MSU Video Codec Comparison 2017

Part III: Full HD Content, Subjective Evaluation

Minory revised on February 25, 2018

Graphics & Media Lab
Video Group

Video group head	Dr. Dmitriy Vatolin
Project head	Dr. Dmitriy Kulikov
Measurements & analysis	Dr. Mikhail Erofeev,
	Stanislav Dolganov,
	Sergey Zvezdakov

Codecs:

H.265

- Kingsoft HEVC Encoder
- nj265
- x265

Non H.265

- nj264
- SIF Encoder
- uAVS2
- x264

CS MSU Graphics & Media Lab, Video Group

November 21, 2017

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

videocodec-testing@graphics.cs.msu.ru

Powered by Subjectify.us

Contents

1. ACKNOWLEDGMENTS

The Graphics & Media Lab Video Group would like to express its gratitude to the following persons, companies and groups of developers for providing the codecs and settings used in this report:

- x264 developer team
- SIF developer team
- AVS2 developer team
- Nanjing Yunyan
- Kingsoft

The Video Group would also like to thank these companies and developers for their help and technical support during the tests.

2. INTRODUCTION

In this report we describe our subjective comparison of video codecs using a method similar to that of our prior objective comparisons. Instead of objective SSIM quality scores, however, we employ subjective scores obtained from a crowdsourced online study conducted using the Subjectify.us platform (a description of which appears in Section 2.2). We provide a detailed description of the study and score-computation method in Appendix B, as well as a short summary of the study conditions in Section 2.1. To show that our study's crowdsourcing approach is accurate enough to compare video encoders, we replicated a study that Netflix conducted in a controlled laboratory environment, verifying that our results match the laboratory results with a high correlation coefficient (see Appendix C).

This report complements our prior report, *HEVC/H.265 Video Codecs Comparison 2017*, since we compare the same set of video codecs by applying them to video sequences using the same command-line arguments. (The complete list of codecs and command-line arguments appears in Appendix E). But we limit the scope of our study to the "Ripping" use case (i.e., all codecs should have a mean encoding speed greater than 1 FPS). Note that command-line arguments for two of the codecs in our comparison (x264 and x265) enable the `--tune ssim` option, which is designed to maximize the SSIM objective score of the encoded sequence but may sacrifice perceived quality. In Appendix D we show that under our study conditions, this option should remain in the arguments.

The rest of the report is organized as follows:

1. Section 3 contains rate-distortion (RD) plots showing the relationship between bitrate and subjective score.
2. Section 4 contains plots depicting the mean speed and quality scores that encoders achieve in our comparison relative to x264 (which we use as a reference).
3. Section 5 shows our results for pairwise codec comparisons (i.e., relative to each other).
4. Finally, Section 6 presents overall relative quality scores (i.e., mean test-encoder bitrate divided by mean reference-encoder bitrate for the same range of subjective scores) and compares them to relative quality scores computed using SSIM.

2.1. Study Conditions

Encoders under comparison: Seven software video encoders (x264, x265, nj264, nj265, SIF Encoder, Kingsoft HEVC Encoder and AVS2) with preselected command-line arguments that deliver at least a 1 FPS encoding speed. See the complete list in Appendix E.

Test video sequences: Four Full HD video sequences with frame rates of 24–25 FPS. See the complete list in Appendix F.

Encoding bitrates: 1 Mbps, 2 Mbps and 4 Mbps.

Test hardware: All codecs ran on an Intel Core i7-6700K (Skylake) @ 4GHz with 8GB of RAM and Windows 8.1.

Computation of subjective quality scores: Using the Subjectify.us platform, we showed study participants pairs of videos encoded at various bitrates by the codecs under evaluation. We asked them to choose the video with the best visual quality from each pair. To filter out responses from participants who made thoughtless decisions, we also asked them hidden quality-control questions. We collected 11,530 valid answers from 325 unique participants and converted pairwise responses to subjective scores using the crowd Bradley-Terry model [1]. A detailed description of this step appears in Section B.2.

Computation of integral quality and speed scores: To summarize an encoder's performance at multiple bitrates, we computed relative quality and speed scores. The relative quality score is the test encoder's mean bitrate divided by the reference encoder's mean bitrate for the same range of quality scores. The relative speed score is the test encoder's mean encoding speed divided by the reference encoder's mean speed for the same bitrate range. A detailed description of integral-score computation appears in Section B.3.

2.2. 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.

Subjectify.us is currently in a private beta stage. To try the platform in your research project, request early access at www.subjectify.us. This demo video shows an overview of the Subjectify.us workflow.

3. RD CURVES

The plots below depict subjective quality scores for encoders at various bitrates and on various test sequences (a description of how we compute the scores for this study is in Section B.2). No one codec is an absolute winner for all sequences. Nevertheless, two are clear leaders: the Kingsoft HEVC Encoder and x265. They are among the top three encoders for all sequences. Judging from the mean quality scores (computed using the method described in Section B.3), first place in the quality competition goes to the Kingsoft HEVC Encoder, second place goes to x265, and third place to AVS2.

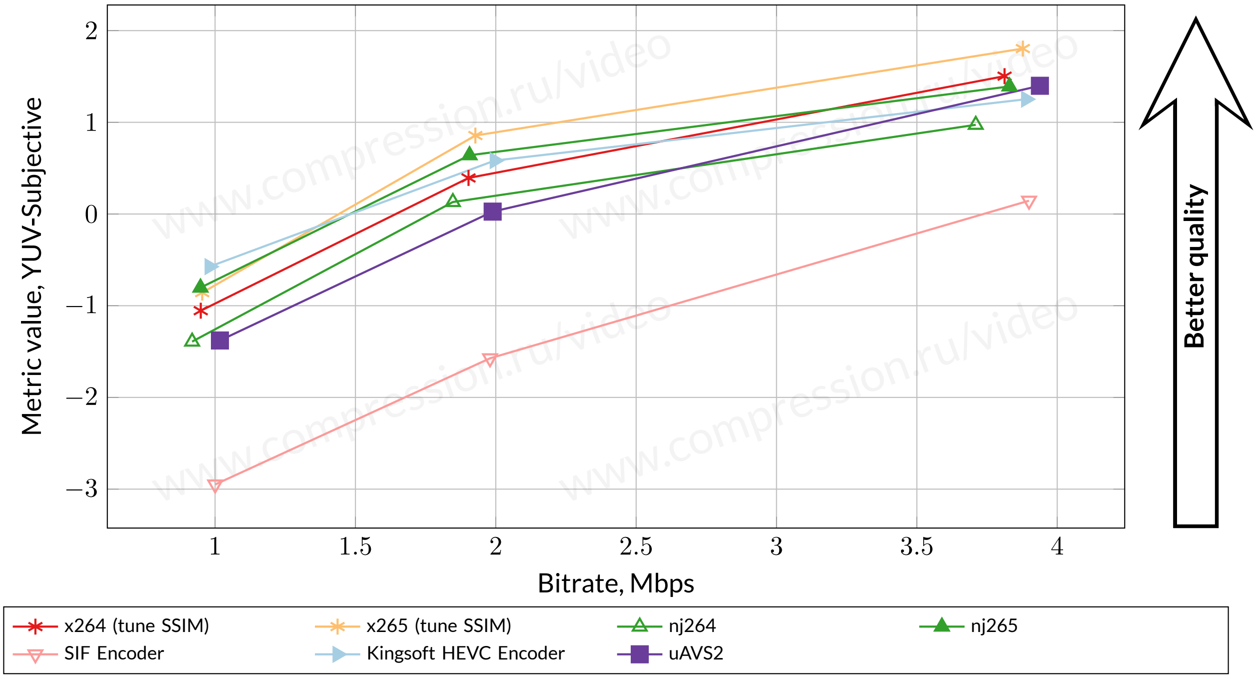


Figure 1: Bitrate/quality—use case "Ripping," *Fountain* sequence, subjective quality scores.

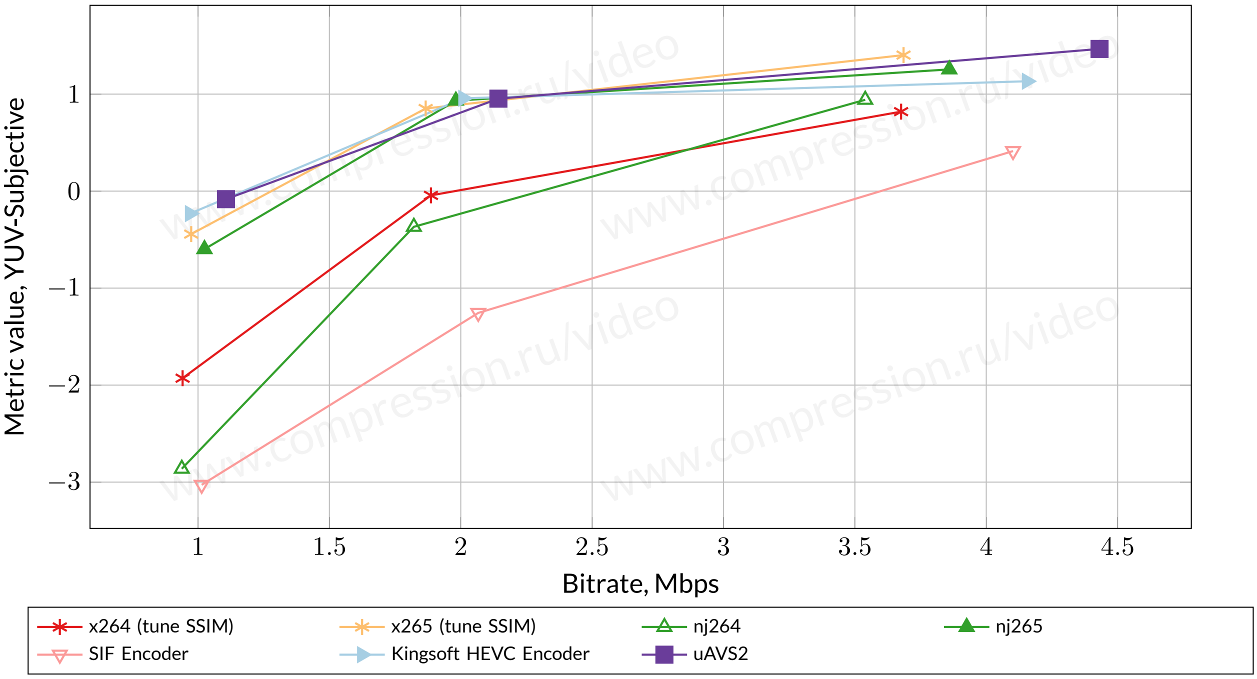


Figure 2: Bitrate/quality—use case "Ripping," *Mountain Bike* sequence, subjective quality scores.

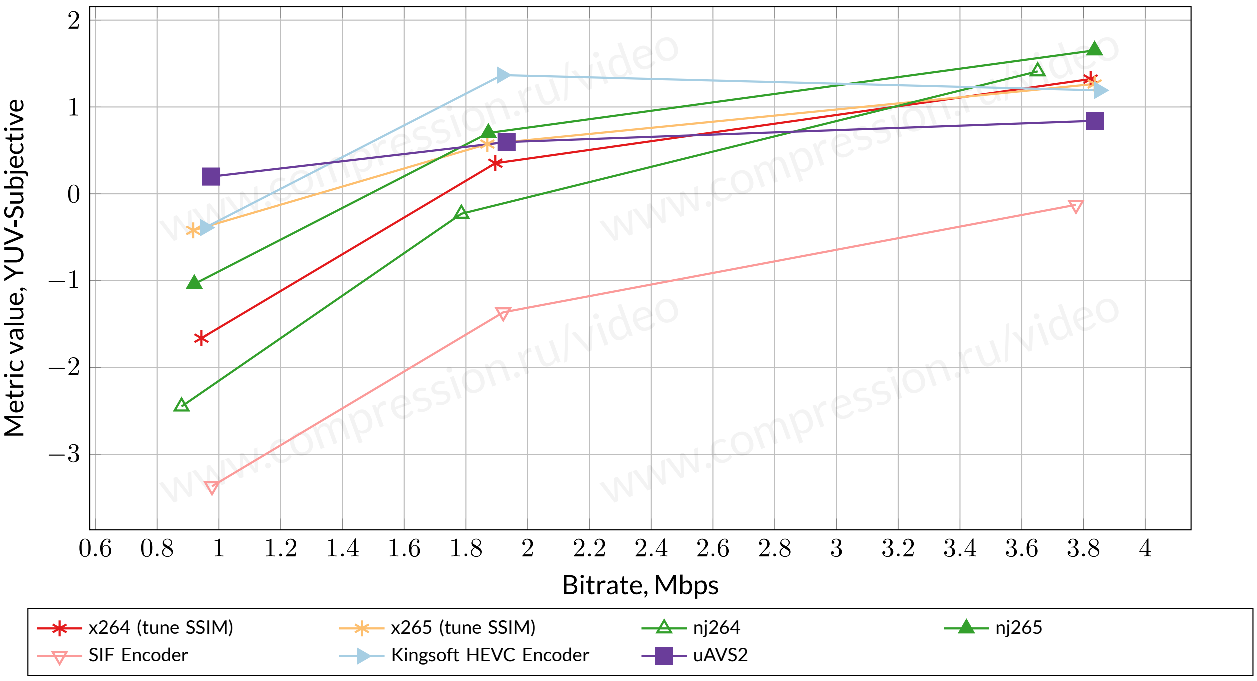


Figure 3: Bitrate/quality—use case "Ripping," *Wedding* sequence, subjective quality scores.

Figure 4 depicts rate-distortion (RD) curves for the *Ziguinchor* sequence. It shows an atypical shape for the AVS2 curve: the quality score decreases as the bitrate increases from 2 Mbps to 4 Mbps. This phenomenon can be explained by the relatively high level of grain noise in the source sequence. At a 2 Mbps bitrate the AVS2 encoder produced a sequence with few compression artifacts and almost no grain from the source sequence (owing to the limited bitrate). A 4 Mbps bitrate, however, enabled it to reproduce the source's grain noise. We believe that during subjective comparisons, observers perceived a higher level of grain noise as an artifact and therefore assigned lower quality scores at 4 Mbps than at 2 Mbps.

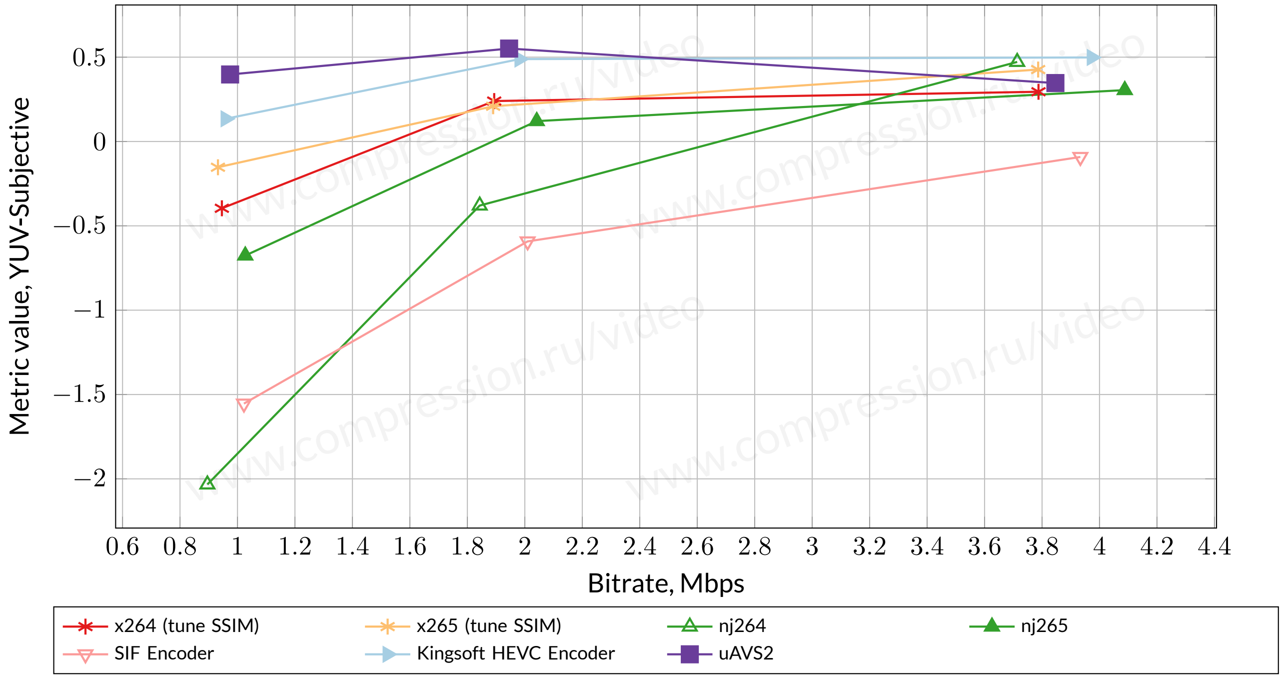


Figure 4: Bitrate/quality—use case "Ripping," *Ziguinchor* sequence, subjective quality scores.

