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

- Intel MSS HEVC Encoder (GPU)
- Kingsoft HEVC Encoder
- nj265
- SHBP H.265 Real time encoder

Non H.265

- nj264
- x264

CS MSU Graphics & Media Lab, Video Group
December 1, 2016
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:

- 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.	Arashiyama	984	24	3840×2160
2.	Butterflies	874	30	3840×2160
3.	Disneyland	317	24	3840×2160
4.	Driving	1747	24	4096×2160
5.	Flight	999	30	3840×2160
6.	Foreman	248	24	3840×2160
7.	Little Girl	1531	30	4096×2160
8.	Mobile	355	24	3840×2160
9.	Outdoor Party	868	30	3840×2160
10.	Quadrocopter	1830	30	3840×2160

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.

2.2 Codecs

	Codec	Developer	Version
1.	Intel MSS HEVC Encoder (GPU)	Intel	Intel Media Server Studio 2017 R1
2.	Kingsoft HEVC Encoder	Kingsoft	Kingsoft Encoder V2.3.3.1
3.	nj264	Nanjing Yunyan Email: jtwen@tsinghua.edu.cn	1.0
4.	nj265	Nanjing Yunyan Email: jtwen@tsinghua.edu.cn	1.0
5.	SHBP H.265 Real time encoder	SHBP Codec's development team Email: lobasso@hotmail.com	1.0
6.	x264	x264 Developer Team	148 r2721 72d53ab

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 B.

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.

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

4 RD CURVES

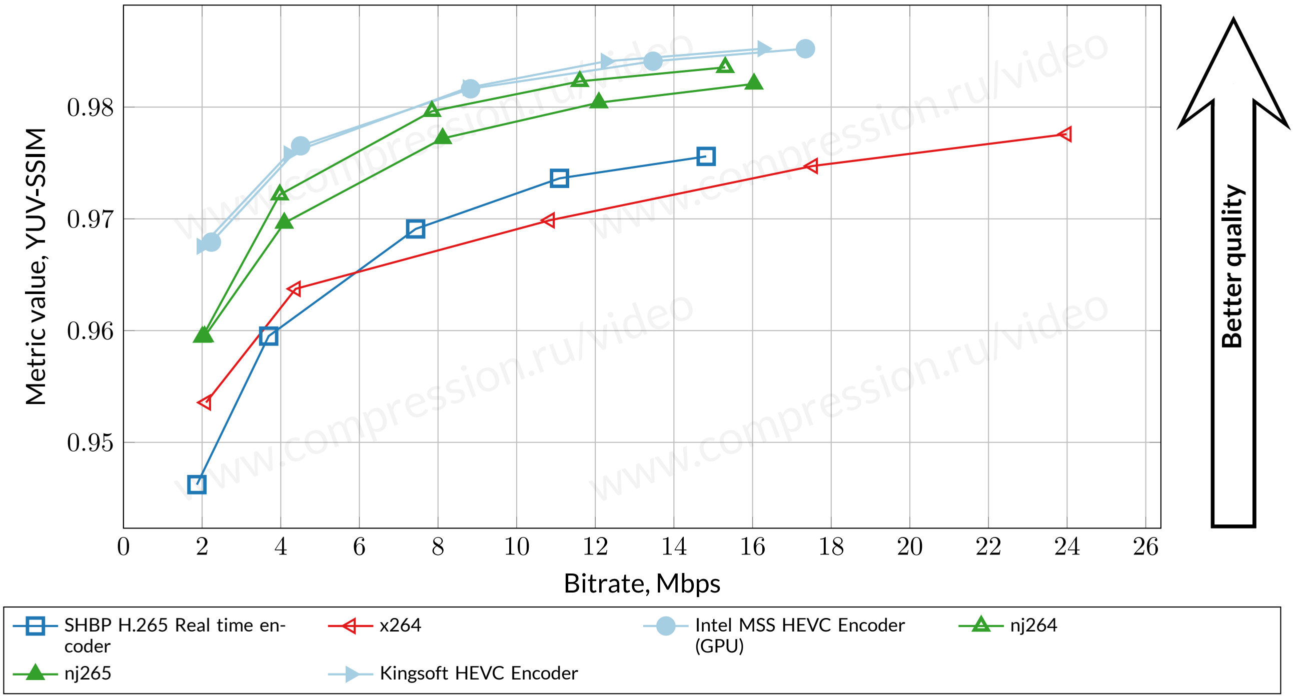


FIGURE 1: Bitrate/quality—usecase "4k Encoding," Disneyland sequence, YUV-SSIM metric

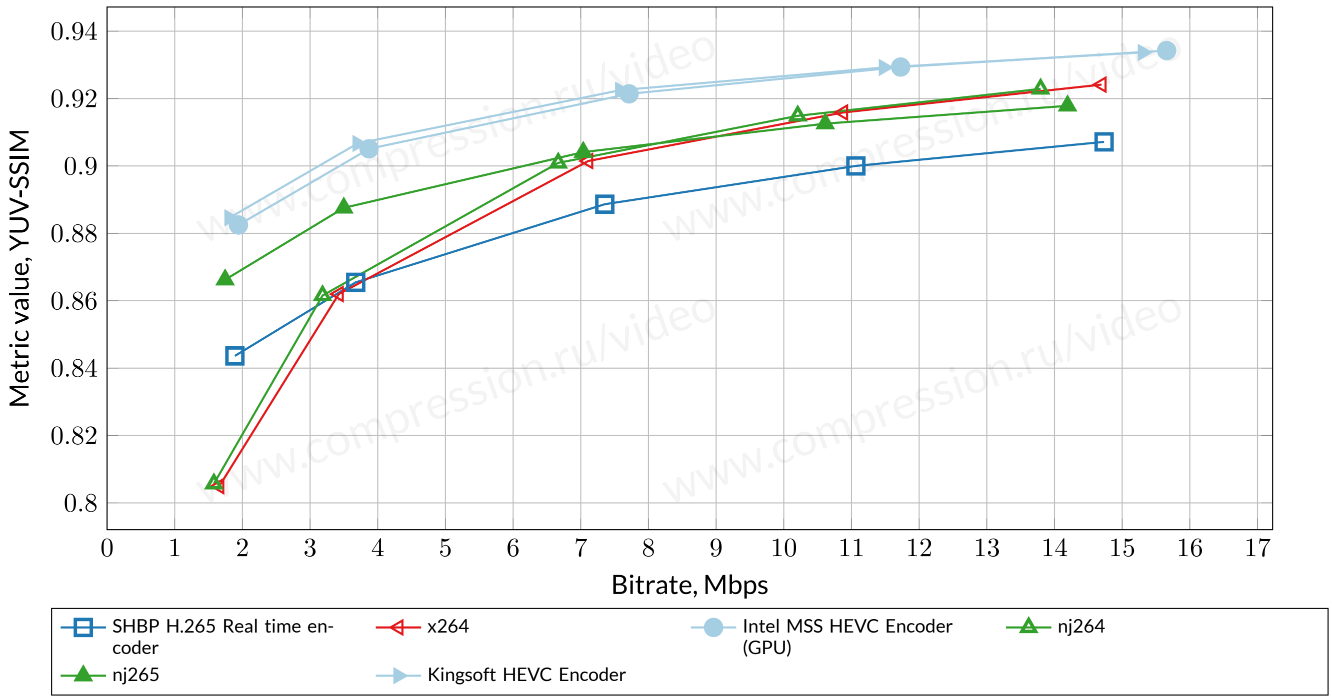


FIGURE 2: Bitrate/quality—usecase "4k Encoding," Outdoor Party sequence, YUV-SSIM metric

