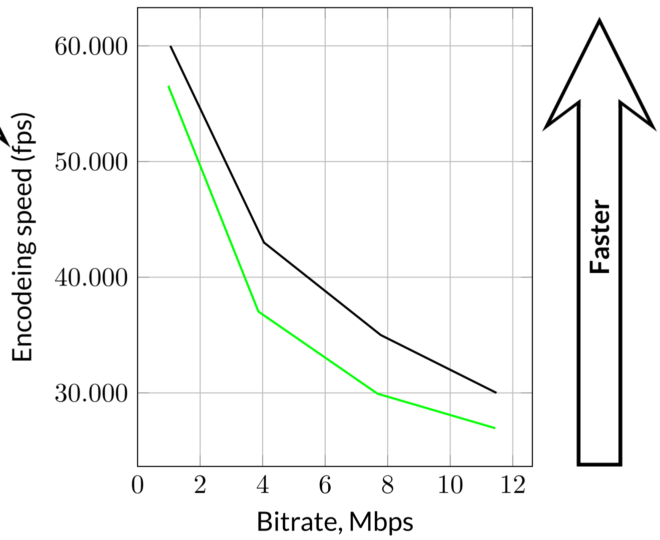
E.4. Bitrate Ratio for the Same Quality

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 24b). All further computations use the inverted graph.

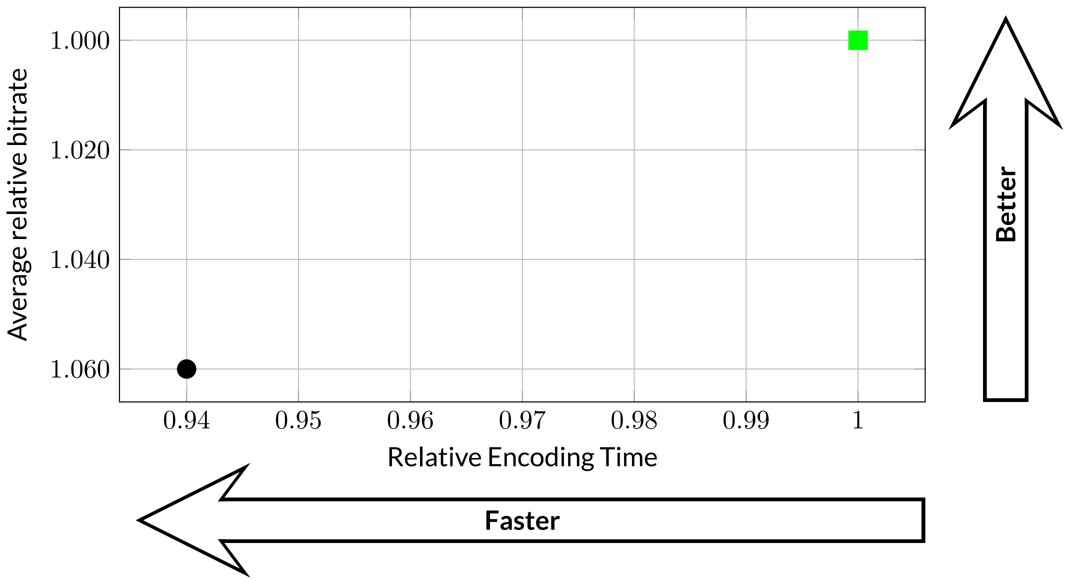
The second step involves averaging the interval over which the quality axis is chosen. The averaging is only over those segments for which both codecs yield results. This limitation is due to the difficulty of developing extrap-



(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 23: Speed/Quality trade-off example

