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, Vladimir Yanushkovsky

Free version

Codecs:

H.265

- f265 H.265 Encoder
- Intel MSS HEVC GAcc
- Intel MSS HEVC Software
- Ittiam HEVC Hardware Encoder
- Ittiam HEVC Software Encoder
- Strongene Lentoid HEVC Encoder
- SHBP H.265 Real time encoder
- x265

Non H.265

- InTeleMax TurboEnc
- SIF Encoder
- VP9 Video Codec
- x264

CS MSU Graphics & Media Lab, Video Group
October 15, 2015

`http://www.compression.ru/video/codec_comparison/index_en.html`
`videocodec-testing@graphics.cs.msu.ru`

Contents

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:

- InTeleMax, Inc.
- Intel Corporation
- Ittiam Sysytems (P) Ltd.
- Strongene Ltd.
- ”System house ”Business partners” company
- SIF Encoder developper team
- The WebM Project team
- x264 developer team
- x265 developer team

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

2 DISCLAIMER

This is free version of the report which contains only overall results for desktop and server platform and detailed analysis of codecs performance on one video sequence for desktop platform. For each individual sequence detailed plots and quality scores computed using various quality metrics please refer to Enterprise version of this report (http://compression.ru/video/codec_comparison/hevc_2015/#version_comparison).

3 OVERVIEW

3.1 Sequences

	Sequence	Number of frames	Frame rate	Resolution
1.	Apple Tree	338	30	1920 × 1080
2.	Bunny	600	24	1920 × 1080
3.	City Crowd	763	30	1920 × 1080
4.	Concert	1533	25	1920 × 1080
5.	Day Cars	1299	25	1920 × 1080
6.	Developers	1500	30	1280 × 720
7.	Fire	601	25	1920 × 1080
8.	Golden Statue	1993	30	1920 × 1080
9.	Hockey	1000	25	1920 × 1080
10.	Kremlin	1899	25	1920 × 1080
11.	Market Walk	688	30	1920 × 1080
12.	Mountain View	398	30	1920 × 1080
13.	Night Cars	1305	25	1920 × 1080
14.	Pine Tree	1130	30	1920 × 1080
15.	River Boats	1061	30	1920 × 1080
16.	Road	877	25	1920 × 1080
17.	Shake Walk	805	25	1920 × 1080
18.	Sita	1000	25	1920 × 1080
19.	Trigans	10500	30	1920 × 1080
20.	Water	1209	25	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 A.

3.2 Codecs

	Codec	Developer	Version
1.	InTeleMax TurboEnc	InTeleMax, Inc.	3.0
2.	f265 H.265 Encoder	f265 Developer Team	0.2
3.	Intel MSS HEVC GAcc	Intel	Intel Media Server Studio 2015 R4 – Professional Edition (release date: April 2015)
4.	Intel MSS HEVC Software	Intel	Intel Media Server Studio 2015 R4 – Professional Edition (release date: April 2015)
5.	Ittiam HEVC Hardware Encoder	Ittiam Systems (P) Ltd.	2_04_4_00
6.	Ittiam HEVC Software Encoder	Ittiam Systems (P) Ltd.	1_14_8_06
7.	Strongene Lentoid HEVC Encoder	Strongene Ltd.	2.2
8.	SHBP H.265 Real time encoder	SHBP Codec's development team Email: lobasso@hotmail.com	
9.	SIF Encoder	SIF Encoder Team Email: info@sifcodec.com	1.30.4
10.	VP9 Video Codec	The WebM Project	1.3.0
11.	x264	x264 Developer Team	146 r2538 121396c
12.	x265	MulticoreWare, Inc.	1.5+460-ac85c775620f

TABLE 2: Short codec 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 B.

4 OBJECTIVES AND TESTING RULES

4.1 HEVC Codec Testing Objectives

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. The comparison was done using settings provided by the developers of each codec. Nevertheless, we required all presets to satisfy minimum speed requirement on the particular use case. The main task of the comparison is to analyze different encoders for the task of transcoding video—e.g., compressing video for personal use.

4.2 Testing Rules

The comparison was performed on two platforms:

- *Desktop*—Core i7 4770R @3.9 GHz, RAM 4 GB, Windows 8.1
- *Server*—Xeon E5 2697v3, RAM 64 GB, Windows Server 2012

For both platforms we considered three key use cases with different speed requirements.

- Desktop
  - Ripping—no minimum speed
  - Universal—minimum 10 FPS
  - Fast transcoding—minimum 30 FPS
- Server
  - Ripping—no minimum speed
  - Universal—minimum 30 FPS
  - Fast transcoding—minimum 60 FPS

5 DESKTOP COMPARISON

5.1 Fast Transcoding

5.1.1 RD curves

The plot below contains RD-curves for one test video sequence. For codecs' performance on other video sequences please refer to Enterprise version of this report `http://compression.ru/video/codec_comparison/hevc_2015/#version_comparison`.

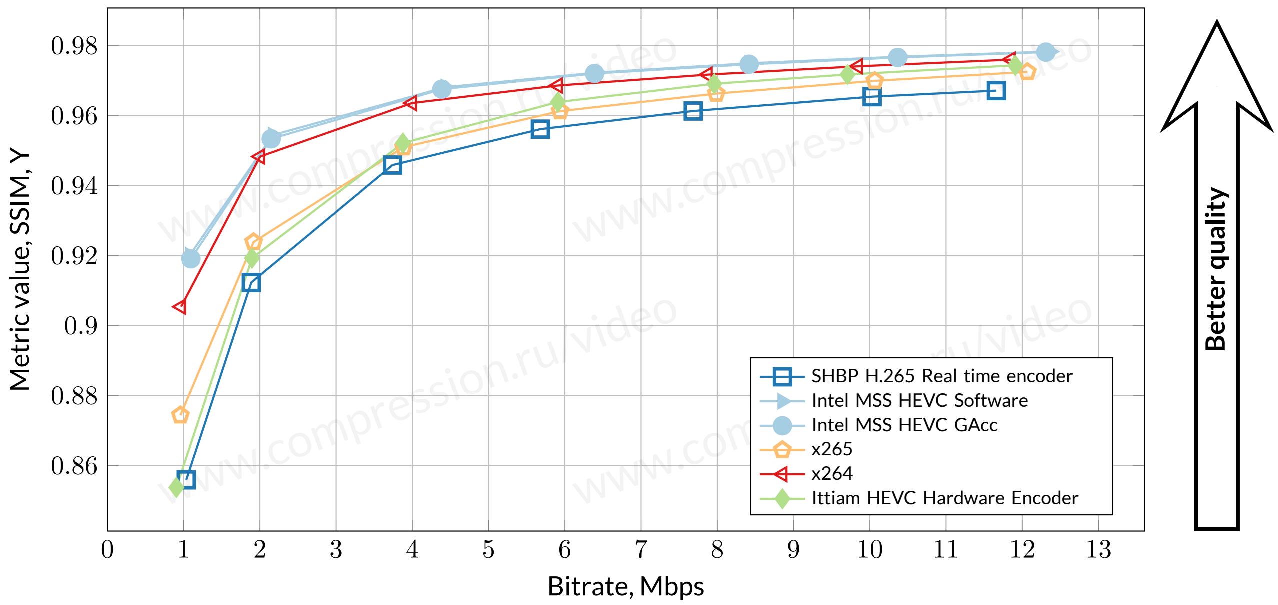


FIGURE 1: Bitrate/quality—usecase "Fast Transcoding," Apple Tree sequence, Y-SSIM metric

5.1.2 Encoding Speed

Figure below shows how participating codecs differ in encoding speed on "Apple Tree" video sequence.

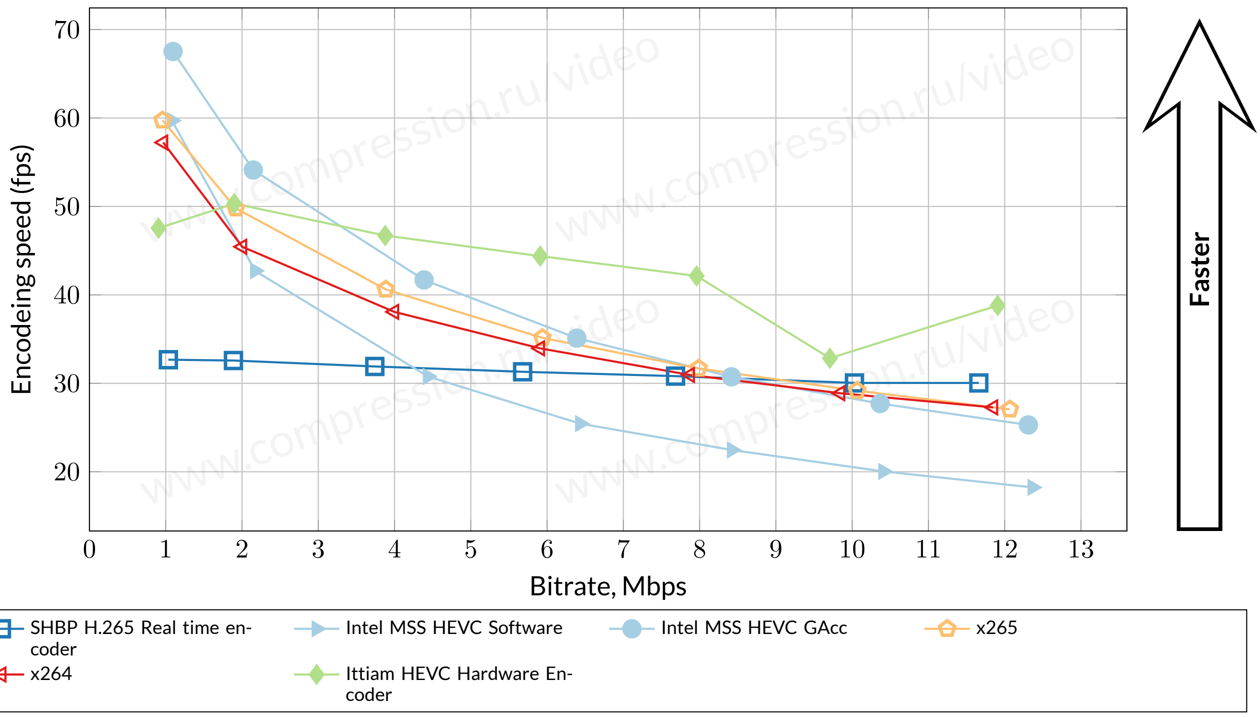


FIGURE 2: Encoding speed—usecase "Fast Transcoding," Apple Tree sequence

5.1.3 Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs can be found in Appendix C. 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.

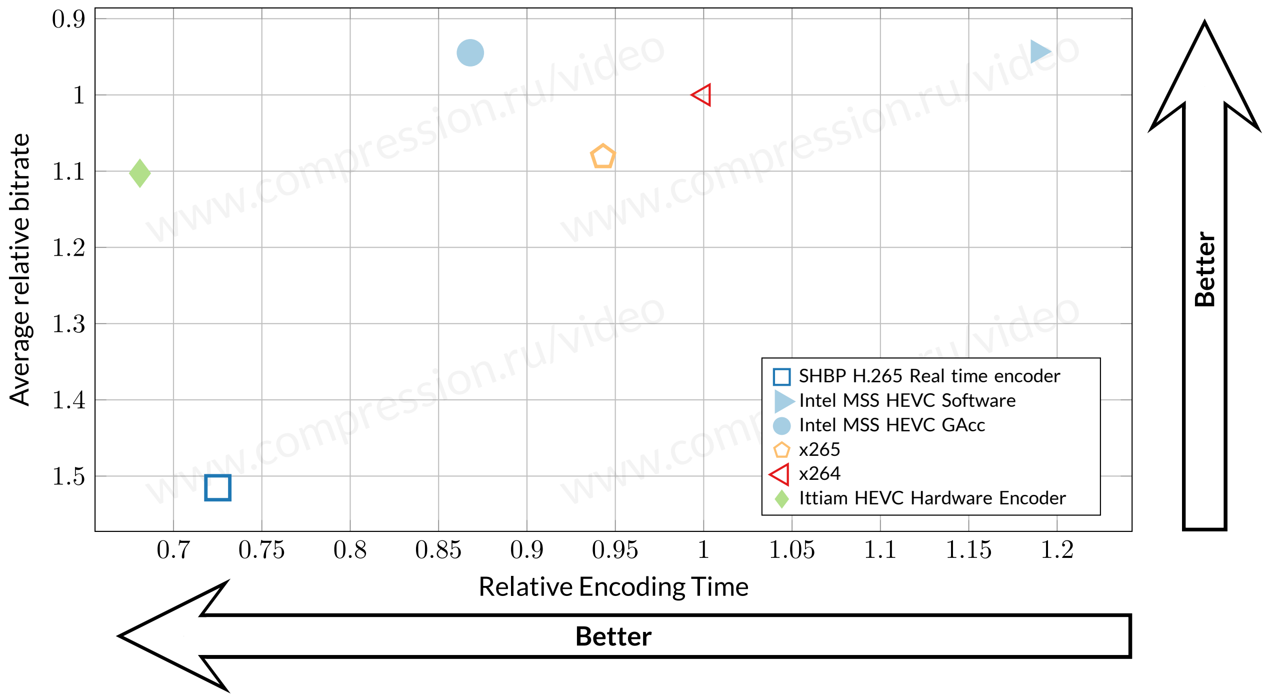


FIGURE 3: Speed/quality trade-off—usecase "Fast Transcoding," all sequences, Y-SSIM metric

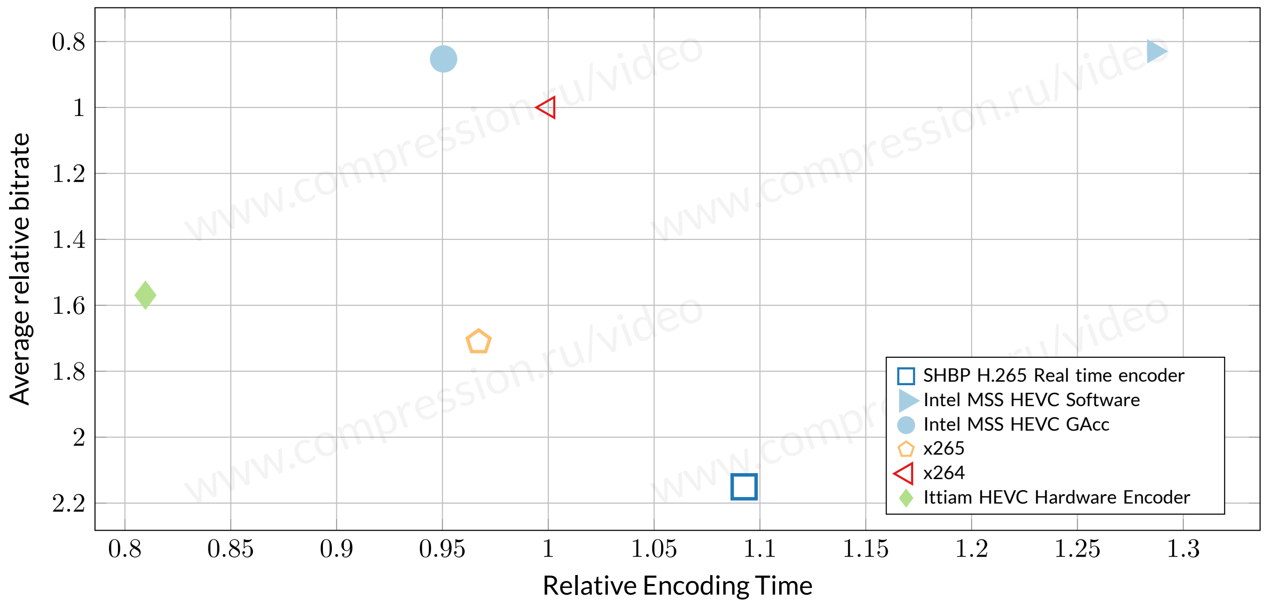


FIGURE 4: Speed/quality trade-off—usecase "Fast Transcoding," Apple Tree sequence, Y-SSIM metric

5.1.4 Bitrate Handling

The plot below shows how accurately encoded stream's real bitrate matches bitrate requested by user on "Apple Tree" video sequence.

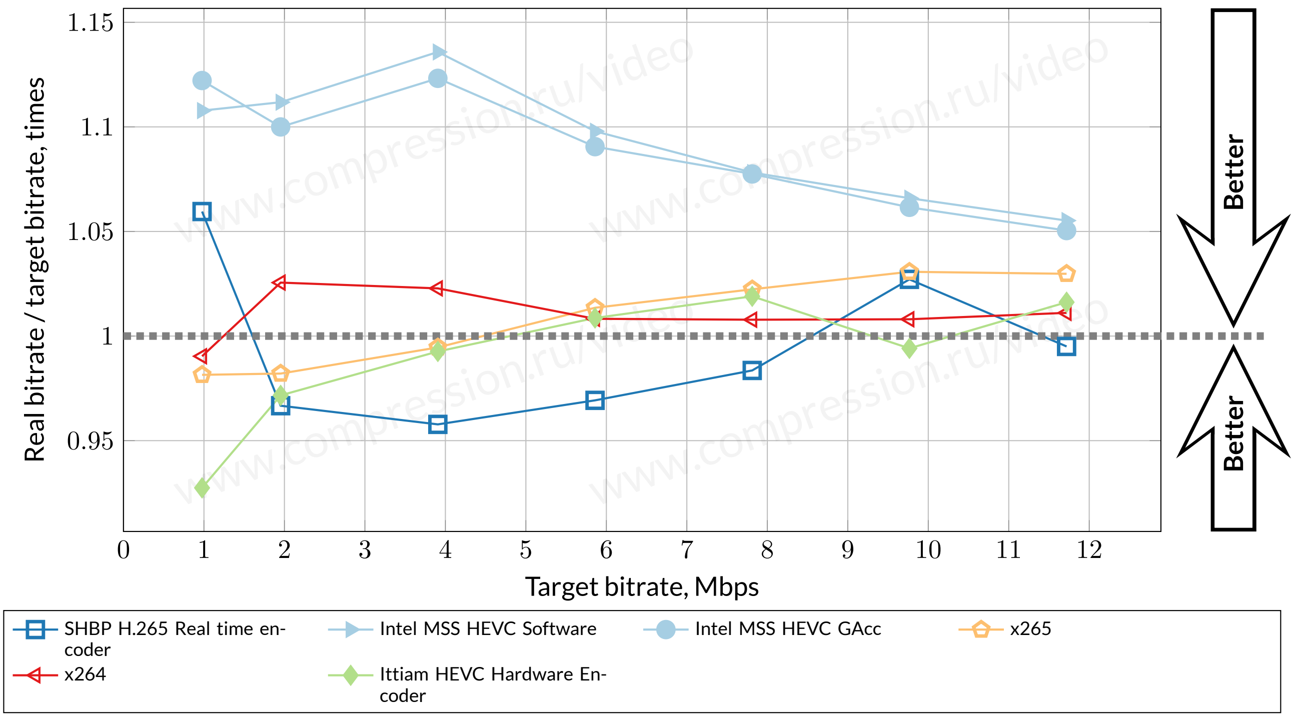


FIGURE 5: Bitrate handling—usecase "Fast Transcoding," Apple Tree sequence

5.1.5 Relative Quality Analysis

Note that each number in the tables below corresponds to some range of bitrates (see Appendix C). 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. Please see Section C.4 for explanation how to read table and plot below.