5 ENCODING SPEED

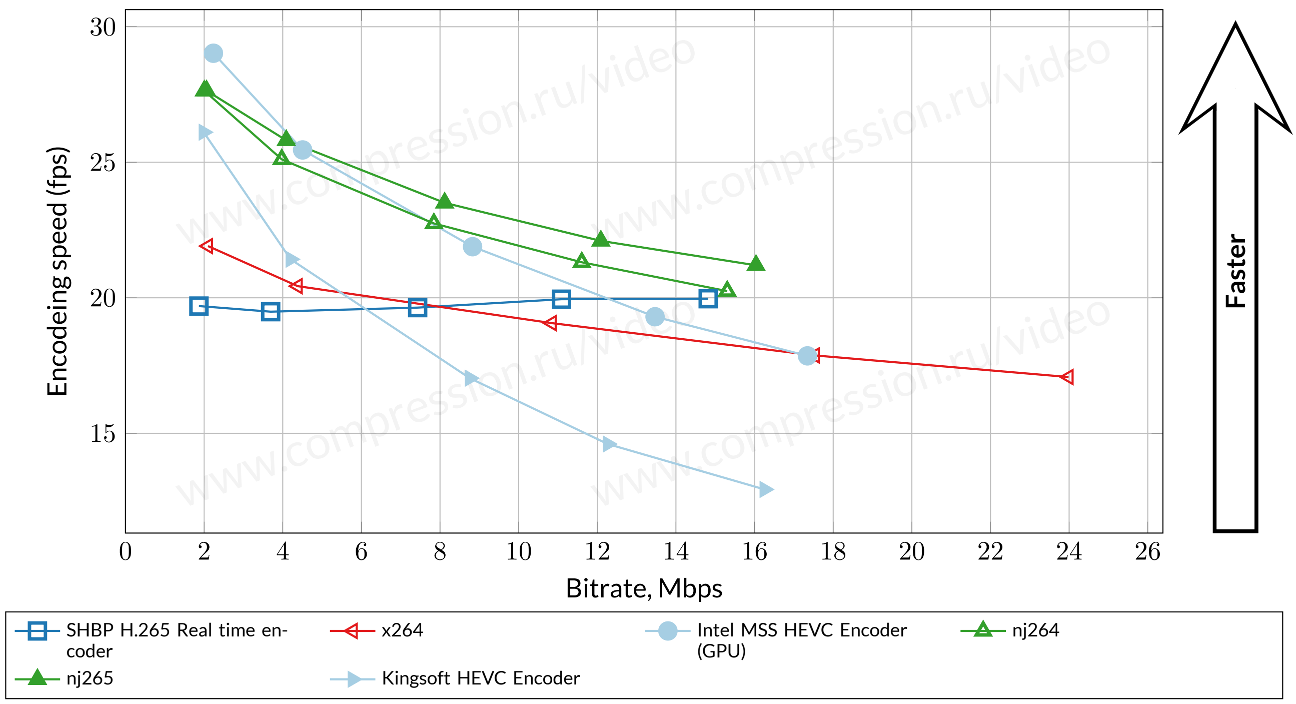


FIGURE 3: Encoding speed—usecase "4k Encoding," Disneyland sequence

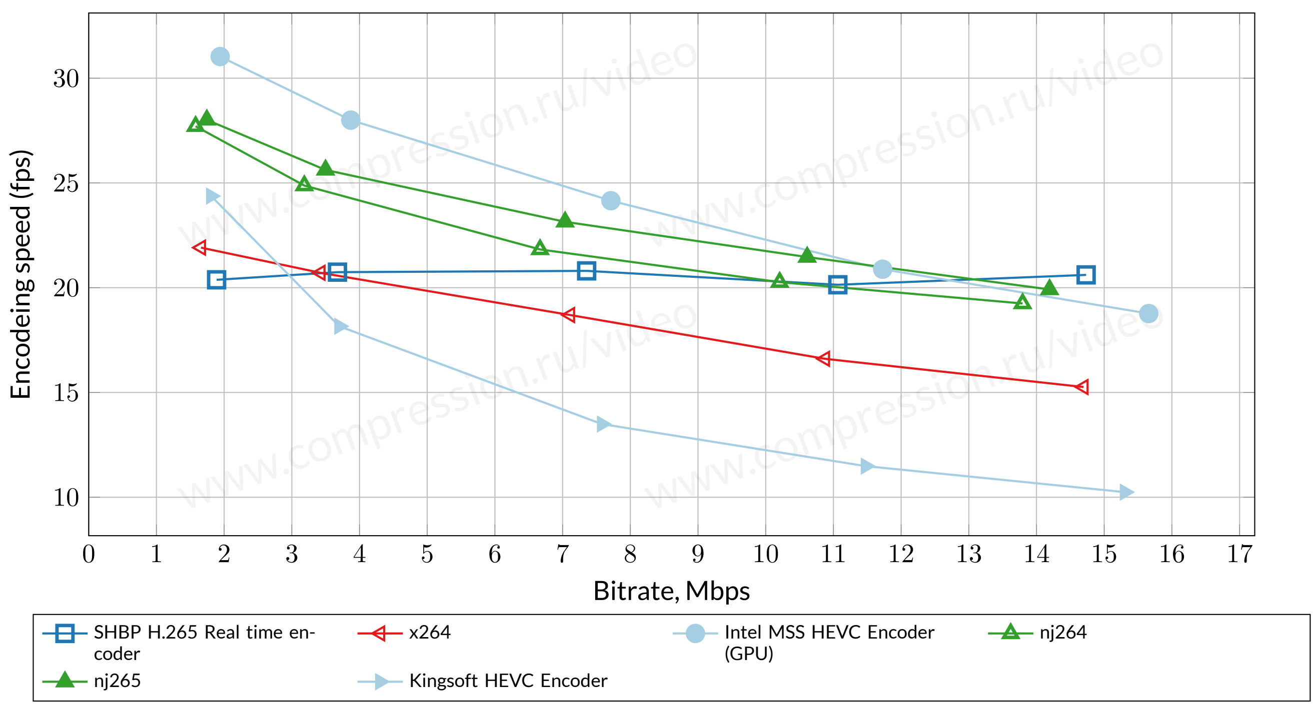


FIGURE 4: Encoding speed—usecase "4k Encoding," Outdoor Party sequence

6 SPEED/QUALITY TRADE-OFF

