MSU Video Codecs Comparison 2020

Part II: FullHD Content, Subjective 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

- Alibaba S265
- Reference x265
- x265
- XCCZM265

AV1

- KAV1_v1
- KAV1_v2
- QAV1
- rav1e

Other

- BVC2
- SIF Codec
- x264

CS MSU Graphics & Media Lab, Video Group
January 21, 2022
http://www.compression.ru/video/codec_comparison/index_en.html
videocodec-testing@graphics.cs.msu.ru

Powered by Subjectify.us

1. REPORT VERSIONS

	Free version	**Enterprise version**
Use cases	Universal (partially)	Fast, Offline
Per-sequence-results	1 of 8 sequences (only Offline use case)	All 8 sequences for all use cases (in interactive charts)
Metric: YUV-Subjective, YUV-SSIM (overall results), 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	54 pages	69 pages
HTML report	5 interactive charts	540+ 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:

- Alibaba Taobao codec Team
- Bytedance Inc.
- CKL DXX ZWJ XQQ CC ZHB ZZW
- iQIYI Inc.
- MulticoreWare, Inc.
- SIF Codec LLC
- The rav1e contributors
- 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. About Subjectify.us

We obtained the subjective scores for this study using Subjectify.us. This platform enables researchers and developers to conduct subjective comparisons of image- and video-processing methods (e.g., compression, inpainting, denoising, matting, etc.) and carry out studies of human quality perception.

To conduct a study, researchers must apply the methods under comparison to a set of test videos (images), upload the results to Subjectify.us and write a task description for study participants. Subjectify.us handles all the laborious steps of a crowdsourced study: it recruits participants, presents uploaded content in a pairwise fashion, filters out responses from participants who cheat or are careless, analyzes collected results, and generates a study report with interactive plots. Thanks to the pairwise presentation, researchers need not invent a quality scale, as study participants just select the best option of the two. The platform is optimized for comparison of large video files: it prefetches all videos assigned to a study participant and loads them into his or her device before asking the first question. Thus, even participants with a slow Internet connection won't experience buffering events that might affect quality perception.

To try the platform in your research project, reach out to `www.subjectify.us`. This demo video shows an overview of the Subjectify.us workflow.

3.2. Sequences

	Sequence	Number of frames	Frame rate	Resolution
1.	blue_hair	300	30	1920×1080
2.	cover_song	300	30	1920×1080
3.	crowd_run	500	50	1920×1080
4.	forest_dog	250	25	1920×1080
5.	gaming	300	30	1920×1080
6.	okeechobee	250	25	1920×1080
7.	pyranha_rafting	240	24	1920×1080
8.	road_timelapse	300	30	1920×1080

Table 1: Summary of video sequences

All sequences were trimmed to 10 seconds duration for both subjective and objective evaluation.

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

3.3. Codecs

Codec	Developer	Version	OS
Alibaba S265	Alibaba Taobao codec Team	S265 v5	Windows
BVC2	Bytedance Inc.	V1	Windows
KAV1_v1	CKL DXX ZWJ XQQ CC ZHB ZZW	8ec71bcc	Windows
KAV1_v2	CKL DXX ZWJ XQQ CC ZHB ZZW	04ea2aa4	Windows
QAV1	iQIYI Inc.		Linux
rav1e	The rav1e contributors		Windows
Reference x265	MulticoreWare, Inc.		Windows
SIF Codec	SIF Codec LLC		Windows
x264	x264 project		Windows
x265	MulticoreWare, Inc.		Windows
XCCZM265	XCCZM Codec Team		Linux

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 D 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 subjective study as well as objective assessment methods to compare the encoding quality of recent HEVC encoders as well as encoders implementing other standards. This effort employed 8 video sequences at 1080p resolution to evaluate codec performance. 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 E).

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.

For subjective quality measurements we used Subjectify.us crowdsourcing platform. We involved 6,100+ participants. After deleting replies from bots we got 236,736 pairwise answers. Bradley-Terry model was used to compute global rank.

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

We didn't use no-reference metrics because most of them 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 F), first place in the quality competition goes to **BVC2**, second place goes to **Alibaba S265**, and third place to **KAV1_v2**.

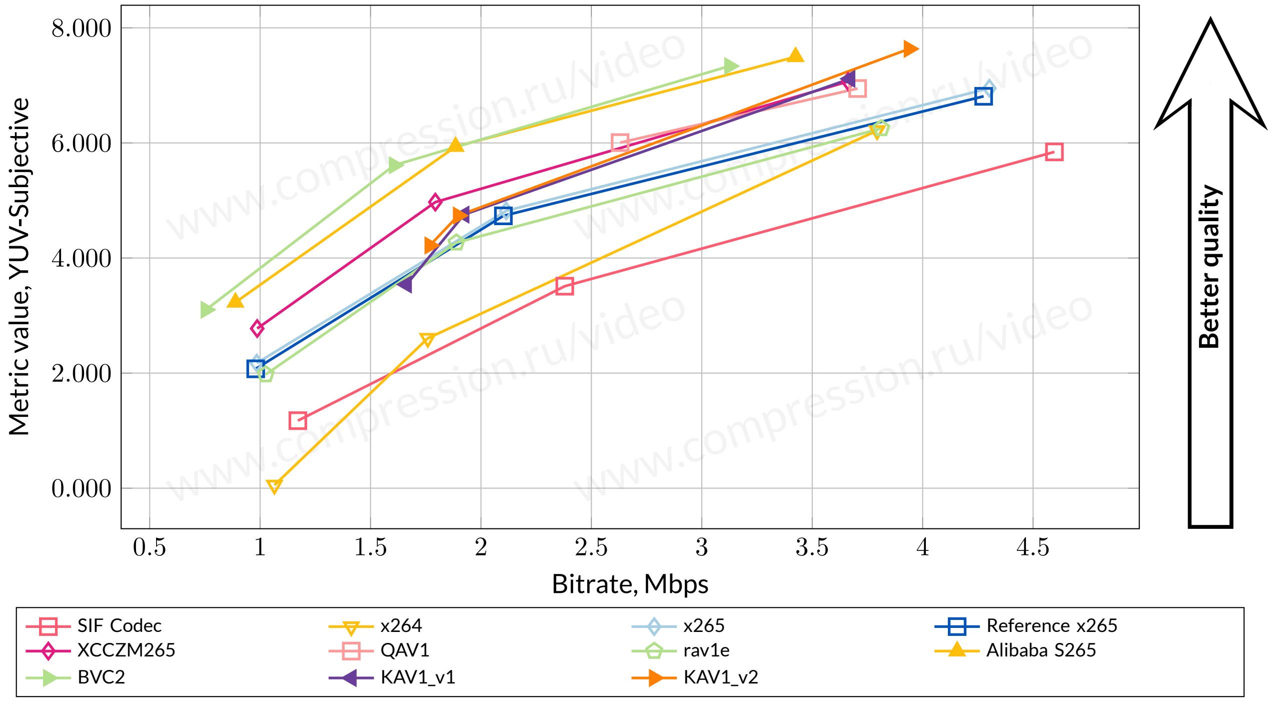
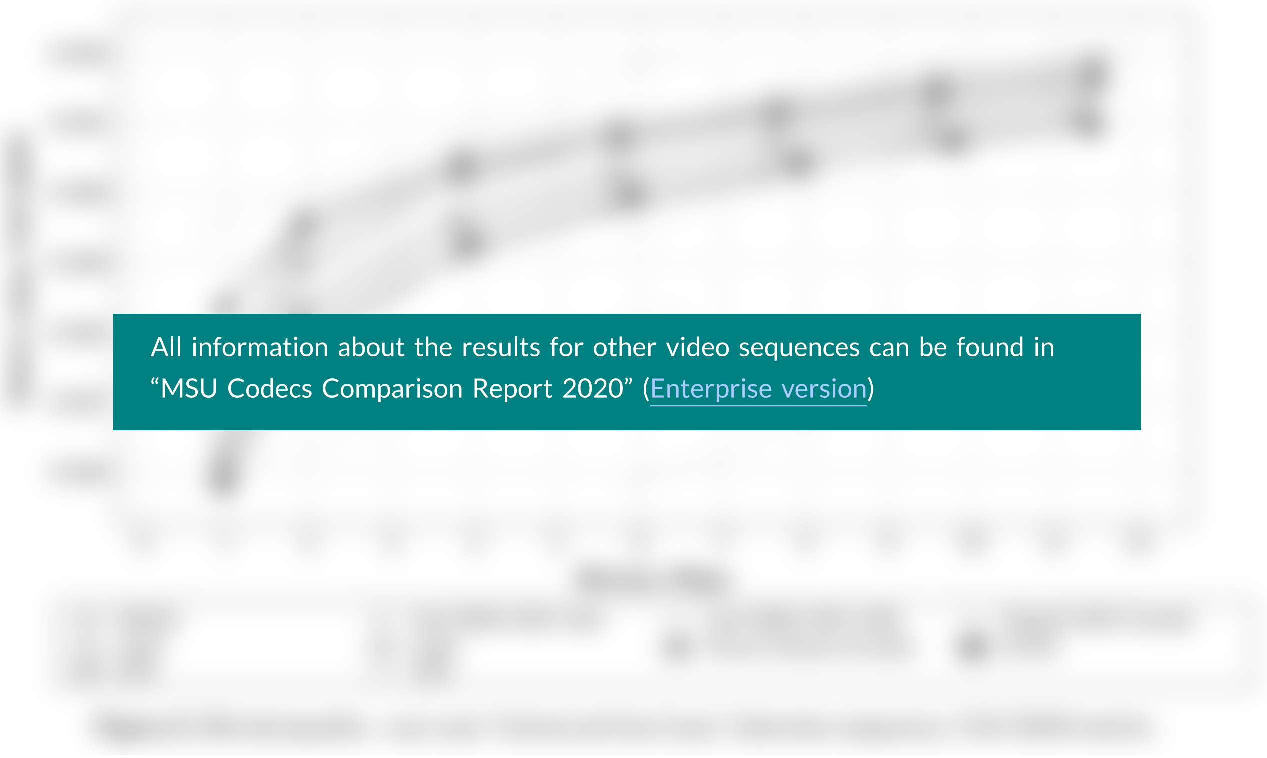


Figure 1: Bitrate/quality—use case "Offline (1 fps)," *crowd_run* sequence, YUV-Subjective metric.

The explanation of measuring on additional bitrates is presented in Section F.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 F), first place in the speed competition goes to **SIF Codec**, second place goes to **x264**, and third place to **QAV1**.

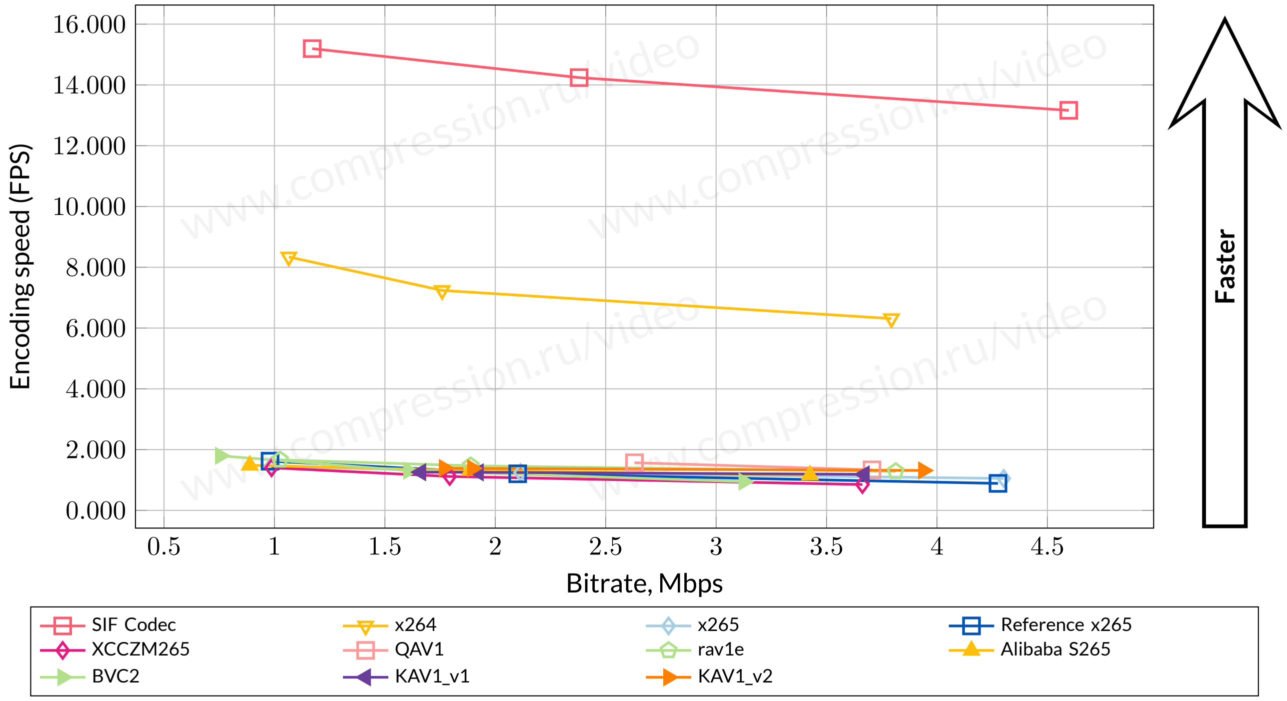
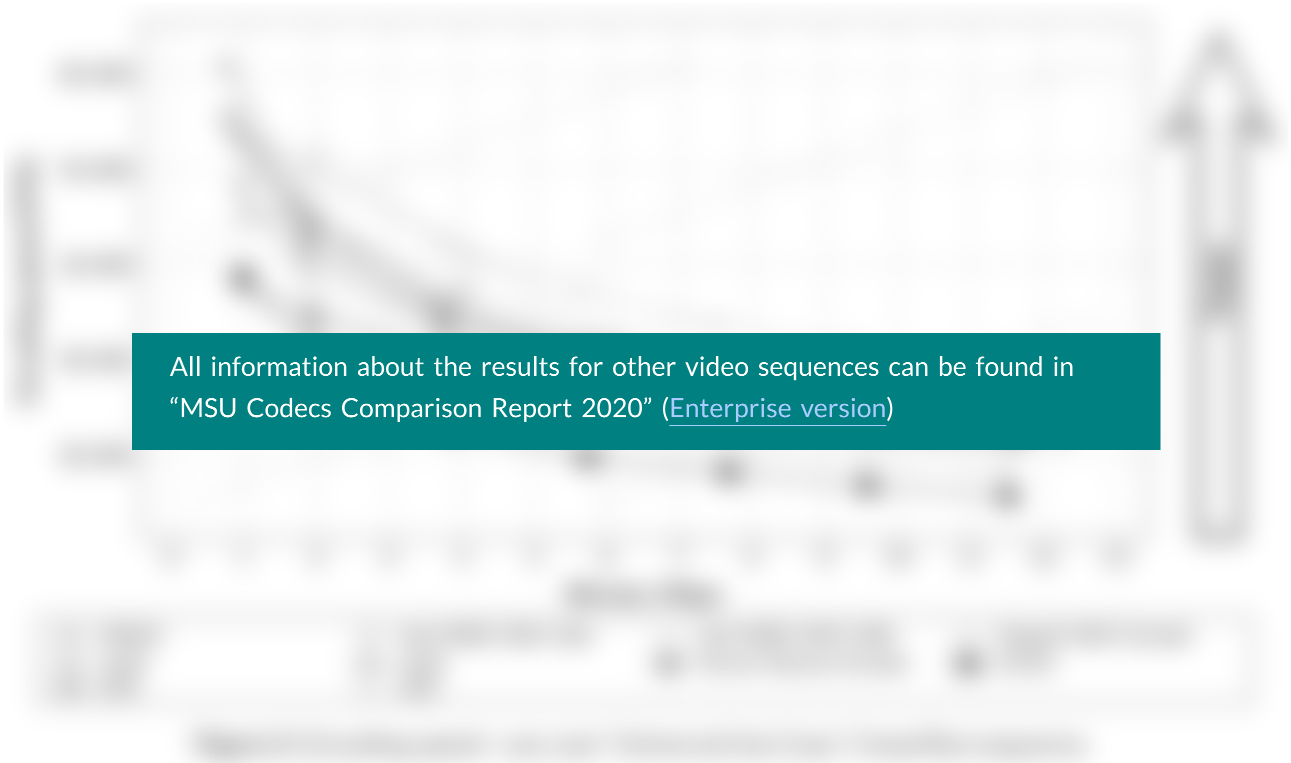


Figure 2: Encoding speed—use case "Offline (1 fps)," *crowd_run* sequence.

The explanation of measuring on additional bitrates is presented in Section F.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 F. 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 six Pareto-optimal encoders: **BVC2**, **Alibaba S265**, **KAV1_v2**, **QAV1**, **x264**, and **SIF Codec**.