	SHBP H.265 Real time encoder	Intel MSS HEVC Software	Intel MSS HEVC GAcc	x265	x264	Ittiam HEVC Hardware Encoder
SHBP H.265 Real time encoder	100% ☺	58% ☺	58% ☺	69% ☺	66% ☺	67% ☺
Intel MSS HEVC Software	210% ☺	100% ☺	100% ☺	120% ☺	106% ☺	120% ☺
Intel MSS HEVC GAcc	210% ☺	100% ☺	100% ☺	120% ☺	106% ☺	120% ☺
x265	189% ☺	87% ☺	87% ☺	100% ☺	92% ☺	103% ☺
x264	186% ☺	97% ☺	97% ☺	117% ☺	100% ☺	114% ☺
Ittiam HEVC Hardware Encoder	180% ☺	86% ☺	85% ☺	102% ☺	91% ☺	100% ☺

TABLE 3: Average bitrate ratio for a fixed quality—usecase "Fast Transcoding," Y-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.

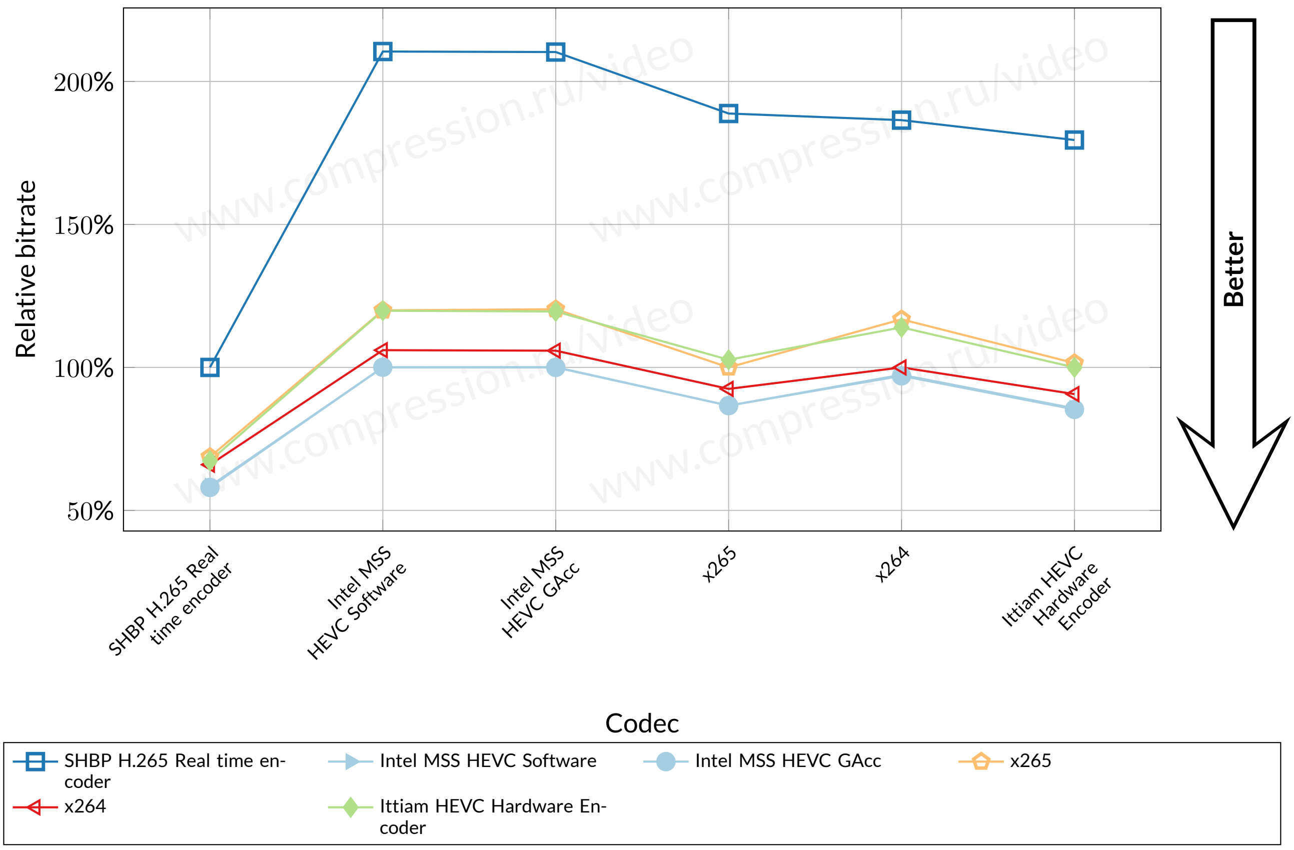


FIGURE 6: Average bitrate ratio for a fixed quality—usecase "Fast Transcoding," Y-SSIM metric

For visual purposes we show the same plot without SHBP H.265 Real time encoder below.

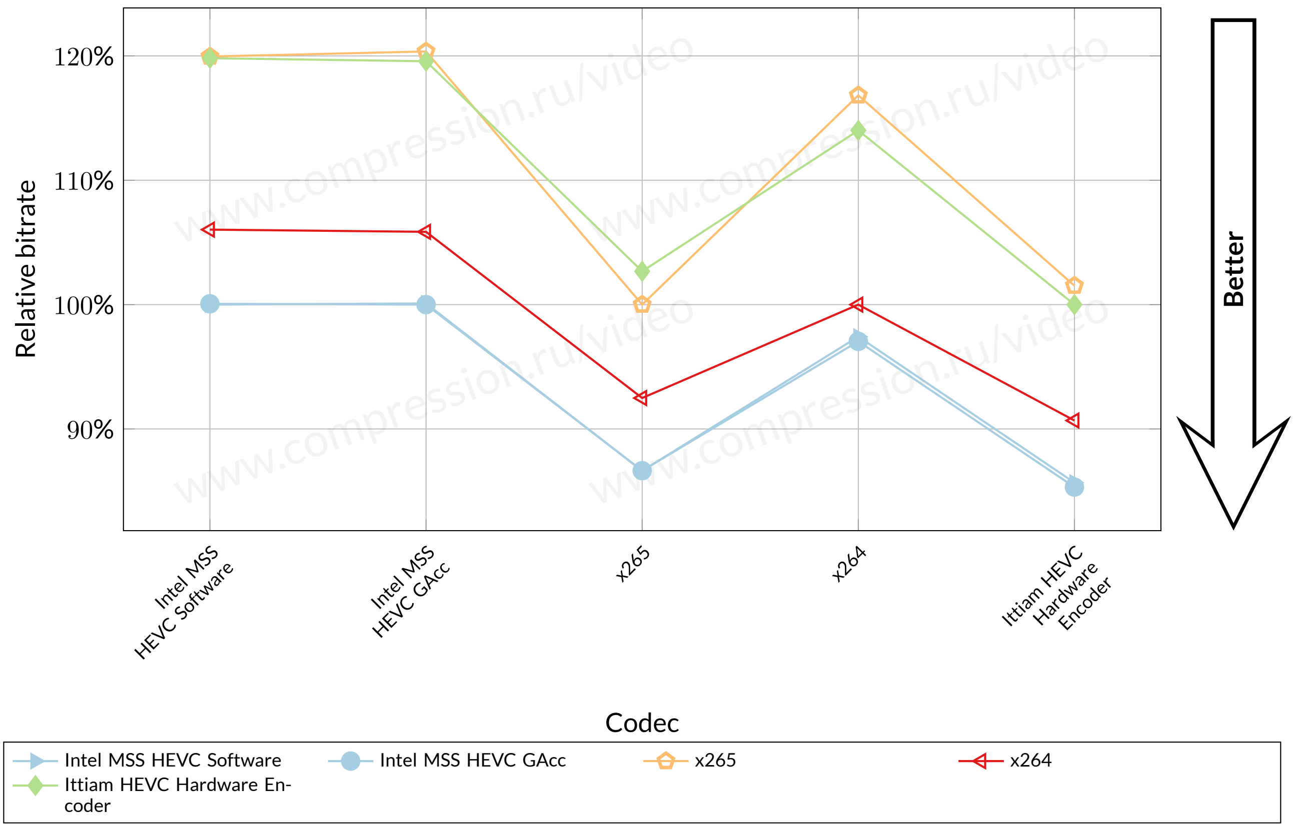


FIGURE 7: Average bitrate ratio for a fixed quality—usecase “Fast Transcoding,” Y-SSIM metric, without SHBP H.265 Real time encoder

5.2 Universal

Now we move to Universal use case which imposes weaker speed requirements (codecs should process at least 10 frames per second on reference video sequence).

5.2.1 RD curves

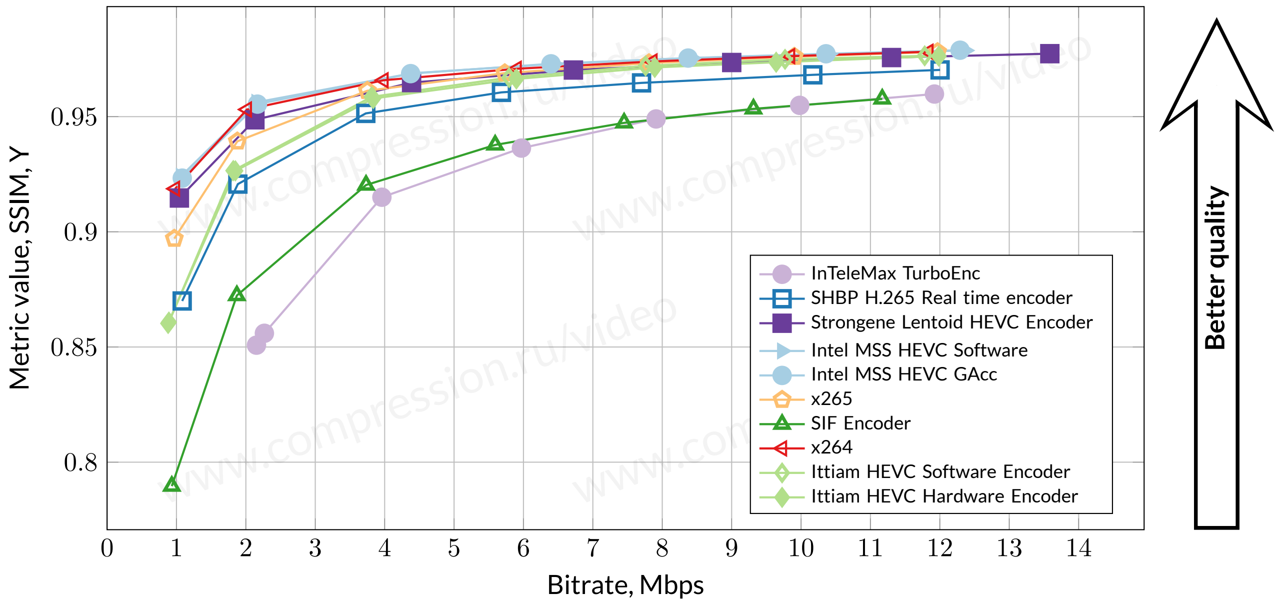


FIGURE 8: Bitrate/quality—usecase "Universal," Apple Tree sequence, Y-SSIM metric

5.2.2 Encoding Speed

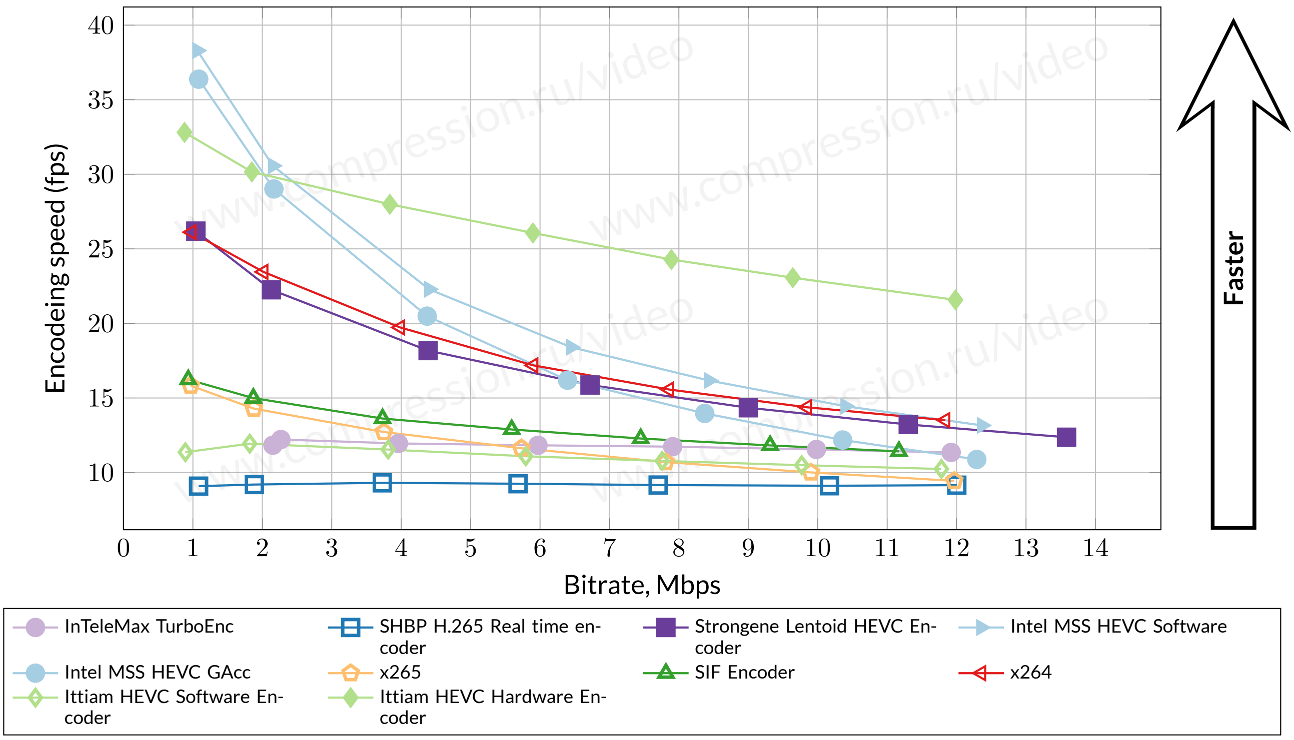


FIGURE 9: Encoding speed—usecase "Universal," Apple Tree sequence

5.2.3 Speed/Quality Trade-Off

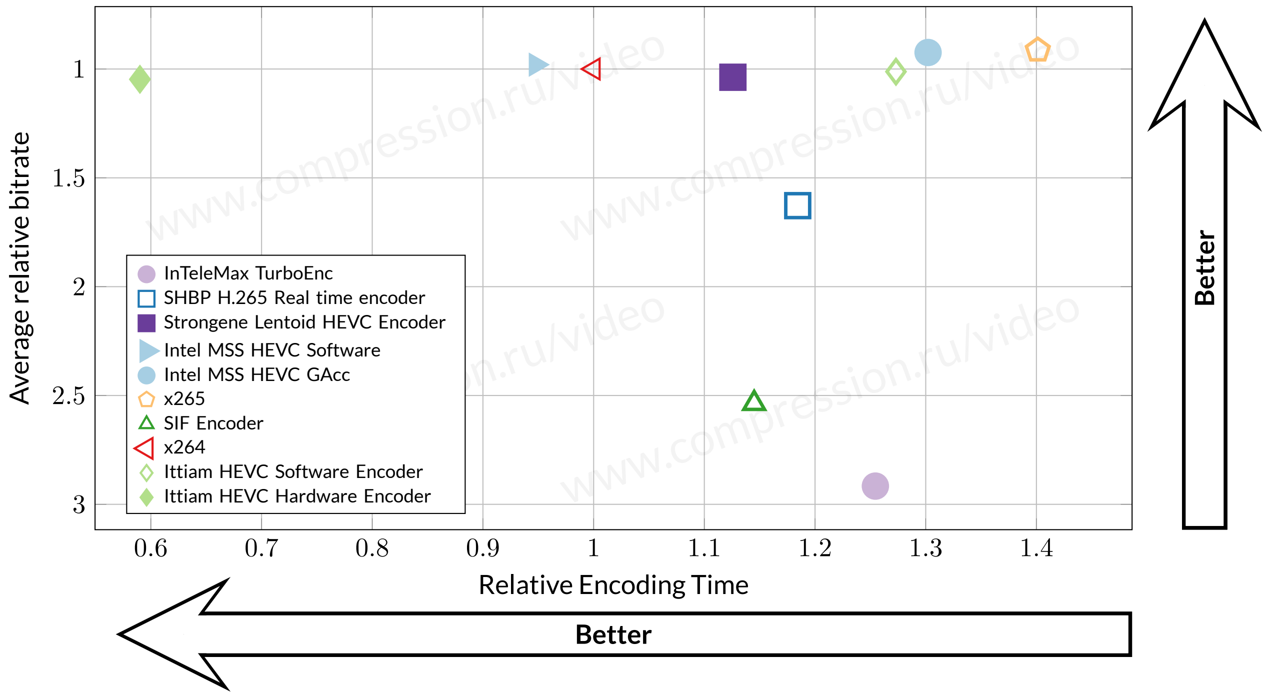


FIGURE 10: Speed/quality trade-off—usecase "Universal," all sequences, Y-SSIM metric

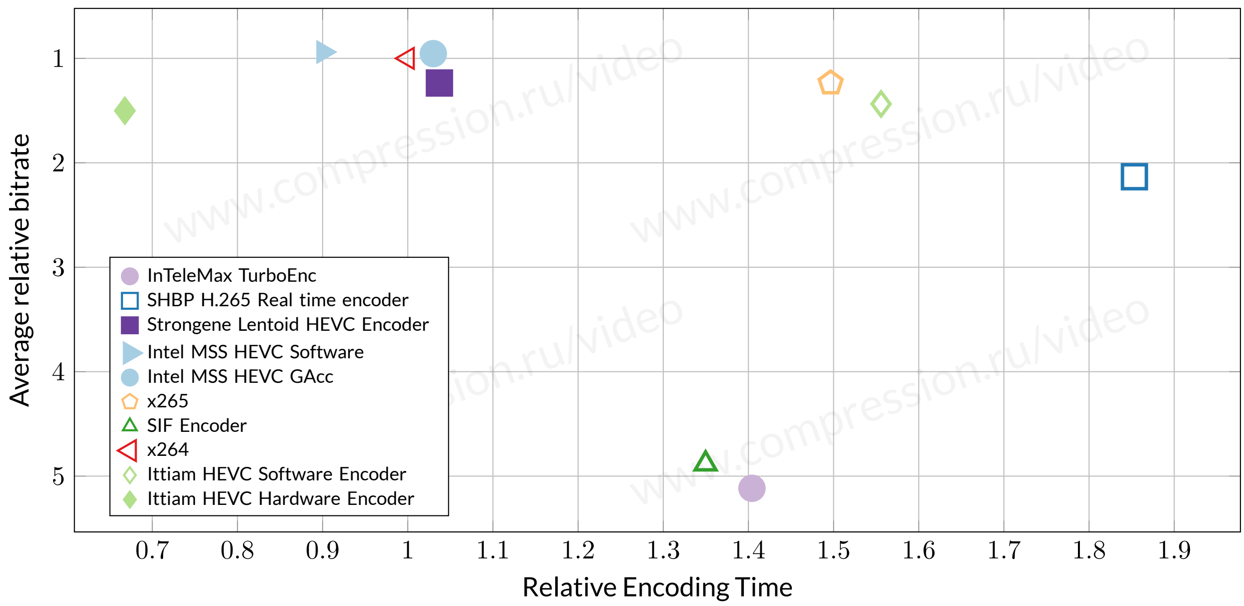


FIGURE 11: Speed/quality trade-off—usecase "Universal," Apple Tree sequence, Y-SSIM metric

