HEVC/H.265 Video Codecs Comparison

Graphics & Media Lab
Video Group

Video group head: Dr. Dmitriy Vatolin

Project head: Dr. Dmitriy Kulikov

Measurements & analysis: Mikhail Erofeev,
Stanislav Dolganov,
Sergey Zvezdakov

Codecs:

H.265

- Chips&Media HEVC Encoder
- Intel MSS HEVC Encoder
- Kingsoft HEVC Encoder
- nj265
- SHBP H.265 Real time encoder
- x265

Non H.265

- nj264
- x264

CS MSU Graphics & Media Lab, Video Group
August 22, 2016
http://www.compression.ru/video/codec_comparison/index_en.html
videocodec-testing@graphics.cs.msu.ru

Contents

1 Acknowledgments **5**

2 Overview **6**
2.1 Sequences . . . 6
2.2 Codecs . . . 7

3 Objectives and Testing Rules **8**

4 Fast Encoding **9**
4.1 RD curves . . . 9
4.2 Encoding Speed . . . 10
4.3 Speed/Quality Trade-Off . . . 12
4.4 Bitrate Handling . . . 14
4.5 Relative Quality Analysis . . . 15

5 Universal Encoding **19**
5.1 RD curves . . . 19
5.2 Encoding Speed . . . 20
5.3 Speed/Quality Trade-Off . . . 21
5.4 Bitrate Handling . . . 23
5.5 Relative Quality Analysis . . . 25

6 Ripping Encoding **28**
6.1 RD curves . . . 28
6.2 Encoding Speed . . . 29
6.3 Speed/Quality Trade-Off . . . 30
6.4 Bitrate Handling . . . 32
6.5 Relative Quality Analysis . . . 33

7 Conclusion **35**
7.1 Fast Encoding . . . 37
7.2 Universal Encoding . . . 38
7.3 Ripping Encoding . . . 39
7.4 Overall . . . 40

A Software and Hardware Encoders Comparison **41**
A.1 RD curves . . . 41
A.2 Encoding Speed . . . 42
A.3 Speed/Quality Trade-Off . . . 44
A.4 Bitrate Handling . . . 46
A.5 Relative Quality Analysis . . . 47

B Participants' Comments **50**
B.1 Kingsoft . . . 50
B.2 x264 . . . 50

C Sequences **51**
C.1 "Apple Tree" . . . 51
C.2 "Arashiyama" . . . 52
C.3 "Bicycle" . . . 53
C.4 "Bunny" . . . 54
C.5 "Butterflies" . . . 55
C.6 "Caterpillar" . . . 56
C.7 "CG Forest" . . . 57
C.8 "Church Concert" . . . 58
C.9 "City Crowd" . . . 59
C.10 "City Views" . . . 60
C.11 "Concert" . . . 61
C.12 "Crowd Run" . . . 62
C.13 "Disneyland" . . . 63
C.14 "Fire" . . . 64
C.15 "Fishes" . . . 65
C.16 "Fountain" . . . 66
C.17 "Neighborhood" . . . 67
C.18 "Old Car" . . . 68
C.19 "Outdoor Party" . . . 69
C.20 "Park Walk" . . . 70
C.21 "Seacoast" . . . 71
C.22 "Shakewalk" . . . 72
C.23 "Ships" . . . 73
C.24 "Steadicam" . . . 74
C.25 "Sunset" . . . 75
C.26 "Water" . . . 76
C.27 "Winter" . . . 77

D Sequences selection **78**

E Codecs **81**
E.1 Intel® Media Server Studio HEVC GPU-accelerated Encoder . . . 81
E.2 SHBP H.265 Real time encoder . . . 82
E.3 x264 . . . 83
E.4 x265 . . . 84
E.5 Chips&Media Hevc Encoder . . . 85
E.6 nj264 . . . 85
E.7 nj265 . . . 86
E.8 KS265 . . . 86

F Figures Explanation **87**
F.1 RD Curves 87
F.2 Relative Bitrate/Relative Time Charts 87
F.3 Graph Example 87
F.4 Bitrates Ratio with the Same Quality 87
F.5 Relative Quality Analysis 89

G Objective Quality Metrics Description **91**
G.1 SSIM (Structural SIMilarity) 91
G.1.1 Brief Description 91
G.1.2 Examples 92

H About the Graphics & Media Lab Video Group **95**

1 ACKNOWLEDGMENTS

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

- Intel Corporation
- ”System house ”Business partners” company
- x264 developer team
- MulticoreWare, Inc.
- Nanjing Yunyan
- Kingsoft
- Chips&Media Inc.

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

2 OVERVIEW

2.1 Sequences

	Sequence	Number of frames	Frame rate	Resolution
1.	Apple Tree	338	30	1920 × 1080
2.	Arashiyama	1029	24	1920 × 1080
3.	Bicycle	780	25	1920 × 1080
4.	Bunny	600	24	1920 × 1080
5.	Butterflies	1076	30	1920 × 1080
6.	Caterpillar	113	30	1920 × 1080
7.	CG Forest	1184	30	1920 × 1080
8.	Church Concert	1095	24	1920 × 1080
9.	City Crowd	763	30	1920 × 1080
10.	City Views	1025	24	1920 × 1080
11.	Concert	1533	25	1920 × 1080
12.	Crowd Run	500	25	1920 × 1080
13.	Disneyland	317	24	1920 × 1080
14.	Fire	601	25	1920 × 1080
15.	Fishes	1025	25	1920 × 1080
16.	Fountain	516	25	1920 × 1080
17.	Neighborhood	360	24	1920 × 1080
18.	Old Car	1125	24	1920 × 1080
19.	Outdoor Party	1513	30	1920 × 1080
20.	Park Walk	914	24	1920 × 1080
21.	Seacoast	1075	30	1920 × 1080
22.	Shakewalk	805	25	1920 × 1080
23.	Ships	1041	25	1920 × 1080
24.	Steadicam	979	24	1920 × 1080
25.	Sunset	1190	24	1920 × 1080
26.	Water	1209	25	1920 × 1080
27.	Winter	997	24	1920 × 1080

TABLE 1: Summary of video sequences

Brief descriptions of the sequences used in our comparison are given in Table 1. More detailed descriptions of these sequences can be found in Appendix C.

2.2 Codecs

	Codec	Developer	Version
1.	Chips&Media HEVC Encoder (speed unverified)	Chips&Media, Inc.	1.3 (r5650)
2.	Intel MSS HEVC Encoder	Intel	Intel Media Server Studio 2017 R1
3.	Kingsoft HEVC Encoder	Kingsoft	Kingsoft Encoder V2.1.1
4.	nj264	Nanjing Yunyan Email: jtwen@tsinghua.edu.cn	1.0
5.	nj265	Nanjing Yunyan Email: jtwen@tsinghua.edu.cn	1.0
6.	SHBP H.265 Real time encoder	SHBP Codec's development team Email: lobasso@hotmail.com	1.0
7.	x264	x264 Developer Team	148 r2665 a01e399
8.	x265	MulticoreWare, Inc.	1.9+169-e5b5bdc3c154

TABLE 2: Short codecs' descriptions

Brief descriptions of the codecs used in our comparison are given in Table 2. x264 was used as a good quality AVC reference codec for comparison purposes. Detailed descriptions of all codecs used in our comparison can be found in Appendix E.

3 OBJECTIVES AND TESTING RULES

The main goal of this report is the presentation of a comparative evaluation of the quality of new HEVC codecs and codecs of other standards using objective measures of assessment. All test video sequences were 1080p video sequences. The comparison was done using settings provided by the developers of each codec. Nevertheless, we required all presets to satisfy minimum speed requirements. The main task of the comparison is to analyze different encoders for the task of transcoding video—e.g., compressing video for personal use.

The comparison was performed on Core i7 6700K (Skylake) @ 4Ghz, RAM 8 GB, Windows 8.1.

Set of test video sequences was selected using method described in Appendix D. More than 30000 videos were automatiucaly analyzed and clusterized and 27 of them were chosen to represent theirs clusters.

Quality of encoded video seqeunces was estimated using YUV-SSIM objective quality metric (see Appendix G.1).

4 FAST ENCODING

4.1 RD curves

Next figures show RD curves for eigtht video sequences on fast transcoding use case. On most sequences Intel's encoder and KingSoft encoders are leaders.

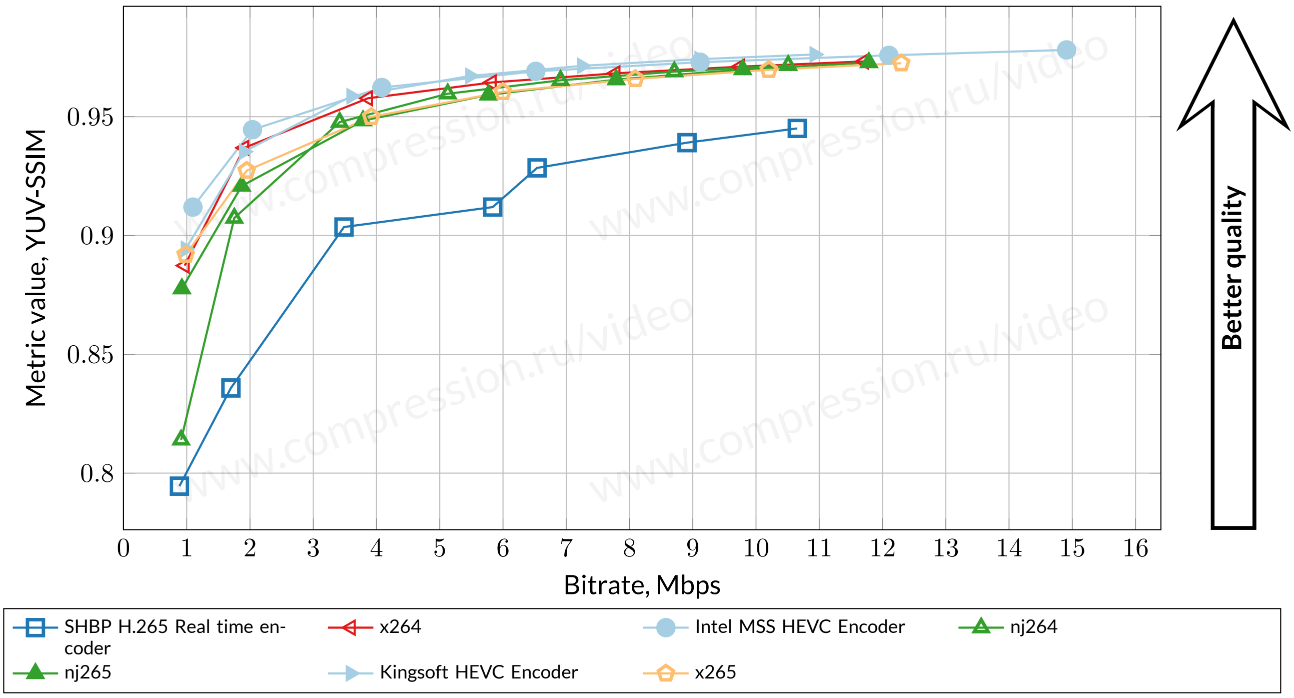


FIGURE 1: Bitrate/quality—usecase "Fast Encoding," Apple Tree sequence, YUV-SSIM metric

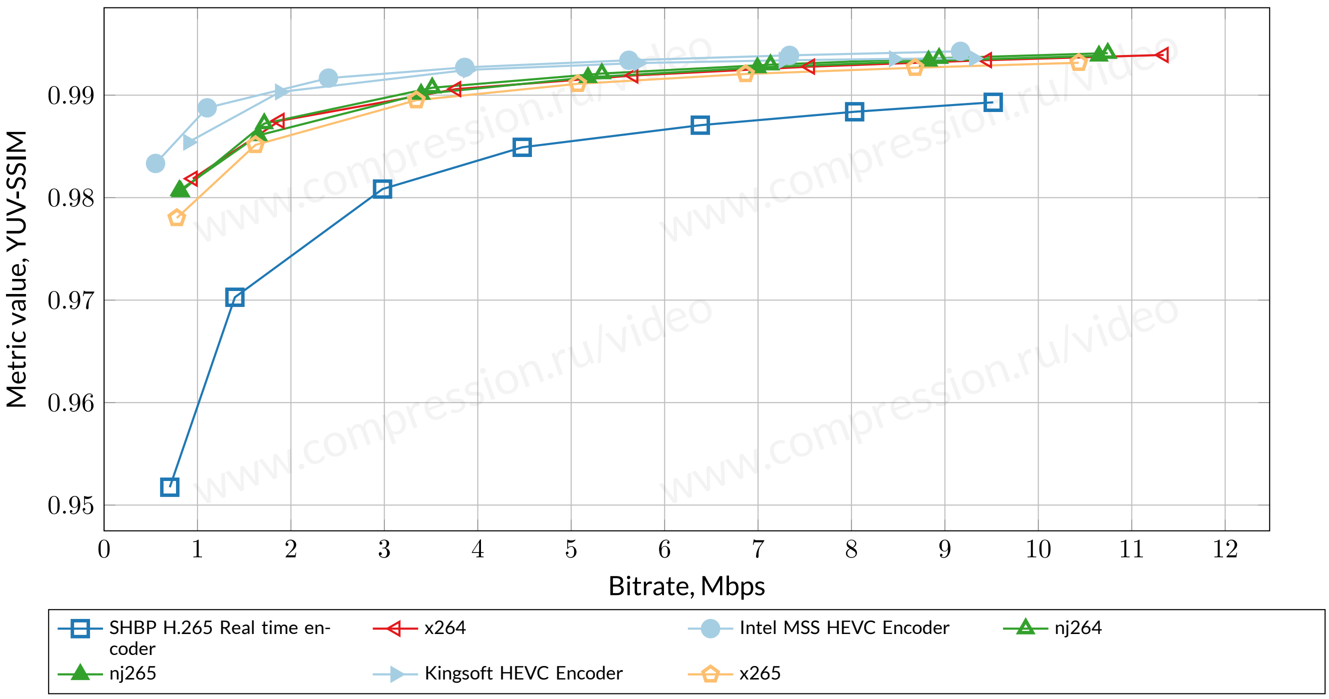


FIGURE 2: Bitrate/quality—usecase "Fast Encoding," Butterflies sequence, YUV-SSIM metric

4.2 Encoding Speed

Figures below show how participating codecs differ in encoding speed. The first places are typically taken by SHBP H.265 Real time encoder followed by Kingsoft HEVC encoder with big gap. Also on most of the plots SHBP has almost constant encoding speed. Kingsoft HEVC encoder has slightly stronger encoding speed dependency on traget bitrate than other encoders.

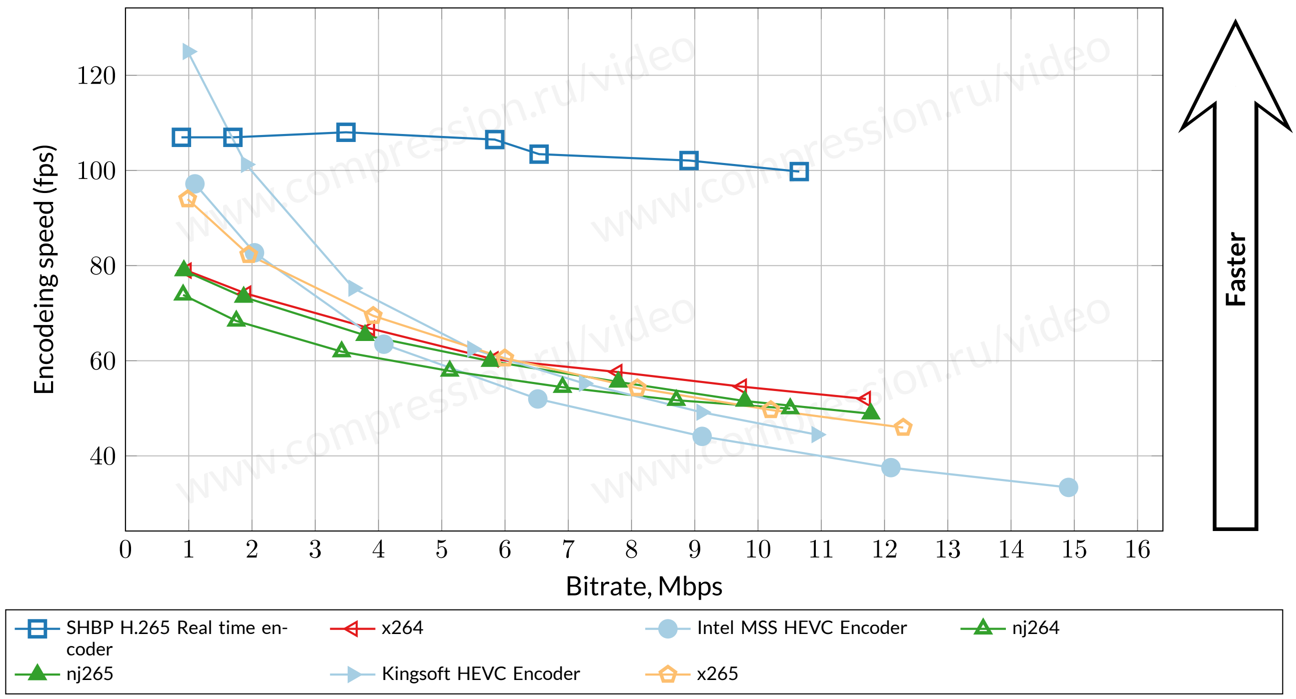


FIGURE 3: Encoding speed—usecase "Fast Encoding," Apple Tree sequence

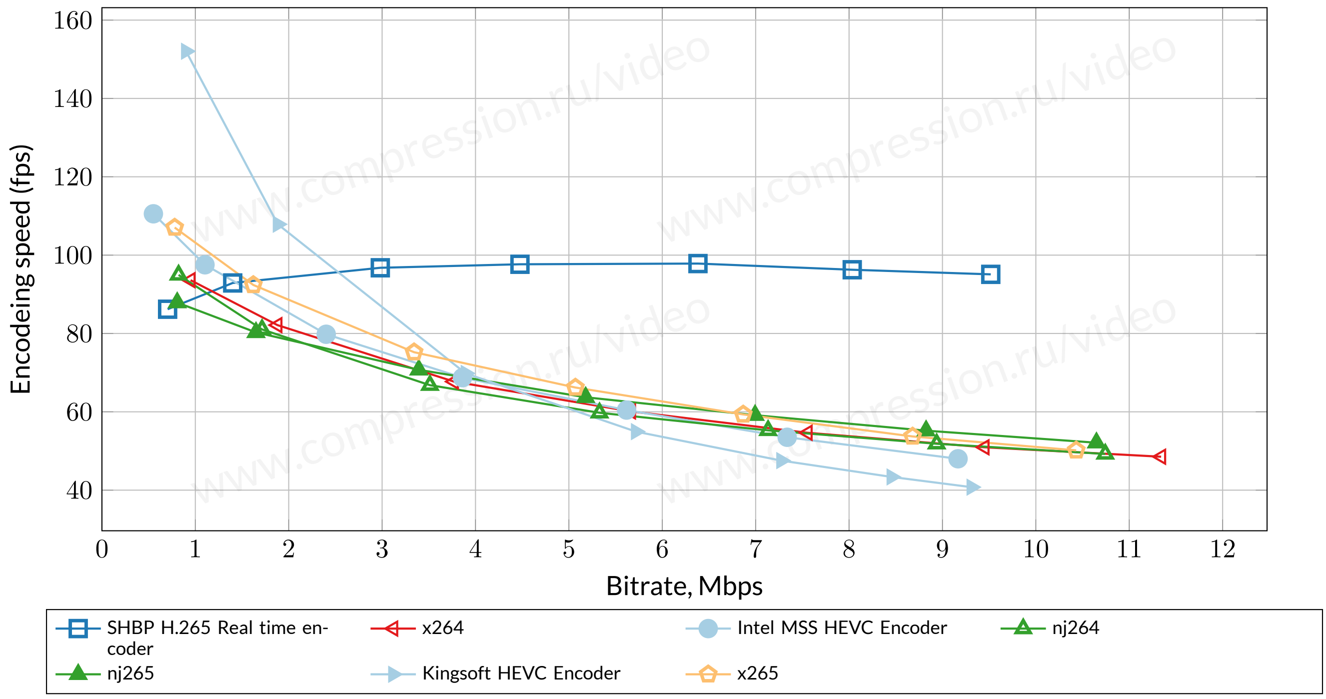


FIGURE 4: Encoding speed—usecase "Fast Encoding," Butterflies sequence

4.3 Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs can be found in Appendix F. Sometimes, codec results are not present in the particular graph owing to the codec's extremely poor performance. The codec's RD curve has no intersection with the reference's RD curve.

The speed/quality trade-off graphs simultaneously show relative quality and encoding speed for the encoders tested in this comparison. x264 is the reference codec, for which both quality and speed are normalized to unity for all of the graphs. The terms "better" and "worse" are used to compare codecs in the same manner as in previous portions of this comparison.

Pareto optimal encoders in terms speed and quality (at average) are SHBP H.265 Real time encoder, Kingsoft HEVC encoder and Intel MSS HEVC encoder. "Pareto optimal" encoder means there is no encoder faster and better then current one in this test.

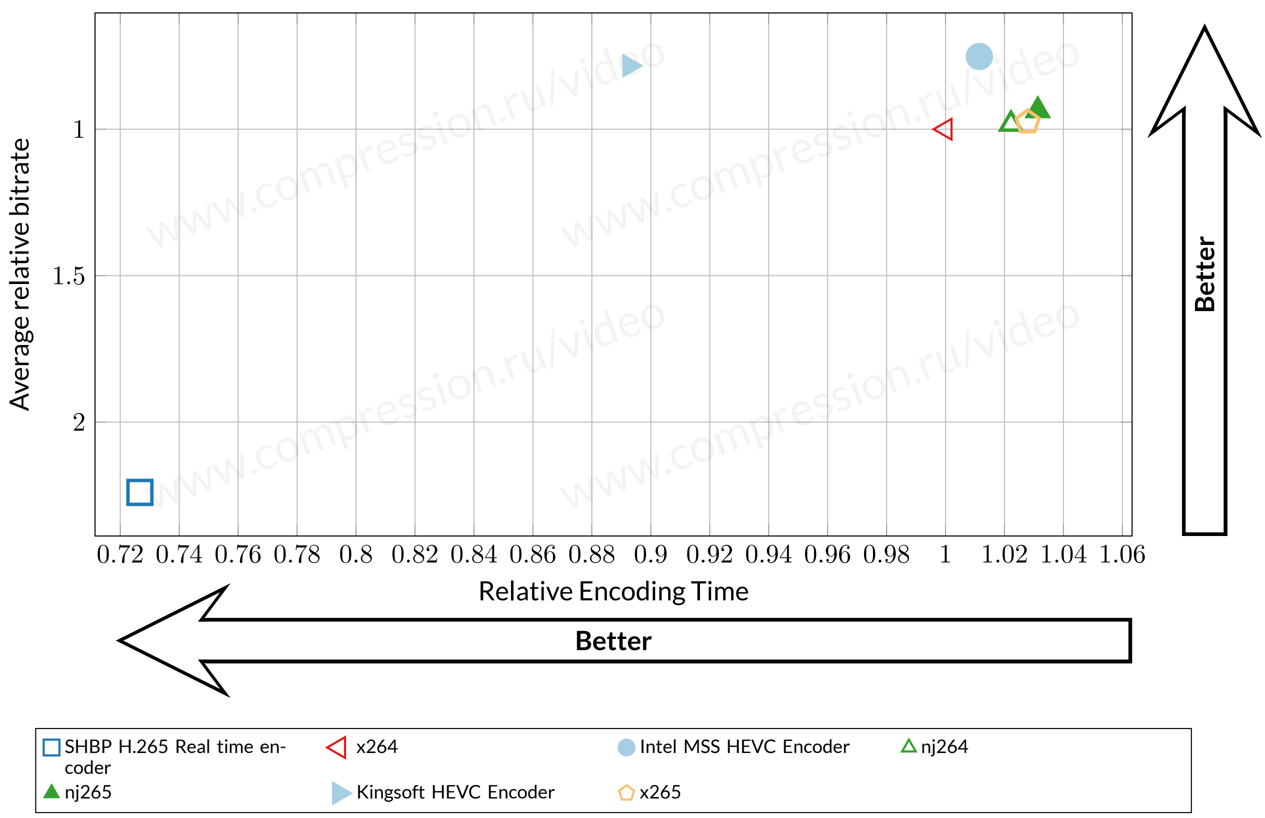


FIGURE 5: Speed/quality trade-off—usecase "Fast Encoding," all sequences, YUV-SSIM metric

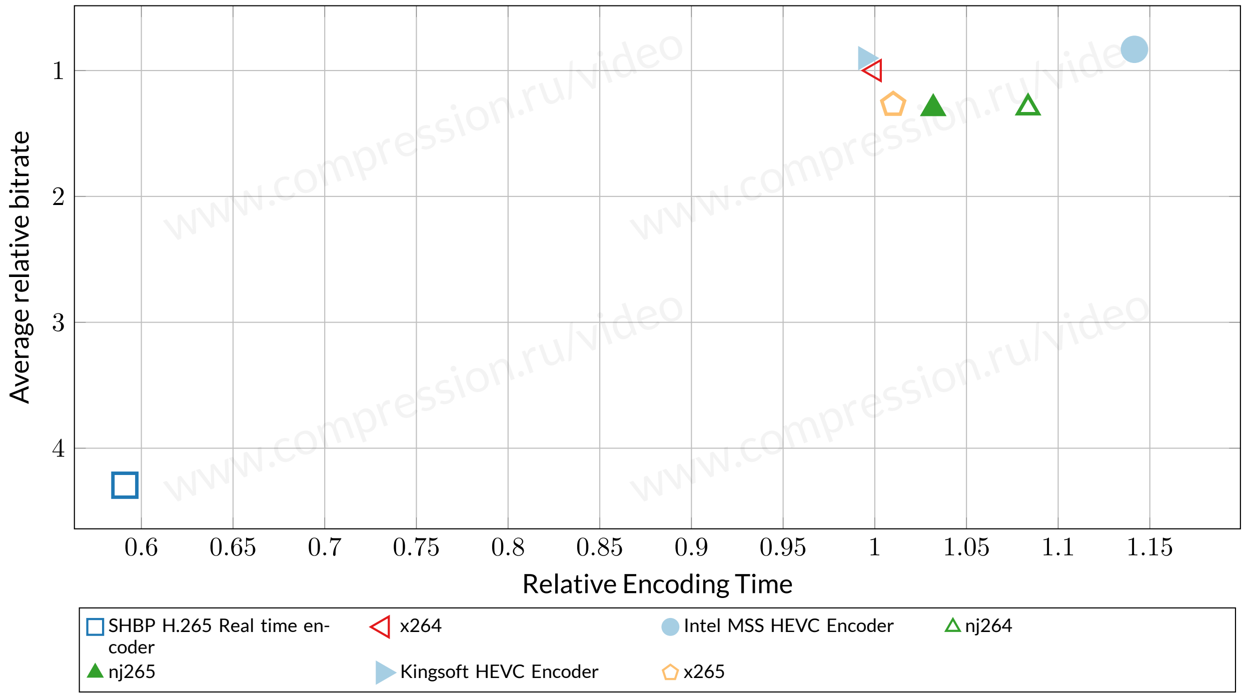


FIGURE 6: Speed/quality trade-off—usecase "Fast Encoding," Apple Tree sequence, YUV-SSIM metric

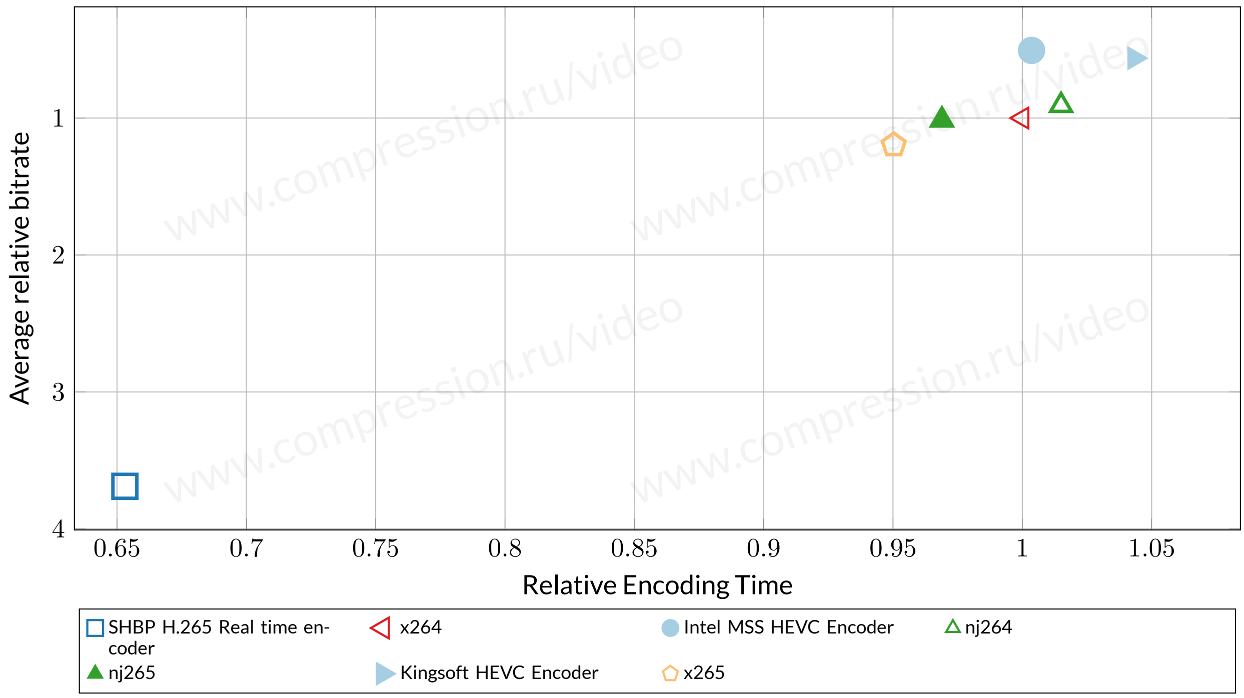


FIGURE 7: Speed/quality trade-off—usecase "Fast Encoding," Butterflies sequence, YUV-SSIM metric

4.4 Bitrate Handling

The plots below show how accurately encoded stream's real bitrate matches bitrate requested by user. Almost all encoders handle bitrate well, but there are issues for some encoders at some sequnces, e.g. nj264, SHBP H.265 Real time encoder.