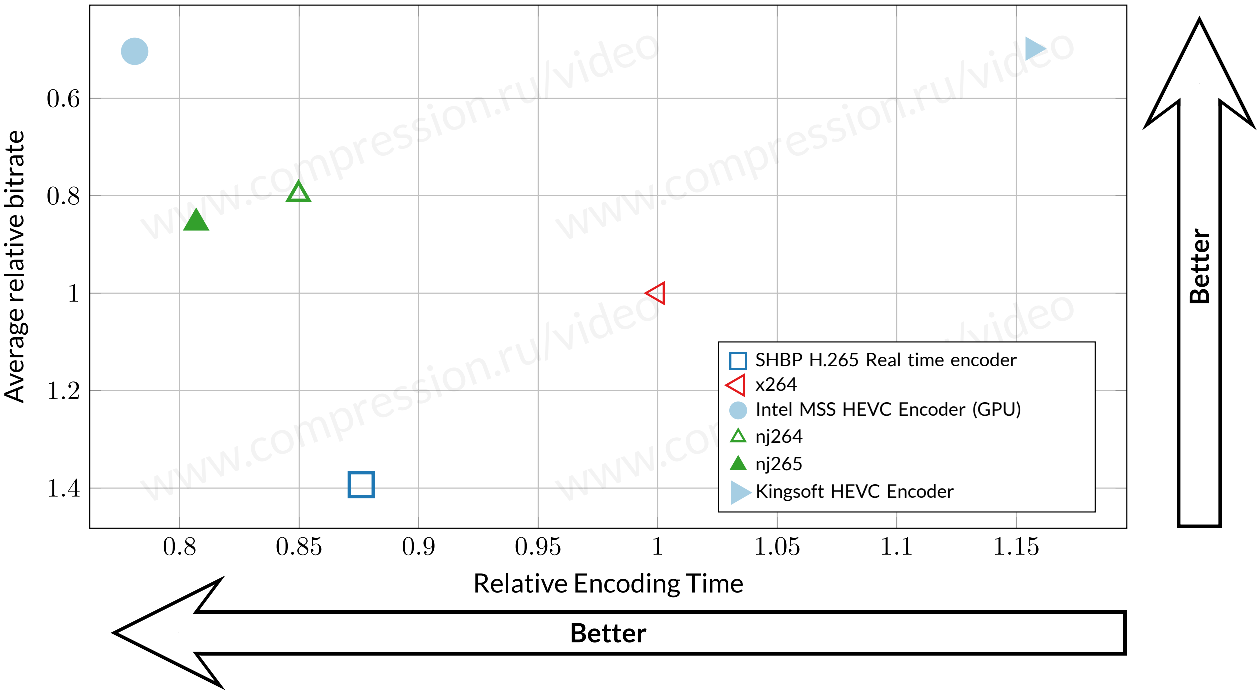


FIGURE 5: Speed/quality trade-off—usecase "4k Encoding," all sequences, YUV-SSIM metric

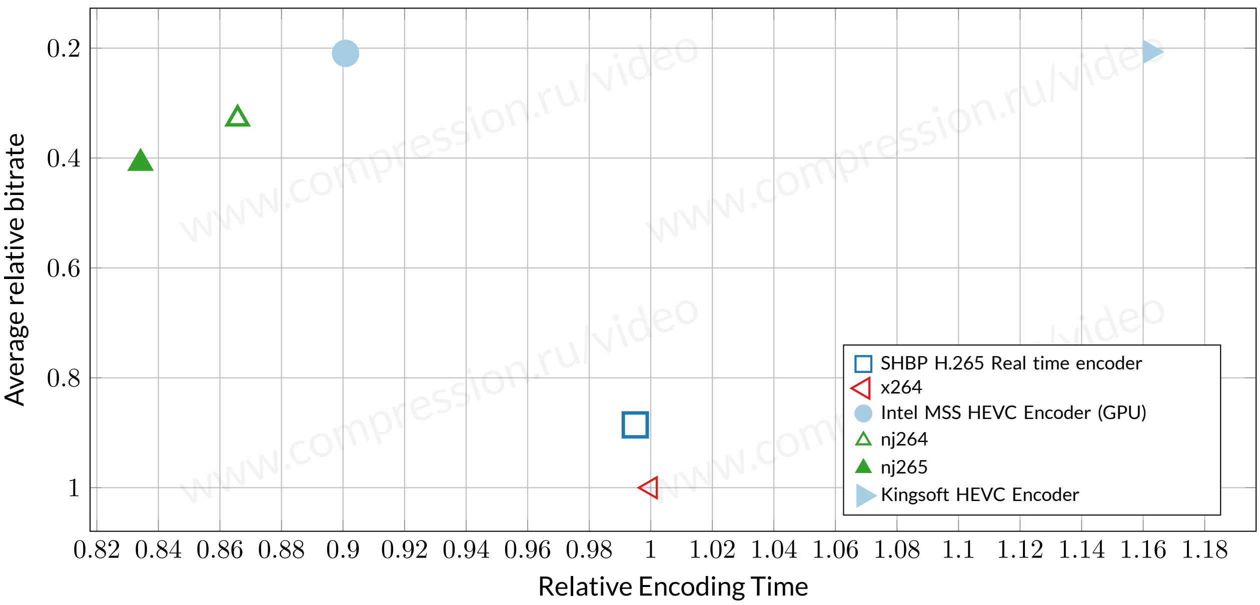


FIGURE 6: Speed/quality trade-off—usecase "4k Encoding," Disneyland sequence, YUV-SSIM metric