5.2.4 Bitrate Handling

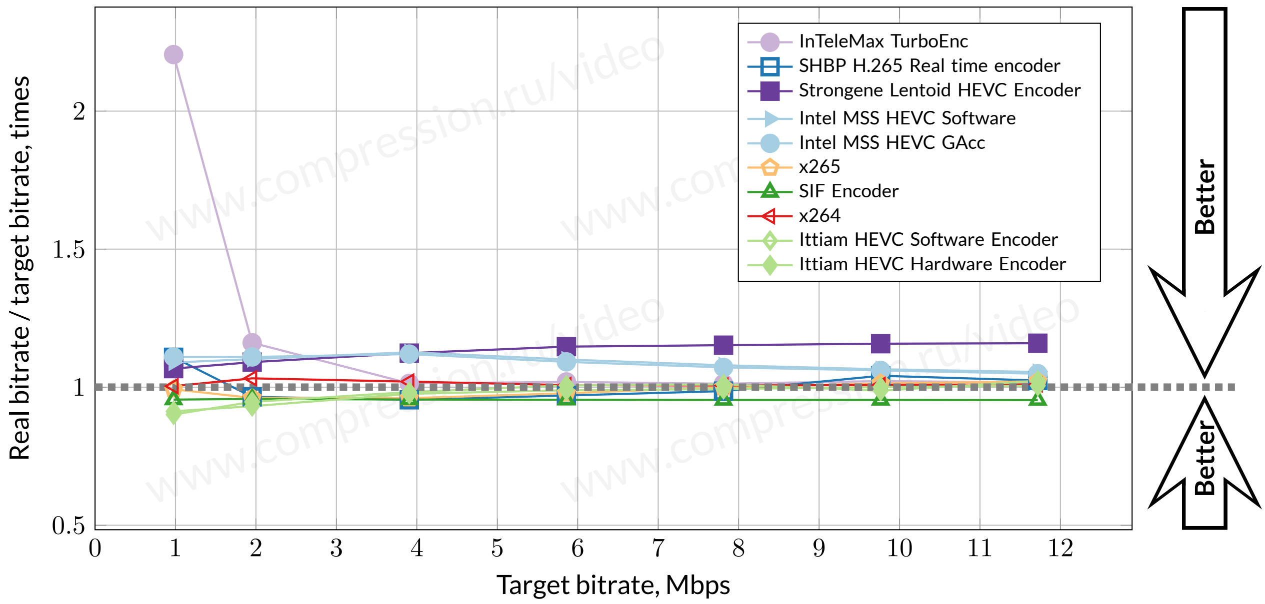


FIGURE 12: Bitrate handling—usecase "Universal," Apple Tree sequence

5.2.5 Relative Quality Analysis

	InTeleMax TurboEnc	SHBP H.265 Real time encoder	Strongene Lentoid HEVC Encoder	Intel MSS HEVC Software	Intel MSS HEVC GAcc	x265	SIF Encoder	x264	Ittiam HEVC Software Encoder	Ittiam HEVC Hardware Encoder
InTeleMax TurboEnc	100% ☺	55% ☺	35% ☺	33% ☺	31% ☺	30% ☺	80% ☺	34% ☺	34% ☺	35% ☺
SHBP H.265 Real time encoder	197% ☺	100% ☺	61% ☺	57% ☺	53% ☺	51% ☺	167% 😐	61% ☺	N/A ☹	60% ☺
Strongene Lentoid HEVC Encoder	334% 😐	215% ☺	100% ☺	95% ☺	88% ☺	88% ☺	279% 😐	96% ☺	98% ☺	102% ☺
Intel MSS HEVC Software	346% 😐	219% ☺	107% ☺	100% ☺	94% ☺	94% ☺	299% 😐	102% ☺	104% ☺	108% ☺
Intel MSS HEVC GAcc	374% 😐	243% ☺	115% ☺	108% ☺	100% ☺	100% ☺	326% 😐	108% ☺	111% ☺	116% ☺
x265	367% 😐	237% ☺	115% ☺	108% ☺	101% ☺	100% ☺	319% 😐	109% ☺	112% ☺	116% ☺
SIF Encoder	134% ☺	65% ☺	38% ☺	37% ☺	33% ☺	33% ☺	100% ☺	39% ☺	36% ☺	38% ☺
x264	330% ☺	203% ☺	107% ☺	101% ☺	93% ☺	94% ☺	276% 😐	100% ☺	104% ☺	108% ☺
Ittiam HEVC Software Encoder	344% 😐	N/A ☹	103% ☺	98% ☺	91% ☺	90% ☺	295% 😐	99% ☺	100% ☺	104% ☺
Ittiam HEVC Hardware Encoder	330% 😐	215% ☺	99% ☺	94% ☺	87% ☺	87% ☺	281% 😐	95% ☺	96% ☺	100% ☺

Confidence ☹ 😐 ☺
0% 50% 100%

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

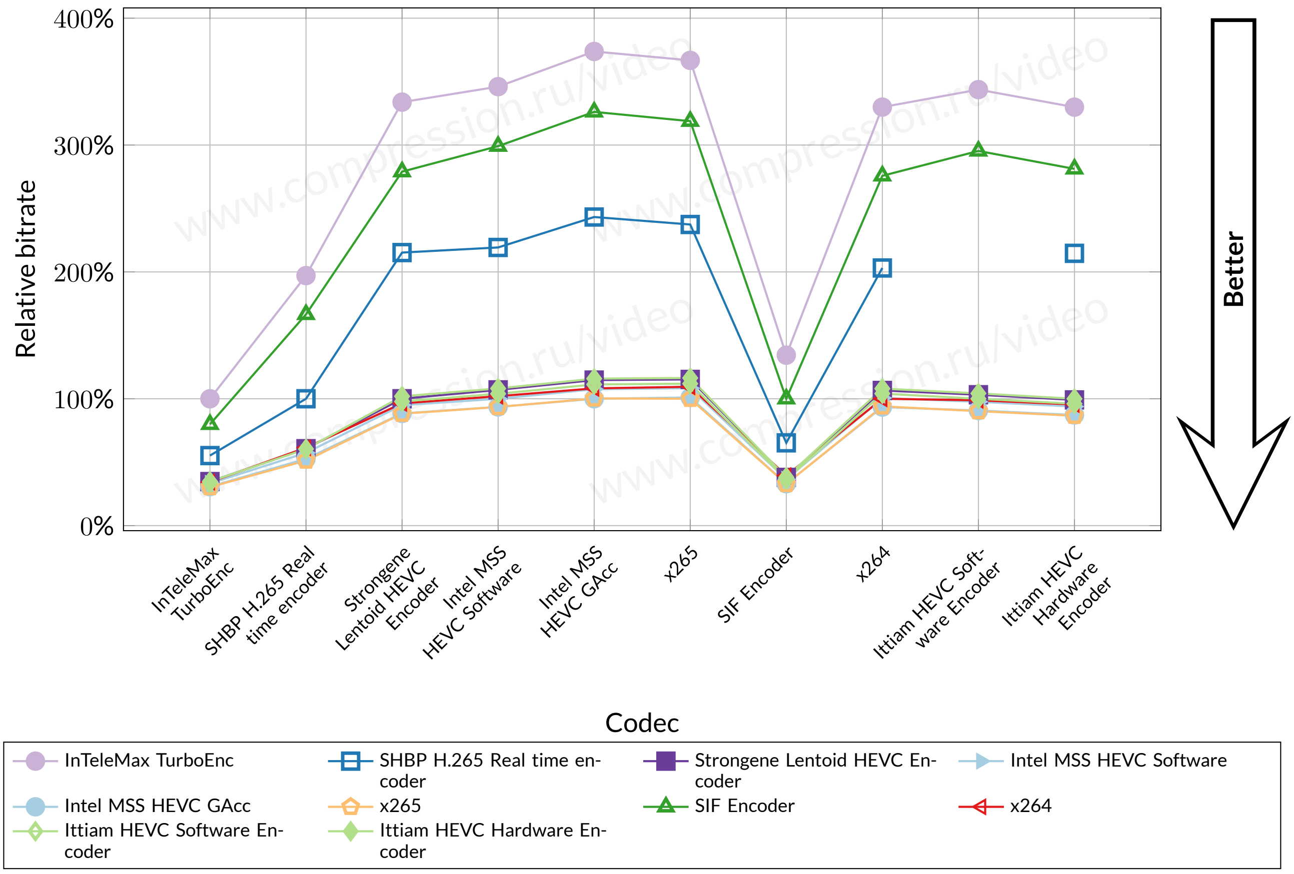


FIGURE 13: Average bitrate ratio for a fixed quality—usecase "Universal," Y-SSIM metric

For visual purposes we show the same plot below without InTeleMax TurboEnc, SIF and SHBP encoders.

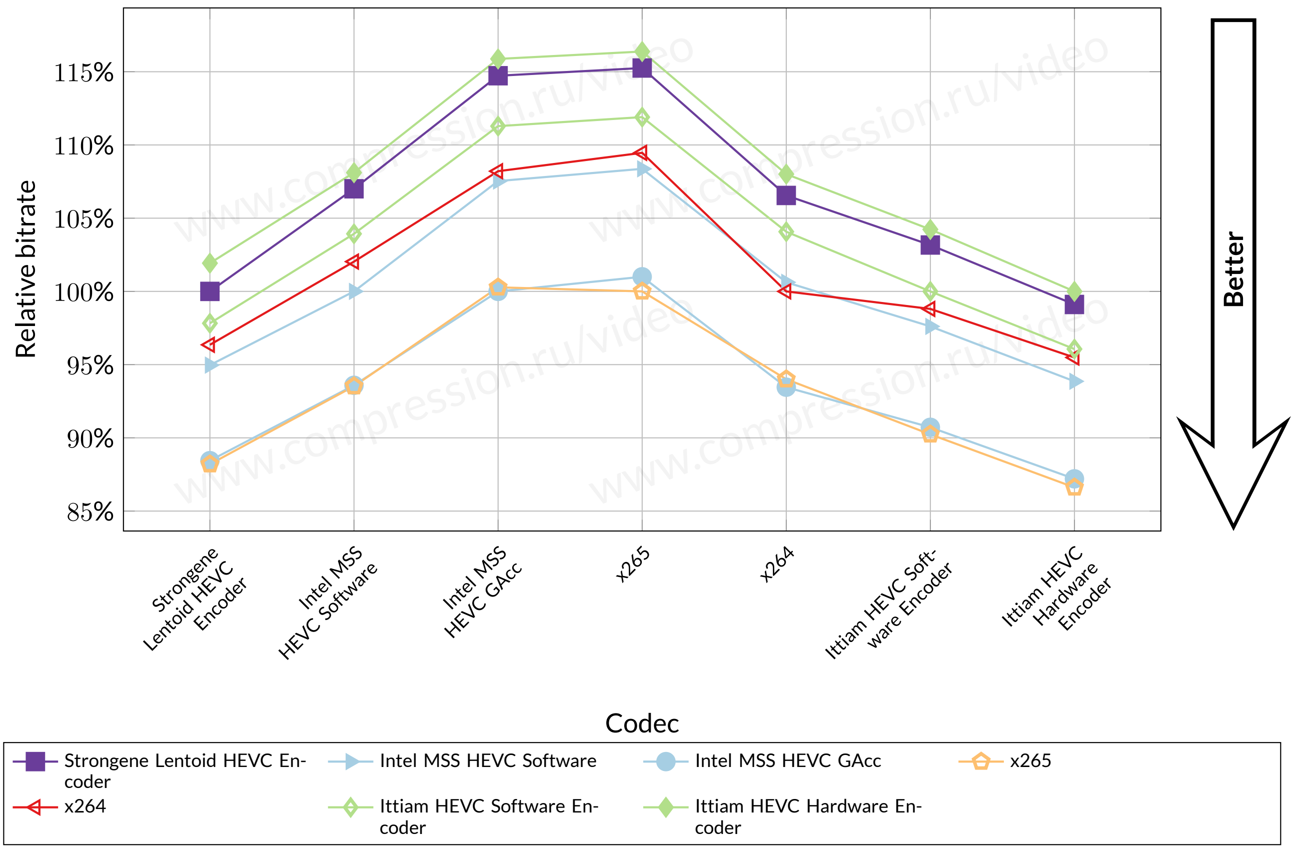


FIGURE 14: Average bitrate ratio for a fixed quality—usecase "Universal," Y-SSIM metric, without SIF Encoder, SHBP H.265 Real time encoder, InTeleMax TurboEnc

5.3 Ripping

This year we have intensive competition in Desktop-Ripping nomination of our comparison, since there are 10 participants.

