MPEG-4 AVC/H.264 Video Codecs Comparison

GRAPHICS & MEDIA LAB
VIDEO GROUP

Short version of report

Project head: Dr. Dmitriy Vatolin
Measurements, analysis: Dr. Dmitriy Kulikov,
Alexander Parshin

Codecs:

DivX H.264

Elecard H.264

Intel® MediaSDK AVC/H.264

MainConcept H.264

Microsoft Expression Encoder

Theora

x264

XviD (MPEG-4 ASP codec)

April 2010

CS MSU Graphics&Media Lab

Video Group

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:

- Elecard Ltd
- Intel Corporation
- MainConcept GmbH
- Theora Development Team
- x264 Development Team
- XviD Development Team

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

2 Overview

2.1 Sequences

Table 1. Summary of video sequences.

Sequence	Number of frames	Frame rate	Resolution
1. Ice Age	2014	24	720x480
2. Up	1920	24	720x480
3. State Enemy	6500	24	720x304
4. Indiana Jones	5000	30	704x288
5. Mobile Calendar	504	50	1280x720
6. Iron Man	600	24	1920x1080
7. Troy	300	24	1920x1072
8. Amazon	1200	24	1280x720

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 5. Test Set of Video Sequences.

2.2 Codecs

Table 2. Short codec descriptions

Codec	Developer	Version
1. DivX AVC/H.264 Video Encoder	DivX, Inc.	version 1.1.1.6
2. Elecard AVC Video Encoder 8-bit edition,	Elecard Ltd	2.1.022202.091207
3. Intel® MediaSDK AVC/H.264 transcoder sample appication	Intel Corp.	1.10.1.15
4. MainConcept AVC/H.264 Video Encoder Console Application	MainConcept GmbH	8.5.0.12837
5. Microsoft Expression Encoder 3	Microsoft Corp.	3.0.1332.0
6. Theora encoder	Xiph.Org.	Provided by developers
7. x264	x264 Development Team	x264 core:85 r1442M 781d300
8. XviD raw mpeg4 bitstream encoder	XviD Development Team	xvid-1.3.0-dev

Brief descriptions of the codecs used in our comparison are given in Table 2. XviD was used as a good quality MPEG-4 ASP reference codec for comparison purposes. Detailed descriptions of all codecs used in our comparison can be found in Appendix 6. Tested Codecs.

2.3 Objectives and Testing Rules

2.4 H.264 Codec Testing Objectives

The main goal of this report is the presentation of a comparative evaluation of the quality of new H.264 codecs using objective measures of assessment. The comparison was done using settings provided by the developers of each codec.

The main task of the comparison is to analyze different H.264 encoders for the task of transcoding video—e.g., compressing video for personal use. Speed requirements are given for a sufficiently fast PC; fast presets are analogous to real-time encoding for a typical home-use PC.

2.5 Testing Rules

- The entire test set was divided into two primary types of applications. These applications differ by resolution, bitrate and encoding speed requirements:
    - Movies (bitrates of 500-2000 kbps)
    - High-definition television (“HDTV”; bitrates of 0.7-10 mbps)
- There are special presets and speed limitations for every type of application:
    - Movies (speed requirements for 750 kbps 4CIF sequences):
        - Minimum 120 fps for "High Speed" preset (1 pass, no B-frames, 1 reference frame)
        - Minimum 80 fps for "Normal" preset (2 passes)
        - Minimum 40 fps for "High Quality" preset
    - HDTV (speed requirements for 3 mbps 1280x720 sequences):
        - Minimum 100 fps for "High Speed" preset (1 pass, no B-frames, 1 reference frame)
        - Minimum 50 fps for "Normal" preset (2 passes)
        - Minimum 20 fps for "High Quality" preset
- Each codec’s developer provided settings for each type of application. Each setting’s individual parameters were, to a large extent, chosen by the developers, except the following:
    - DivX H.264
    - Microsoft Expression Encoder
    - Theora
- Each codec was tested for speed three times; the minimum score was then used as the representative time.
- During the testing process, source video sequences were in the YV12 format (.yuv file extension) for all codecs except Theora (.y4m with YV12)

- For all measurements the PRO version of the YUVsoft Video Codec Scoring System was used (http://www.yuvsoft.com/technologies/vicos/index.html).
- The following computer configuration was used for the main tests:
  - 4-cores processor: Intel Core i7 920, 2.67GHz
  - OS Name: Microsoft Windows 7 Professional 64-bit
  - Total Physical Memory: 6 GB
  - GPU: NVIDIA GeForce GTX 275

During the evaluation the following measures were used:

- SSIM (Y components)
- PSNR (Y components)

More detailed information about these measures may be found on the Internet at the following URL:

http://www.compression.ru/video/quality_measure/info.html

3 Comparison Results

3.1 Movies

The full results for (High Speed, Normal and High Quality presets) could be found in professional versions of this report:

- *H.264 Comparison Report Pro 2010 – Movies edition*
- *H.264 Comparison Report Pro 2010 – Enterprise edition*

See http://compression.ru/video/codec_comparison/h264_2010 webpage for purchase links.

3.1.1 RD Curves

3.1.1.1 Normal Preset

The Normal preset results for each sequence are presented in Figure 1 through **Error! Reference source not found.**. The first four figures show the Y-SSIM results.

SSIM metric: The leader is x264; MainConcept placed second, and DivX H.264 placed third. For the "Ice Age" and "Up" sequences, the MediaSDK and Elecard encoders exhibited similar results. For the "Indiana Jones" and "State Enemy" sequences, the XviD encoder placed fourth. Theora has a strange RD-curve quality drop at 1,000 kbps.

Theora was second to last on average for Y-SSIM, but for high bitrates, it had better results than did some codecs according to the Y-SSIM metric.

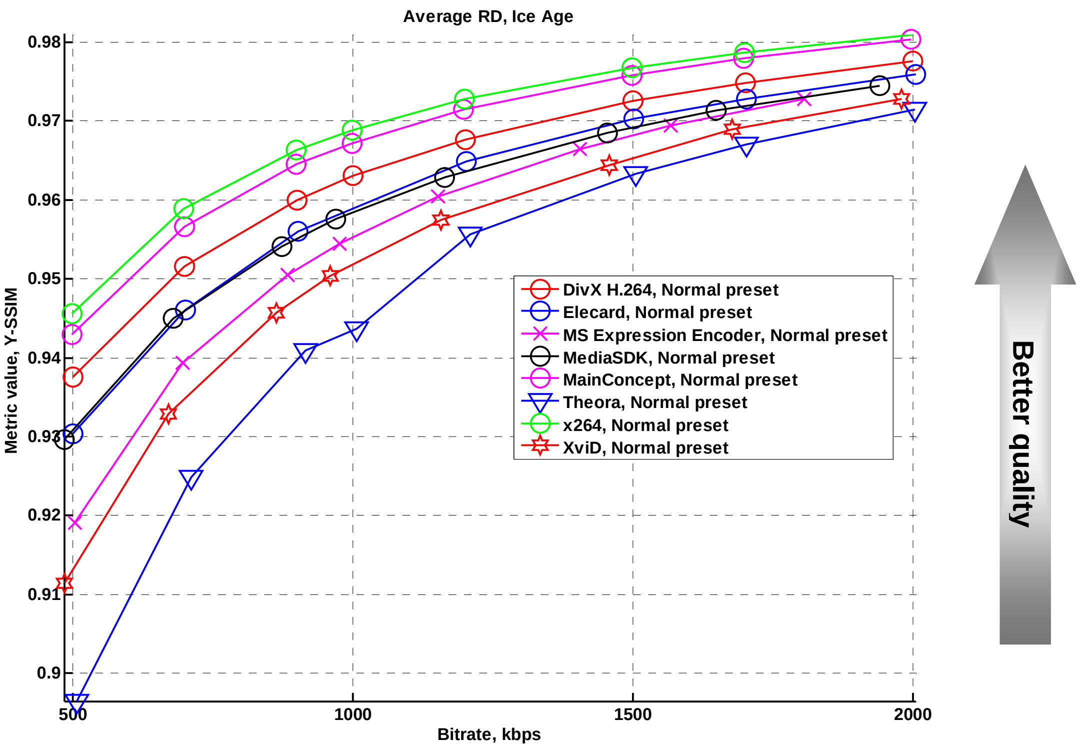


Figure 1. Bitrate/quality—usage area "Movies," "Ice Age" sequence, Normal preset, Y-SSIM metric

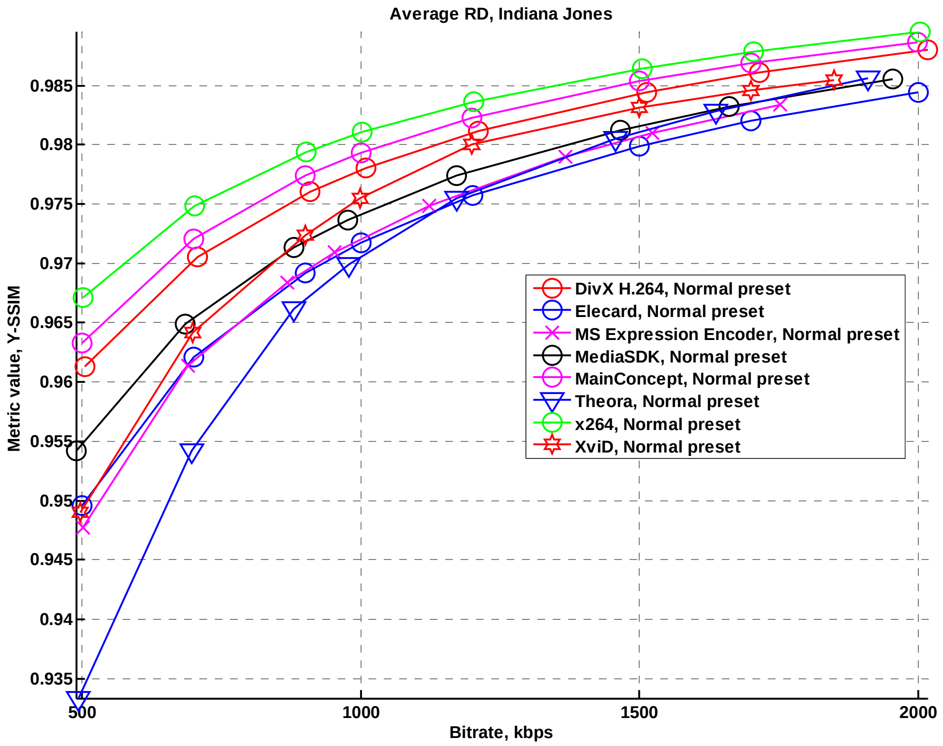


Figure 2. Bitrate/quality—usage area "Movies," "Indiana Jones" sequence, Normal preset, Y-SSIM metric

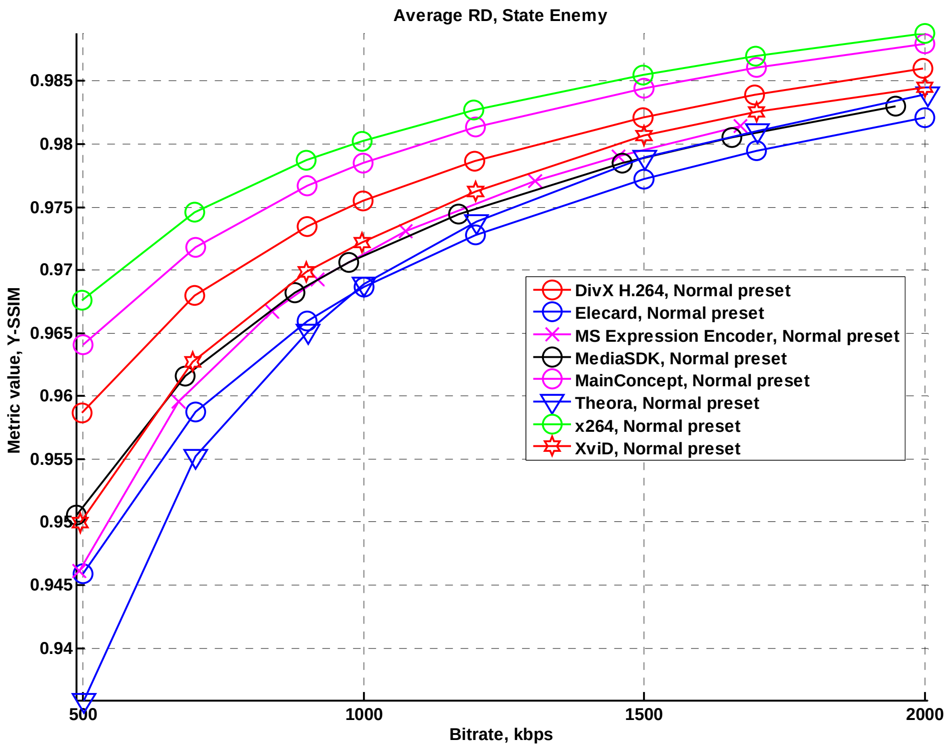


Figure 3. Bitrate/quality—usage area "Movies," "State Enemy" sequence,

Normal preset, Y-SSIM metric

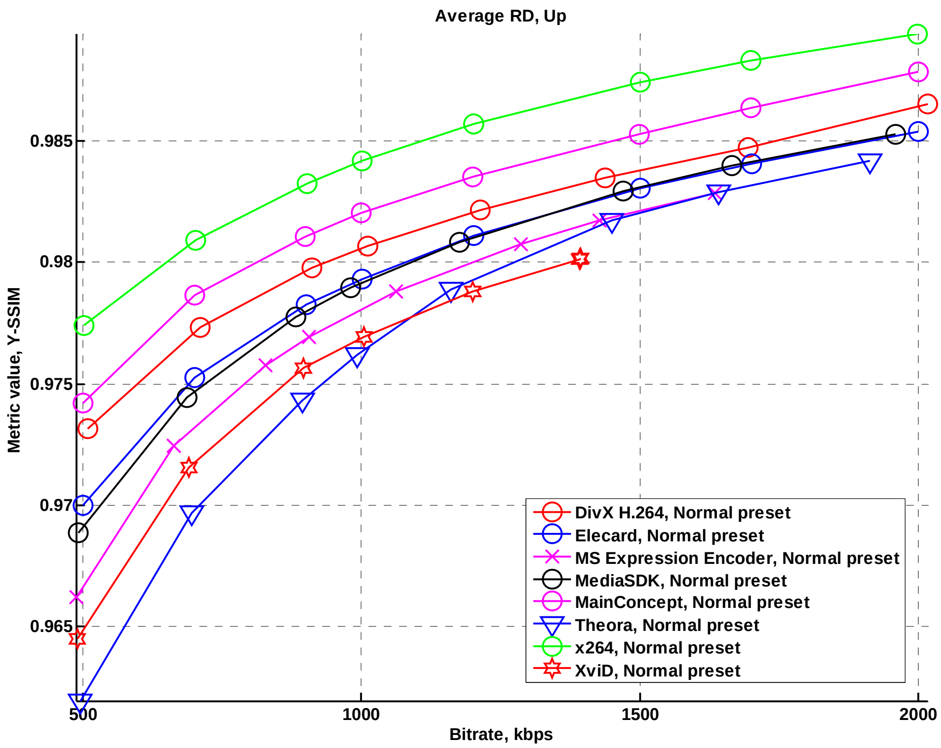


Figure 4. Bitrate/quality—usage area "Movies," "Up" sequence, Normal preset, Y-SSIM metric

3.1.2 Encoding Speed

3.1.2.1 Normal Preset

Absolute speed results are presented in Figure 5 through Figure 8. All the encoders except Microsoft Expression and Theora have a similar growth rate for encoding time versus increasing bitrate. Elecard is the fastest encoder. The encoding speed of the Microsoft Expression and Theora encoders exhibits almost no dependency on bitrate, and Microsoft Expression's encoding speed decreases at 1,200 kbps for the "Ice Age" sequence. XviD's encoding speed exhibits little dependency on bitrate for the "Up" sequence.

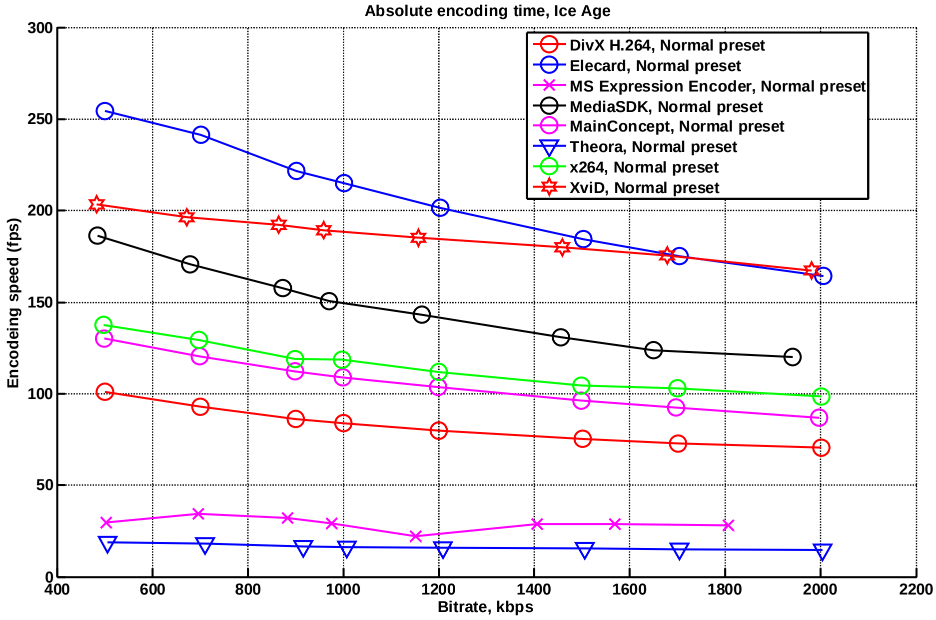


Figure 5. Encoding speed—usage area "Movies" "Ice Age" sequence, Normal preset

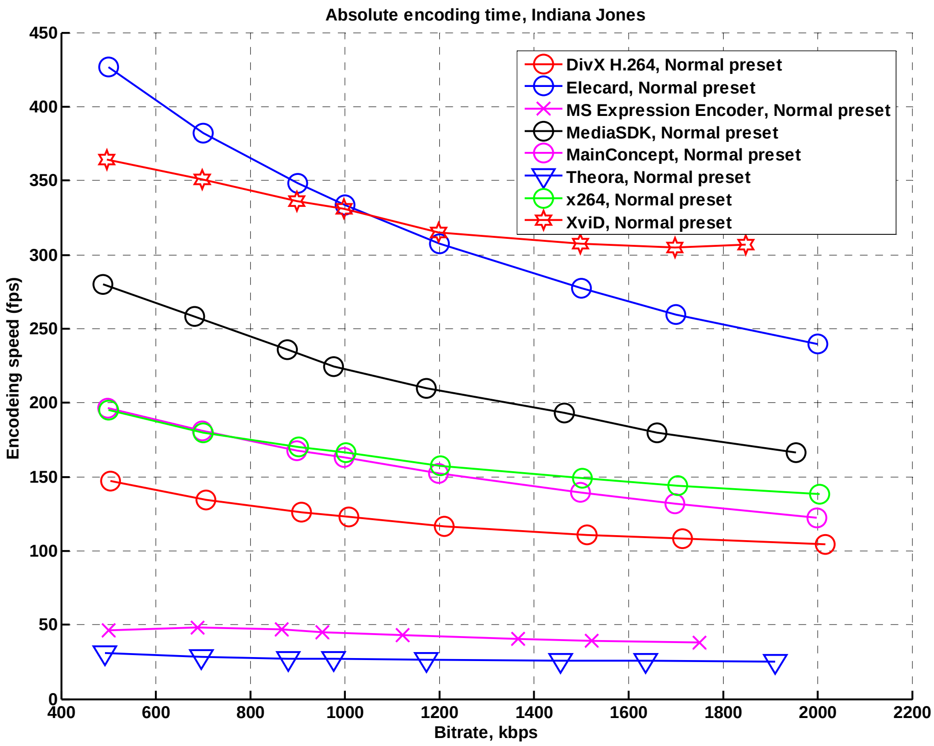


Figure 6. Encoding speed—usage area "Movies" "Indiana Jones" sequence, Normal preset