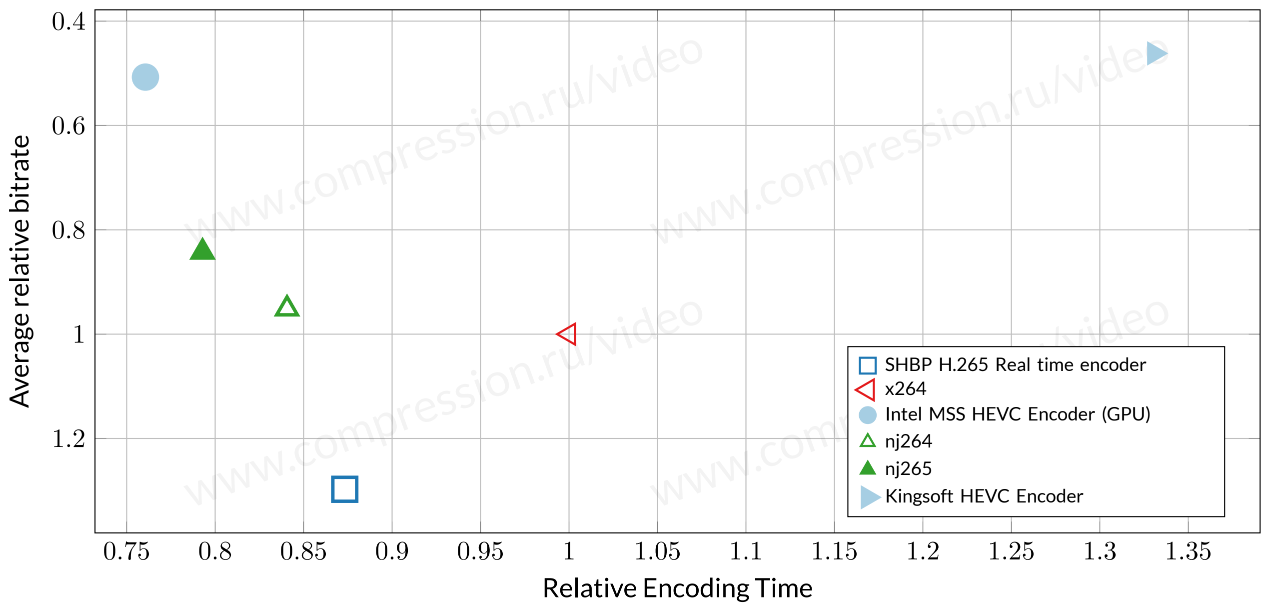


FIGURE 7: Speed/quality trade-off—usecase "4k Encoding," Outdoor Party sequence, YUV-SSIM metric