4. SPEED/QUALITY TRADE-OFF

In the figures below we show overall and per-sequence relative speed and quality scores computed using the method described in Section B.3. Figure 5 shows overall scores, demonstrating that no codec is the absolute leader in both speed and quality. We can, however, identify Pareto-optimal candidates: the Kingsoft HEVC Encoder, x265, AVS2 and the SIF Encoder.

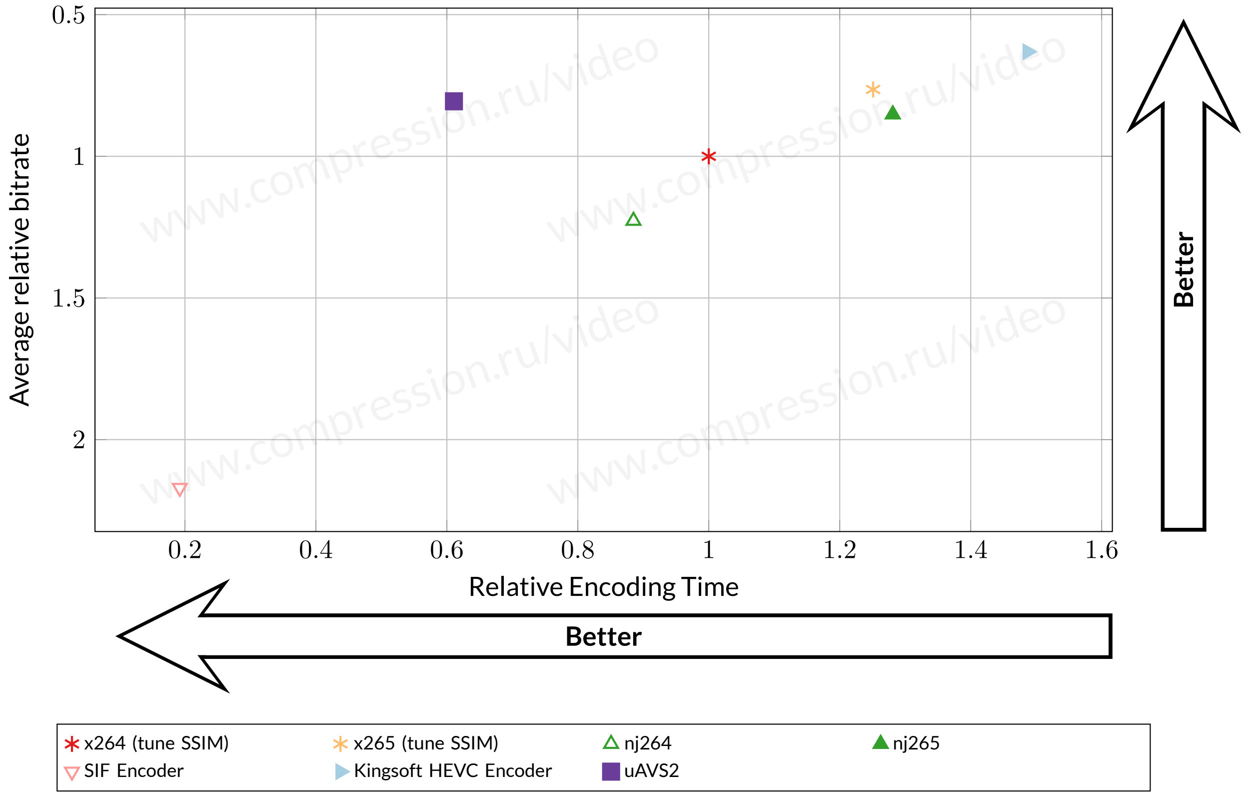


Figure 5: Speed/quality trade-off—use case "Ripping," all sequences, subjective quality scores.

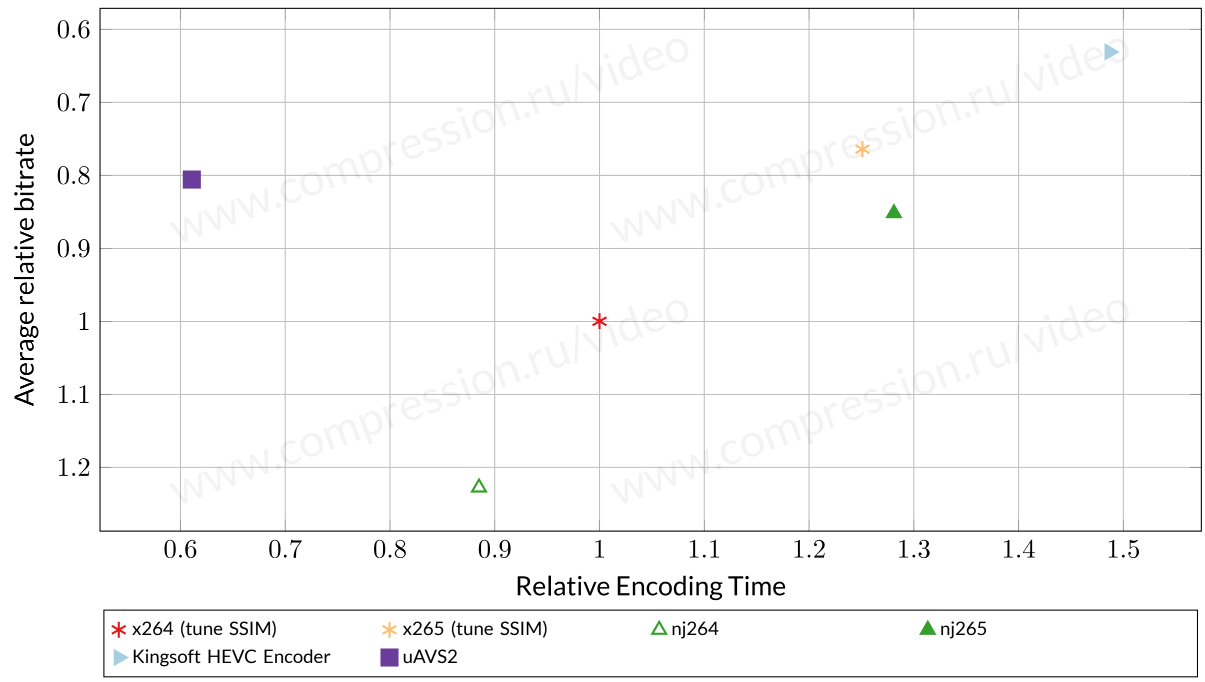


Figure 6: Speed/quality trade-off—use case "Ripping," all sequences, subjective quality scores, without SIF Encoder.

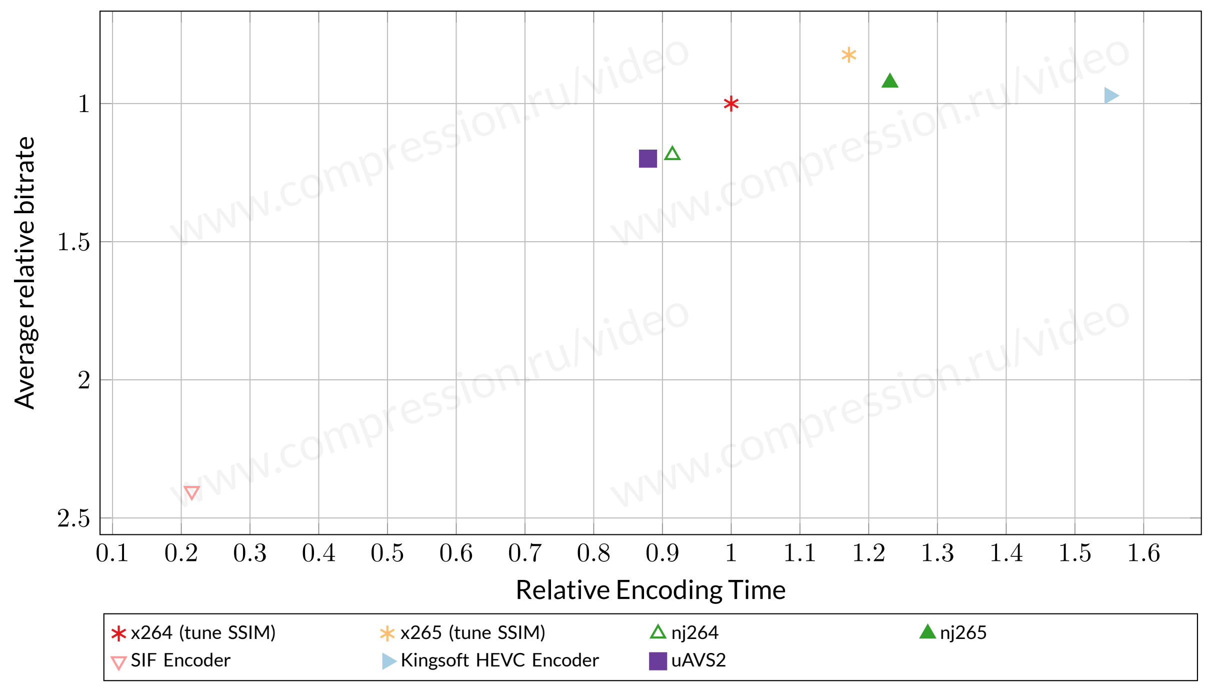


Figure 7: Speed/quality trade-off—use case "Ripping," *Fountain* sequence, subjective quality scores.

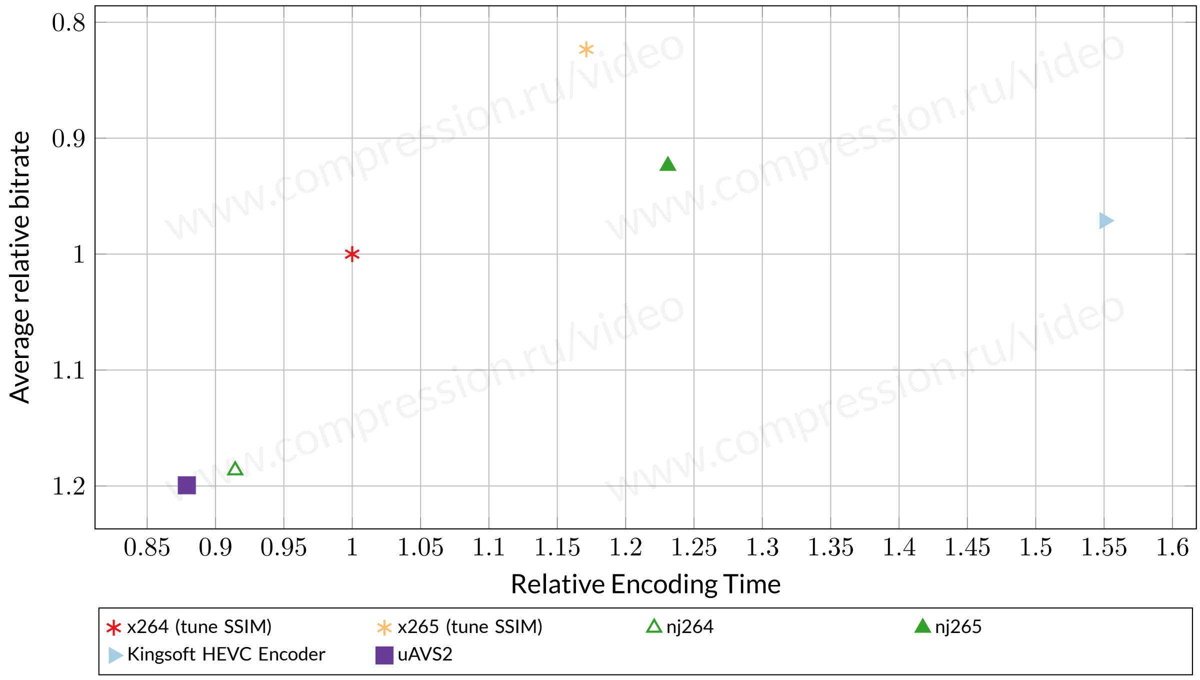


Figure 8: Speed/quality trade-off—use case "Ripping," *Fountain* sequence, subjective quality scores, without SIF Encoder.

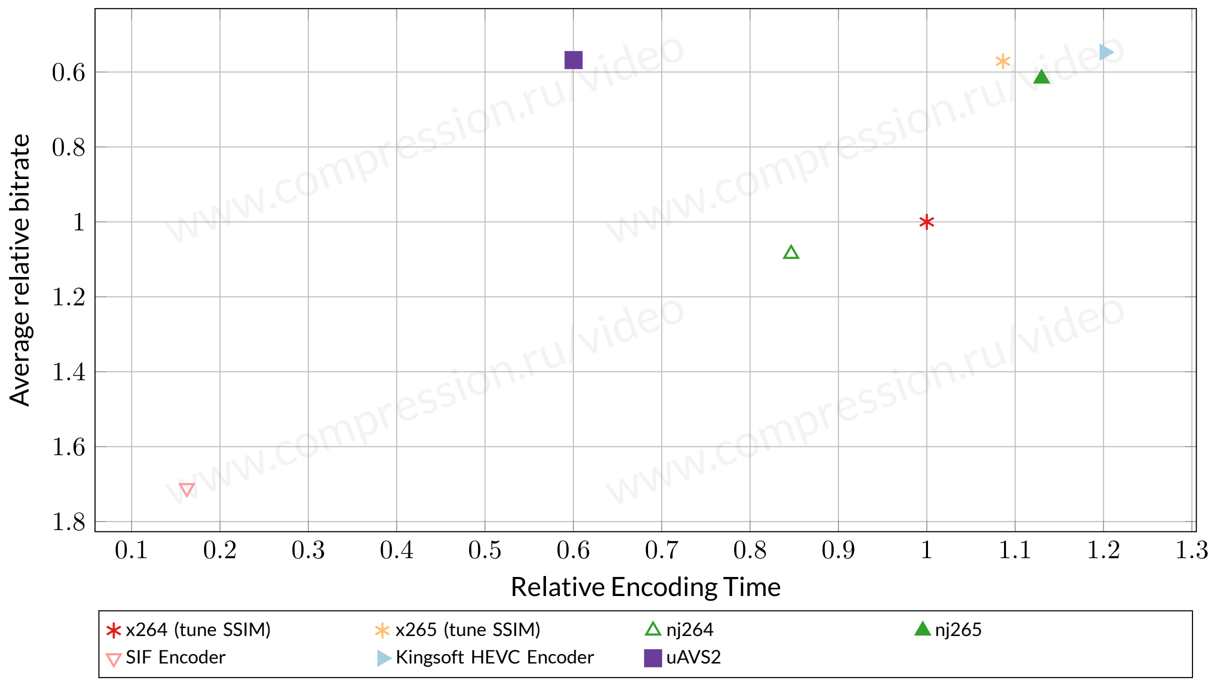


Figure 9: Speed/quality trade-off—use case "Ripping," *Mountain Bike* sequence, subjective quality scores.