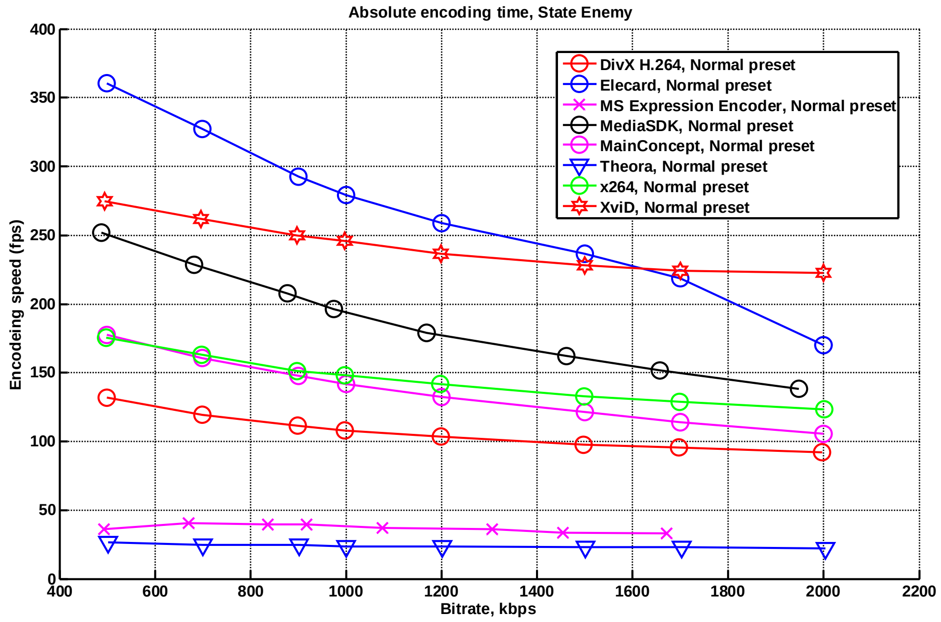


Figure 7. Encoding speed—usage area "Movies" "State Enemy" sequence, Normal preset

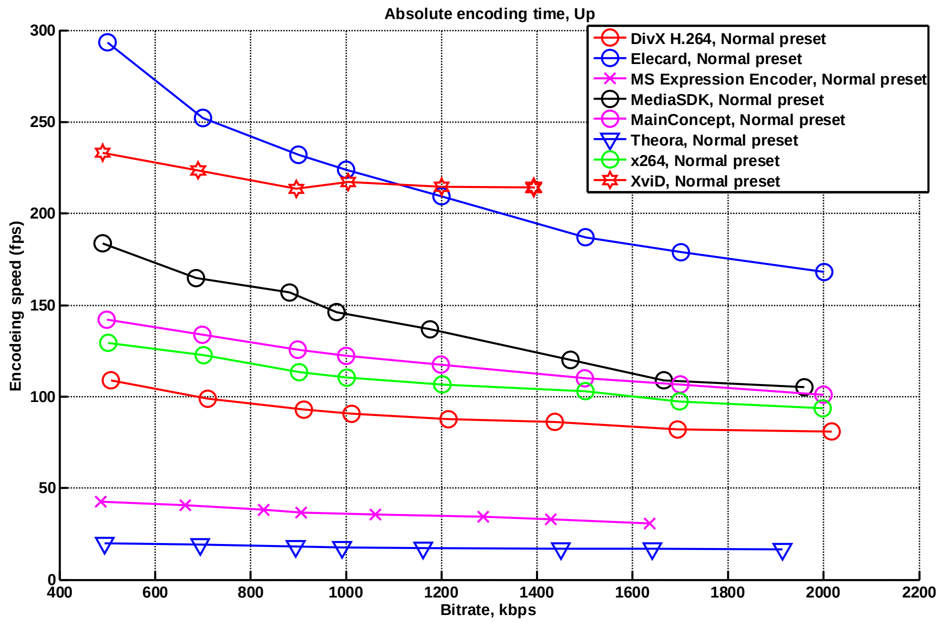


Figure 8. Encoding speed—usage area "Movies" "Up" sequence, Normal preset

3.1.3 Speed/Quality Trade-Off

Detailed descriptions of the speed/quality trade-off graphs can be found in Appendix 7. Figures Explanation. 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. XviD 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.

Please note that the method of averaging among all sequences assumes that all codecs produced results for each sequence. When this is not the case, only existing results are taken into account.

3.1.3.1 Normal Preset

Figure 9 through **Error! Reference source not found.** show results for the Normal preset. The results differ depending on the chosen metric.

Y-SSIM: The three best codecs (no codec performs faster with higher quality) in terms of speed and quality are XviD, Elecard and x264 on average, except for the "Ice Age" and "Indiana Jones" sequences. The x264 encoder is better on average than DivX H.264 and MainConcept.

Almost all encoders except Microsoft Expression exhibited better results on average than did Theora.

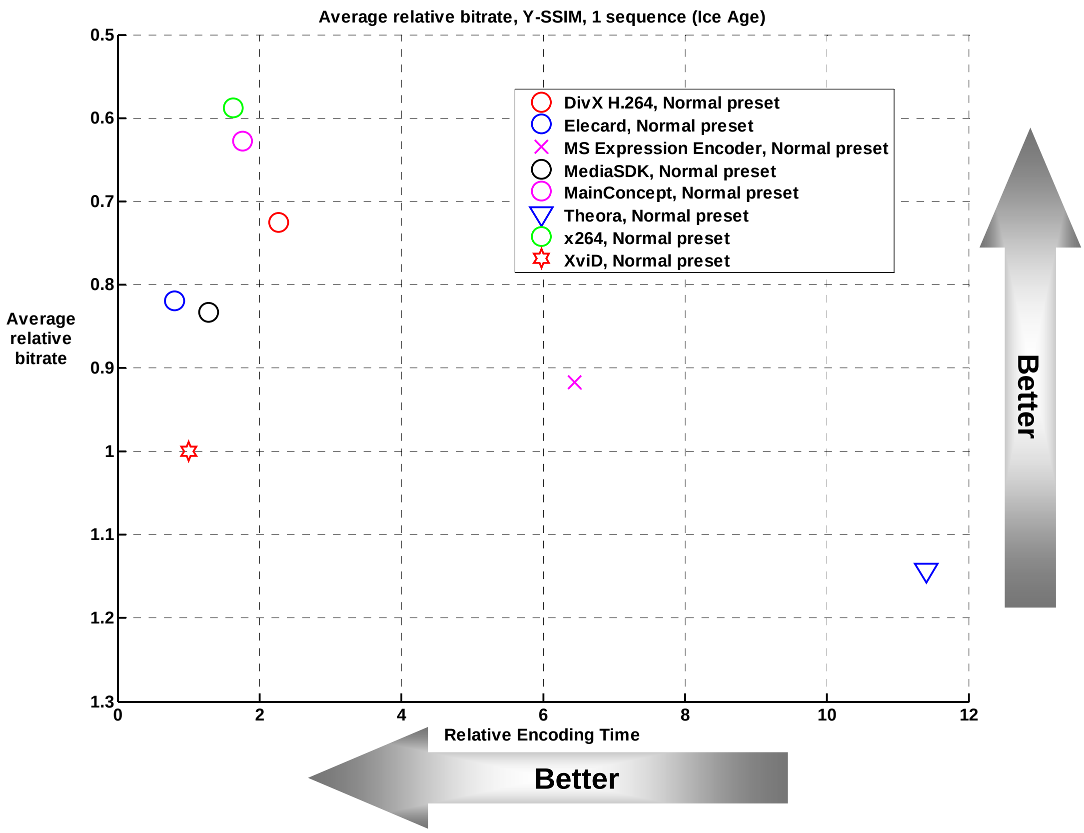


Figure 9. Speed/quality trade-off—usage area "Movies," "Ice Age" sequence, Normal preset, Y-SSIM metric

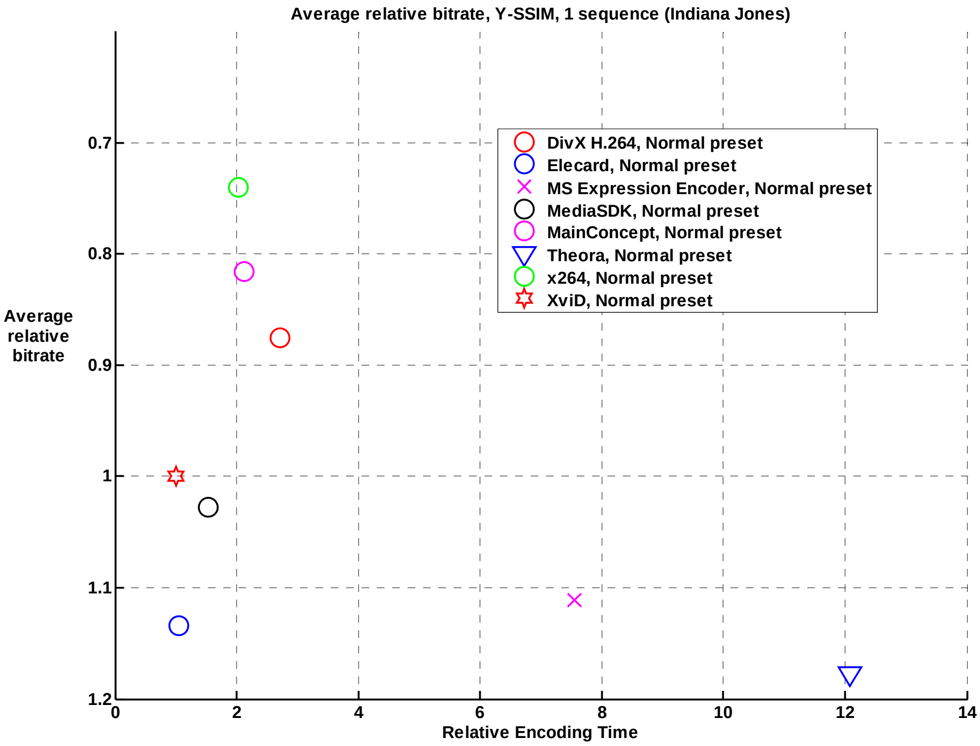


Figure 10. Speed/quality trade-off—usage area "Movies," "Indiana Jones" sequence, Normal preset, Y-SSIM metric

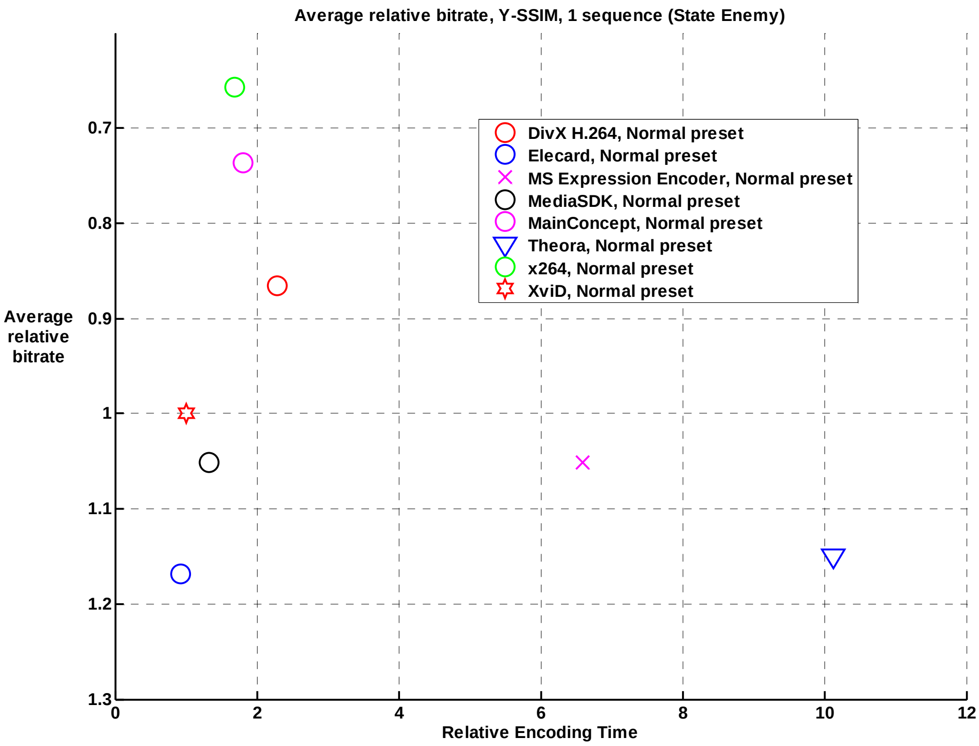


Figure 11. Speed/quality trade-off—usage area "Movies," "State Enemy" sequence, Normal preset, Y-SSIM metric

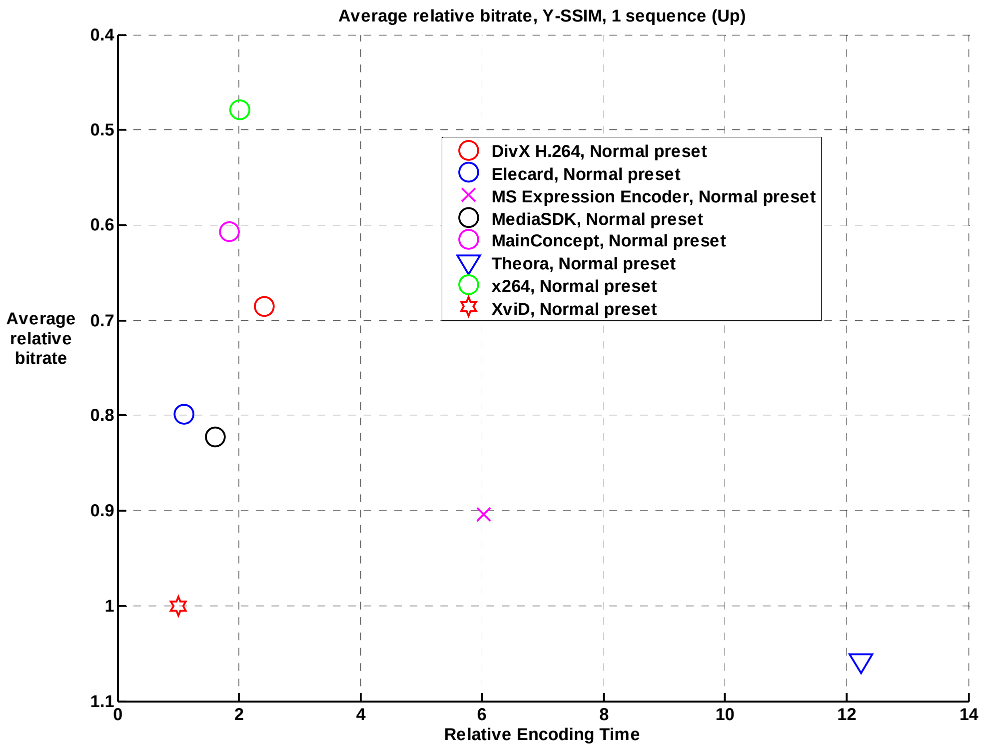


Figure 12. Speed/quality trade-off—usage area "Movies," "Up" sequence, Normal preset, Y-SSIM metric

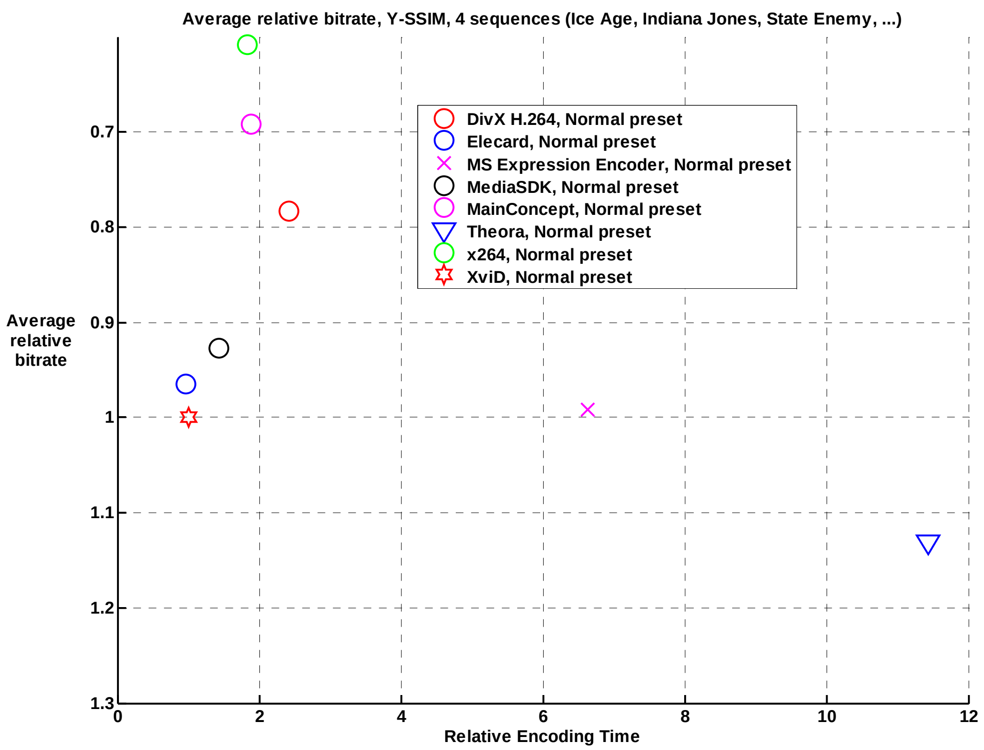


Figure 13. Speed/quality trade-off—usage area "Movies," All "Movie" sequences, Normal preset, Y-SSIM metric

3.1.4 Bitrate Handling

3.1.4.1 Normal Preset

Encoders with Normal presets, except the Microsoft Expression encoder, demonstrate good bitrate handling for all sequences. For the "Up" sequence, XviD showed less than stellar results, especially at high bitrates. For all sequences, MainConcept, x264, Elecard and DivX H.264 showed the best results compared with other encoders. DivX H.264's bitrate decreases at 1,500 kbps. MediaSDK slightly decreases bitrates, but the results are stable. Theora has a good bitrate handling mechanism but is not very stable.