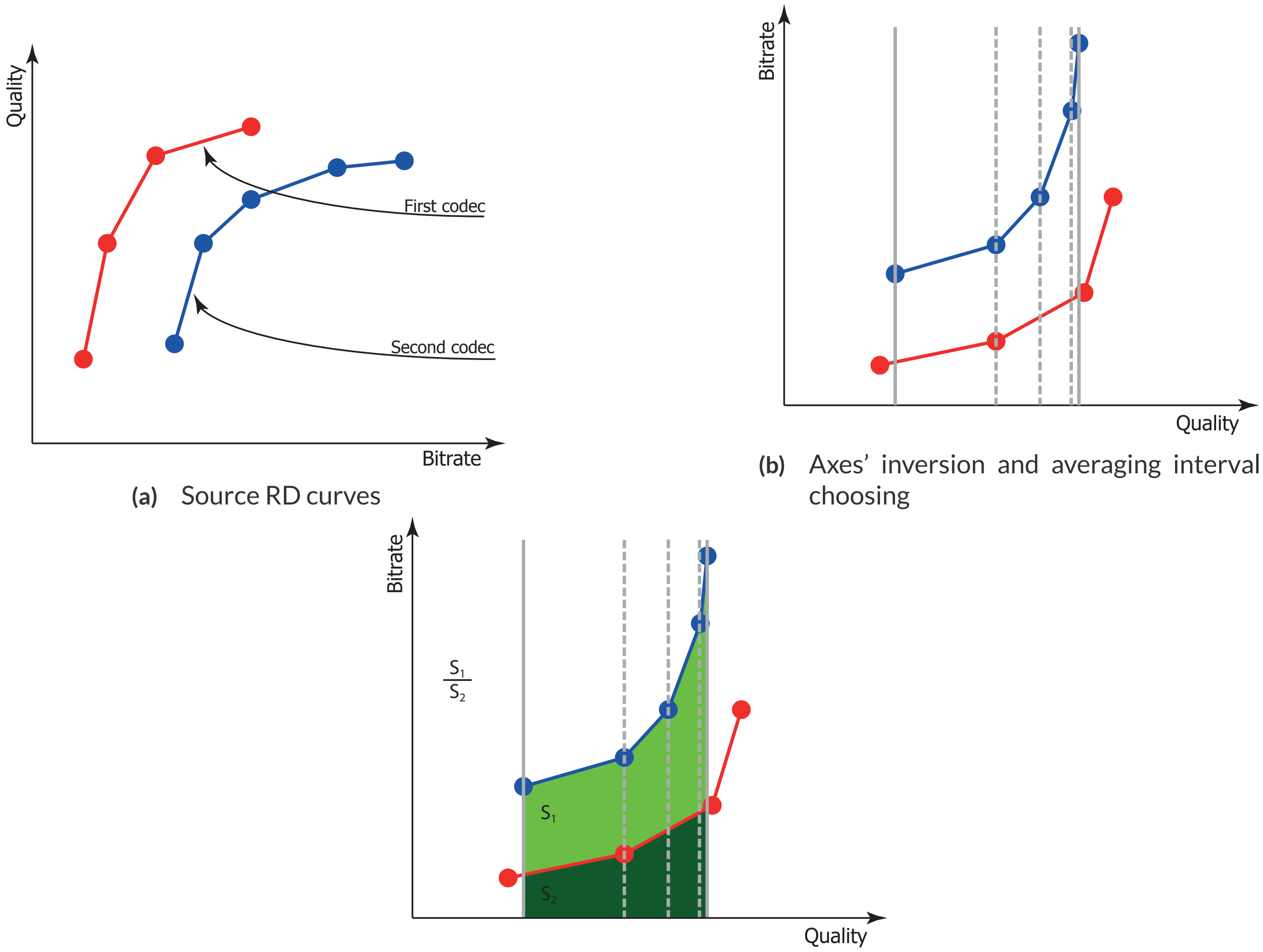


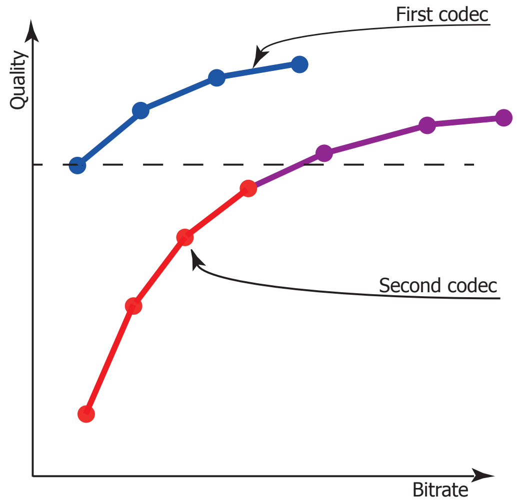
Figure 24: Average bitrate ratio computation

olation methods for classic RD curves; nevertheless, even linear methods are acceptable when interpolating RD curves.