7 BITRATE HANDLING

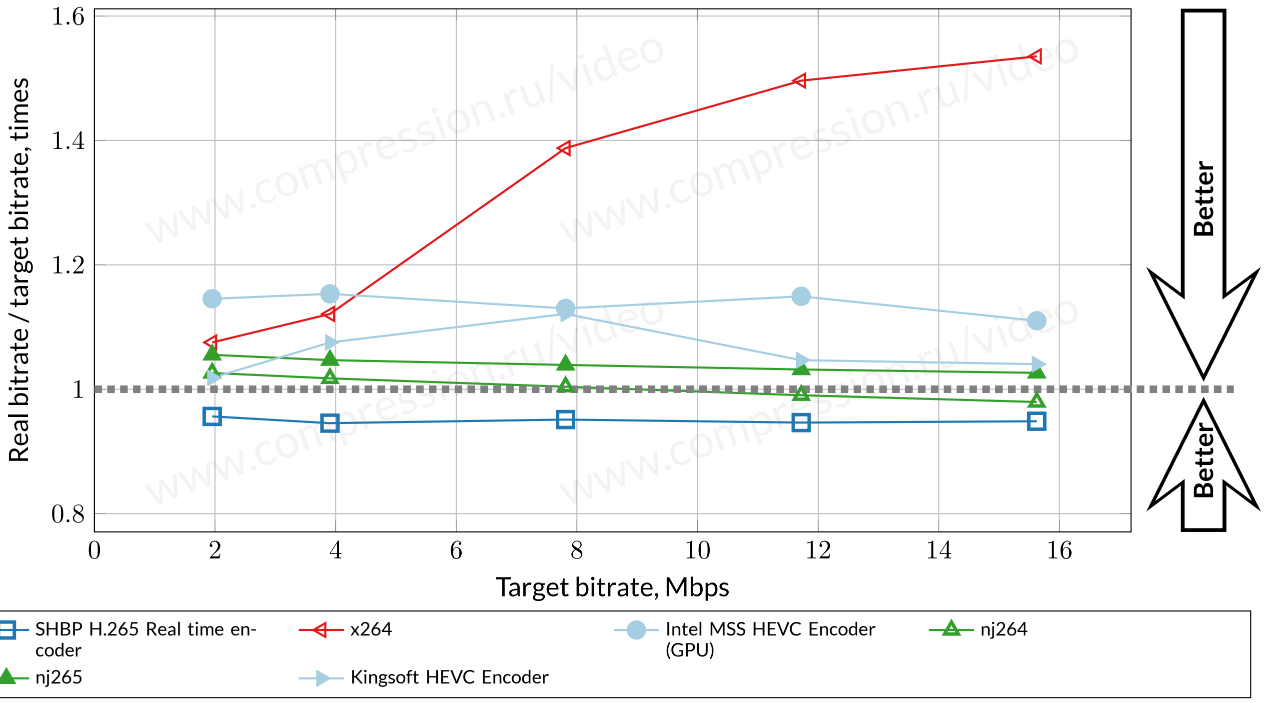


FIGURE 8: Bitrate handling—usecase "4k Encoding," Disneyland sequence

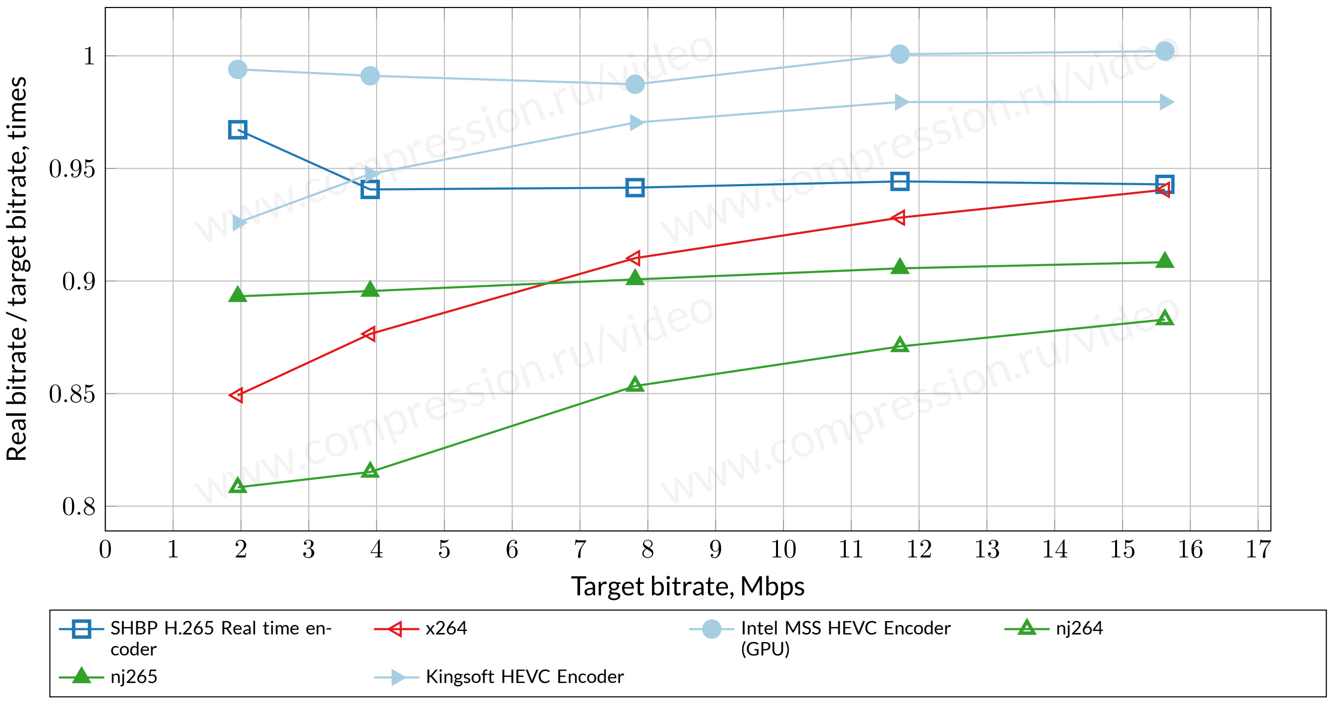


FIGURE 9: Bitrate handling—usecase "4k Encoding," Outdoor Party sequence

8 RELATIVE QUALITY ANALYSIS

	SHBP H.265 Real time encoder	x264	Intel MSS HEVC Encoder (GPU)	nj264	nj265	Kingsoft HEVC Encoder
SHBP H.265 Real time encoder	100%	72%	26%	61%	50%	26%
x264	146%	100%	57%	89%	95%	58%
Intel MSS HEVC Encoder (GPU)	413%	199%	100%	159%	199%	101%
nj264	173%	125%	64%	100%	111%	64%
nj265	217%	117%	53%	94%	100%	54%
Kingsoft HEVC Encoder	441%	201%	100%	161%	209%	100%

TABLE 3: Average bitrate ratio for a fixed quality—usecase "4k Encoding," YUV-SSIM metric

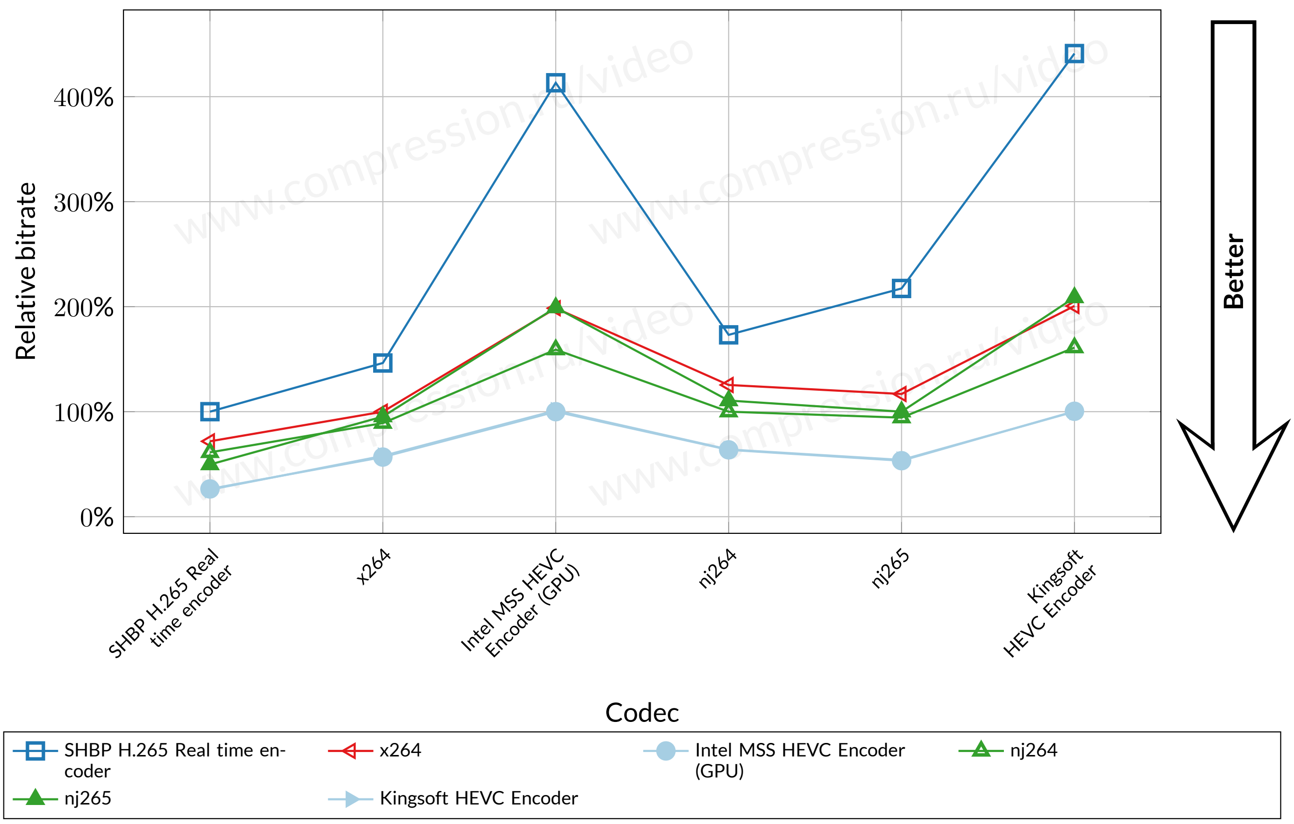


FIGURE 10: Average bitrate ratio for a fixed quality—usecase "4k Encoding," YUV-SSIM metric

9 CONCLUSION

Intel MSS HEVC Encoder (GPU) is the best codec in terms of speed-quality: it has almost the same encoding quality as Kingsoft HEVC Encoder and it is 1.5 times faster at average. If we consider only software encoders (i.e. non-hardware accelerated), there are 3 Pareto-optimal codecs: nj264, nj265, and Kingsoft HEVC Encoder.