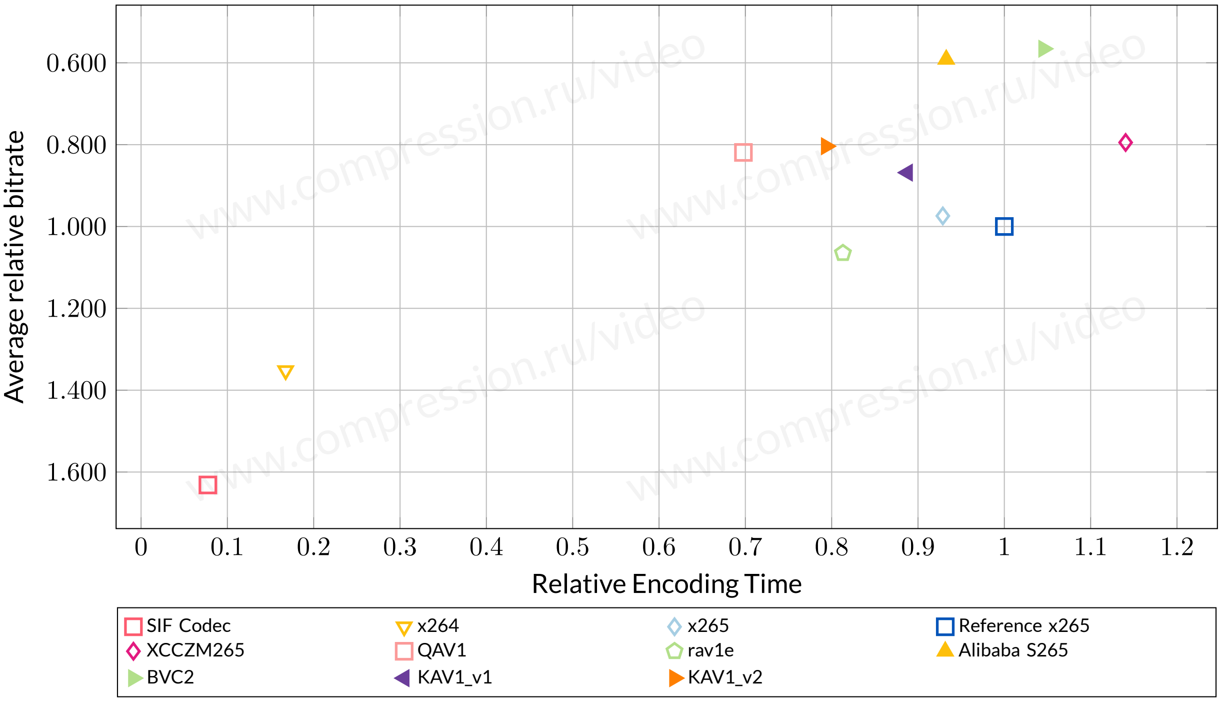
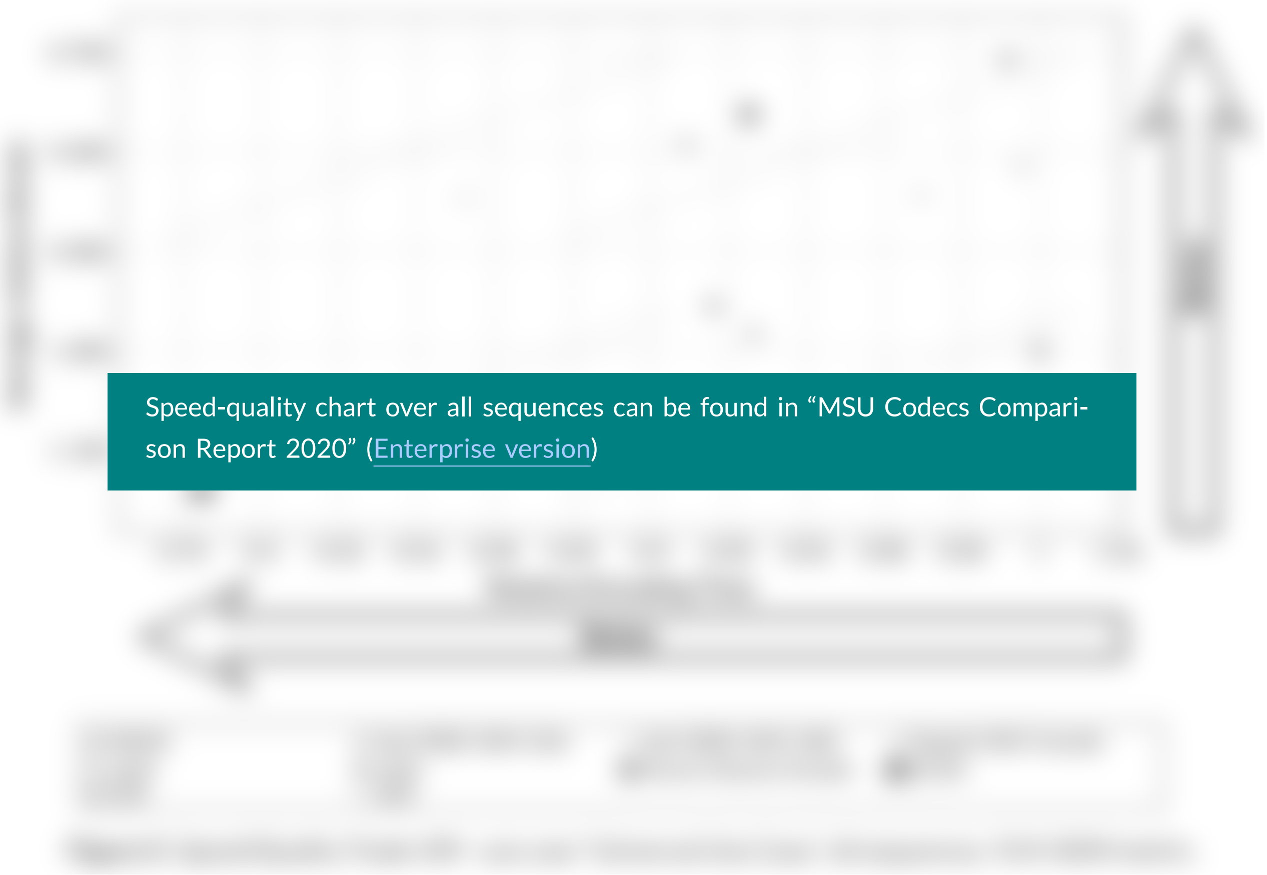


Figure 3: Speed/Quality Trade-Off—use case "Offline (1 fps)," *crowd_run* sequence, YUV-Subjective metric.



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

5.4. Offline (1 fps) YUV-Subjective

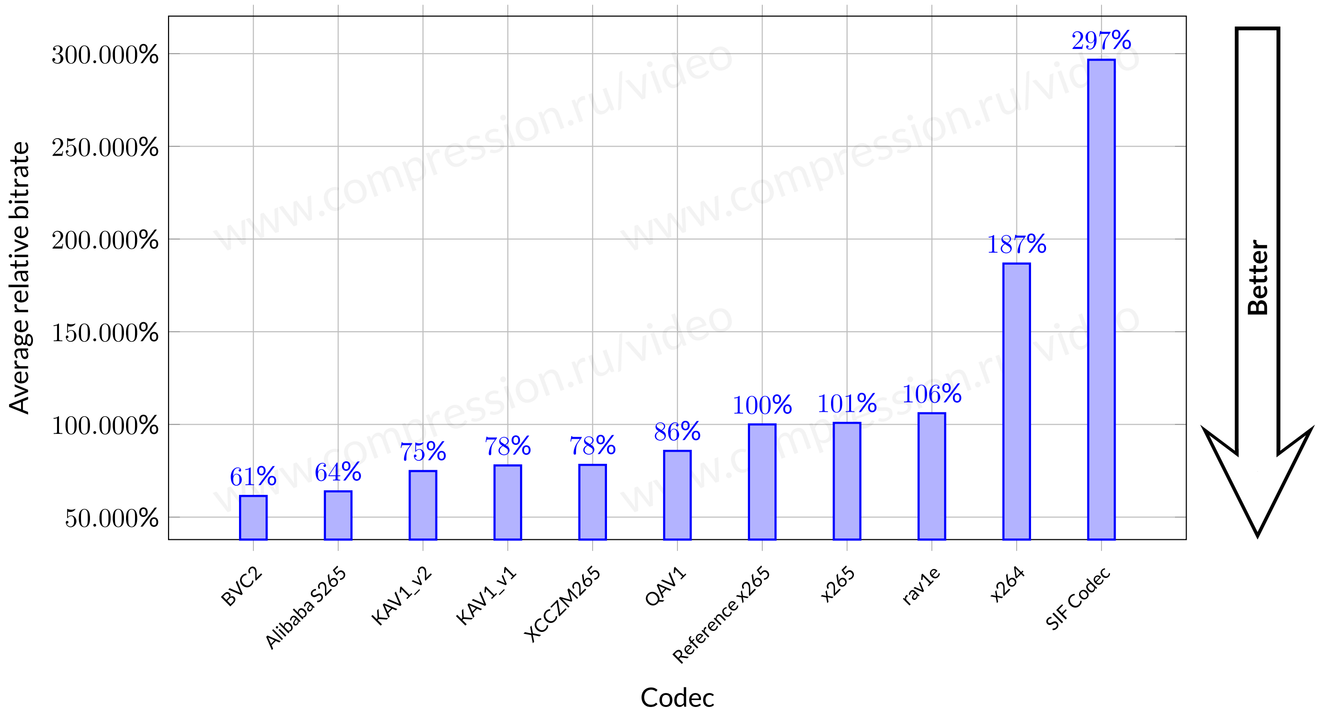


Figure 4: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-Subjective metric.

5.5. Offline (1 fps) YUV-SSIM

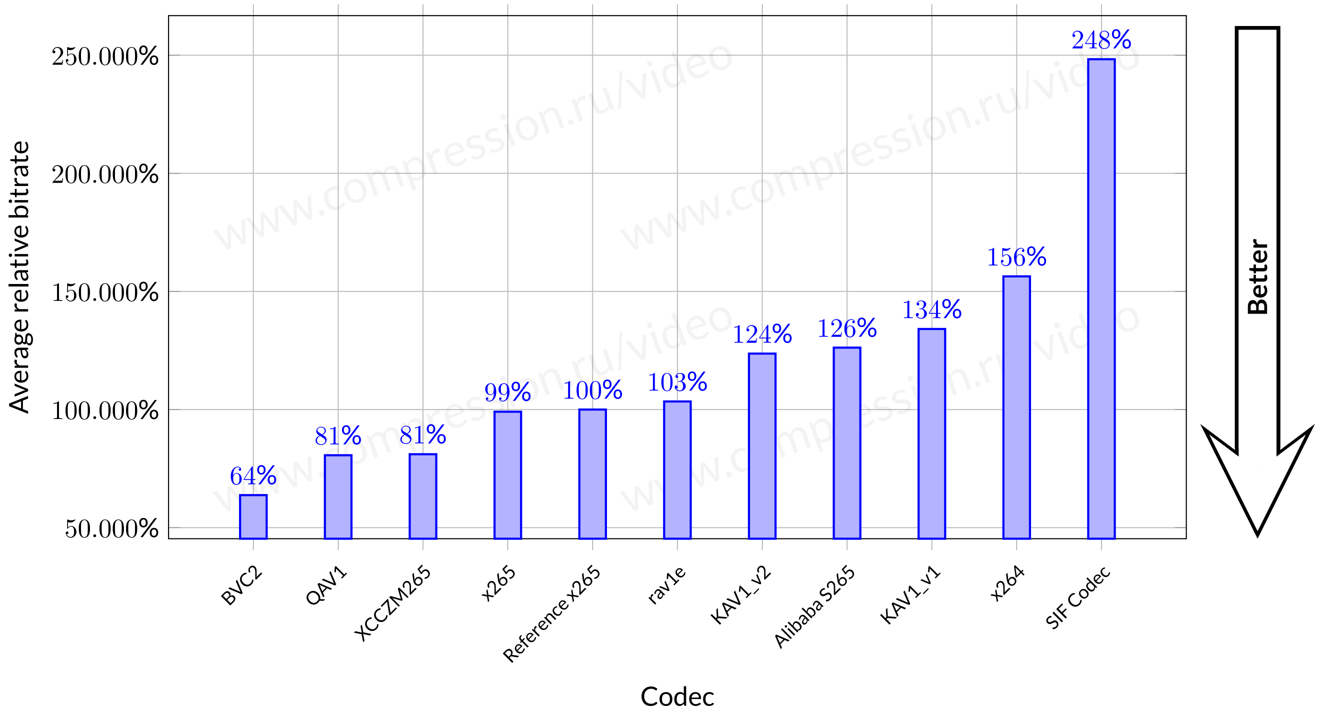


Figure 5: Average bitrate ratio for a fixed quality—use case “Offline (1 fps),” all sequences, YUV-SSIM metric.

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

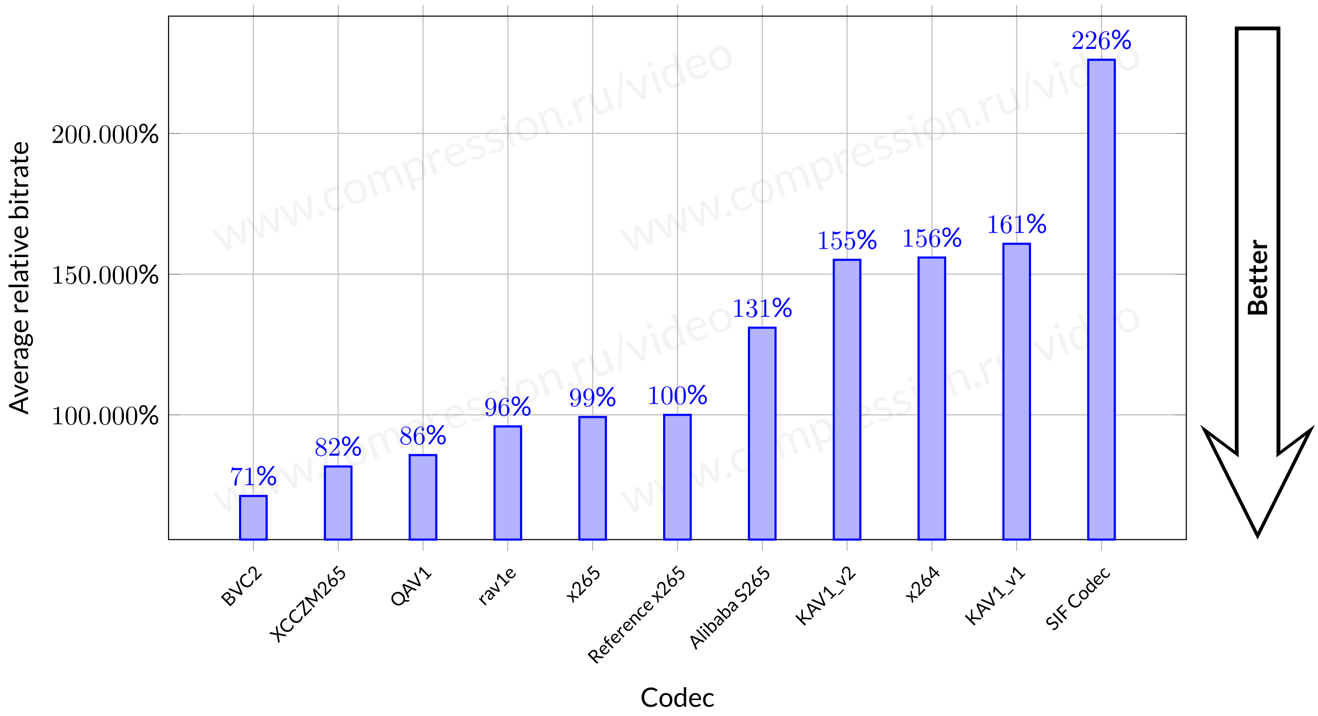


Figure 6: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-PSNR (avg. MSE) metric.