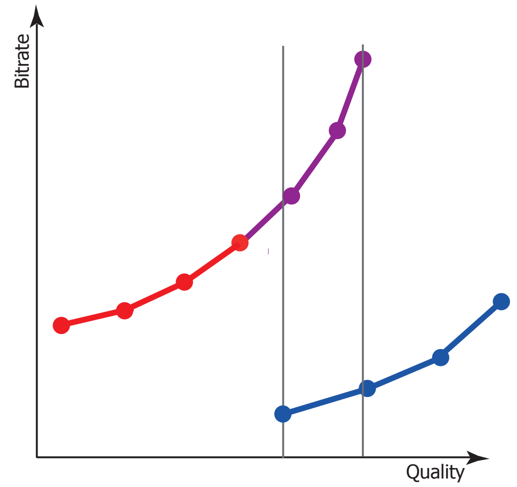
The final step is calculation of the area under the curves in the chosen interpolation segment and determination of their ratio (see Figure 24c). This result is an average bitrate ratio at a fixed quality for the two codecs. When considering more than two codecs, one of is defined as a reference codec, and the quality of the others is compared with that of the reference.

E.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 25) 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

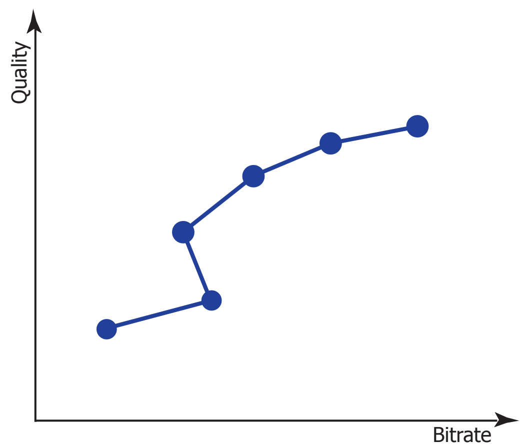


(b) Axes' inversion and averaging interval choosing

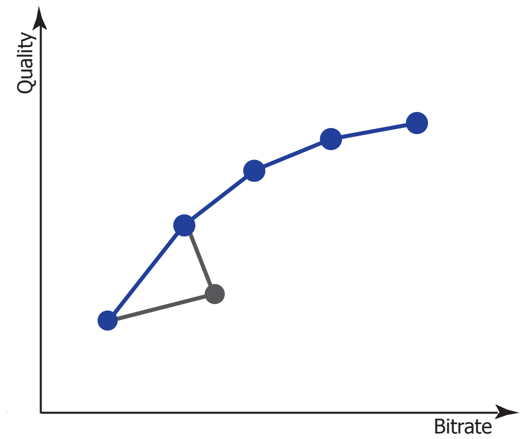
Figure 25: Measuring codec on additional bitrates to make it cross with other codecs over the quality axis.

E.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 26.



(a) Non-monotonic RD-curve.



(b) Points that were used to calculate integral.

Figure 26: Processing non-monotonic RD-curves.

F. OBJECTIVE-QUALITY METRIC DESCRIPTION

F.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.

F.1.1. Brief Description

Wang, et al.[5] 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

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

where

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

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

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

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 $K1, K2 \ll 1$.

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.

[5] 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.

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.

F.1.2. Examples

Figure 27 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 27: SSIM example for compressed image

Figure 28 depicts various distortions applied to the original image, and Figure 29 shows SSIM values for these distortions.