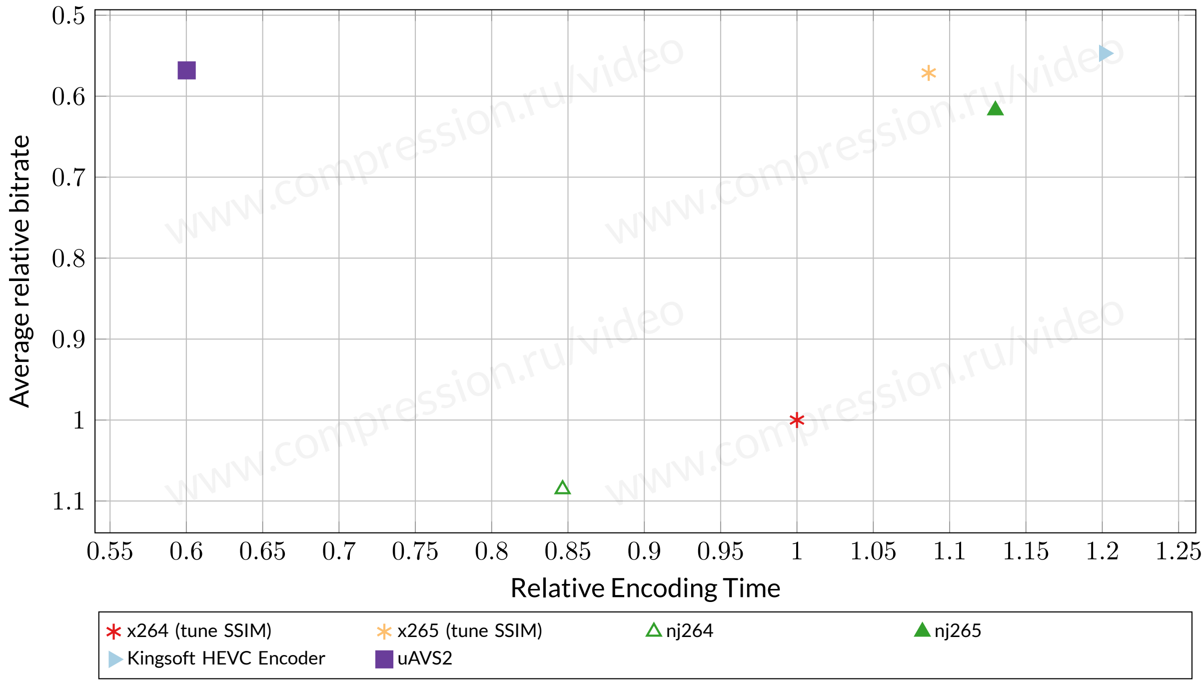


Figure 10: Speed/quality trade-off—use case "Ripping," *Mountain Bike* sequence, subjective quality scores, without SIF Encoder.

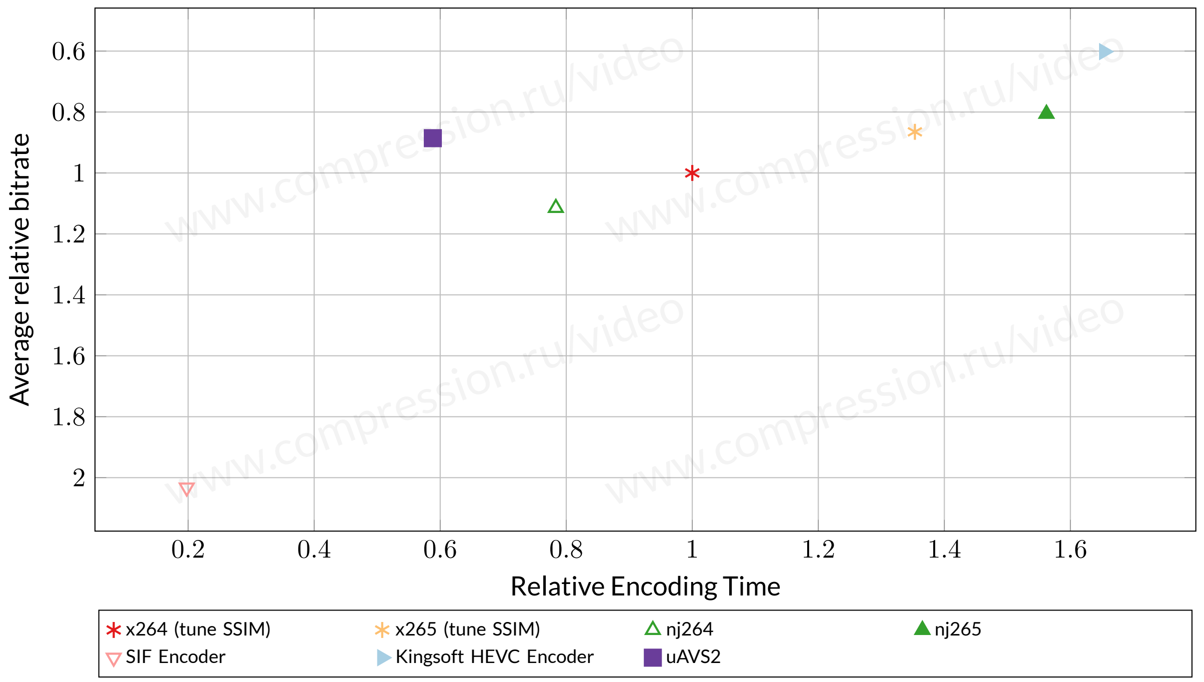


Figure 11: Speed/quality trade-off—use case "Ripping," *Wedding* sequence, subjective quality scores.

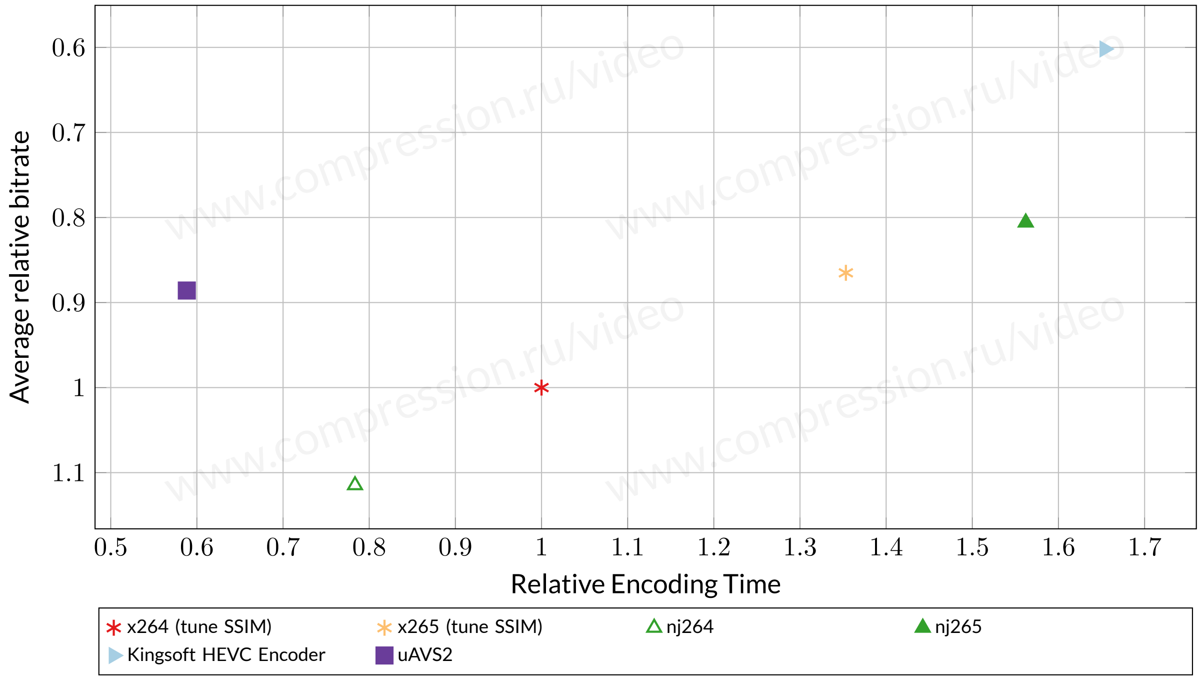


Figure 12: Speed/quality trade-off—use case "Ripping," *Wedding* sequence, subjective quality scores, without SIF Encoder.

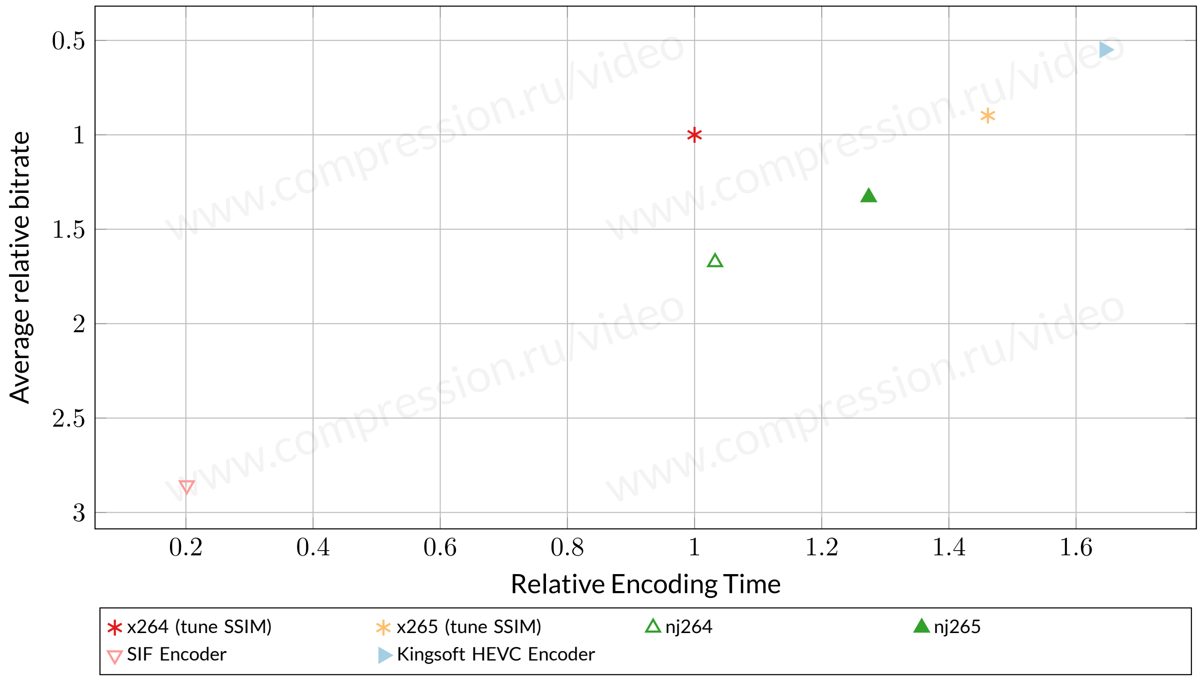


Figure 13: Speed/quality trade-off—use case "Ripping," *Ziguinchor* sequence, subjective quality scores.

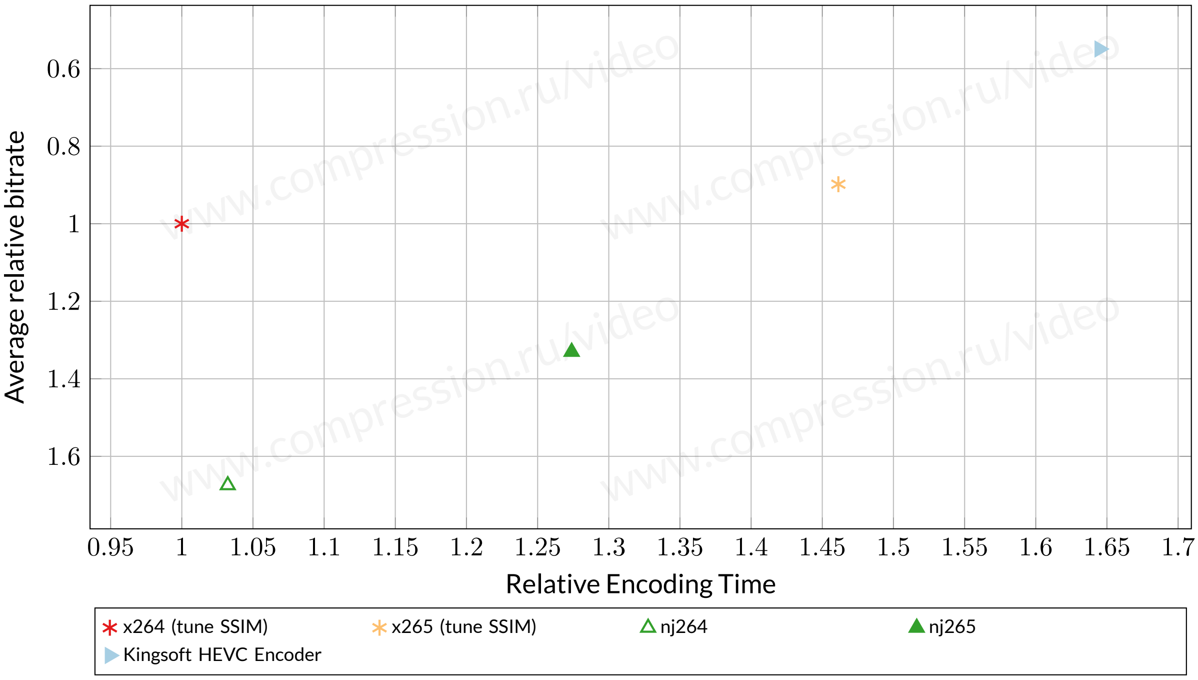


Figure 14: Speed/quality trade-off—use case "Ripping," *Ziguinchor* sequence, subjective quality scores, without SIF Encoder.

5. RELATIVE QUALITY ANALYSIS

The data we present in this section is useful for pairwise encoder comparison. Section B.3.2 provides an explanation of the table and plot below. Note that because of our study's integral-score computation method, we calculated each number in the table below for a particular quality range, which may differ drastically among encoders owing to differences in their performance. These differences can lead to inadequate results when using the table to compare three or more codecs at once.

	x264 (tune SSIM)	x265 (tune SSIM)	nj264	nj265	SIF Encoder	Kingsoft HEVC Encoder	uAVS2
x264 (tune SSIM)	100.0% ☺	131.0% ☺	81.0% ☺	117.0% ☺	46.0% 😐	158.0% 😐	124.0% 😐
x265 (tune SSIM)	79.0% ☺	100.0% ☺	67.0% ☺	89.0% ☺	32.0% ☹	128.0% ☺	151.0% 😐
nj264	127.0% ☺	151.0% 😐	100.0% ☺	142.0% 😐	64.0% ☺	182.0% 😐	176.0% 😐
nj265	92.0% ☺	115.0% ☺	71.0% ☺	100.0% ☺	41.0% 😐	149.0% 😐	90.0% 😐
SIF Encoder	225.0% 😐	316.0% ☹	165.0% ☺	248.0% 😐	100.0% ☺	305.0% ☹	239.0% ☹
Kingsoft HEVC Encoder	67.0% ☺	86.0% ☺	57.0% ☺	79.0% ☺	33.0% ☹	100.0% ☺	101.0% ☺
uAVS2	88.0% ☺	96.0% ☺	68.0% ☺	112.0% ☺	43.0% ☹	107.0% ☺	100.0% ☺

Confidence ☹ 😐 ☺

0% 50% 100%

Table 1: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, subjective quality scores.

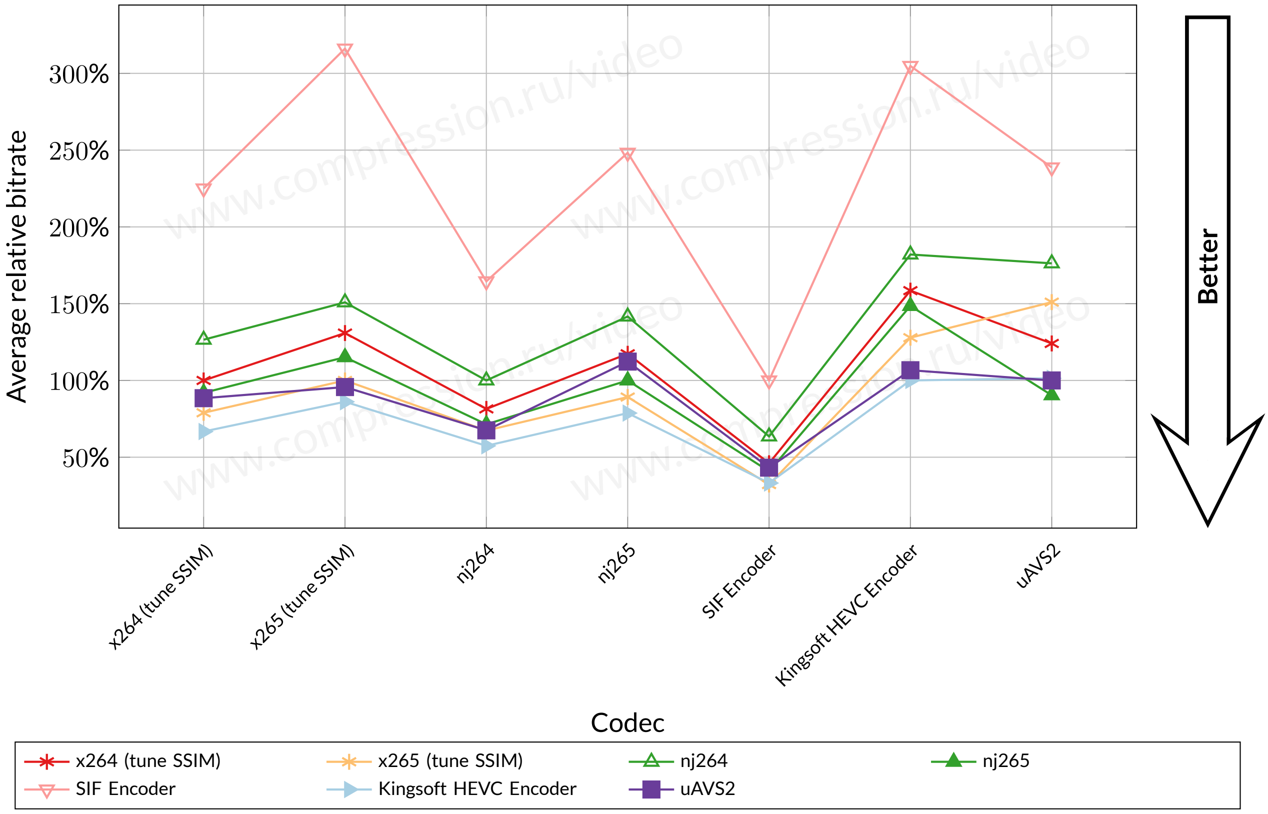


Figure 15: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, subjective quality scores.