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

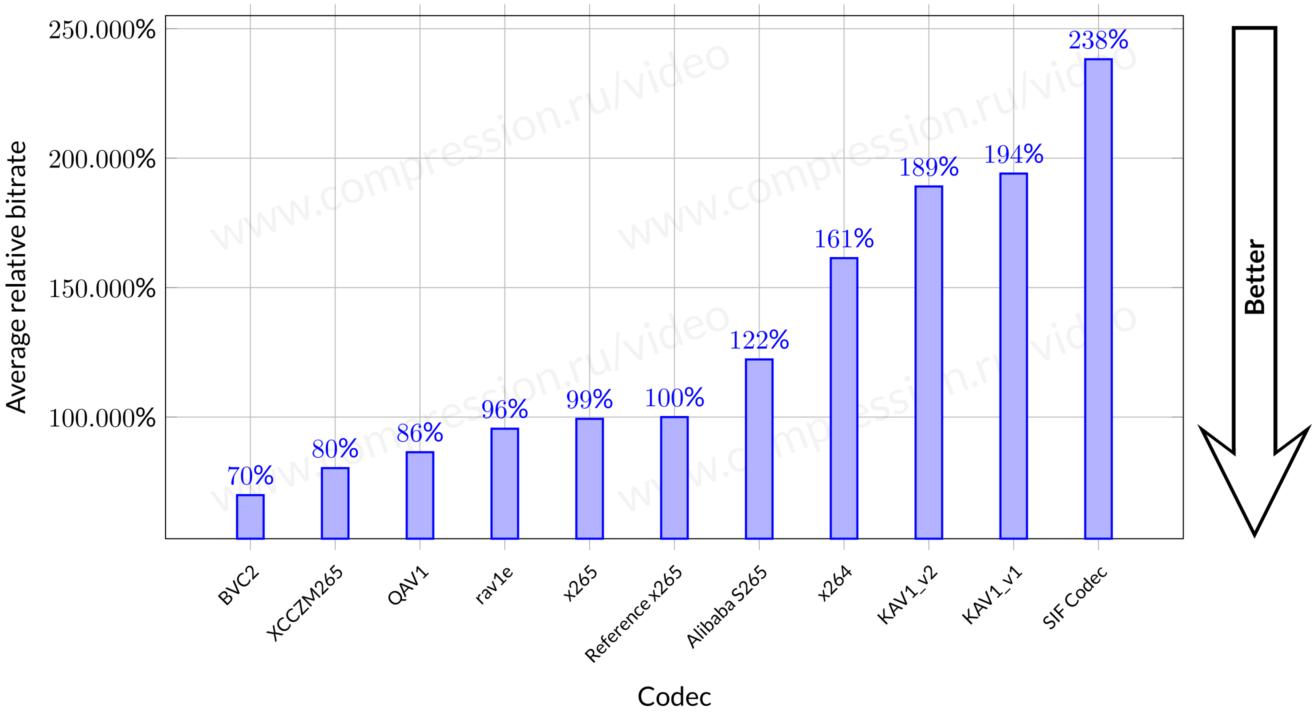


Figure 7: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, YUV-PSNR (avg. log) metric.

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

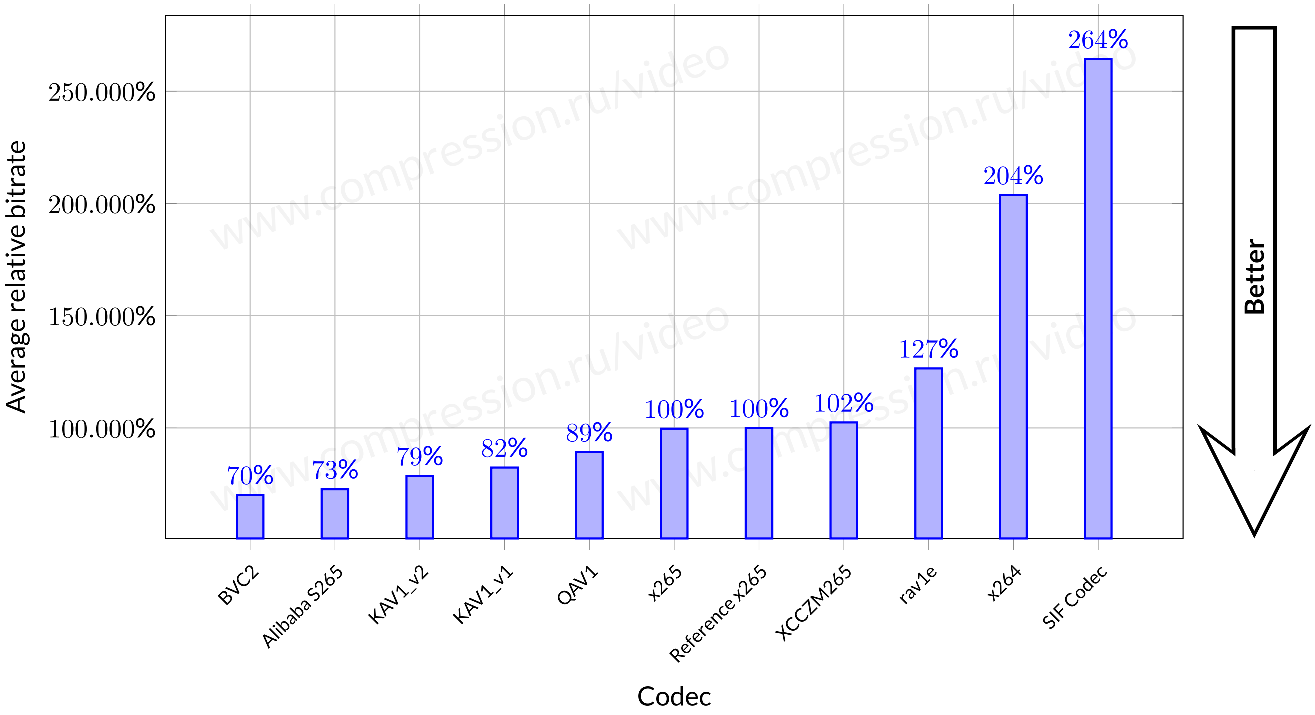
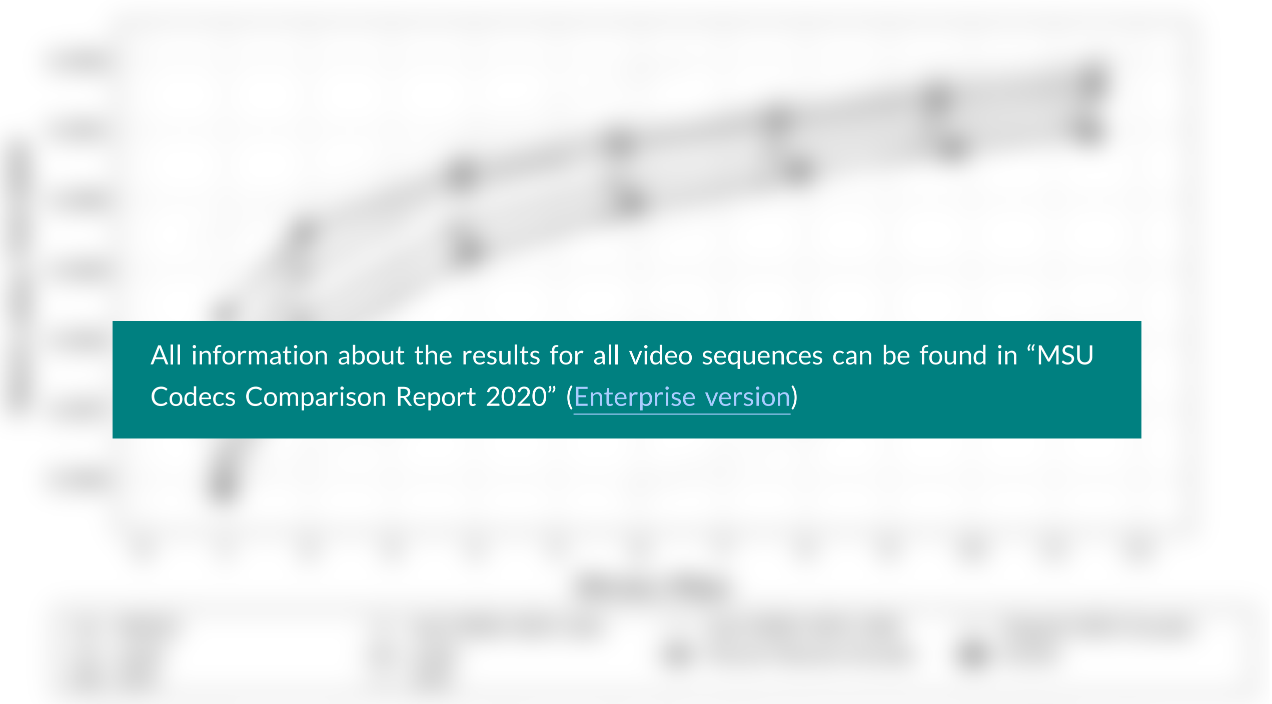


Figure 8: Average bitrate ratio for a fixed quality—use case "Offline (1 fps)," all sequences, Y-VMAF (v0.6.3) metric.

6. ONLINE (30 FPS)

6.1. RD Curves

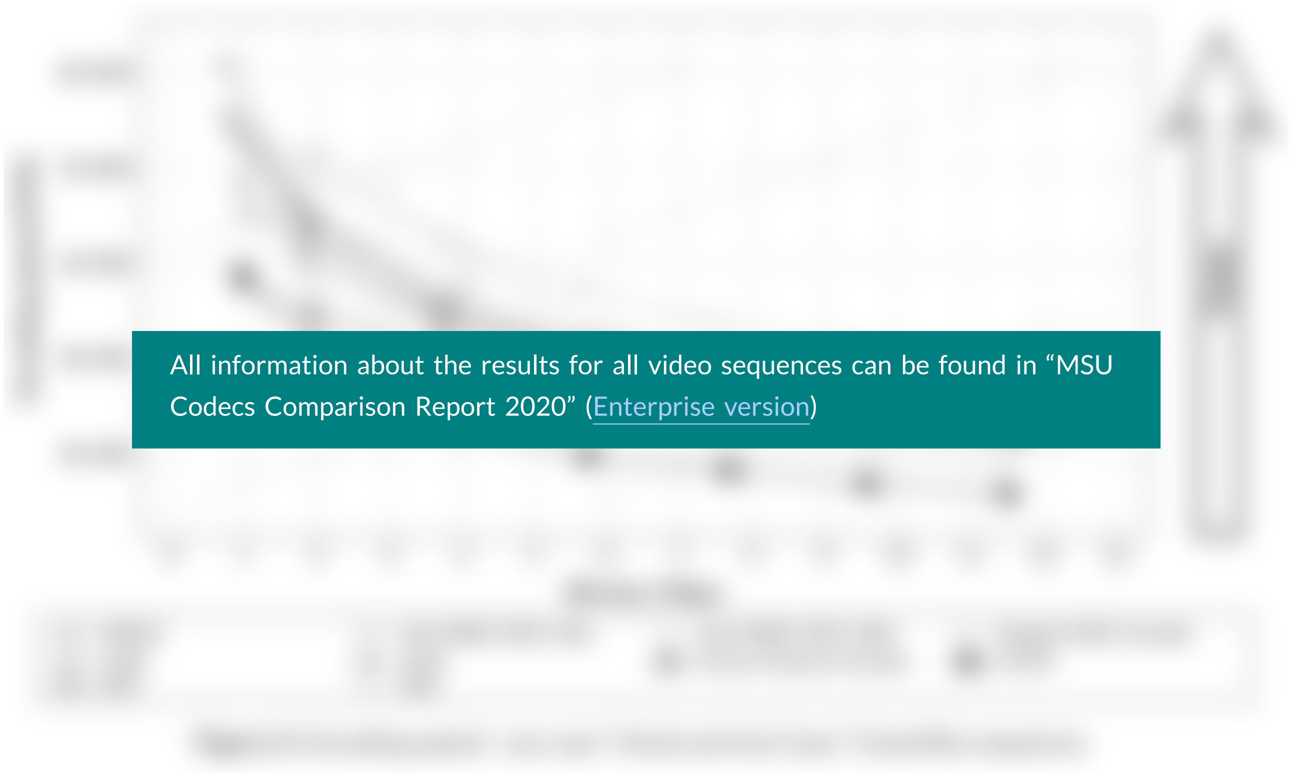
Judging from the mean quality scores (computed using the method described in Section F), first place in the quality competition goes to **Alibaba S265**, second place goes to **XCCZM265**, and third place to **x265** and **Reference x265**.



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 F), first place in the speed competition goes to **SIF Codec**, **x264**, **x265**, and **Reference x265**, second place goes to **XCCZM265**, and third place to **Alibaba S265**.



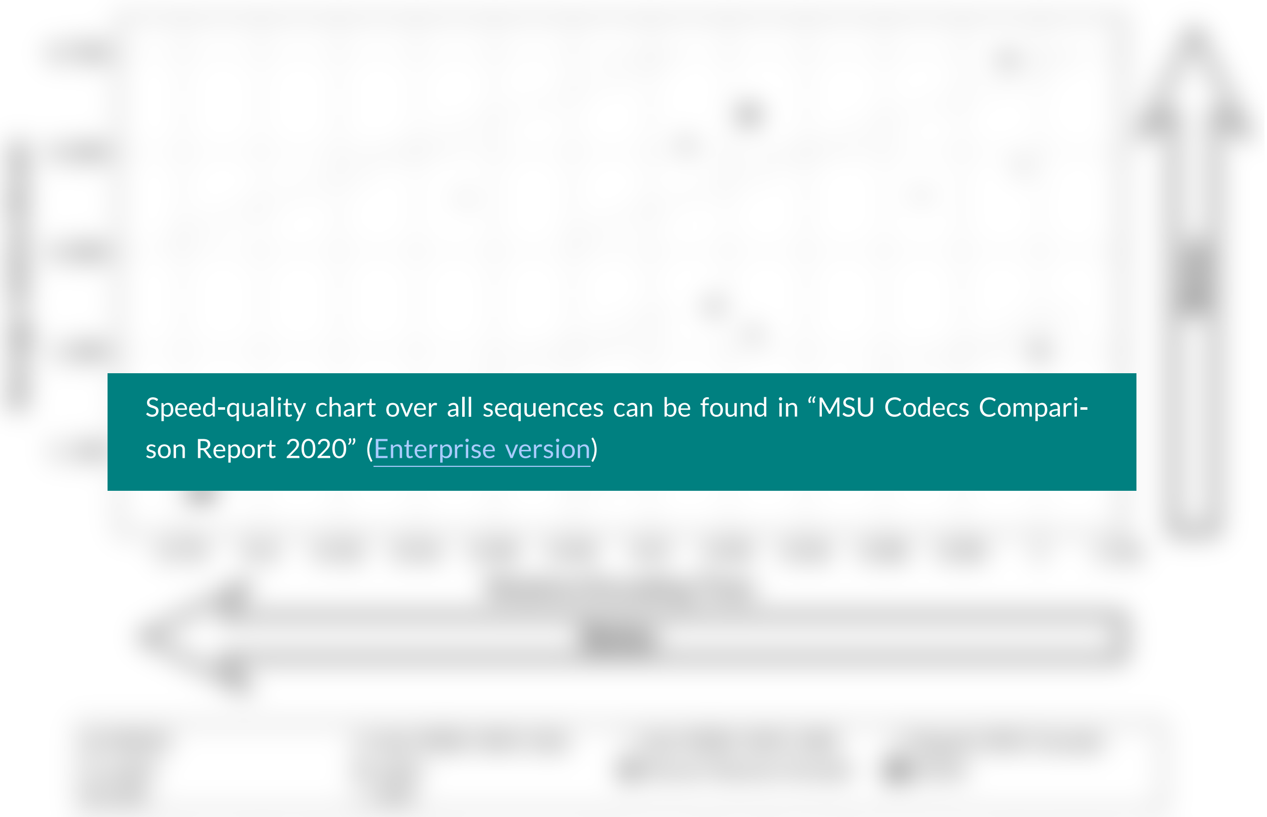
All information about the results for all video sequences can be found in “MSU Codecs Comparison Report 2020” (Enterprise version)

6.3. Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs are in Appendix F. 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 five Pareto-optimal encoders: **Alibaba S265**, **XCCZM265**, **x265**, **x264**, and **SIF Codec**.



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

[illegible]

6.4. Online (30 fps) YUV-Subjective

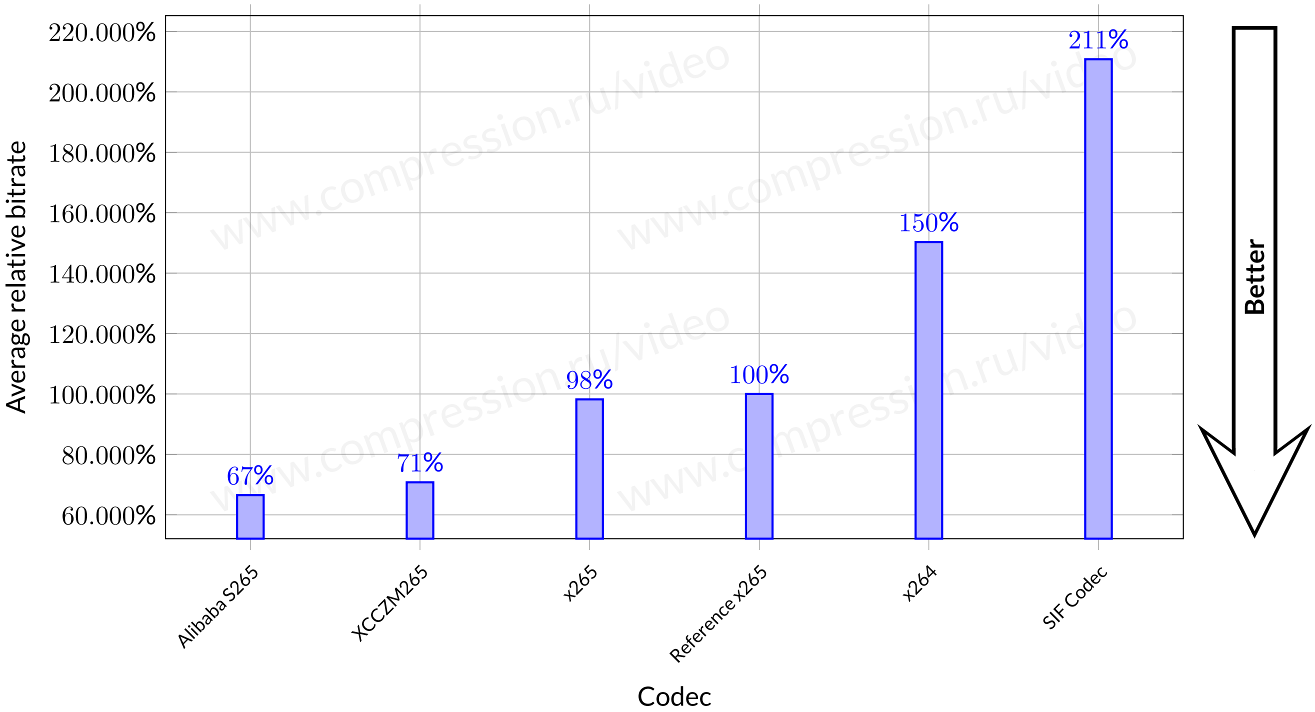


Figure 9: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-Subjective metric.