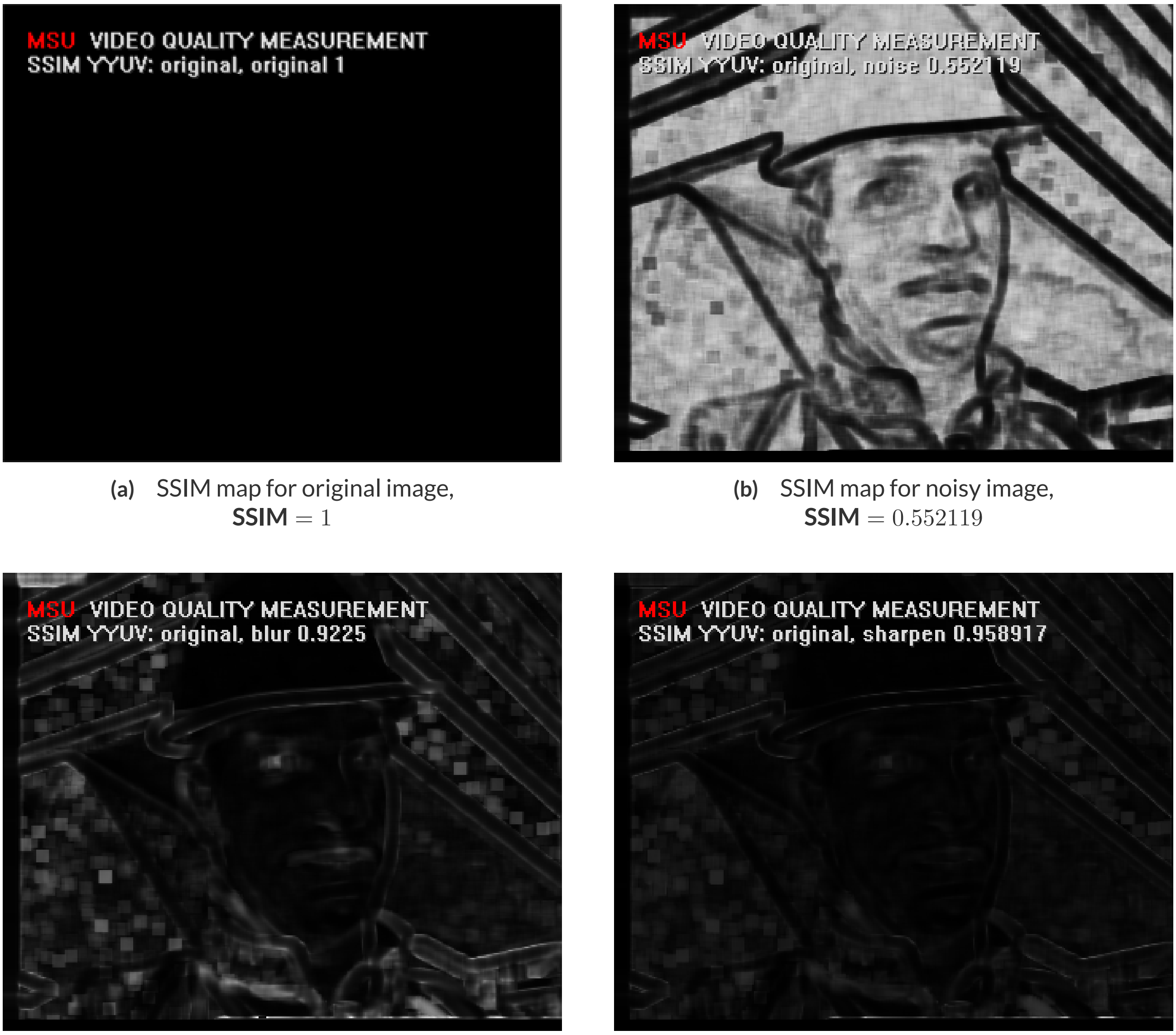
(a) Original image

(b) Image with added noise

(c) Blurred image

(d) Sharpen image

Figure 28: 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 29: SSIM values for original and processed images

F.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.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 $n \times m$:

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

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

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

F.2.1. PSNR (avg. MSE)

For two videos V and $\hat{V}$:

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

F.2.2. PSNR (avg. log)

For two videos V and $\hat{V}$:

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

G. 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

GRAPHICS & MEDIA LAB

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 PSNR
VQM	MSU Drop Frame Metric
SSIM	MSU Noise Estimation Metric
MS-SSIM	MSU Scene Change Detector
3-SSIM	MSU Blocking Metric
VMAF	NIQE (no-reference comparison)