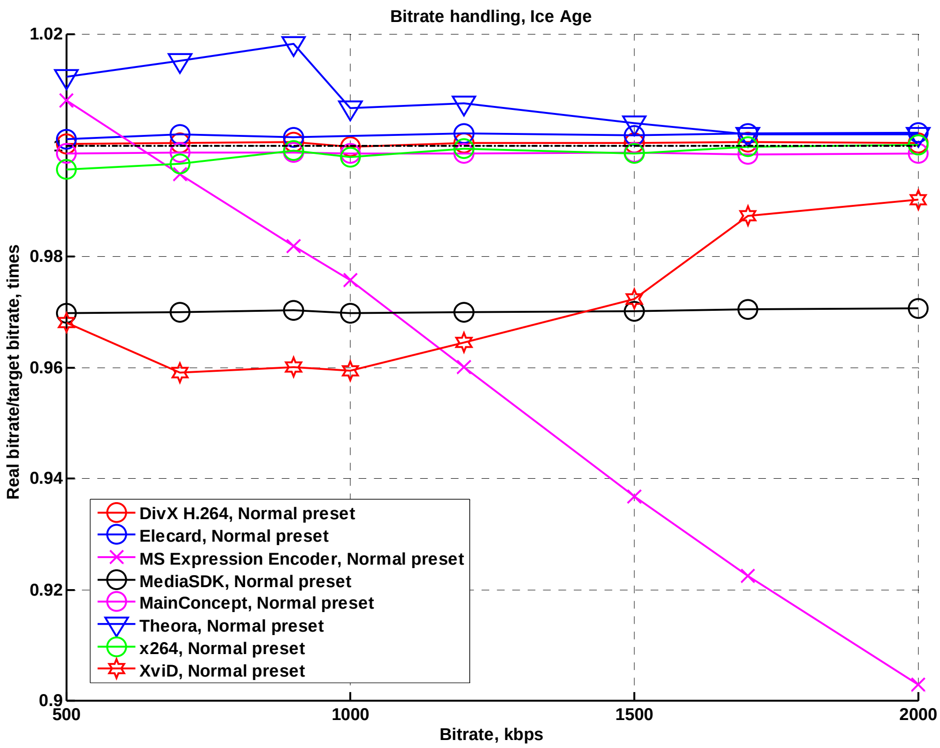


Figure 14. Bitrate handling—usage area "Movies," "Ice Age" sequence, Normal preset

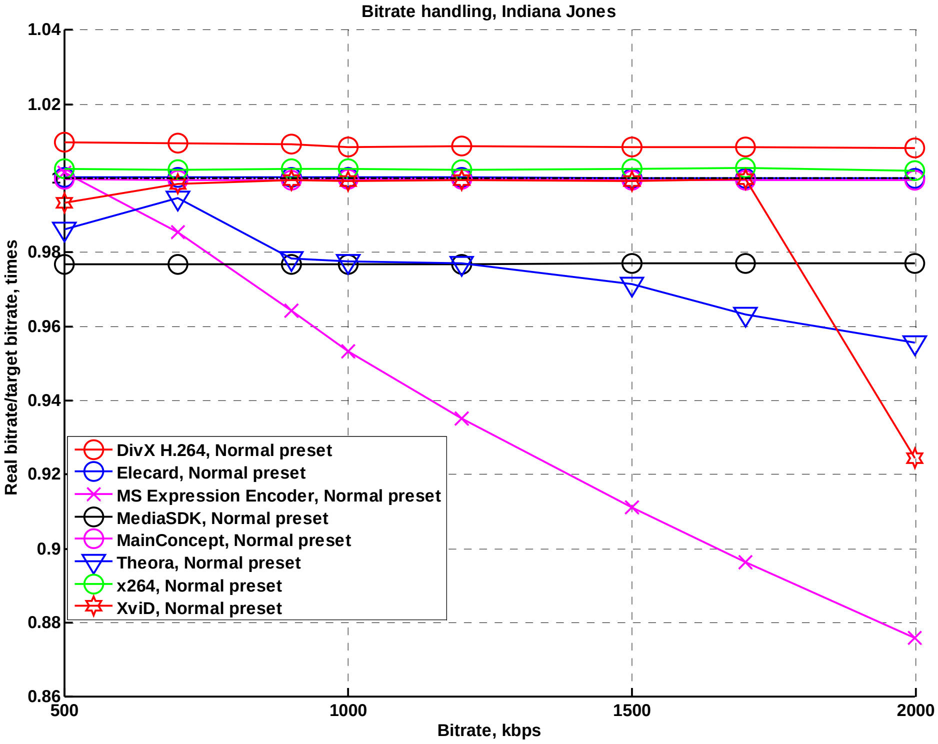


Figure 15. Bitrate handling—usage area "Movies," "Indiana Jones" sequence, Normal preset

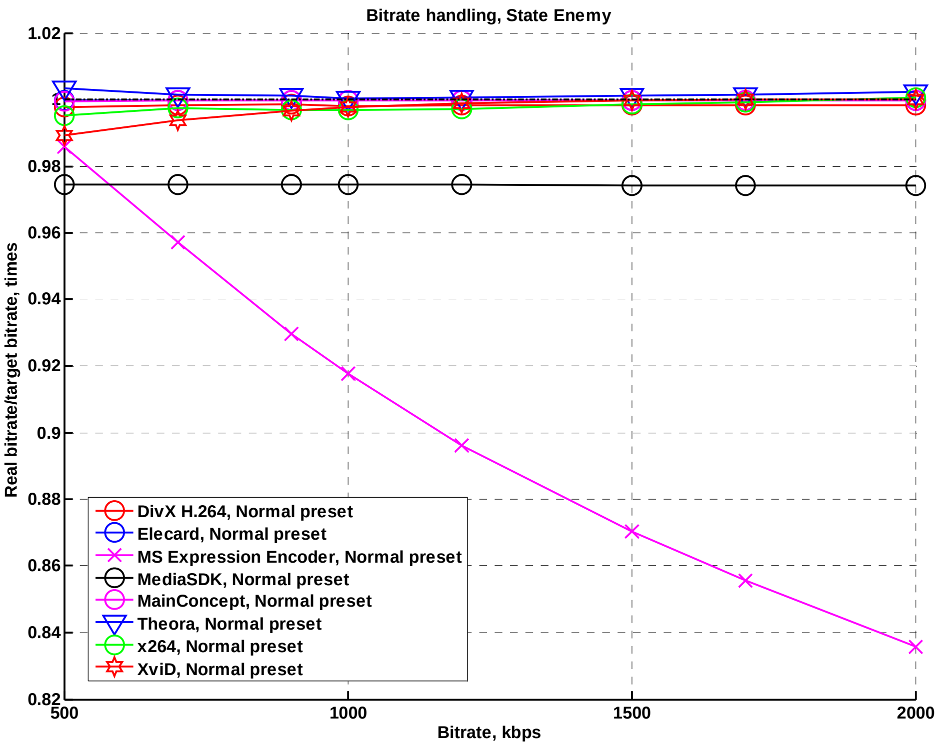


Figure 16. Bitrate handling—usage area "Movies," "State Enemy" sequence, Normal preset

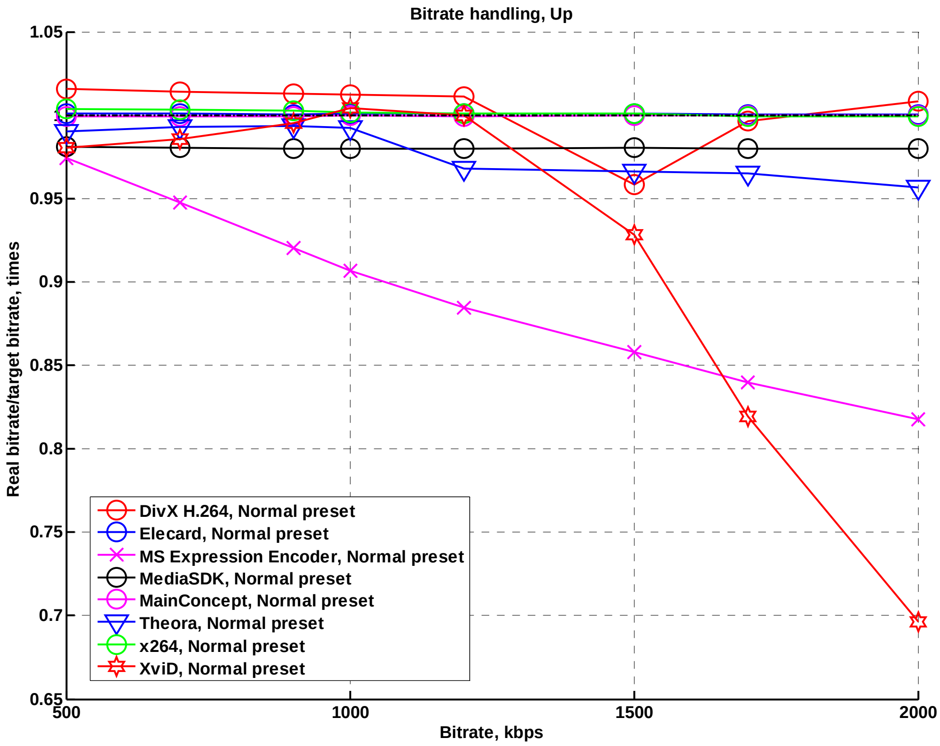


Figure 17. Bitrate handling—usage area "Movies," "Up" sequence, Normal preset

3.1.5 Relative Quality Analysis

Error! Reference source not found. through **Error! Reference source not found.** show relative bitrates for a fixed quality output for all codecs and presets. Note that these tables do not include information about the speed of the encoder.

Note that each number in the tables below corresponds to some range of bitrates (see Appendix 7. Figures Explanation for more details). 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. This comparison technique will be improved in the future.

Table 3 and **Error! Reference source not found.** present the Normal preset results for the Y-SSIM and Y-PSNR quality metrics, respectively. The results are similar to those of the High Speed preset: the leaders are x264 and MainConcept, depending on the quality metric (the bitrate difference is 18% for a fixed quality).

Table 3. Average bitrate ratio for the same quality. Usage area "Movie". "Normal" preset, Y-SSIM.

	DivX H.264	Elecard	Expression	MediaSDK	MainConcept	Theora	x264	XviD
DivX H.264	100%	124%	127%	118%	87%	138%	76%	128%
Elecard	81%	100%	103%	95%	70%	115%	61%	104%
Expression	79%	97%	100%	93%	69%	114%	61%	101%
Media SDK	84%	105%	107%	100%	74%	120%	64%	108%
MainConcept	115%	142%	144%	136%	100%	157%	88%	145%
Theora	72%	87%	88%	84%	64%	100%	57%	88%
x264	131%	163%	164%	155%	114%	176%	100%	164%
XviD	78%	96%	99%	93%	69%	113%	61%	100%

Error! Reference source not found. *depicts the data from the tables above. Each line in the figures 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.*

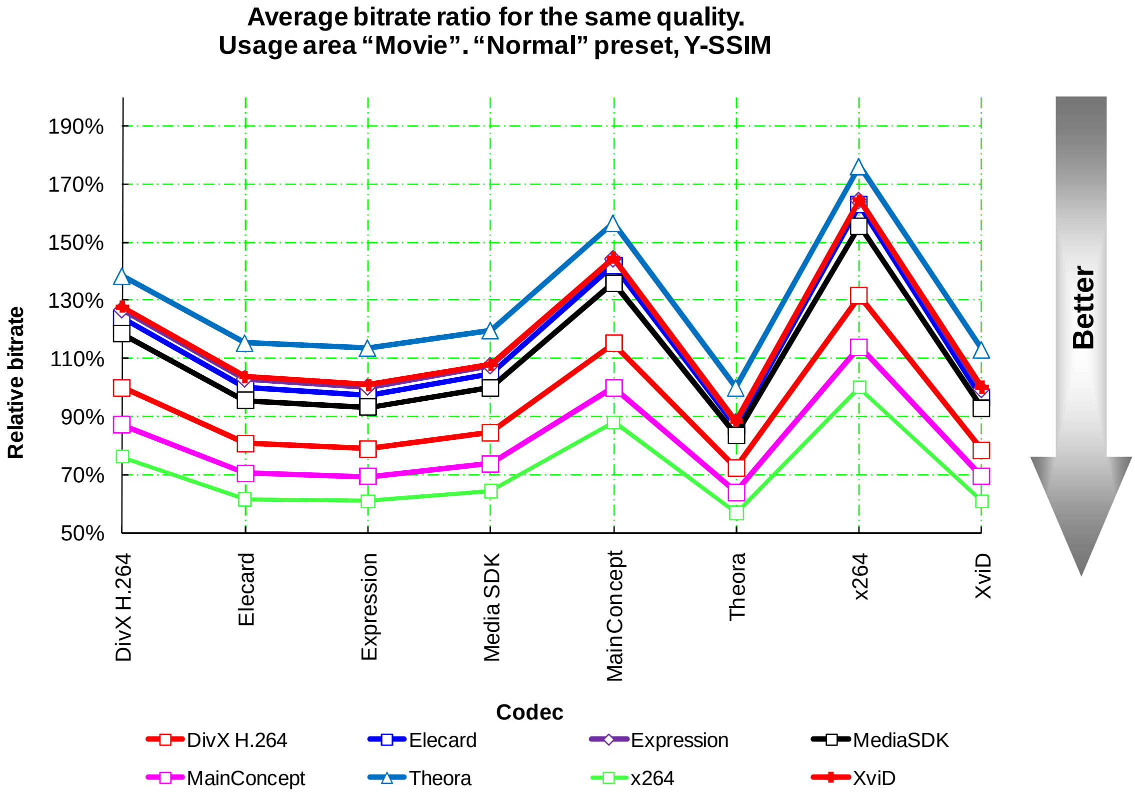


Figure 18. Average bitrate ratio for a fixed quality—usage area "Movies". Normal preset, Y-SSIM metric.

3.2 HDTV

The results for HDTV could be found in professional versions of this report:

- *H.264 Comparison Report Pro 2010 – HDTV edition*
- *H.264 Comparison Report Pro 2010 – Enterprise edition*

See http://compression.ru/video/codec_comparison/h264_2010 webpage for purchase links.

3.3 Conclusions

3.3.1 Movies

The leading encoders in this usage area are MainConcept and x264. The quality of the Theora encoder is rather low.

3.3.1.1 High Speed Preset

The x264 encoder demonstrates better quality on average, and MainConcept shows slightly lower quality. These codecs' bitrate handling algorithm is acceptable for this usage area. The MediaSDK codec places third. The top three codecs for this preset are the following:

1. x264
2. MainConcept
3. MediaSDK

3.3.1.2 Normal Preset

The results for the Normal preset differ from those for the High Speed presets only in third place. The x264 encoder demonstrates better quality on average, and MainConcept shows slightly lower quality. The DivX H.264 encoder holds third place. The top three codecs for this preset are the following:

1. x264
2. MainConcept
3. DivX H.264

3.3.1.3 High Quality Preset

The results for this preset are similar to those of the Normal preset. The leaders are the x264 and MainConcept codecs. DivX H.264 takes third place. The top three codecs for this preset are the following:

1. x264
2. MainConcept
3. DivX H.264

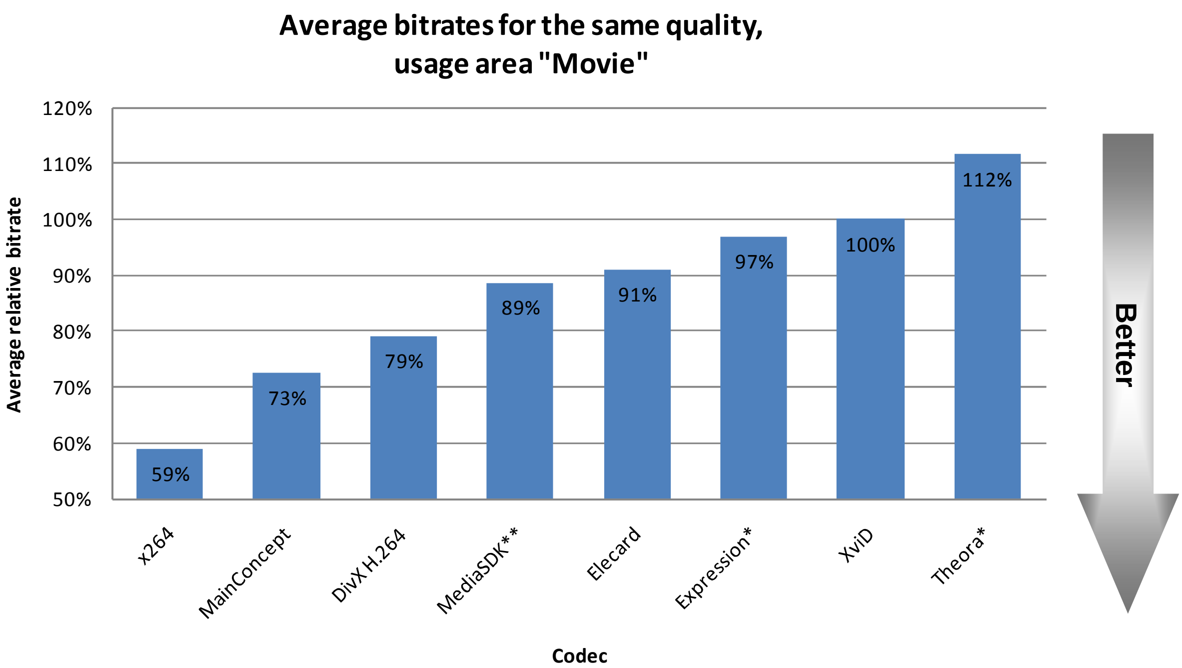


Figure 19. Average bitrate ratio for a fixed quality—usage area "Movies," all presets, Y-SSIM.

**The encoding speed of these encoders is not in the requirements range owing to our chosen presets or to a lack of options*

***The MediaSDK encoder does not meet the requirements for High Speed presets–specifically, no B-frames and exactly one reference frame*

3.3.2 HDTV

The leaders in the HDTV area are x264, DivX H.264 and MainConcept. The XviD encoder trails all other H.264 encoders. Theora demonstrates rather poor results compared with other encoders.