6.5. Online (30 fps) YUV-SSIM

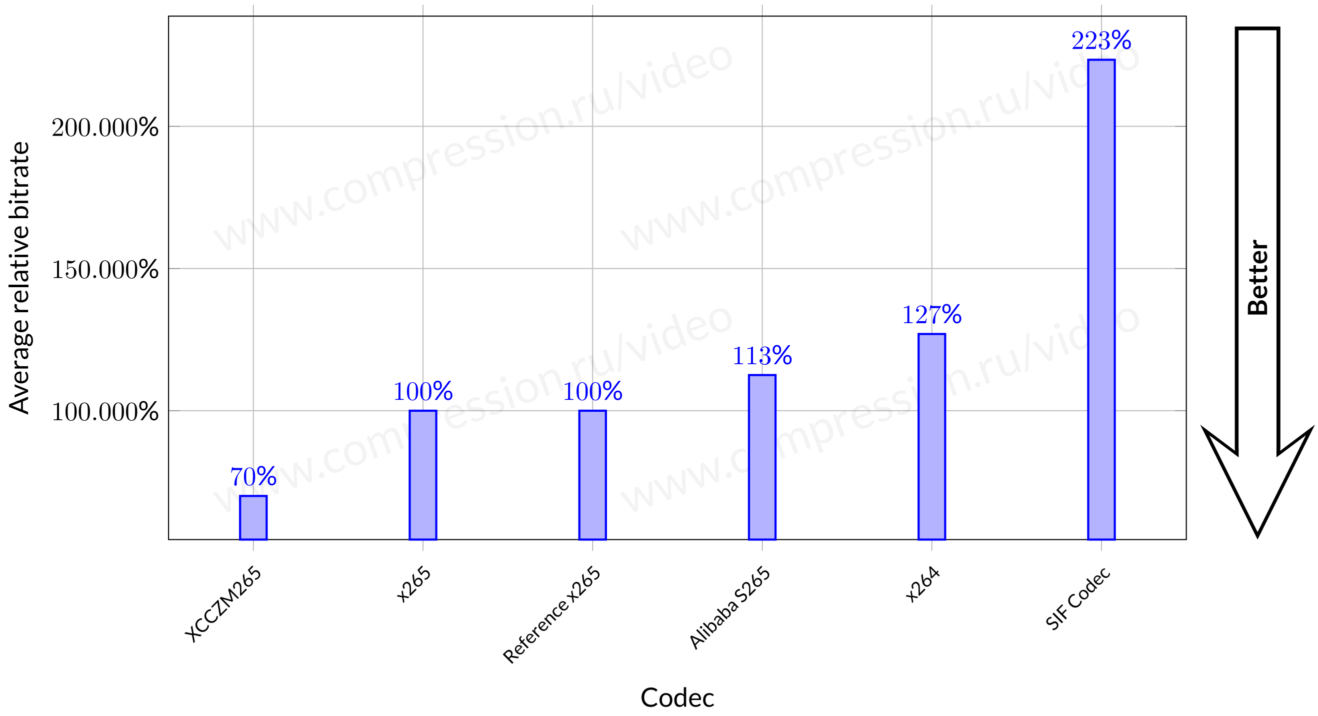


Figure 10: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-SSIM metric.

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

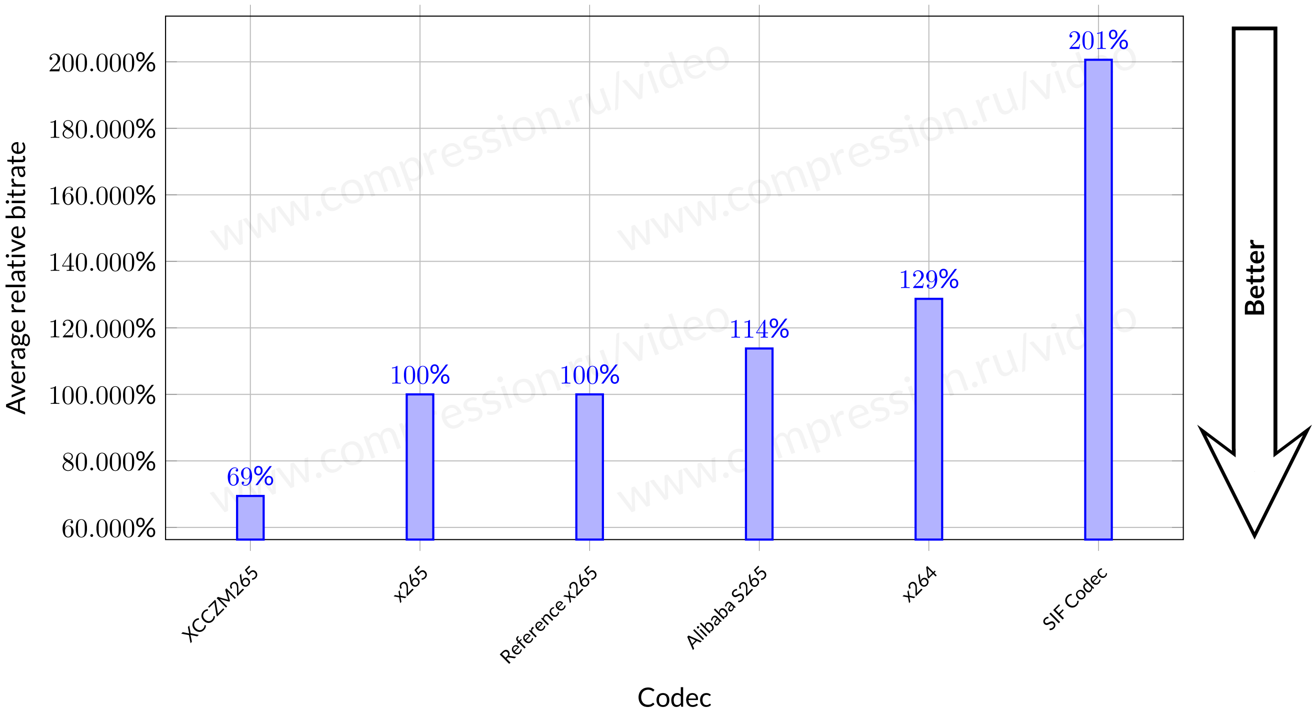


Figure 11: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-PSNR (avg. MSE) metric.

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

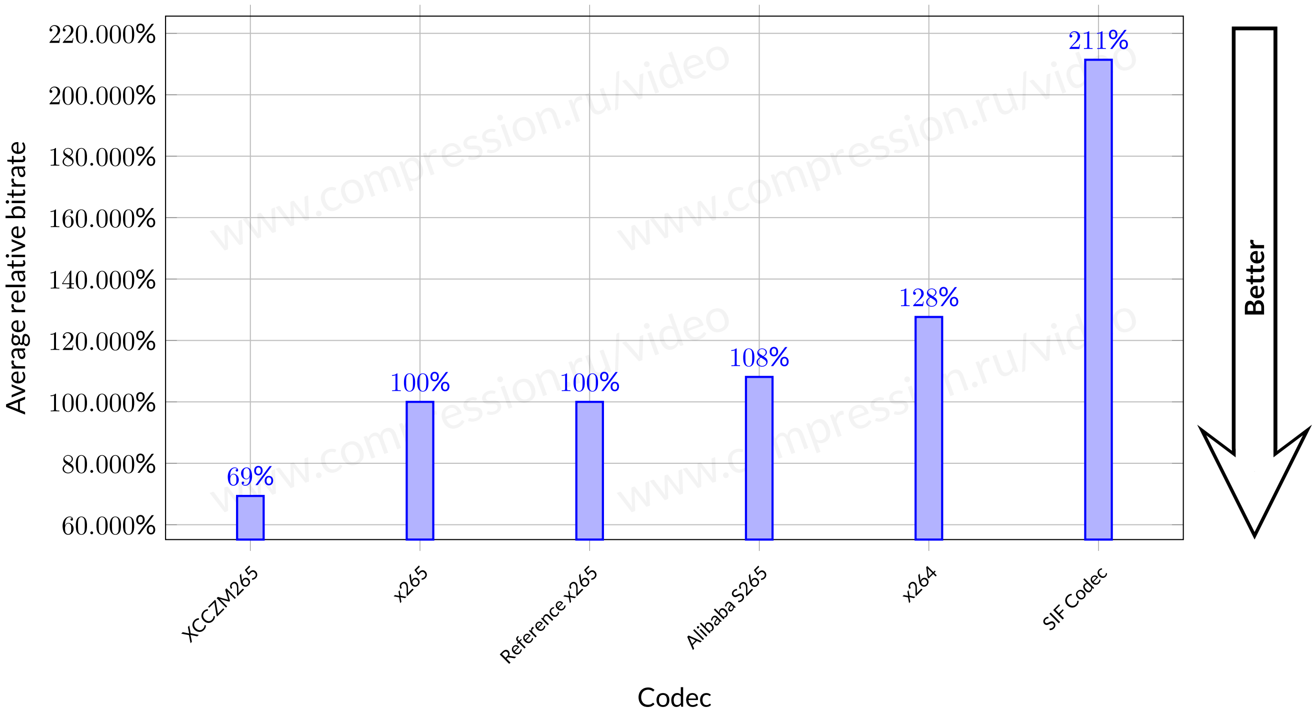


Figure 12: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, YUV-PSNR (avg. log) metric.

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

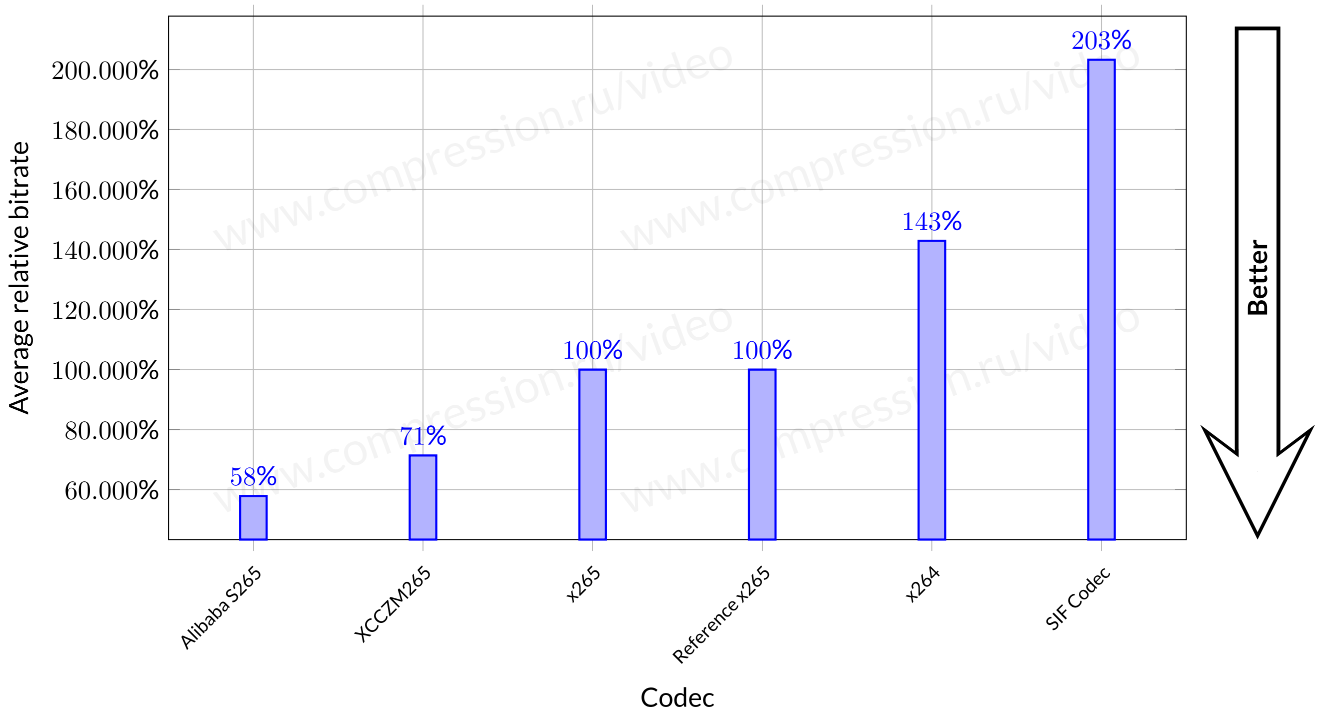


Figure 13: Average bitrate ratio for a fixed quality—use case "Online (30 fps)," all sequences, Y-VMAF (v0.6.3) metric.

7. CONCLUSION

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

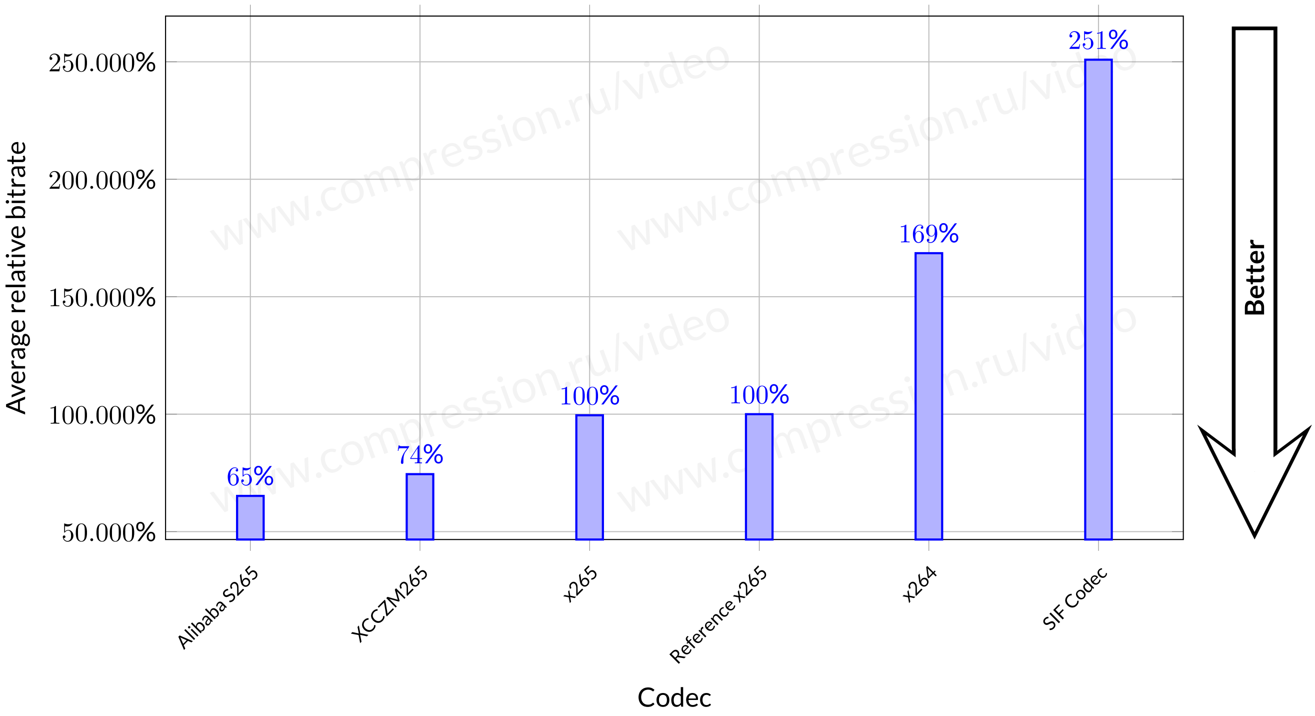


Figure 14: Average bitrate ratio for a fixed quality—all sequences, YUV-Subjective metric.