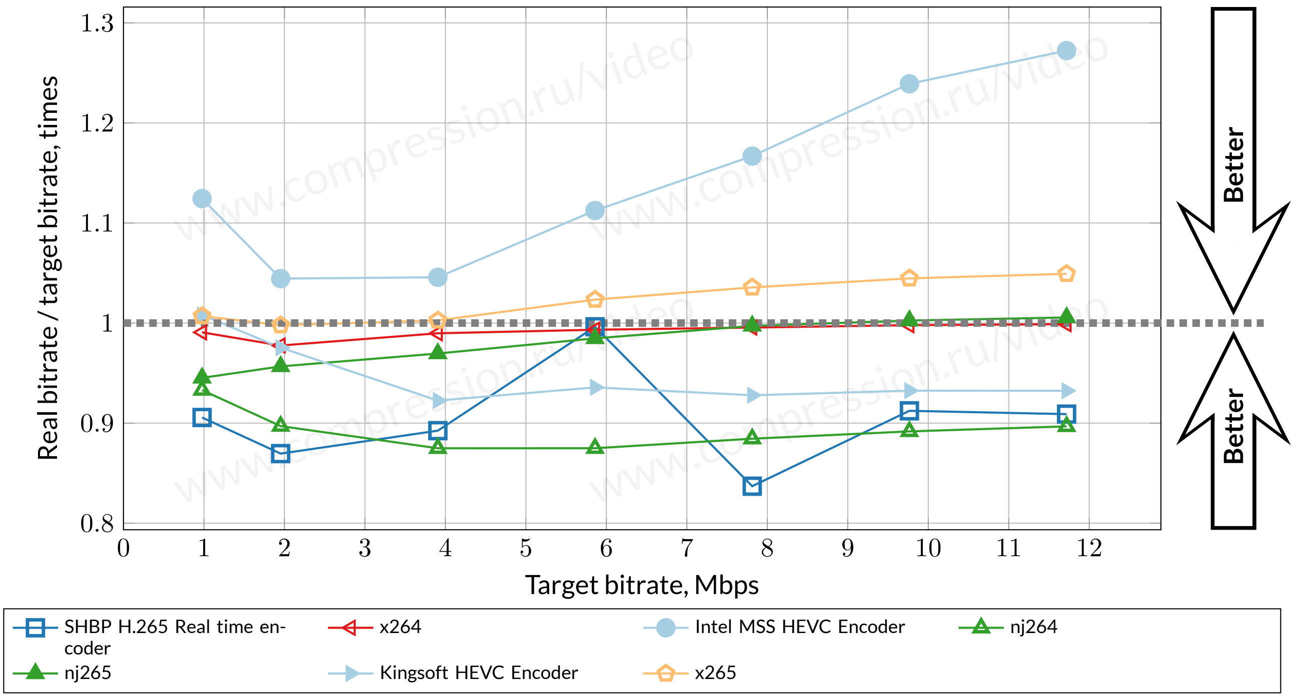


FIGURE 8: Bitrate handling—usecase "Fast Encoding," Apple Tree sequence

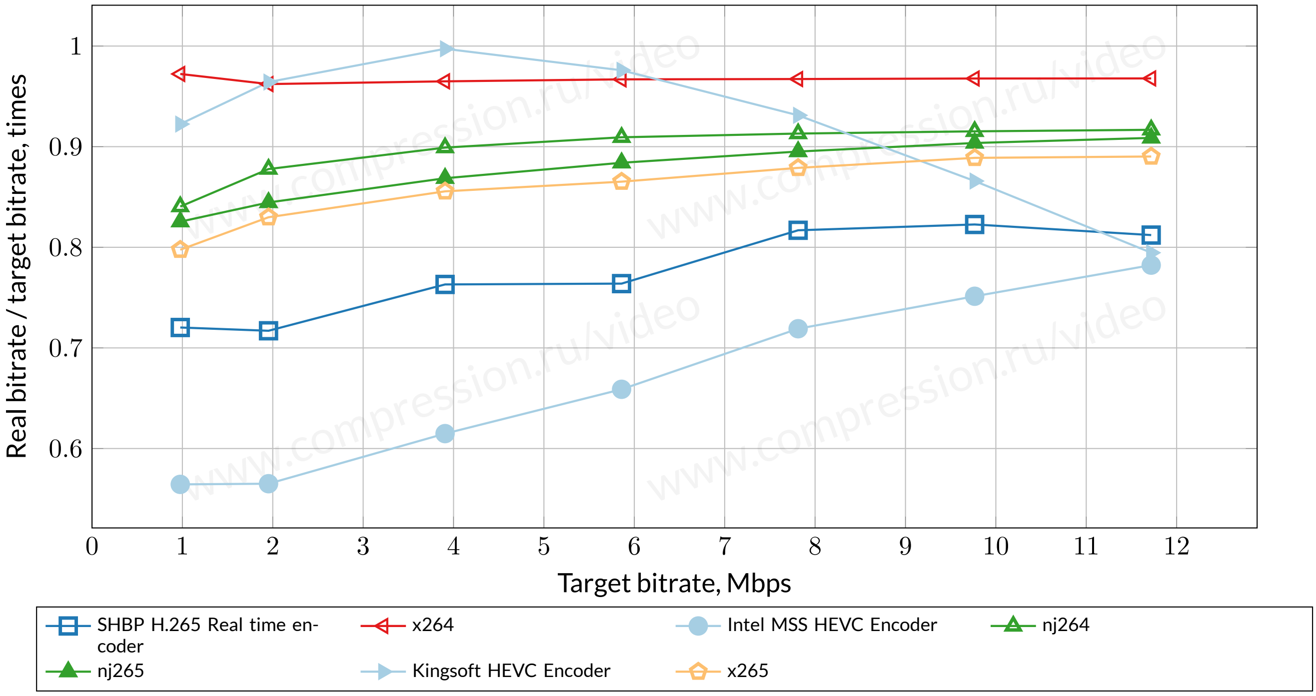


FIGURE 9: Bitrate handling—usecase "Fast Encoding," Butterflies sequence

4.5 Relative Quality Analysis

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

	SHBP H.265 Real time encoder	x264	Intel MSS HEVC Encoder	nj264	nj265	Kingsoft HEVC Encoder	x265
SHBP H.265 Real time encoder	100%	45%	29%	46%	38%	31%	39%
x264	258%	100%	78%	101%	96%	82%	102%
Intel MSS HEVC Encoder	411%	133%	100%	135%	127%	107%	137%
nj264	266%	102%	78%	100%	98%	82%	105%
nj265	295%	107%	80%	109%	100%	86%	106%
Kingsoft HEVC Encoder	390%	128%	95%	129%	122%	100%	131%
x265	284%	103%	77%	105%	96%	82%	100%

Confidence: 0% — 50% — 100%

TABLE 3: Average bitrate ratio for a fixed quality—usecase "Fast Encoding," YUV-SSIM metric

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

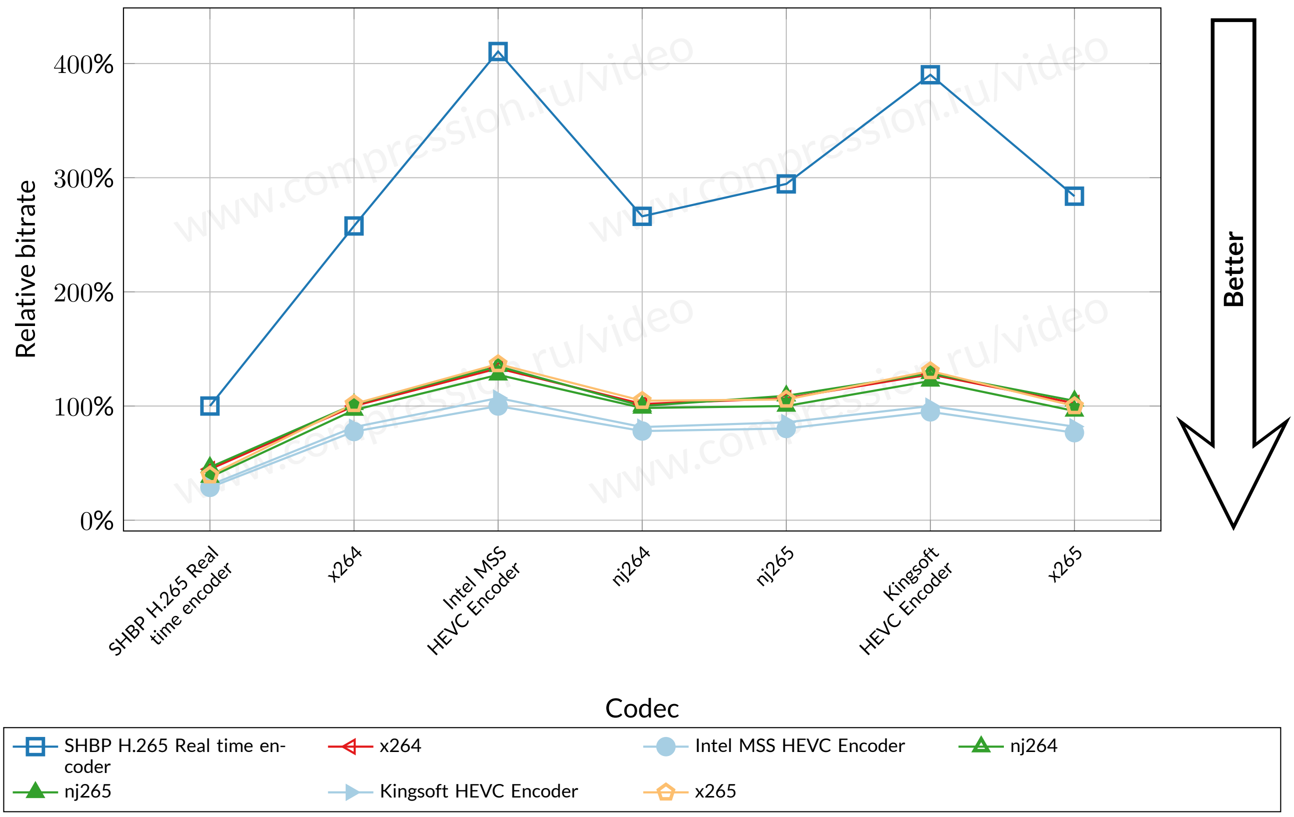


FIGURE 10: Average bitrate ratio for a fixed quality—usecase "Fast Encoding," YUV-SSIM metric

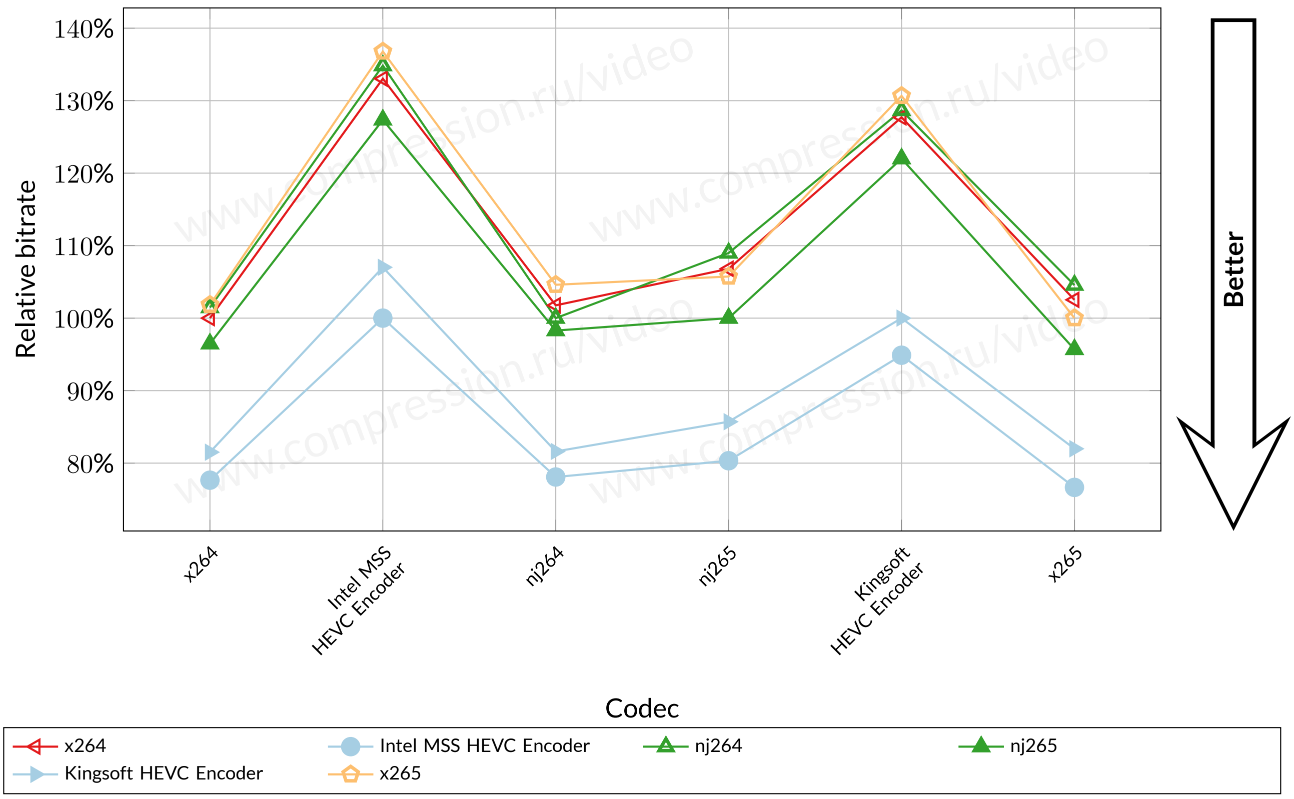


FIGURE 11: Average bitrate ratio for a fixed quality—usecase "Fast Encoding," YUV-SSIM metric, without SHBP H.265 Real time encoder

5 UNIVERSAL ENCODING

5.1 RD curves

The first encoder is Intel MSS HEVC encoder followed by Kingsoft HEVC encoder, x265 and nj265.

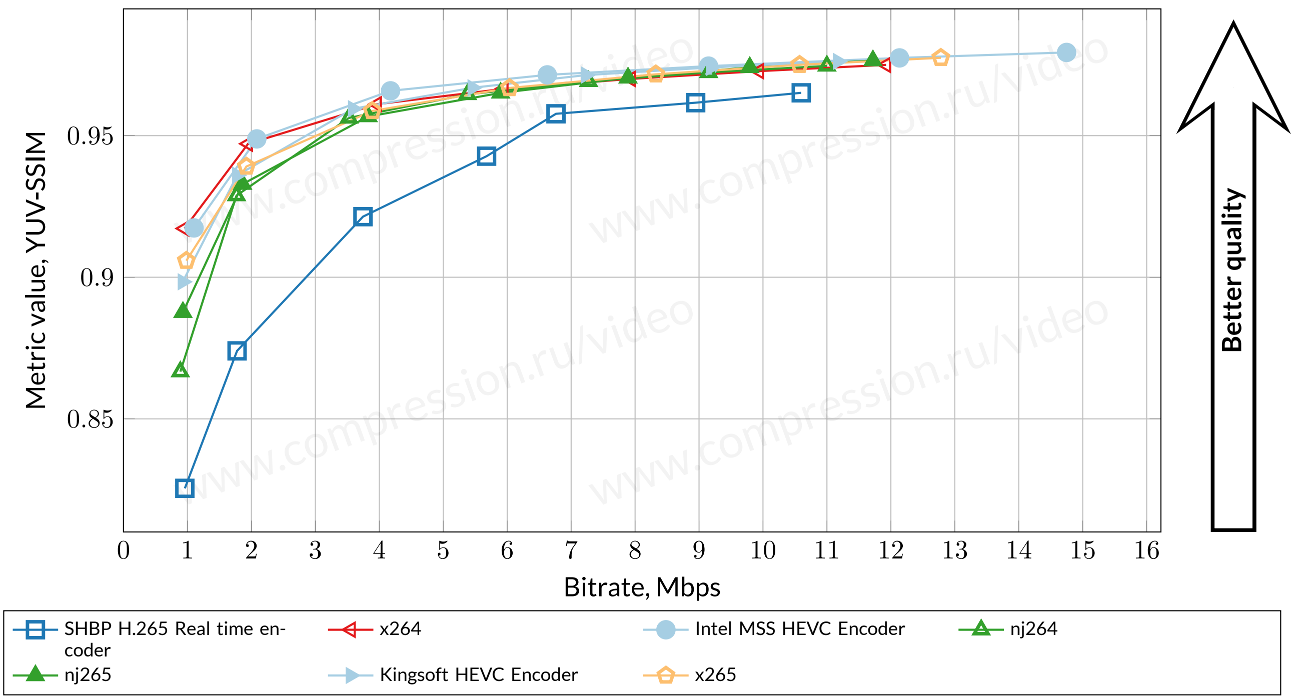


FIGURE 12: Bitrate/quality—usecase “Universal Encoding,” Apple Tree sequence, YUV-SSIM metric

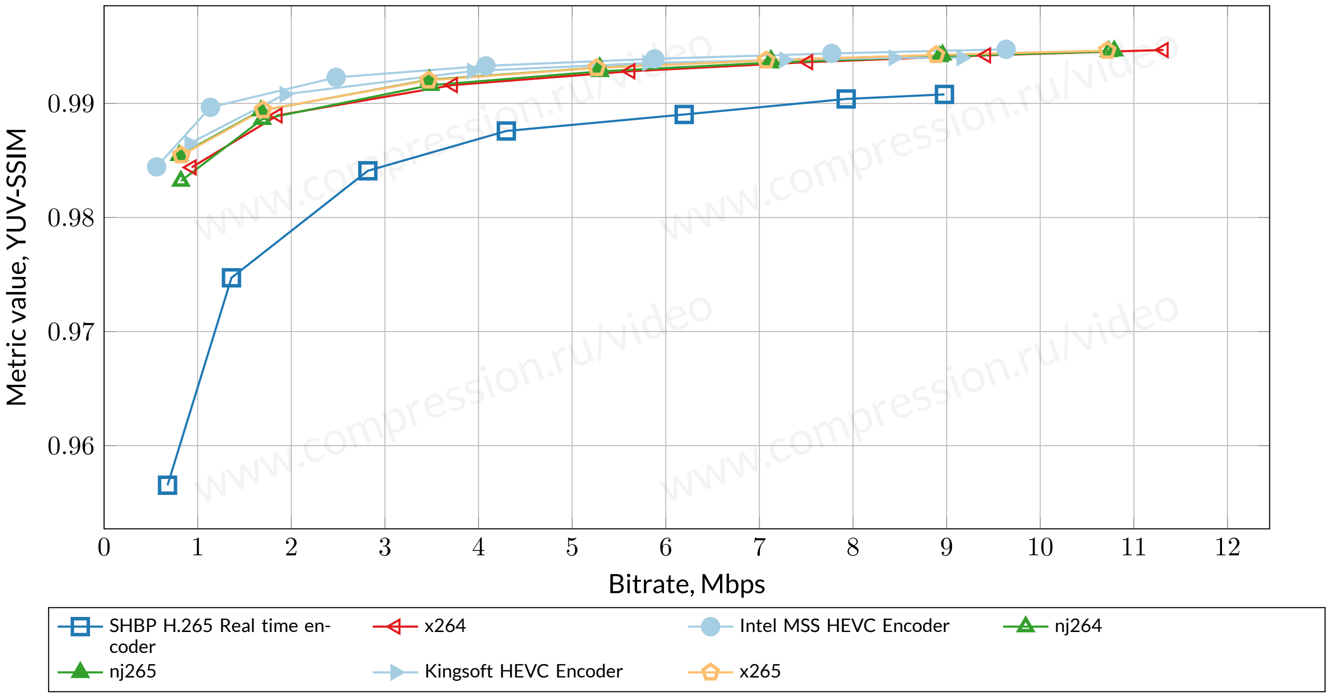


FIGURE 13: Bitrate/quality—usecase "Universal Encoding," Butterflies sequence, YUV-SSIM metric

5.2 Encoding Speed

The leaders are Kingsoft HEVC encoder and SHBP H.265 Real time encoder at average.

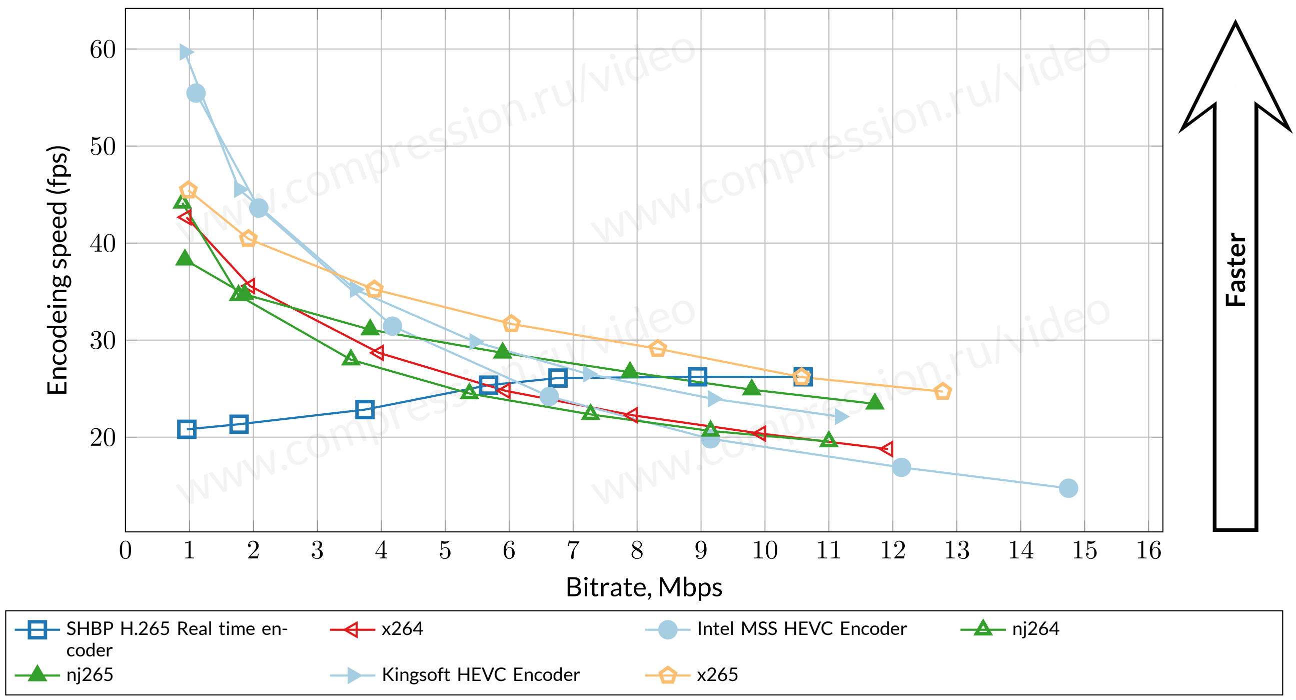


FIGURE 14: Encoding speed—usecase "Universal Encoding," Apple Tree sequence

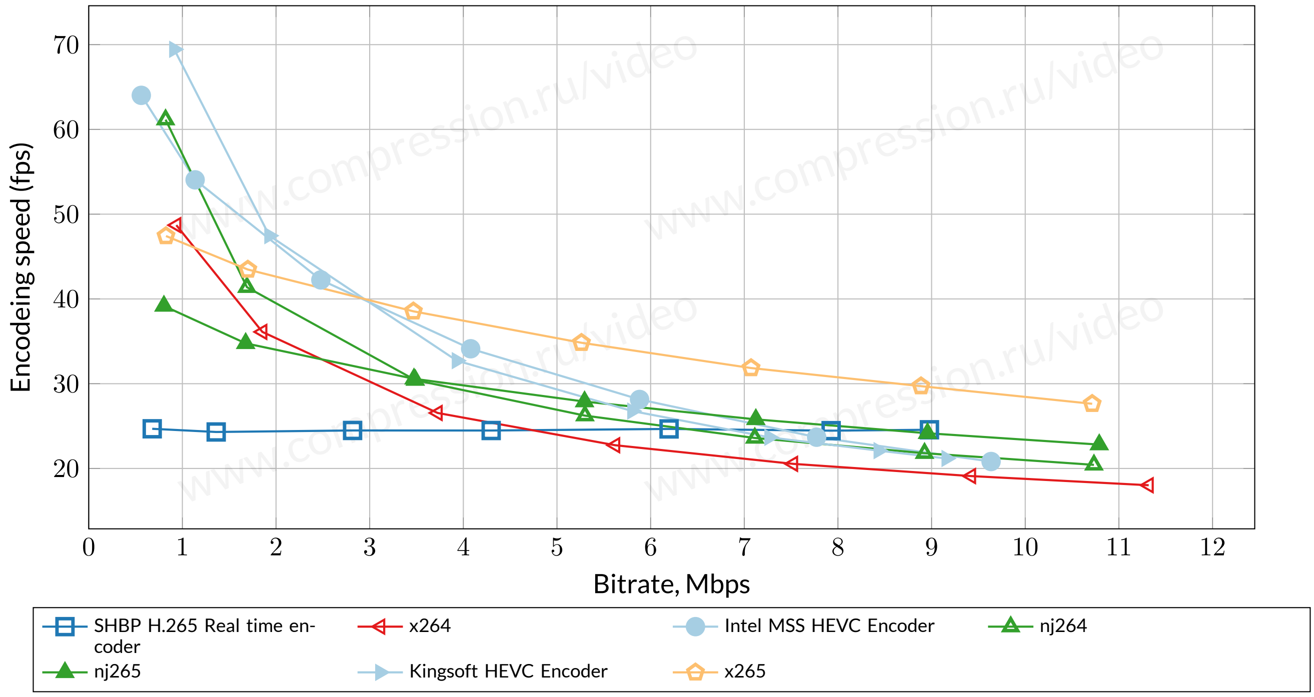


FIGURE 15: Encoding speed—usecase "Universal Encoding," Butterflies sequence

5.3 Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs can be found in Appendix F. Sometimes, codec results are not present in the particular graph owing to the codec's extremely poor performance. The codec's RD curve has no intersection with the reference's RD curve.