3.3.2.1 High Speed Preset

The x264 encoder demonstrates better quality on average, and MainConcept shows slightly lower quality. The MainConcept codec holds third place. The top three codecs for this preset are the following:

1. x264
2. DivX H.264 (High Speed preset does not meet speed requirements)
3. MainConcept

3.3.2.2 Normal Preset

The results for the Normal preset differ from those of the High Speed presets. MainConcept shows the best results; DivX H.264 and x264 share second place (their quality results are very similar, with x264 being slightly better and DivX H.264 being 15% faster), and MainConcept holds third. The top four codecs for this preset are the following:

1. MainConcept
2. x264 and DivX H.264
3. MainConcept

3.3.2.3 High Quality Preset

The results for the High Quality preset are very interesting: x264 held first place, MainConcept was in second place, and two codecs (DivX H.264 and Elecard) shared third place (their quality results are very close, but DivX H.264 is 1.2 times faster than Elecard). The top four codecs for this preset are the following:

1. x264
2. MainConcept
3. DivX H.264 and Elecard

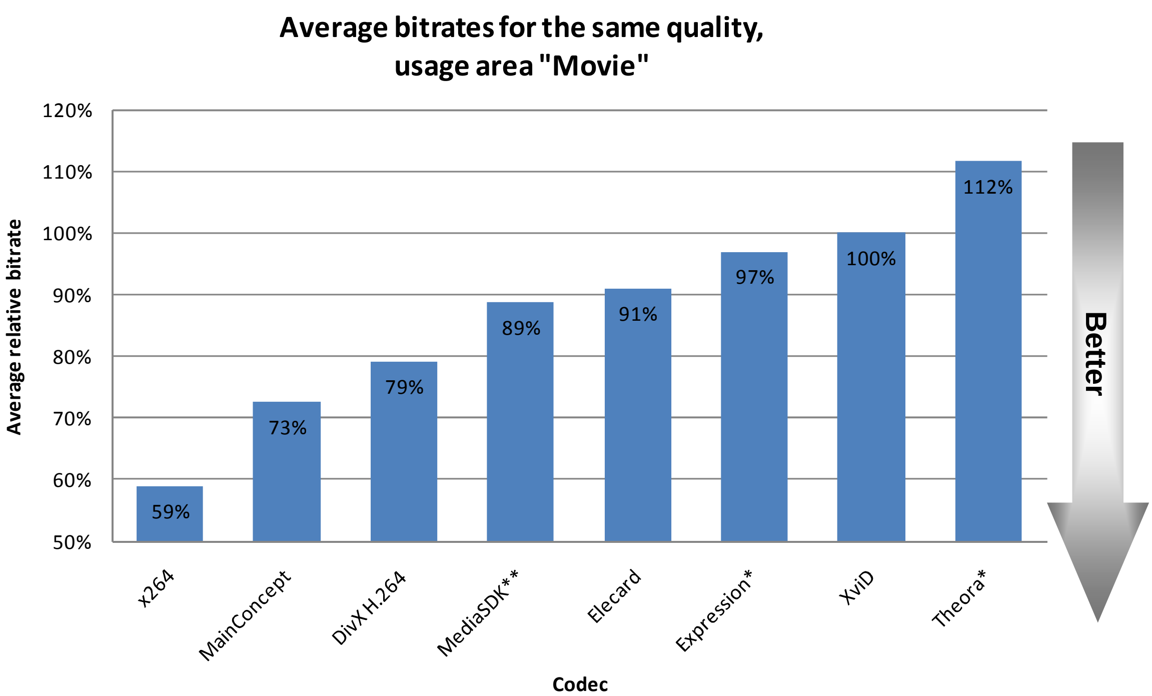


Figure 20. Average bitrate ratio for a fixed quality—usage area “HDTV,” all presets, Y-SSIM.

**The encoding speed of this encoder is not in the requirements range owing to our chosen presets or to a lack of options*

***The DivX H.264 High Speed preset for HDTV does not meet the speed requirements*

3.3.3 Overall Conclusions

Overall, the leader in this comparison is x264, followed by MainConcept and DivX H.264. The Theora encoder demonstrates the worst results among all codecs tested.

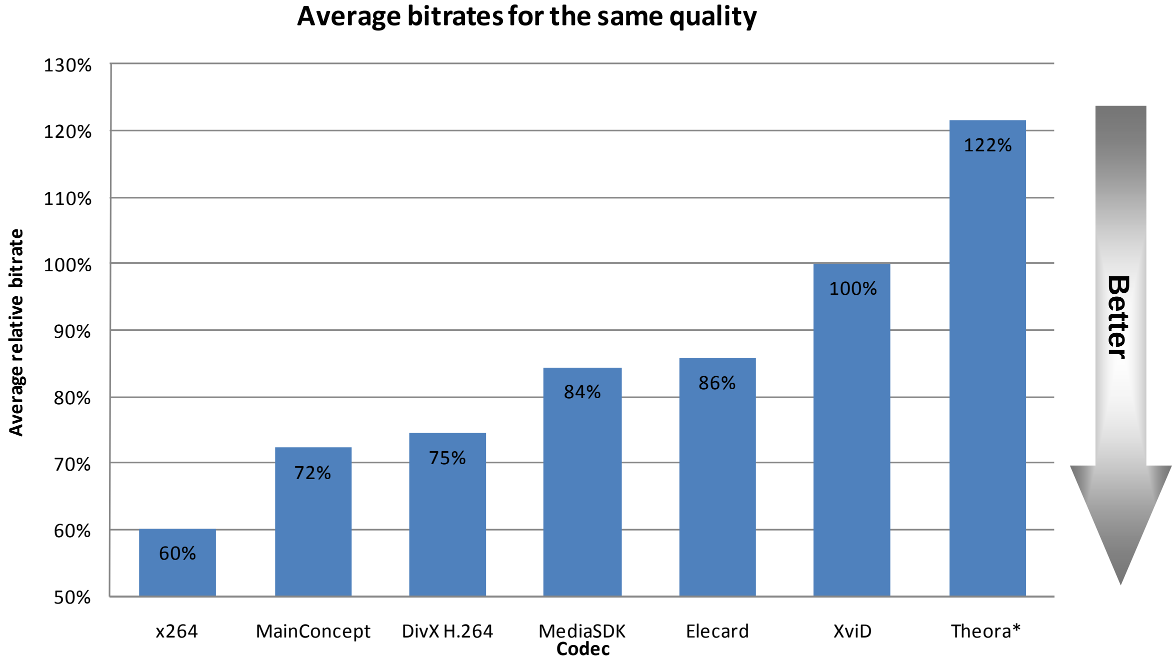


Figure 21. Average bitrate ratio for a fixed quality for all categories and all presets (Y-SSIM).

**The encoding speed of this codec does not fall in the required range owing to our chosen presets or to a lack of options.*

The overall ranking of the codecs tested in this comparison is as follows:

1. x264
2. MainConcept
3. DivX H.264
4. MediaSDK
5. Elecard
6. XviD
7. Theora

Microsoft Expression Encoder 3 could not be placed in this list because of its much longer encoding time compared with other encoders (except Theora).

Using the standard Theora interface, we could not find any multithreading options; Theora works only in single-thread mode.

The leader in this comparison is x264—its quality difference (according to the SSIM metric) could be explained by the special encoding option ("tune-SSIM"). Interestingly, using the PSNR metric for MainConcept yielded results comparable with or better than those of x264. This means that no encoder can achieve the best results for both SSIM and PSNR when using the same parameters.

The difference between the MainConcept and DivX H.264 encoders is not overly significant, so these encoders tied for second in this comparison. The developers of the Elecard encoder do not provide a High Speed preset, so its ranking is based solely on the results for the Normal and High Quality presets.

This rank is based only on the encoders' quality results (see Figure 21). Encoding speed is not considered here.

3.3.4 Codec Conclusions

- **DivX H.264—**quite balanced encoder with not very big number of parameters, this fact could be comfortable for users. This encoder is designed as a free sample application for DivX Plus HD compliant video encoding, and is a feature-constrained, for-purpose application.
- **Elecard—**very fast codec with good encoding quality and very flexible settings. Many adjustable encoding settings are provided. This encoder has a very good bitrate handling mechanism (especially for the “Movies” usage area).
- **Microsoft Expression Encoder—**the low encoding speed could result from the GUI loading every time a video is encoded. Unfortunately, we could not automate the encoder to eliminate this GUI loading.
- **MediaSDK—**balanced encoder with a limited number of parameters but several use cases (speed/quality presets), making the encoder very comfortable for users.
- **MainConcept—**one of the best codecs by encoding quality; has many encoding settings that can be adjusted**.** This encoder has a very good bitrate handling mechanism.
- **Theora—**not an H.264 codec. Using our testing methodology (objective metrics) and test set, this encoder yields lower quality than do H.264 codecs and even MPEG-4 ASP.
- **x264—**one of the best codecs by encoding quality; has very user-friendly predefined presets, as well as many adjustable encoding settings.
- **XviD—**an MPEG-4 ASP codec; its quality could be very close to or even higher than that of some commercial H.264 standard implementations, especially for encoding “Movie” sequences, but not for “HDTV” sequences.

3.3.5 Comments from Developers

3.3.5.1 x264

This comparison had very different restrictions as compared to previous years. In particular, the speed requirements were much more restrictive, which seems to have hurt many competing encoders more than it did x264. In addition to the effects of these rule changes, a variety of improvements on x264's side likely contributed to its relatively high 14.8% margin of victory as compared to last year's test.

Macroblock-tree ratecontrol is a new feature (added August 2009) which provided very significant compression improvements, both when measured via SSIM and PSNR. This is likely the reason for some of the very large gaps between x264 and other encoders in certain tests (e.g. Up and Amazon). Weighted P-frame prediction also slightly improved compression in the higher quality modes. A new adaptive quantization mode, which was used in the MSU test, significantly improves SSIM. The many performance improvements since the last test also helped a great deal considering the more restrictive speed requirements.

One criticism we have is that a lot of the tests were on very grainy – often even already-compressed sources, such as Amazon. We think that better results would be achieved by sticking to relatively clean, uncompressed sources. The primary reason for this is that noise tends to reduce the gap between encoders, making judgments more difficult. Measuring grain retention is not really possible with SSIM and PSNR to begin with, so using grainy sources is not very useful.

3.3.5.2 Xvid

Xvid's rate-control is basically targeted towards encoding longer sequences (at least a couple of minutes), so rate-control inaccuracies could in part be attributed to short test sequences. In real world scenarios, Xvid's two-pass rate-control has proven to be very accurate. When comparing MPEG-4 ASP and H.264 encoders it should be noted that quantizer scales are different: MPEG-4 ASP can employ quantization parameters ranging from 1 to 31 while in contrast H.264 spans from 0 to 51 and has a logarithmic scale. Because of this, a much wider bitrate range can be covered by H.264 than by MPEG-4 ASP encoders. That's shown also by the report's results: Xvid's rate-control works quite well at mid-bitrate ranges but then over-/undershoots at the very low-/high bitrate test points. This is not a rate-control problem in Xvid but rather caused by the narrower quantizer range of MPEG-4 ASP that does not permit to cover the same wide bitrate range than H.264.

Test results may indicate that the (pre-)compression format of the input sequence has major influence on (re-)compression results: As soon as a video was once lossy compressed, there will be compression artifacts present even if invisible to the human eye. So when (re-)compressing such material, those codecs that can best "imitate" the compression artifacts already present in the input will have an advantage. E.g. for input that was precompressed with H.264 or VC-1, H.264 encoders have an inherent advantage over MPEG-4 ASP at "imitating" the artifacts in the input (because of 4x4 transform). Other than for the Movie test case, several of the HDTV test sequences are H.264/VC-1 precompressed, which could explain the relative worse performance of Xvid for HDTV compared to the Movie use-case. But in contrast, Xvid consistently shows very competitive results at both SD and HD resolutions when the test input is natural video precompressed in MPEG-2 format.

4 Appendix 1. Subjective Comparison and Psycho-Visual Enhancements

4.1 Brief Description

In this work, we analyzed video codecs not only with help of objective metrics, but also using subjective comparisons using SAMVIQ (Subjective Assessment Method for Video Quality evaluation) methodology. This new method was created by the EBU (European Broadcasting Union).

Another task in this part of comparison was to analyze the psycho-visual enhancement quality during encoding. We tested five video codecs, and one codec was tested with two almost identical presets: one without psycho-visual enhancement and another with psycho-visual enhancement.

Methodology	SAMVIQ
Number of experts	42
Number of sequences	5
Number of codecs (presets)	6

4.1.1 SAMVIQ Description

During testing, each expert is able to play any sequence from the test set and give it a mark, and he is able to play a reference video. Marks are in the range of 0 to 100. More information about the methodology can be found in the publication *SAMVIQ: A new EBU methodology for video quality evaluations in in multimedia (*Kozamernik, F., Steinmann, V., Sunna, P. and Wyckens, E., *SMPTE journal,* 2005 04 April).

4.1.2 Subjective Assessment Description

Forty-two experts participated in this subjective assessment. The experts represent a variety of individuals: males and females of ages 18 to 40 who are PC users or video and graphics experts. Each user watched five video groups, with eight videos in each group. Each video group contains six encoded videos and one reference video (the viewers were unaware of which type they were viewing), as well as a reference video specified as such. Users gave each video a mark of 0 to 100, and the marks for all users and sequences were averaged. Some extreme results were discarded before obtaining the final result.

4.2 Video Sequences

Five different video sequences were used in this comparison.

4.2.1 "Battle"

Sequence title	Battle
Resolution	1280x544
Number of frames	586
Color space	RGB24
Frames per second	25

Figure 22. Battle sequence, frame 192

This sequence is a fragment from the beginning of the movie *Terminator 2.* The compression of this sequence is the most difficult among all of the sequences in the analysis. This difficulty is mainly due to three reasons: continual brightness variation (resulting from explosions and laser flashes as seen in the picture above), very fast motion and frequent scene changes. These characteristics often cause codecs to compress frames as I-frames.

4.2.2 "Football"

Sequence title	Football
Resolution	1280x720
Number of frames	564
Color space	RGB24
Frames per second	25

Figure 23. Football sequence, frame 400

This sequence is a part of sport translation. The sequence has a high contrast level, strong motion and rich colors.

4.2.3 "Italy"

Sequence title	Italy
Resolution	1280x720
Number of frames	493
Color space	RGB24
Frames per second	25

Figure 24. Italy sequence, frame 368

This sequence has many scenes with fading transitions between them. This fading can cause encoding difficulties. The sequence contains many sharp details.

4.2.4 "Quadbike"

Sequence title	Quadbike
Resolution	1280x720
Number of frames	562
Color space	RGB24
Frames per second	25

Figure 25. Quadbike sequence, frame 191

This sequence has slow motion with an almost static camera and few scene changes.

4.2.5 “Simpsons”

Sequence title	Simpsons
Resolution	1280x528
Number of frames	514
Color space	RGB24
Frames per second	25

Figure 26. Simpsons sequence, frame 310

This sequence is a part of the *Simpsons* animated movie; it has high contrast and different types of motion.

4.3 Video Codecs

Five codecs were used in the comparison:

- DivX MPEG-4 ASP
- DivX H.264

- Elecard
- MainConcept
- x264 (two presets: with psycho-visual enhancement and without)

4.4 Results

The following graphs show results for the subjective comparison. Reference bars indicate the visual quality of the uncompressed original sequence as estimated by experts. Other bars indicate the quality of the encoded sequences.

4.4.1 "Battle" Sequence

For the "Battle" sequence, x264 shows the highest quality. x264 with psycho-visual enhancement shows very similar (almost undistinguishable) results. The codecs can be rated by visual quality as follows:

1. x264
2. x264 with psycho-visual enhancement
3. MainConcept
4. DivX H.264
5. Elecard
6. DivX ASP

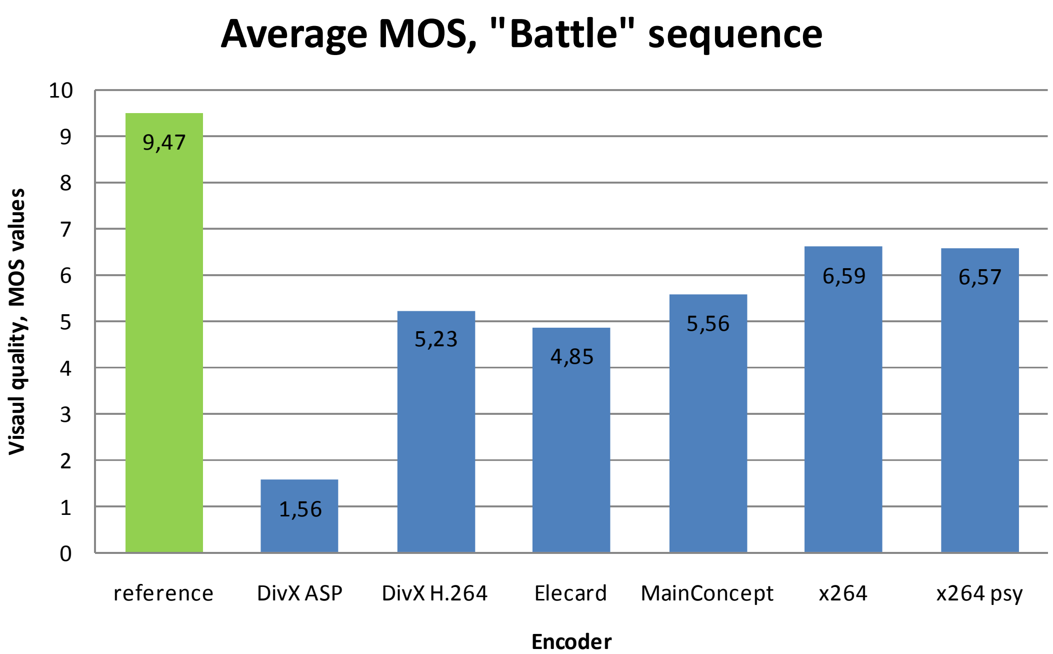


Figure 27. Mean opinion score (MOS), "Battle" sequence

The full results for Subjective Comparison could be found in professional version of this report – H.264 Comparison Report Pro 2010 – Enterprise edition

See http://compression.ru/video/codec_comparison/h264_2010 webpage for purchase links.

4.5 Conclusion

Analysis of all tested sequences yields the following codec rankings:

1. x264
2. x264 with psycho-visual enhancement
3. DivX H.264
4. MainConcept
5. Elecard
6. DivX ASP[1]

A crucial conclusion that can be drawn from the subjective comparison is that psycho-visual enhancement yields poorer results on average than does the unenhanced codec. DivX ASP is a for-purpose encoder specializing in interoperability with DivX certified devices. It is not an AVC/H.264 encoder.

5 Appendix 2. x264 Comparison Over Time

The quality of an H.264 codec, over several years, can be compared for a given video sequence. The x264 encoder was chosen for this task because it is present in almost every MSU VIDEO MPEG-4 AVC/H.264 codec comparison, and it produces good results compared with other encoders. Figure 28 through Figure 31 show the position of the x264 codec compared with other codecs for the “Battle” sequence. For all years except 2005, x264 shows the best results. For years 2006–2009, we have shown results using Y-SSIM as the quality metric; for 2005, we did not use this as the main metric. In light of these results, x264 could be a good reference encoder for analyzing the overall progress of H.264 encoders over time.