5.3.1 RD curves

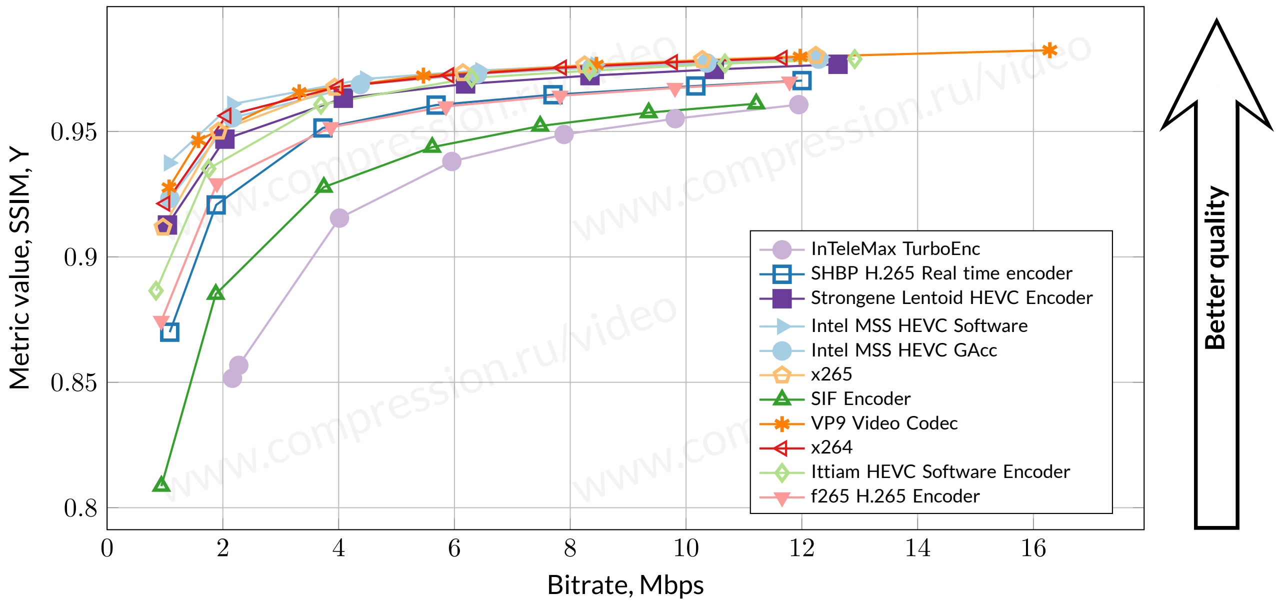


FIGURE 15: Bitrate/quality—usecase "Ripping," Apple Tree sequence, Y-SSIM metric

5.3.2 Encoding Speed

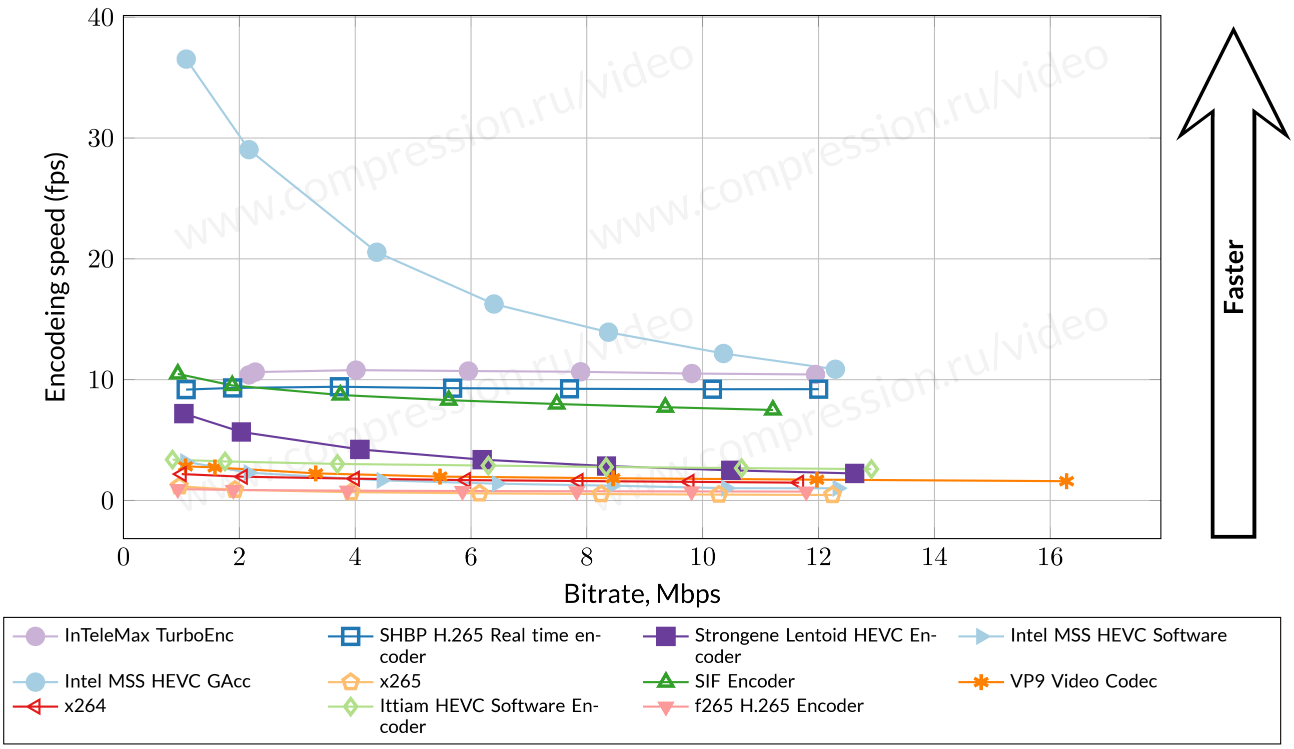


FIGURE 16: Encoding speed—usecase "Ripping," Apple Tree sequence

5.3.3 Speed/Quality Trade-Off

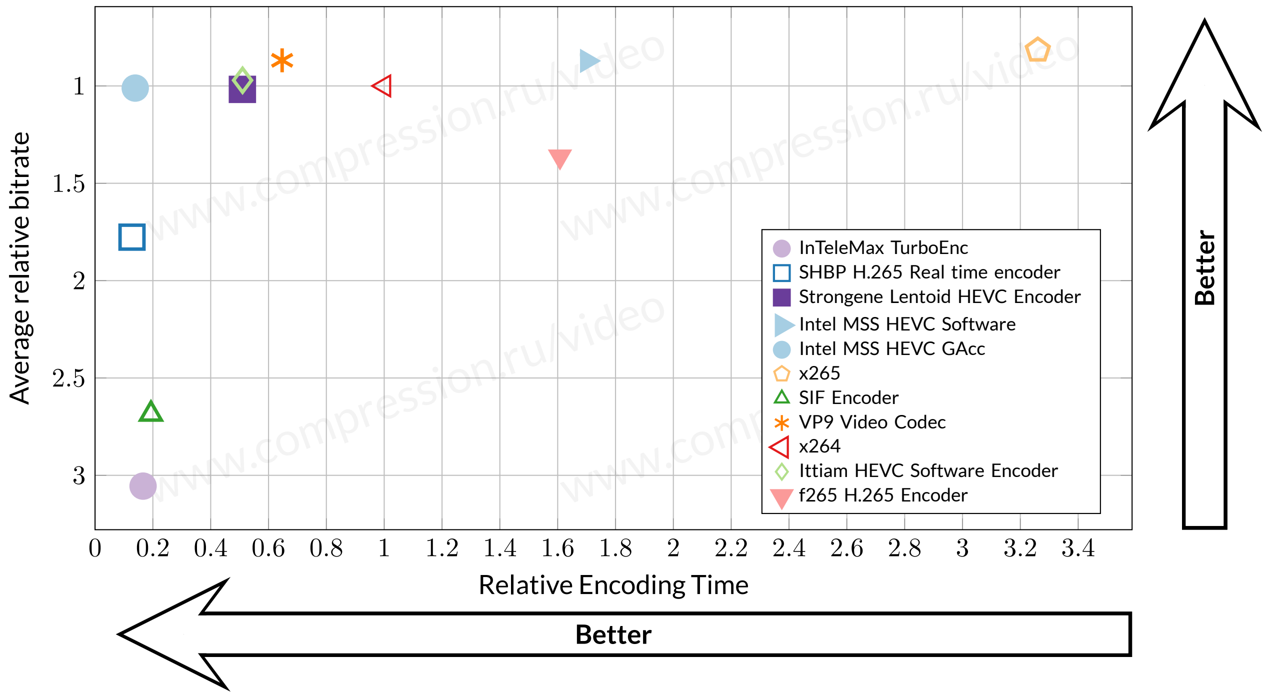


FIGURE 17: Speed/quality trade-off—usecase “Ripping,” all sequences, Y-SSIM metric

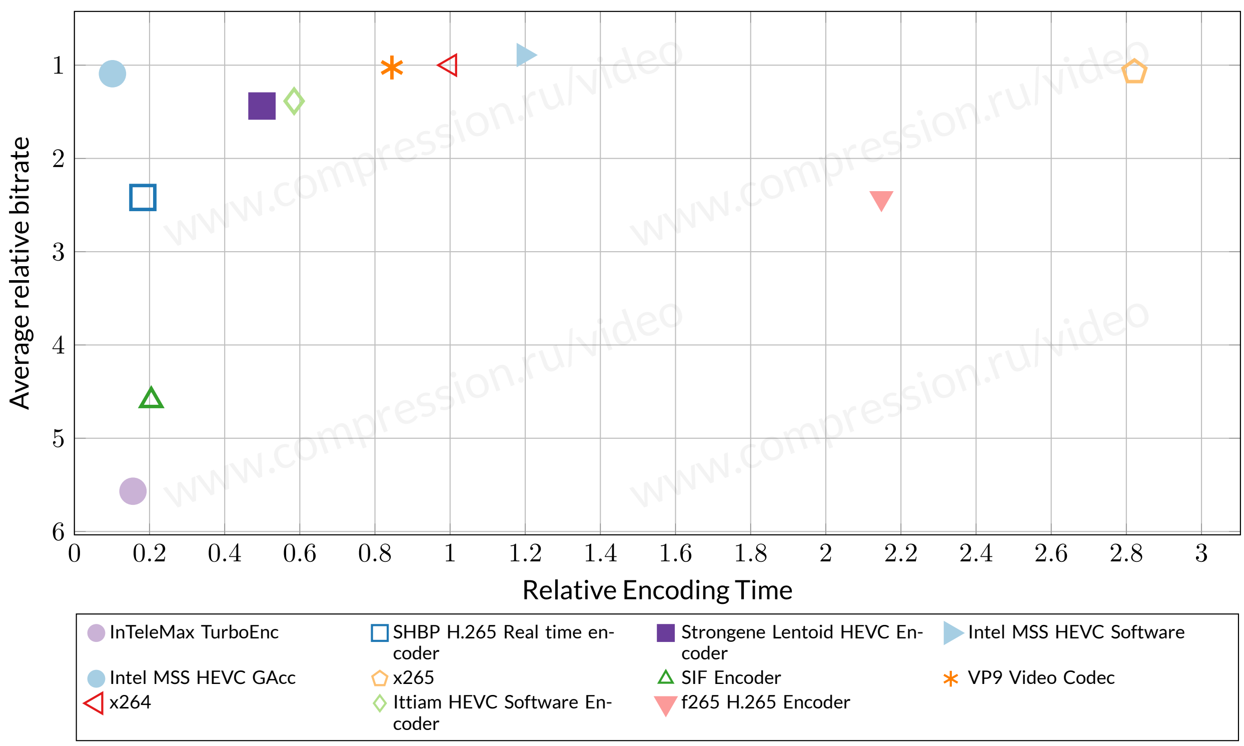


FIGURE 18: Speed/quality trade-off—usecase “Ripping,” Apple Tree sequence, Y-SSIM metric

5.3.4 Bitrate Handling

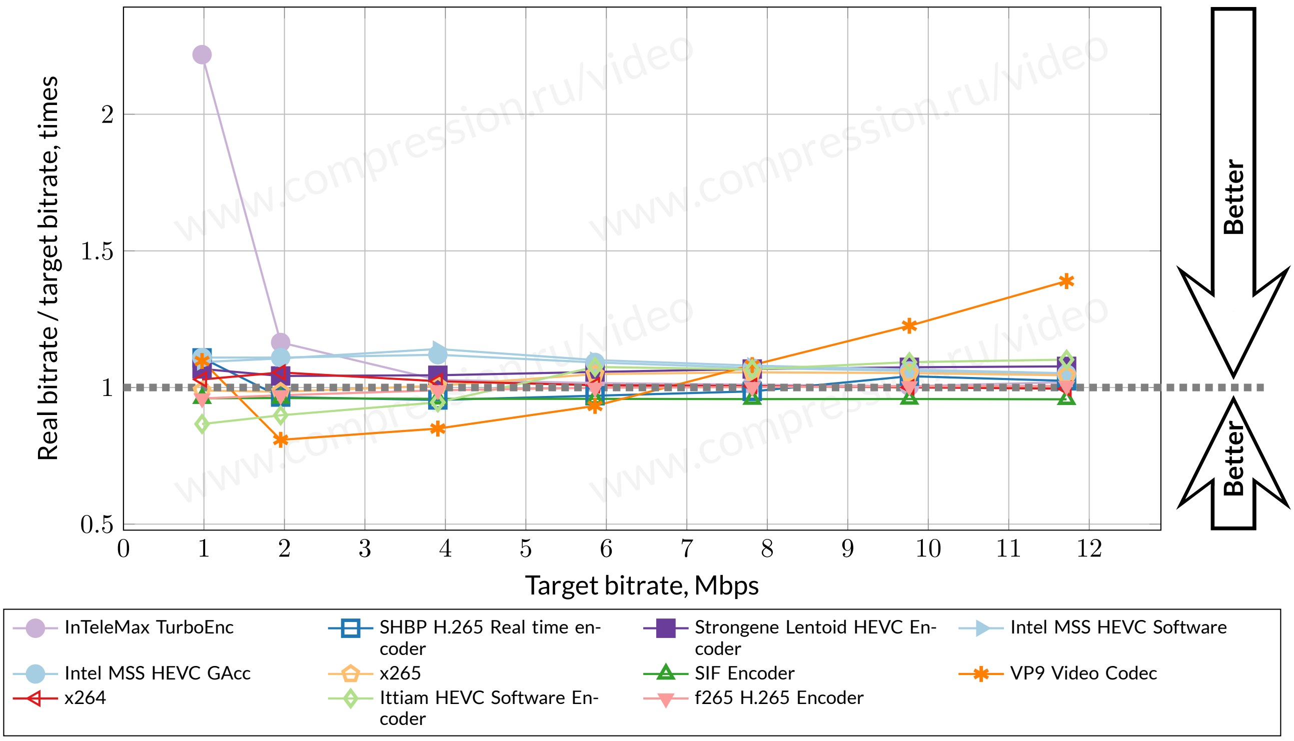


FIGURE 19: Bitrate handling—usecase "Ripping," Apple Tree sequence

5.3.5 Relative Quality Analysis

	InTeleMax TurboEnc	SHBP H.265 Real time encoder	Strongene Lentoid HEVC Encoder	Intel MSS HEVC Software	Intel MSS HEVC GAcc	x265	SIF Encoder	VP9 Video Codec	x264	Ittiam HEVC Software Encoder	f265 H.265 Encoder
InTeleMax TurboEnc	100%	57%	32%	27%	32%	26%	80%	N/A	33%	N/A	46%
SHBP H.265 Real time encoder	189%	100%	N/A	N/A	53%	N/A	161%	N/A	56%	N/A	83%
Strongene Lentoid HEVC Encoder	360%	N/A	100%	84%	100%	79%	310%	84%	98%	96%	146%
Intel MSS HEVC Software	435%	N/A	120%	100%	118%	93%	390%	98%	115%	114%	179%
Intel MSS HEVC GAcc	357%	243%	102%	85%	100%	80%	314%	84%	99%	96%	150%
x265	432%	N/A	128%	108%	126%	100%	382%	106%	122%	121%	187%
SIF Encoder	133%	68%	35%	28%	34%	27%	100%	29%	37%	32%	54%
VP9 Video Codec	N/A	N/A	122%	103%	120%	95%	388%	100%	115%	116%	179%
x264	351%	229%	105%	89%	103%	83%	295%	89%	100%	99%	153%
Ittiam HEVC Software Encoder	N/A	N/A	105%	89%	105%	83%	338%	87%	103%	100%	156%
f265 H.265 Encoder	255%	158%	71%	60%	72%	58%	203%	61%	73%	69%	100%

Confidence ☹ 0% — 😐 50% — ☺ 100%

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

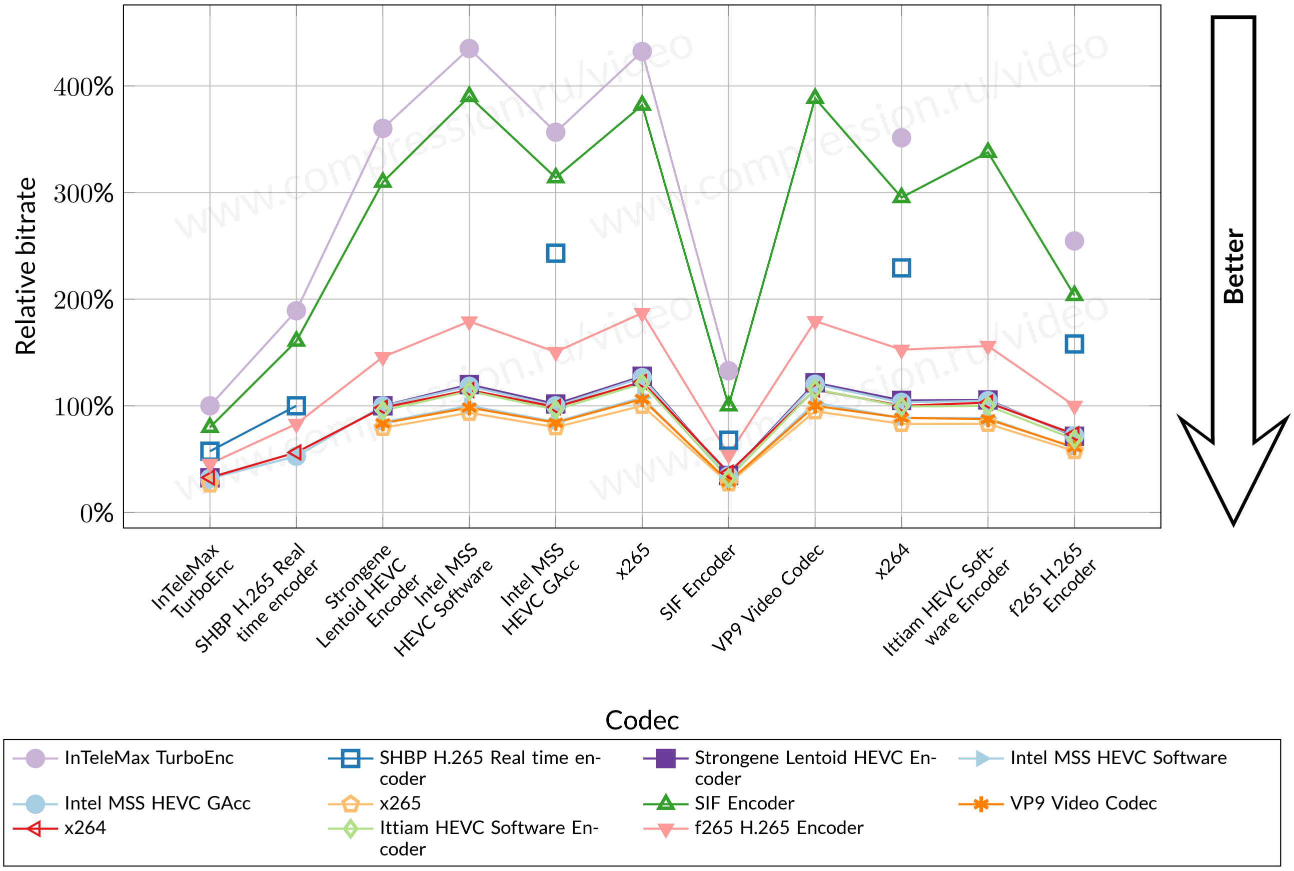


FIGURE 20: Average bitrate ratio for a fixed quality—usecase "Ripping," Y-SSIM metric

For visual purposes we show the same plot below without SIF, SHBP and TurboEnc encoders.

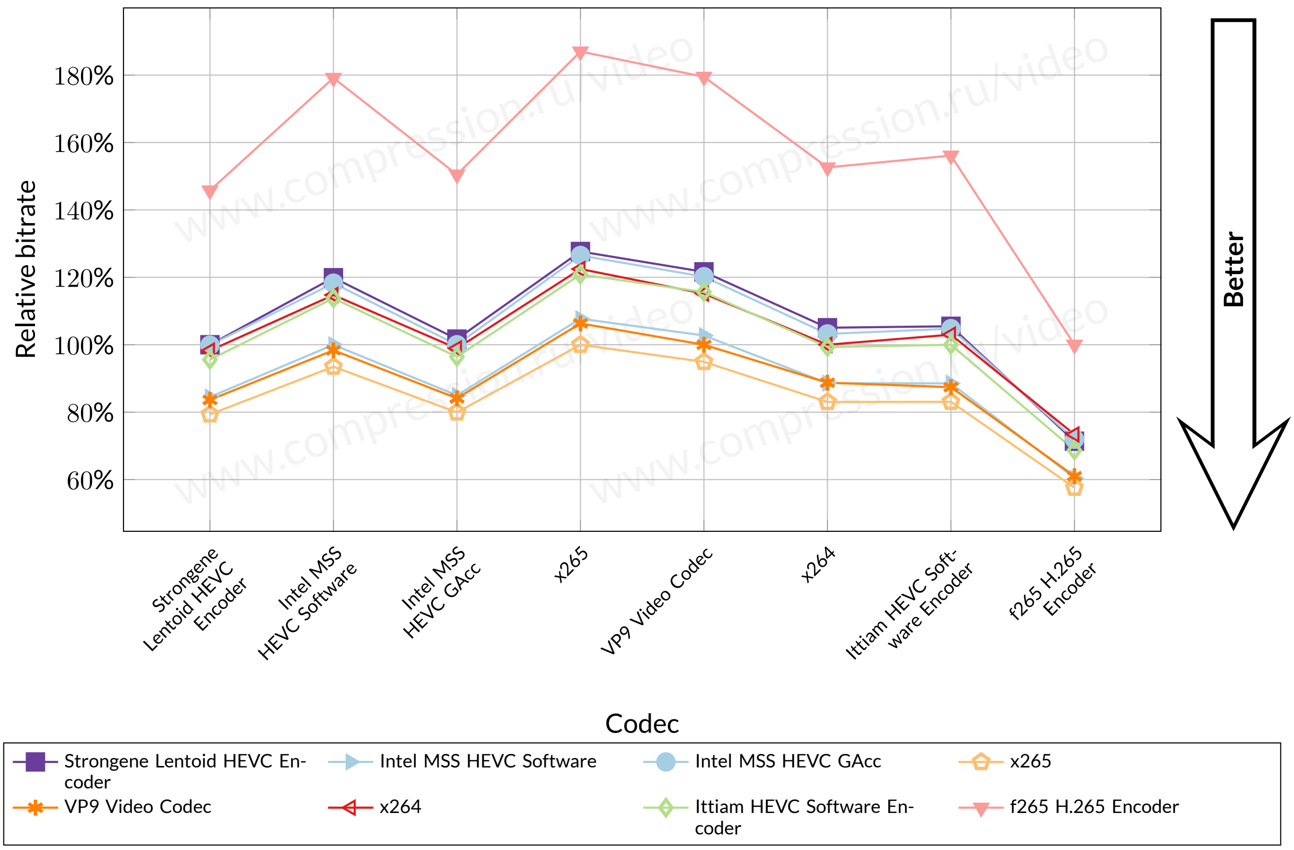


FIGURE 21: Average bitrate ratio for a fixed quality—usecase "Ripping," Y-SSIM metric, without SIF Encoder, SHBP H.265 Real time encoder, InTeleMax TurboEnc

6 CONCLUSION

6.1 Desktop Comparison

6.1.1 Fast Transcoding

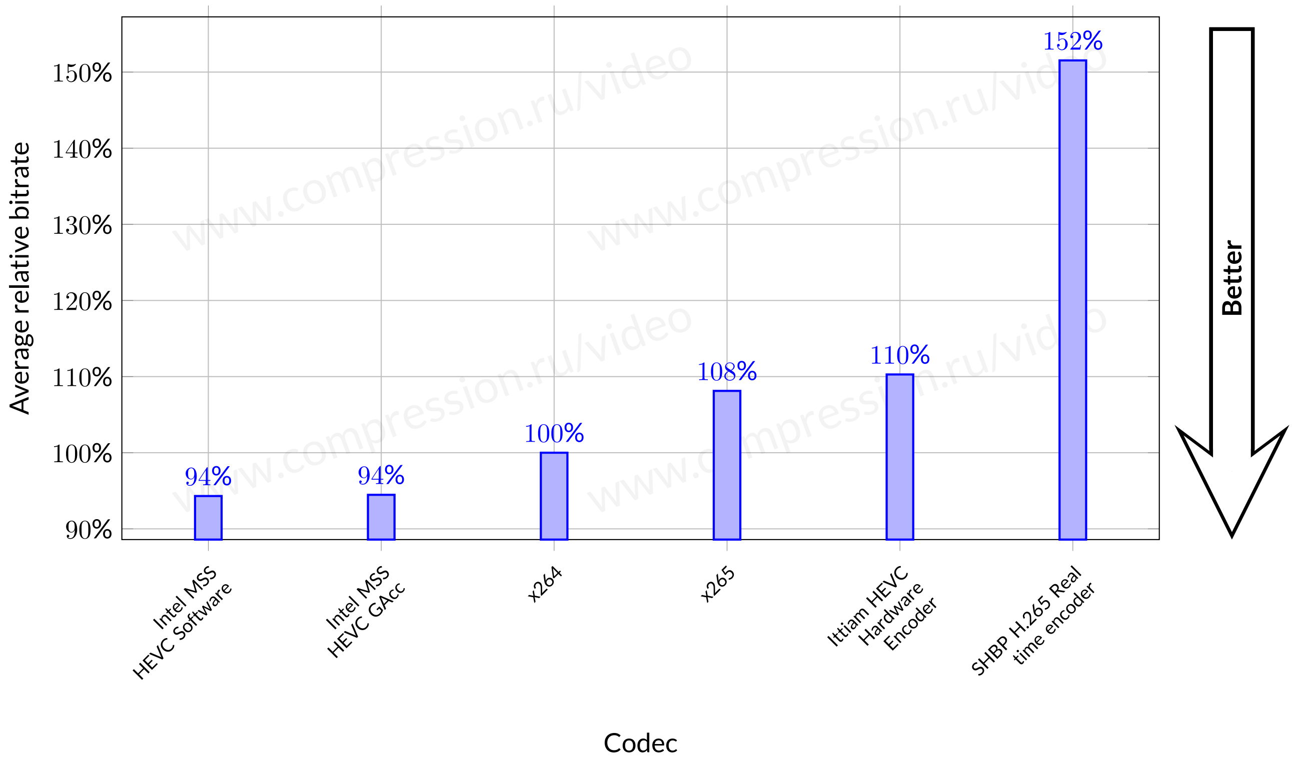


FIGURE 22: Average bitrate ratio for a fixed quality—usecase "Fast Transcoding," Y-SSIM metric.