The speed/quality trade-off graphs simultaneously show relative quality and encoding speed for the encoders tested in this comparison. x264 is the reference codec, for which both quality and speed are normalized to unity for all of the graphs. The terms "better" and "worse" are used to compare codecs in the same manner as in previous portions of this comparison.

Pareto optimal encoders in terms of speed and quality (at average) are Kingsoft HEVC encoder and Intel MSS HEVC encoder. "Pareto optimal" encoder means there is no encoder faster and better then current one in this test.

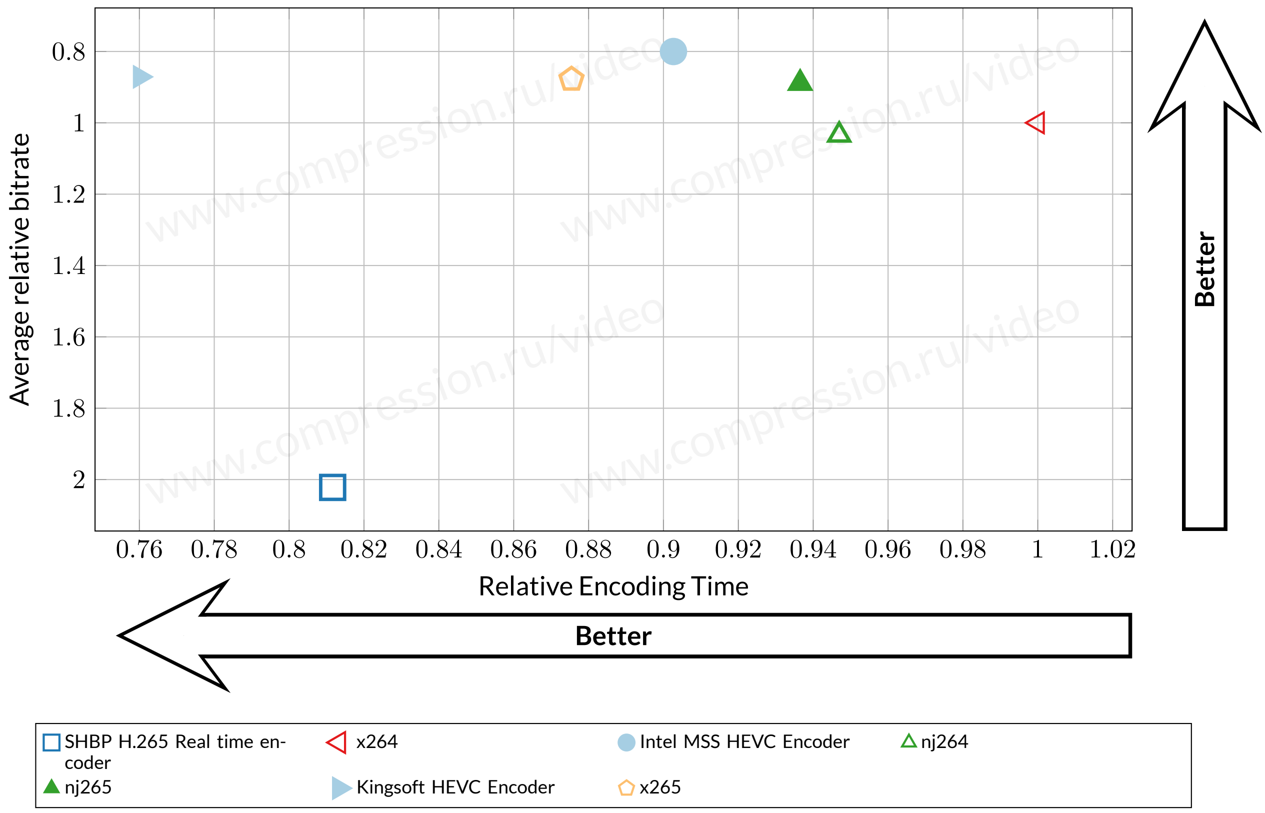


FIGURE 16: Speed/quality trade-off—usecase "Universal Encoding," all sequences, YUV-SSIM metric

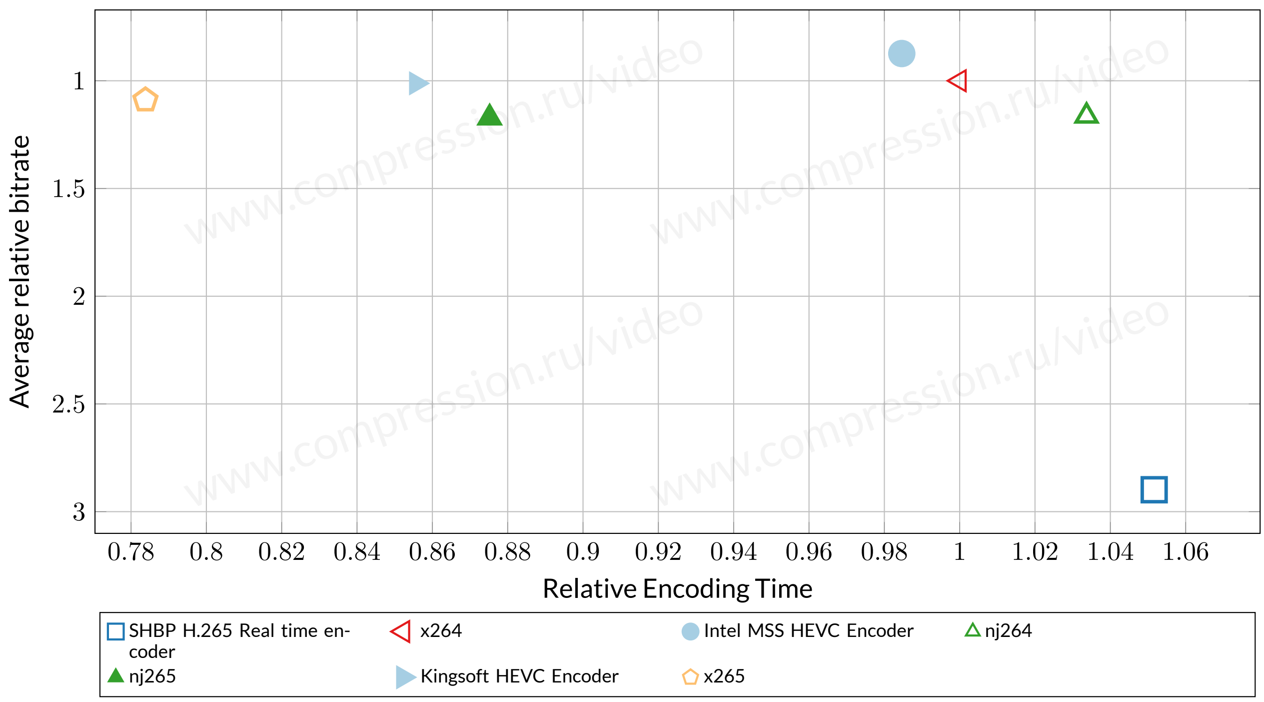


FIGURE 17: Speed/quality trade-off—usecase "Universal Encoding," Apple Tree sequence, YUV-SSIM metric

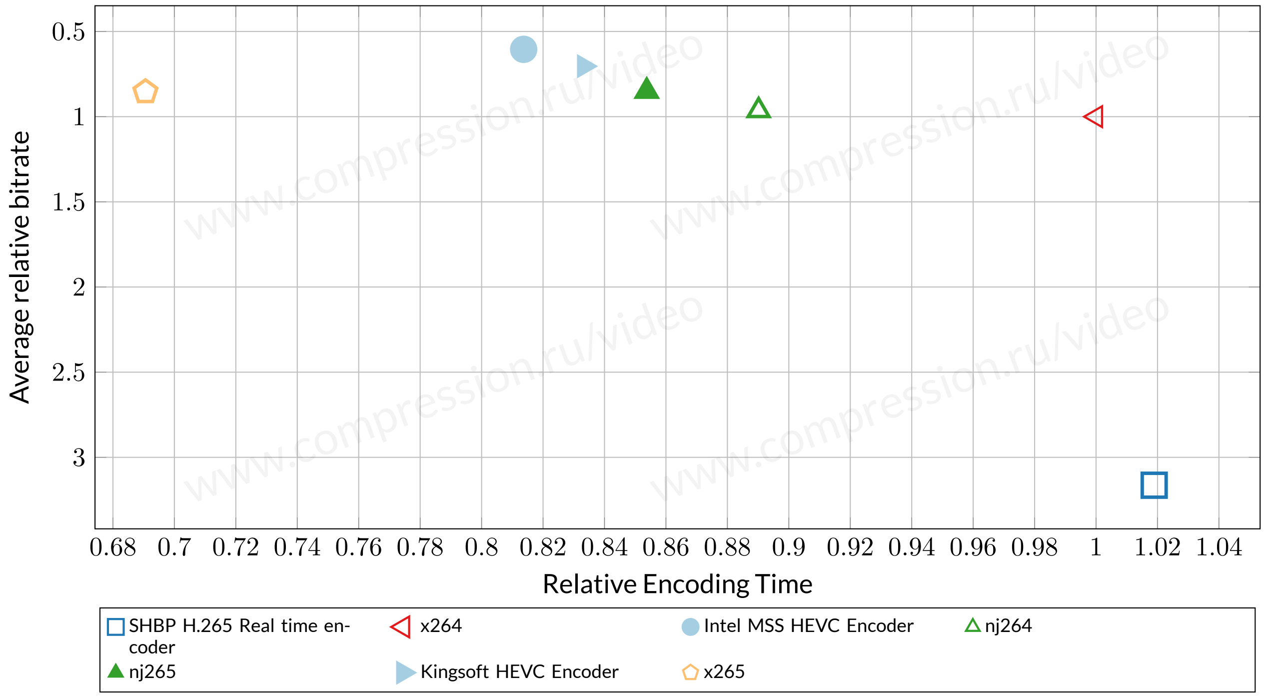


FIGURE 18: Speed/quality trade-off—usecase "Universal Encoding," Butterflies sequence, YUV-SSIM metric

5.4 Bitrate Handling

The plots below show how accurately encoded stream's real bitrate matches bitrate requested by user. Almost all encoders handle bitrate well, but there are issues for some encoders at some sequnces, e.g. nj264, SHBP H.265 Real time encoder and Intel MSS HEVC encoder.

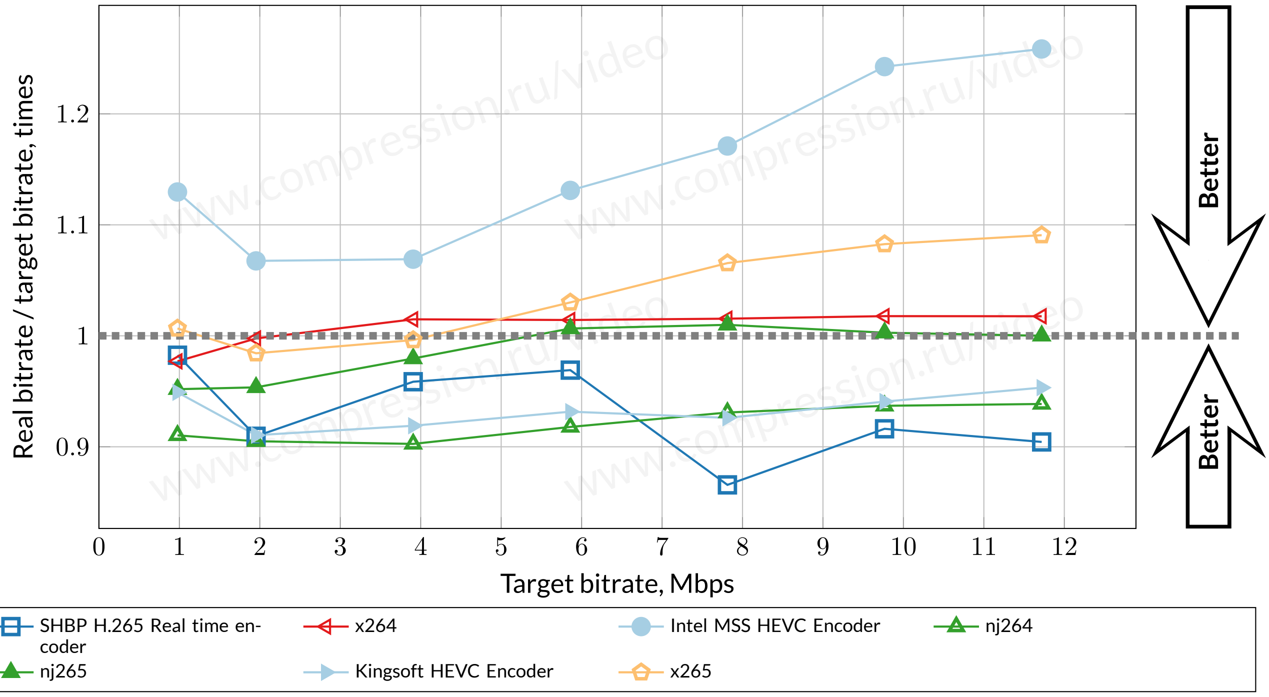


FIGURE 19: Bitrate handling—usecase "Universal Encoding," Apple Tree sequence

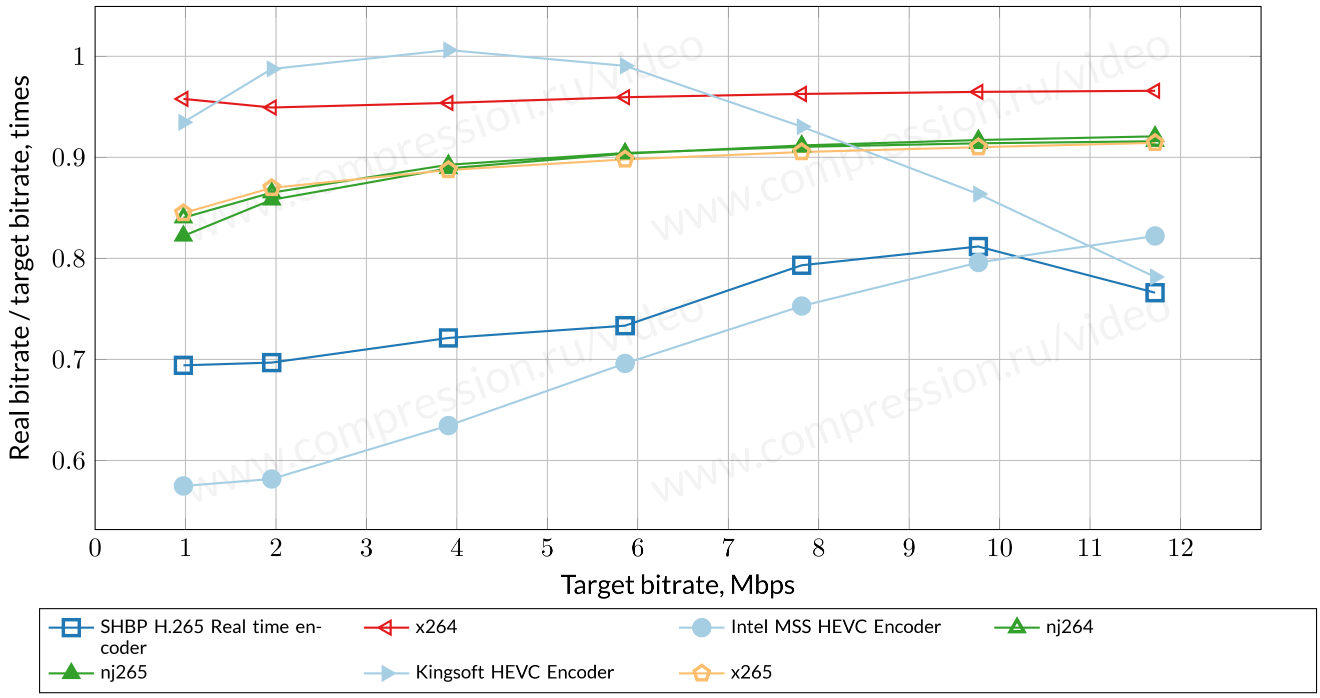


FIGURE 20: Bitrate handling—usecase "Universal Encoding," Butterflies sequence

5.5 Relative Quality Analysis

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

	SHBP H.265 Real time encoder	x264	Intel MSS HEVC Encoder	nj264	nj265	Kingsoft HEVC Encoder	x265
SHBP H.265 Real time encoder	100%	49%	36%	54%	41%	39%	41%
x264	217%	100%	82%	107%	90%	89%	89%
Intel MSS HEVC Encoder	314%	125%	100%	136%	113%	113%	110%
nj264	215%	97%	77%	100%	85%	84%	85%
nj265	271%	112%	90%	121%	100%	100%	99%
Kingsoft HEVC Encoder	282%	115%	90%	122%	101%	100%	100%
x265	271%	114%	92%	124%	102%	102%	100%

TABLE 4: Average bitrate ratio for a fixed quality—usecase "Universal Encoding," YUV-SSIM metric

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

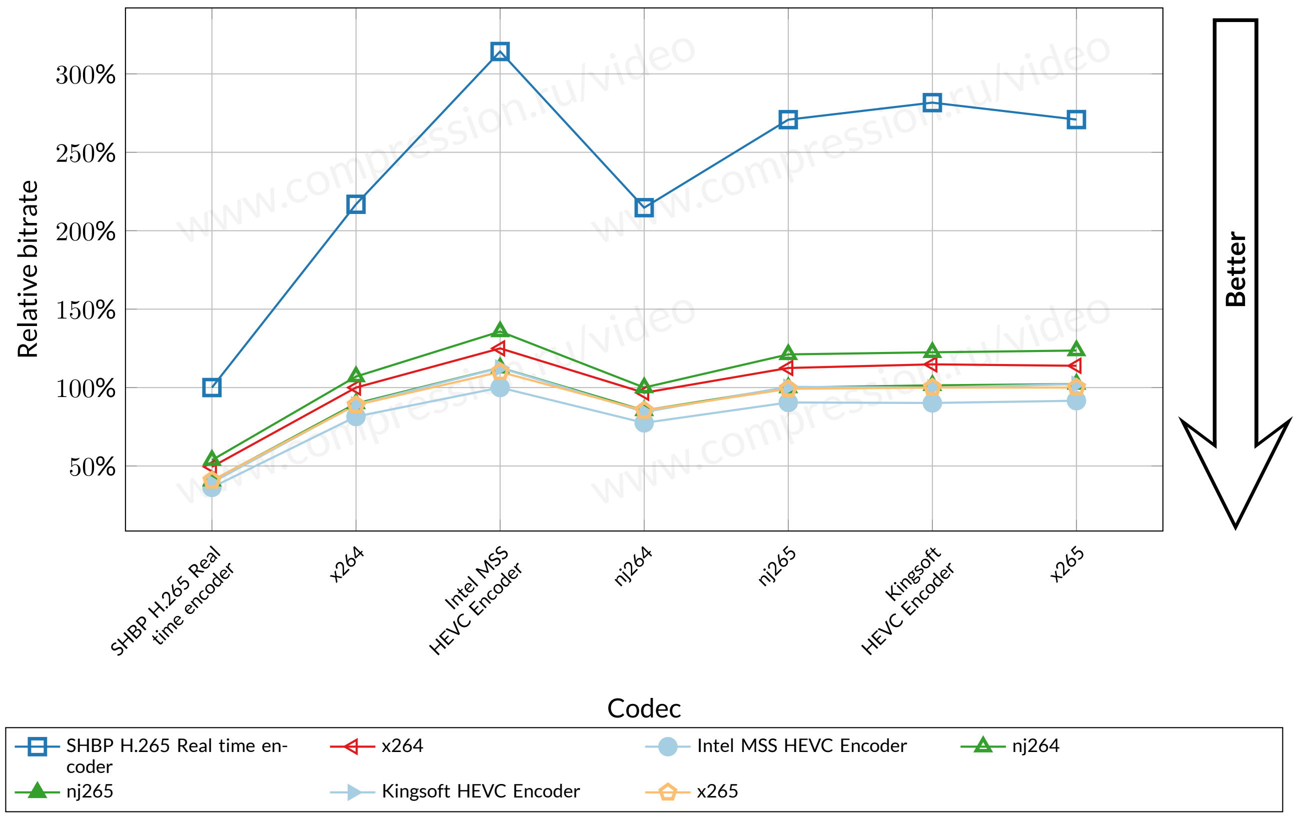


FIGURE 21: Average bitrate ratio for a fixed quality—usecase “Universal Encoding,” YUV-SSIM metric

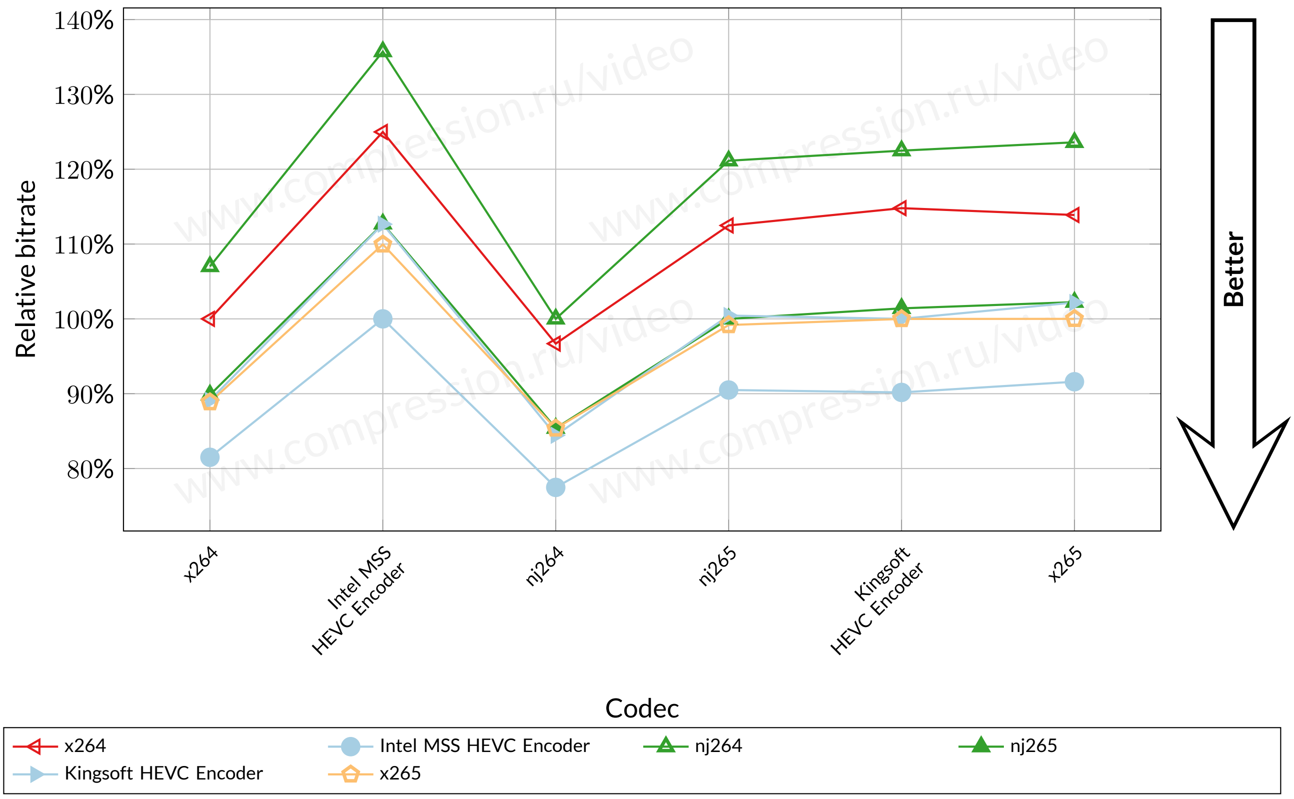


FIGURE 22: Average bitrate ratio for a fixed quality—usecase "Universal Encoding," YUV-SSIM metric, without SHBP H.265 Real time encoder

6 RIPPING ENCODING

6.1 RD curves

The first encoder at most sequecnes by quality is Intel MSS HEVC encoder.