[1]

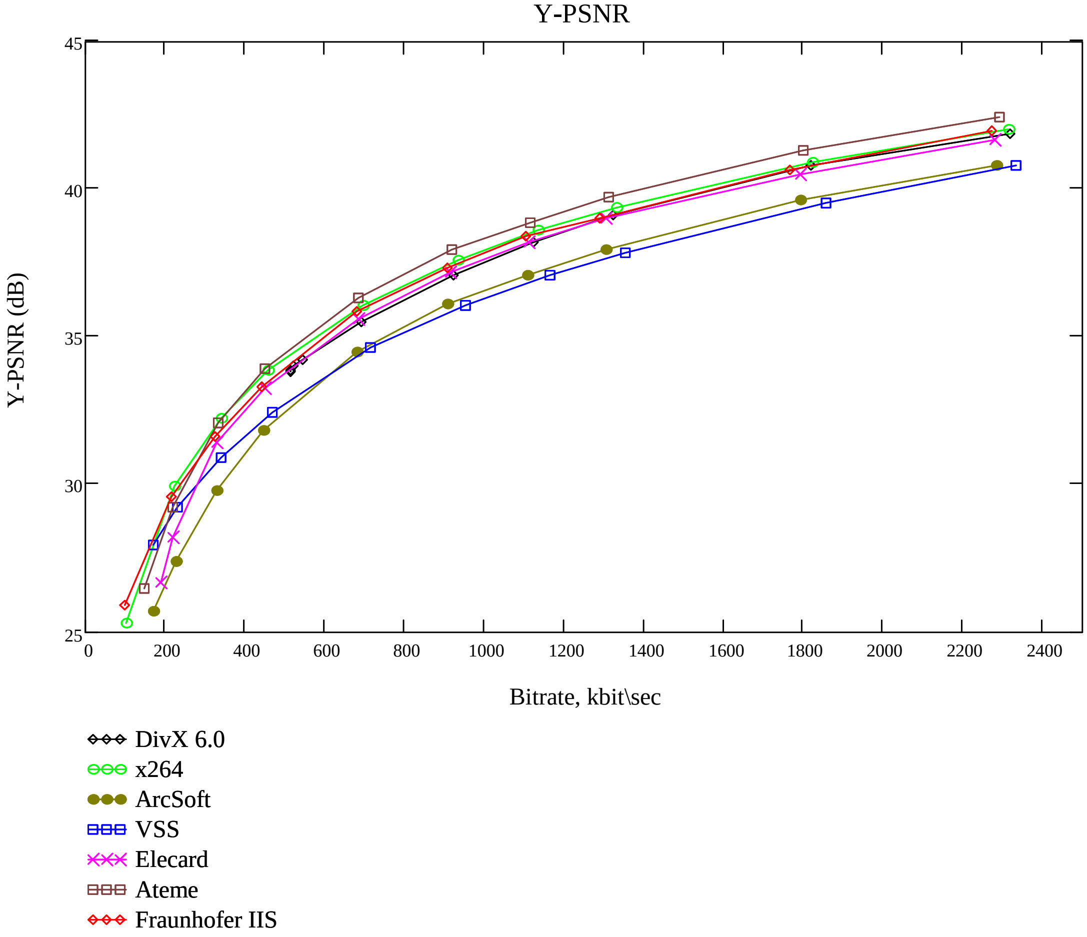


Figure 28. Bitrate/quality for 2005—usage area "Movies," "Battle" sequence, High Quality preset, Y-PSNR metric

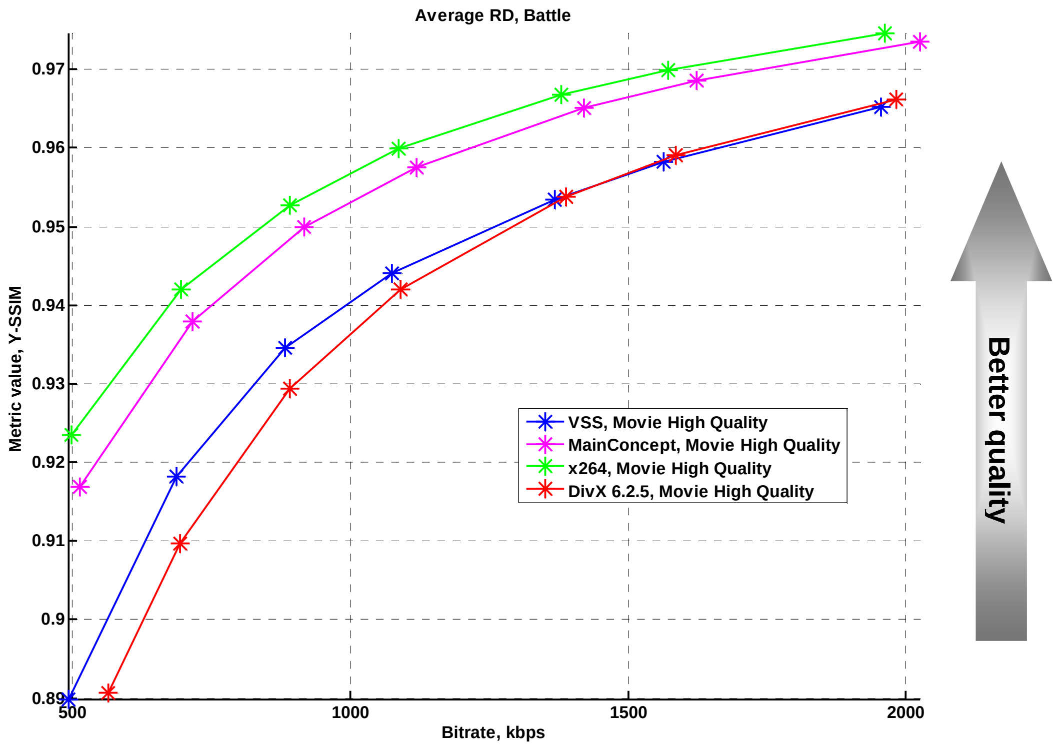


Figure 29. **Bitrate/quality for 2006—usage area "Movies," "Battle" sequence, High Quality preset, Y-SSIM metric**

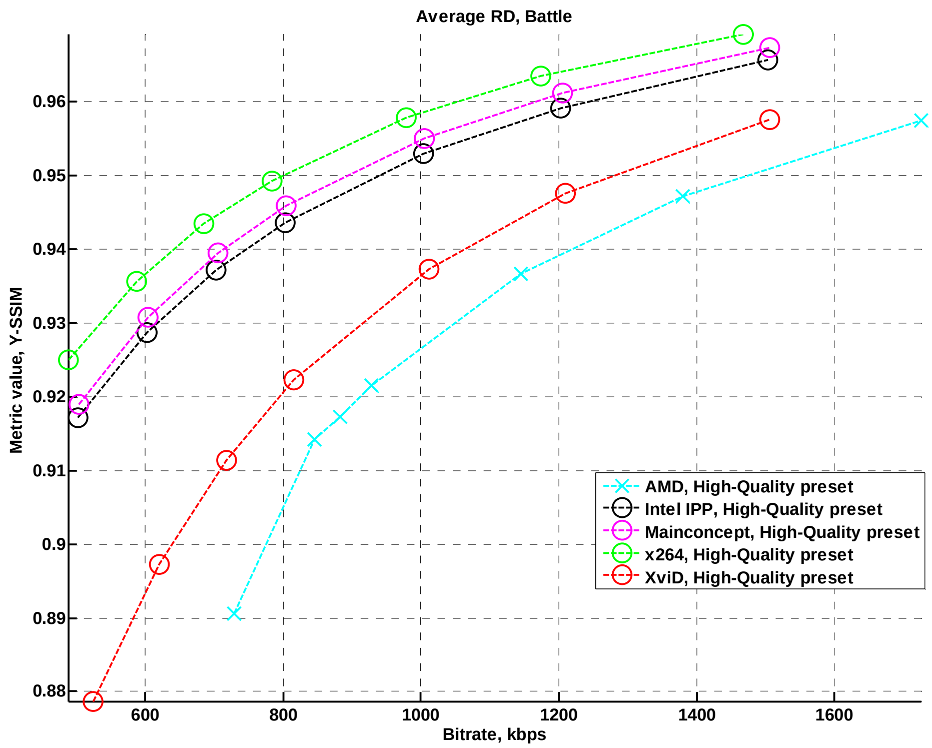


Figure 30. **Bitrate/quality for 2007—usage area "Movies," "Battle" sequence, High Quality preset, Y-SSIM metric**

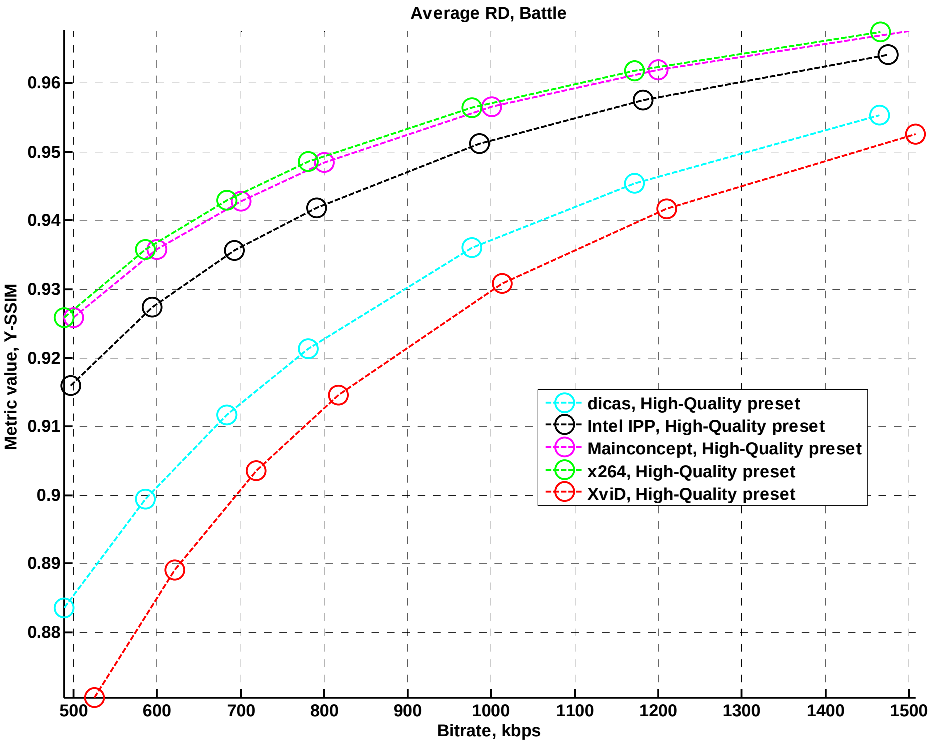


Figure 31. Bitrate/quality for 2009—usage area "Movies," "Battle" sequence, High Quality preset, Y-SSIM metric

Figure 32 shows the RD curve for the "Battle" sequence using x264 encoders from different years. The best encoder is this year's x264; the worst is the 2005 version. Using SSIM, the codecs can be ranked as follows:

1. x264 (2010)
2. x264 (2009)
3. x264 (2007)
4. x264 (2006)
5. x264 (2005)

These results are shown in Figure 33. This figure indicates that the overall progress is very good, and that the x264 encoder has increased in speed and quality over recent years. But the old x264 does not use multithreading, so encoding speed varies considerably.

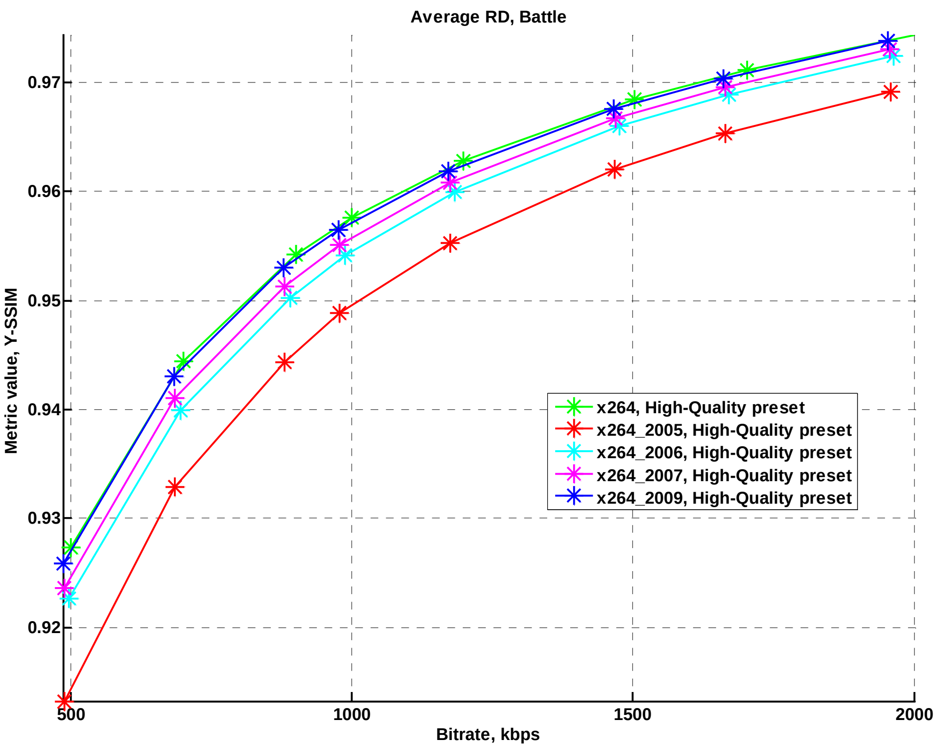


Figure 32. Bitrate/quality for different x264 encoder versions—usage area "Movies," "Battle" sequence, High Quality preset, Y-SSIM metric

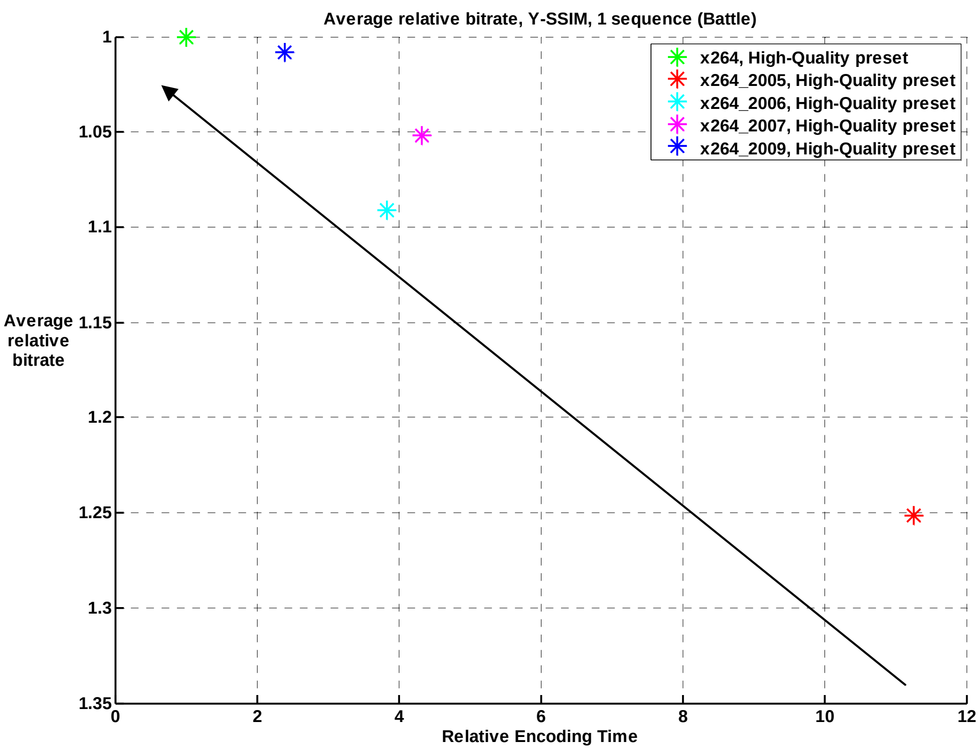


Figure 33. Progress of the x264 encoder over several years—Y-SSIM metric

Interestingly, if Y-PSNR is used as the quality metric, x264 does not exhibit the kind of progress shown in Figure 34 or Figure 35. This difference is because for past years, the x264 developers specified, for use in testing, the presets optimized for the SSIM metric. Also, note that no encoder (among the different x264 versions) produces the best results simultaneously for both SSIM and PSNR.