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

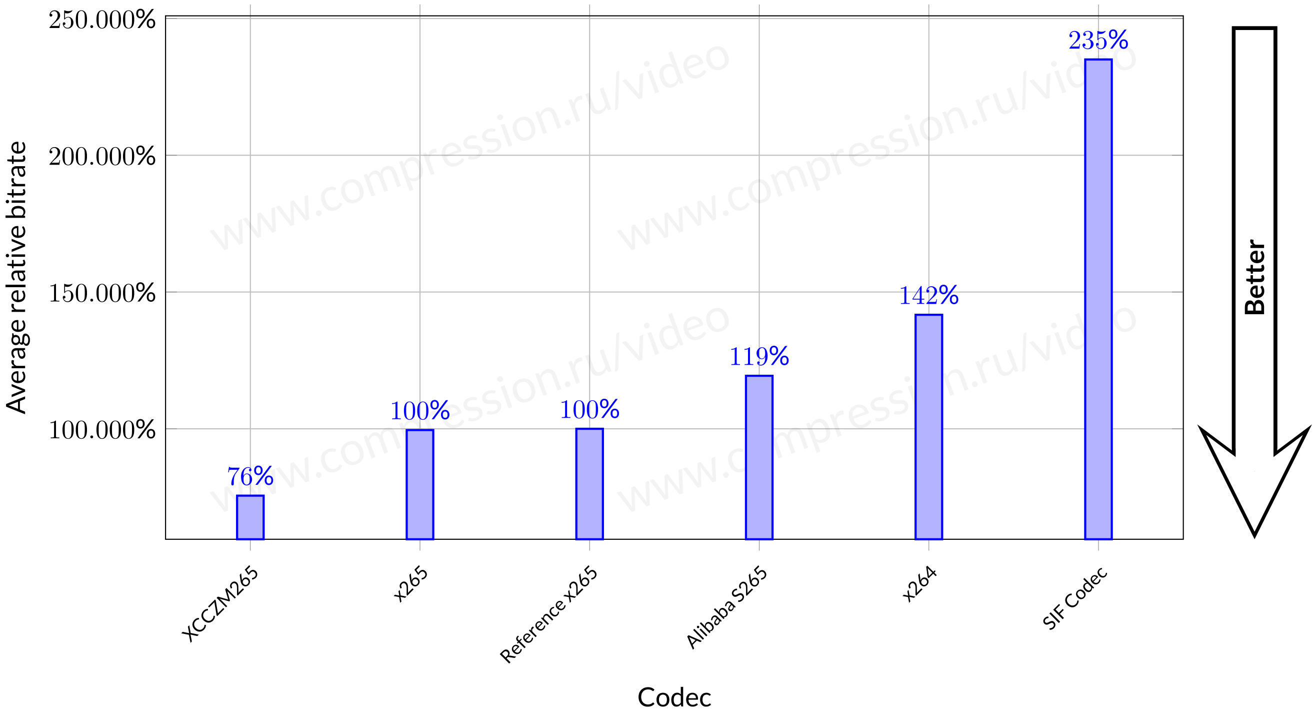


Figure 15: Average bitrate ratio for a fixed quality—all sequences, YUV-SSIM metric.

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

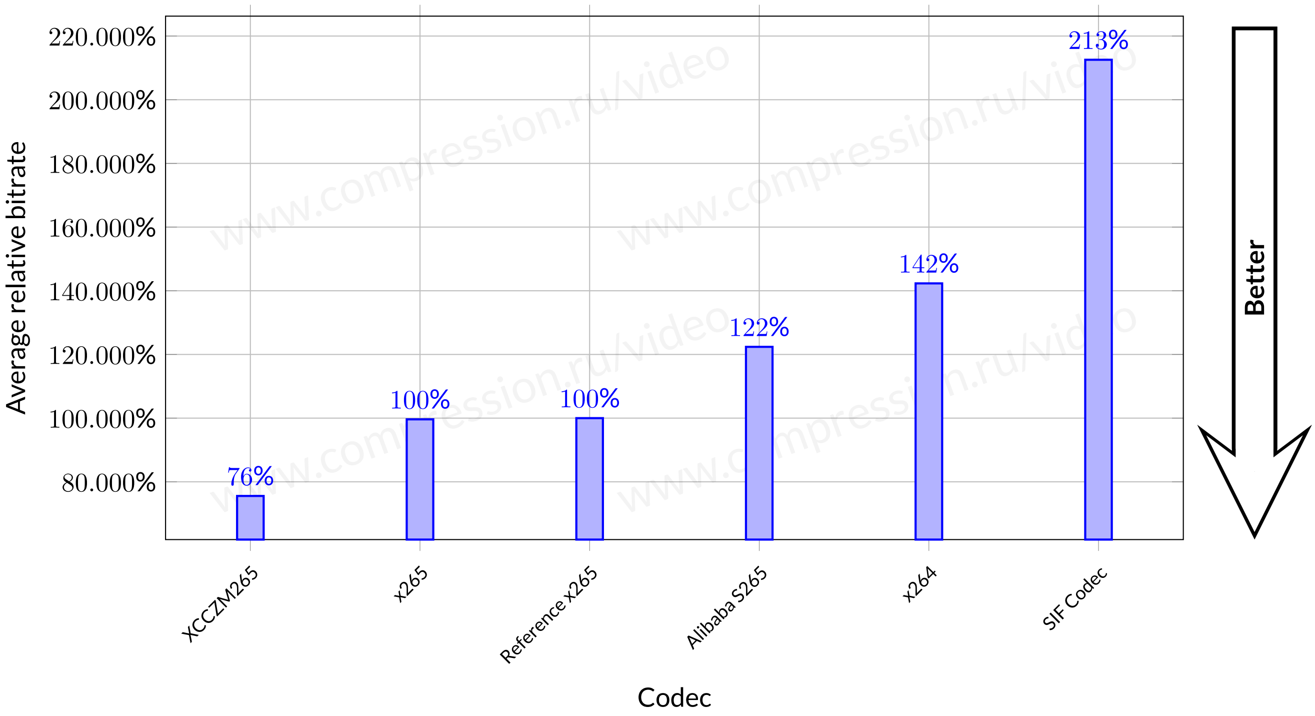


Figure 16: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. MSE) metric.

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

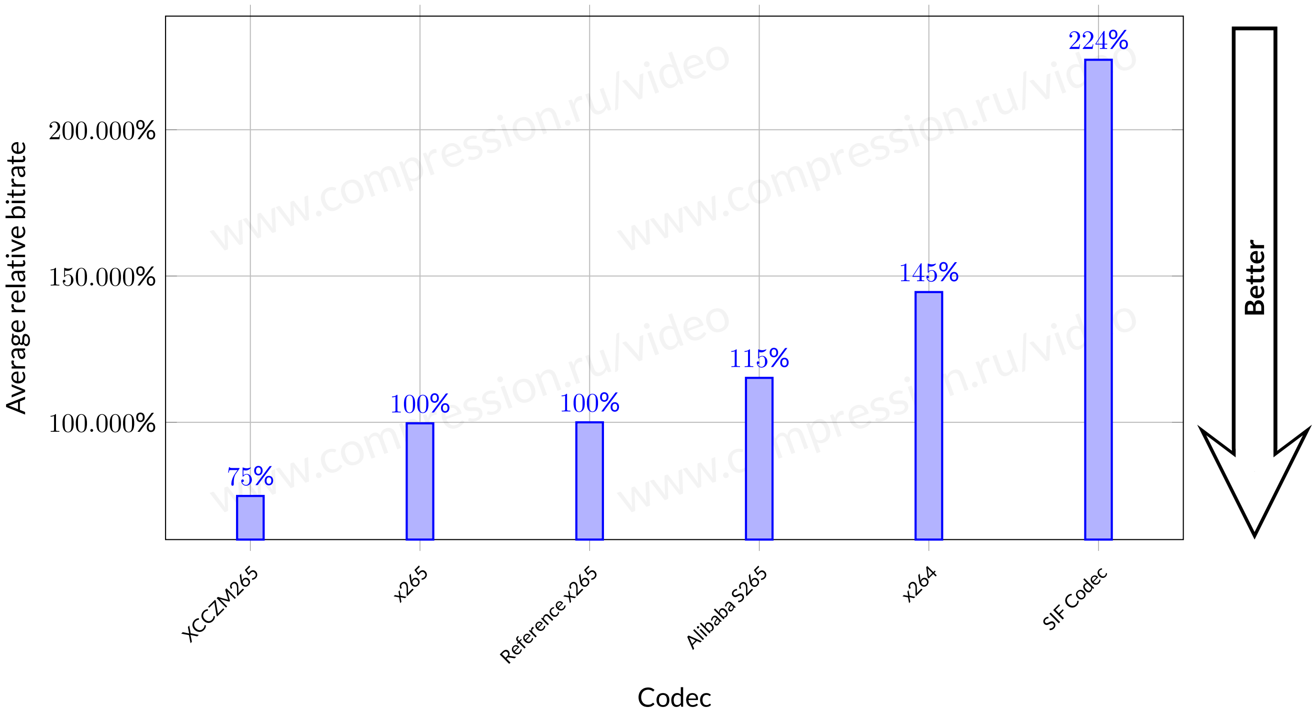


Figure 17: Average bitrate ratio for a fixed quality—all sequences, YUV-PSNR (avg. log) metric.

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

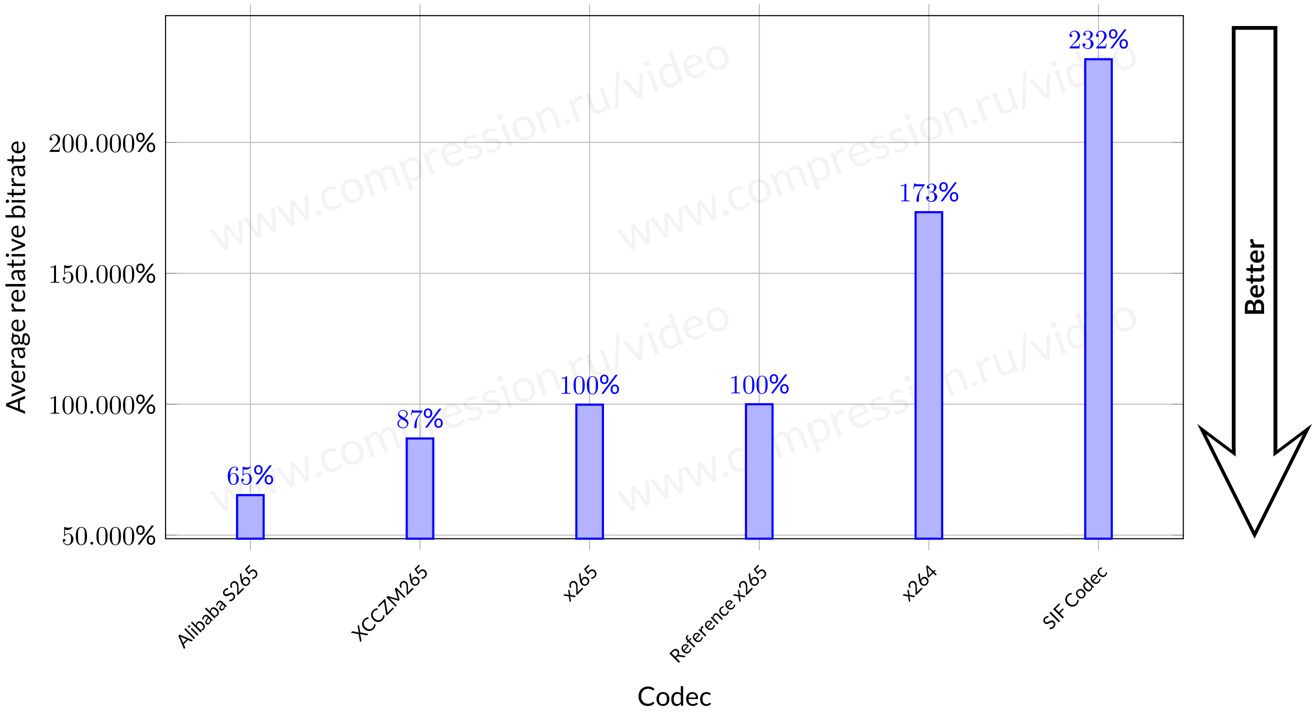


Figure 18: Average bitrate ratio for a fixed quality—all sequences, Y-VMAF (v0.6.3) metric.

A. PARTICIPANTS' COMMENTS

A.1. Alibaba S265

We think subjective quality is different from objective metrics, Alibaba S265 imported JND model to optimize QP from CTU level and some video pre-process technology to improve Subjective quality.

A.2. rav1e

There is a direct focus on color quality in rav1e that is reflected in chroma-only objective results and in subjective results. There is also a preference for preservation of macroblock variance which may contribute to performance in subjective tests. Compared to the reference, a similar subjective result is achieved at a substantially higher speed.

B. LIST OF CHANGES

- **January, 21, 2022** Added description of Subjectify.us service in the Introduction.

C. SEQUENCES

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

C.1. blue_hair

Sequence title	blue_hair
Resolution	1920×1080
Number of frames	300
Color space	YV12
Frames per second	30
Source resolution	FullHD

Shaky handheld vlog video. ID: Vlog_1080P-35cd

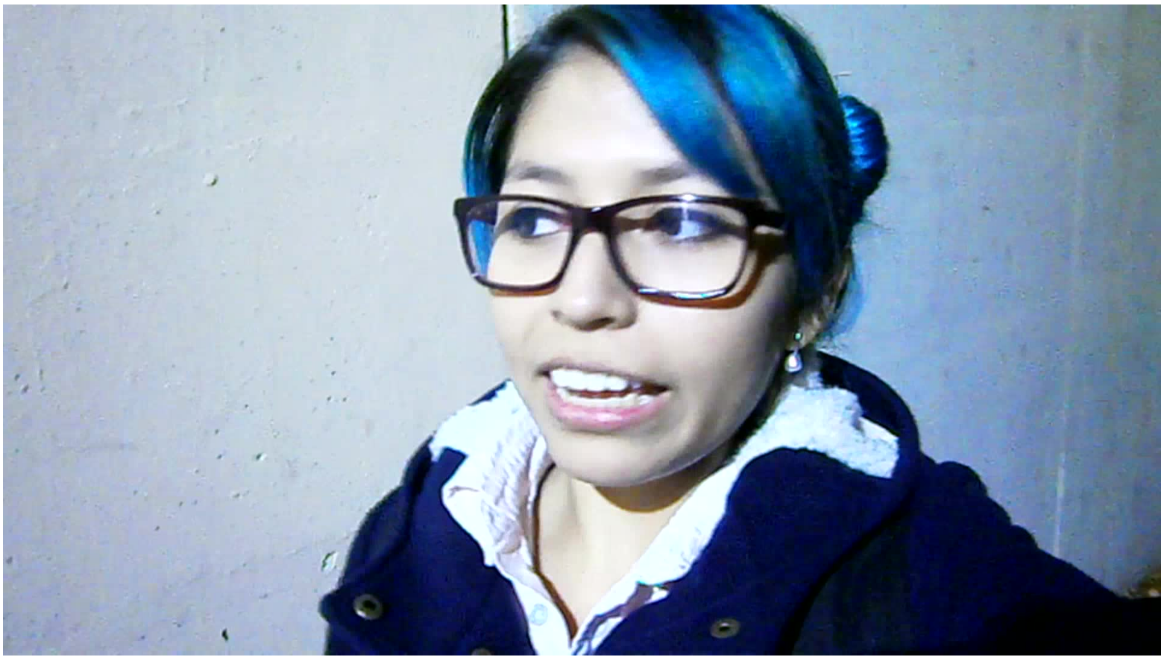


Figure 19: blue_hair sequence, frame 235

C.2. cover_song

Sequence title	cover_song
Resolution	1920×1080
Number of frames	300
Color space	YV12
Frames per second	30
Source resolution	FullHD