6.1.2 Universal

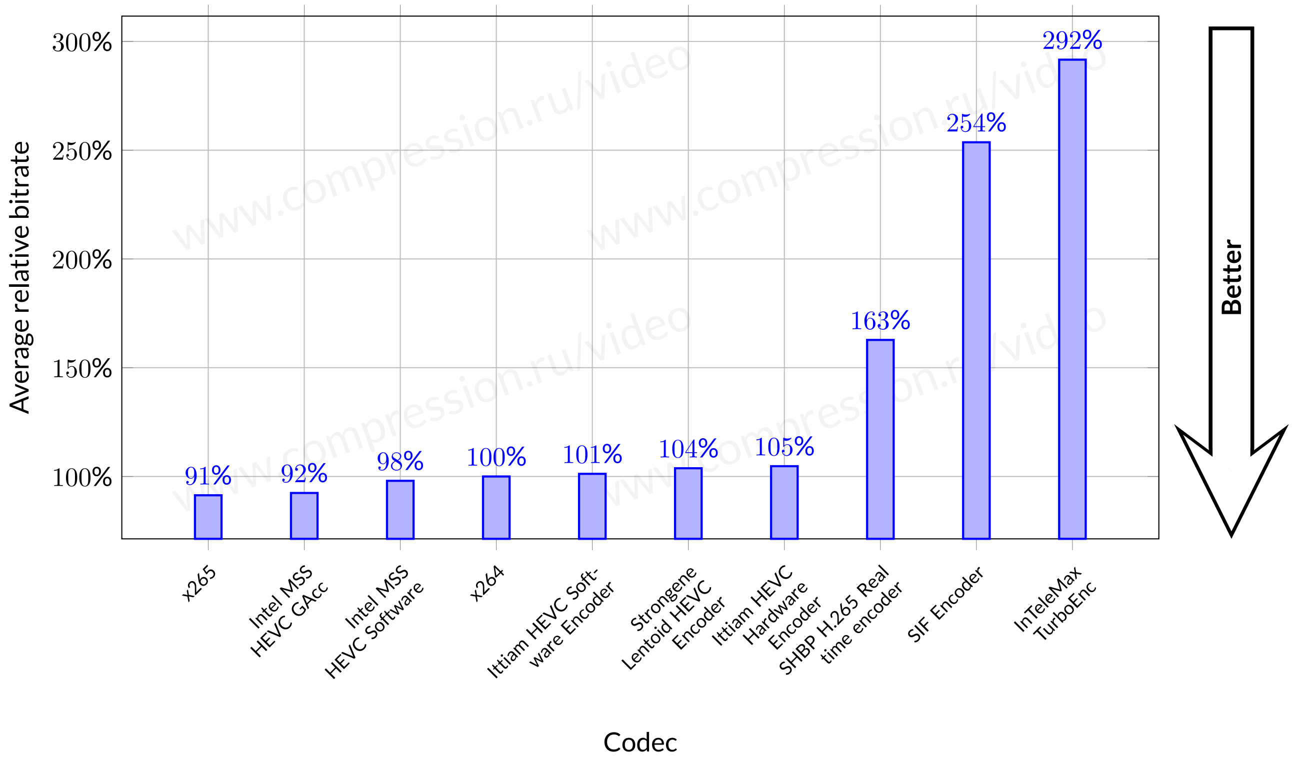


FIGURE 23: Average bitrate ratio for a fixed quality—usecase “Universal,” Y-SSIM metric.

6.1.3 Ripping

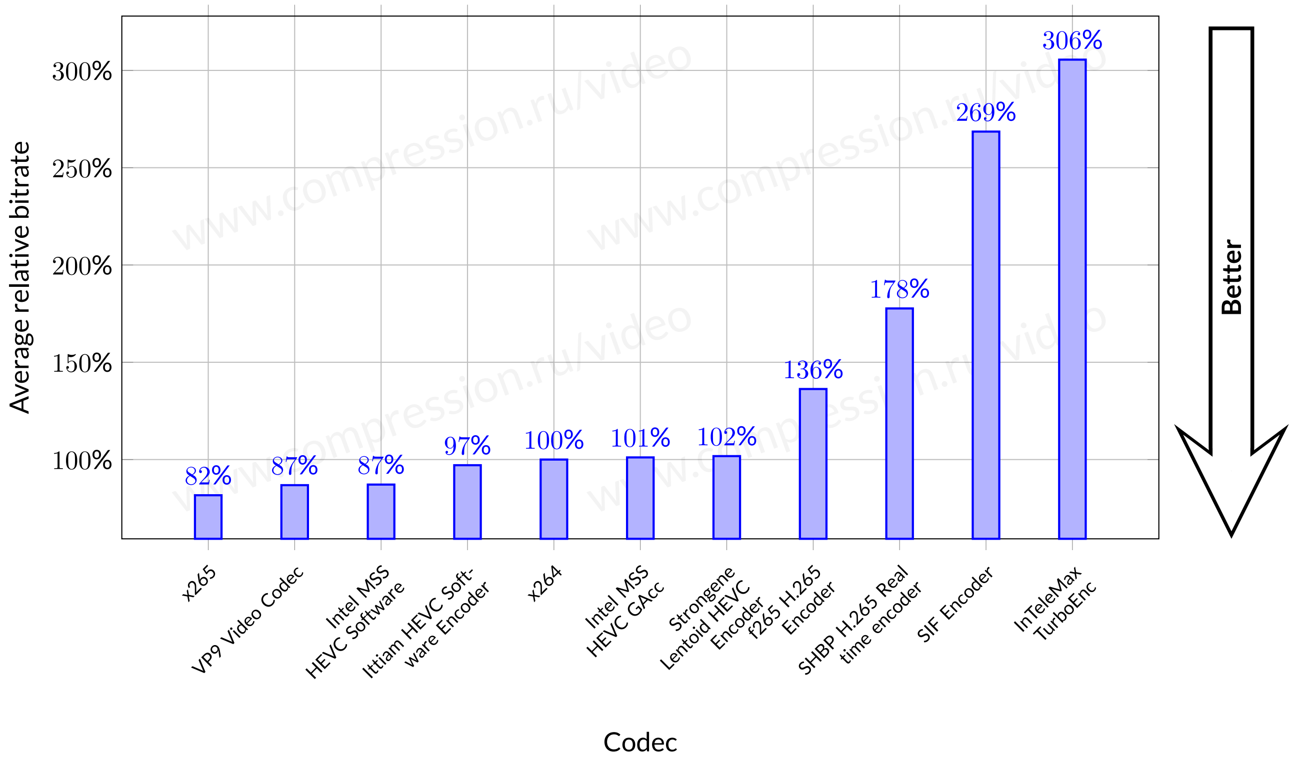


FIGURE 24: Average bitrate ratio for a fixed quality—usecase "Ripping," Y-SSIM metric.

6.1.4 Overall

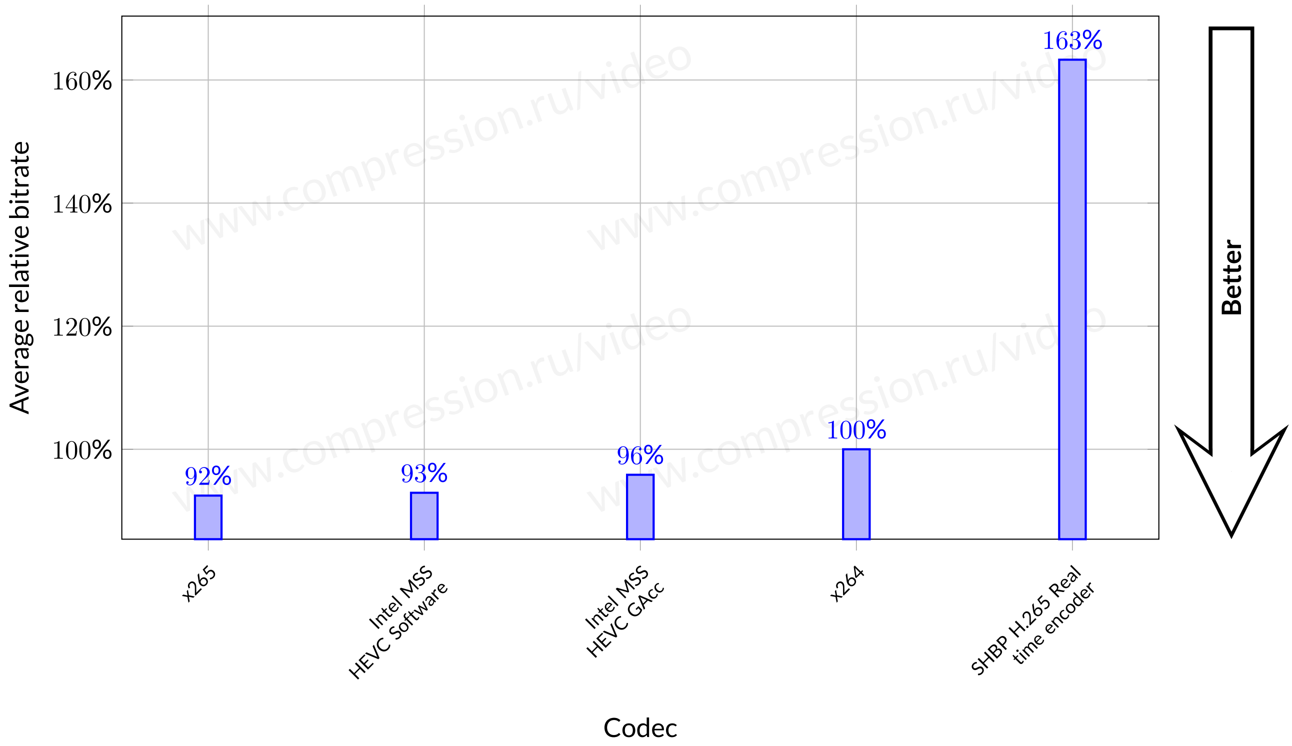


FIGURE 25: Average bitrate ratio for a fixed quality—Y-SSIM metric.

6.2 Server Comparison

6.2.1 Fast Transcoding

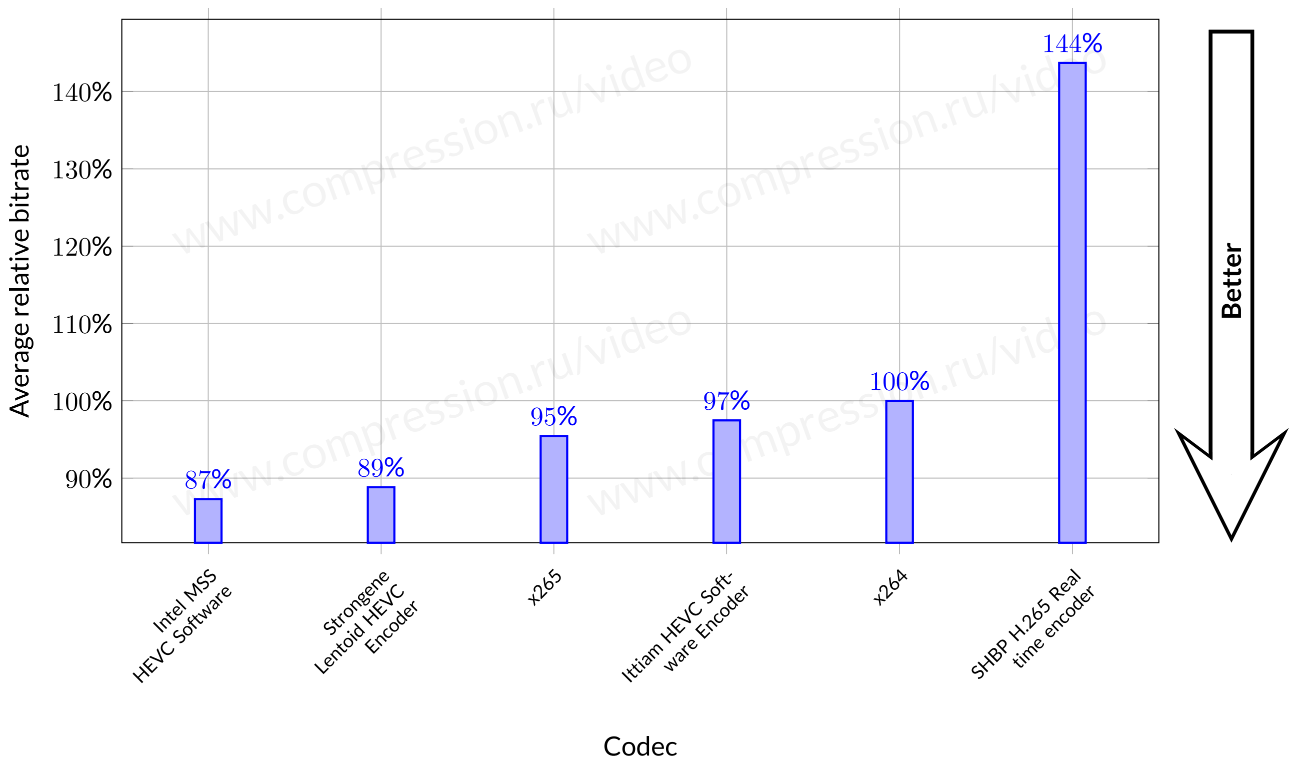


FIGURE 26: Average bitrate ratio for a fixed quality—usecase "Fast Transcoding," Y-SSIM metric.

6.2.2 Universal

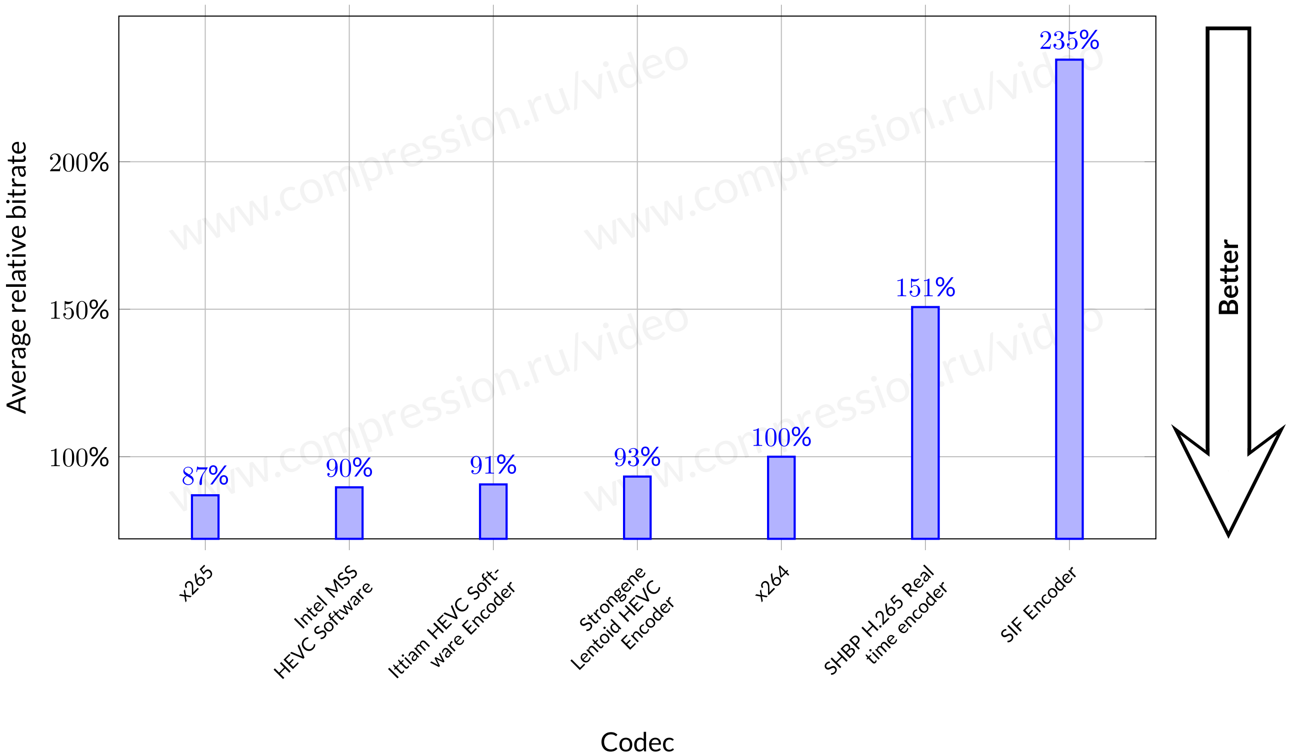


FIGURE 27: Average bitrate ratio for a fixed quality—usecase "Universal," Y-SSIM metric.