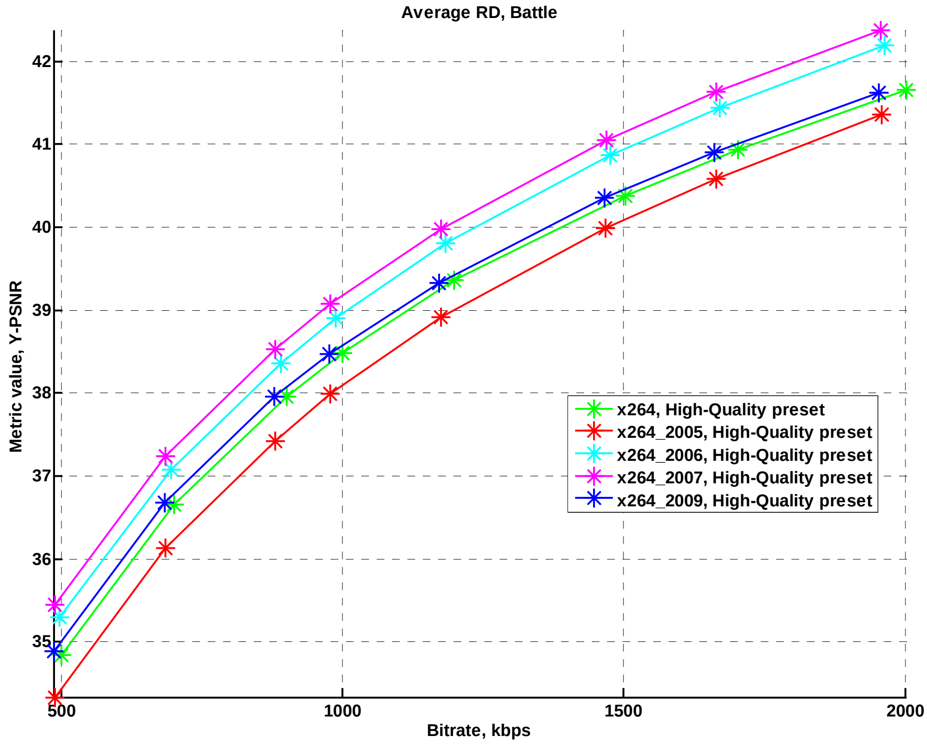


Figure 34. Bitrate/quality for different x264 encoder versions—usage area "Movies," "Battle" sequence, High Quality preset, Y-PSNR metric

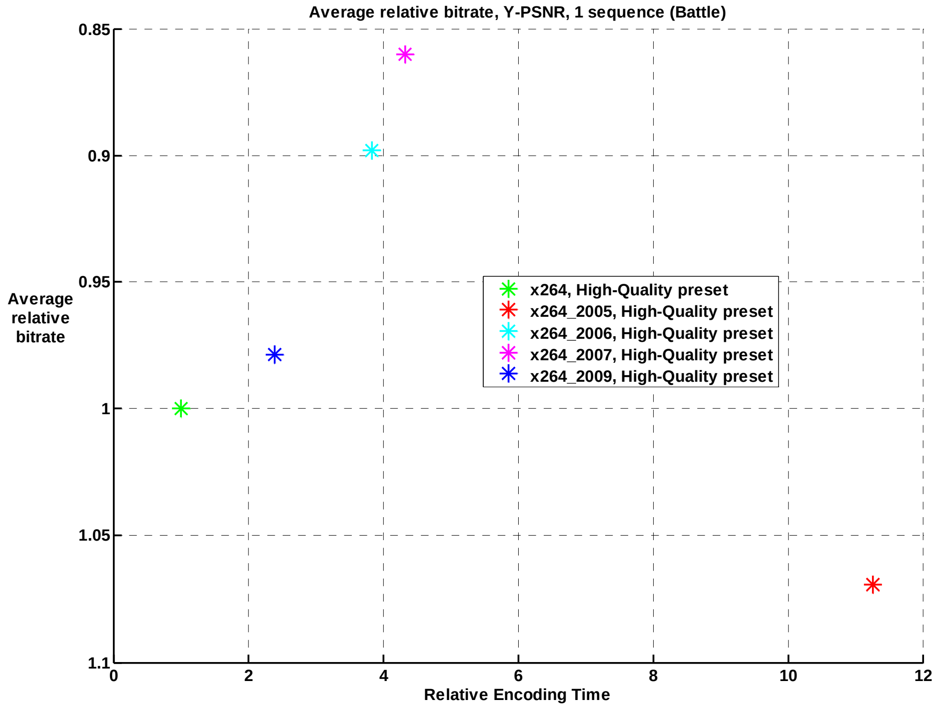


Figure 35. Progress of the x264 encoder over several years—Y-PSNR metric

The bitrate handling mechanism for the x264 encoder is quite good for each version, as Figure 36 indicates. Results for previous x264 versions (0.98 of target bitrate) could be explained by a different interpretation of kbps (1,024 versus 1,000 bits per second).

The per-frame analysis presented in Figure 37 shows that the main encoding mechanism did not changed significantly.

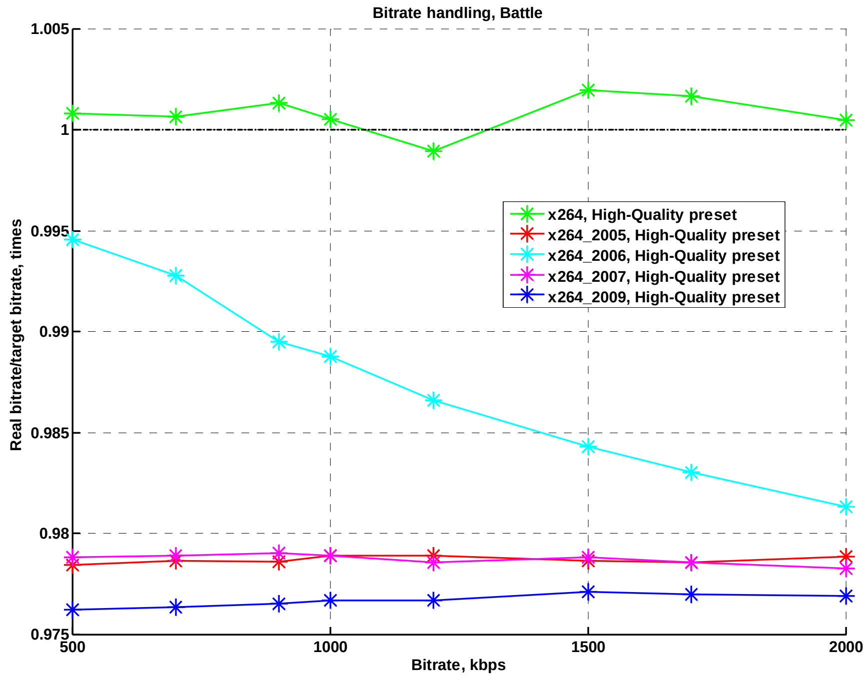


Figure 36. Bitrate handling for different x264 encoder versions

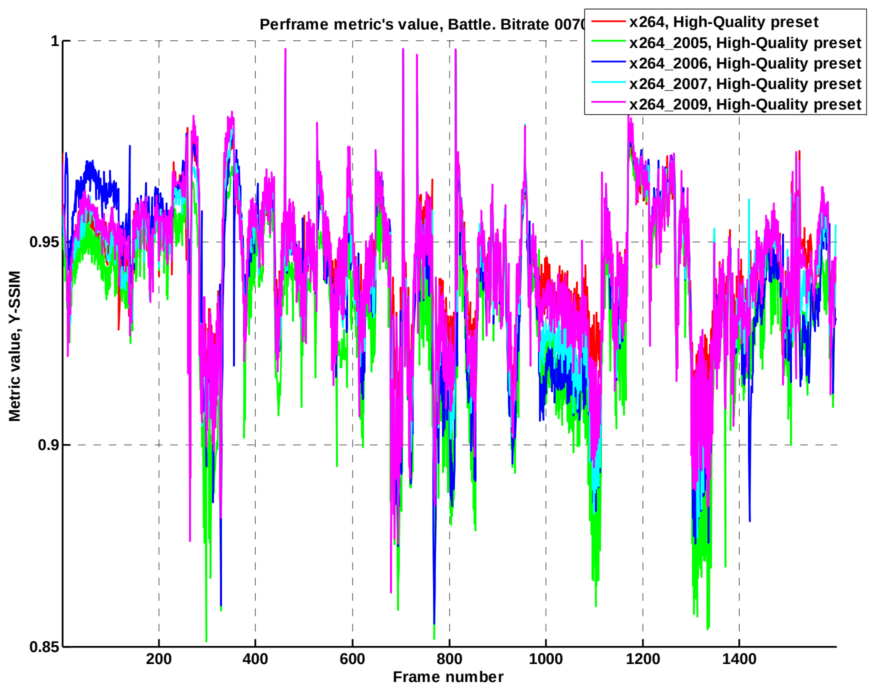


Figure 37. Per-frame analysis for different x264 versions—"Battle" sequence, 700kbps

6 Appendix 3. Video Codec Analyzers

Three codec analyzers were used in this comparison:

- Synthetic Motion analyzer—synthetic video sequences are used to analyze a codec's ME algorithm.
- Tail Area analyzer—synthetic video sequences are used to estimate a codec's quality in occlusion areas near moving objects.
- Spatially Variable Noise analyzer—analyzes a codec's macro-block level rate control by adding various types of noise in each video frame.

The results for Video Codec Analyzers could be found in professional version of this report – H.264 Comparison Report Pro 2010 – Enterprise edition

See http://www.compression.ru/video/codec_comparison/mpeg-4_avc_h264_2010.html webpage for purchase links.

7 Appendix 4. Theora vs. x264 Single-thread Comparison

Multithreaded encoding is not supported by the Theora encoder, but CPUs with four cores are used for the tests. This situation leads to an incorrect speed comparison between Theora and other codecs.

To eliminate this disparity, we have turned on the x264 encoder's single-thread mode and compared the results with those of Theora. The overall speed/quality trade-off for "Movies" and "HDTV" use cases is depicted in Figure 38 and Figure 39.

The main conclusion is that even in single-thread mode, Theora is a little bit slower and produces much poorer quality than does x264.

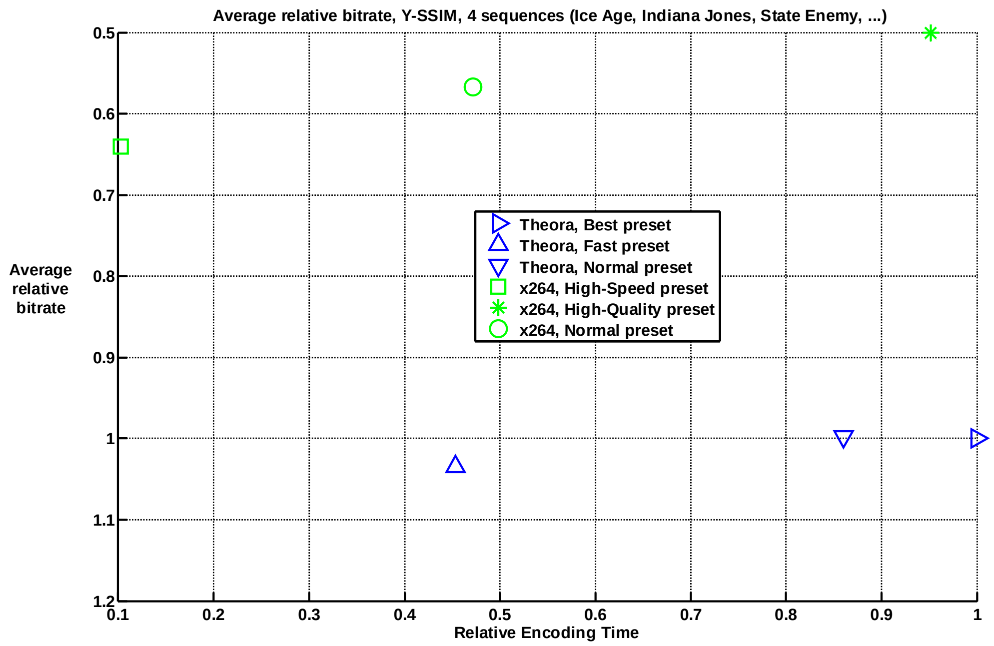


Figure 38. Speed/quality trade-off for x264 and Theora single-thread mode—usage area "Movies," all sequences, Y-SSIM metric

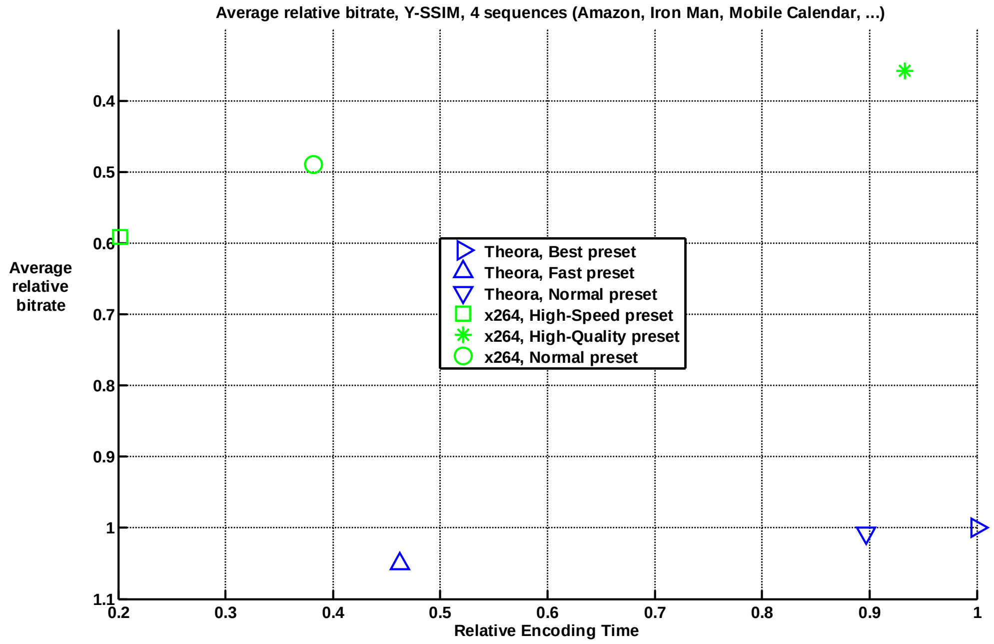


Figure 39. Speed/quality trade-off for x264 and Theora single-thread mode—usage area "HDTV," all sequences, Y-SSIM metric

8 Appendix 5. Test Set of Video Sequences

8.1 Movie Sequences

8.1.1 “Ice Age”

Sequence title	Ice Age
Resolution	720x480
Number of frames	2014
Color space	YV12
Frames per second	24
Source	MPEG-2 (DVD9), 5.7Mbps

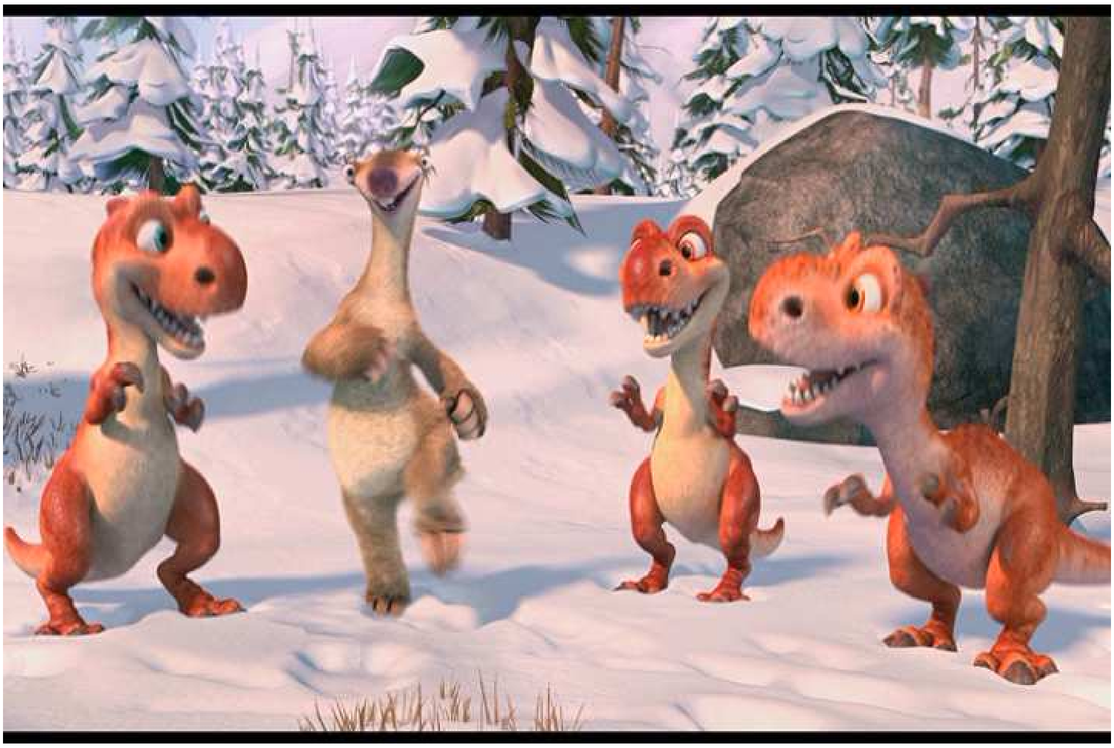


Figure 40. Ice Age sequence, frame 500

This sequence is a fragment from the *Ice Age 3* animated movie. This movie has low-contrast portions and high-contrast portions, and it has many types of motion: camera panning, slow motion and very fast motion. Also, it has a scene with colors that differ completely from those of other scenes. Small black letterboxes appear at the top and bottom of the video.

8.1.2 "Indiana Jones"

Sequence title	Indiana Jones
Resolution	704x288
Number of frames	5000
Color space	YV12
Frames per second	30
Source	MPEG-2 (DVD), FlaskMPEG deinterlace

Figure 41. Indiana Jones sequence, frame 1

This sequence is a fragment from the *Indiana Jones* movie. Compression of this sequence is difficult for two main reasons: the presence of low-contrast scenes and the high level of motion in different scenes. Also, several scenes have very different types of motion, ranging from almost static scenes with talking people to scenes with strong motion (for example, the scene where stones fall).

8.1.3 "State Enemy"

Sequence title	State Enemy
Resolution	720x304
Number of frames	6500
Color space	YV12
Frames per second	24
Source	MPEG-2 (DVD), FlaskMPEG deinterlace

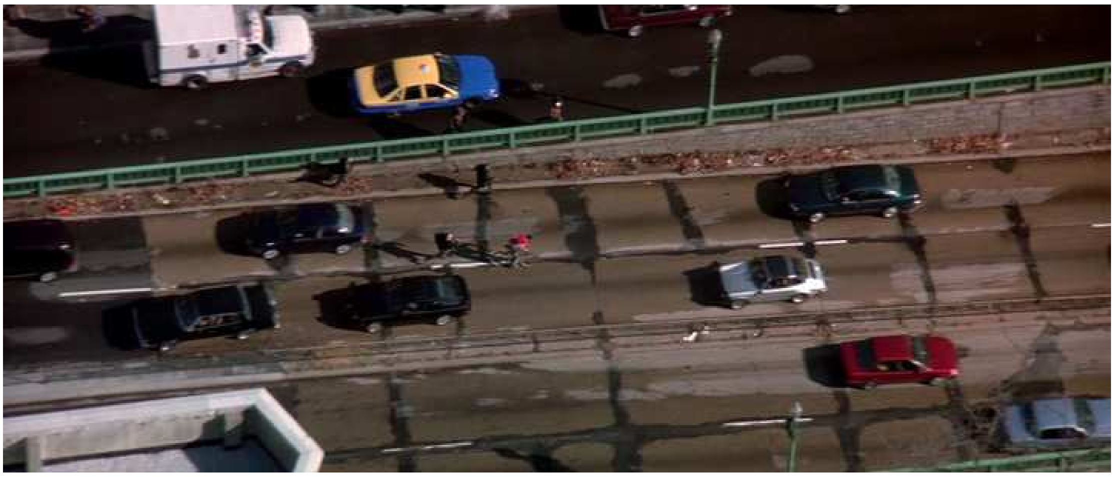


Figure 42. State Enemy sequence, frame 1115

This sequence is a fragment from the *Enemy of the State* movie. This sequence includes outdoor scenes with strong motion at the beginning when the bicyclist runs, as well as scenes with low motion and indoor scenes with normal motion. This sequence has scenes with different lighting conditions.

8.1.4 "Up"

Sequence title	Up
Resolution	720x480
Number of frames	1920
Color space	YV12
Frames per second	24
Source	MPEG-2 (DVD9), 6.5Mbps

Figure 43. Up sequence, frame 638

This sequence is a fragment from the *Up* animated movie. The sequence contains low-contrast scenes with almost static brightness and high-colored scenes, and it contains many scenes with a few frames that include quick scene changes.