Handheld static camera and scene changes. ID: CoverSong_1080p-5430

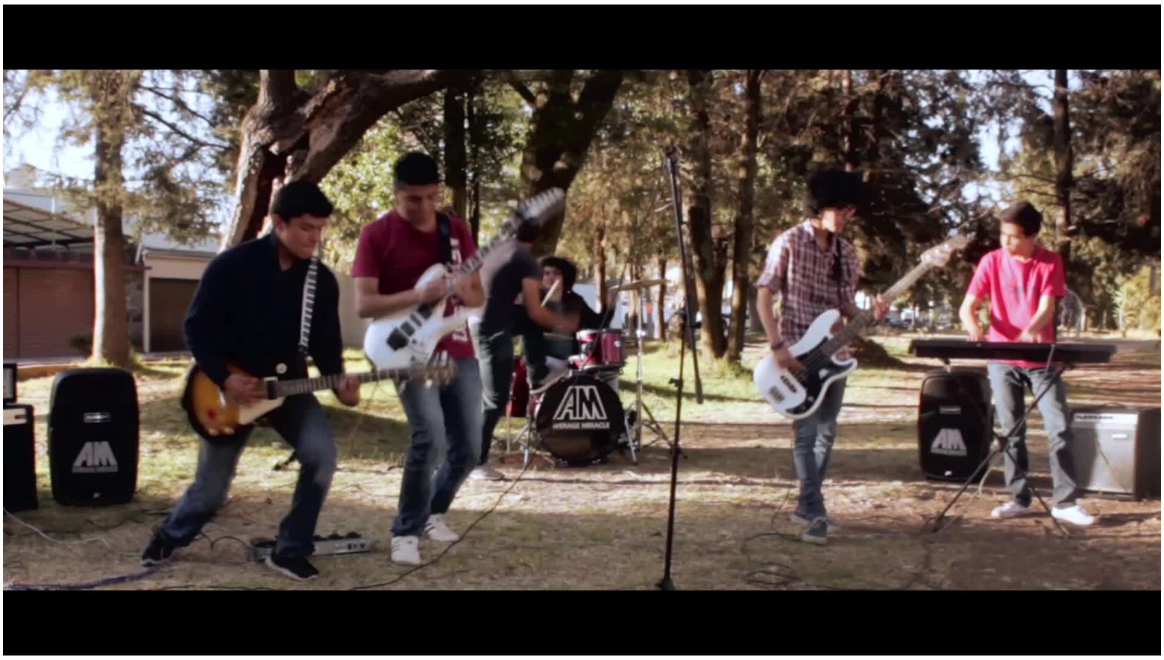


Figure 20: cover_song sequence, frame 29

C.3. crowd_run

Sequence title	crowd_run
Resolution	1920×1080
Number of frames	500
Color space	YV12
Frames per second	50
Source resolution	FullHD
Bitrate	1244.16

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

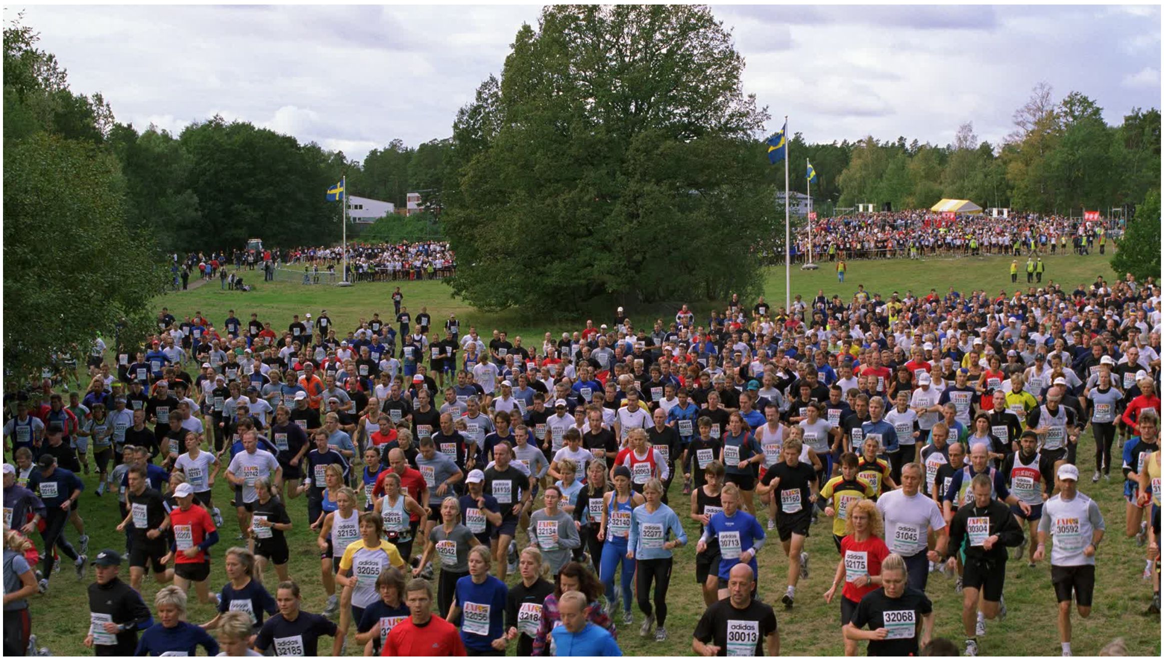


Figure 21: crowd_run sequence, frame 179

C.4. forest_dog

Sequence title	forest_dog
Resolution	1920×1080
Number of frames	250
Color space	YV12
Frames per second	25
Source resolution	FullHD

Macro shot of flora. Black dog running.

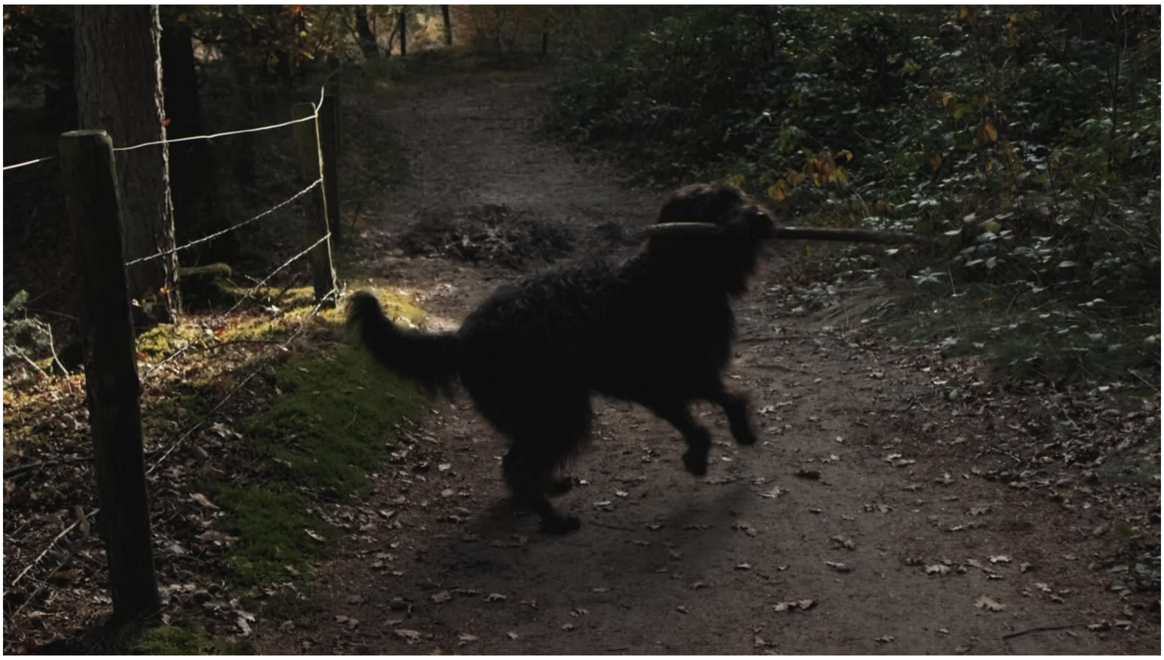


Figure 22: forest_dog sequence, frame 44

C.5. gaming

Sequence title	gaming
Resolution	1920×1080
Number of frames	300
Color space	YV12
Frames per second	30
Source resolution	FullHD

The Forest gameplay. ID: Gaming_1080P-72c8

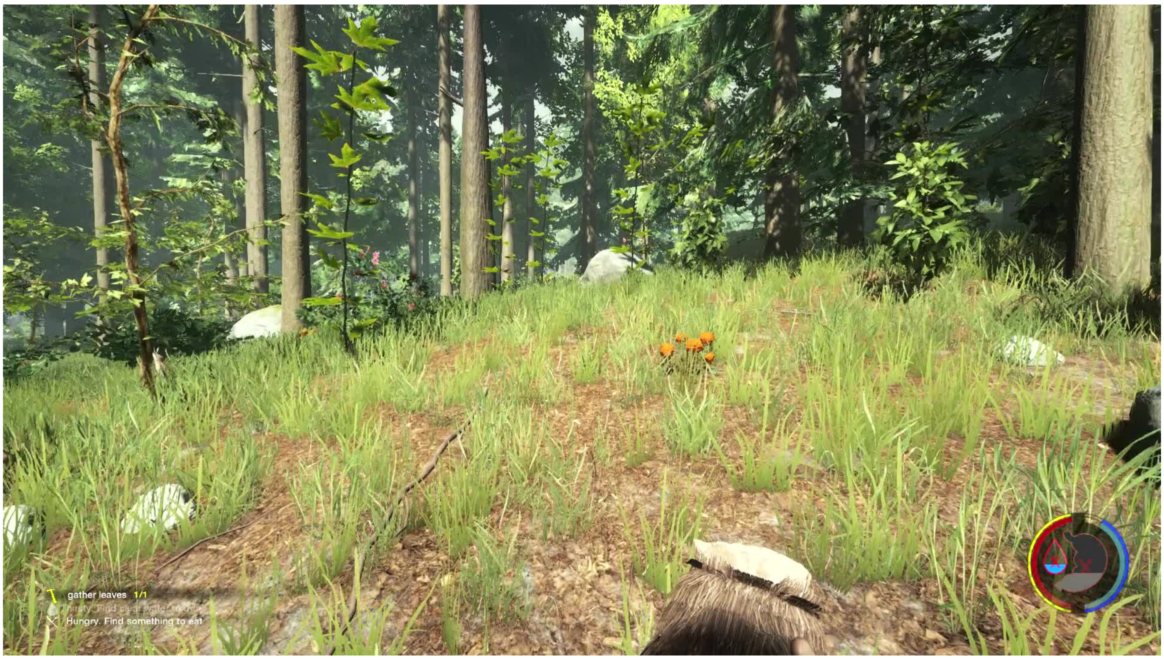

gather leaves 1/1
Thirsty. Find clear water to drink
Hungry. Find something to eat

Figure 23: gaming sequence, frame 40

C.6. okeechobee

Sequence title	okeechobee
Resolution	1920×1080
Number of frames	250
Color space	YV12
Frames per second	25
Source resolution	FullHD

Concert record shot from different points.

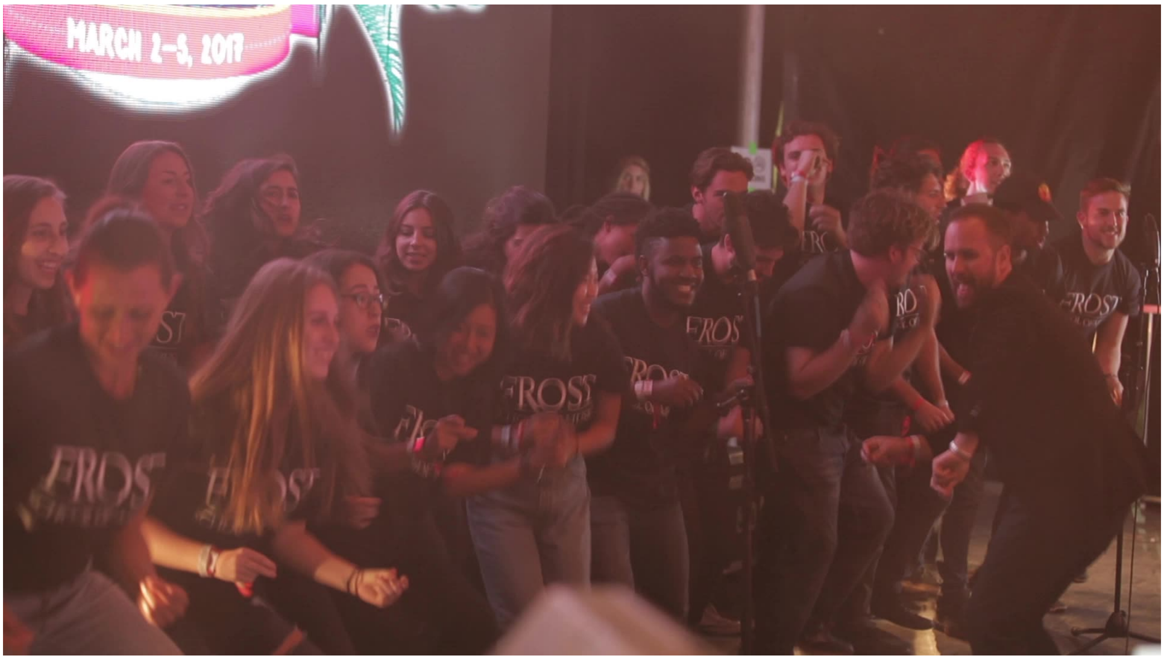


Figure 24: okeechobee sequence, frame 195

C.7. pyranha_rafting

Sequence title	pyranha_rafting
Resolution	1920×1080
Number of frames	240
Color space	YV12
Frames per second	24
Source resolution	FullHD

Panning shots of whitewater rafting.

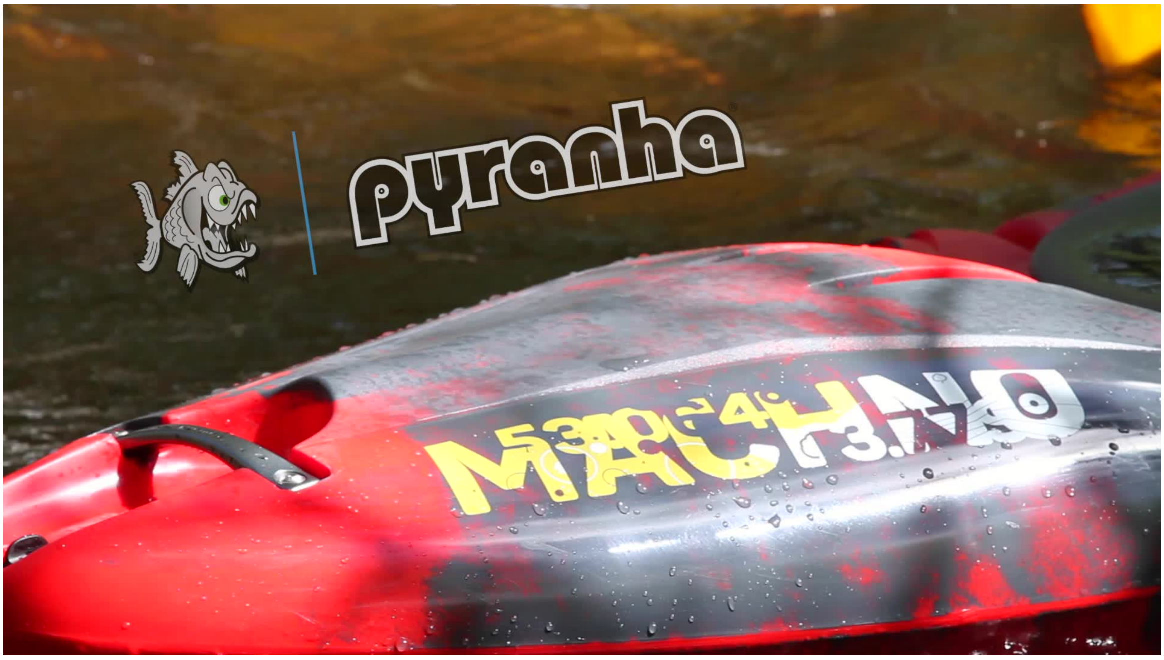


Figure 25: pyranha_rafting sequence, frame 54

C.8. road_timelapse

Sequence title	road_timelapse
Resolution	1920×1080
Number of frames	300
Color space	YV12
Frames per second	30
Source resolution	FullHD

Sped up footage from a dash cam. Objects reflecting in the windshield.