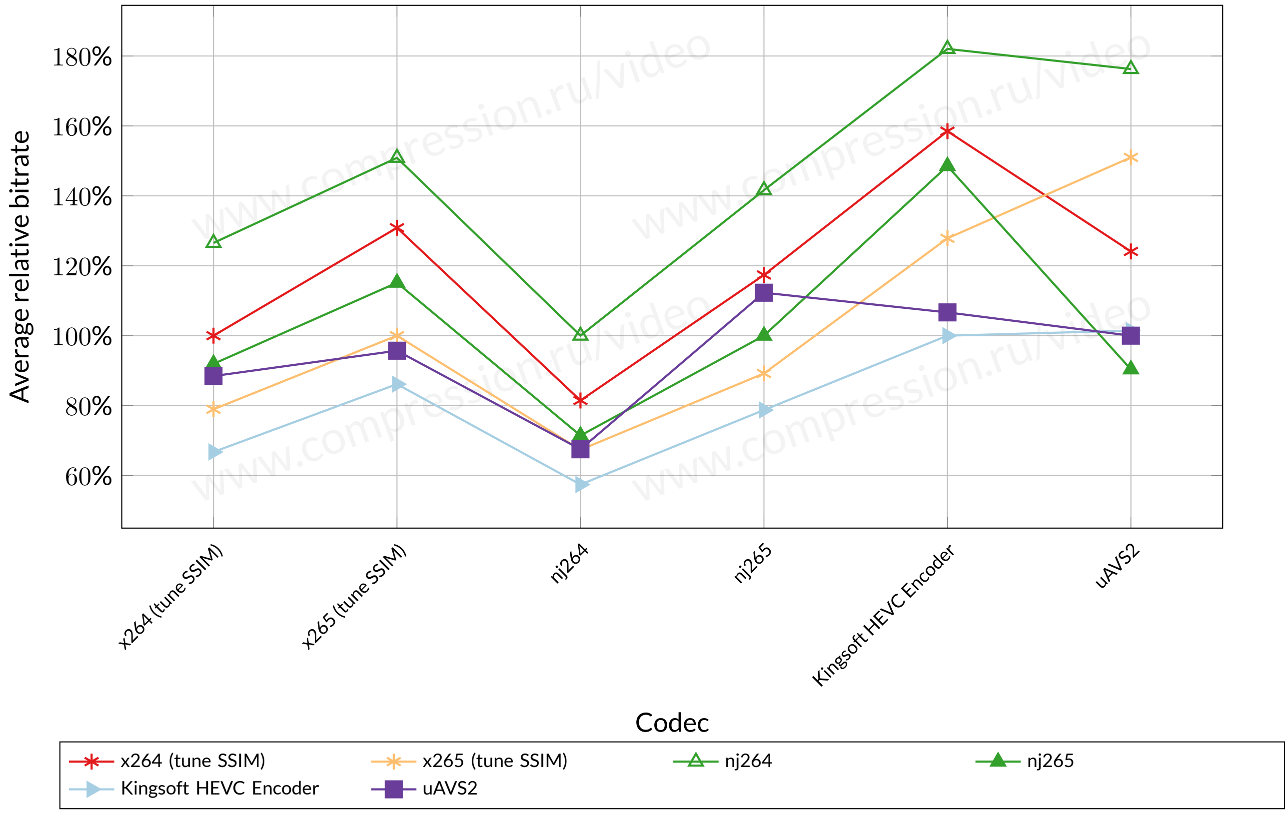


Figure 16: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, subjective quality scores, without SIF Encoder.

6. CONCLUSION

The plot below shows overall quality scores for the encoders in our comparison (see Section B.3 for a description of the integral-score computation method). First place goes to the Kingsoft HEVC Encoder. Second place goes to x265 and third place to AVS2.

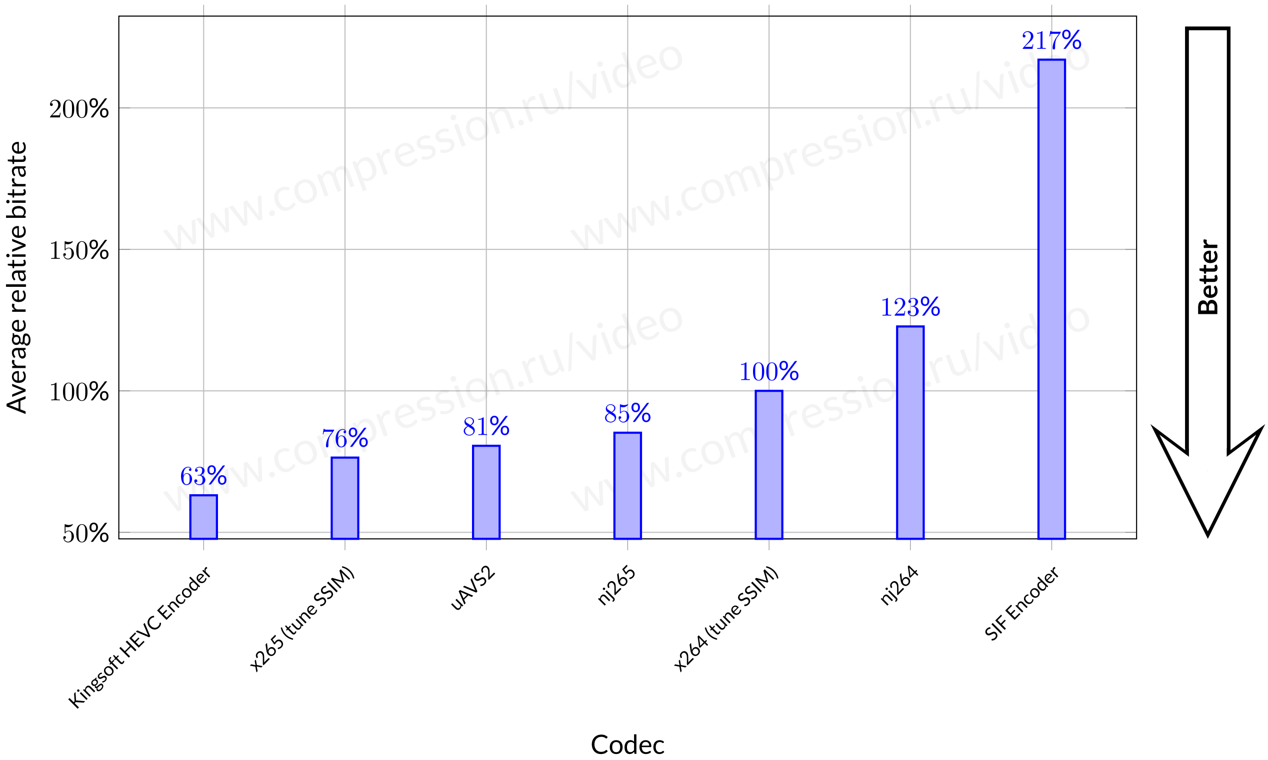


Figure 17: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, subjective quality scores.

Finally, we compare the plot above with a plot generated using the same method but based on *objective* SSIM quality scores [3] instead of *subjective* scores (see Figure 18). As the plots indicate, although the absolute scores exhibit significant differences, the order of encoders is almost the same (except AVS2 and nj265 have swapped places). We believe this similarity is evidence that the method we used to estimate subjective scores is accurate (see Appendix C for a more formal evaluation of that method) and also highlights the importance of conducting subjective tests alongside objective ones.

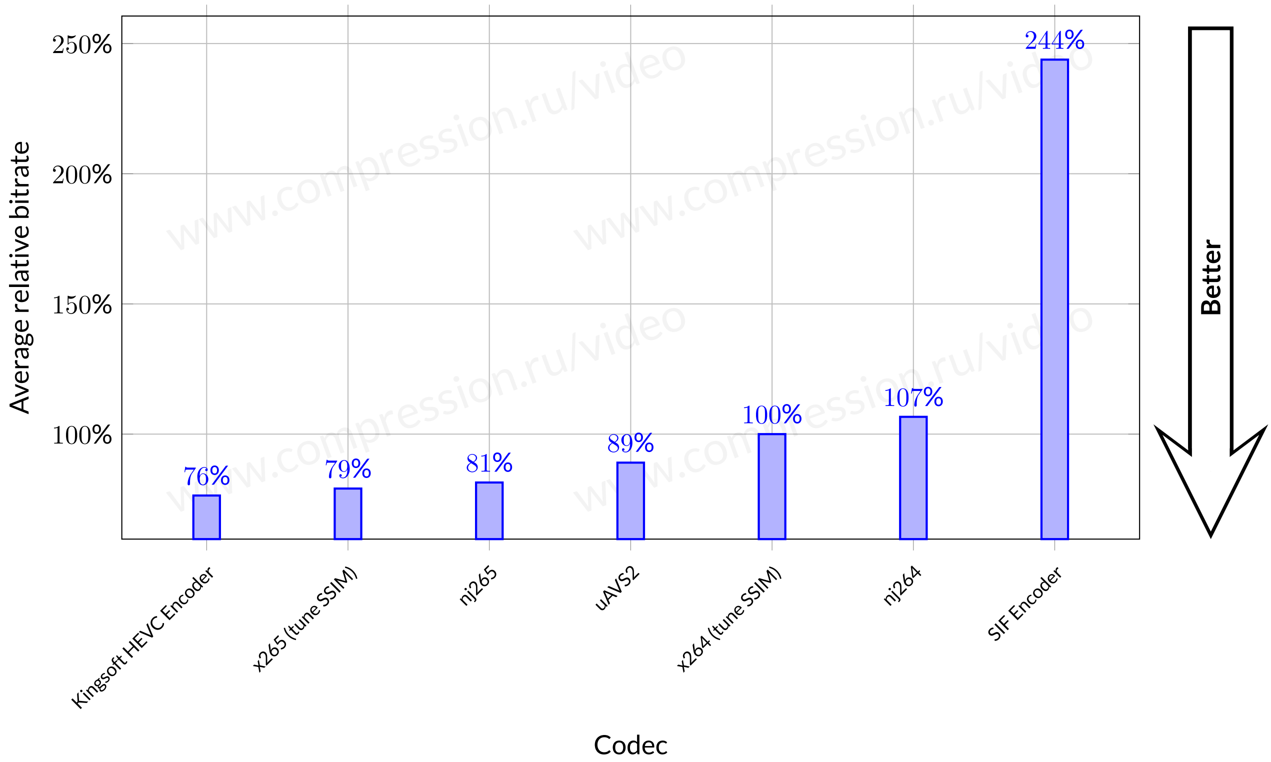


Figure 18: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, YUV-SSIM metric.

A. PARTICIPANTS' COMMENTS

A.1. Kingsoft

Thanks a lot for MSU's support of this comparison and the results sound good for us. Nevertheless, 2 pass can still be set to improve the BDRate by 3% with slight time increasing, when first pass only requires superfast preset. Moreover, Placebo preset is never carefully tunned and might not have a good trade off between speed and compression ratio.

B. STUDY METHOD

The goal of our study is to rank modern video codecs according to the subjective visual quality of the compressed videos they produce. We therefore employed the method that prior MSU objective-codec comparisons employed, but we replaced the objective SSIM quality scores with subjective scores estimated using Subjectify.us. For the sake of completeness, however, we provide a full description of the method below (including internal details of Subjectify.us).

The method comprises three main steps:

1. Video encoders launches (see Section B.1)
2. Subjective-score estimation (see Section B.2)
3. Integral-score estimation (see Section B.3)

B.1. Running Codecs

In this study we compare the same encoders as in our previous report, *HEVC/H.265 Video Codecs Comparison 2017*, using the same command-line arguments from the "Ripping" use case (i.e., a minimum encoding speed of 1 FPS).

We applied software encoders with preselected command-line arguments (see the full list of codecs in Appendix E) to four Full HD test video sequences (see the full list in Appendix F) at three bitrates: 1 Mbps, 2 Mbps and 4 Mbps. All encoders ran on a computer with an Intel Core i7-6700K (Skylake) processor @ 4GHz, 8GB of RAM and Windows 8.1. The source and encoded files, the encoder executable, and the operation system all resided on an SSD. We ran every encoder three times and recorded the duration of the fastest run for each.

B.2. Subjective-Score Estimation

To conduct an online crowdsourced comparison, we uploaded encoded streams from the previous step to Subjectify.us and then showed them to study participants 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 $0.05 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. Study participants could take part in the study up to 10 times. In total we collected 11,530 valid answers from 325 unique participants.

To convert the collected pairwise results to subjective scores, we used the crowd Bradley-Terry model [1]. 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 3 shows these curves.

B.3. Integral-Score Estimation

To compare not just individual encoder runs but all runs for a single sequence and, to obtain an overall score for each encoder, we computed integral relative scores: the relative quality score and the relative speed score.

Relative quality score is the test encoder's mean bitrate divided by a reference encoder's mean bitrate for the same range of quality scores. The relative quality score x% for test encoder *A* means the following: *A* must deliver x% of the reference encoder's bitrate to achieve the same visual quality. To compute this score, we employ the following procedure:

1. Transpose the RD curve for both the test codec and reference codec (see Figures 19a and 19b).
2. Find the RD curves' projection onto the quality axis—that is, the largest quality range for which both curves are defined (see Figure 19b).
3. Compute the area under the curves for the quality range from the previous step (see Figure 19c).
4. Define the relative quality score as the area under the test codec's RD curve divided by the area under the reference codec's RD curve.
5. Additionally, to score the estimate from the previous step, define the confidence as the length of the quality range used to compute the area divided by the length of quality range for which the reference RD curve is defined. Section 5 depicts these scores with emoticons.

To compute an overall quality score for the test encoder, we average its relative quality scores for the individual test sequences. Section 6 shows overall quality scores.

Relative speed score is the mean encoding speed of the test encoder divided by the mean encoding speed of the reference encoder for the same bitrate range. To compute this score, we employ the same procedure as above but compute the area under the encoding-speed curves, rather than the RD curves, along the bitrate axis.

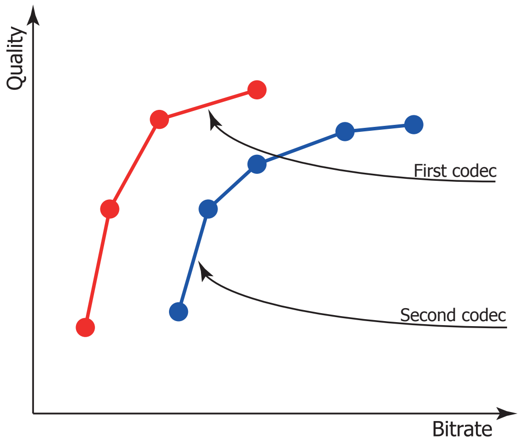
B.3.1. Speed/Quality Trade-Off Plots

To compare both relative quality and relative speed scores, in Section 4 we show speed/quality trade-off plots (the x-axis corresponds to the speed score and the y-axis to quality score). These plots enable us to see whether a codec won first place in both categories (speed and quality). If no absolute winner emerges, the plot helps in finding Pareto-optimal encoders (i.e., encoders for which no competitor has a higher score in both speed and quality).