8.2 HDTV Sequences

8.2.1 "Amazon"

Sequence title	Amazon
Resolution	1280x720
Number of frames	1200
Color space	YV12
Frames per second	24
Source	Windows Media (6.4Mbps)

Figure 44. Amazon sequence, frame 200

This sequence contains scenes with the camera panning over a landscape view of the Amazon; also, some scene changes take place with the camera panning again. The video frames consist of many edges and high-contrast, sharp details, such as leaves and branches. This sequence was downloaded from the Microsoft website:

http://www.microsoft.com/windows/windowsmedia/musicandvideo/hdvideo/contentshowcase.aspx

8.2.2 "Iron Man"

Sequence title	Iron Man
Resolution	1920x1080
Number of frames	600
Color space	YV12
Frames per second	24
Source	H.264, 14Mbps

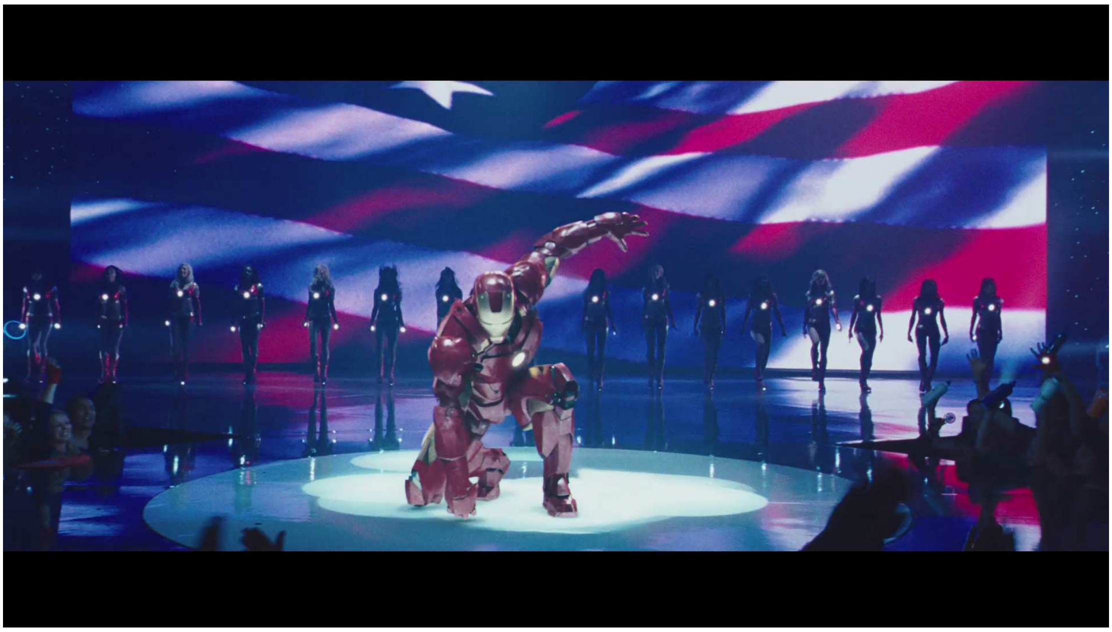


Figure 45. Iron Man sequence, frame 455

This sequence is a part of the *Iron Man 2* movie trailer. It has low-brightness scenes at the beginning, followed by a scene with very fast motion and scenes with slow camera panning. Some frames are very blurry. Also, the sequence contains scenes with flashing lights, which could be a big problem for some encoders. Black letterboxes appear at the top and bottom of the video.

8.2.3 “Mobile Calendar”

Sequence title	Mobile Calendar
Resolution	1280x720
Number of frames	504
Color space	YV12
Frames per second	50
Source	Uncompressed, progressive

Figure 46. Mobile Calendar sequence, frame 416

This close-up sequence is similar to “Mobile&Calendar” and includes a moving calendar with text and a detailed photo of the Vasa ship. It also includes a moving train with colorful toys. The background has two types of wallpaper: one is brown with details, and the other is yellow with drawn figures. The sequence is very detailed and is normally demanding. The main potential compression difficulty is the many small, sharp details on the calendar and on the background.

8.2.4 "Troy"

Sequence title	Troy
Resolution	1920x1072
Number of frames	300
Color space	YV12
Frames per second	24
Source	MPEG-2

Figure 47. Troy sequence, frame 1

This sequence is a fragment of the "Troy" movie and contains three parts with sharp scene changes. The video includes medium scene motion and slow camera motion. In terms of compression, this sequence is difficult to compress because of the many small details.

9 Appendix 6. Tested Codecs and Presets

9.1 Codecs

9.1.1 DivX AVC/H.264 Video Encoder

- Console encoding program version 1.1.1.6
- Presets were chosen by ourselves to meet the comparison requirements

***Remarks**: Owing to our choice of presets, the results for the DivX H.264 encoder could be slightly diminished compared with the case where the developers provide the presets.*

```
DivX AVC/H.264 Video Encoder (version 1.1.1.6) Copyright (c) 2009 DivX, Inc.
Usage: [options] -i <input file> -o <output file>

Available options are:

General:

                    -help   This help information
                       -h
              -noprogress   Do not display progress information
                 -version   Display version information
                 -v <0|1>   Verbose level

Input/Output:

          -i <input file>   AVI file or AVISynth script (avi|avs)
                              Pixel format in one of these formats:
                              YV12 IYUV YUY2 YVYU UYVY BGR24 BGR32
                            or raw yuv (use - for stdin) (requires -y)
         -o <output file>   Raw AVC bit stream
                            (Annex B raw byte stream format, type II)
      -y <width>x<height>   Input resolution (e.g. 1920x1080)
       -fps <int>[/<num>]   Override input frame rate (e.g. 30 or 30/1.001)
                     -tff   Interlaced input, top field first
                     -bff   Interlaced input, bottom field first
    -sar <width>:<height>   Sample Aspect Ratio [1:1]
             -start <int>   First frame to encode
            -frames <int>   Maximum number of frames to encode

Rate control:

                -br <int>   Target bitrate in kbps
                -qf <int>   Target Quality Factor
                              Lower number results in higher quality [0..51]

  Multipass:

             -npass <1|2>   Specify multipass mode
          -sf <stat file>   Specify multipass statistics file name [divx264_stat
.dat]

Encoder:

             -aqo <0|1|2>   Algorithm quality optimized for:
                             0 = Fast encoding
                             1 = Balanced performance/quality (default)
                             2 = Highest quality
                -I <1..4>   Gop length (seconds) [4]
             -fmode <1|2>   Interlace coding mode
                             1 = MBAFF
                             2 = Field
              -ref <1..4>   Maximum number of reference frames [4]
                 -pyramid   Enables pyramid encoding (implies -bref)
                    -bref   Enables B as reference
               -bf <0..3>   Maximum consecutive B-frames [2]
           -threads <int>   Maximum number of threads [auto]


The following frame rates for DivX Plus are permitted:

  60 Hz
  60000/1001 Hz
  50 Hz
  30 Hz
  30000/1001 Hz (i)
  25 Hz (i)
  24 Hz
  24000/1001 Hz

This pre-release version will expire on Sat Sep 04 01:00:00 2010
Please check http://labs.divx.com for new versions.
```

Figure 48. DivX AVC/H.264 video encoder

9.1.2 Elecard AVC Video Encoder 8-bit edition,

- Console encoding program version 2.1.022202.091207
- Codec and presets were provided by Elecard Ltd Company specifically for this test

```
Elecard AVC Video Encoder 8-bit edition, ver. 2.1.022202.091207
usage: avcenc.exe config.cfg [parameters list]
```

Figure 49. Elecard AVC Video Encoder 8-bit edition

9.1.3 Intel® MediaSDK AVC/H.264

- Console encoding program, version 1.10.1.15
- Codec and presets were provided by Intel Corp. specifically for this test

```
mfx_transcoder, version 1.10.1.15
mfx_transcoder is DXVA2 test application developed by Intel/SSG/VCSD/CIP.
This application is for Intel INTERNAL use only!

Usage: mfx_transcoder.exe [Parameters]
```

Figure 50. MediaSDK encoder

9.1.4 MainConcept AVC/H.264 Video Encoder Console Application

- Console encoding version 8.5.0
- Codec and presets were provided by MainConcept AG Company specifically for this test

```
MC H.264/AVC encoder (build 8.5.0 at 2010/01/20 rev. 12837)
Copyright (c) 2010 MC

THIS SOFTWARE IS FOR EVALUATION PURPOSES ONLY!
```

Figure 51. MainConcept H.264/AVC encoder

9.1.5 Microsoft Expression Encoder 3

- GUI encoding program
- Presets chosen ourselves for the analysis

Remarks: Owing to a long initial GUI loading time, the encoding time for Microsoft Expression Encoder is significantly higher than for other encoders. Unfortunately, we were unable to encode multiple files without the GUI loading each time. This is likely one of reasons for the encoder's poor time results.

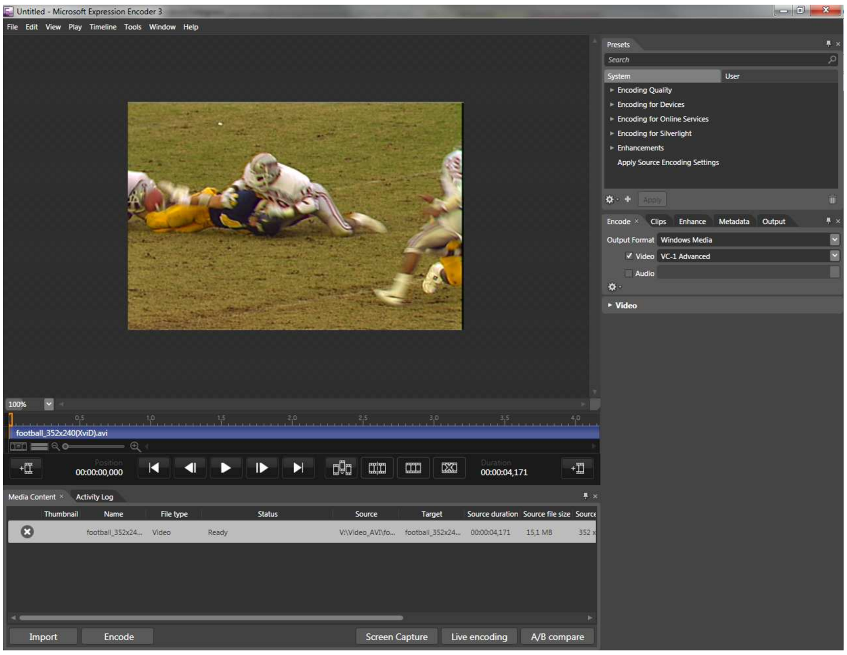


Figure 52. Microsoft Expression encoder

9.1.6 Theora encoder

- Both encoder and decoder were provided by developers

Remarks: Theora's low encoding speed is explained by the version's lack of multithreading support. Thus, Theora was tested in single-thread mode.

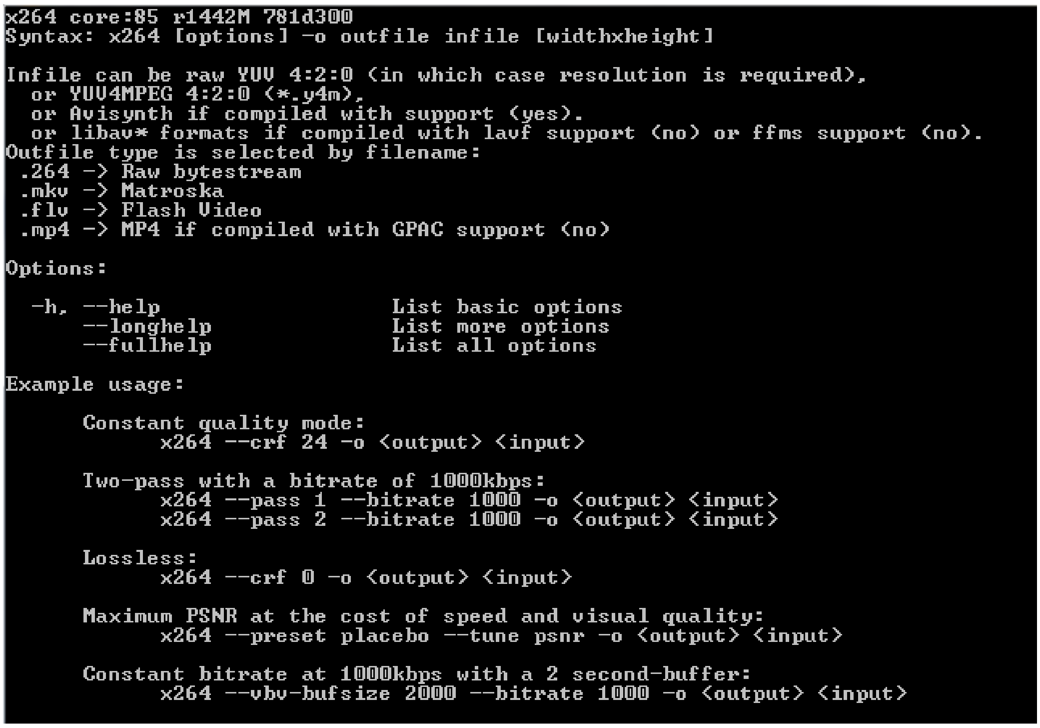
```
encoder_example.exe: invalid option -- h
Usage: encoder_example [options] [audio_file] video_file

Options:

  -o --output <filename.ogv>      file name for encoded output;
                                  If this option is not given, the
                                  compressed data is sent to stdout.

  -A --audio-rate-target <n>      bitrate target for Vorbis audio;
                                  use -a and not -A if at all possible,
                                  as -a gives higher quality for a given
                                  bitrate.

  -V --video-rate-target <n>      bitrate target for Theora video

     --soft-target                Use a large reservoir and treat the rate
                                  as a soft target; rate control is less
                                  strict but resulting quality is usually
                                  higher/smoother overall. Soft target also
                                  allows an optional -v setting to specify
                                  a minimum allowed quality.

     --two-pass                   Compress input using two-pass rate control
                                  This option requires that the input to the
                                  to the encoder is seekable and performs
                                  both passes automatically.

     --first-pass <filename>      Perform first-pass of a two-pass rate
                                  controlled encoding, saving pass data to
                                  <filename> for a later second pass

     --second-pass <filename>     Perform second-pass of a two-pass rate
                                  controlled encoding, reading first-pass
                                  data from <filename>.  The first pass
                                  data must come from a first encoding pass
                                  using identical input video to work
                                  properly.
```

Figure 53. Theora encoder

9.1.7 x264

- Console encoding program version core:85 r1442M 781d300
- Codec and presets were provided by developers specifically for this test

Remarks: The presets provided by the developers for this comparison were specifically chosen for the SSIM metric.

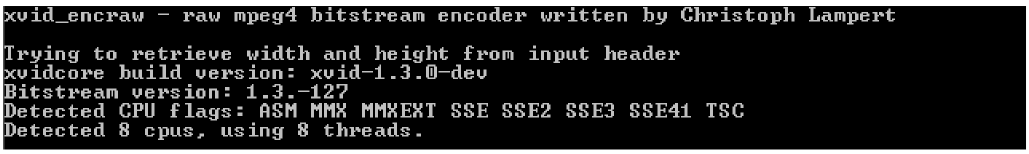


```
x264 core:85 r1442M 781d300
Syntax: x264 [options] -o outfile infile [widthxheight]

Infile can be raw YUV 4:2:0 (in which case resolution is required),
  or YUV4MPEG 4:2:0 (*.y4m),
  or Avisynth if compiled with support (yes).
  or libav* formats if compiled with lavf support (no) or ffms support (no).
Outfile type is selected by filename:
 .264 -> Raw bytestream
 .mkv -> Matroska
 .flv -> Flash Video
 .mp4 -> MP4 if compiled with GPAC support (no)

Options:

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

Example usage:

      Constant quality mode:
            x264 --crf 24 -o <output> <input>

      Two-pass with a bitrate of 1000kbps:
            x264 --pass 1 --bitrate 1000 -o <output> <input>
            x264 --pass 2 --bitrate 1000 -o <output> <input>

      Lossless:
            x264 --crf 0 -o <output> <input>

      Maximum PSNR at the cost of speed and visual quality:
            x264 --preset placebo --tune psnr -o <output> <input>

      Constant bitrate at 1000kbps with a 2 second-buffer:
            x264 --vbv-bufsize 2000 --bitrate 1000 -o <output> <input>
```

Figure 54. x264 encoder

9.1.8 XviD raw mpeg4 bitstream encoder

- Console encoding program
- Codec and presets were provided by developers especially for this test



```
xvid_encraw - raw mpeg4 bitstream encoder written by Christoph Lampert

Trying to retrieve width and height from input header
xvidcore build version: xvid-1.3.0-dev
Bitstream version: 1.3.-127
Detected CPU flags: ASM MMX MMXEXT SSE SSE2 SSE3 SSE41 TSC
Detected 8 cpus, using 8 threads.
```

Figure 55. XviD encoder

9.2 Presets

The table below lists the settings used in this comparison for all of the codecs.