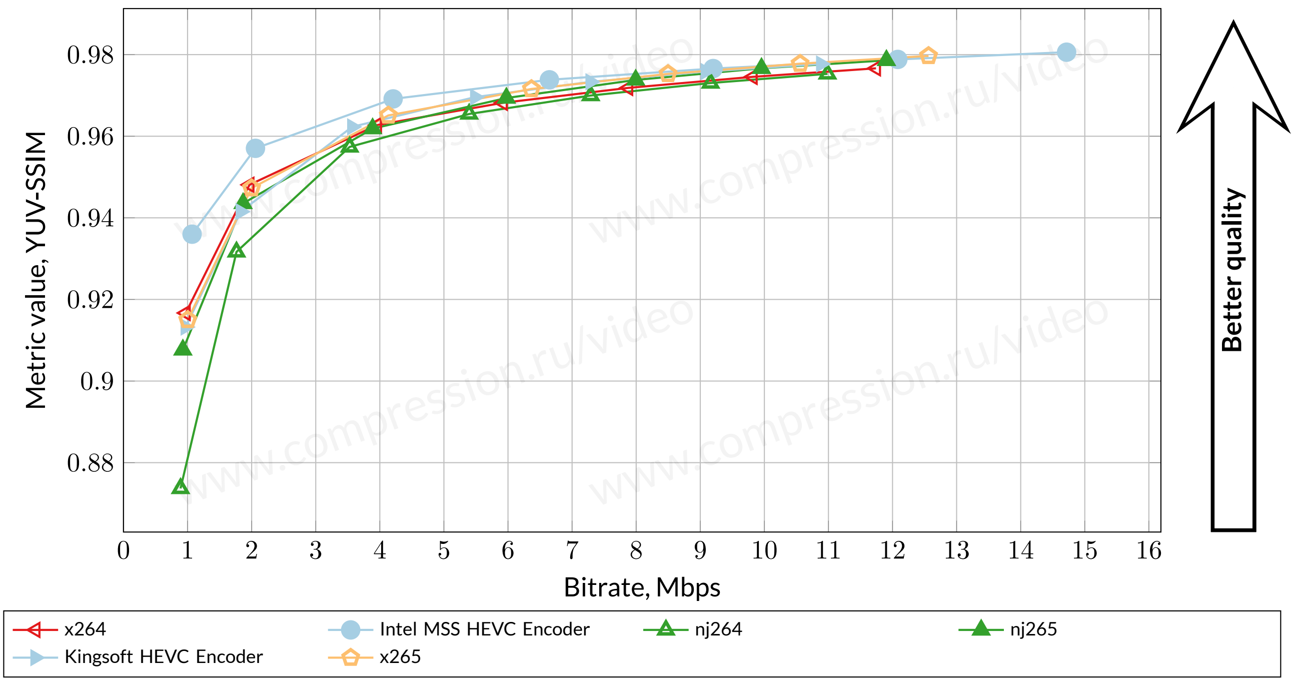


FIGURE 23: Bitrate/quality—usecase "Ripping Encoding," Apple Tree sequence, YUV-SSIM metric

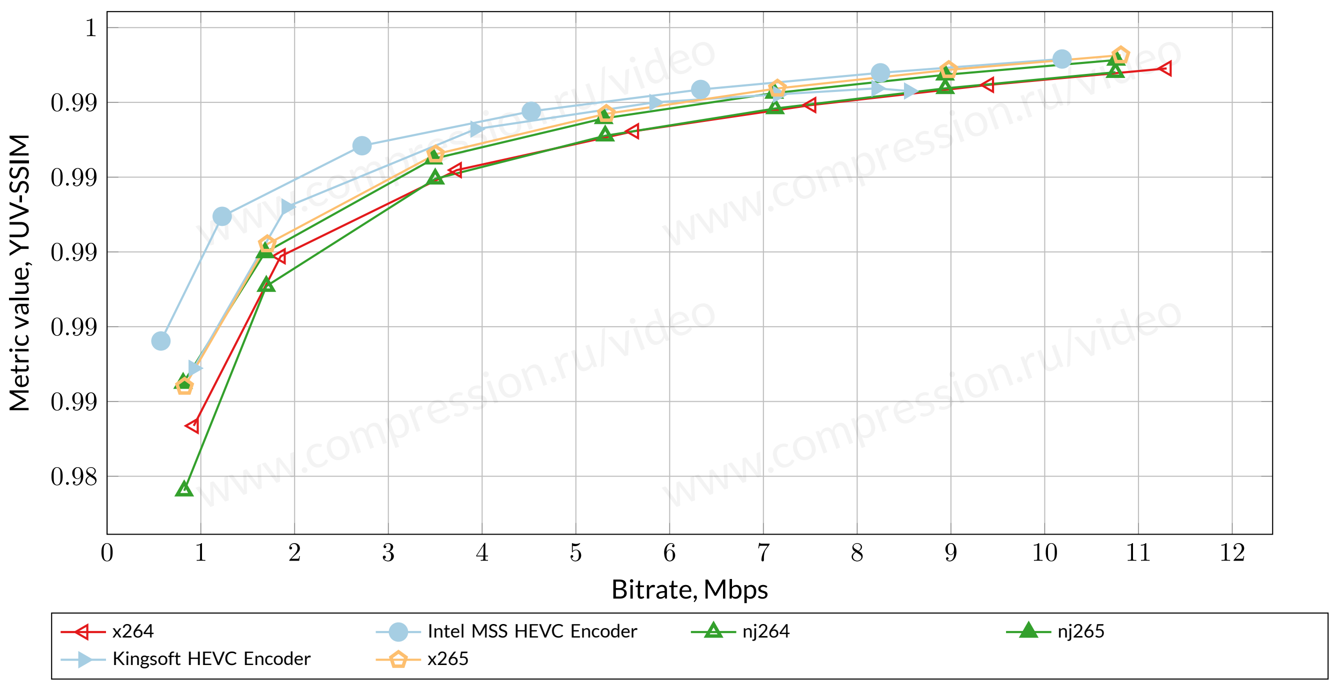


FIGURE 24: Bitrate/quality—usecase "Ripping Encoding," Butterflies sequence, YUV-SSIM metric

6.2 Encoding Speed

According to encoding speed analysis, the first places go to nj264, x264.

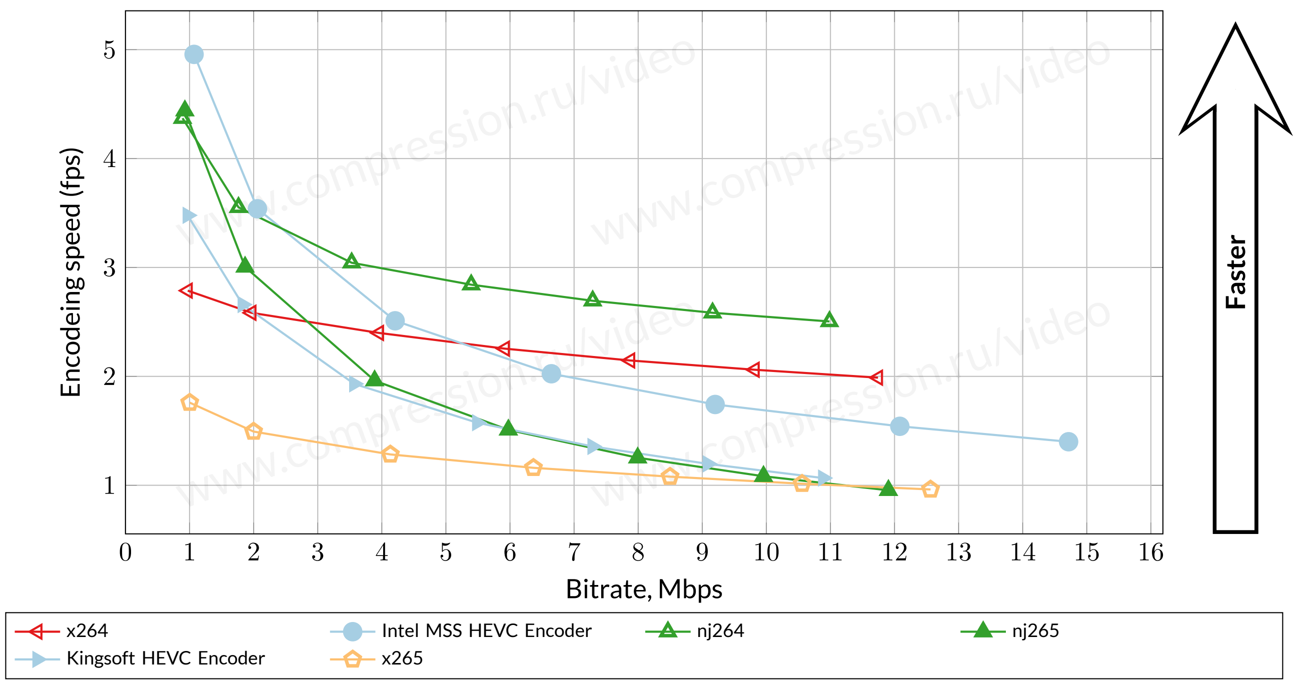


FIGURE 25: Encoding speed—usecase "Ripping Encoding," Apple Tree sequence

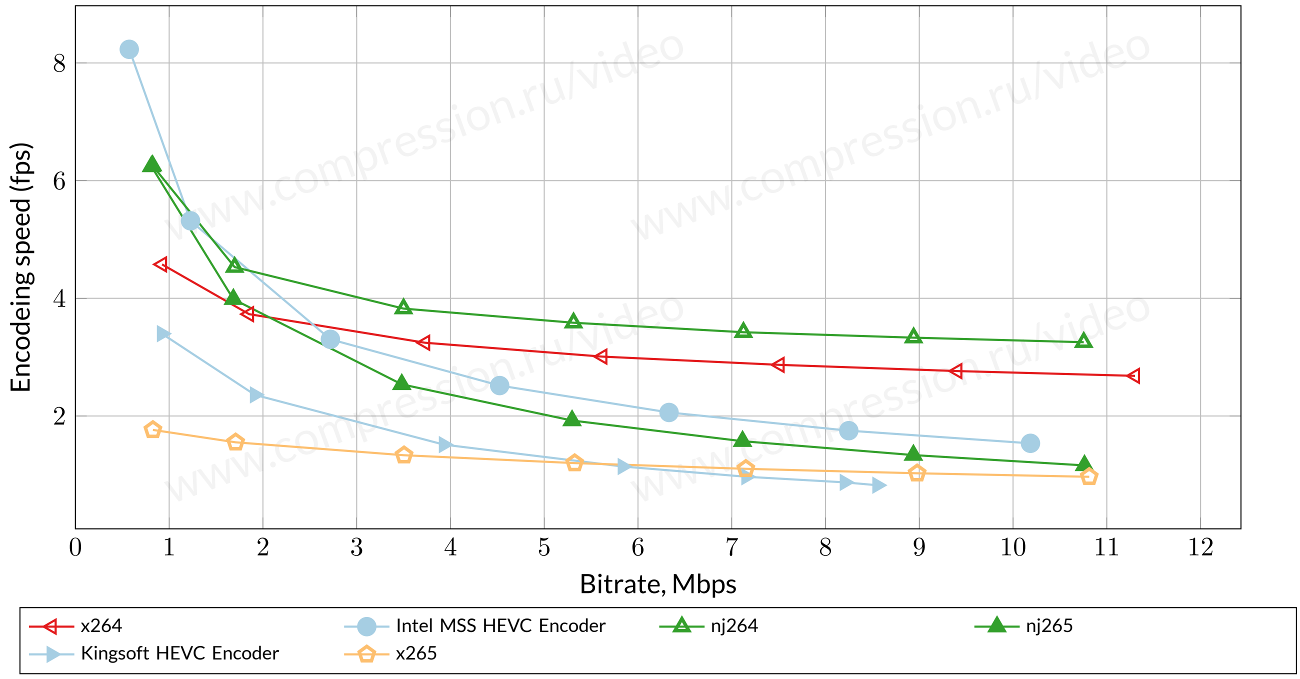


FIGURE 26: Encoding speed—usecase "Ripping Encoding," Butterflies sequence

6.3 Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs can be found in Appendix F. Sometimes, codec results are not present in the particular graph owing to the codec's extremely poor performance. The codec's RD curve has no intersection with the reference's RD curve.

The speed/quality trade-off graphs simultaneously show relative quality and encoding speed for the encoders tested in this comparison. x264 is the reference codec, for which both quality and speed are normalized to unity for all of the graphs. The terms "better" and "worse" are used to compare codecs in the same manner as in previous portions of this comparison.

Pareto optimal encoders in terms of speed and quality (at average) are nj264, x264 and Intel MSS HEVC encoder. "Pareto optimal" encoder means there is no encoder faster and better then current one in this test.

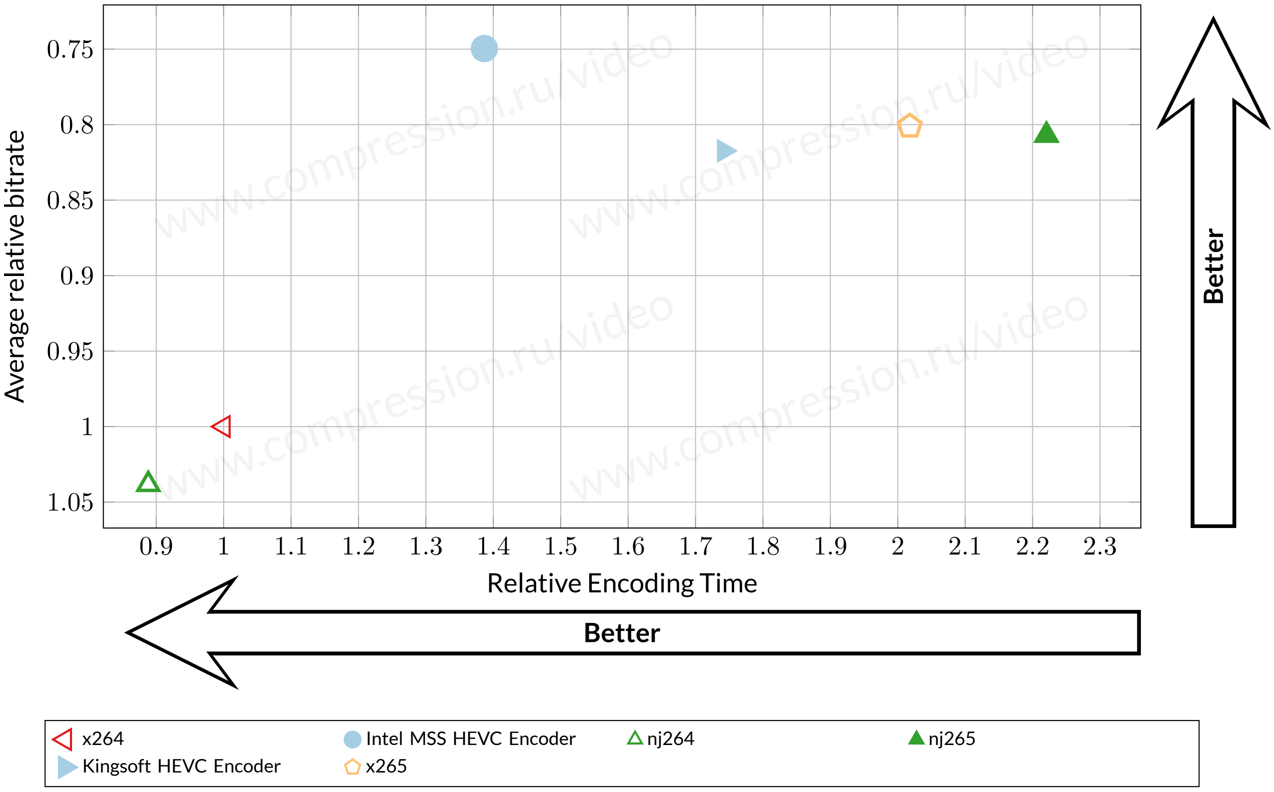


FIGURE 27: Speed/quality trade-off—usecase “Ripping Encoding,” all sequences, YUV-SSIM metric

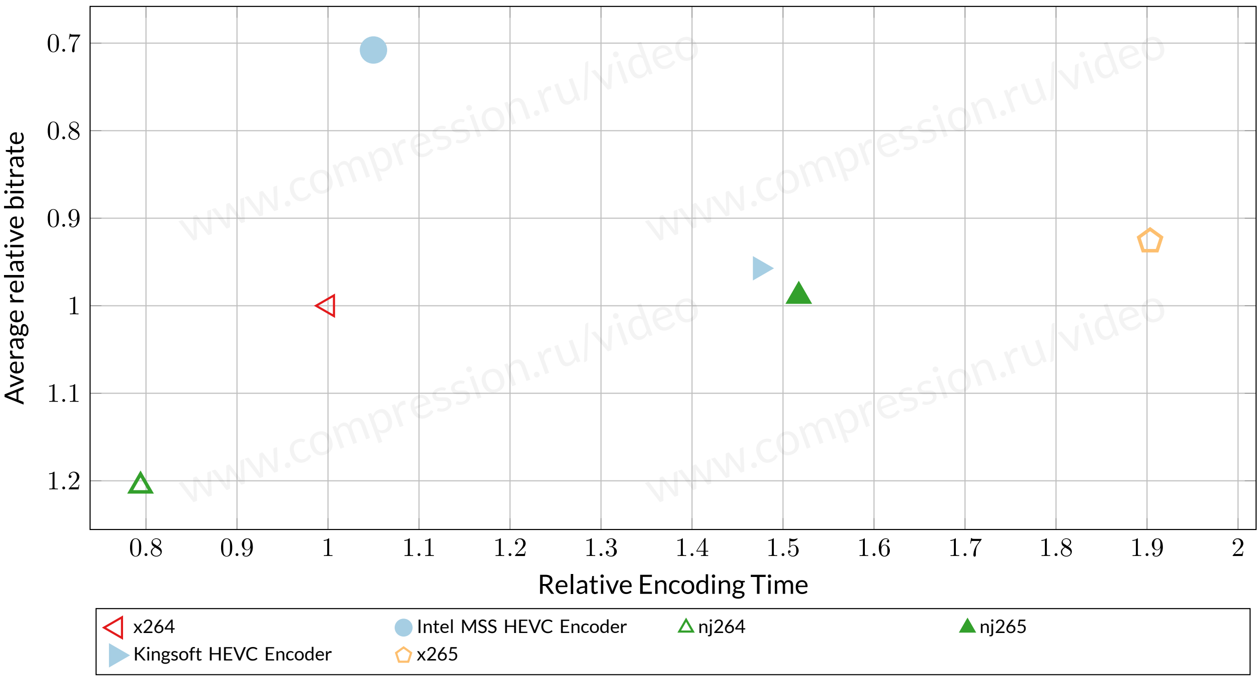


FIGURE 28: Speed/quality trade-off—usecase “Ripping Encoding,” Apple Tree sequence, YUV-SSIM metric

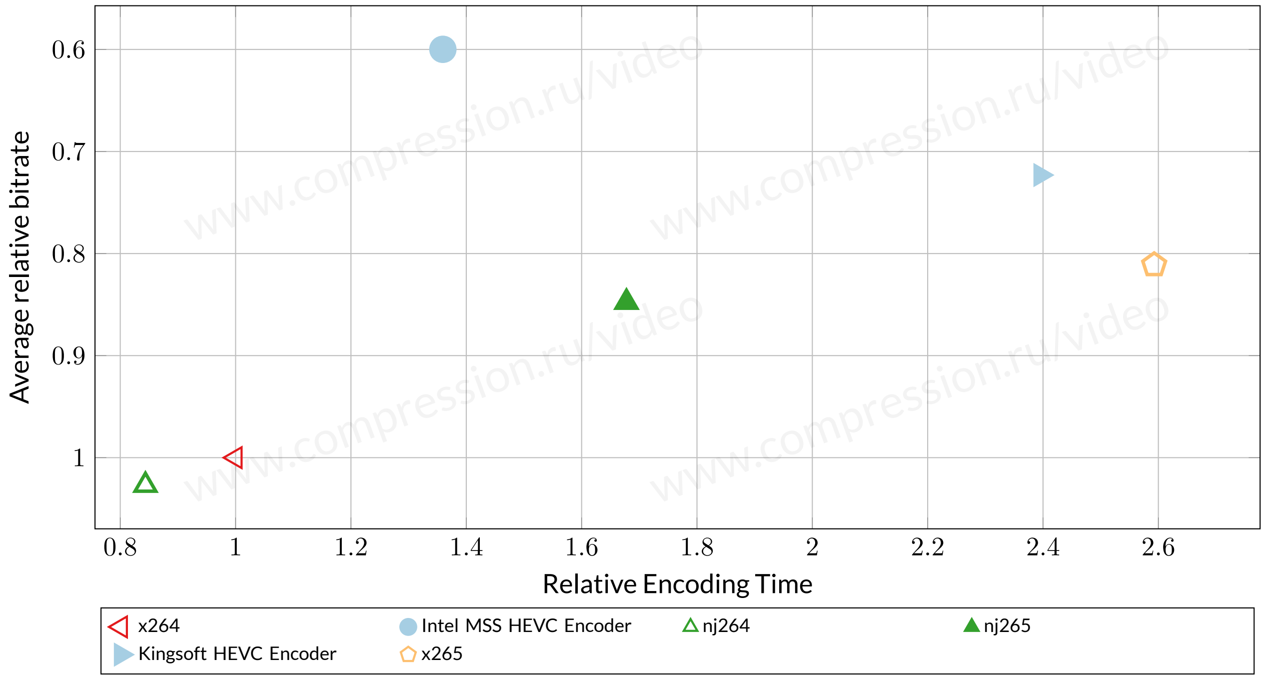


FIGURE 29: Speed/quality trade-off—usecase "Ripping Encoding," Butterflies sequence, YUV-SSIM metric

6.4 Bitrate Handling

The plots below show how accurately encoded stream's real bitrate matches bitrate requested by user. Encoders sometimes fail to correctly handle bitrate on some video sequences.

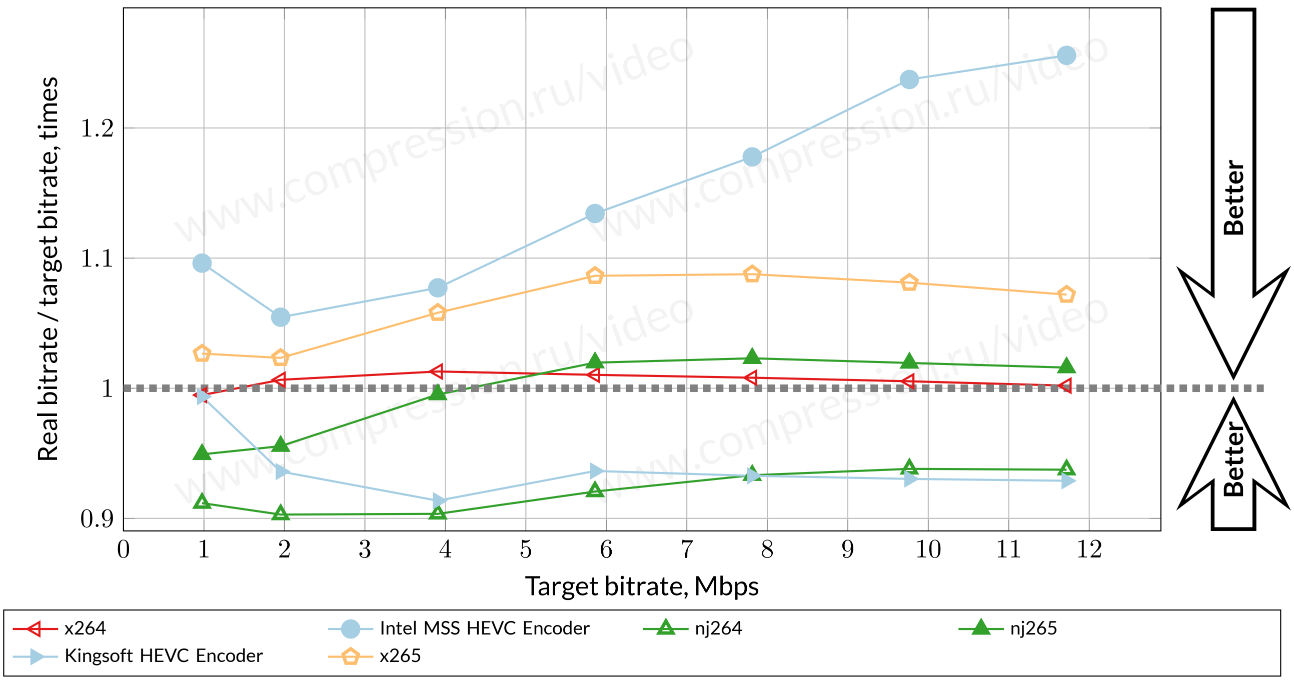


FIGURE 30: Bitrate handling—usecase "Ripping Encoding," Apple Tree sequence

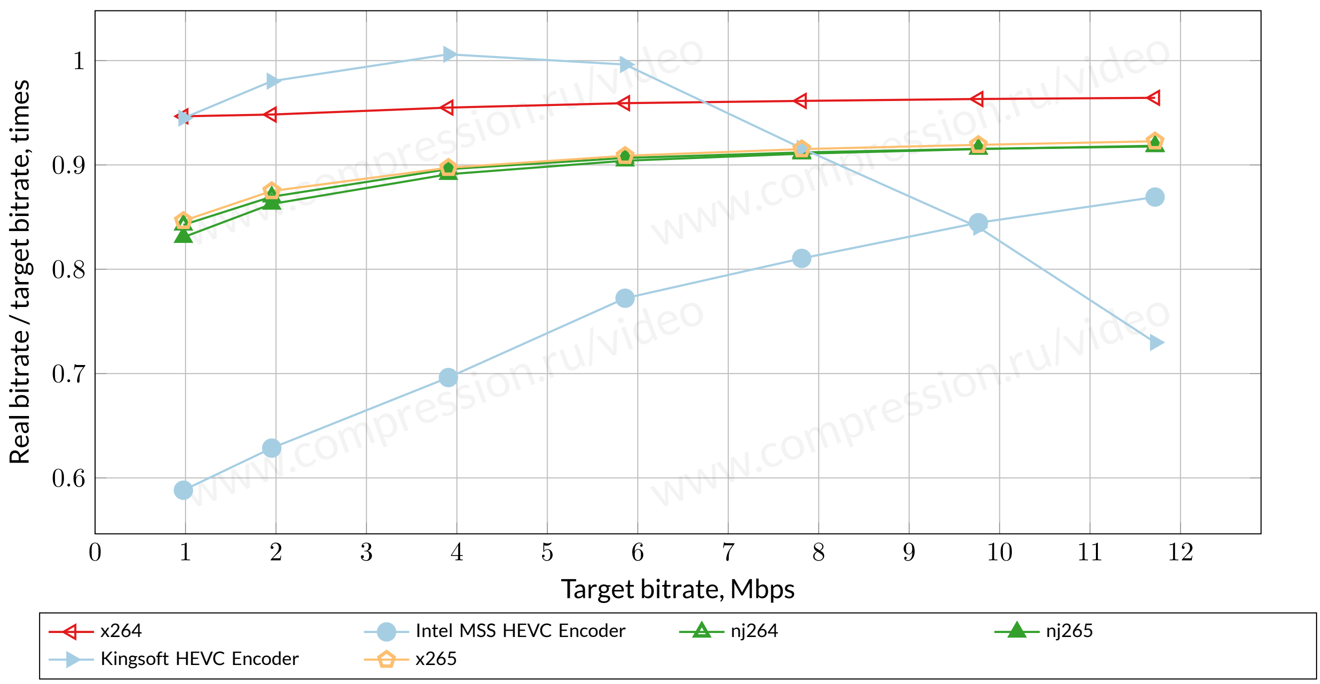


FIGURE 31: Bitrate handling—usecase "Ripping Encoding," Butterflies sequence

6.5 Relative Quality Analysis

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

	x264	Intel MSS HEVC Encoder	nj264	nj265	Kingsoft HEVC Encoder	x265
x264	100%☺	76%☺	106%☺	81%☺	84%☺	81%☺
Intel MSS HEVC Encoder	133%☺	100%☺	144%☺	108%☺	112%☺	106%☺
nj264	96%☺	72%☺	100%☺	78%☺	79%☺	78%☺
nj265	124%☺	94%☺	133%☺	100%☺	104%☺	100%☺
Kingsoft HEVC Encoder	122%☺	91%☺	131%☺	99%☺	100%☺	97%☺
x265	125%☺	95%☺	134%☺	101%☺	104%☺	100%☺

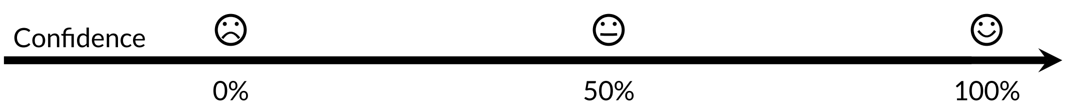


TABLE 5: Average bitrate ratio for a fixed quality—usecase "Ripping Encoding," YUV-SSIM metric

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

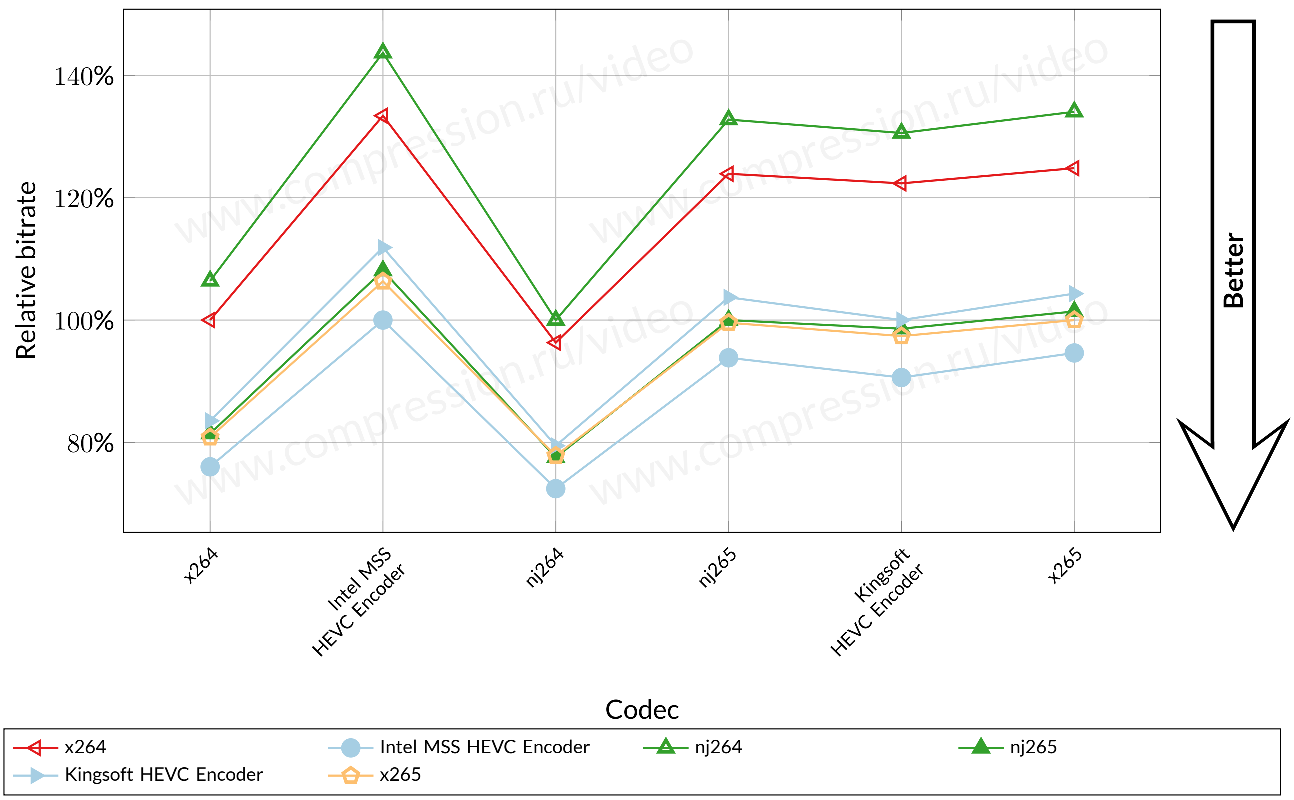


FIGURE 32: Average bitrate ratio for a fixed quality—usecase "Ripping Encoding," YUV-SSIM metric

7 CONCLUSION

All encoders could be ranged by quality in the following way:

- First place is for Intel MSS HEVC encoder
- Second place is for Kingsoft HEVC encoder
- Third place is for nj265 and x265.