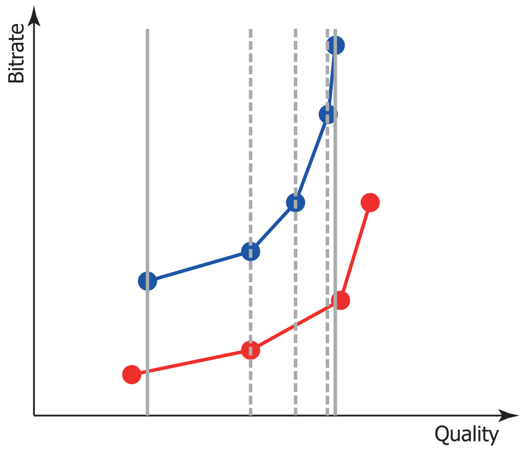
Consider the simplified example in Figure 20. As Figure 20a shows, the "green" codec outperforms the "black" codec in quality scores. But the black codec is faster, according to Figure 20b. We can make the same observation at a glance by considering the speed/quality trade-off plot in Figure 20c: green earned a higher quality score by (possibly) sacrificing speed, thus falling short of black in that category. In this example, neither competitor is the absolute winner. Both, however, are Pareto-optimal candidates.

B.3.2. Relative Quality Analysis

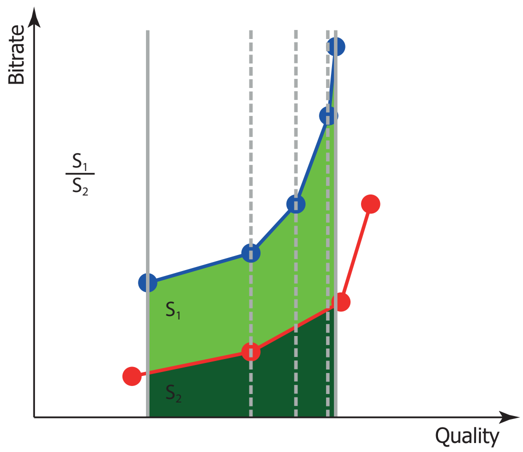
In Section 5 we compare encoders pairwise (i.e., one versus another). Using each codec as a reference, we then compute quality scores for all the others relative to that codec. These results are useful when comparing two



(a) Source RD curves.



(b) Curves transposition and scores computation interval choice.



(c) Areas under curves ratio.

Figure 19: Relative quality score computation.

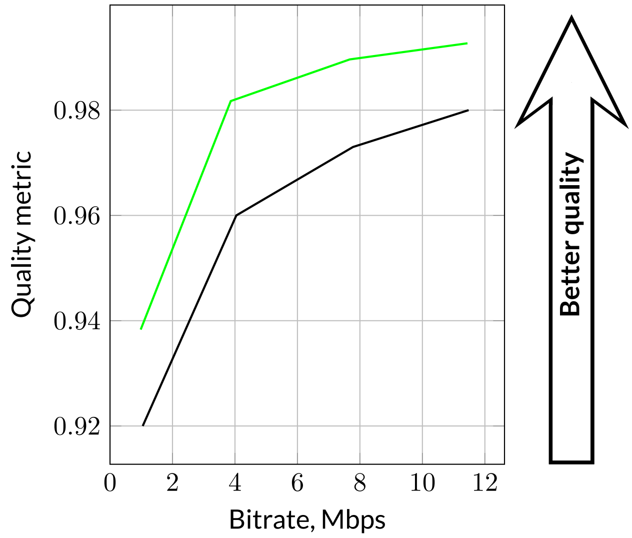
	A	B
A	100% ☺	75% ☹
B	134% ☹	100% ☺

Table 2: Example of relative quality analysis table.

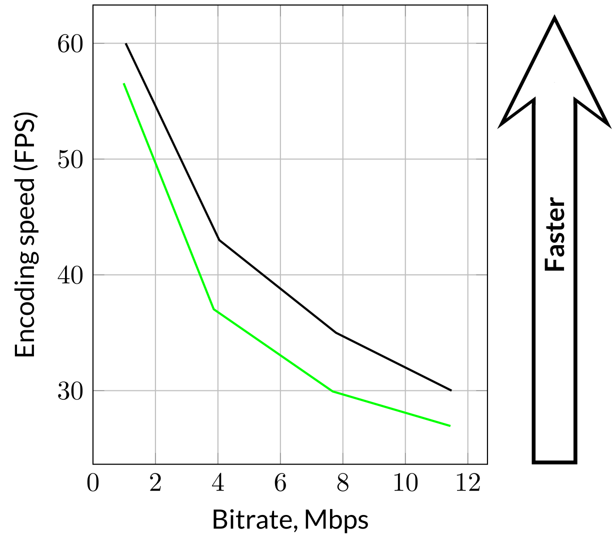
particular encoders.

In Table 2 we show a simplified example comparison of hypothetical codecs *A* and *B*. Consider row *B*, column *A*, which contains the value 134%. This value means that to produce an encoded stream of the same quality as *A*, *B* must deliver 134% of *A*'s bitrate. The emoticon in the cell depicts the confidence for this estimate.

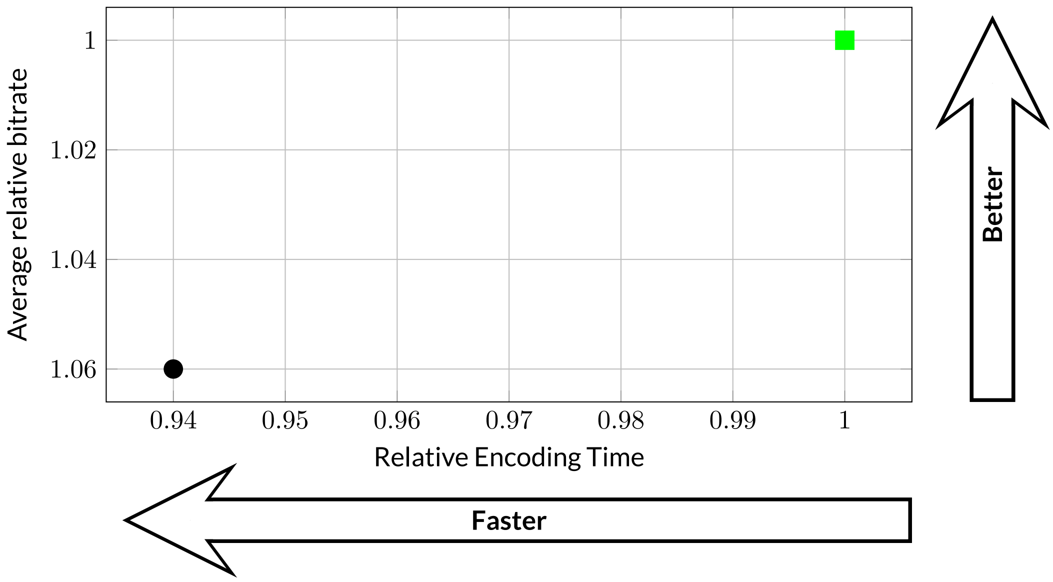
The plot in Section 5 depicts data from the table: each line on the plot corresponds to a table row.



(a) RD curve.



(b) Encoding speed (frames per second).



(c) Trade-off plot.

Figure 20: Speed/quality trade-off example.

C. STUDY-METHOD EVALUATION

Undoubtedly, a subjective study conducted in a controlled laboratory environment with a huge panel of observers is the gold standard for comparison of image- and video-processing algorithms. Unfortunately, crowdsourced studies don't allow researchers to control content-viewing conditions, as the study participants are viewing content in their homes using various devices. Recruiting numerous participants for a crowdsourced study, however, is much easier than bringing the same number of people to the laboratory. Thus, noise introduced by poorly controlled viewing conditions in a crowdsourced study can be reduced by increasing the number of participants.

Here we show that the results of our study conducted using Subjectify.us are close to the results obtained in the laboratory. For this purpose we used Subjectify.us to replicate the user study that Netflix conducted in a controlled laboratory environment, then we compared the results. Furthermore, our comparison shows that the results obtained using Subjectify.us have significantly higher correlation with the laboratory results than scores computed using objective metrics (e.g., PSNR, SSIM and VMAF, which Netflix recently proposed).

C.1. Data Set

The data set Netflix employed in its subjective study contains videos of various types (e.g., animation, fast motion and landscape footage) compressed using the H.264 encoder at various bitrates and resolutions. Comparison of such content is challenging for a crowdsourced study, since poor viewing conditions can make barely noticeable differences in high-bitrate videos entirely invisible.

The public portion of the Netflix data set consists of nine test video sequences, each containing 6–10 distorted videos as well as 1 original undistorted video. For our study we randomly selected seven of these nine sequences and uploaded both the distorted and undistorted files to the Subjectify.us platform.

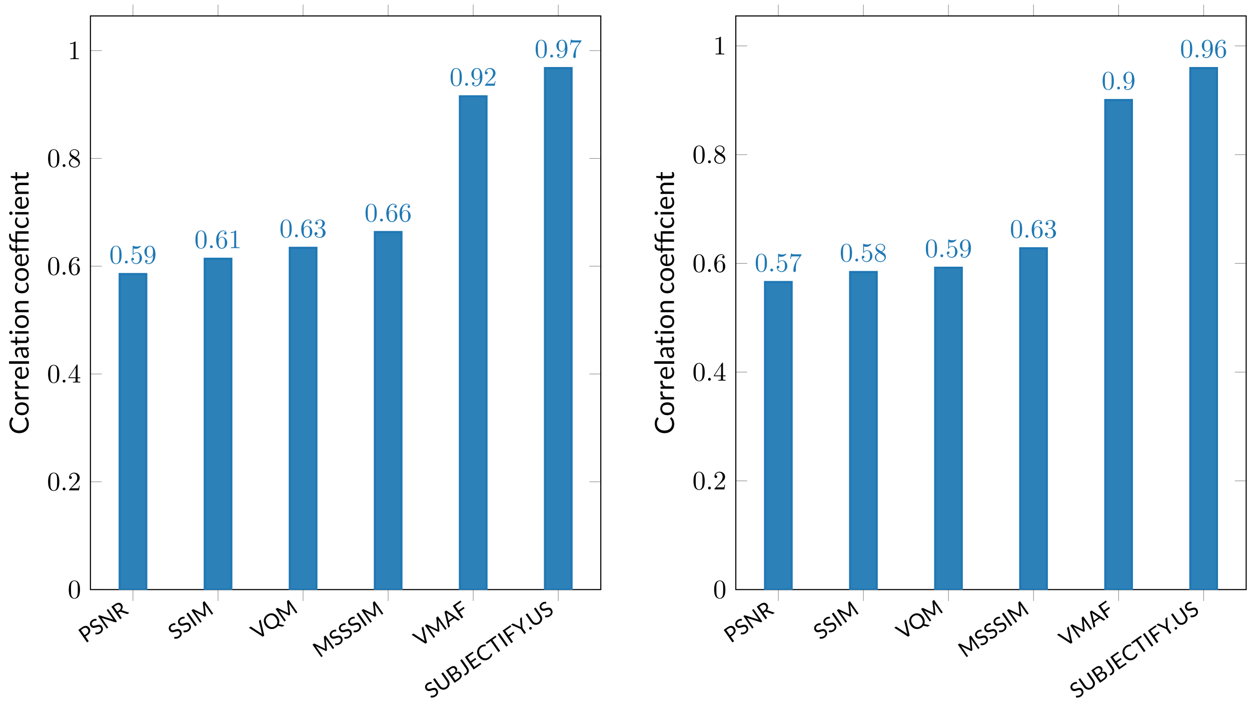
C.2. Perceptual-Data Collection

The uploaded videos were shown to study participants in a pairwise fashion. The videos were displayed in full-screen mode one after another. After each pair, participants were asked to choose the video with the best visual quality or indicate that the quality is equal. They also had an option to replay the videos.

Each study participant compared 10 video pairs including 2 hidden quality-control pairs, which compared the original undistorted video and a video compressed at 375 kbps. The responses of participants who failed one or more quality-control comparisons were rejected. Participants were allowed to complete the process up to five times. In total we collected 11,235 responses from 375 unique participants. Subjectify.us converted these responses to subjective quality scores using the crowd Bradley-Terry [1] model.

C.3. Analysis

To evaluate the quality of the scores that Subjectify.us obtained using a crowdsourced approach, we computed the correlation between these scores and the DMOS scores that Netflix obtained by conducting its experiment



(a) Pearson correlation coefficients.

(b) Spearman correlation coefficients.

Figure 21: Correlation coefficients between DMOS scores collected in laboratory environment and scores estimated by Subjectify.us compared with correlation coefficients for objective quality metrics.

in a laboratory. As a baseline we also computed the correlation between the DMOS scores and scores estimated by the following widely used objective quality metrics:

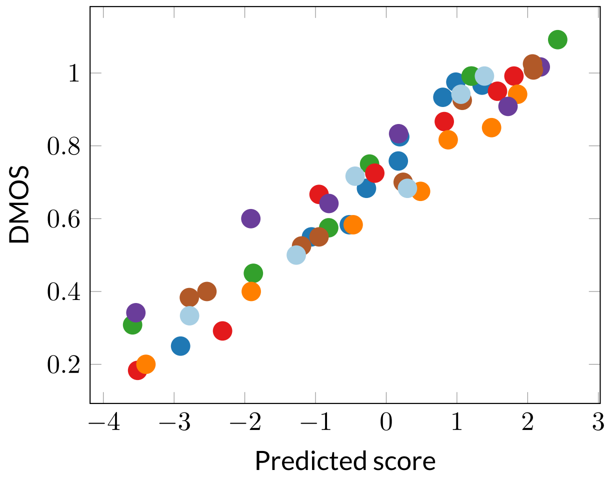
- PSNR
- SSIM [3]
- MSSSIM [2]
- VQM [4]
- VMAF, which Netflix recently proposed alongside the data set we used in this study

Figure 21 depicts the computed correlation coefficients.

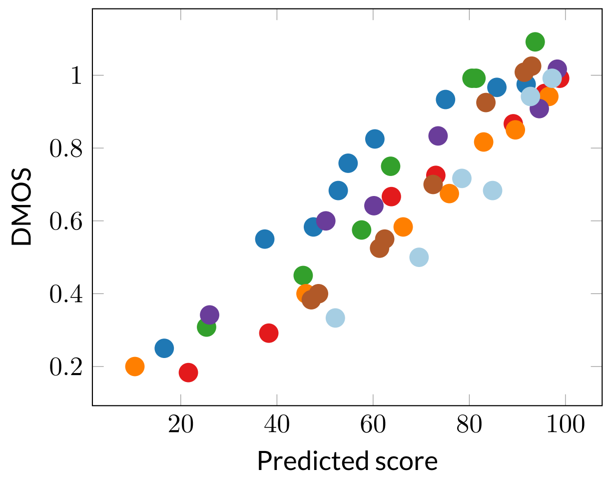
It shows that scores estimated using Subjectify.us are highly correlated with ground-truth DMOS scores according to both the Pearson (0.9614) and Spearman (0.9567) coefficients. Moreover, these correlation coefficients are significantly higher than those achieved by objective quality metrics. Notably, Netflix designed VMAF using the data set we employed in our study; this metric may therefore be overtrained for this data. In our experiment, however, Subjectify.us garnered higher correlation coefficients than VMAF.

To visually evaluate how well various methods can predict ground-truth DMOS scores, we show them on scatter-plots (see Figure 22).

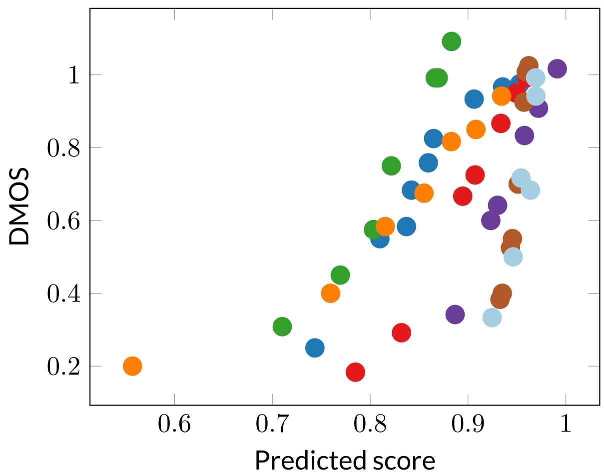
Finally, we evaluated the relationship between the number of collected responses and the correlation of estimated scores with ground-truth DMOS scores. Figure 23 shows this relationship, along with the correlation coefficients for the VMAF, MSSSIM and SSIM quality metrics.