Bitrate handling:

- x264 has bitrate handling issues
- Kingsoft HEVC encoder has also some issues
- there are some sequences that are difficult in terms of bitrate handling for almost all codecs

Encoding speed: There are no codecs with significant encoding-speed issues. Nevertheless, encoding speed is notably affected by the type of input sequences.

The similar observation can be made for speed-quality tradeoff. Intel MSS HEVC Encoder takes the first place according to both speed and quality for "Driving" sequence. Nevertheless, for "Disneyland" it is among Pareto-optimal codecs: Intel MSS HEVC Encoder, nj264, nj265, and Kingsoft HEVC Encoder .

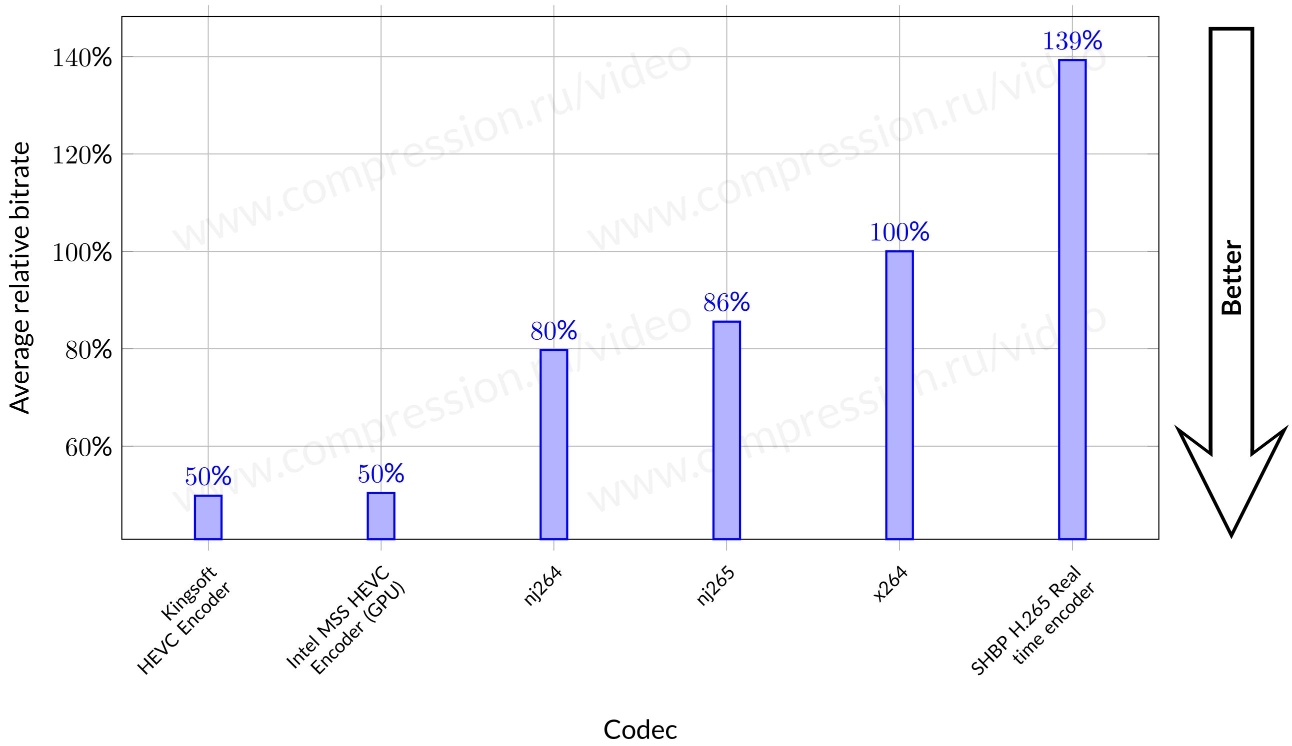


FIGURE 11: Average bitrate ratio for a fixed quality—usecase "4k Encoding," YUV-SSIM metric.

A SEQUENCES

A.1 "Arashiyama"

Sequence title	Arashiyama
Resolution	3840×2160
Number of frames	984
Color space	YV12
Frames per second	24
Source	https://vimeo.com/142480565#t=85

Several sights of famous Japanese nature park Arashiyama. There are some static camera views and mostly rotating views occur.

FIGURE 12: Arashiyama sequence, frame 24

A.2 “Butterflies”

Sequence title	Butterflies
Resolution	3840×2160
Number of frames	874
Color space	YV12
Frames per second	30
Source	https://vimeo.com/128901681#t=70

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

FIGURE 13: Butterflies sequence, frame 30

A.3 "Disneyland"

Sequence title	Disneyland
Resolution	3840×2160
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 14: Disneyland sequence, frame 24

A.4 “Driving”

Sequence title	Driving
Resolution	4096×2160
Number of frames	1747
Color space	YV12
Frames per second	24

The camera is set on the car rapidly driving Mulholland Drive in the evening.

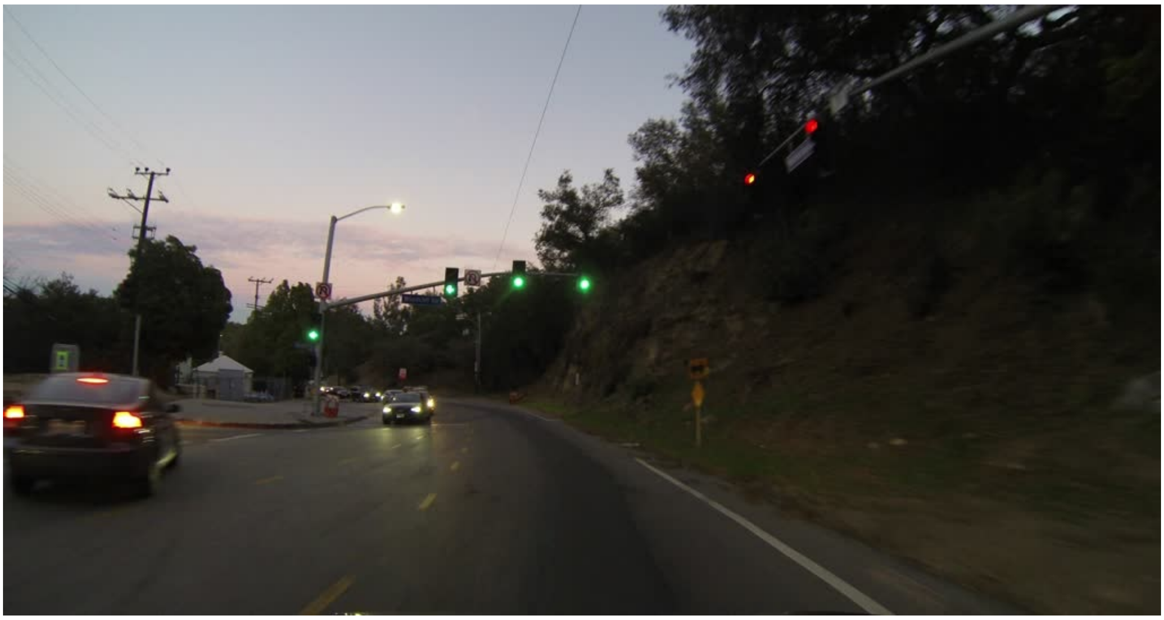


FIGURE 15: Driving sequence, frame 24

A.5 “Flight”

Sequence title	Flight
Resolution	3840×2160
Number of frames	999
Color space	YV12
Frames per second	25
Source	https://vimeo.com/147882650#t=40

A quadrocopter shooting views of the forest and some houses on the hill.