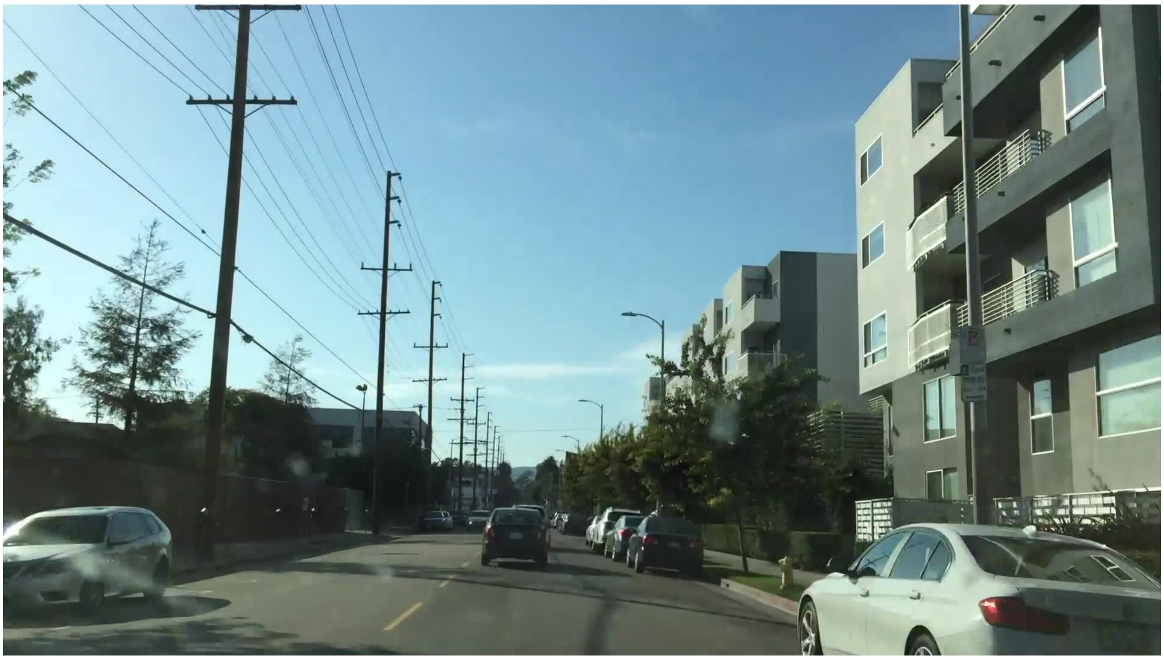


Figure 26: road_timelapse sequence, frame 141

D. CODECS

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

E. 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 Table 3 shows the number of videos in our video collection.

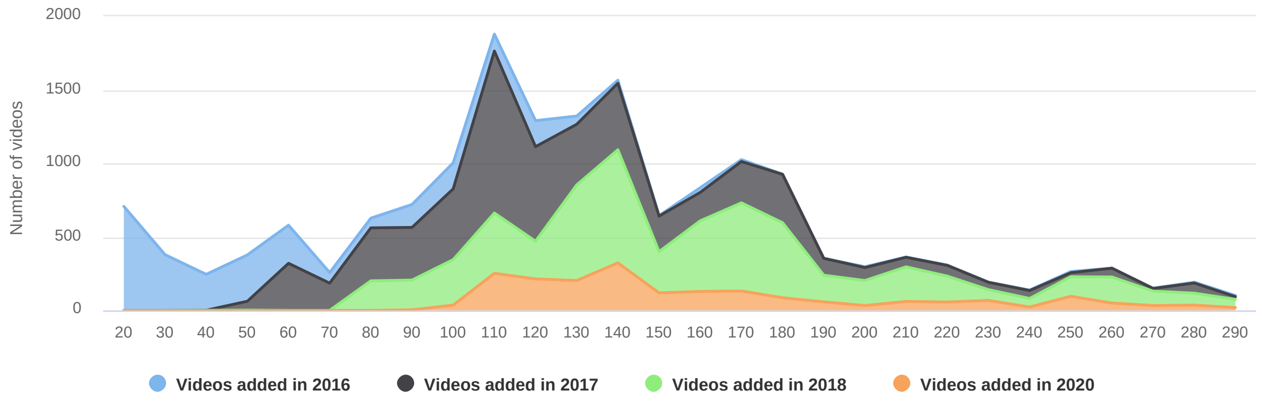


Figure 27: Bit rate distributions for comparison video set.

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.

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 28.

---

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

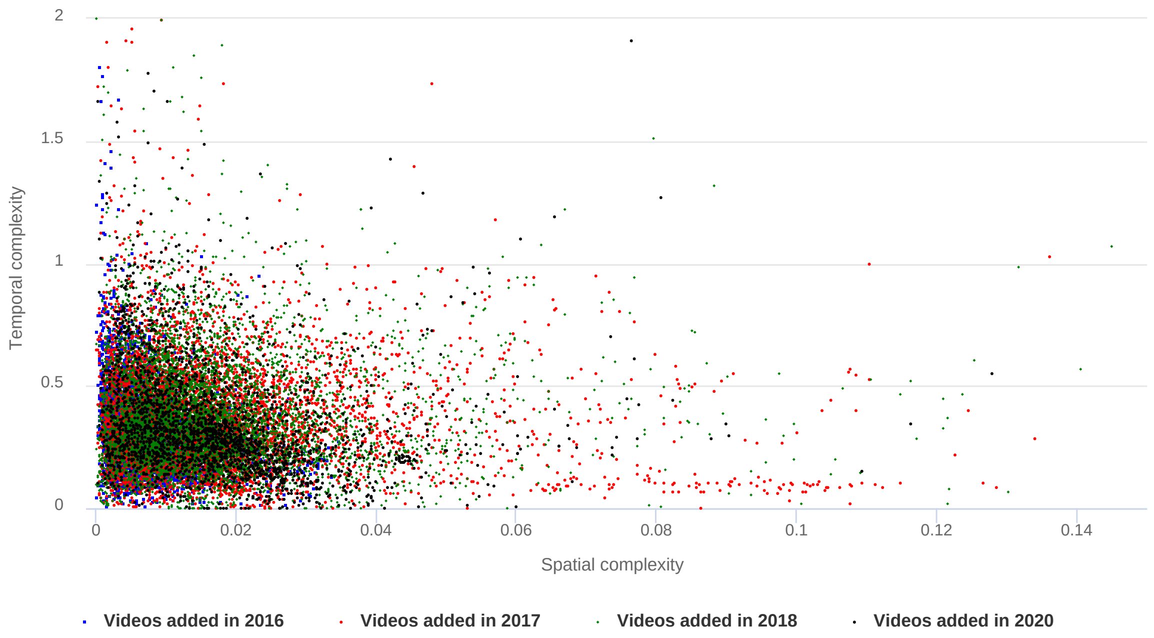


Figure 28: Distribution of obtained samples.

This year, we conducted a voting to choose final set of 50 videos for Main 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.

For subjective report we have selected 5 videos from Main report and added 3 videos from YouTube UGC Dataset.

F. 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.

F.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.

F.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.

F.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.

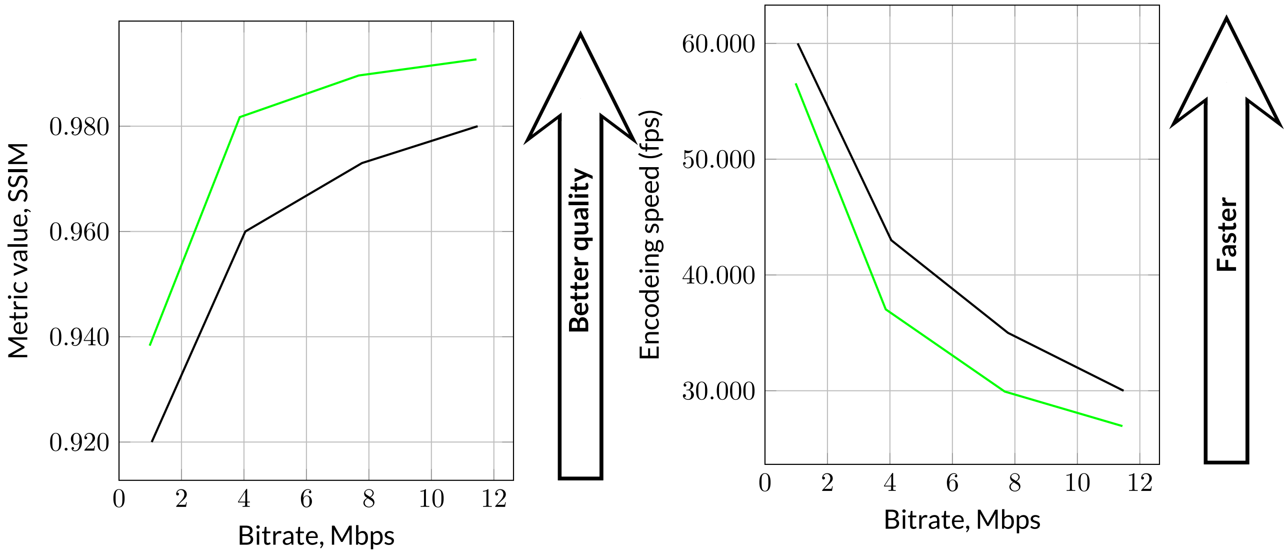
Owing to these advantages, we frequently use relative-bitrate/relative-time graphs in this report because they assist in evaluating the codecs in the test set, especially when the number of codecs is large.

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

F.4. Bitrate Ratio for the Same Quality

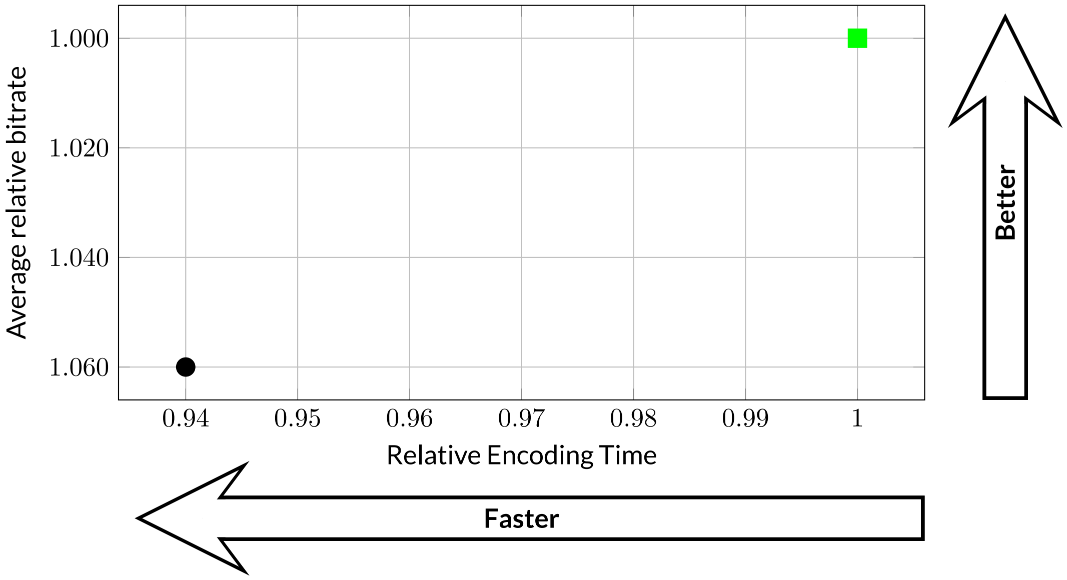
The first step in computing the average bitrate ratio for a fixed quality is to invert the axes of the bitrate/quality graph (see Figure 30b). All further computations use the inverted graph.

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-



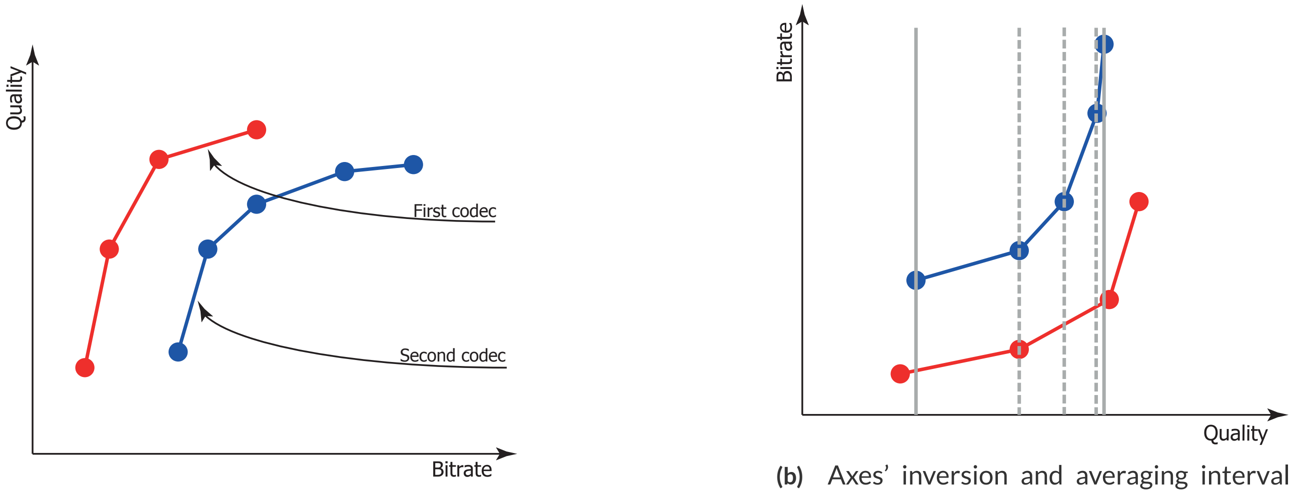
(a) RD curve. "Green" codec is better!

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



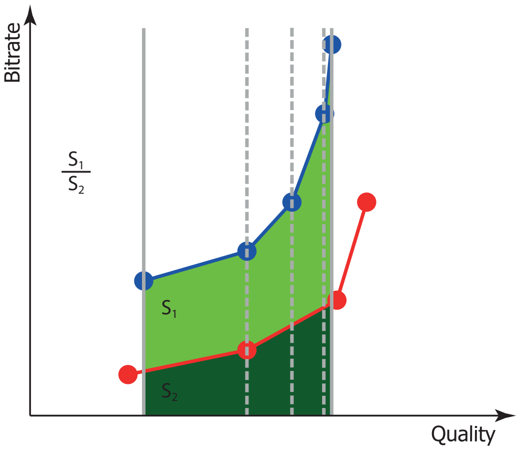
(c) Integral situation with codecs. This plot shows the situation more clearly

Figure 29: Speed/Quality trade-off example



(a) Source RD curves

(b) Axes’ inversion and averaging interval choosing



(c) Areas under curves ratio

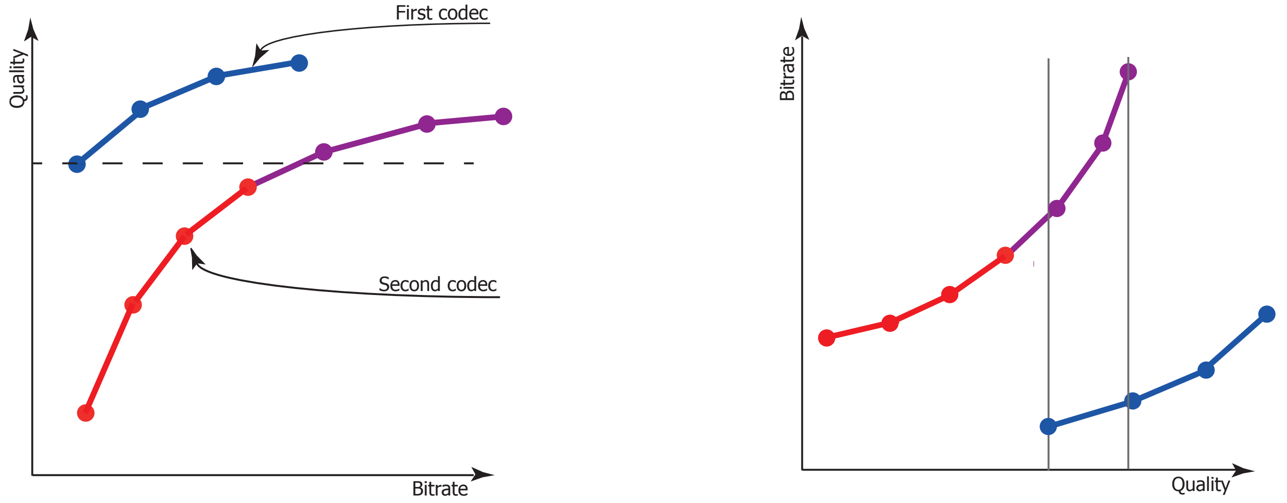
Figure 30: Average bitrate ratio computation

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.

F.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.



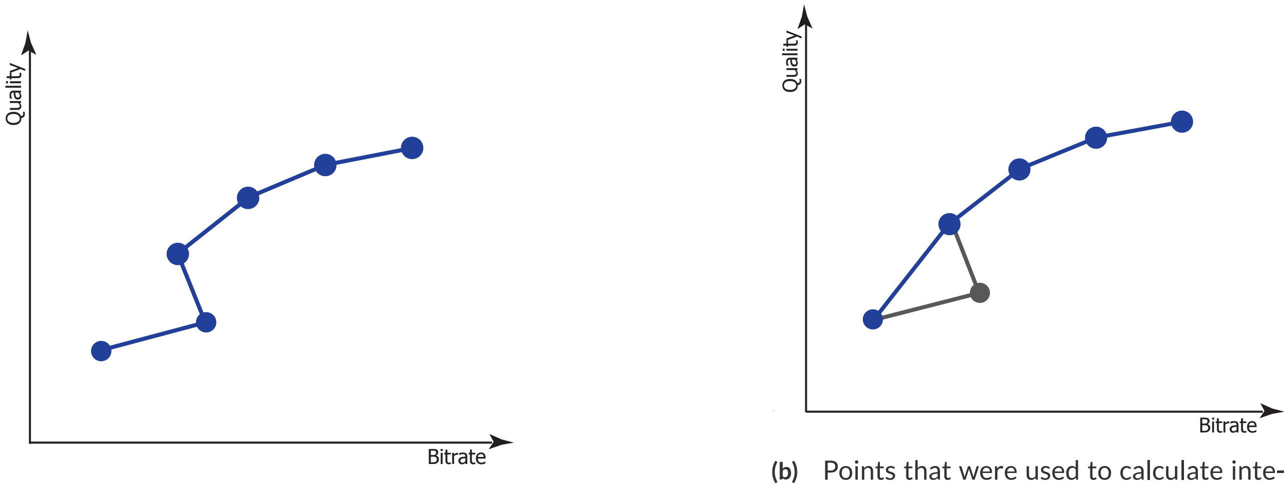
(a) Source RD curves, purple color indicates results for extra bitrates

(b) Axes' inversion and averaging interval choosing

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

F.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.