HDR support

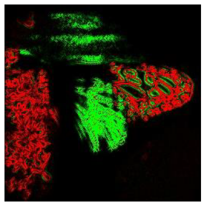
Hundreds Video and 30+ Image Formats

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

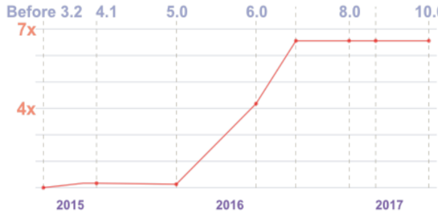
Visualization Examples

Allows easily detect where codec/filter fails



MSU Blurring Metric

MSU Blocking Metric



VQMT average Speedup

Fastest Video Quality Measurement

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

Multi-core Processors Support

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:

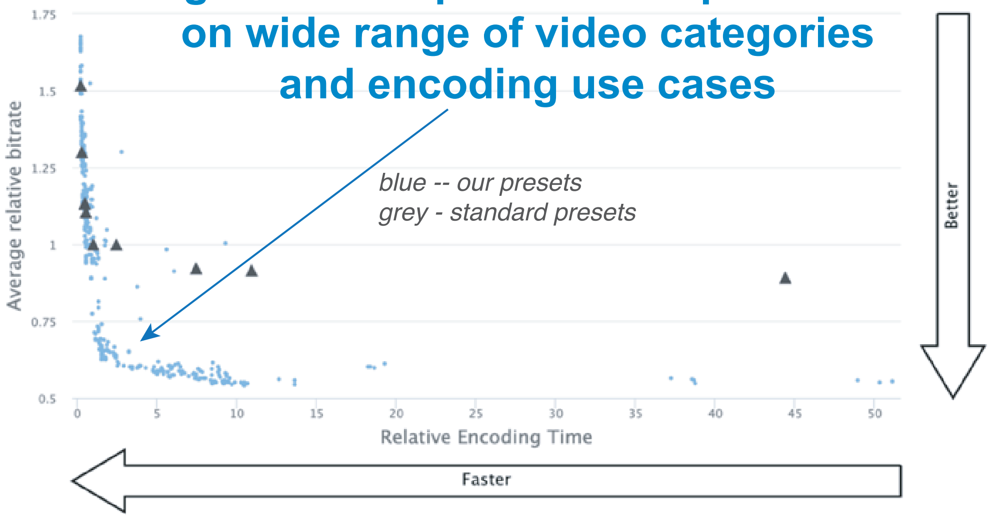
Video-Codec Advanced Tuning

Boost up your codec without changing code!

Effective Video Transcoding

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 ~2.2*10^13 computing centuries (~488 000 Earth ages)

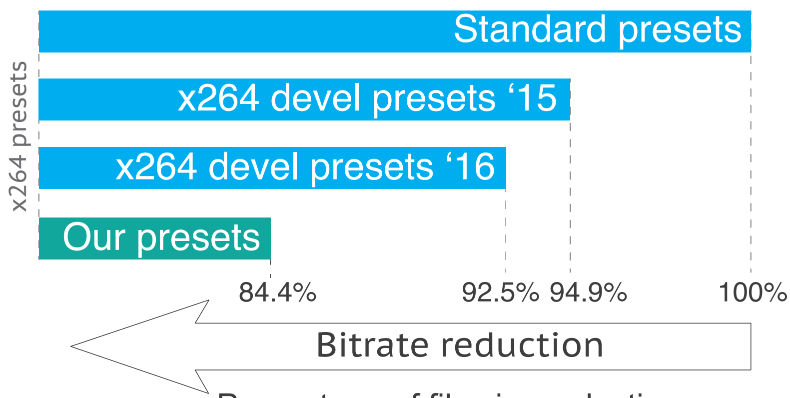
15% bitrate savings in average

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!



Percentage of file size reduction in average for a set of 77 UGC-videos

We can find best encoding presets for your videos

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

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

evt@compression.ru

In cooperation with Lomonosov MSU Graphics&Media Lab