FIGURE 16: Flight sequence, frame 25

A.6 "Foreman"

Sequence title	Foreman
Resolution	3840×2160
Number of frames	248
Color space	YV12
Frames per second	24
Source	http://www.elementaltechnologies.com/resources/4k-test-sequences

The foreman in front emotionally tells something and waves his hand. Then camera quickly rotates to the right showing some object under construction.

FIGURE 17: Foreman sequence, frame 24

A.7 "Little Girl"

Sequence title	Little Girl
Resolution	4096×2160
Number of frames	1531
Color space	YV12
Frames per second	30

The little girl is playing toy blocks. The camera slowly moves to the left.

FIGURE 18: Little Girl sequence, frame 30

A.8 "Mobile"

Sequence title	Mobile
Resolution	3840×2160
Number of frames	355
Color space	YV12
Frames per second	24
Source	`http://www.elementaltechnologies.com/resources/4k-test-sequences`

The toy train is pushing the ball in front of the picture. The camera slowly follows the train in second part of the video.

FIGURE 19: Mobile sequence, frame 24

A.9 "Outdoor Party"

Sequence title	Outdoor Party
Resolution	3840×2160
Number of frames	868
Color space	YV12
Frames per second	30
Source	https://vimeo.com/133945050#t=35

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

FIGURE 20: Outdoor Party sequence, frame 30

A.10 “Quadrocopter”

Sequence title	Quadrocopter
Resolution	3840×2160
Number of frames	1830
Color space	YV12
Frames per second	30
Source	https://vimeo.com/132733109#t=0

Real estate advertisement with computer render of a quadrocopter.

FIGURE 21: Quadrocopter sequence, frame 240

B CODECS

B.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
4K	`mfx_transcoder.exe -h265 -hw -sys -encode_plugin mfxplugin64_hevce_gacc.dll -i %SOURCE_FILE% -w %WIDTH% -h %HEIGHT% -f %FPS% -o %TARGET_FILE% -b %BITRATE_BPS% -avbr -u 7 -async 3`

B.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 22: SHBP H.265 Real time encoder

Preset name	Encoder parameters
4K	`sh_hevc_enc.exe -w %WIDTH% -h %HEIGHT% -f %FPS% -n %FRAMES_NUM% -p 20.0 -b %BITRATE_KBPS% -i %SOURCE_FILE% -o %TARGET_FILE%`

B.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 23: x264 encoder

Preset name	Encoder parameters
4K	`x264 --preset fast --keyint infinite --tune ssim --bitrate %BITRATE_KBPS% %SOURCE_FILE% --input-res %WIDTH%x%HEIGHT% --fps %FPS% -o %TARGET_FILE%`

B.4 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
4K	`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%`

B.5 nj265

Encoder title	nj265
Version	1.0
Developed by	Nanjing Yunyan

Preset name	Encoder parameters
4K	`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%`

B.6 KS265

Encoder title	Kingsoft Encoder
Version	V2.1.1
Developed by	Kingsoft

Preset name	Encoder parameters
4K	`AppEncoder_x64.exe -i %SOURCE_FILE% -wdt %WIDTH% -hgt %HEIGHT% -fr %FPS% -br %BITRATE_KBPS% -b %TARGET_FILE% -preset slow -rc 1 -part 0 -sao 1 -me 1 -iNxN 0`

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 24 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.

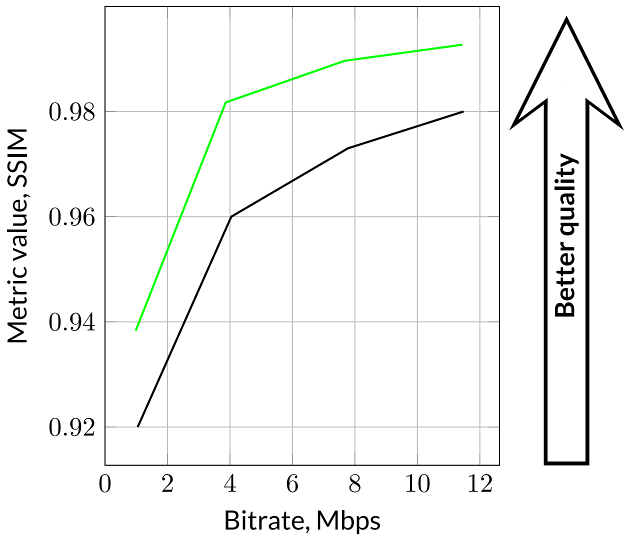
A more detailed description of the preparation of these graphs is given below.

C.4 Bitrates Ratio with the Same Quality

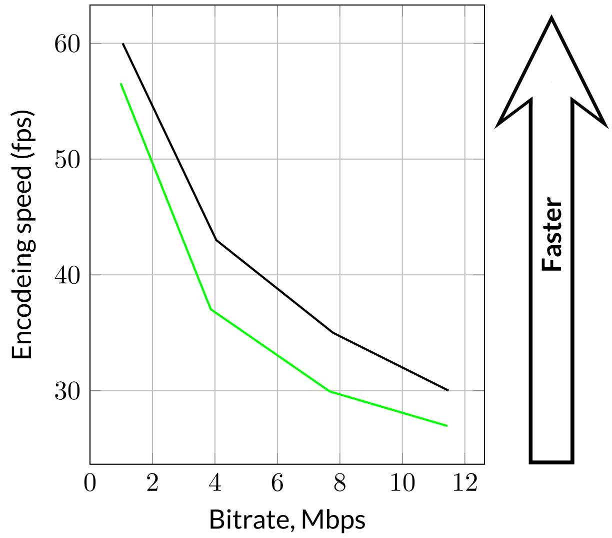
The first step in computing the average bitrate ratio for a fixed quality is inversion of the axes of the bitrate/quality graph (see Figure 25b). All further computations are performed using the inverted graph.