In this report we introduce new plot illustrating speed/quality relation of all presets used in our comparison. x264 with ripping preset was choosen as the reference point. Each line on the plot corresponds to encoder and each point on the line corresponds to preset. Along x-axis we put mean speed of encoder's preset on our test datset. Position along y-axis is determined by preset's bitrate relative to reference (i.e. how much or less bits encoder needs to gain same quality as reference). Detailed description of relative bitrate computation can be found in Appendix F.4.

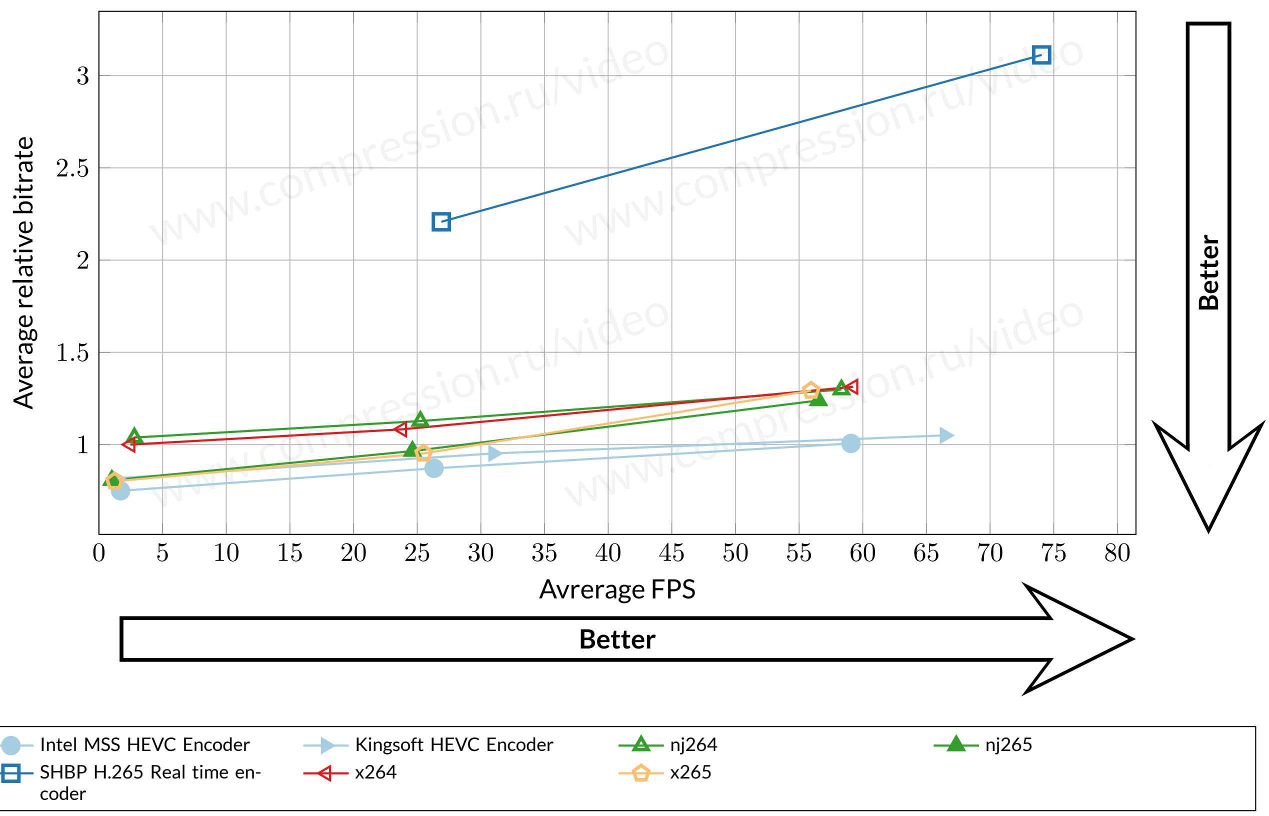


FIGURE 33: Average bitrate ratio and speed for various presets–YUV-SSIM metric.

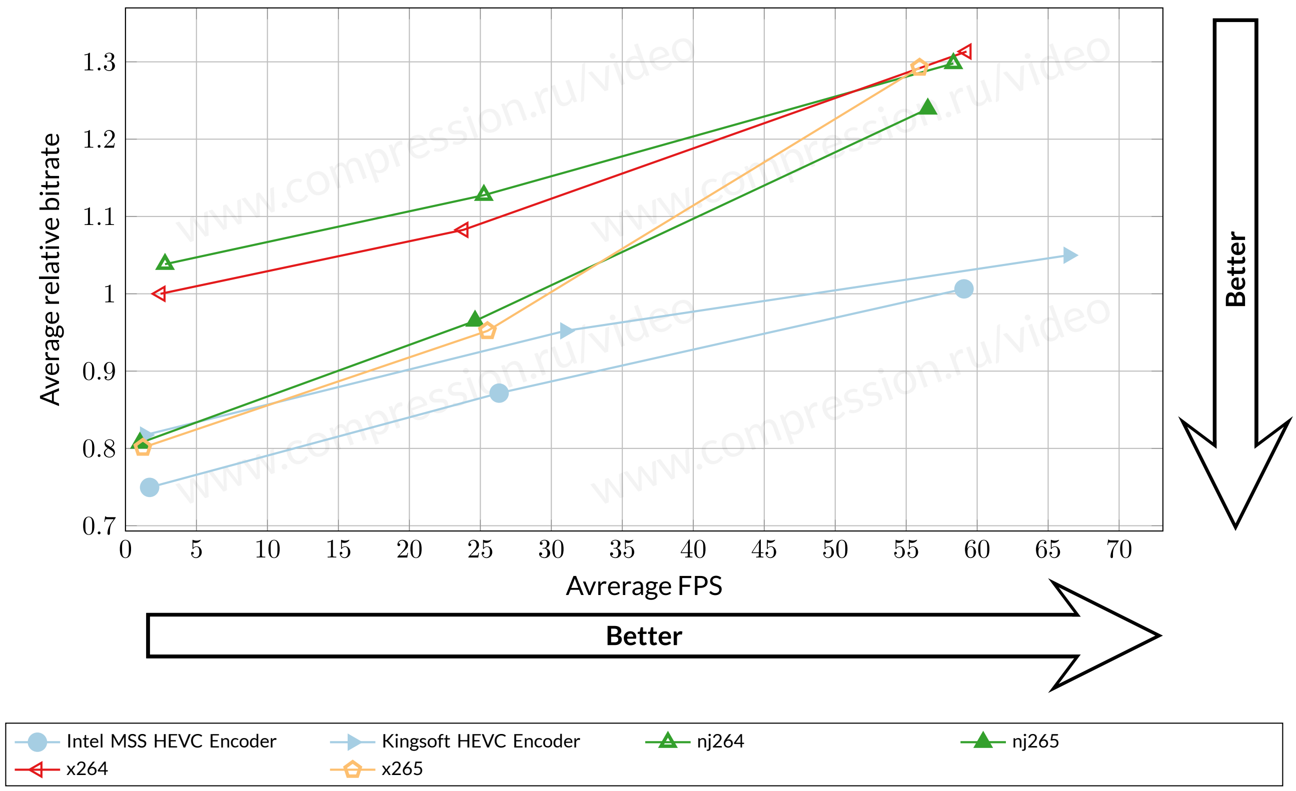


FIGURE 34: Average bitrate ratio and speed for various presets–YUV-SSIM metric, without SHBP H.265 Real time encoder.

7.1 Fast Encoding

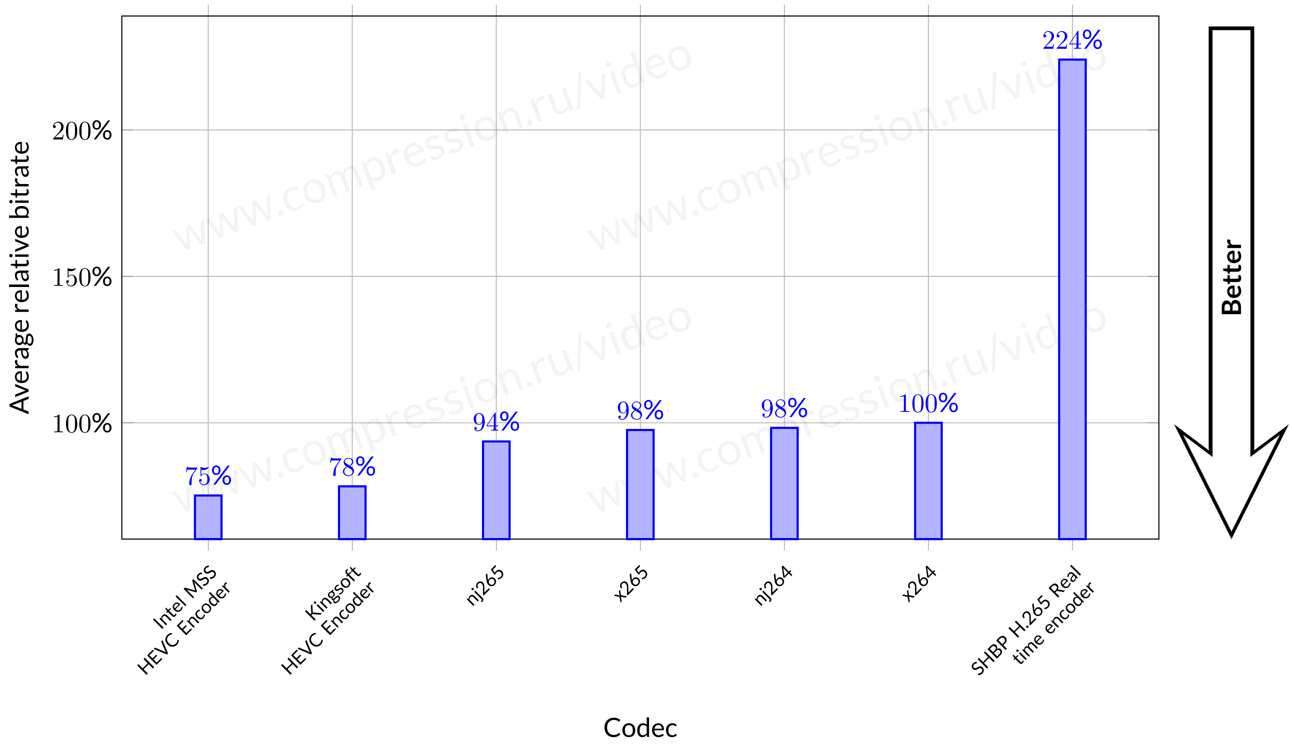


FIGURE 35: Average bitrate ratio for a fixed quality—usecase "Fast Encoding," YUV-SSIM metric.

7.2 Universal Encoding

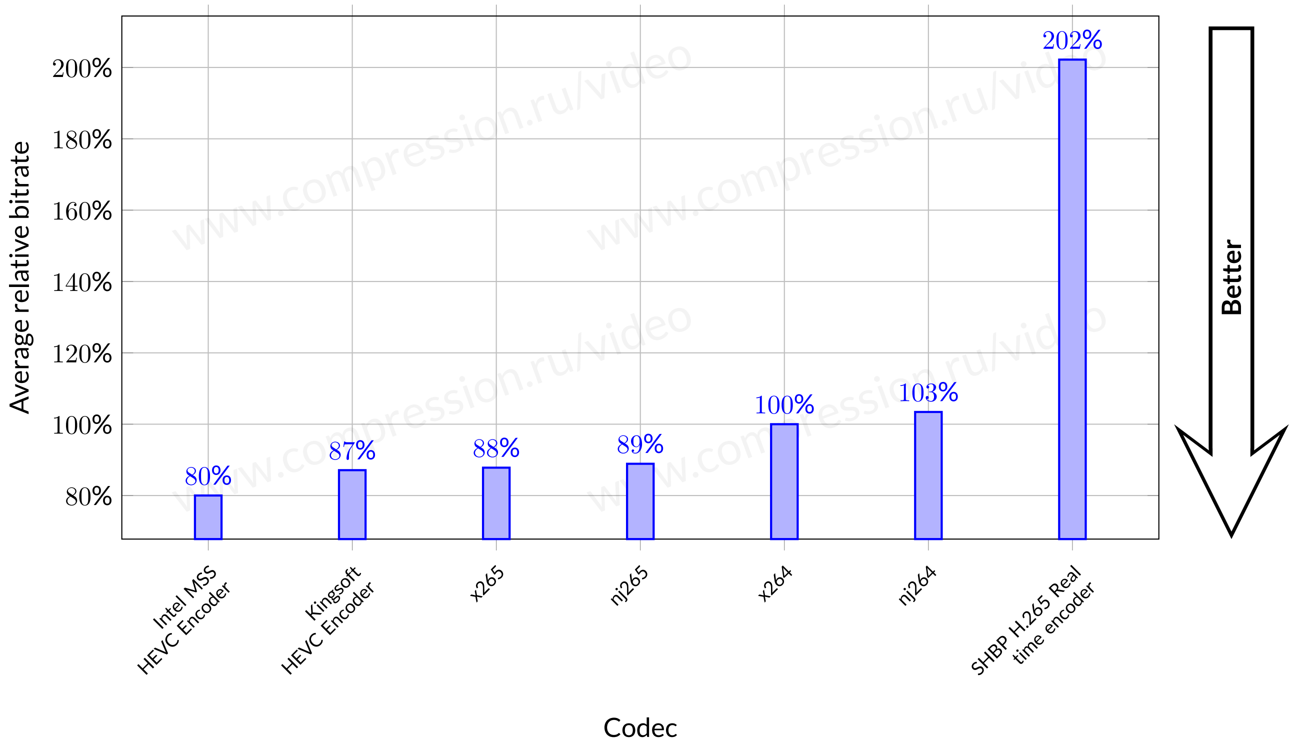


FIGURE 36: Average bitrate ratio for a fixed quality—usecase "Universal Encoding," YUV-SSIM metric.

7.3 Ripping Encoding

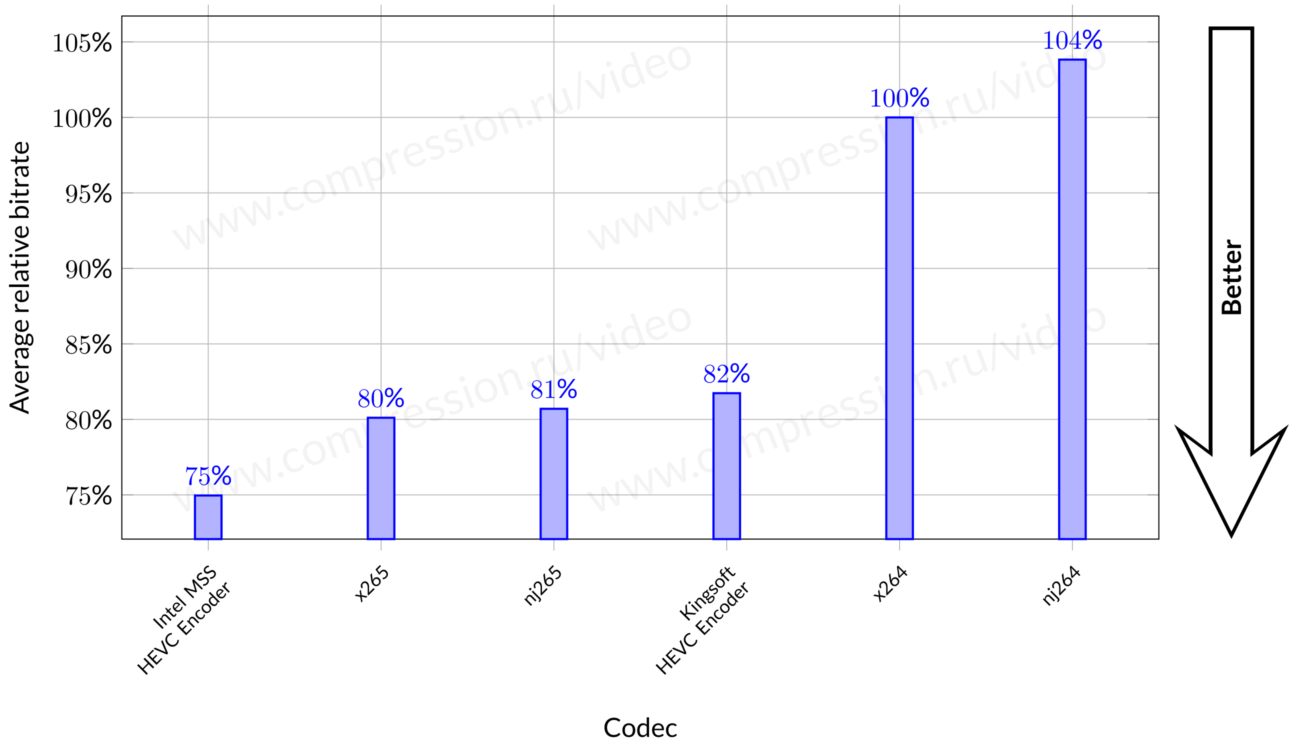


FIGURE 37: Average bitrate ratio for a fixed quality—usecase "Ripping Encoding," YUV-SSIM metric.

7.4 Overall

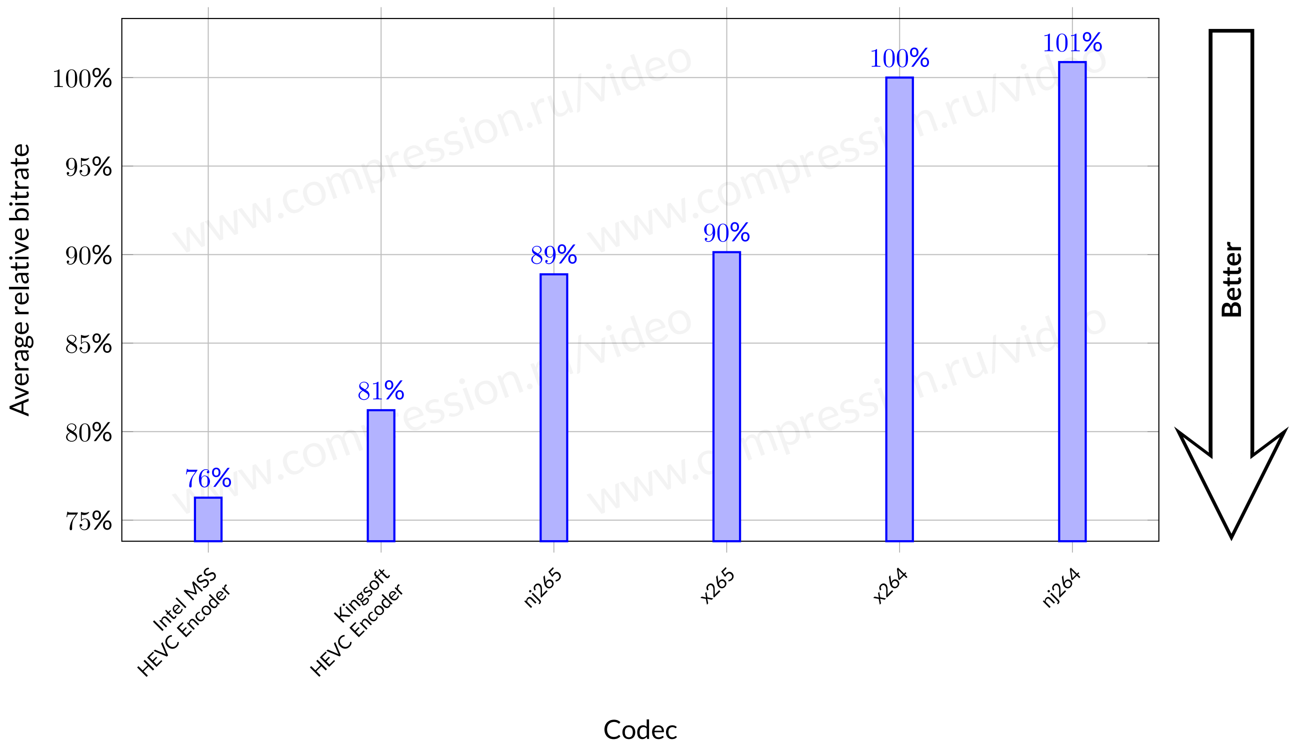


FIGURE 38: Average bitrate ratio for a fixed quality—YUV-SSIM metric.

A SOFTWARE AND HARDWARE ENCODERS COMPARISON

A.1 RD curves

Chips&Media HEVC Encoder by quality is in top 3 encoders and at some sequences (e.g. Apple Tree) it shows the best result.

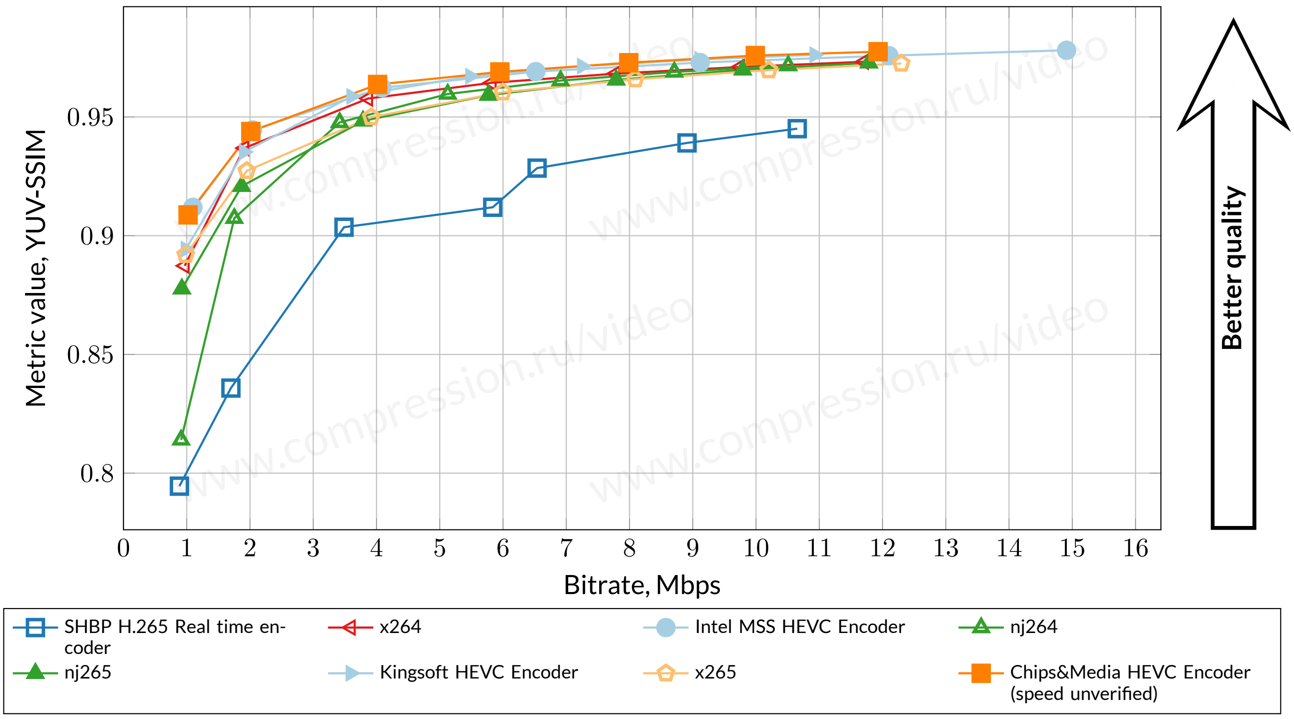


FIGURE 39: Bitrate/quality—usecase "Software and Hardware Encoders Comparison," Apple Tree sequence, YUV-SSIM metric