6.2.3 Ripping

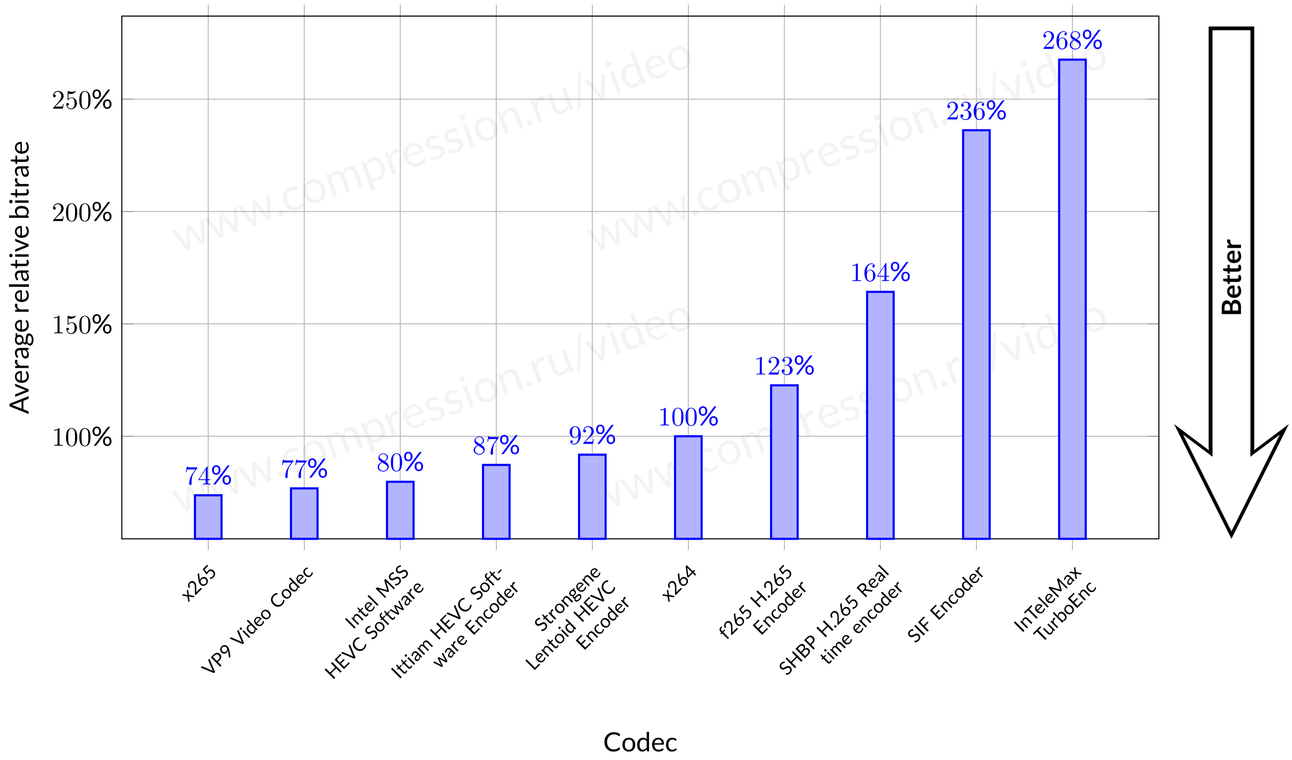


FIGURE 28: Average bitrate ratio for a fixed quality—usecase "Ripping," Y-SSIM metric.

6.2.4 Overall

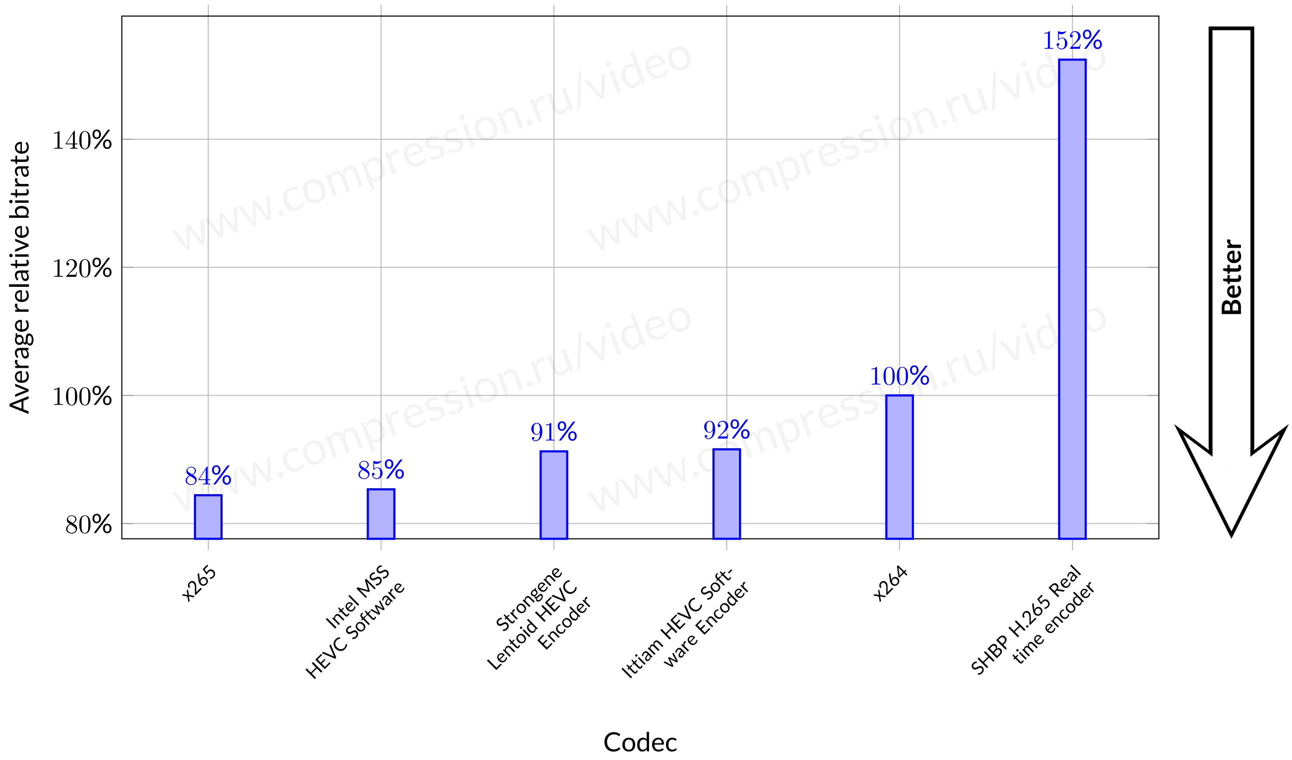


FIGURE 29: Average bitrate ratio for a fixed quality—Y-SSIM metric.

7 PARTICIPANTS' COMMENTS

7.1 WebM codec team—VP9 Encoder

1. We noted that the VP9 encoder in the test is running with parameter (`--cpu-used=1`). This setting trades off some compression performance for the encoding speed. One can use parameter (`--cpu-used=0`) to improve the compression performance by 8%-10%.

2. Although the test environment allows up to 32 cores, our current VP9 encoder is implemented to support up to 8 cores for encoding.

3. It is definitely possible to incorporate more computing resources to significantly speedup the VP9 encoding process while achieving the same compression statistics. As an open source project, we certainly welcome contributions from the video coding community to make it happen.

4. Regarding the rate control variance noted on page 25, the VP9 encode settings are optimized for VoD. This use case values a high compression ratio over precisely matching target bitrates, so the encoder allows the actual bitrate to be within $\pm$50% of the target. The acceptable range of variance could be reduced, with some loss in compression performance.

7.2 Ittiam Systems(P) Ltd.

Ittiam is happy that its encoder has been rated as pareto optimal in 5 out of 6 cases (across desktop and server), and, faster by 40+% than the reference x264 encoder in the fast transcoding and universal use-cases. We would like to point out that the configurations provided by us were not tuned for SSIM, but more for subjective viewing. Also, in cases other than ripping, our encoder was configured for single pass and CBR, while 2-pass VBR encoding (as done for x264) would have further improved the compression/BDRATE. The metric chosen Y-SSIM fails to take the chroma quality into account and we hope that MSU will factor this in their next round of comparisons. In our own comparisons against publicly available encoders, our chroma quality has been better than these encoders. For short sequences, the look-ahead configured tends to be comparable and reflects as a reduced throughput. Also, we feel that configuring the desktop in high performance mode could have further differentiated our encoder.

7.3 MulticoreWare, Inc.—x265 encoder

x265 is developed to optimize subjective visual quality, not objective quality metrics. As such, our objective quality measurements are not as high as they would be if we optimized for PSNR or SSIM.

Encoding speed has been greatly improved in the latest version of x265 (1.7 + 478), thanks to a combination of algorithmic improvements and other performance optimizations.

7.4 SIF Encoder Team

Thank you for invitation to participate in video codec test. We'd like to inform you that SIF codec version we presented is "work-in progress". Version you've tested was very close to release which was made about 3 years ago with very limited resources. SIF Codec core has a significant potential for improvement and we are going to realize it with our new team and additional resources in 2015 - 2016. We hope to present an improved version of SIF Codec for your next test.

A SEQUENCES

A.1 "Apple Tree"

Sequence title	Apple Tree
Resolution	1920×1080
Number of frames	338
Color space	YV12
Frames per second	30.0

FIGURE 30: Apple Tree sequence, frame 169

Video sequence with zoom out and lot of small details.

A.2 "Bunny"

Sequence title	Bunny
Resolution	1920×1080
Number of frames	600
Color space	YV12
Frames per second	24.0

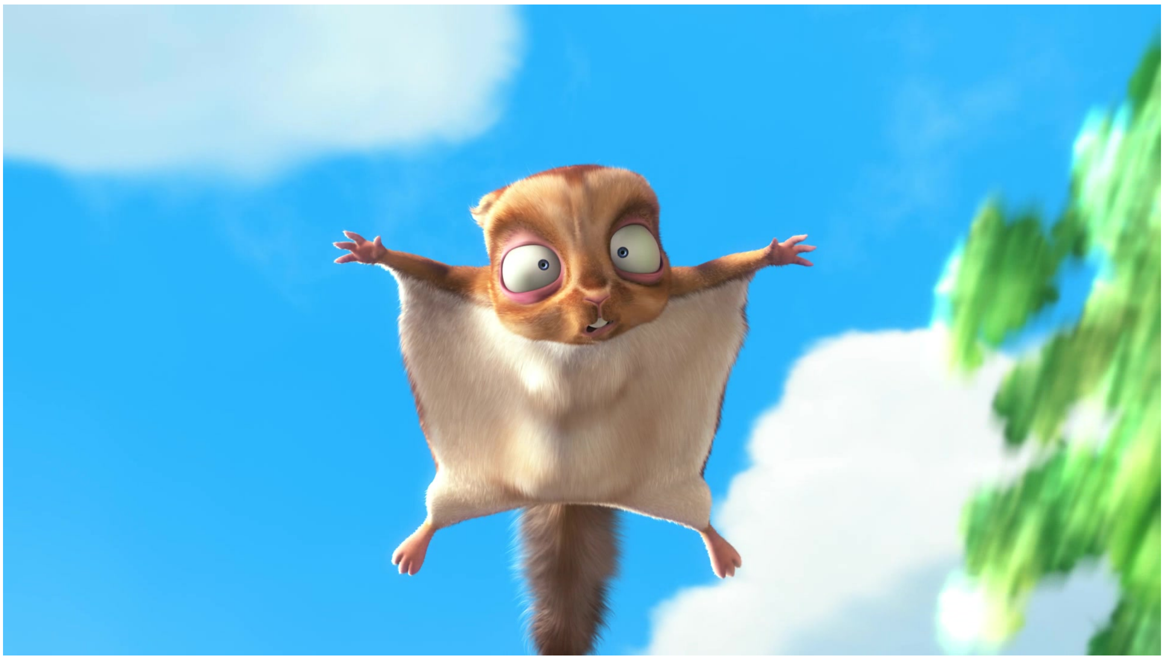


FIGURE 31: Bunny sequence, frame 300

Scene from a cartoon movie Big Buck Bunny. Contains a lot of movement, very bright colors, different type of motion.

A.3 "City Crowd"

Sequence title	City Crowd
Resolution	1920×1080
Number of frames	763
Color space	YV12
Frames per second	30.0

FIGURE 32: City Crowd sequence, frame 382

Video sequence with walking crowd.

A.4 "Concert"

Sequence title	Concert
Resolution	1920×1080
Number of frames	1533
Color space	YV12
Frames per second	25.0

FIGURE 33: Concert sequence, frame 766

This sequence is a part from concert record. This concert has not a lot of movement, but a lot of flashes and flashing color lights make this sequence pretty hard for the encoding. Camera zooms in the first part of the scene.

A.5 "Day Cars"

Sequence title	Day Cars
Resolution	1920×1080
Number of frames	1299
Color space	YV12
Frames per second	25.0

FIGURE 34: Day Cars sequence, frame 650

This sequence contains one of the Moscow large streets with big number of moving cars. Camera is static in this sequence.

A.6 “Developers”

Sequence title	Developers
Resolution	1280×720
Number of frames	1500
Color space	YV12
Frames per second	30.0

FIGURE 35: Developers sequence, frame 750

Video with some movement and facial expressions in foreground and some very bright movement at background (man in red shirt walking at background). Typical videoconference use-case.

A.7 "Fire"

Sequence title	Fire
Resolution	1920×1080
Number of frames	601
Color space	YV12
Frames per second	25.0

FIGURE 36: Fire sequence, frame 300

This scene contains campfire shoot at the evening. Big part of the frame is dark background.

A.8 "Golden Statue"

Sequence title	Golden Statue
Resolution	1920×1080
Number of frames	1993
Color space	YV12
Frames per second	30.0

FIGURE 37: Golden Statue sequence, frame 996

Video sequence with crowded street and moving cars on foreground.

A.9 "Hockey"

Sequence title	Hockey
Resolution	1920×1080
Number of frames	1000
Color space	YV12
Frames per second	25.0

FIGURE 38: Hockey sequence, frame 500

Record of Moscow State University (MSU) and Moscow State University of Civil Engineering hockey teams training. Recorded on the non-professional video camera, contains fast-moving players in the half-lighted ice rink.

A.10 "Kremlin"

Sequence title	Kremlin
Resolution	1920×1080
Number of frames	1899
Color space	YV12
Frames per second	25.0

FIGURE 39: Kremlin sequence, frame 950

Panoramic scene of Moscow Kremlin in the cloudy day. Slow camera movement.

A.11 "Market Walk"

Sequence title	Market Walk
Resolution	1920×1080
Number of frames	688
Color space	YV12
Frames per second	30.0

FIGURE 40: Market Walk sequence, frame 344

Video sequence with shaking camera.

A.12 "Mountain View"

Sequence title	Mountain View
Resolution	1920×1080
Number of frames	398
Color space	YV12
Frames per second	30.0

FIGURE 41: Mountain View sequence, frame 199

Video sequecne with view from fast moving car.

A.13 "Night Cars"

Sequence title	Night Cars
Resolution	1920×1080
Number of frames	1305
Color space	YV12
Frames per second	25.0

FIGURE 42: Night Cars sequence, frame 652

Same as the Day cars sequence, but this scene was shot at night � contains a lot of moving car�s lights and dark background.

A.14 "Pine Tree"

Sequence title	Pine Tree
Resolution	1920×1080
Number of frames	1130
Color space	YV12
Frames per second	30.0

FIGURE 43: Pine Tree sequence, frame 565

Video sequence with focus and defocus.

A.15 "River Boats"

Sequence title	River Boats
Resolution	1920×1080
Number of frames	1061
Color space	YV12
Frames per second	30.0

FIGURE 44: River Boats sequence, frame 530

Video sequence contains static and moving parts of the frame.

A.16 "Road"

Sequence title	Road
Resolution	1920×1080
Number of frames	877
Color space	YV12
Frames per second	25.0

FIGURE 45: Road sequence, frame 438

Sequence captured from the moving car on the Moscow streets. Contains small moving objects (pedestrians), trees and moving cars (cars speed differs from the camera speed).

A.17 "Shake Walk"

Sequence title	Shake Walk
Resolution	1920×1080
Number of frames	805
Color space	YV12
Frames per second	25.0

FIGURE 46: Shake Walk sequence, frame 402

Sequence recorded while walking with a strong camera movement.

A.18 "Sita"

Sequence title	Sita
Resolution	1920×1080
Number of frames	1000
Color space	YV12
Frames per second	25.0

FIGURE 47: Sita sequence, frame 500

Part of a cartoon movie Sita sings the blues. Contains a lot of contrast shapes with strict edges. Scenes contains only monotonous movement.

A.19 “Trigans”

Sequence title	Trigans
Resolution	1920×1080
Number of frames	10500
Color space	YV12
Frames per second	30.0

FIGURE 48: Trigans sequence, frame 5250

Long sequences with many scene changes.

A.20 “Water”

Sequence title	Water
Resolution	1920×1080
Number of frames	1209
Color space	YV12
Frames per second	25.0

FIGURE 49: Water sequence, frame 604

Scene contains pouring and highlighted water flow. Sometimes camera focuses on the bottom of the creek.

A.21 “Water Ripple”

Sequence title	Water Ripple
Resolution	1920×1080
Number of frames	220
Color space	YV12
Frames per second	30.0

FIGURE 50: Water Ripple sequence, frame 110

Water ripple, hard to compress sequence.

B CODECS

B.1 InTeleMax TurboEnc

Encoder title	InTeleMax TurboEnc
Version	3.0
Developed by	InTeleMax, Inc.