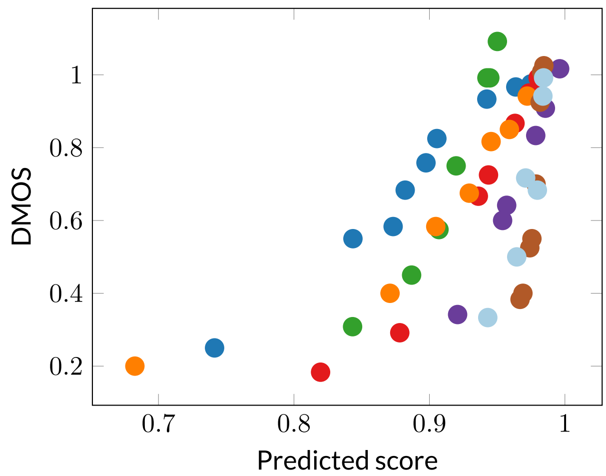
(a) SUBJECTIFY.US.



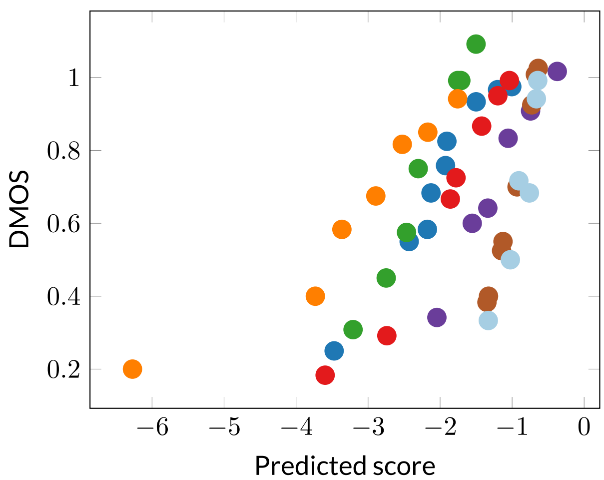
(b) VMAF.



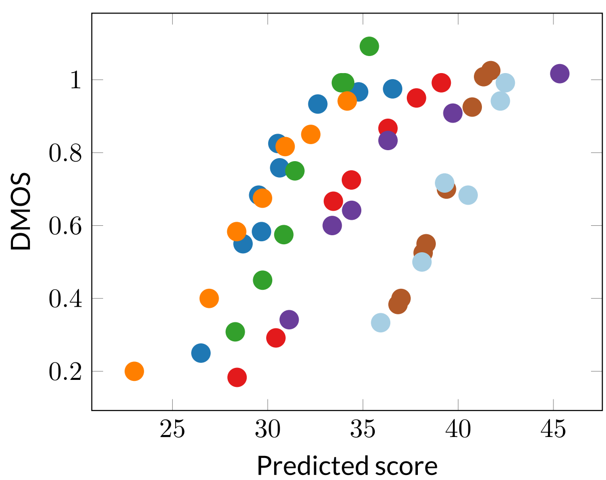
(c) SSIM.



(d) MSSSIM.

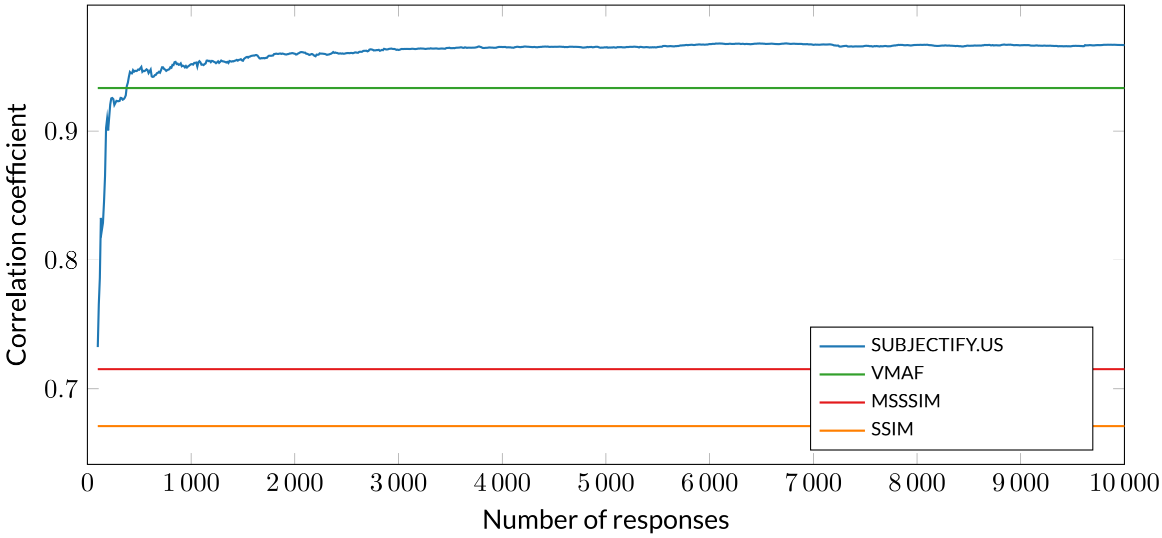


(e) VQM.

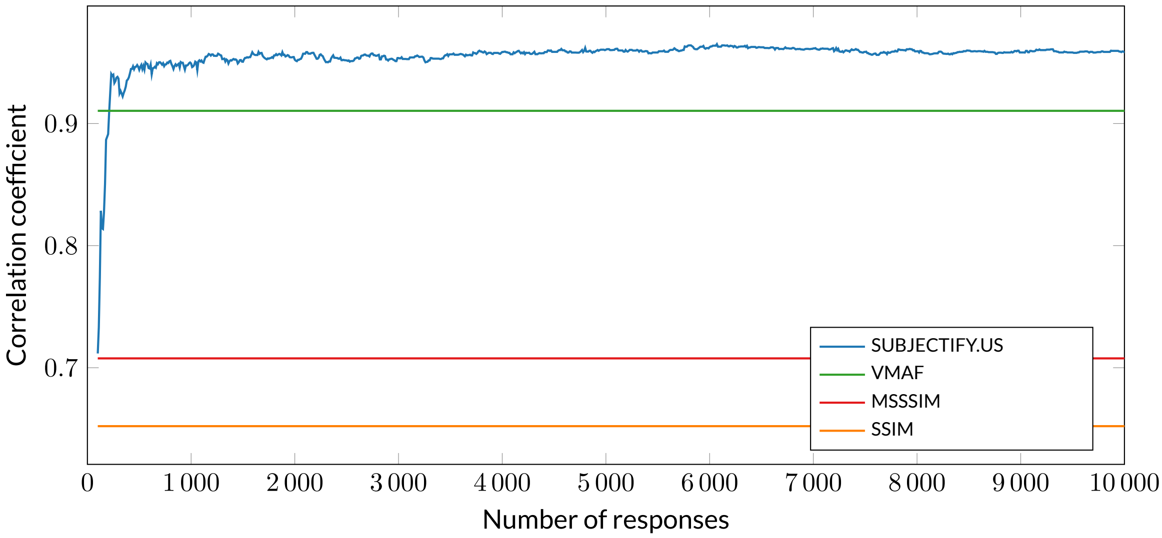


(f) PSNR.

Figure 22: DMOS scores estimated in laboratory environment vs. scores predicted by Subjectify.us and well-known objective quality metrics. Points corresponding to one source video sequence have unique color.



(a) Pearson correlation.



(b) Spearman correlation.

Figure 23: Correlation coefficients between ground-truth DMOS scores and scores computed by Subjectify.us for various number of participants' responses. The baselines indicate correlation coefficients gained by objective quality metrics: VMAF, MSSSIM, and SSIM.

As expected, the correlation coefficients increase with the number of collected responses. Subjectify.us outperforms MSSSIM, SSIM, VQM and PSNR almost immediately. It outperforms VMAF when using 300 or more responses.

C.4. Conclusion

In this auxiliary study we show that quality scores computed using the Subjectify.us crowdsourcing platform are better for comparing compressed videos than scores estimated using objective quality metrics. The reason is that Subjectify.us scores have a higher correlation with the results of experiments conducted in the laboratory. Moreover, correlation coefficients between the Subjectify.us and ground-truth scores have high absolute values (0.9614 for Pearson and 0.9567 for Spearman) in our study, indicating they are close to the laboratory results. These results justify the use of subjective crowdsourced online studies for video-codec comparisons.

D. ANALYSIS OF "TUNE SSIM" OPTION

Since the x264 and x265 command-line arguments from the "Ripping" use case in *HEVC/H.265 Video Codecs Comparison 2017* employ the `--tune ssim` option, which is designed to improve SSIM scores while sacrificing the visual quality that humans perceive, we performed an additional study to determine whether these options should be disabled for our subjective comparison.

We used the exact same method as the main study, but we compared just four options:

1. x264 encoder using the command-line arguments from the "Ripping" use case, depicted as "x264 (tune SSIM)" in the plots below.
2. x264 encoder using the same command-line arguments but with the `--tune ssim` option disabled, depicted as "x264" in the plots below.
3. x265 encoder using the command-line arguments from the "Ripping" use case, depicted as "x265 (tune SSIM)" in the plots below.
4. x265 encoder using the same command-line arguments but with the `--tune ssim` option disabled, depicted as "x265" in the plots below.

To compute subjective scores for this auxiliary study, we collected 3,280 pairwise judgments from 219 unique participants.

D.1. RD Curves

According to the RD curves shown below, the subjective quality score earned by the codec with the `--tune ssim` option enabled is always higher than the score of the same codec with this option disabled.

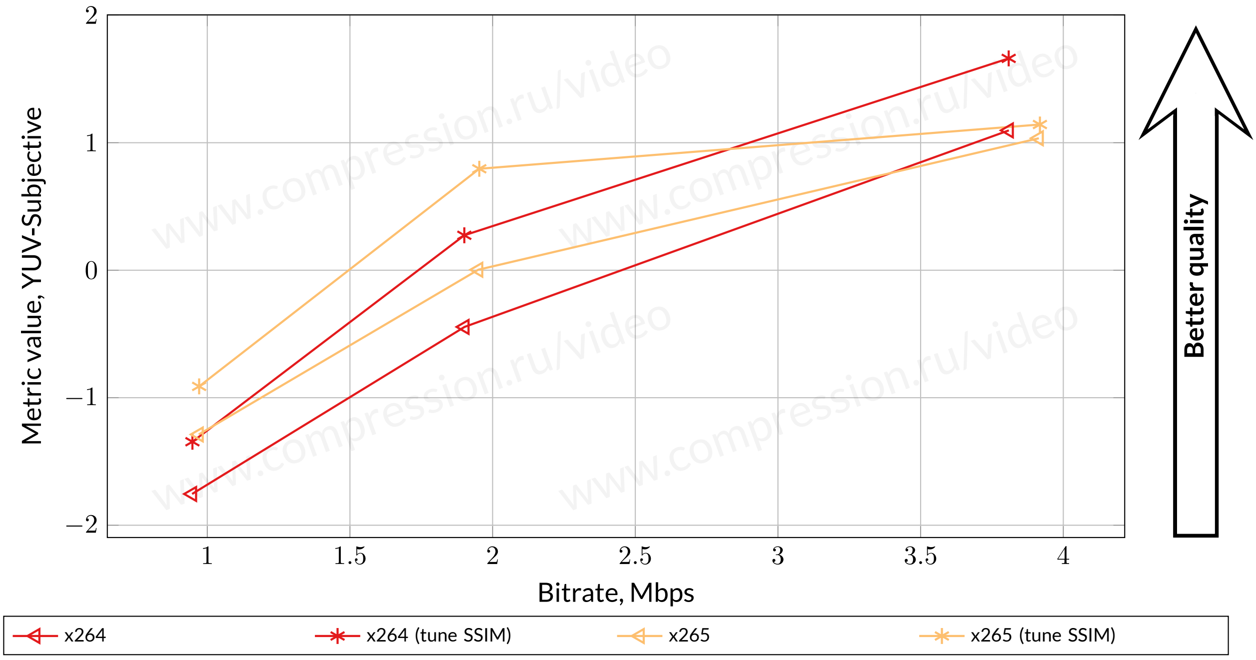


Figure 24: Bitrate/quality—use case "Ripping," *Fountain* sequence, subjective quality scores.

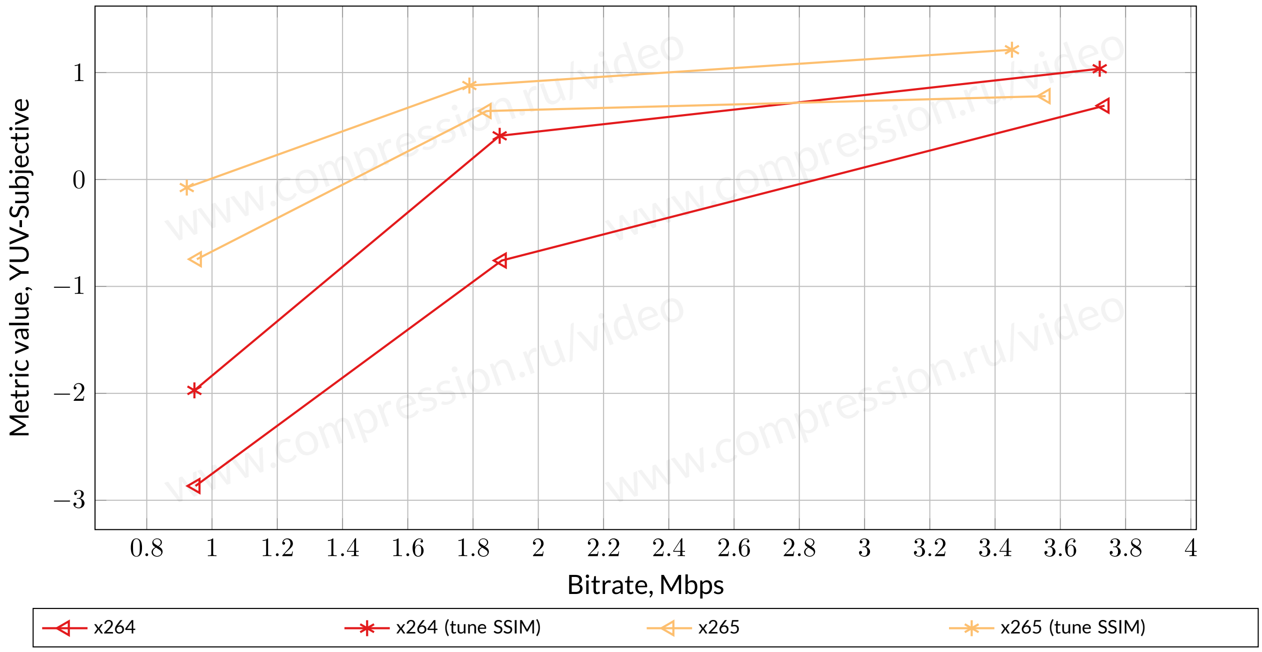


Figure 25: Bitrate/quality—use case "Ripping," *Mountain Bike* sequence, subjective quality scores.