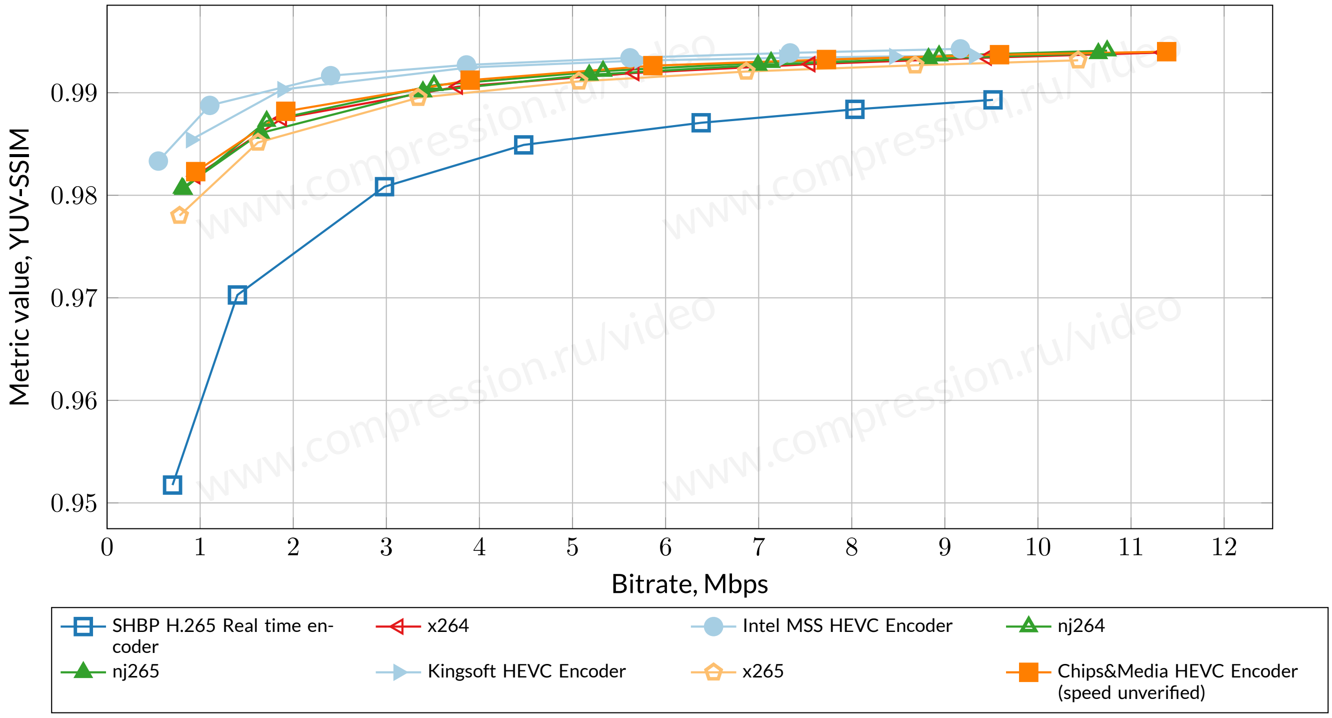


FIGURE 40: Bitrate/quality—usecase "Software and Hardware Encoders Comparison," Butterflies sequence, YUV-SSIM metric

A.2 Encoding Speed

Chips&Media HEVC Encoder speed is 240 fps at FullHD (data is provided by developer)

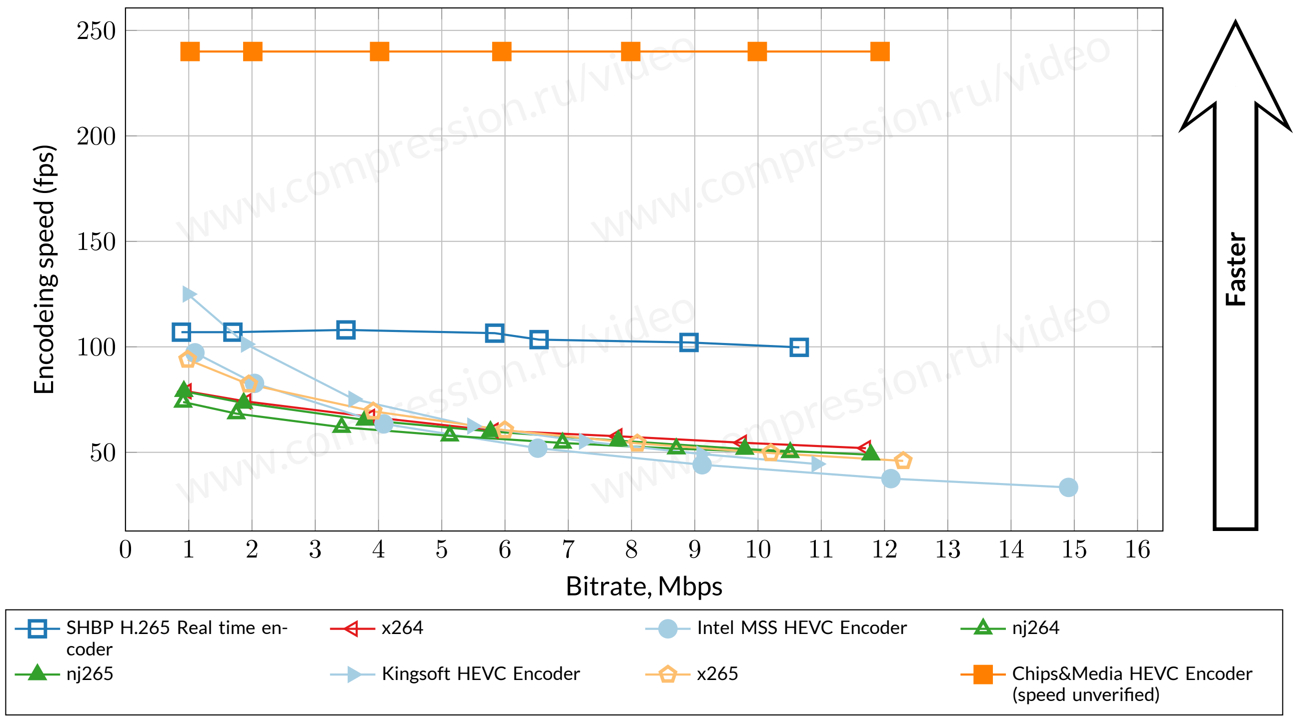


FIGURE 41: Encoding speed—usecase "Software and Hardware Encoders Comparison," Apple Tree sequence

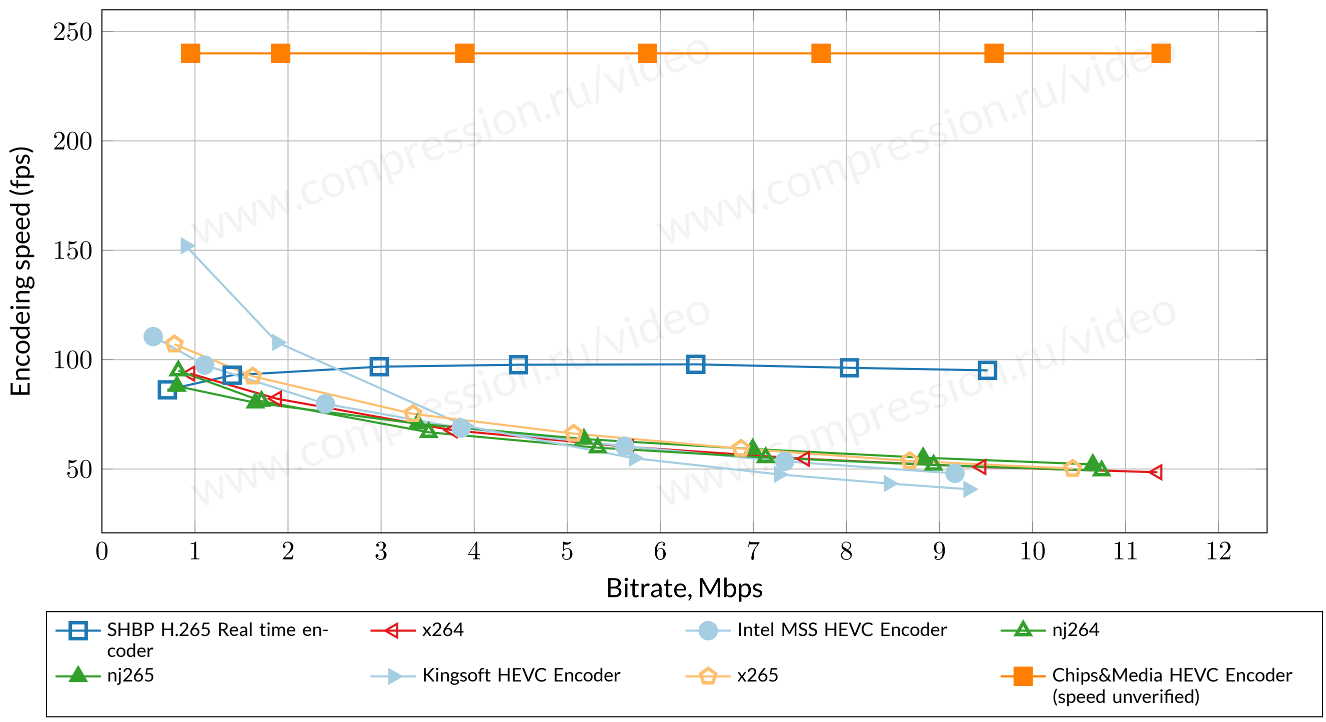


FIGURE 42: Encoding speed—usecase "Software and Hardware Encoders Comparison," Butterflies sequence

A.3 Speed/Quality Trade-Off

Chips&Media HEVC Encoder is in pareto-optimal (in terms speed/quality) encoders list with Kingsoft HEVC encoder and Intel MSS HEVC encoder.

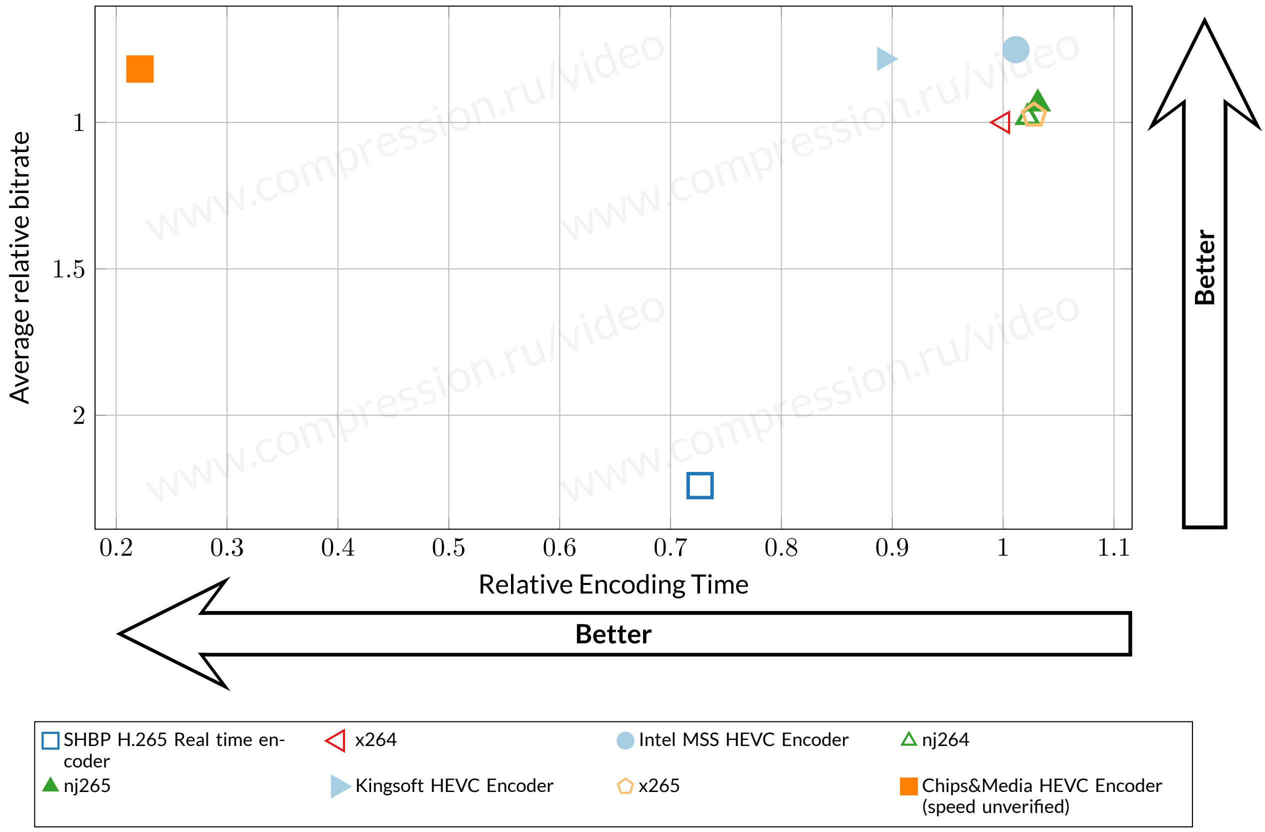


FIGURE 43: Speed/quality trade-off—usecase "Software and Hardware Encoders Comparison," all sequences, YUV-SSIM metric

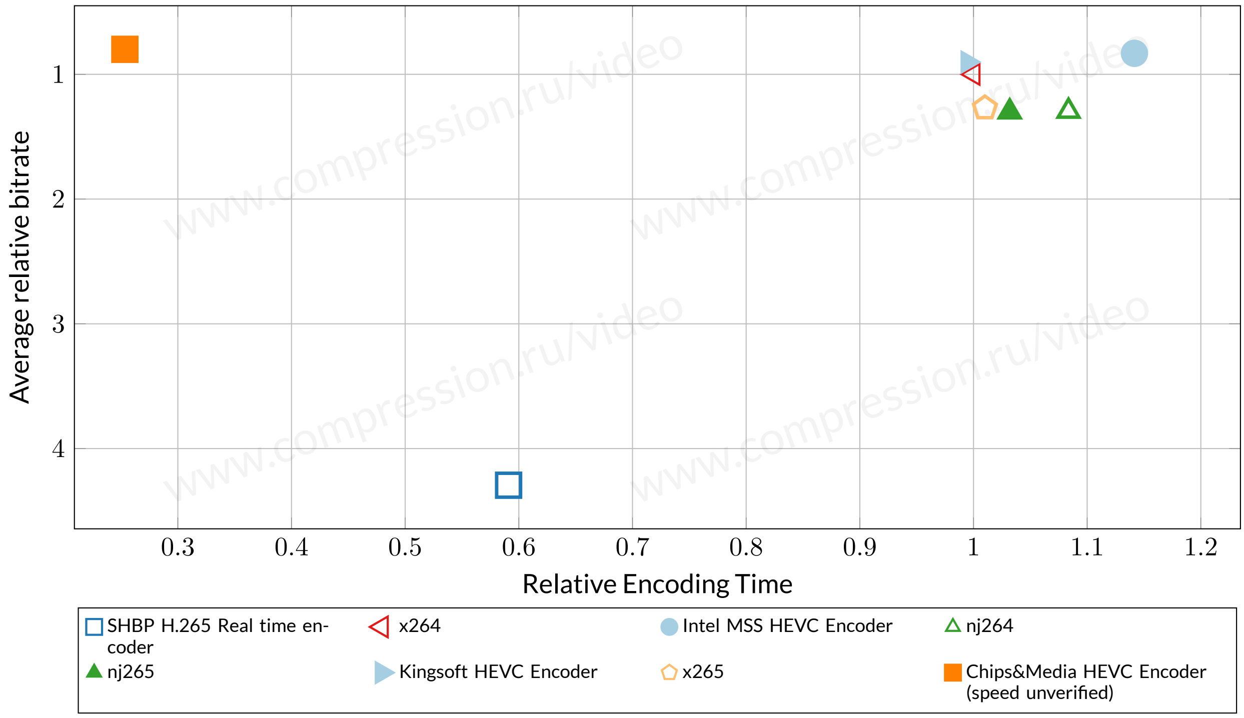


FIGURE 44: Speed/quality trade-off—usecase "Software and Hardware Encoders Comparison," Apple Tree sequence, YUV-SSIM metric

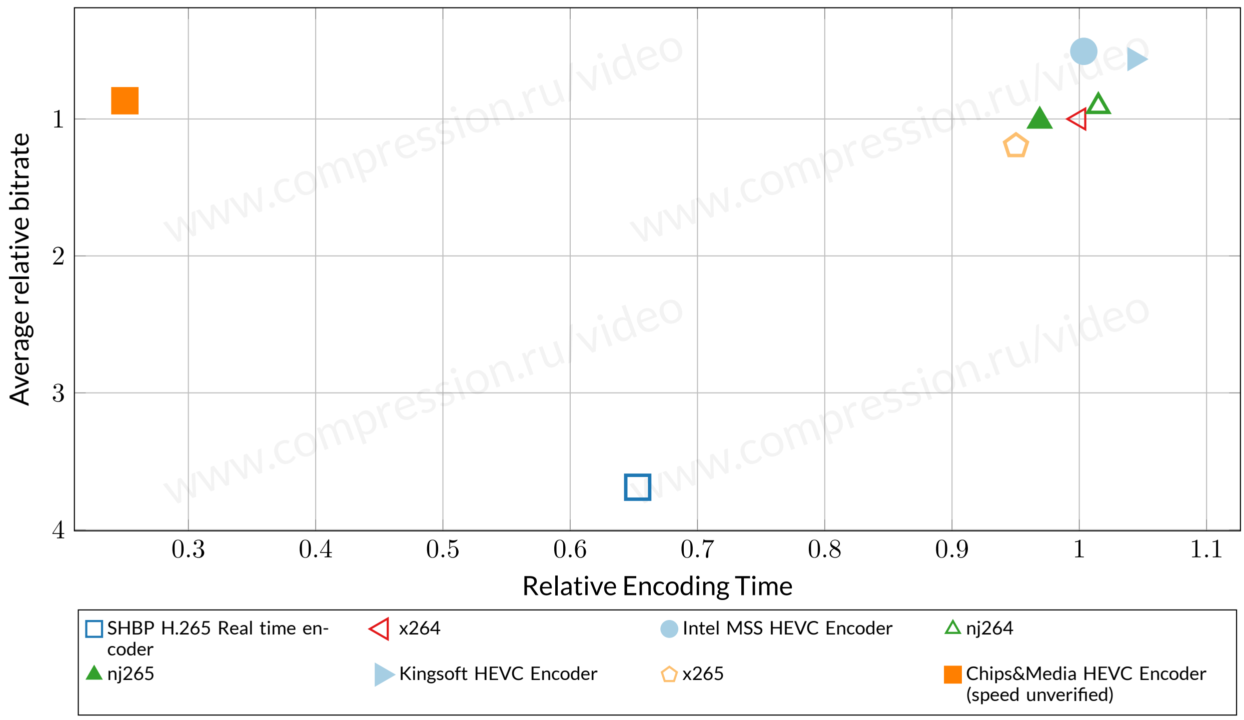


FIGURE 45: Speed/quality trade-off—usecase "Software and Hardware Encoders Comparison," Butterflies sequence, YUV-SSIM metric

A.4 Bitrate Handling

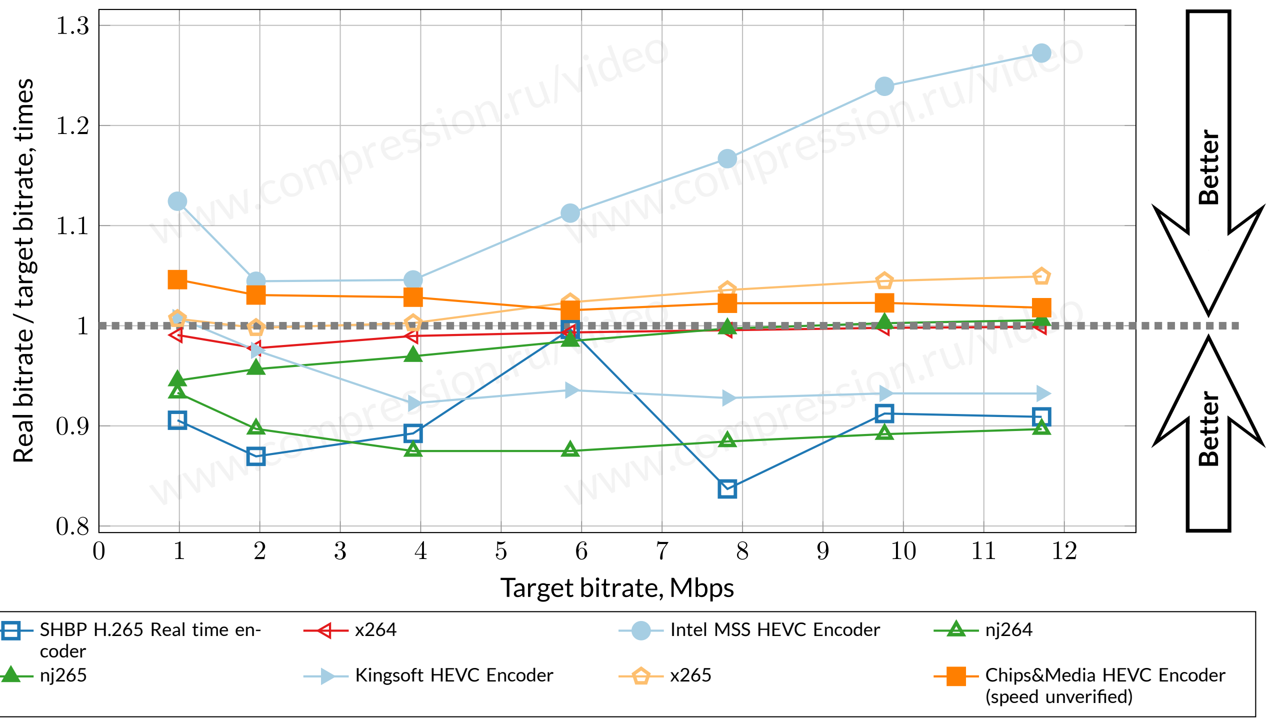


FIGURE 46: Bitrate handling—usecase "Software and Hardware Encoders Comparison," Apple Tree sequence

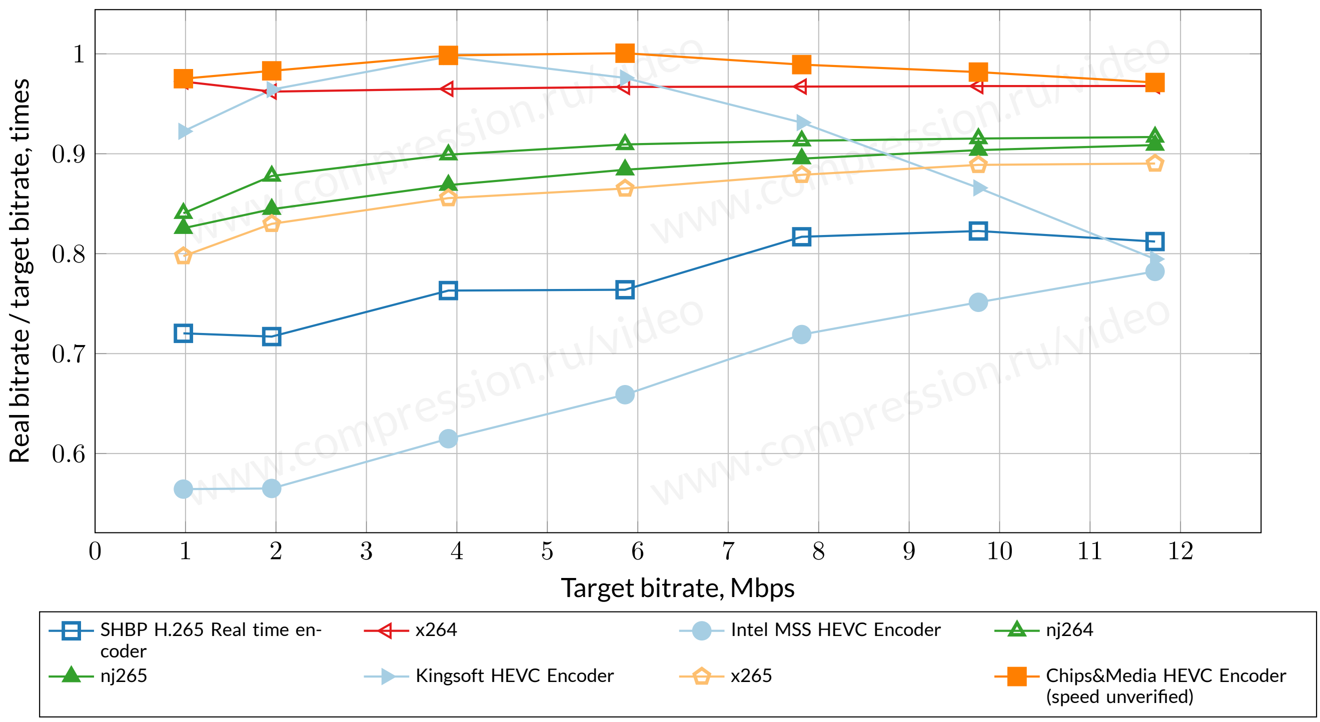


FIGURE 47: Bitrate handling—usecase "Software and Hardware Encoders Comparison," Butterflies sequence

A.5 Relative Quality Analysis

	SHBP H.265 Real time encoder	x264	Intel MSS HEVC Encoder	nj264	nj265	Kingsoft HEVC Encoder	x265	Chips&Media HEVC Encoder (speed unverified)
SHBP H.265 Real time encoder	100% ☺	45% ☺	29% ☺	46% ☺	38% ☺	31% ☺	39% ☺	32% ☺
x264	258% ☺	100% ☺	78% ☺	101% ☺	96% ☺	82% ☺	102% ☺	82% ☺
Intel MSS HEVC Encoder	411% 😐	133% ☺	100% ☺	135% ☺	127% ☺	107% ☺	137% ☺	109% ☺
nj264	266% ☺	102% ☺	78% ☺	100% ☺	98% ☺	82% ☺	105% ☺	83% ☺
nj265	295% ☺	107% ☺	80% ☺	109% ☺	100% ☺	86% ☺	106% ☺	87% ☺
Kingsoft HEVC Encoder	390% 😐	128% ☺	95% ☺	129% ☺	122% ☺	100% ☺	131% ☺	103% ☺
x265	284% 😐	103% ☺	77% ☺	105% ☺	96% ☺	82% ☺	100% ☺	83% ☺
Chips&Media HEVC Encoder (speed unverified)	358% 😐	122% ☺	94% ☺	125% ☺	119% ☺	100% ☺	126% ☺	100% ☺

Confidence ☹ 0% — 😐 50% — ☺ 100%

TABLE 6: Average bitrate ratio for a fixed quality—usecase "Software and Hardware Encoders Comparison," YUV-SSIM metric