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.

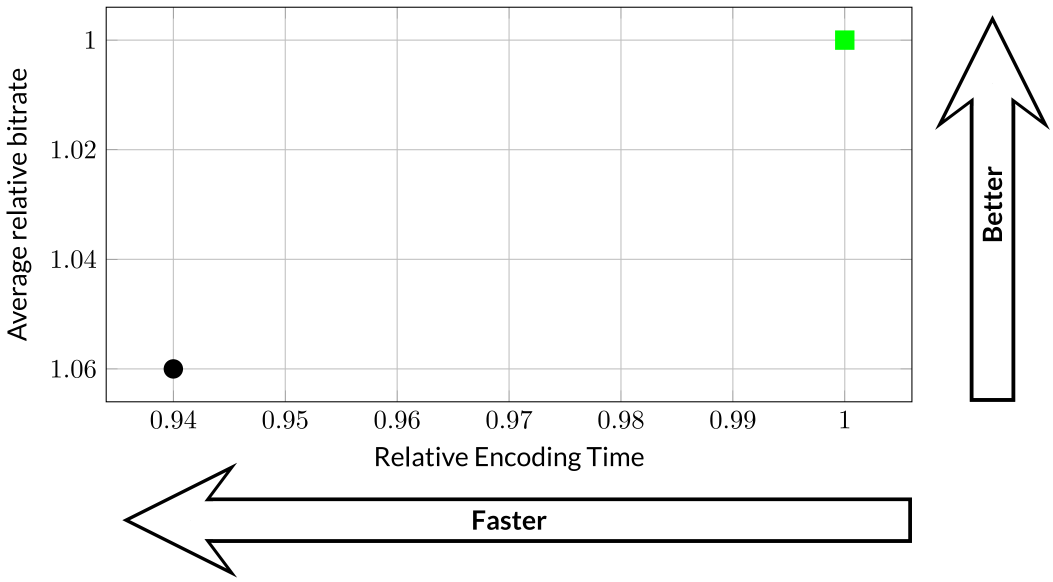
The final step is calculation of the area under the curves in the chosen interpolation segment and determination of their ratio (see Figure 25c). This result is an average bitrate ratio for a fixed quality for the two codecs. If



(a) RD curve. “Green” codec is better!

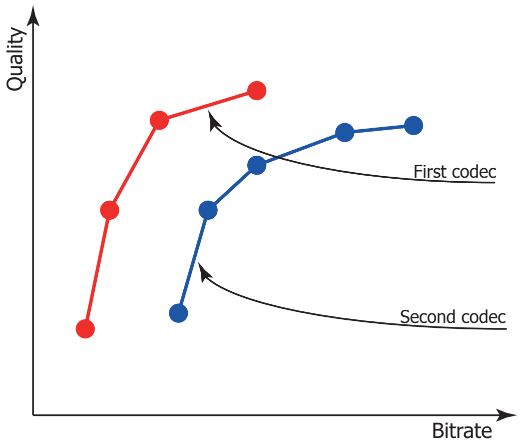


(b) Encoding speed (frames per second). “Green” codec is slower!

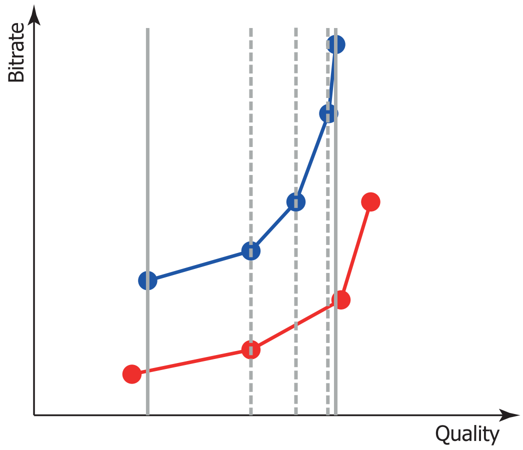


(c) Integral situation with codecs. This plot shows the situation more clearly

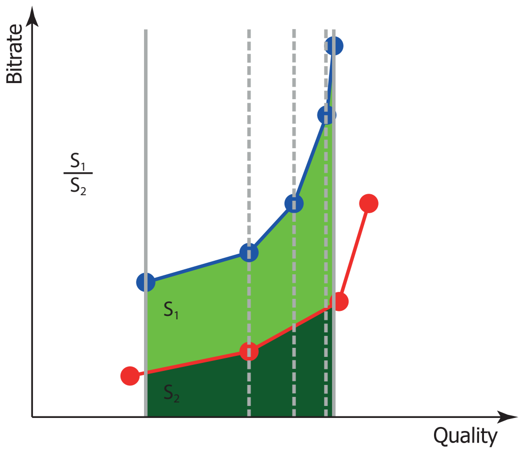
FIGURE 24: Speed/Quality trade-off example



(a) Source RD curves



(b) Axes' inversion and averaging interval choosing



(c) Areas under curves ratio

FIGURE 25: 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 4: 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 25) 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)

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.

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:

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

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.

D.1.2 Examples

Figure 26 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 26: SSIM example for compressed image

Figure 27 depicts various distortions applied to original image and Figure 28 shows SSIM values for these distortions.

(a) Original image

(b) Image with added noise

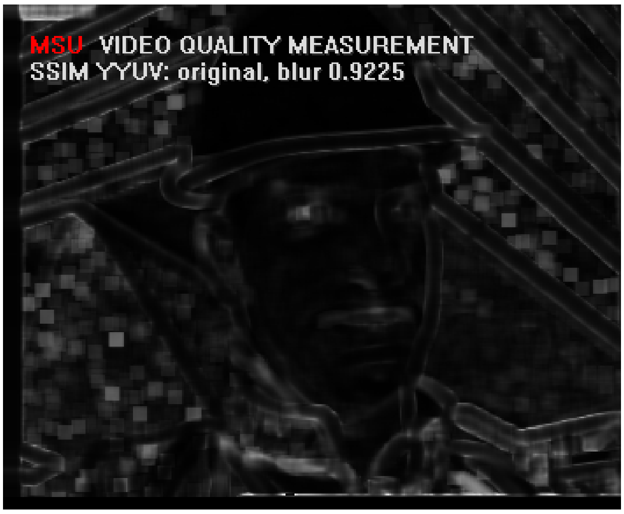
(c) Blurred image

(d) Sharpen image

FIGURE 27: 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 28: SSIM values for original and processed images

E ABOUT THE GRAPHICS & MEDIA LAB VIDEO GROUP

Graphics & Media Lab Video Group

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