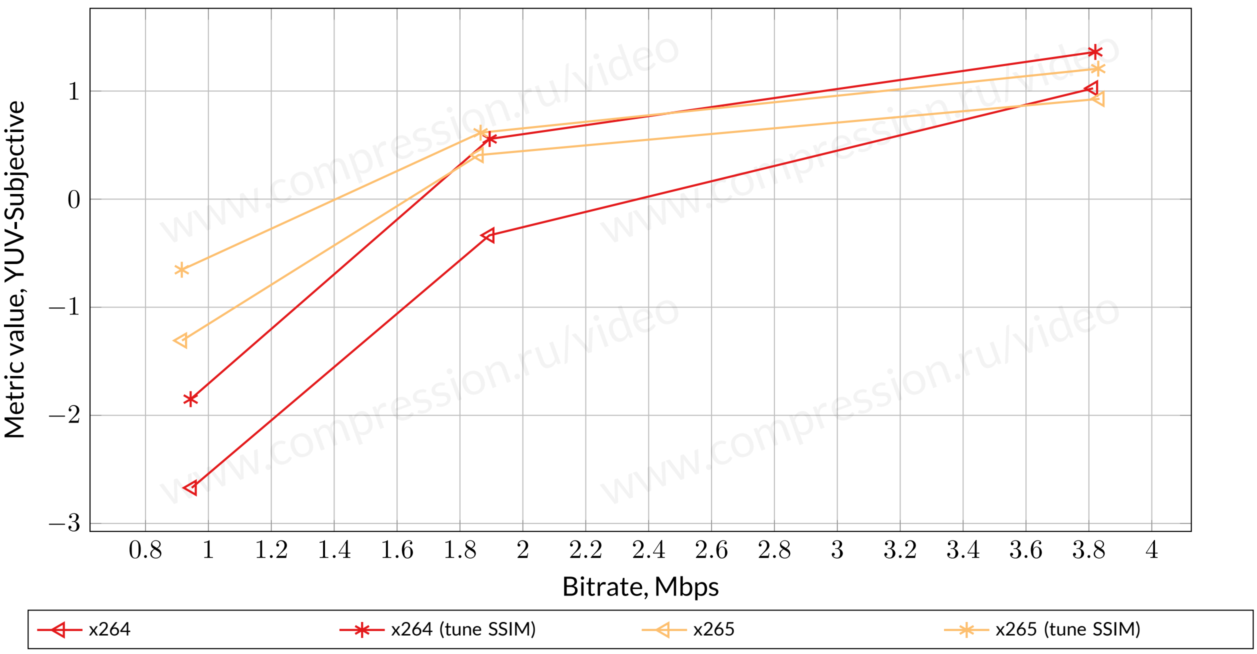


Figure 26: Bitrate/quality—use case "Ripping," *Wedding* sequence, subjective quality scores.

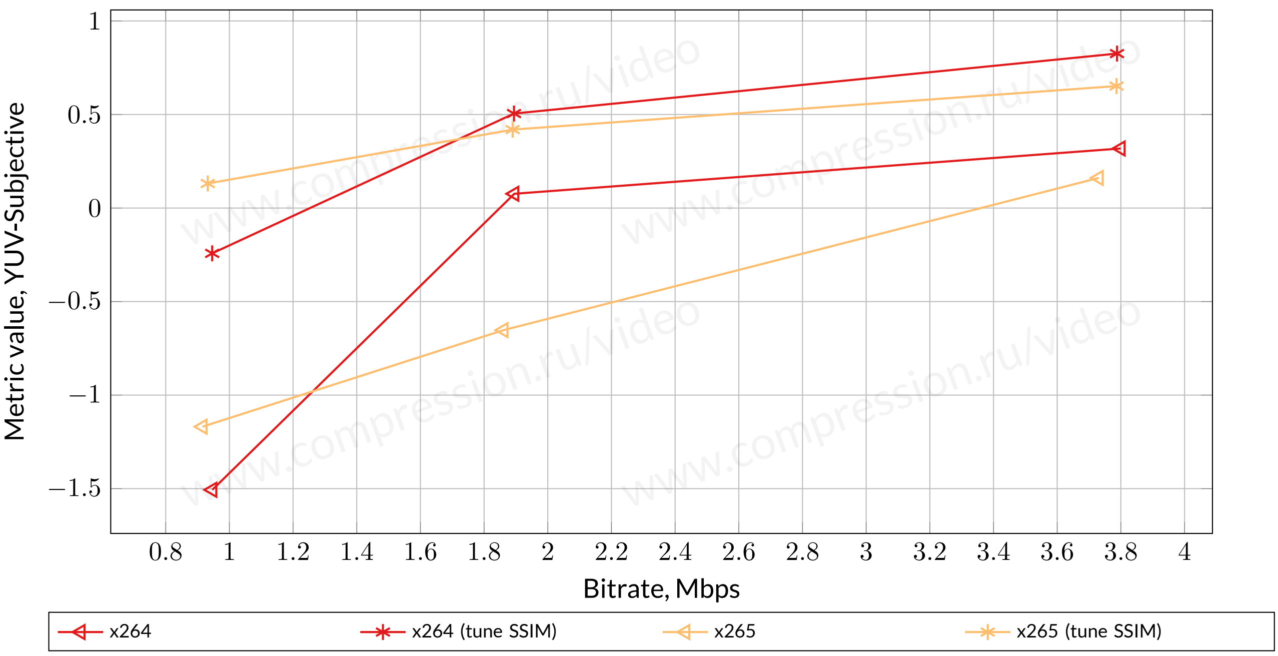


Figure 27: Bitrate/quality—use case "Ripping," *Ziguinchor* sequence, subjective quality scores.

D.2. Speed/Quality Trade-off

Furthermore, according to speed/quality trade-off plots, the encoder with `--tune ssim` enabled outperforms its sibling with `--tune ssim` disabled not only in quality but also in speed.

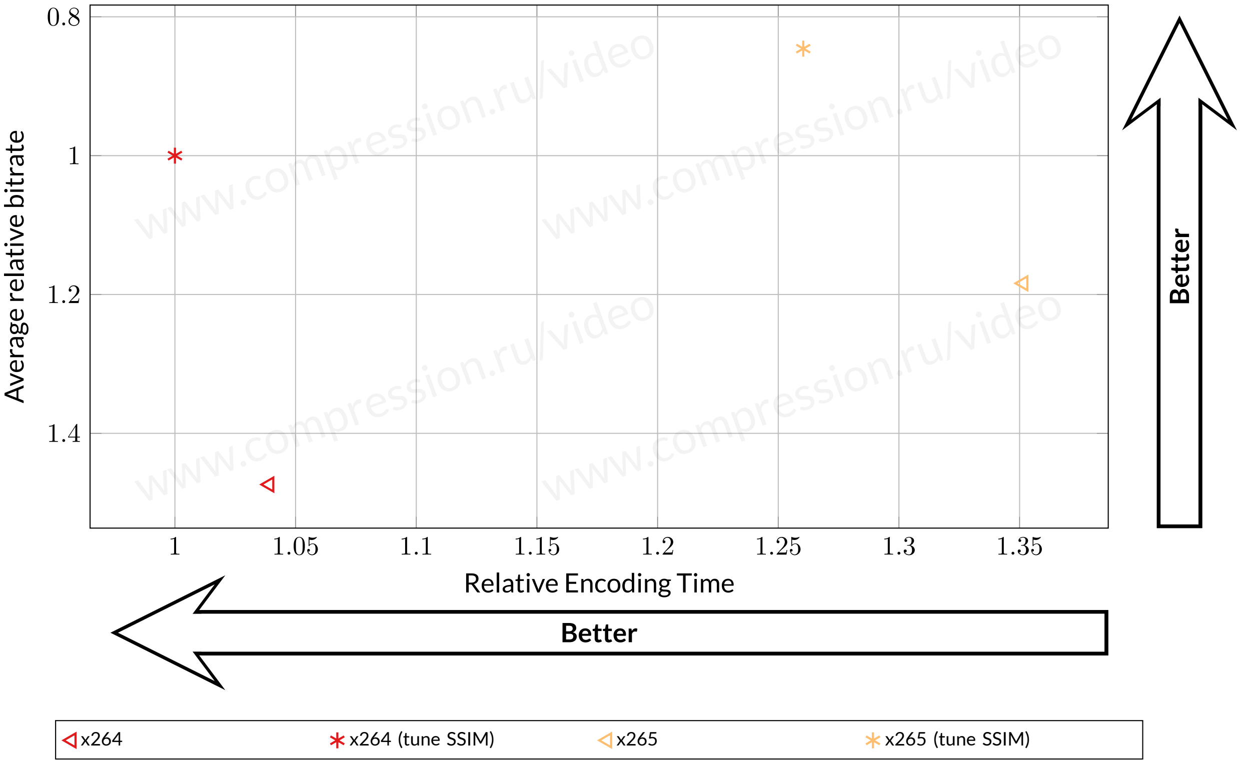


Figure 28: Speed/quality trade-off—use case "Ripping," all sequences, subjective quality scores.

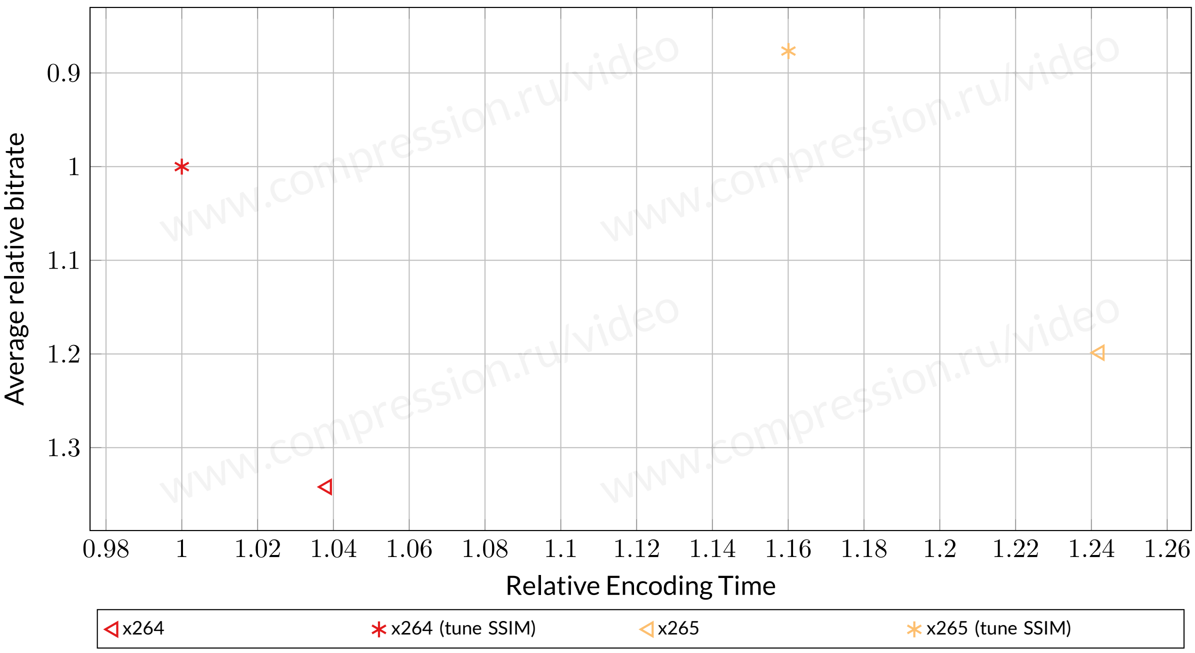


Figure 29: Speed/quality trade-off—use case "Ripping," *Fountain* sequence, subjective quality scores.

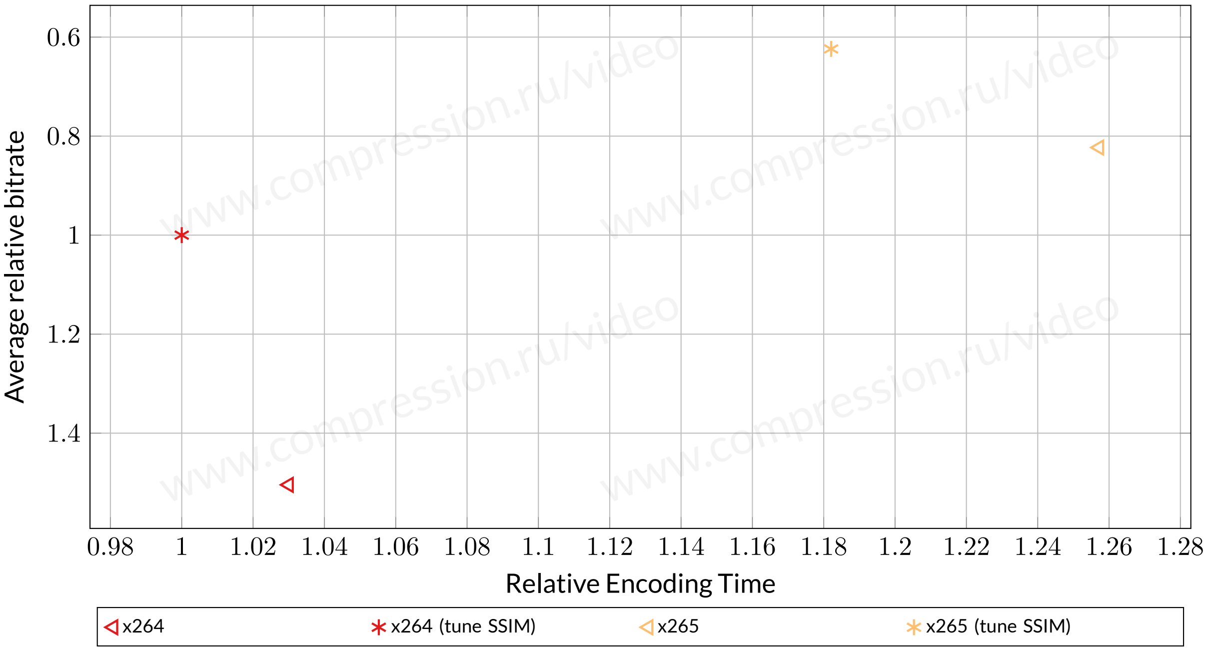


Figure 30: Speed/quality trade-off—use case “Ripping,” *Mountain Bike* sequence, subjective quality scores.

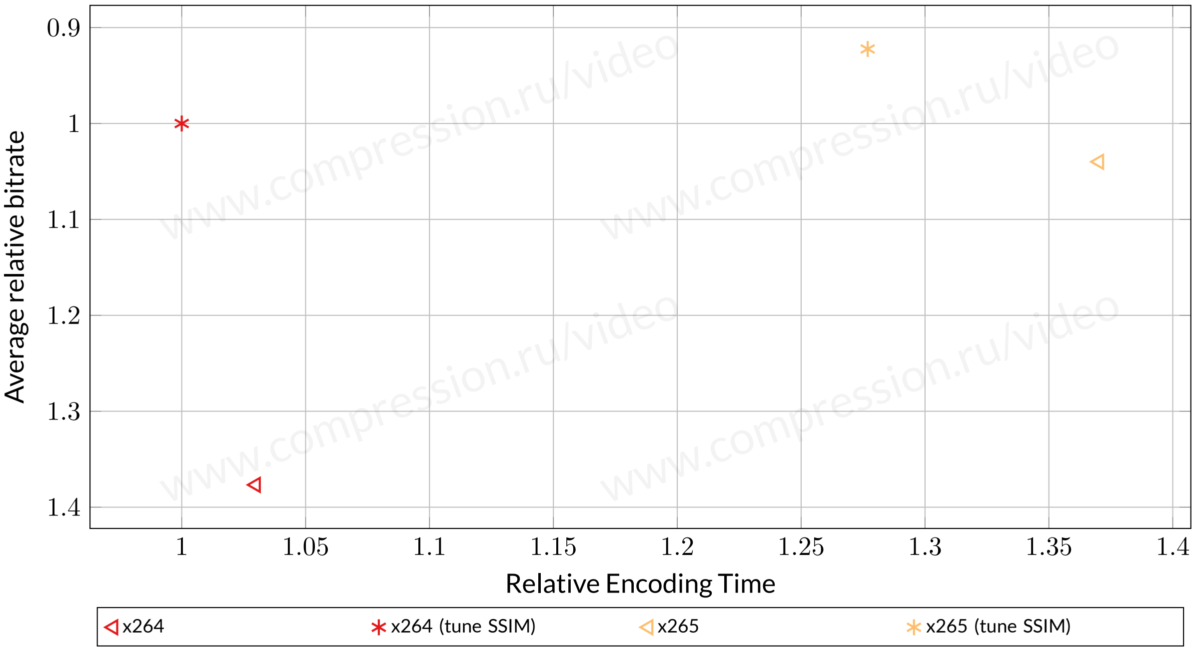


Figure 31: Speed/quality trade-off—use case “Ripping,” *Wedding* sequence, subjective quality scores.