(a) Non-monotonic RD-curve.

(b) Points that were used to calculate integral.

Figure 32: Processing non-monotonic RD-curves.

F.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 F.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% ☺

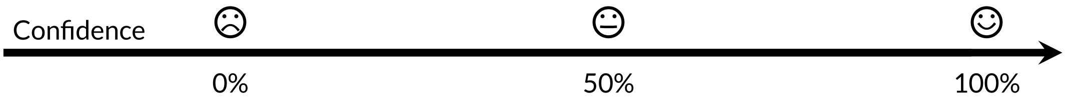


Table 4: Example of average bitrate ratio for a fixed quality table

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 30) 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.

G. OBJECTIVE-QUALITY METRIC DESCRIPTION

G.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.

G.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

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

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.

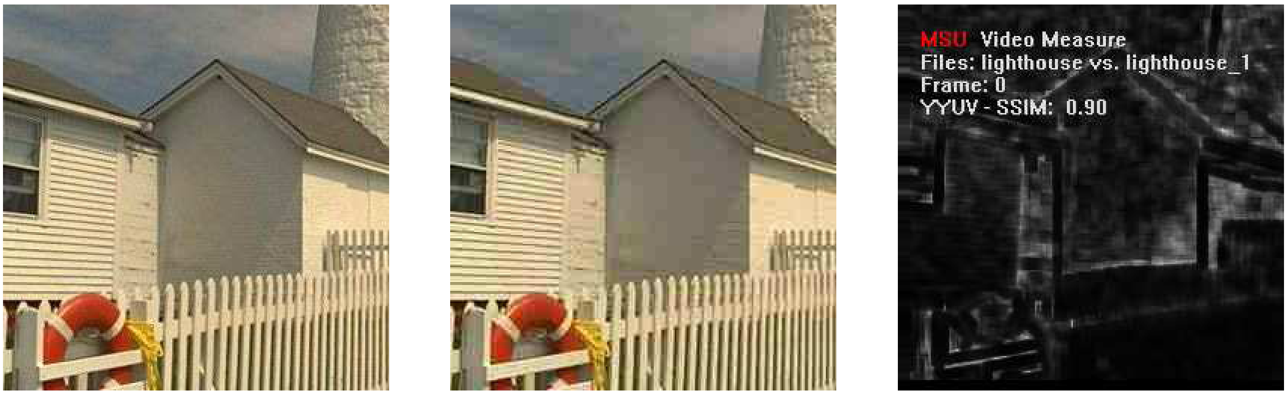
---

[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.

G.1.2. Examples

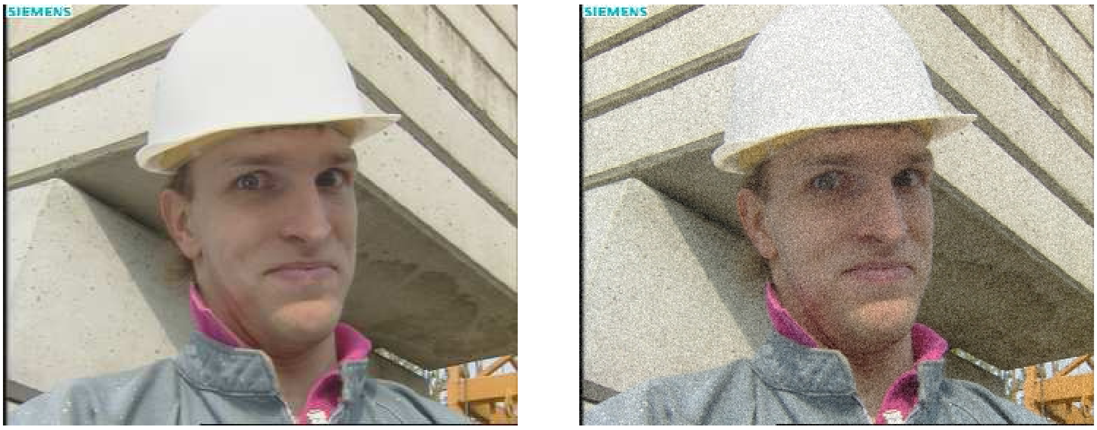
Figure 33 shows an example SSIM result for an original and processed (compressed with lossy compression) image. The value of 0.9 demonstrates that the two images are very similar.



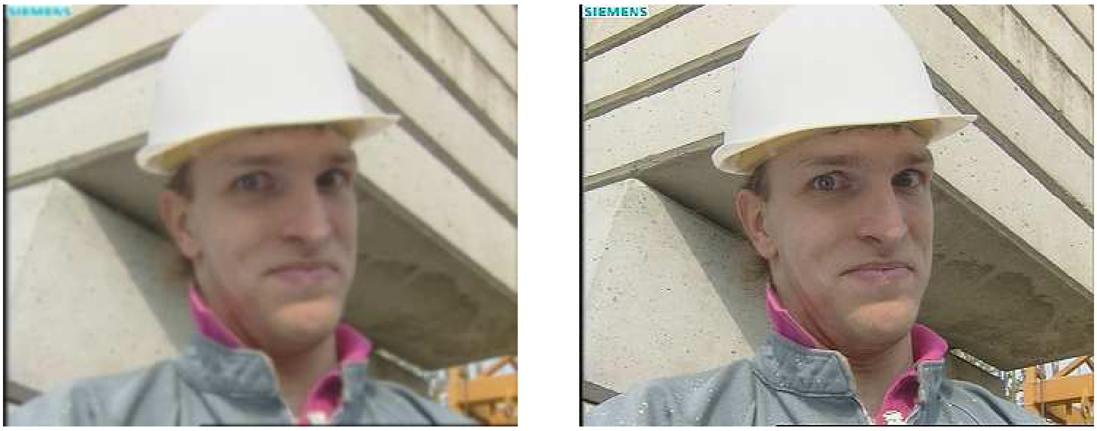
(a) Original **(b)** Compressed **(c)** SSIM

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.

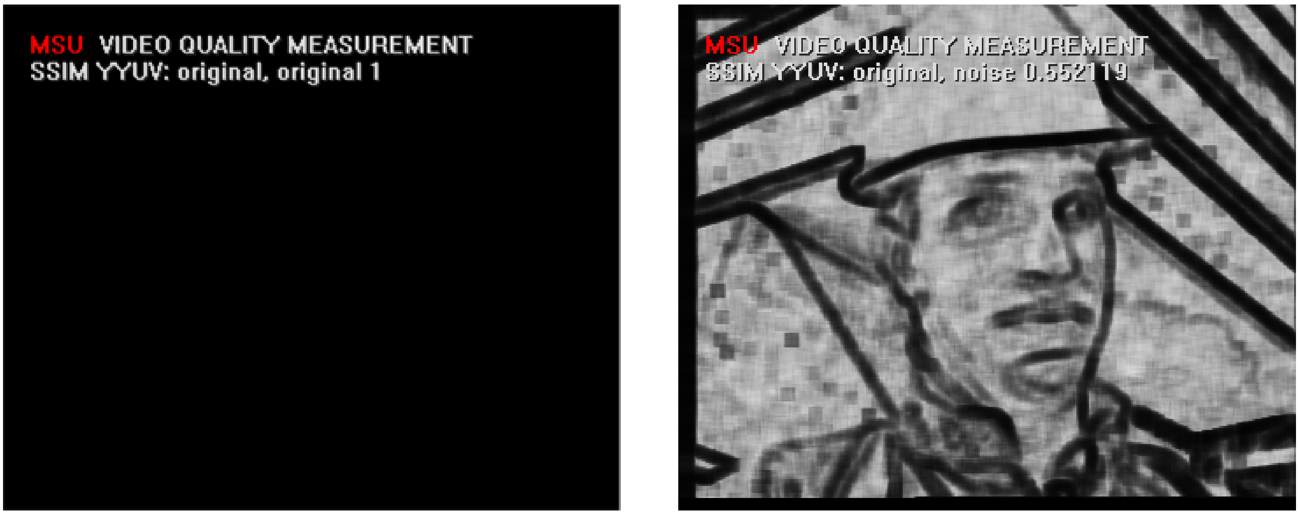


(a) Original image (b) Image with added noise



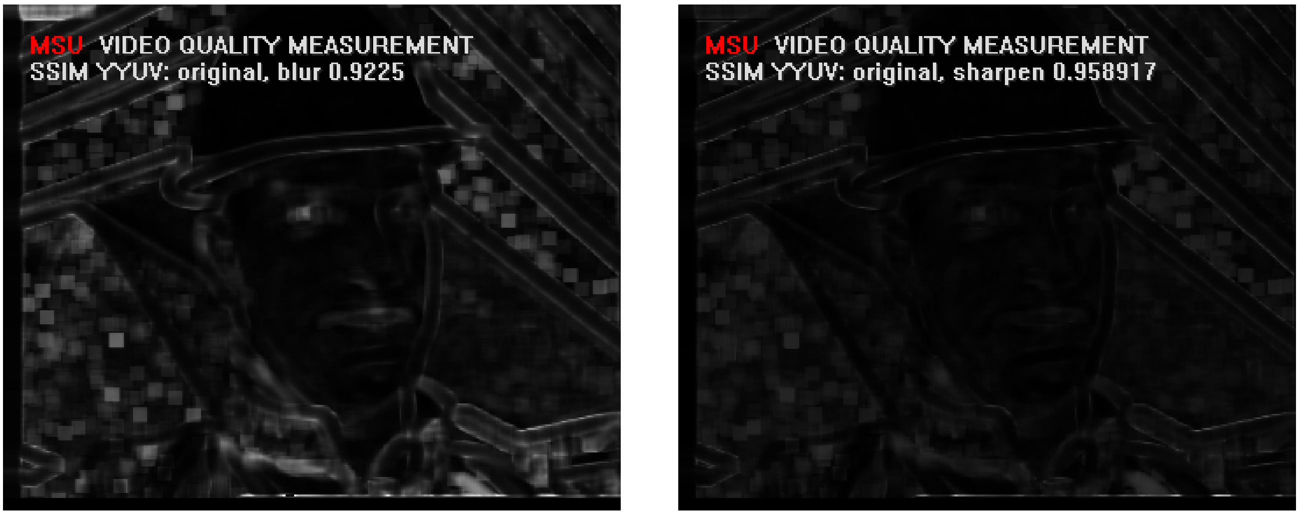
(c) Blurred image (d) Sharpen image

Figure 34: Examples of processed images


MSU VIDEO QUALITY MEASUREMENT
SSIM YYUV: original, original 1
MSU VIDEO QUALITY MEASUREMENT
SSIM YYUV: original, noise 0.552119

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

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


MSU VIDEO QUALITY MEASUREMENT
SSIM YYUV: original, blur 0.9225
MSU VIDEO QUALITY MEASUREMENT
SSIM YYUV: original, sharpen 0.958917

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

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

Figure 35: SSIM values for original and processed images

G.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.

G.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.

G.2.1. PSNR (avg. MSE)

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

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

G.2.2. PSNR (avg. log)

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

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

H. SUBJECTIVE-SCORE ESTIMATION

To conduct an online crowdsourced comparison, we uploaded encoded streams from the previous step to Subjectify.us. The platform hired study participants and showed the upload streams to them in pairs. Each pair consisted of two variants of the same test video sequence encoded by various codecs at various bitrates. Videos from each pair were presented to study participant sequentially (i.e., one after another) in full-screen mode. After viewing each pair, participants were asked to choose the video with the best visual quality. They also had the option to play the videos again or to indicate that the videos have equal visual quality. We assigned each study participant 10 pairs, including 2 hidden quality-control pairs, and each received money reward after successfully completing the task. The quality-control pairs consisted of test videos compressed by the x264 encoder at 1 Mbps and 4 Mbps. Responses from participants who failed to choose the 4 Mbps sequence for one or more quality-control questions were excluded from further consideration. In total we collected 236,736 valid answers from 6,100+ unique participants.

To convert the collected pairwise results to subjective scores, we used the Bradley-Terry model. Thus, each codec run received a quality score. We then linearly interpolated these scores to get continuous rate-distortion (RD) curves, which show the relationship between the real bitrate (i.e., the actual bitrate of the encoded stream) and the quality score. Section ”RD Curves” shows these curves.

I. 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`

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
MSAD
Delta
MSE
VQM
SSIM
MS-SSIM
3-SSIM
VMAF

Spatio-Temporal SSIM
MSU Blurring Metric
MSU Brightness Flicking Metric
MSU Brightness Independent
PSNR
MSU Drop Frame Metric
MSU Noise Estimation Metric
MSU Scene Change Detector
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

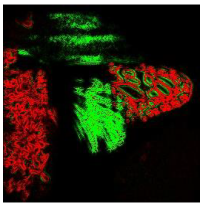
Fastest Video Quality Measurement

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

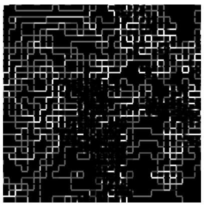
Multi-core Processors Support

Visualization Examples

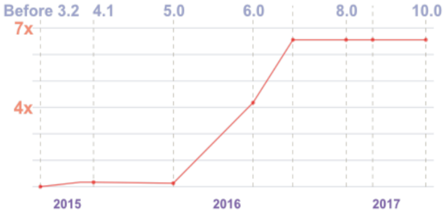
Allows easily detect where codec/filter fails



MSU Blurring Metric



MSU Blocking Metric



VQMT average Speedup

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:

intel, Walt Disney, NVIDIA, skype, NASA, cisco, ORACLE, Microsoft
SONY, MOTOROLA, AMD, facebook, HBO, DOLBY, OBS, QUICKPLAY, NETFLIX, AVerMedia
HUAWEI, aws elemental, harmonic, sky, Red Bull MEDIA HOUSE, ERICSSON, Fraunhofer IDMT, SFU, FOX, NBCUniversal
amazon Lab126, Bell Media, ARM, ATI, AMETEK, ANALOG DEVICES, UBIQUITY, AVAYA, TELUS, AT&T

Effective Video Transcoding

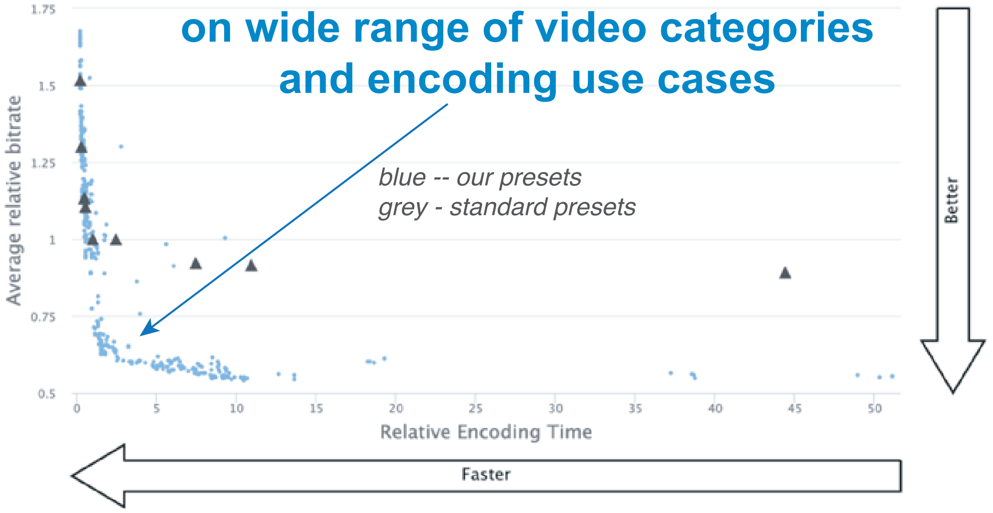
Video-Codec Advanced Tuning

Boost up your codec without changing code!

GRAPHICS & MEDIA LAB

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)

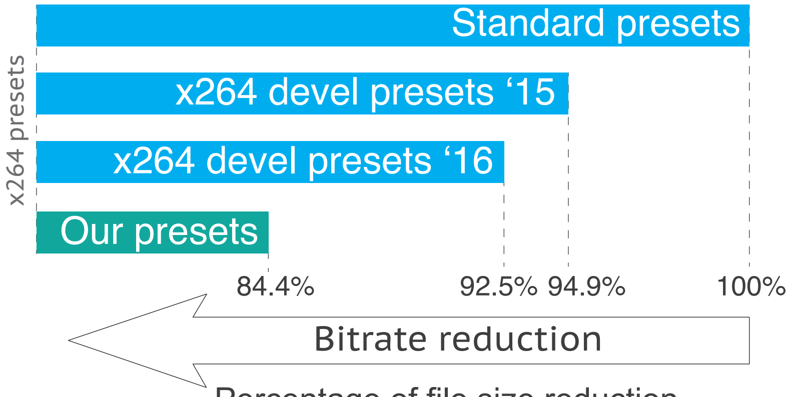
15% bitrate savings in average

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

GRAPHICS & MEDIA LAB VIDEO GROUP