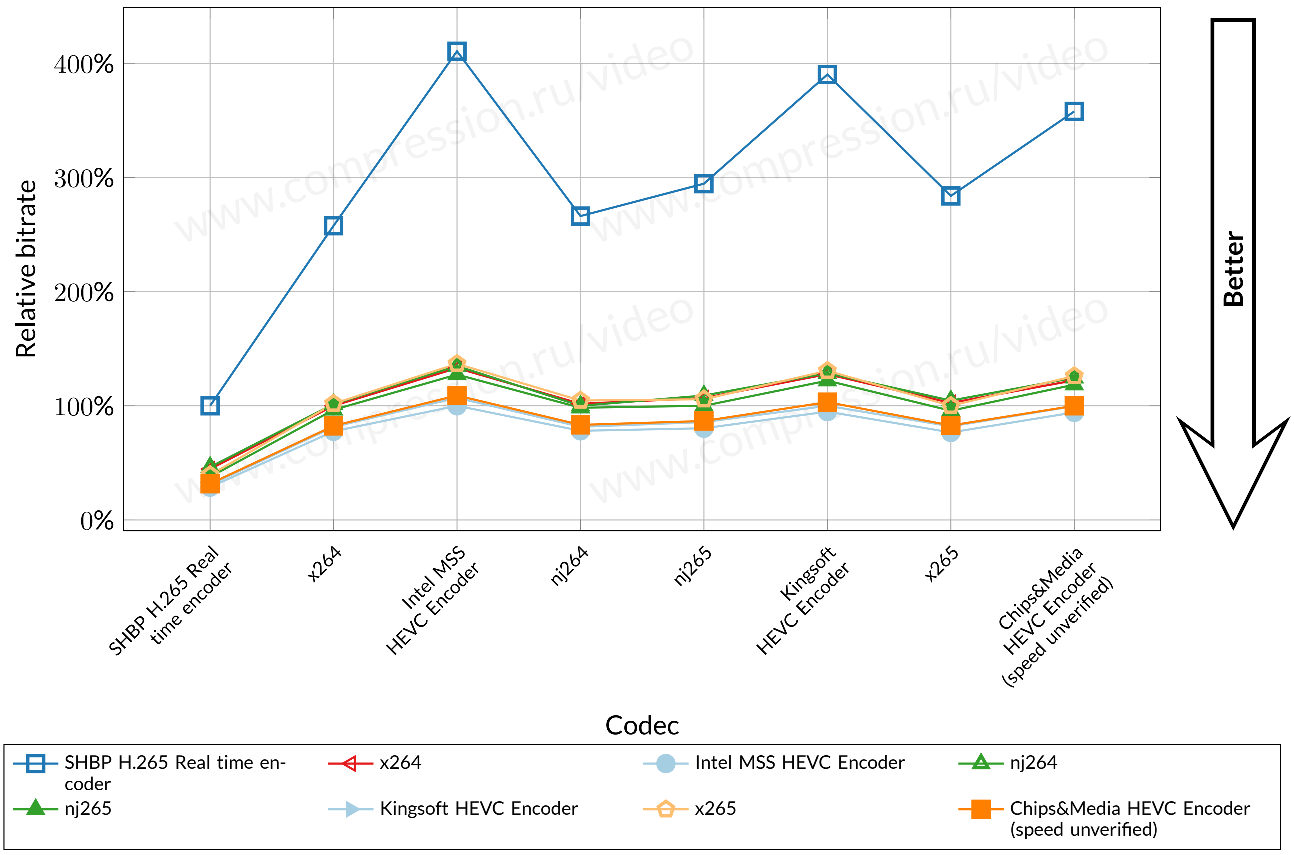


FIGURE 48: Average bitrate ratio for a fixed quality—usecase "Software and Hardware Encoders Comparison," YUV-SSIM metric

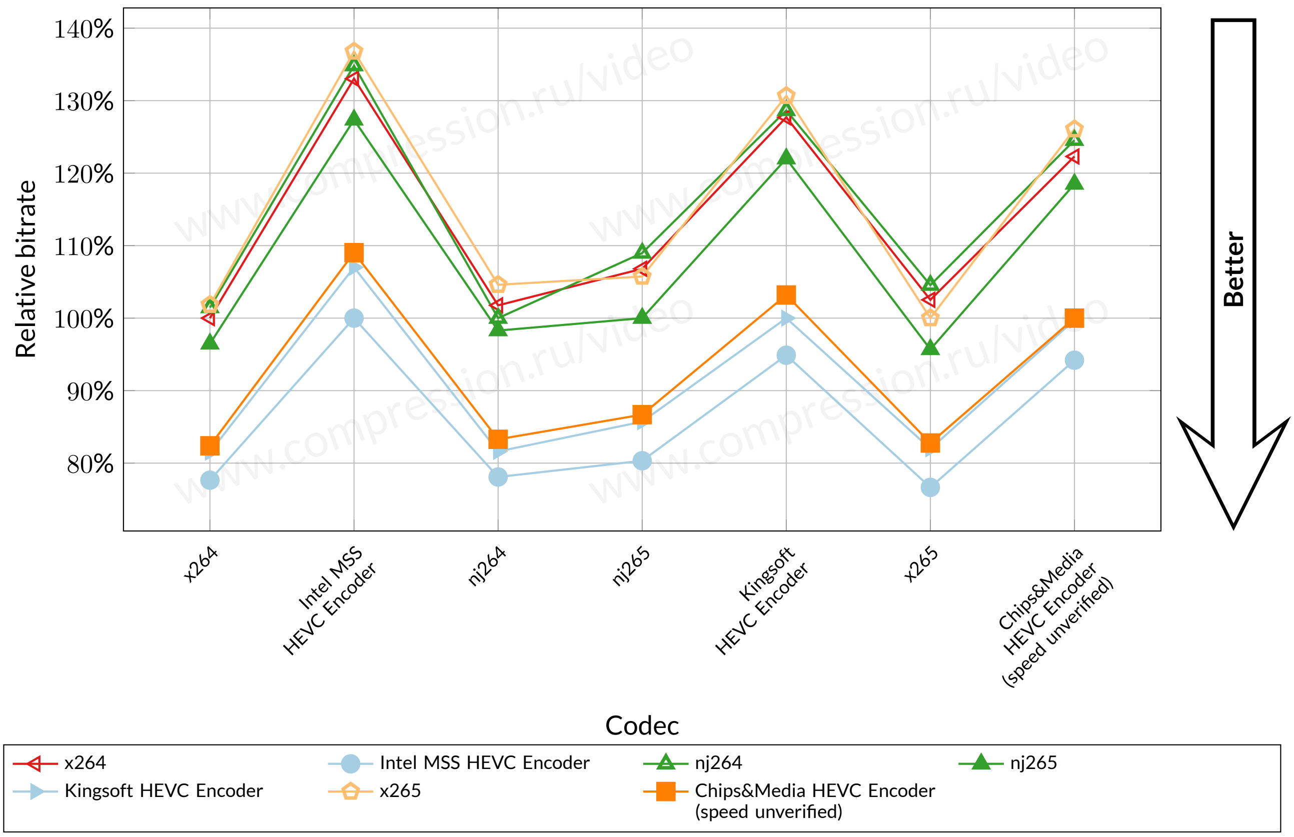


FIGURE 49: Average bitrate ratio for a fixed quality—usecase "Software and Hardware Encoders Comparison," YUV-SSIM metric, without SHBP H.265 Real time encoder

B PARTICIPANTS' COMMENTS

B.1 Kingsoft

Kingsoft HEVC encoder is majorly designed to support internet applications, including video livestreaming, video conferencing, video transcoding on mobile phones, personal computers and the cloud service. Over the past 2.5 years, Kingsoft HEVC encoder makes great improvement.

1. Regarding the speed problem of HEVC encoder, Kingsoft HEVC encoder is developed to optimize the encoder speed for real-time and living stream case. The Kingsoft HEVC encoder has achieved more than 30% bitrate saving over x264's veryfast to placebo presets at the similar encoding speed. We will take more effort to improve ripping case in future.

2. Due to the various internet video coding requirements, Kingsoft HEVC encoder is designed to compress internet videos to different kinds of resolutions, from QCIF to 4K.

3. Kingsoft HEVC encoder works better in livestraming and conferencing applications, because the rate control will be stricter in such cases.

B.2 x264

We selected the slowest encoder settings that met the speed targets of each of the fast/universal/ripping use cases.

x264 can achieve significantly faster encoding speeds with good results when speed is crucial.

C SEQUENCES

C.1 "Apple Tree"

Sequence title	Apple Tree
Resolution	1920×1080
Number of frames	338
Color space	YV12
Frames per second	30

Camera zooms out from an apple tree with an average speed.

FIGURE 50: Apple Tree sequence, frame 30

C.2 "Arashiyama"

Sequence title	Arashiyama
Resolution	1920×1080
Number of frames	1029
Color space	YV12
Frames per second	24
Source	https://vimeo.com/142480565#t=41

People are walking in the center of famous Japanese nature park Arashiyama. Static camera changes views several times.

FIGURE 51: Arashiyama sequence, frame 24

C.3 "Bicycle"

Sequence title	Bicycle
Resolution	1920×1080
Number of frames	780
Color space	YV12
Frames per second	25
Source	https://vimeo.com/158959434#t=49

Old man drives a bicycle through city views. Camera follows the man. Several times scene changes to one where this man is interviewed on a couch at some house. This scene has static camera.

FIGURE 52: Bicycle sequence, frame 25

C.4 "Bunny"

Sequence title	Bunny
Resolution	1920×1080
Number of frames	600
Color space	YV12
Frames per second	24

CG scene where squirrel flies over the forest and meets a bunny, then squirrel falls down.

FIGURE 53: Bunny sequence, frame 24

C.5 “Butterflies”

Sequence title	Butterflies
Resolution	1920×1080
Number of frames	1076
Color space	YV12
Frames per second	30
Source	https://vimeo.com/128901681#t=33

Many scenes with static camera and a butterfly sitting on a flower.

FIGURE 54: Butterflies sequence, frame 30

C.6 "Caterpillar"

Sequence title	Caterpillar
Resolution	1920×1080
Number of frames	113
Color space	YV12
Frames per second	30
Source	https://vimeo.com/122051624#t=0

Caterpillar climbs out of the cocoon. Static camera.

FIGURE 55: Caterpillar sequence, frame 30

C.7 "CG Forest"

Sequence title	CG Forest
Resolution	1920×1080
Number of frames	1184
Color space	YV12
Frames per second	30
Source	https://vimeo.com/150319285#t=0

Several views of CG forest from a star wars video game. Slow moving camera.

FIGURE 56: CG Forest sequence, frame 30

C.8 “Church Concert”

Sequence title	Church Concert
Resolution	1920×1080
Number of frames	1095
Color space	YV12
Frames per second	24
Source	https://vimeo.com/118449040#t=84

Many people staying near the scene where some guy playing guitar and singing. Camera flows a lot and rotates.

FIGURE 57: Church Concert sequence, frame 24

C.9 "City Crowd"

Sequence title	City Crowd
Resolution	1920×1080
Number of frames	763
Color space	YV12
Frames per second	30

People walking on a city street and in the center of scene a train runs into static camera.

FIGURE 58: City Crowd sequence, frame 30

C.10 "City Views"

Sequence title	City Views
Resolution	1920×1080
Number of frames	1025
Color space	YV12
Frames per second	24
Source	https://vimeo.com/118501540#t=39

Several views of city. Many of them include water. Camera is shaking a bit.

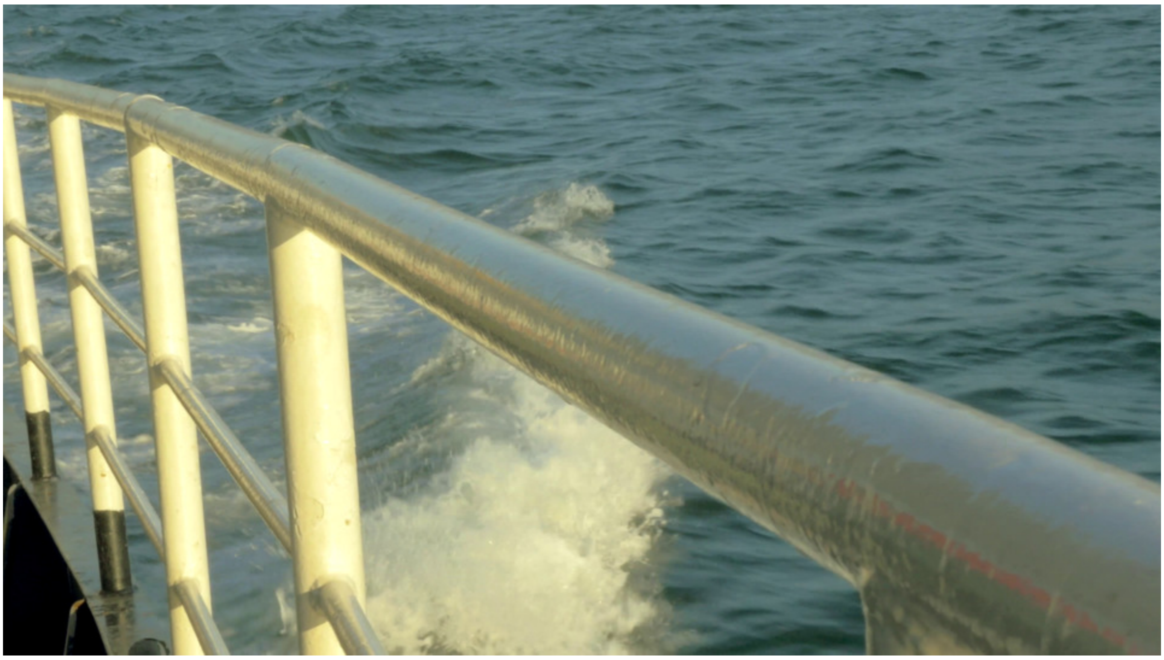


FIGURE 59: City Views sequence, frame 24

C.11 “Concert”

Sequence title	Concert
Resolution	1920×1080
Number of frames	1533
Color space	YV12
Frames per second	25

Dark scene of a captured rock concert with a lot of light flickering. Camera zooms in and out and shakes.

FIGURE 60: Concert sequence, frame 25

C.12 "Crowd Run"

Sequence title	Crowd Run
Resolution	1920×1080
Number of frames	500
Color space	YV12
Frames per second	25

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

FIGURE 61: Crowd Run sequence, frame 50

C.13 "Disneyland"

Sequence title	Disneyland
Resolution	1920×1080
Number of frames	317
Color space	YV12
Frames per second	24
Source	https://vimeo.com/152119430#t=0

Time lapse of disneyland castle located in a park with people. Camera slowly zooms in.

FIGURE 62: Disneyland sequence, frame 24

C.14 “Fire”

Sequence title	Fire
Resolution	1920×1080
Number of frames	601
Color space	YV12
Frames per second	25

Shooting of a bonfire. Static camera at the beginning and then it shakes a lot.

FIGURE 63: Fire sequence, frame 25

C.15 "Fishes"

Sequence title	Fishes
Resolution	1920×1080
Number of frames	1025
Color space	YV12
Frames per second	25
Source	https://vimeo.com/148326661#t=79

Several scenes of fishes swimming. Every scene camera follows a fish.

FIGURE 64: Fishes sequence, frame 25

C.16 "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

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

FIGURE 65: Fountain sequence, frame 25

C.17 "Neighborhood"

Sequence title	Neighborhood
Resolution	1920×1080
Number of frames	360
Color space	YV12
Frames per second	24
Source	https://vimeo.com/144904785#t=0

Video of several houses and a river captured by flying quadcopter.

FIGURE 66: Neighborhood sequence, frame 24

C.18 "Old Car"

Sequence title	Old Car
Resolution	1920×1080
Number of frames	1125
Color space	YV12
Frames per second	24
Source	https://vimeo.com/115884794#t=0

Different views of an old muscle car — its interior and engine.

FIGURE 67: Old Car sequence, frame 24

C.19 "Outdoor Party"

Sequence title	Outdoor Party
Resolution	1920×1080
Number of frames	1513
Color space	YV12
Frames per second	30
Source	https://vimeo.com/133945050#t=226

Children relax on a grass in a park. Camera shakes a bit.

FIGURE 68: Outdoor Party sequence, frame 30

C.20 “Park Walk”

Sequence title	Park Walk
Resolution	1920×1080
Number of frames	914
Color space	YV12
Frames per second	24
Source	https://vimeo.com/111171315#t=210

Views of a park, girl walking on a bridge and an interview of a man. Camera either static or slowly moving.

FIGURE 69: Park Walk sequence, frame 24

C.21 "Seacoast"

Sequence title	Seacoast
Resolution	1920×1080
Number of frames	1075
Color space	YV12
Frames per second	30
Source	https://vimeo.com/109321182#t=66

Woman walking near seacoast. Camera rotates over her and captures some views of a water.

FIGURE 70: Seacoast sequence, frame 30

C.22 "Shakewalk"

Sequence title	Shakewalk
Resolution	1920×1080
Number of frames	805
Color space	YV12
Frames per second	25

A man walking in the park and holding camera in front of him and shaking this camera a lot.

FIGURE 71: Shakewalk sequence, frame 25

C.23 “Ships”

Sequence title	Ships
Resolution	1920×1080
Number of frames	1041
Color space	YV12
Frames per second	25
Source	https://vimeo.com/137256075#t=119

Several views of people walking and taking photos of large ships.

FIGURE 72: Ships sequence, frame 25

C.24 "Steadicam"

Sequence title	Steadicam
Resolution	1920×1080
Number of frames	979
Color space	YV12
Frames per second	24
Source	https://vimeo.com/118449040#t=0

Interior of a church captured with steadicam.

FIGURE 73: Steadicam sequence, frame 96

C.25 "Sunset"

Sequence title	Sunset
Resolution	1920×1080
Number of frames	1190
Color space	YV12
Frames per second	24
Source	https://vimeo.com/129172145#t=161

A girl participates in photo session during sunset.

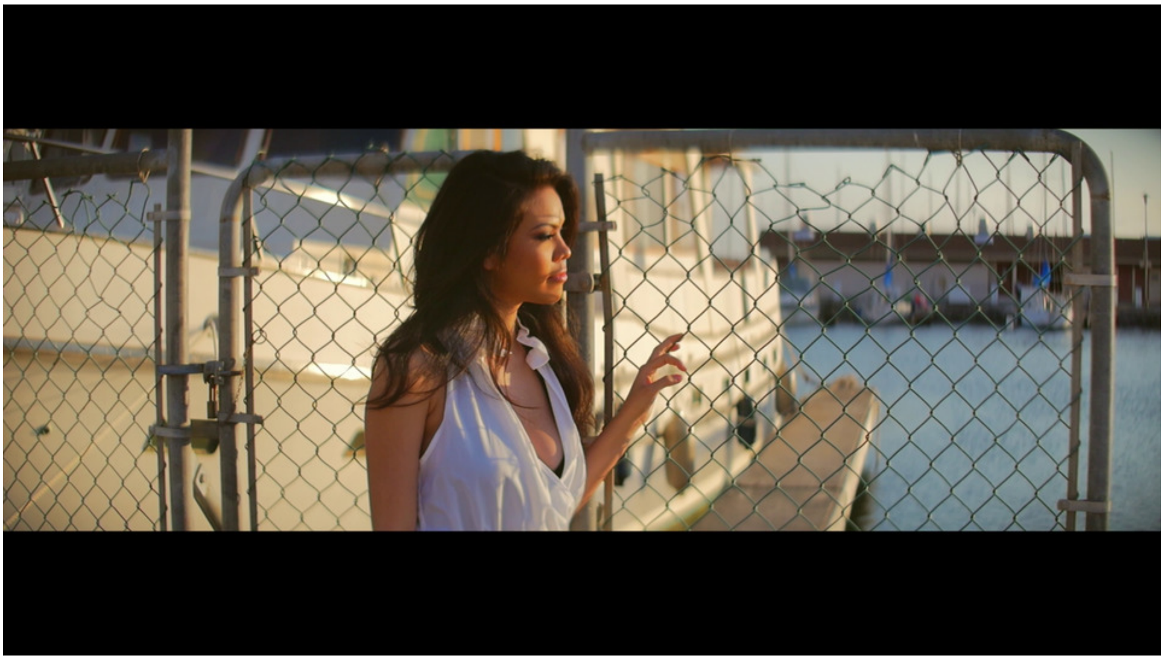


FIGURE 74: Sunset sequence, frame 24

C.26 "Water"

Sequence title	Water
Resolution	1920×1080
Number of frames	1209
Color space	YV12
Frames per second	25

Camera shoots water ripples, shakes and zooms in.

FIGURE 75: Water sequence, frame 25

C.27 “Winter”

Sequence title	Winter
Resolution	1920×1080
Number of frames	997
Color space	YV12
Frames per second	24
Source	https://vimeo.com/117702311#t=0

Camera slowly moves and shoots views of winter park.

FIGURE 76: Winter sequence, frame 24

D SEQUENCES SELECTION

Video sequences for all of MSU video codec comaprison reports were choosen by MSU team through manual selection. Various videos were selected to help to find the strengths and weaknesses of video encoders. This comparison's test dataset was significantly updated. Our goal was to create dataset with videos that encoders are facing in everyday life.

For this purpose, 30000 videos from Vimeo service were analyzed and 885 4K videos with high bitrate were downloaded. After that all videos were resized and cropped to FullHD resolution to ensure absence of compression artifacts. All videos were cut to samples, with 1000 frames approximate length. Incision points were located at scene change points. From 885 videos we created 2908 different samples.

To evaluate spatial and temporal complexity all samples were ecnoded using x264 encoder with constant quantization parameter (QP) equal to 28. For all samples temporal and spatial complexity were calculated. Spatial complexity was defined as average size of I-frame normalized by sample's uncompressed frame size. Temporal complexity was defined as average size of P-frame divided by average size of I-frame. Distribution of obtained samples compared to sequences from previous codec comparison dataset is shown in Figure 77.

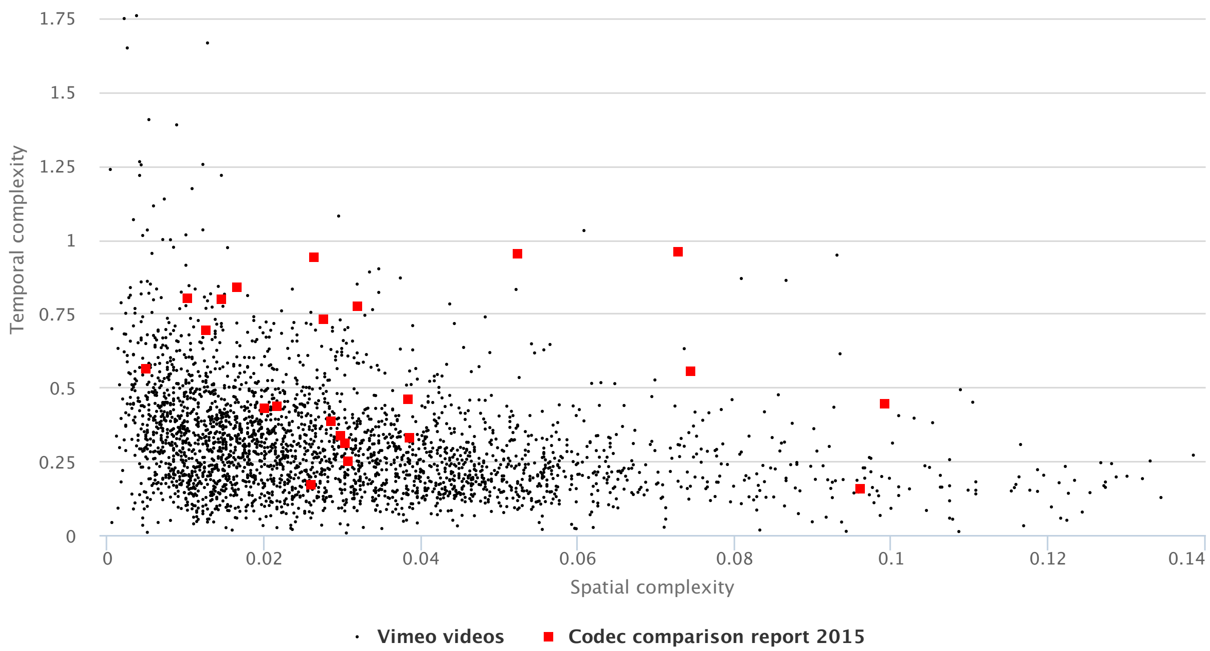


FIGURE 77: Distribution of all sequences

Figure 77 reveals, that dataset from MSU HEVC Comaprison 2015 (old dataset) doesn't fully cover all distribution. In order to prepare dataset with better coverage the following process was used.

We divided the space into 30 clusters with K-Means. Sequences from old dataset were given 10 times higher weight than Vimeo sequences to avoid complete update of sequences list. For each cluster we selected video sequence closest to its center having license enabling derivatives and commercial usage. The cluster's boundaries and chosen sequences are shown at Figure 78.

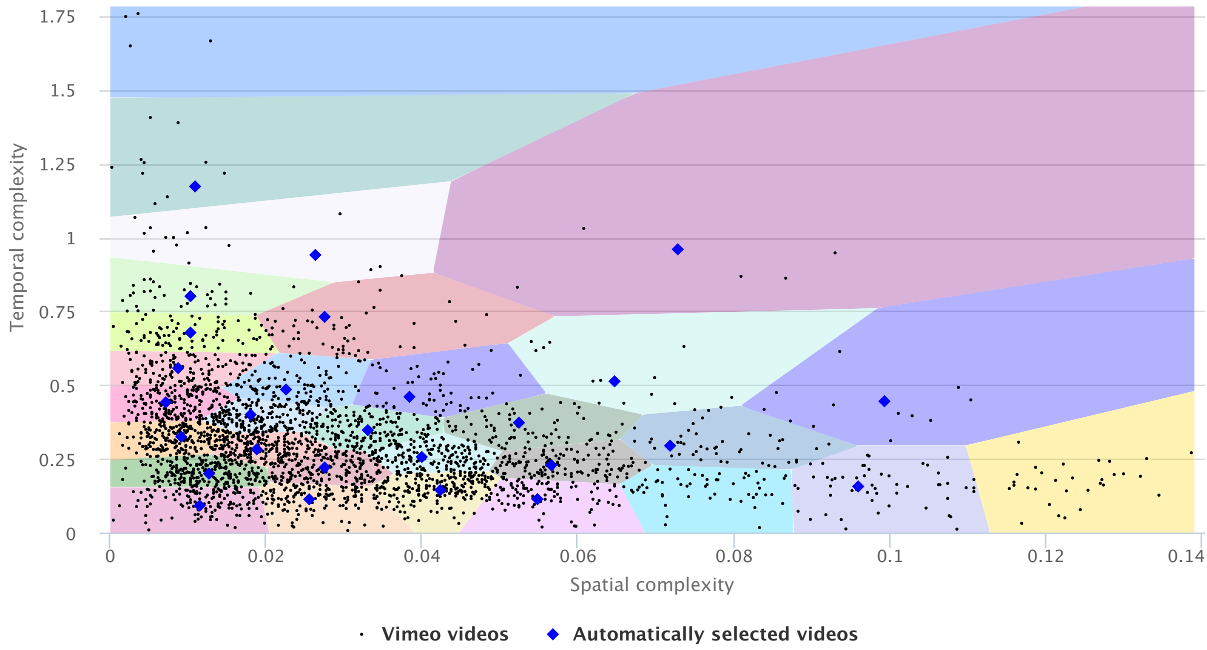


FIGURE 78: Segmentation of the sequence space

At Figure 79 we visualize how sequences from old dataset correspond to cluster boundaries. There are many clusters not covered by videos from old dataset, and there are some clusters that have more than one video.

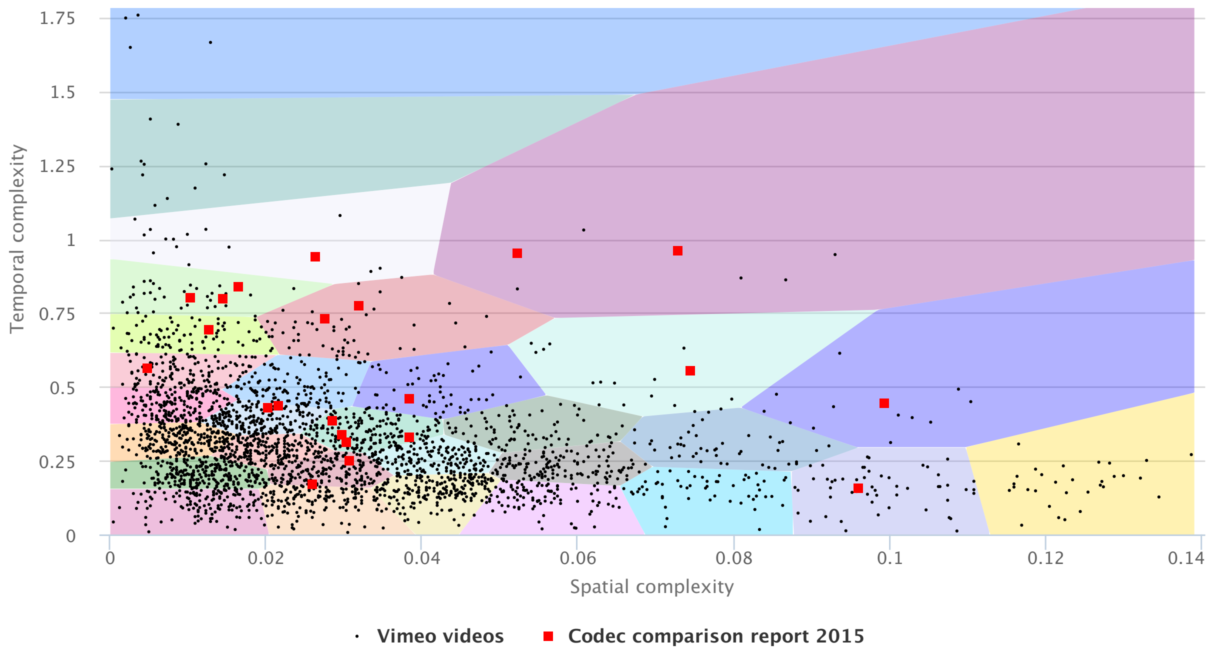


FIGURE 79: Segmentation of the sequence space compared to old dataset

Some of automatically chosen samples contain company names or have other copyright issues, so we replaced that samples with other samples in that clusters with suitable license. Figure 80 illustrates applied adjustments.