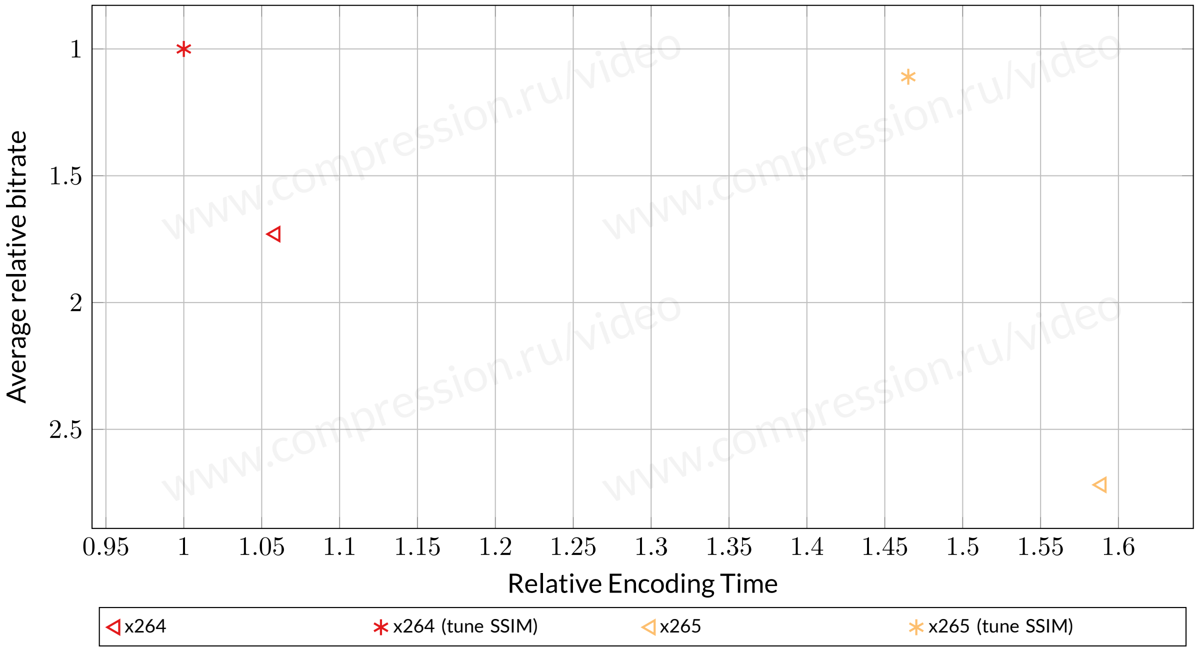


Figure 32: Speed/quality trade-off—use case "Ripping," *Ziguinchor* sequence, subjective quality scores.

D.3. Conclusion

Finally, in Figure 33 we show overall subjective quality scores computed relative to the "x264 (tune ssim)" option. The plot reveals that codecs with the `--tune ssim` option enabled outperform their siblings with this option disabled by a huge margin.

We believe this counterintuitive observation can be explained by our study conditions (see Section 2.1) and, particularly, our bitrate range (1–4 Mbps). We believe the psychovisual optimizations that are disabled by the `--tune ssim` option aren't meant for use at these relatively low bitrates. This explanation agrees with the comments we received from x264 developers:

> If you wanted to check psychovisual optimizations of x264 and especially psy-rd, then IMHO 1–4 Mbps is a very low bitrate for Full HD video encoding with it. At low bitrates it tends to produce ringing/blocking artifacts, which lower subjective quality. So, psy-rd is supposed to be used only with high-bitrate encodes, where it improves sharpness and ringing artifacts aren't visible.
>
> Also `--tune ssim` changes `--aq-mode` from 1 to 2. And `--aq-mode 2` needs less tweaking for the source owing to its auto-strength component, while `--aq-mode 1` may need `--aq-strength` tweaking for the source. When tweaked correctly it can produce higher quality than `--aq-mode 2`, but this may need per-source tweaking.
>
> *x264 developers*

But the results of this auxiliary study justify keeping the `--tune ssim` option enabled in the main study that this report describes.

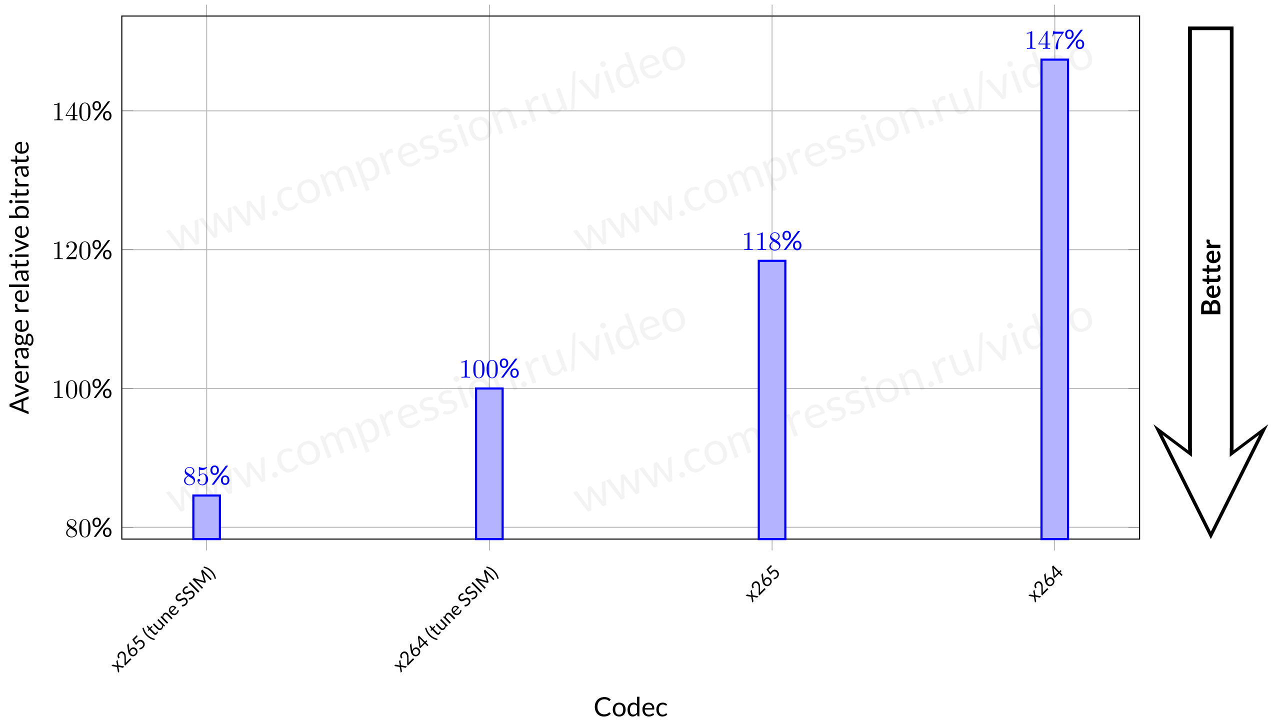


Figure 33: Average bitrate ratio for a fixed quality—use case "Ripping," all sequences, subjective quality scores.

E. CODECS

E.1. SIF-1

Encoder title	SIF
Version	1.43.0
Developed by	SIF Encoder Team

Preset name	Encoder parameters
Ripping	`ConsoleEnc.exe %SOURCE_FILE% --fps=%FPS_TMP%/1000 --comp_mode=vbr_all_p --sub_me_mode=fastest --out_bitrt=%BITRATE_KBPS% --rc_buf_s=2 --entropy_mode=8_threads -w %WIDTH% -h %HEIGHT% -o %TARGET_FILE%.avi`

E.2. x264

Encoder title	x264
Version	r2833 df79067
Developed by	x264 Developer Team

Preset name	Encoder parameters
Ripping	`x264 --preset placebo --me umh --merange 32 --keyint infinite --tune ssim --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o NUL` `x264 --preset placebo --me umh --merange 32 --keyint infinite --tune ssim --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o %TARGET_FILE%`

E.3. x265

Encoder title	x265
Version	2.3+23-97435a0870befe35
Developed by	x265 Developer Team

Preset name	Encoder parameters
Ripping	`x265_64-8bit[gcc] -p veryslow --tune ssim --bitrate %BITRATE_KBPS% --ssim %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`

E.4. nj264

Encoder title	nj264
Version	V1.0
Developed by	Nanjing Yunyan

The encoder is recipient of the Frost & Sullivan 2016 Global Enabling Technology Leadership of the Year Award for AVC Video Encoding.

Preset name	Encoder parameters
Ripping	`nj264.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj264 -preset ripping -nj264-params bitrate=%BITRATE_KBPS% -f h264 -y %TARGET_FILE%`

E.5. nj265

Encoder title	nj265
Version	V1.0
Developed by	Nanjing Yunyan

Preset name	Encoder parameters
Ripping	`nj265.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj265 -preset ripping -nj265-params bitrate=%BITRATE_KBPS% -f hevc -y %TARGET_FILE%`

E.6. KS265

Encoder title	Kingsoft Encoder
Version	V2.5.2
Developed by	Kingsoft

Preset name	Encoder parameters
Ripping	`AppEncoder_x64.exe -i %SOURCE_FILE% -preset placebo -threads 0 -wdt %WIDTH% -hgt %HEIGHT% -fr %FPS% -br %BITRATE_KBPS% -b %TARGET_FILE%`

E.7. uAVS2

Encoder title	uAVS2
Version	V1.0
Developed by	Digital Media R&D Center, Peking University, Shenzhen Graduate School

Preset name	Encoder parameters
Ripping	`Ripping\utest_x64.exe -f Ripping\encoder_ra.cfg -p InputFile=%SOURCE_FILE% -p OutputFile=%TARGET_FILE% -p SourceWidth=%WIDTH% -p SourceHeight=%HEIGHT% -p FrameRate=%FPS% -p FramesToBeEncoded=%FRAMES_NUM% -p TargetBitRate=%BITRATE_KBPS%`

F. TEST VIDEO SEQUENCES

F.1. "Fountain"

Sequence title	Fountain
Resolution	1920×1080
Number of frames	516
Color space	YV12
Frames per second	25
Source	https://vimeo.com/92772980#t=0
Source resolution	4K
Bitrate	78.856 Mbps

Static camera captures people passing by in front of a fountain in a city.

Figure 34: Fountain sequence, frame 25

F.2. “Mountain Bike”

Sequence title	Mountain Bike
Resolution	1920×1080
Number of frames	1063
Color space	YV12
Frames per second	24
Source	https://vimeo.com/188799676#t=38
Source resolution	FullHD
Bitrate	71.226 Mbps

The sequence films bikers riding in the forest. Consists of quadcopter shooting, slowmotion and close-up shots.

Figure 35: Mountain Bike sequence, frame 25

F.3. “Wedding”

Sequence title	Wedding
Resolution	1920×1080
Number of frames	948
Color space	YV12
Frames per second	24
Source	https://vimeo.com/180841074#t=625
Source resolution	FullHD
Bitrate	112.827 Mbps

Outdoor shooting of a wedding. The camera changes view several times.

Figure 36: Wedding sequence, frame 25

F.4. “Ziguinchor”

Sequence title	Ziguinchor
Resolution	1920×1080
Number of frames	994
Color space	YV12
Frames per second	25
Source	https://vimeo.com/184550115#t=120
Source resolution	FullHD
Bitrate	259.92 Mbps

Indoor and outdoor shooting of people’s conversations.

Figure 37: Ziguinchor sequence, frame 25

G. ABOUT THE GRAPHICS & MEDIA LAB VIDEO GROUP

The Graphics & Media Lab Video Group is part of the Computer Science Department of 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

H. References

[1] Xi Chen et al. "Pairwise ranking aggregation in a crowdsourced setting". In: *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*. Association for Computing Machinery (ACM), 2013. DOI: 10.1145/2433396.2433420.

[2] Z. Wang, E. P. Simoncelli, and A. C. Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*. Vol. 2. Nov. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.

[3] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861.

[4] Feng Xiao et al. "DCT-based video quality evaluation". In: *Final Project for EE392J* 769 (2000).

I. LIST OF MINARY FIXES

We are sorry for mistakes and formatting defects in the release version of our report. This year we used new version of report generation system, that caused some inaccuracies passed while manual report checking. In this report version the following mistakes were corrected:

1. x265 codec version was unified and corrected in all mentions and report parts. Before this changes, some of the x265 mentions included an old (1.9+169-e5b5bdc3c154) version. This happened due to cut&paste from previous 2016 report and some mentions was passed while changing to a correct version (2.3+23-97435a0870befe35)
2. The name uAVS2 was corrected on the title page of Part 1
3. In Part 3, overlapping of x264 description was fixed (in an appendix with codecs)
4. In Part 4, text overlapping in Section 2 (with codecs descriptions) was corrected
5. List of video sequences and their descriptions were completed in Part 4
6. All screenshots from all sequences were converted to JPEG due to make the PDF file size smaller

MSU Video Quality Measurement Tool

MSU Graphics & Media Lab. Video Group.

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

1. Widest Range of Metrics & Formats

1.1 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)**

1.2 HDR support

1.3 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

1.4 2k, 4k, 8k support

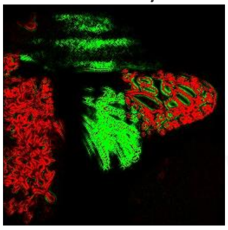
2. Fastest Video Quality Measurement

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

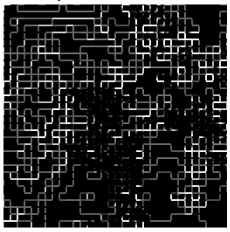
2.2 Multi-core Processors Support

Visualization Examples

Allows easily detect where codec/filter fails



MSU Blurring Metric



MSU Blocking Metric

VQMT average Speedup

3. Easy Integration

3.1 Linux support

DEB & RPM packages

3.2 Batch Processing with JSON and CSV output

3.3 Plugins SDK

4. Professional Analysis

4.1 Comparative Analysis

4.2 Metric Visualization

MSU VQMT Official Page

Tool was downloaded more than 200 000 times!

Free and Professional versions are available

Big thanks to our contributors:

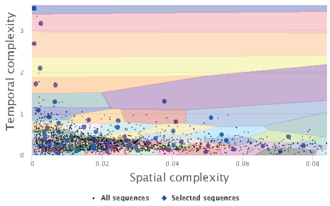
Reduce video file size or encoding speed with optimal codec settings

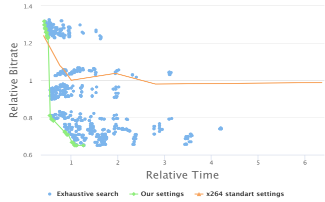
Effective Video Transcoding

For almost 14 years, Lomonosov MSU Graphics&Media Lab's video group has been conducting video codecs comparisons. We know that almost always there is a possibility to find efficient encoding options for every video

We created a representative dataset of **385 videos** chosen from **9000+ FullHD&4K** videos

12 million encoder launches were done on Intel Xeon E3-1125v3

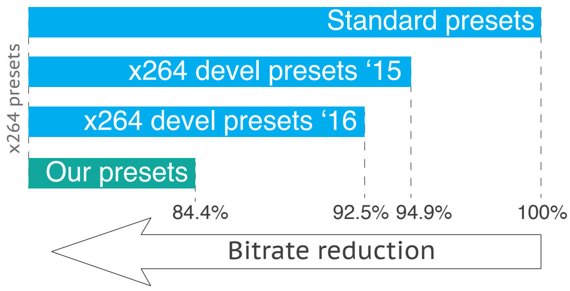




Full-size charts are available on our project page

15% bitrate savings in average

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



Percentage of file size reduction in average for a representative dataset of 77 videos

We developed a way to find optimal presets for a large number of video classes

Everything is fair! We don’t declare an “up-to-x%” bitrate reduction — average file size reduction is 15% higher comparing to standard x264 presets

We find presets that **do not reduce encoding speed and objective quality of encoded video**
You give limitations, and we guarantee the same or higher objective quality and encoding speed

You use standard presets and don’t believe that it will work for your videos? Give us a chance — request a demo, for free!

We can find best 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

Subjective comparisons
Receive subjective quality comparison results for your videos

Codec analysis
Find out strong and weak parts of your codec

Saliency-adaptive encoding
Bitrate savings given by adaptive encoding of salient regions

Gaze maps construction
Raw viewers’ gaze points on your video

Encoding with extremely low bitrates
Get your video of highest quality for low bitrates

4K and 360-degree encoding
Best presets for high-quality formats encoding

contact **evt@compression.ru** to get them!

Our project page compression.ru/video/video_codec_optimization/

In cooperation with Lomonosov MSU Graphics&Media Lab