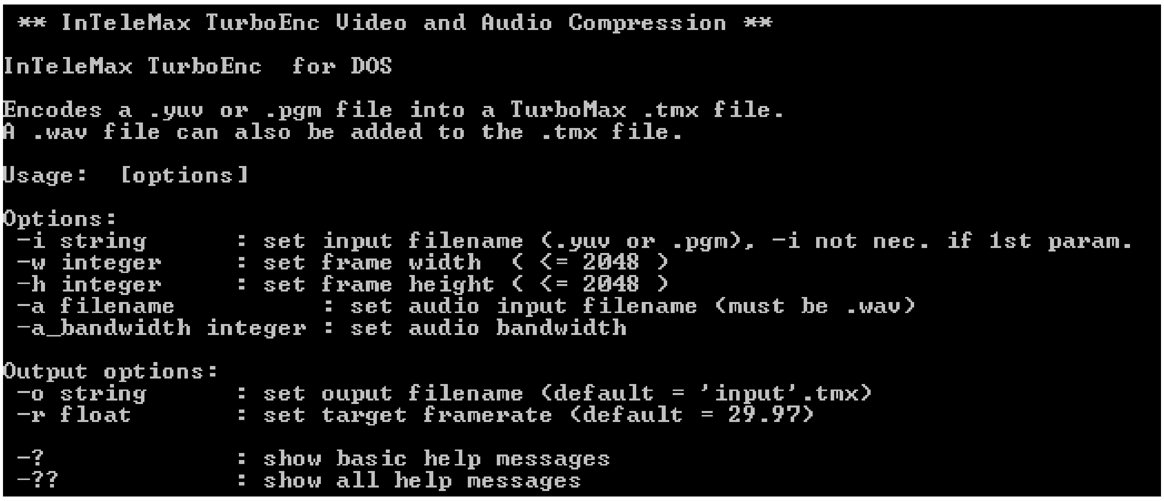


FIGURE 51: InTeleMax TurboEnc

Platform	Preset name	Encoder parameters
Desktop	Ripping	`Encode_V3.0.exe %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -o %TARGET_FILE% -bitrate %BITRATE_BPS% -r %FPS% -cf 4`
	Universal	`Encode_V3.0.exe %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -o %TARGET_FILE% -bitrate %BITRATE_BPS% -r %FPS% -cf 2`
Server	Ripping	`Encode_V3.0.exe %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -o %TARGET_FILE% -bitrate %BITRATE_BPS% -r %FPS% -cf 4`

B.2 f265

Encoder title	f265
Version	0.2
Developed by	f265 Developer Team

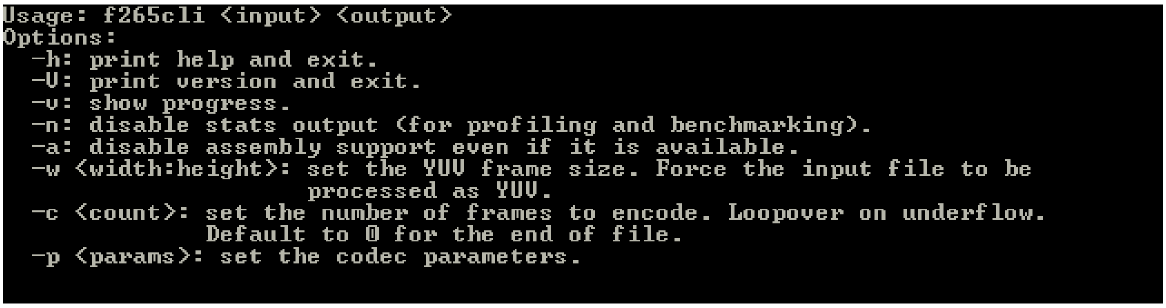


FIGURE 52: f265 encoder

Platform	Preset name	Encoder parameters
Desktop, Server	Ripping	`f265cli.exe -p "quality=25 rc=abr bitrate=%BITRATE_KBPS1000%" -w %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`

B.3 Intel Media Server Studio HEVC Software

Encoder title	Intel Media Server Studio HEVC Software
Version	Intel Media Server Studio 2015 R4 – Professional Edition (release date: April 2015)
Developed by	Intel

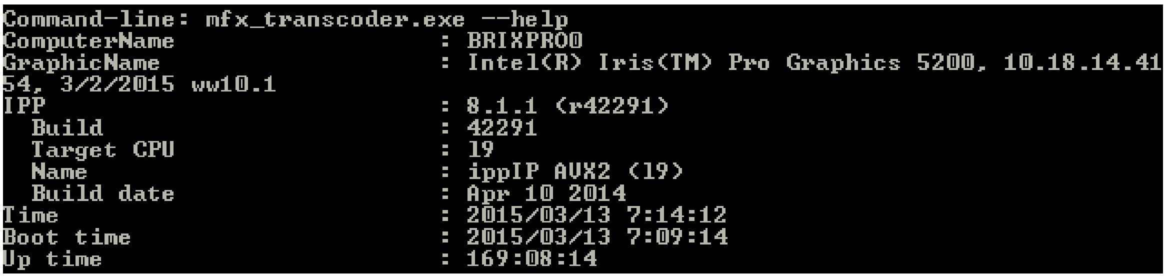


FIGURE 53: Intel Media Server Studio HEVC Software

Platform	Preset name	Encoder parameters
Desktop, Server	Ripping	`mfx_transcoder.exe h265 --preset slow -encode_plugin mfxplugin64_hevce_sw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 1 -async 3`
	Universal	`mfx_transcoder.exe h265 --preset medium -encode_plugin mfxplugin64_hevce_sw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 5 -async 3`
	Fast	`mfx_transcoder.exe h265 --preset fast -encode_plugin mfxplugin64_hevce_sw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 7 -async 3`

B.4 Intel Media Server Studio HEVC GAcc

Note: GAcc – Graphics Accelerated (GPGPU codec).

Encoder title	Intel Media Server Studio HEVC Graphics Accelerated
Version	Intel Media Server Studio 2015 R4 – Professional Edition (release date: April 2015)
Developed by	Intel

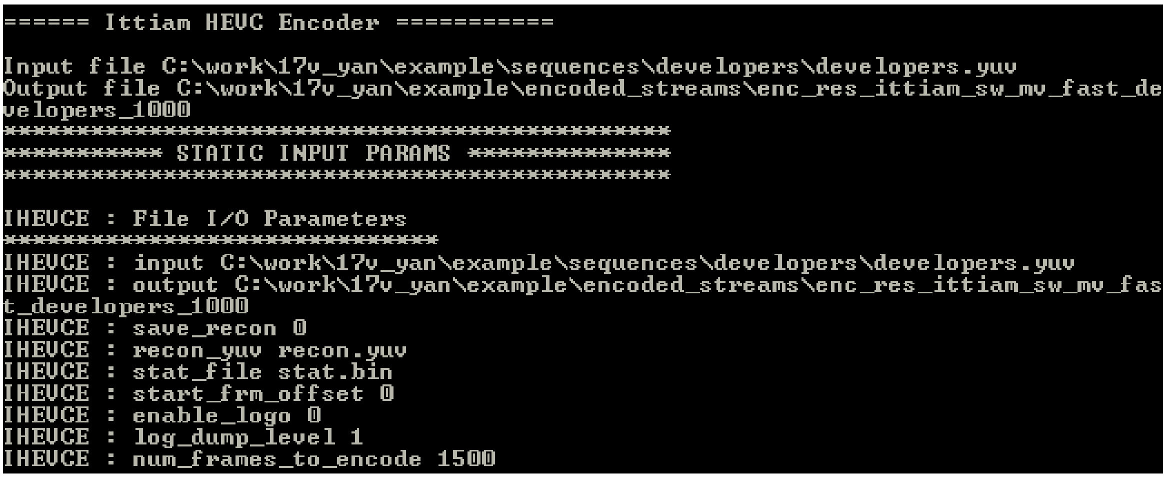
```
Command-line: mfx_transcoder.exe --help
ComputerName                                : BRIXPRO0
GraphicName                                 : Intel(R) Iris(TM) Pro Graphics 5200, 10.18.14.41
54, 3/2/2015 ww10.1
IPP                                         : 8.1.1 (r42291)
  Build                                     : 42291
  Target CPU                                : 19
  Name                                      : ippIP AVX2 (19)
  Build date                                : Apr 10 2014
Time                                        : 2015/03/13 7:14:12
Boot time                                   : 2015/03/13 7:09:14
Up time                                     : 169:08:14
```

FIGURE 54: Intel Media Server Studio HEVC GAcc

Platform	Preset name	Encoder parameters
Desktop	Ripping	`mfx_transcoder.exe h265 --preset slow -encode_plugin mfxplugin64_hevce_hw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 4 -async 3`
	Universal	`mfx_transcoder.exe h265 --preset medium -encode_plugin mfxplugin64_hevce_hw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 4 -async 3`
	Fast	`mfx_transcoder.exe h265 --preset fast -encode_plugin mfxplugin64_hevce_hw.dll -mfxdll libmfxrt64.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 7 -async 3`

B.5 Ittiam HEVC Software Encoder

Encoder title	Ittiam HEVC Software Encoder
Version	1_14_8_06
Developed by	Ittiam Systems (P) Ltd.



```
====== Ittiam HEVC Encoder ===========

Input file C:\work\17v_yan\example\sequences\developers\developers.yuv
Output file C:\work\17v_yan\example\encoded_streams\enc_res_ittiam_sw_mv_fast_de
velopers_1000
*********************************************
*********** STATIC INPUT PARAMS **************
*********************************************

IHEVCE : File I/O Parameters
******************************
IHEVCE : input C:\work\17v_yan\example\sequences\developers\developers.yuv
IHEVCE : output C:\work\17v_yan\example\encoded_streams\enc_res_ittiam_sw_mv_fas
t_developers_1000
IHEVCE : save_recon 0
IHEVCE : recon_yuv recon.yuv
IHEVCE : stat_file stat.bin
IHEVCE : start_frm_offset 0
IHEVCE : enable_logo 0
IHEVCE : log_dump_level 1
IHEVCE : num_frames_to_encode 1500
```

FIGURE 55: Ittiam HEVC Software Encoder

Platform	Preset name	Encoder parameters
Desktop, Server	Ripping	`ihevce_x64_sw_eval.exe -c vid_enc_cfg_FirstPass_vbr_1080p_ES_ripping.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
		`ihevce_x64_sw_eval.exe -c vid_enc_cfg_SecondPass_vbr_1080p_PQ_ripping.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
	Universal	`ihevce_x64_sw_eval.exe -c vid_enc_cfg_Cbr_1080p_MS_universal.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
	Fast	`ihevce_x64_sw_eval.exe -c vid_enc_cfg_Cbr_1080p_ES_fast.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`

B.6 Ittiam HEVC Hardware Encoder

Encoder title	Ittiam HEVC Hardware Encoder
Version	2_04_4_00
Developed by	Ittiam Systems (P) Ltd.

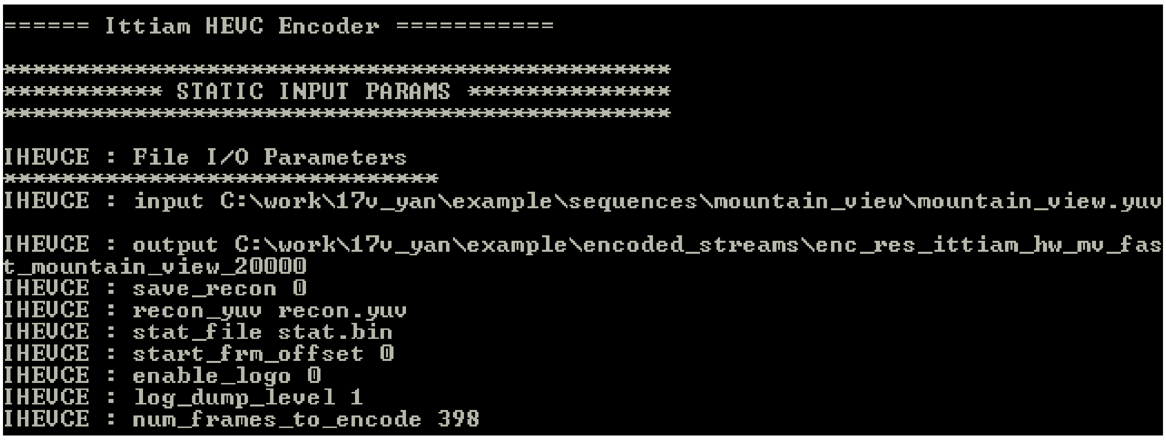


FIGURE 56: Ittiam HEVC Hardware Encoder

Platform	Preset name	Encoder parameters
Desktop	Universal	`ihevce_x64_hw_eval.exe -c vid_enc_cfg_Cbr_1080p_MS_hw_universal.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
	Fast	`ihevce_x64_hw_eval.exe -c vid_enc_cfg_Cbr_1080p_ES_hw_fast.txt --src_width %WIDTH% --src_height %HEIGHT% --tgt_width %WIDTH% --tgt_height %HEIGHT% --num_frames_to_encode %FRAMES_NUM% --src_frame_rate_num %FPS_NUM% --src_frame_rate_denom %FPS_DENOM% --tgt_bitrate %BITRATE_BPS% -i %SOURCE_FILE% -o %TARGET_FILE%`

B.7 Strongene Lentoid HEVC Encoder

Encoder title	Strongene Lentoid HEVC Encoder
Version	2.2 (r3833 for client, r3771 for server)
Developed by	Strongene Ltd.

```
lentenc: strongene lentoid HEVC encoder app.
Ver: 2.2 (r3771)
usage: lentenc [options] <input_filename> [<output_filename>]
options:
  -h|--help                display this help page
     --wait                wait debuger
  -s|--size <W>x<H>        video resolution, the width and height must be a
                           multiple of 8, default is 1920x1080
  -n|--frames <Int>        stop after a specified number of frames encoded
     --compat <Ver>        version of HM software for compatibility, can be 9.1
                           or 10 or 12, default is 12
  -c|--complex <Int>       Setting encoding complexity [0~4]
     --tune <Enum1>[,<Enum2>[,<Enum3>]]
                           Tune for a particular type of source or situation
                           There are three types of tuning, they are :
                               Application situation
                               Metric
                               Expand
                           In the same types only one tuning can be used at a time
```

FIGURE 57: Strongene Lentoid HEVC Encoder

Platform	Preset name	Encoder parameters
Desktop	Ripping	`lentenc -c 4 --frames %FRAMES_NUM% -y -b %BITRATE_KBPS% -r %FPS% -s %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`
	Universal	`lentenc -c 2 --frames %FRAMES_NUM% -y -b %BITRATE_KBPS% -r %FPS% -s %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`
Server	Ripping	`lentenc -c 4 --frames %FRAMES_NUM% -y -b %BITRATE_KBPS% -r %FPS% -s %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`
	Universal	`lentenc -c 2 --frames %FRAMES_NUM% -y -b %BITRATE_KBPS% -r %FPS% -s %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`
	Fast	`lentenc -c 1 --frames %FRAMES_NUM% -y -b %BITRATE_KBPS% -r %FPS% -s %WIDTH%x%HEIGHT% %SOURCE_FILE% %TARGET_FILE%`

B.8 SHBP H.265 Real time encoder

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

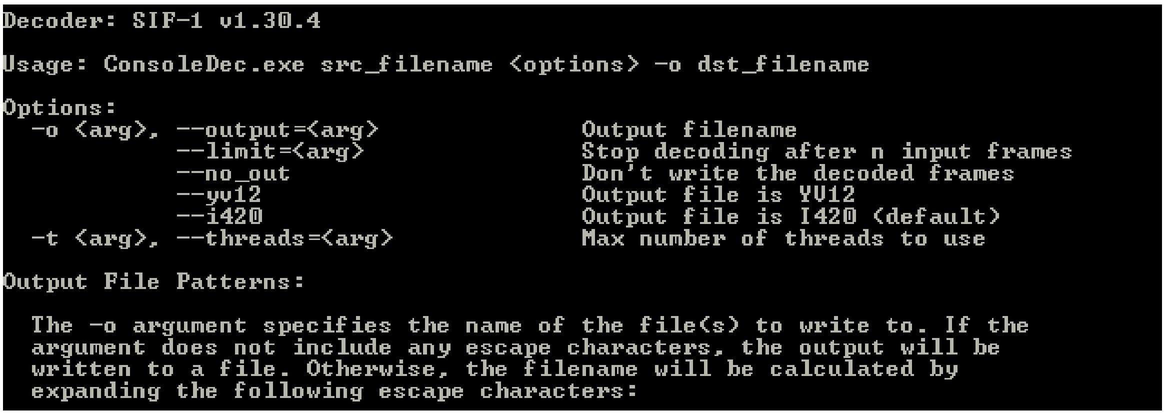
```
Usage: sh_hevc_enc.exe <options>

  -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 (24.0)
  -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 (120)
  -q <n>        n - quantization parameter for variable bitrate [1, 51] (disabled
-b option)
  -p <n>        n - performance level (0 - fastest)
Incorrect/missing input filename: .
```

FIGURE 58: SHBP H.265 Real time encoder

Platform	Preset name	Encoder parameters
Desktop, Server	Ripping	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 3 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
	Universal	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 2 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`
	Fast	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 1 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`

B.9 SIF-1

Encoder title	SIF-1
Version	1.30.4
Developed by	www.mysif.ru



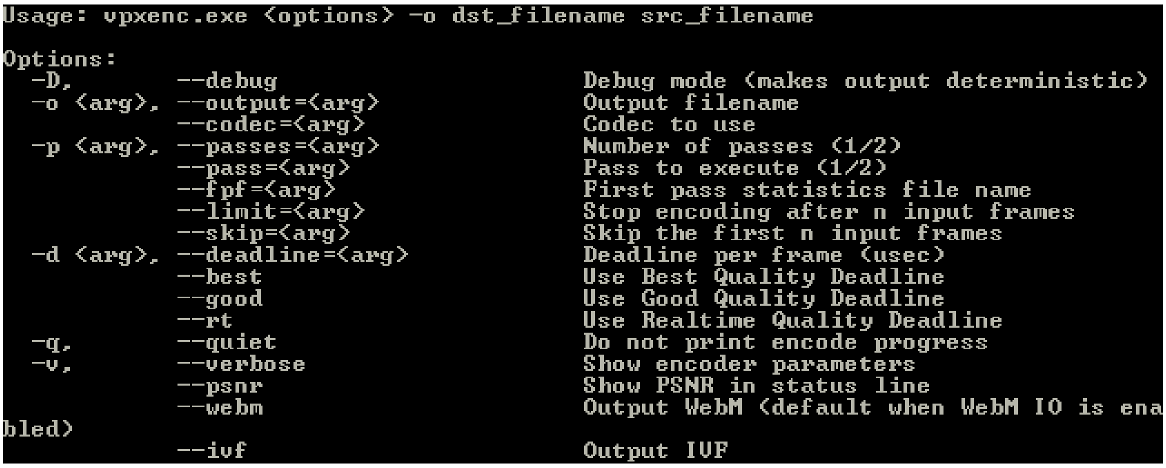
```
Decoder: SIF-1 v1.30.4

Usage: ConsoleDec.exe src_filename <options> -o dst_filename

Options:
  -o <arg>, --output=<arg>            Output filename
            --limit=<arg>             Stop decoding after n input frames
            --no_out                  Don't write the decoded frames
            --yv12                    Output file is YV12
            --i420                    Output file is I420 (default)
  -t <arg>, --threads=<arg>           Max number of threads to use

Output File Patterns:

  The -o argument specifies the name of the file(s) to write to. If the
  argument does not include any escape characters, the output will be
  written to a file. Otherwise, the filename will be calculated by
  expanding the following escape characters:
```

FIGURE 59: SIF-1

Platform	Preset name	Encoder parameters
Desktop	Ripping	`ConsoleEnc.exe %SOURCE_FILE% --me_mode=slow --comp_mode=vbr_all_p --out_bitrt=%BITRATE_KBPS% --viz_dist=100 --sub_me_mode=fastest --rc_buf_s=2 --entropy_mode=4_threads -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% -o %TARGET_FILE%.avi`
	Universal	`ConsoleEnc.exe %SOURCE_FILE% --me_mode=standard --comp_mode=vbr_all_p --out_bitrt=%BITRATE_KBPS% --viz_dist=150 --sub_me_mode=fastest --rc_buf_s=2 --no_chroma_me --entropy_mode=4_threads -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% -o %TARGET_FILE%.avi`
Server	Ripping	`ConsoleEnc.exe %SOURCE_FILE% --me_mode=slow --comp_mode=vbr_all_p --out_bitrt=%BITRATE_KBPS% --viz_dist=100 --sub_me_mode=fastest --rc_buf_s=2 --entropy_mode=8_threads --threads=28 -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% -o %TARGET_FILE%.avi`
	Universal	`ConsoleEnc.exe %SOURCE_FILE% --me_mode=ultrafast --comp_mode=vbr_all_p --out_bitrt=%BITRATE_KBPS% --viz_dist=350 --sub_me_mode=fastest --rc_buf_s=1 --no_chroma_me --entropy_mode=8_threads --threads=28 -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% -o %TARGET_FILE%.avi`