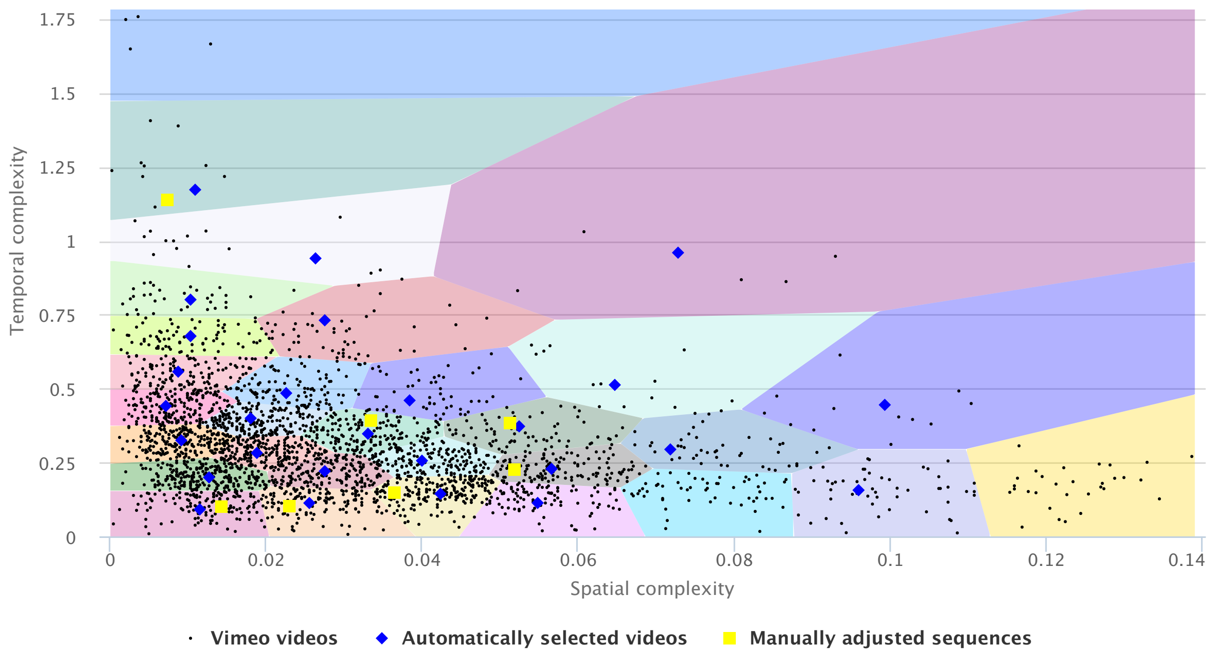


FIGURE 80: Adjustments to test dataset

Figure 81 shows final distribution of sequences in dataset used in this report.

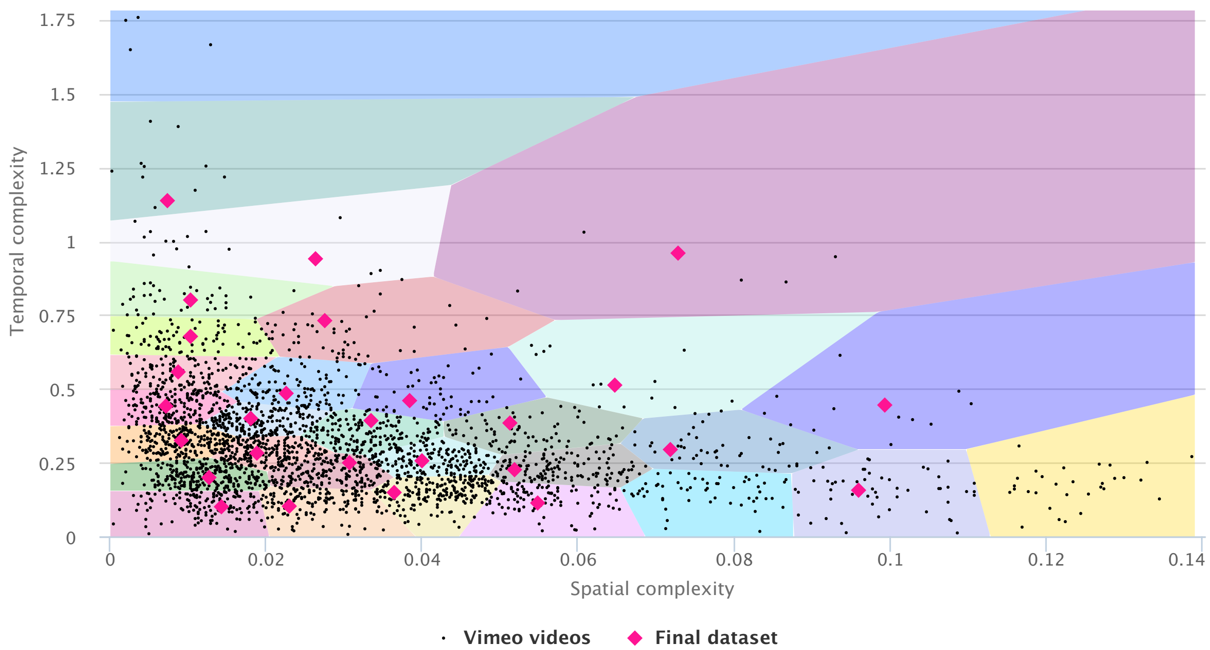


FIGURE 81: Distribution of sequences in final dataset

New dataset consists of 27 video sequences, 8 videos from old dataset, and 19 new videos from Vimeo. 13 sequences from old dataset were excluded.

E CODECS

E.1 Intel® Media Server Studio HEVC GPU-accelerated Encoder

Encoder title	Intel® Media Server Studio HEVC GPU-accelerated Encoder
Version	Intel® Media Server Studio 2017 R1
Developed by	Intel

Preset name	Encoder parameters
Fast	`mfx_transcoder.exe h265 -encode_plugin mfxplugin64_hevce_gacc.dll -hw -sys -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 6 -async 3`
Universal	`mfx_transcoder.exe h265 -encode_plugin mfxplugin64_hevce_gacc.dll -hw -sys -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 4 -async 3`
Ripping	`mfx_transcoder.exe h265 -encode_plugin mfxplugin64_hevce_sw.dll -mfxdll libmfxsw64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 1 -async 3`

E.2 SHBP H.265 Real time encoder

Encoder title	SHBP H.265 Real time encoder
Version	0.8
Developed by	SHBP Codec's development team

```
Usage: sh_hevc_enc.exe <options>

  -help         display this information
  -i <s>        s - input YUV filename
  -w <n>        n - input frames width
  -h <n>        n - input frames height
  -f <f>        f - frames per second value (25.0)
  -n <n>        n - number of frames to encode(0 - all)
  -o <s>        s - output binary filename
  -r <s>        s - reconstructed YUV filename(none)
  -c <s>        s - config txt filename with advanced parameters(none)
  -id <n>       n - input device id(0) (0 - file, 1 - hw emu)
  -od <n>       n - output device id(0) (0 - file, 1 - hw emu)
  -b <n>        n - target bitrate in kb per second
  -g <n>        n - GOP size in frames (104)
  -q <n>        n - quantization parameter [1, 51] (disabled -b option)
  -p <f>        f - performance level in fps (0 - auto)
```

FIGURE 82: SHBP H.265 Real time encoder

Preset name	Encoder parameters
Fast	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 60.0 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
Universal	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 25.0 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`

E.3 x264

Encoder title	x264
Version	148 r2665 a01e399
Developed by	x264 Developer Team

```
x264 core:148 r2638 7599210
Syntax: x264 [options] -o outfile infile

Infile can be raw (in which case resolution is required),
  or YUV4MPEG (*.y4m),
  or Avisynth if compiled with support (yes).
  or libav* formats if compiled with lavf support (yes) or ffms support (no).
Outfile type is selected by filename:
 .264 -> Raw bytestream
 .mkv -> Matroska
 .flv -> Flash Video
 .mp4 -> MP4 if compiled with GPAC or L-SMASH support (no)
Output bit depth: 8 (configured at compile time)

Options:

  -h, --help                  List basic options
      --longhelp              List more options
      --fullhelp              List all options
```

FIGURE 83: x264 encoder

Preset name	Encoder parameters
Fast	`x264 --preset fast --subme 5 --b-adapt 0 --keyint infinite --tune ssim --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o NUL` `x264 --preset fast --subme 5 --b-adapt 0 --keyint infinite --tune ssim --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o %TARGET_FILE%`
Universal	`x264 --preset slow --me hex --trellis 2 --keyint infinite --tune ssim --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o NUL` `x264 --preset slow --me hex --trellis 2 --keyint infinite --tune ssim --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o %TARGET_FILE%`
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.4 x265

Encoder title	x265
Version	1.9+169-e5b5bdc3c154
Developed by	x265 Developer Team

```
Syntax: x265 [options] infile [-o] outfile
    infile can be YUV or Y4M
    outfile is raw HEVC bitstream

Executable Options:
-h/--help                         Show this help text and exit
-V/--version                      Show version info and exit

Output Options:
-o/--output <filename>            Bitstream output file name
-D/--output-depth 8|10|12         Output bit depth (also internal bit depth)
```

FIGURE 84: x265 encoder

Preset name	Encoder parameters
Fast	`x265 -p ultrafast --tune ssim --me 1 --ref 2 --limit-refs 3 --signhide --b-intra --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
Universal	`x265 -p medium --tune ssim --rd 2 --early-skip --bframes 3 --max-merge 3 --ref 4 --b-intra --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
Ripping	`x265 -p veryslow --tune ssim --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`

E.5 Chips&Media Hevc Encoder

Encoder title	Chips&Media Hevc Encoder
Version	1.3 (r5650)
Developed by	Chips&Media, Inc.

```
HEVC Encoder Version [1.3][r]

Usage: TAppEncoder.exe [OPTION]

  -c, --ConfigFile                        Defines configuration file to use.
                                          Multiple configuration files may be us
ed with repeated -c options

  -i, --InputFile                         Specifies the input video file.
                                          Video data must be in a raw 4:2:0 plan
ar format (Y0CbCr).
```

FIGURE 85: Chips&Media Hevc Encoder

E.6 nj264

Encoder title	nj264
Version	1.0
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
Fast	`nj264.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj264 -preset speed -nj264-params bitrate=%BITRATE_KBPS% -f h264 -y %TARGET_FILE%`
Universal	`nj264.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj264 -preset quality -nj264-params bitrate=%BITRATE_KBPS% -f h264 -y %TARGET_FILE%`
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.7 nj265

Encoder title	nj265
Version	1.0
Developed by	Nanjing Yunyan

Preset name	Encoder parameters
Fast	`nj265.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj265 -preset ultrafast -nj265-params bitrate=%BITRATE_KBPS% -f hevc -y %TARGET_FILE%`
Universal	`nj265.exe -s %WIDTH%x%HEIGHT% -framerate %FPS% -i %SOURCE_FILE% -c:v libnj265 -preset balanced -nj265-params bitrate=%BITRATE_KBPS% -f hevc -y %TARGET_FILE%`
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.8 KS265

Encoder title	Kingsoft Encoder
Version	V2.1.1
Developed by	Kingsoft

Preset name	Encoder parameters
Fast	`AppEncoder_x64.exe -i %SOURCE_FILE% -preset medium -wdt %WIDTH% -hgt %HEIGHT% -fr %FPS% -rc 1 -br %BITRATE_KBPS% -iper 300 -b %TARGET_FILE% -iNxN 0 -mrgnum 3 -sao 1 -ctx 0 -imdI 0 -ltr 0`
Universal	`AppEncoder_x64.exe -i %SOURCE_FILE% -preset slow -wdt %WIDTH% -hgt %HEIGHT% -fr %FPS% -rc 1 -br %BITRATE_KBPS% -iper 300 -b %TARGET_FILE% -part 0`
Ripping	`AppEncoder_x64.exe -i %SOURCE_FILE% -preset veryslow -wdt %WIDTH% -hgt %HEIGHT% -fr %FPS% -rc 1 -br %BITRATE_KBPS% -iper 300 -b %TARGET_FILE% -cfm 0 -fdointra 0 -estop 0 -cu-costd 0 -goup 0`

F FIGURES EXPLANATION

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

F.1 RD Curves

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

F.2 Relative Bitrate/Relative Time Charts

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

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

F.3 Graph Example

Figure 86 shows a case where these graphs can be useful. In the top left graph, it is apparent that the "Green" codec encodes with significantly 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: it is clearly visible in the bottom graph that one of the codecs is slower, but yields higher visual quality, and that the other codec is faster, but yields lower visual quality.

As a result of these advantages, relative bitrate/relative time graphs are used frequently in this report since they assist in the evaluation of the codecs in the test set, especially when number of codecs is large.

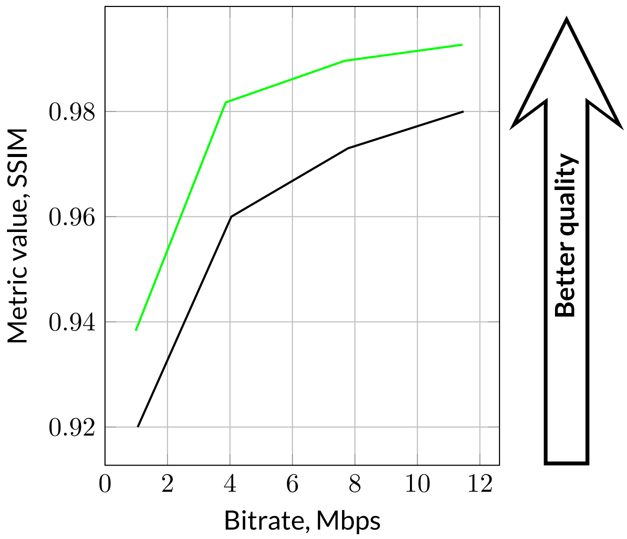
A more detailed description of the preparation of these graphs is given below.

F.4 Bitrates Ratio with the Same Quality

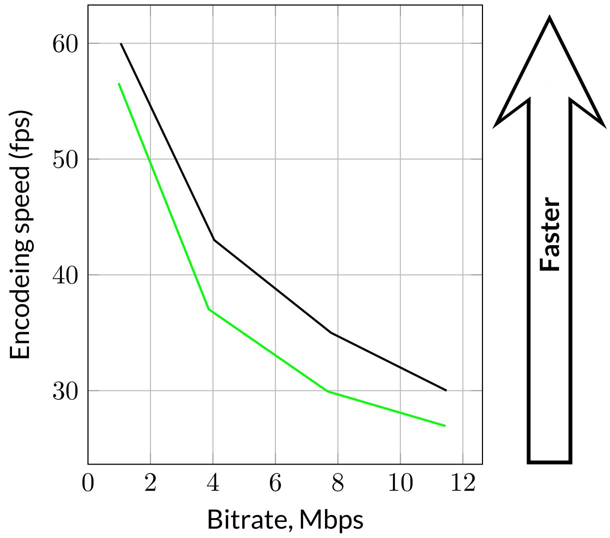
The first step in computing the average bitrate ratio for a fixed quality is inversion of the axes of the bitrate/quality graph (see Figure 87b). All further computations are performed using the inverted graph.

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

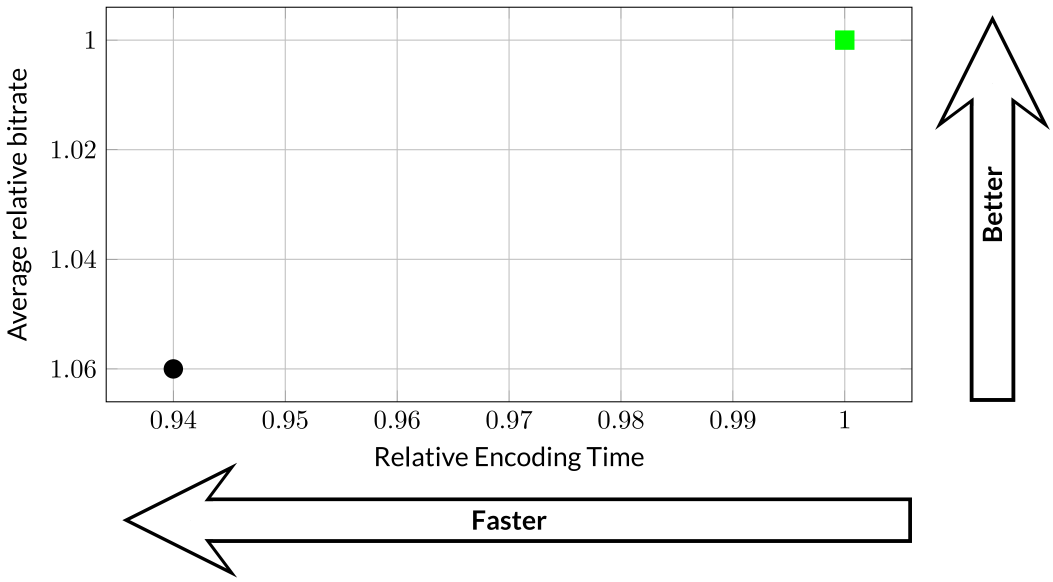
The final step is calculation of the area under the curves in the chosen interpolation segment and determination of their ratio (see Figure 87c). This result is an average bitrate ratio for a fixed quality for the two codecs. If



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

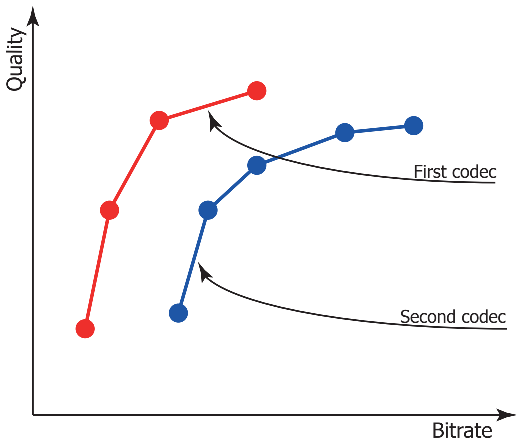


(b) Encoding speed (frames per second). "Green" codec is slower!

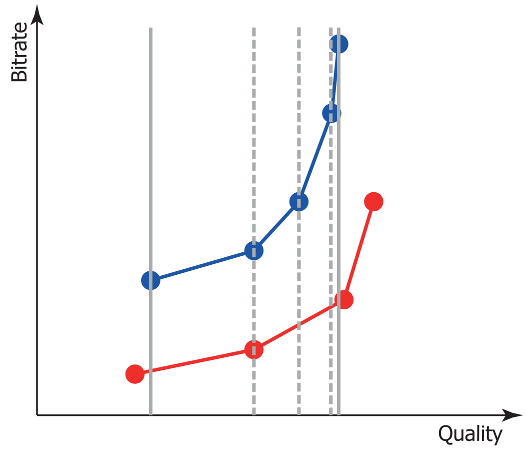


(c) Integral situation with codecs. This plot shows the situation more clearly

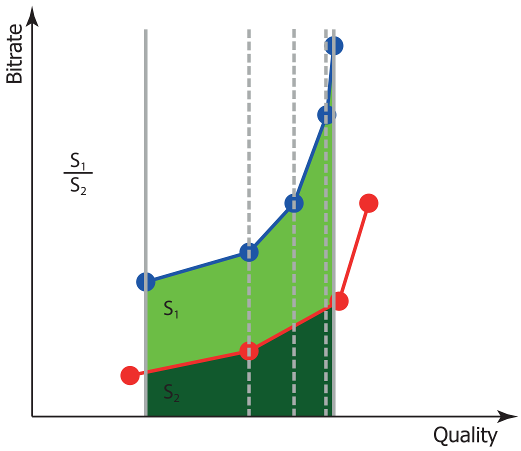
FIGURE 86: Speed/Quality trade-off example



(a) Source RD curves



(b) Axes' inversion and averaging interval choosing



(c) Areas under curves ratio

FIGURE 87: Average bitrate ratio computation

more than two codecs are considered, then one of them is defined as a reference codec and the quality of others is compared to that of the reference.

F.5 Relative Quality Analysis

While most figures in this report provide codec scores relative to reference encoder (i.e. x264) the "Relative Quality Analysis" sections show bitrate ratio with fixed quality (see Section F.4) score for each codec pair. This might be useful if one is interested in comparison of codec A relative to codec B only.

Below we show simplified example of "Average bitrate ratio for a fixed quality" table for two codecs only:

	A	B
A	100% ☺	75% ☹
B	134% ☹	100% ☺

TABLE 7: Example of average bitrate ratio for a fixed quality table

Let's consider column "B" row "A" of the table containing value 75% this should be read in the following way: average bitrate for a fixed quality of codec B is 75% less relative to codec A. The icon in the cell depicts confidence of this estimate. If projections of codecs' RD curves on quality axis (see Figure 87) have relatively large common area you will see happy icon. If size of this intersection is small and thus bitrate score can't be computed reliably the sad icon will be shown.

"Average bitrate ratio for a fixed quality" plots are visualizations these tables. Each line in such plot depicts values from one column of corresponding table.

G OBJECTIVE QUALITY METRICS DESCRIPTION

G.1 SSIM (Structural SIMilarity)

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

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

Brief description of SSIM metric computation is given below.

G.1.1 Brief Description

The original paper on the SSIM metric was published by Wang, et al.[1] The paper can be found at `http://ieeexplore.ieee.org/iel5/83/28667/01284395.pdf`. The SSIM author homepage is found at `http://www.cns.nyu.edu/~lcv/ssim/`

The scheme of SSIM calculation can be presented as follows. 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 the following:

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

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 grayscale images), and $K1, K2 \ll 1$.

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

For the implementation used in this comparison, 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 of

[1] 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.

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

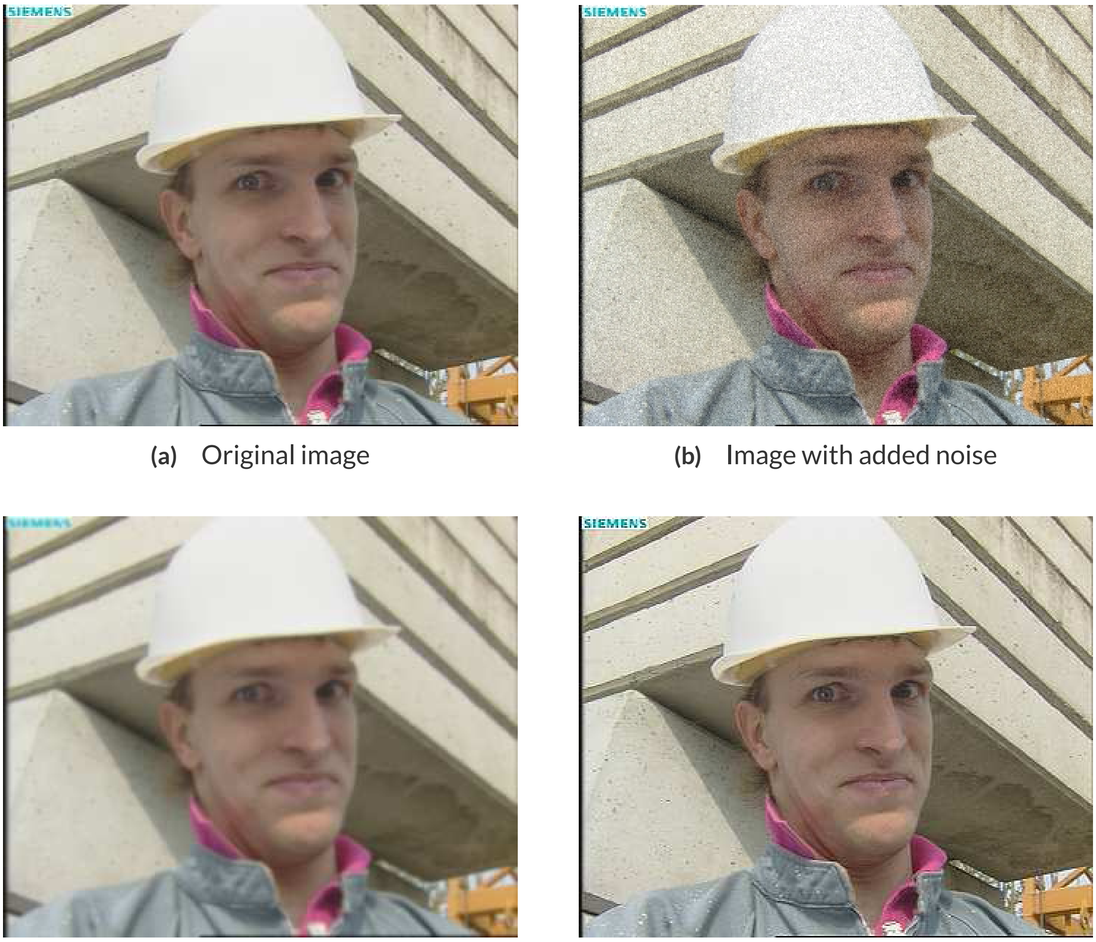
G.1.2 Examples

Figure 88 shows the example of an SSIM result for an original and processed (compressed with lossy compression) image. The resulting value of 0.9 demonstrates that the two images are very similar.

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

FIGURE 88: SSIM example for compressed image

Figure 89 depicts various distortions applied to original image and Figure 90 shows SSIM values for these distortions.



(a) Original image

(b) Image with added noise

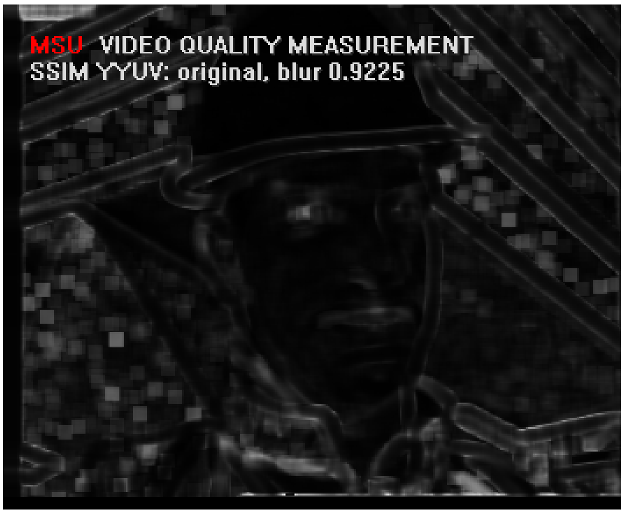
(c) Blurred image

(d) Sharpen image

FIGURE 89: Examples of processed images

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

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



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

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

FIGURE 90: SSIM values for original and processed images

H 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`