B.10 WebM Project VP9 Encoder v1.3.0

Encoder title	WebM Project VP9 Encoder v1.3.0
Version	1.3.0
Developed by	The WebM Project



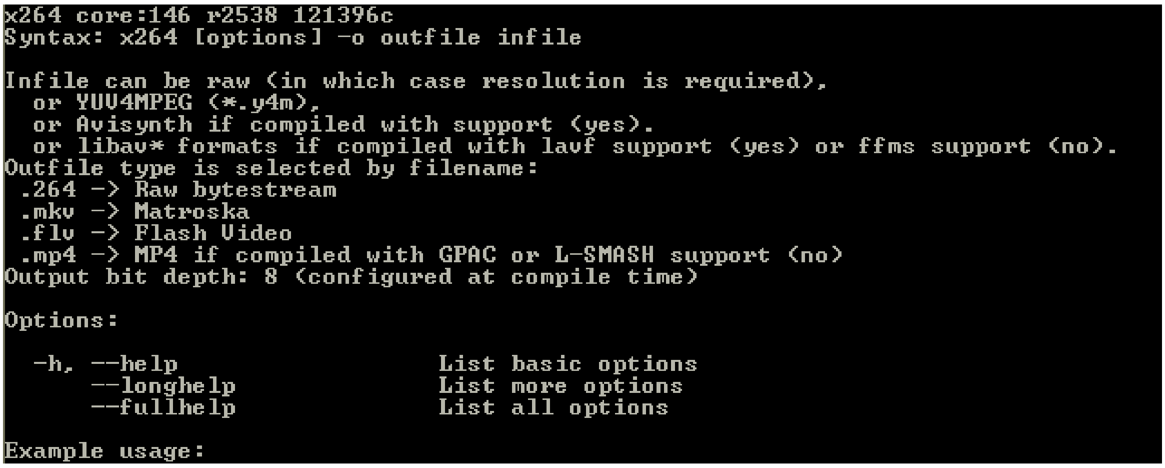
```
Usage: vpxenc.exe <options> -o dst_filename src_filename

Options:
  -D,          --debug                    Debug mode (makes output deterministic)
  -o <arg>, --output=<arg>                Output filename
               --codec=<arg>              Codec to use
  -p <arg>, --passes=<arg>                Number of passes (1/2)
               --pass=<arg>               Pass to execute (1/2)
               --fpf=<arg>                First pass statistics file name
               --limit=<arg>              Stop encoding after n input frames
               --skip=<arg>               Skip the first n input frames
  -d <arg>, --deadline=<arg>              Deadline per frame (usec)
               --best                     Use Best Quality Deadline
               --good                     Use Good Quality Deadline
               --rt                       Use Realtime Quality Deadline
  -q,          --quiet                    Do not print encode progress
  -v,          --verbose                  Show encoder parameters
               --psnr                     Show PSNR in status line
               --webm                     Output WebM (default when WebM IO is ena
bled)
               --ivf                      Output IVF
```

FIGURE 60: WebM Project VP9 Encoder v1.3.0

Platform	Preset name	Encoder parameters
Desktop	Ripping	`vpxenc.exe %SOURCE_FILE% -o %TARGET_FILE% -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% --test-decode=warn --target-bitrate=%BITRATE_KBPS% --codec=vp9 --profile=0 --kf-max-dist=90000 --static-thresh=0 --cpu-used=1 --tile-columns=3 -t 8 --min-q=2 --max-q=52 --undershoot-pct=25 --overshoot-pct=50 --buf-sz=1000 --buf-initial-sz=500 --buf-optimal-sz=600 --max-intra-rate=300 --resize-allowed=0 -p 2 --lag-in-frames=25 --end-usage=vbr --frame-parallel=0`
Server	Ripping	`vpxenc.exe %SOURCE_FILE% -o %TARGET_FILE% -w %WIDTH% -h %HEIGHT% --fps=%FPS_NUM%/%FPS_DENOM% --test-decode=warn --target-bitrate=%BITRATE_KBPS% --codec=vp9 --profile=0 --kf-max-dist=90000 --static-thresh=0 --cpu-used=1 --tile-columns=3 -t 8 --min-q=2 --max-q=62 --undershoot-pct=50 --overshoot-pct=50 --buf-sz=1000 --buf-initial-sz=500 --buf-optimal-sz=600 --max-intra-rate=300 --resize-allowed=0 -p 2 --lag-in-frames=25 --end-usage=vbr --frame-parallel=0`

B.11 x264

Encoder title	x264
Version	146 r2538 121396c
Developed by	x264 Developer Team



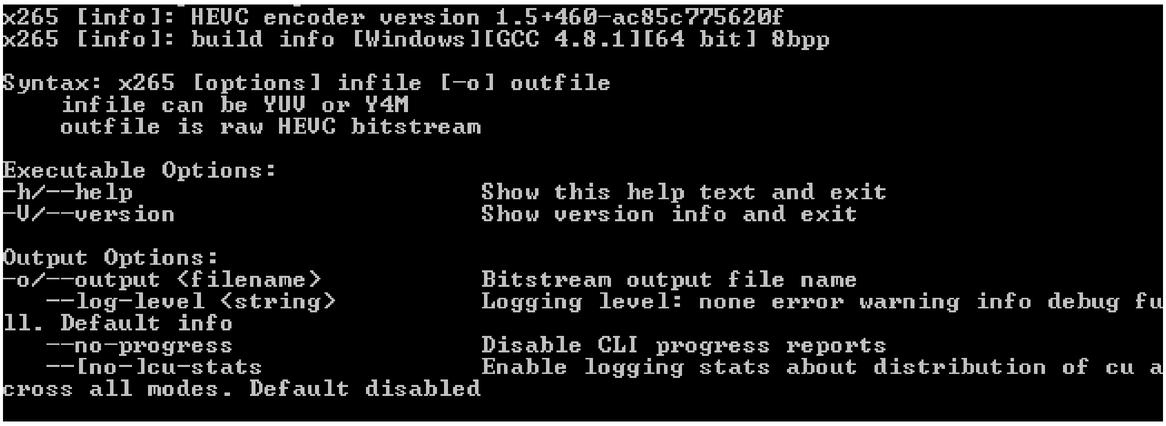
```
x264 core:146 r2538 121396c
Syntax: x264 [options] -o outfile infile

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

Example usage:
```

FIGURE 61: x264 encoder

Platform	Preset name	Encoder parameters
Desktop, Server	Ripping	`x264 --tune ssim --preset placebo --me umh --merange 32 --keyint infinite --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
		`x264 --tune ssim --preset placebo --me umh --merange 32 --keyint infinite --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Universal	`x264 --tune ssim --preset slow --trellis 2 --keyint infinite --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
		`x264 --tune ssim --preset slow --trellis 2 --keyint infinite --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Fast	`x264 --tune ssim --preset medium --keyint infinite --pass 1 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
		`x264 --tune ssim --preset medium --keyint infinite --pass 2 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`

B.12 x265

Encoder title	x265
Version	1.5+460-ac85c775620f
Developed by	x265 Developer Team



```
x265 [info]: HEVC encoder version 1.5+460-ac85c775620f
x265 [info]: build info [Windows][GCC 4.8.1][64 bit] 8bpp

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
   --log-level <string>          Logging level: none error warning info debug fu
ll. Default info
   --no-progress                 Disable CLI progress reports
   --[no-]cu-stats               Enable logging stats about distribution of cu a
cross all modes. Default disabled
```

FIGURE 62: x265 encoder

Platform	Preset name	Encoder parameters
Desktop	Ripping	`x265 -p veryslow --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Universal	`x265 -p medium --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Fast	`x265 -p ultrafast --ref 3 --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
Server	Ripping	`x265 -p veryslow --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Universal	`x265 -p fast --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`
	Fast	`x265 -p superfast --no-scenecut --bitrate %BITRATE_KBPS% %SOURCE_FILE% -o %TARGET_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS%`

C 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.

C.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.

C.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.

C.3 Graph Example

Figure 63 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.

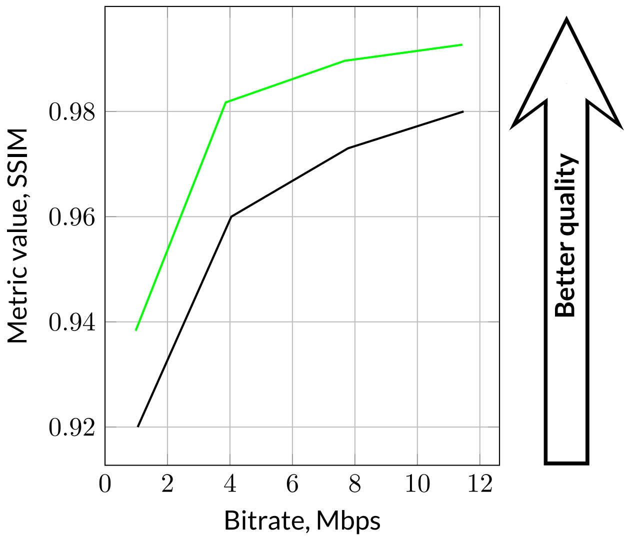
A more detailed description of the preparation of these graphs is given below.

C.4 Bitrates Ratio with the Same Quality

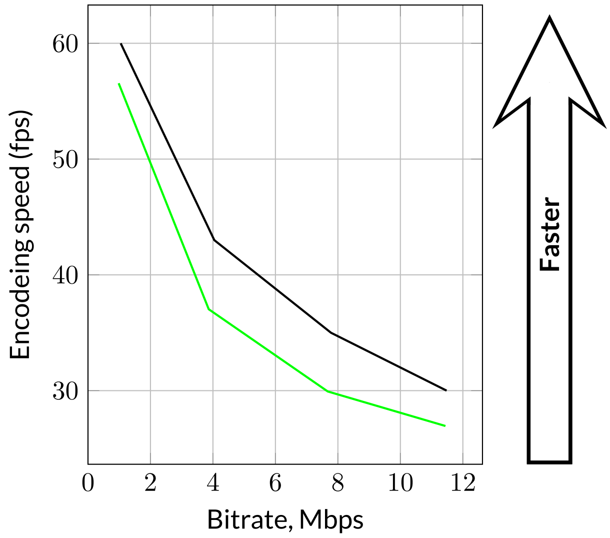
The first step in computing the average bitrate ratio for a fixed quality is inversion of the axes of the bitrate/quality graph (see Figure 64b). All further computations are performed using the inverted graph.

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.

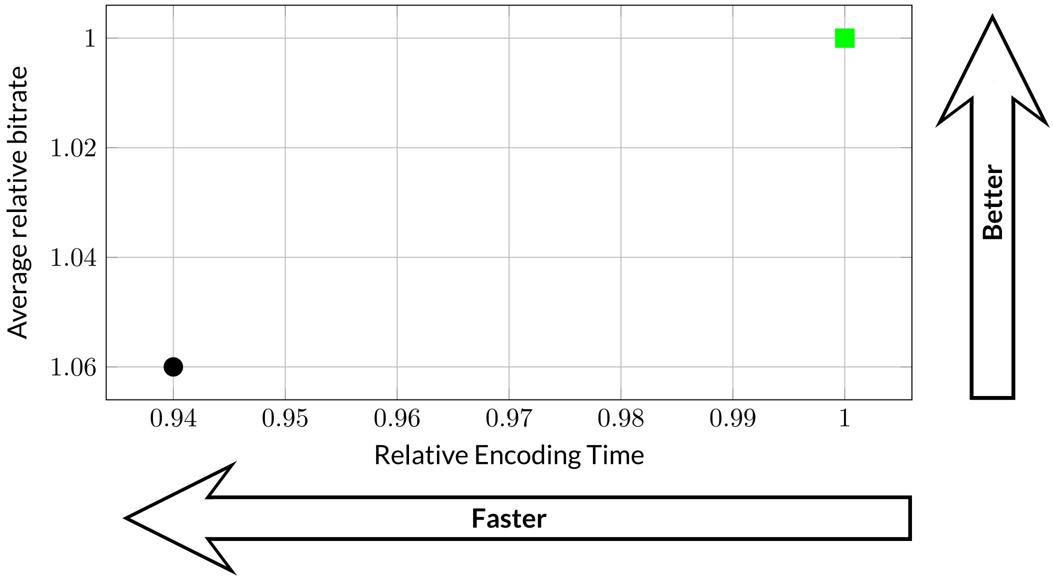
The final step is calculation of the area under the curves in the chosen interpolation segment and determination of their ratio (see Figure 64c). This result is an average bitrate ratio for a fixed quality for the two codecs. If



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

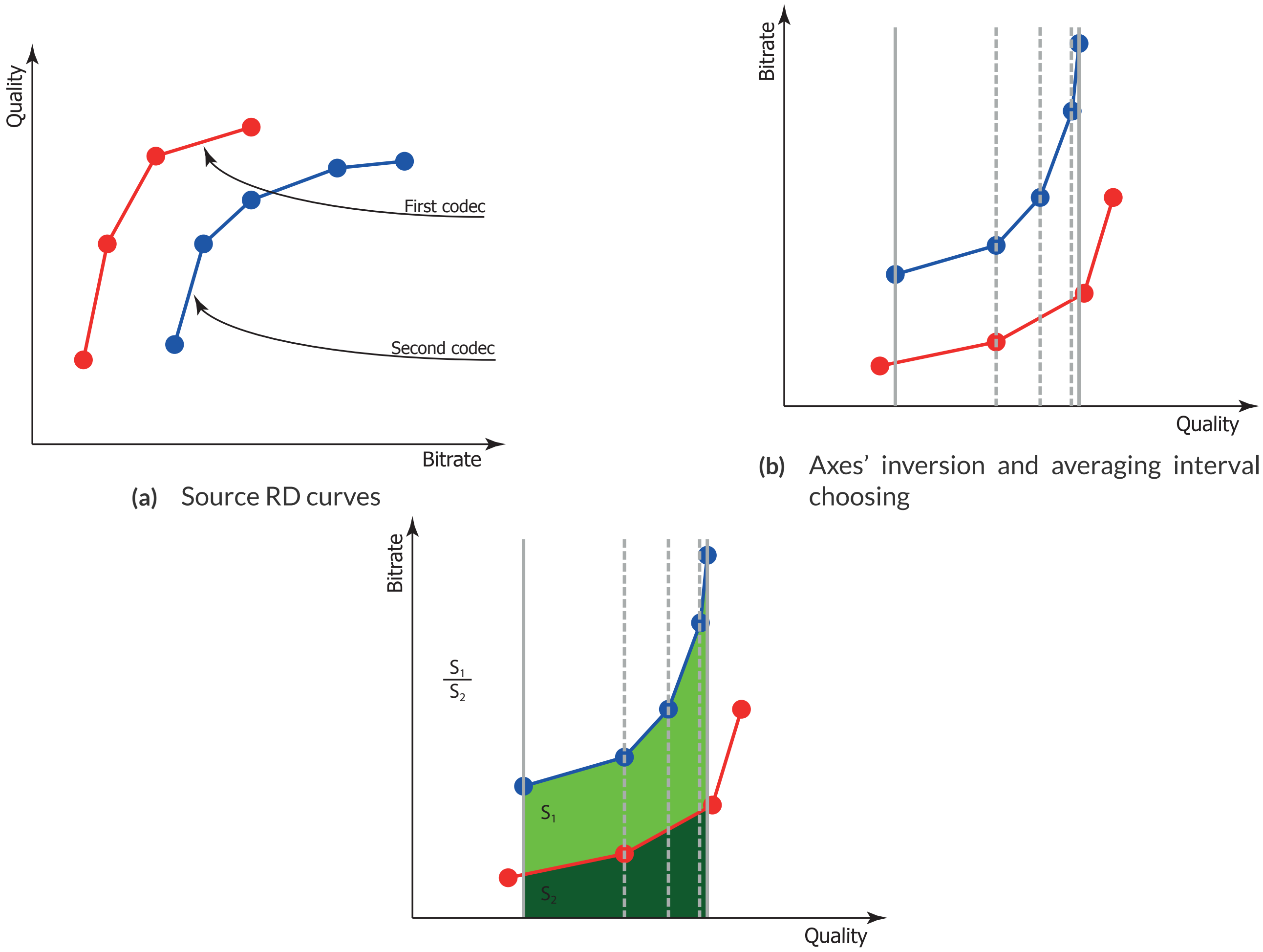


(b) Encoding speed (frames per second). "Green" codec is slower!



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

FIGURE 63: Speed/Quality trade-off example



(a) Source RD curves

(b) Axes' inversion and averaging interval choosing

(c) Areas under curves ratio

FIGURE 64: 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.

C.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 C.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 6: 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 64) 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.

D OBJECTIVE QUALITY METRICS DESCRIPTION

D.1 SSIM (Structural SIMilarity)

D.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:

$$\textbf{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x + \mu_y + C_1)(\sigma_x + \sigma_y + C_2)}, \tag{1}$$

where

$$\mu_x = \sum_{i=1}^{N} \omega_i x_i, \tag{2}$$

$$\sigma_x = \sqrt{\sum_{i=1}^{N} \omega_i (x_i - \mu_x)}, \tag{3}$$

$$\sigma_{xy} = \sum_{i=1}^{N} \omega_i (x_i - \mu_x)(y_i - \mu_y). \tag{4}$$

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 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.

D.1.2 Examples

Figure 65 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.

[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.

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

FIGURE 65: SSIM example for compressed image

Figure 66 depicts various distortions applied to original image and Figure 67 shows SSIM values for these distortions.

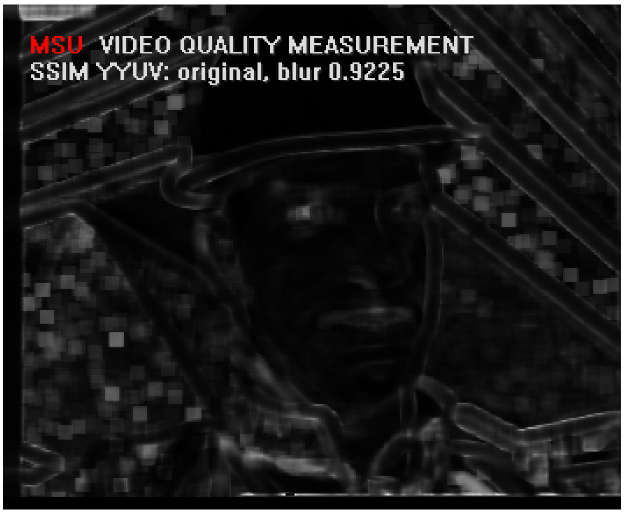
(a) Original image **(b)** Image with added noise

(c) Blurred image **(d)** Sharpen image

FIGURE 66: 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 67: SSIM values for original and processed images

E 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`