<table>
<tr><th>Codec</th><th>Preset Name</th><th>Preset</th></tr>
<tr><td rowspan="5">DivX H.264</td><td>Movie “High Speed”</td><td>-aqo 0 -ref 1 -bf 0</td></tr>
<tr><td>Movie “Normal”</td><td>Default presets</td></tr>
<tr><td>Movie “High Quality”</td><td>1-st pass: -npass 1
2-nd pass: -npass 2</td></tr>
<tr><td>HDTV “High Speed”</td><td>-aqo 0 -ref 1 -bf 0</td></tr>
<tr><td>HDTV “Normal”</td><td>-aqo 0</td></tr>
<tr><td></td><td>HDTV “High Quality”</td><td>-bf 3 -pyramid -bref</td></tr>
<tr><td>Elecard</td><td>Movie, HDTV “Normal”</td><td>
<table>
<tr><th>Parameter name</th><th>Value</th><th>Comment</th></tr>
<tr><td>AffMode</td><td>0</td><td>0 - frame</td></tr>
<tr><td>BMax</td><td>2</td><td>max number of b-frames</td></tr>
<tr><td>BMode</td><td>0</td><td>0 - plain vanilla</td></tr>
<tr><td>Lookahead</td><td>3</td><td>lookahead length in seconds</td></tr>
<tr><td>OffsetCb</td><td>1</td><td>[-10,+10] i prefer 0 or -1</td></tr>
<tr><td>OffsetCr</td><td>1</td><td>[-10,+10] i prefer 0 or -1.</td></tr>
<tr><td>AQMode</td><td>0</td><td>0 - do not use</td></tr>
<tr><td>DeblockAlpha</td><td>-1</td><td>[-6,+6] really depends on source</td></tr>
<tr><td>DeblockBeta</td><td>-1</td><td>[-6,+6] really depends on source.</td></tr>
<tr><td>ModeDecision</td><td>0</td><td>0 - SAD</td></tr>
</table>
</td></tr>
<tr><td></td><td>Movie, HDTV “High Quality”</td><td>
1-st pass:
<table>
<tr><th>Parameter name</th><th>Value</th><th>Comment</th></tr>
<tr><td>AffMode</td><td>0</td><td>0 - frame</td></tr>
<tr><td>BMax</td><td>2</td><td>max number of b-frames</td></tr>
<tr><td>BMode</td><td>0</td><td>0 - plain vanilla</td></tr>
<tr><td>Lookahead</td><td>0</td><td>lookahead length in seconds</td></tr>
<tr><td>OffsetCb</td><td>1</td><td>[-10,+10] i prefer 0 or -1</td></tr>
<tr><td>OffsetCr</td><td>1</td><td>[-10,+10] i prefer 0 or -1.</td></tr>
<tr><td>AQMode</td><td>0</td><td>0 - do not use</td></tr>
<tr><td>Pass</td><td>3</td><td>3 - fast analyse pass</td></tr>
<tr><td>DeblockAlpha</td><td>-1</td><td>[-6,+6] really depends on source</td></tr>
<tr><td>DeblockBeta</td><td>-1</td><td>[-6,+6] really depends on source.</td></tr>
<tr><td>ModeDecision</td><td>0</td><td>0 - SAD</td></tr>
</table>
2-nd pass:
<table>
<tr><th>Parameter</th><th>Value</th><th>Comment</th></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><th>name</th><th></th><th></th></tr>
<tr><td>AffMode</td><td>0</td><td>0 - frame</td></tr>
<tr><td>BMax</td><td>2</td><td>max number of b-frames</td></tr>
<tr><td>BMode</td><td>0</td><td>0 - plain vanilla</td></tr>
<tr><td>Lookahead</td><td>0</td><td>lookahead length in seconds</td></tr>
<tr><td>OffsetCb</td><td>1</td><td>[-10,+10] i prefer 0 or -1</td></tr>
<tr><td>OffsetCr</td><td>1</td><td>[-10,+10] i prefer 0 or -1.</td></tr>
<tr><td>AQMode</td><td>0</td><td>0 - do not use</td></tr>
<tr><td>Pass</td><td>2</td><td>2 - encoding pass</td></tr>
<tr><td>DeblockAlpha</td><td>-1</td><td>[-6,+6] really depends on source</td></tr>
<tr><td>DeblockBeta</td><td>-1</td><td>[-6,+6] really depends on source.</td></tr>
<tr><td>ModeDecision</td><td>2</td><td>2 - RDO</td></tr>
</table>
</td></tr>
<tr><td>Microsoft Expression Encoder 3</td><td>Movie “High Speed”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Fastest</td></tr>
<tr><td></td><td>Movie “Normal”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Normal</td></tr>
<tr><td></td><td>Movie “High Quality”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Best</td></tr>
<tr><td></td><td>HDTV “High Speed”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Fastest</td></tr>
<tr><td></td><td>HDTV “Normal”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Normal</td></tr>
<tr><td></td><td>HDTV “High Quality”</td><td>Encode->OutputFormat = MP4
Encode->Video = H.264 - Main
Encode->Video->Complexity = Best</td></tr>
<tr><td>MediaSDK</td><td>Movie “High Speed” and "Normal"</td><td>-h264 -sw -sys -async 10 -s 0 -l 1 -u 4</td></tr>
<tr><td></td><td>Movie “High Quality”</td><td>-h264 -sw -sys -async 10 -s 0 -l 1 -u 2</td></tr>
<tr><td></td><td>HDTV “Normal”</td><td>-h264 -sw -sys -async 10 -s 0 -l 1 -u 4</td></tr>
<tr><td></td><td>HDTV “High Quality”</td><td>-h264 -sw -sys -async 10 -s 0 -l 1 -u 3</td></tr>
</table>

MainConcept

Movie "High Speed"

Parameter name	Value	Comment
BFramesCount	0	Maximum number of B-frames [0, 3]
BFramesReference	0	
PyramidCoding	0	
AdaptiveB	0	
Pass	0	RC pass number: # 0 - single pass encoding
SubPelMode	2	2 - full, half and quarter pels
SubBlockMode	1	Sub-block motion search: # 1 - use blocks downto 8x8
NumRefFrames	1	Number of reference frames [0, 16]
EnableInter_4x4	1	Enable intra 4x4 mode in inter slices
DeblockMode	0	Deblocking filter mode: # 0 - enable

Movie "Normal"

1-st pass:

Parameter name	Value	Comment
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	1	1 - first pass (gather and write statistics)
SubPelMode	0	Sub-pel motion accuracy: # 0 - full pels only
SubBlockMode	0	Sub-block motion search: # 0 - use 16x16 blocks only
NumRefFrames	4	Number of reference frames [0, 16]
EnableInter_4x4	0	Enable intra 4x4 mode in inter slices
DeblockMode	1	Deblocking filter mode: # 1 - disable

2-nd pass:

Parameter name	Value	Comment
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	2	2 - second pass (use and update

		statistics)
SubPelMode	0	Sub-pel motion accuracy: # 0 - full pels only
SubBlockMode	0	Sub-block motion search: # 0 - use 16x16 blocks only
NumRefFrames	4	Number of reference frames [0, 16]

Movie, HDTV “High Quality”

1-st pass:

Parameter name	Value	Comment
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	1	1 - first pass (gather and write statistics)
SubPelMode	2	Sub-pel motion accuracy: # 0 - full pels only
SubBlockMode	0	Sub-block motion search: # 0 - use 16x16 blocks only
NumRefFrames	4	Number of reference frames [0, 16]
EnableInter_4x4	1	Enable intra 4x4 mode in inter slices
DeblockMode	0	Deblocking filter mode: # 0 - enable

2-nd pass:

Parameter name	Value	Comment
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	2	2 - second pass (use and update statistics)
SubPelMode	0	Sub-pel motion accuracy: # 0 - full pels only
SubBlockMode	0	Sub-block motion search: # 0 - use 16x16 blocks only
NumRefFrames	4	Number of reference frames [0, 16]
FastIntraDecision	0	Enable fast intra modes decisions
FastInterDecision	0	Enable fast inter modes

		decisions
FastMRME	0	Enable fast multi-reference ME
FastSBME	2	Enable fast sub-block ME

HDTV “High Speed”

Parameter name	Value	Comment
TransformType	2	Transform size: 2 - 8x8 only
EnableInter_16x16	0	Enable intra 16x16 mode in inter slices
EnableInter_8x8	0	Enable intra 8x8 mode in inter slices
EnableInter_4x4	0	Enable intra 4x4 mode in inter slices

HDTV “Normal”

1-st pass:

Parameter name	Value	Comment
TransformType	2	Transform size: 2 - 8x8 only
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	1	1 - first pass (gather and write statistics)
SubPelMode	0	Sub-pel motion accuracy: # 0 - full pels only
SubBlockMode	0	Sub-block motion search: # 0 - use 16x16 blocks only
NumRefFrames	3	Number of reference frames [0, 16]
SearchRange	63	Search range in full pel units
EnableInter_4x4	0	Enable intra 4x4 mode in inter slices
EnableInter_16x16	0	Enable intra 16x16 mode in inter slices
DeblockMode	1	Deblocking filter mode: # 1 - disable

2-nd pass:

Parameter name	Value	Comment
BFramesCount	3	Maximum number of B-frames [0, 3]
BFramesReference	1	Enable reference B-frames
PyramidCoding	1	Use pyramid GOP structure
AdaptiveB	1	Enable Adaptive B-frames placement
Pass	2	2 - second pass (use and update statistics)

<table>
<tr><td></td><td></td><td><table><tr><td>SubPelMode</td><td>0</td><td>Sub-pel motion accuracy: # 0 - full pels only</td></tr><tr><td>SubBlockMode</td><td>0</td><td>Sub-block motion search: # 0 - use 16x16 blocks only</td></tr><tr><td>NumRefFrames</td><td>4</td><td>Number of reference frames [0, 16]</td></tr></table></td></tr>
<tr><td rowspan="3">Theora</td><td>Fast</td><td>-v 1 -d 3000 -k 300</td></tr>
<tr><td>Normal</td><td>-v 1 --two-pass -k 300</td></tr>
<tr><td>Best</td><td>-z 0 -v 1 --two-pass -k 300</td></tr>
<tr><td>x264</td><td>Movie “High Speed”</td><td>--b-pyramid normal --tune ssim --keyint 500 --preset fast --ref 1 --bframes 0</td></tr>
<tr><td rowspan="5"></td><td>Movie “Normal”</td><td>1-st pass:
--b-pyramid normal --tune ssim --keyint 500 --preset fast --pass 1 --weightp 0 --subme 5

2-nd pass:
--b-pyramid normal --tune ssim --keyint 500 --preset fast --pass 2 --weightp 0 --subme 5</td></tr>
<tr><td>Movie “High Quality”</td><td>1-st pass:
--b-pyramid normal --tune ssim --keyint 500 --preset slow --pass 1

2-nd pass:
--b-pyramid normal --tune ssim --keyint 500 --preset slow --pass 2</td></tr>
<tr><td>HDTV “High Speed”</td><td>--b-pyramid normal --tune ssim --keyint 500 --preset faster --ref 1 --bframes 0 --subme 3 --trellis 0 --weightp 0</td></tr>
<tr><td>HDTV “Normal”</td><td>1-st pass:
--b-pyramid normal --tune ssim --keyint 500 --preset faster --pass 1 --weightp 0 --subme 3

2-nd pass:
--b-pyramid normal --tune ssim --keyint 500 --preset faster --pass 2 --weightp 0 --subme 3</td></tr>
<tr><td>HDTV “High Quality”</td><td>1-st pass:
--b-pyramid normal --tune ssim --keyint 500 --preset slow --pass 1 --ref 4

2-nd pass:
--b-pyramid normal --tune ssim --</td></tr>
</table>

		keyint 500 --preset slow --pass 2 --ref 4
XviD	Movie "High Speed"	-type 0 -quality 5 -vhqmode 1 -max_bframes 0 -reaction 8 -averaging 50 -smoother 50
	Movie "Normal"	1-st pass: -type 0 -pass1 -quality 6 -vhqmode 1 -ostrength 20 -oimprove 10 -odegrade 10 2-nd pass: -type 0 -pass2 -quality 6 -vhqmode 1 -ostrength 20 -oimprove 10 -odegrade 10
	Movie "High Quality"	1-st pass: -type 0 -pass1 -quality 6 -vhqmode 4 -bvhq -qpel -ostrength 20 -oimprove 10 -odegrade 10 2-nd pass: -type 0 -pass2 -quality 6 -vhqmode 4 -bvhq -qpel -ostrength 20 -oimprove 10 -odegrade 10
	HDTV "High Speed"	-type 0 -quality 5 -vhqmode 1 -max_bframes 0 -reaction 8 -averaging 50 -smoother 50
	HDTV "Normal"	1-st pass: -type 0 -pass1 -quality 6 -vhqmode 1 -ostrength 20 -oimprove 10 -odegrade 10 2-nd pass: -type 0 -pass2 -quality 6 -vhqmode 1 -ostrength 20 -oimprove 10 -odegrade 10
	HDTV "High Quality"	1-st pass: -type 0 -pass1 -quality 6 -vhqmode 4 -bvhq -qpel -ostrength 20 -oimprove 10 -odegrade 10 2-nd pass: -type 0 -pass2 -quality 6 -vhqmode 4 -bvhq -qpel -ostrength 20 -oimprove 10 -odegrade 10

10 Appendix 7. 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.

10.1.1.1 RD curves

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

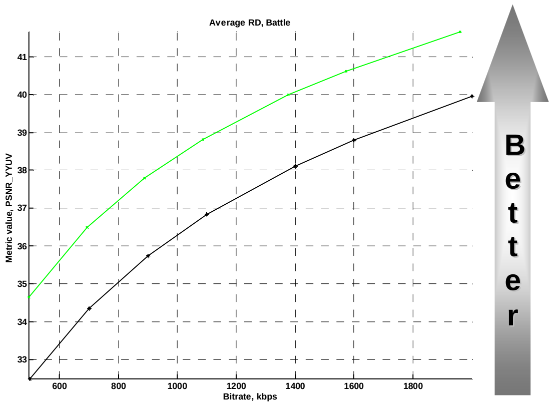
10.1.1.2 Relative Bitrate/Relative Time Charts

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.

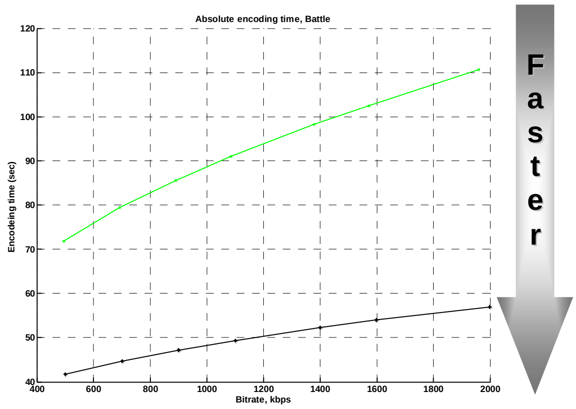
The X-axis shows the relative encoding time for the codec under test. Larger values indicate a slower codec. For example, a value of 2.5 means that the codec under test works 2.5 times slower, on average, than the reference codec.

10.1.1.3 Graph Example

Figure 56 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.



RD curve. “Green” codec is better!



Encoding time (seconds). “Green codec is slower!

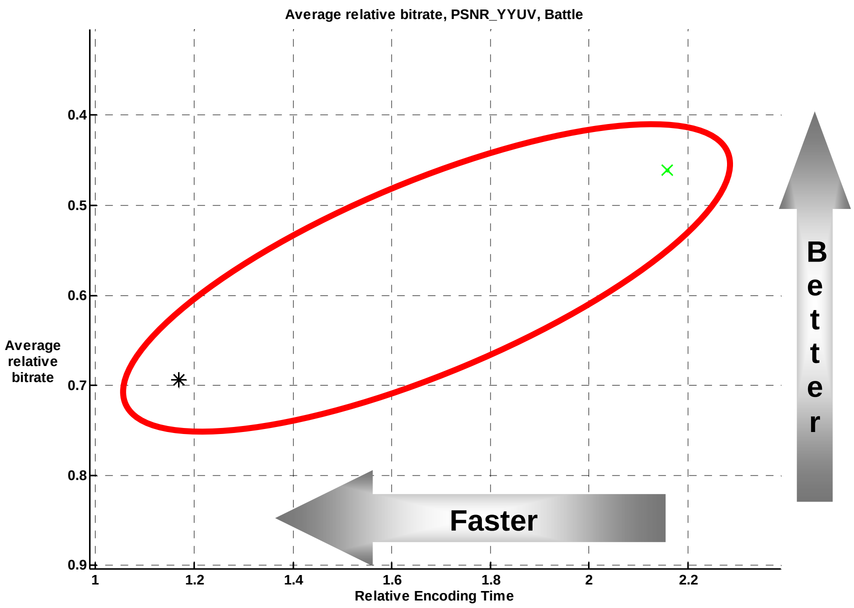


Figure 56. Integral situation with codecs. This plot shows the situation more clearly.

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.

10.2 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 58). 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 59). This result is an average bitrate ratio for a fixed quality for the two codecs. If 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.

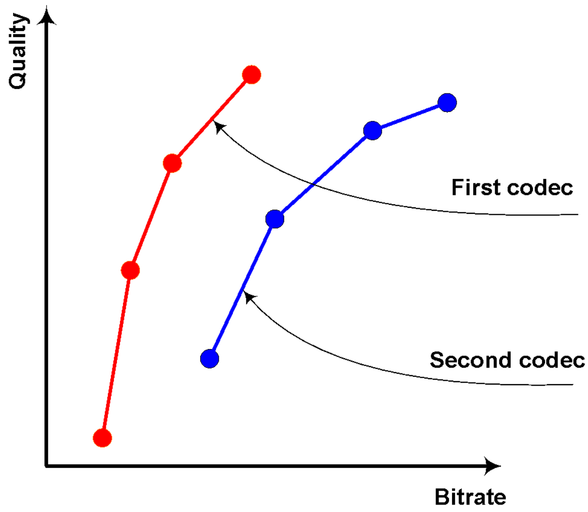


Figure 57. Source Data

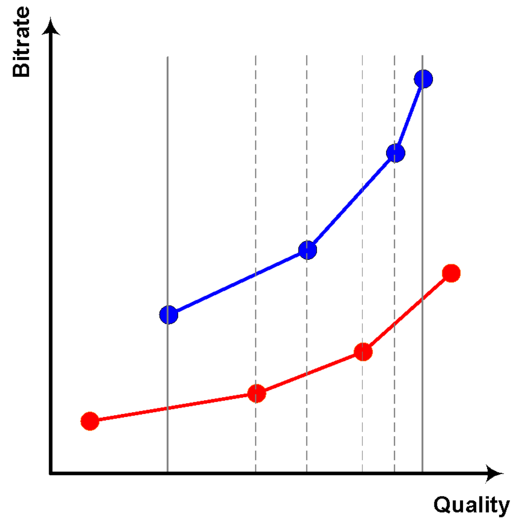


Figure 58. Axes' Inversion and Averaging Interval Choosing

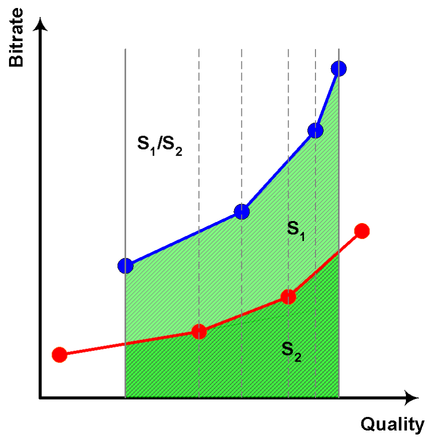


Figure 59. Areas' under Curves Ratio

10.3 Relative Codec Encoding Time Computation

To compute the relative processing time of two codecs for a particular video sequence, the encoding time is calculated for both codecs (the encoding times are summed for all bitrates) and the ratio is taken. For three or more codecs, one codec is chosen as a reference and the ratio of its encoding time to that of the others is calculated.

For multiple sequences, each codec is assigned an arithmetic mean of average relative encoding times for each sequence.

11 Appendix 7. Objective Quality Metrics Description

11.1 SSIM (Structural SIMilarity)

11.1.1 Brief Description

The original paper on the SSIM metric was published by Wang, et al.[2] The paper can be found at the following URL: http://ieeexplore.ieee.org/iel5/83/28667/01284395.pdf

The SSIM author homepage is found at the following URL: 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:

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

where

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

$$\sigma_x = \left(\sum_{i=1}^{N} \omega_i (x_i - \mu_x) \right)^{\frac{1}{2}}$$

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

The constants C_1 and C_2 are defined according to the following expressions:

$$C_1=(K_1L)^2$$
$$C_2=(K_2L)^2$$

where L is the dynamic range of the pixel values (255 for 8-bit grayscale images), and $K_1, K_2 << 1$.

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

[2] 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.

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.

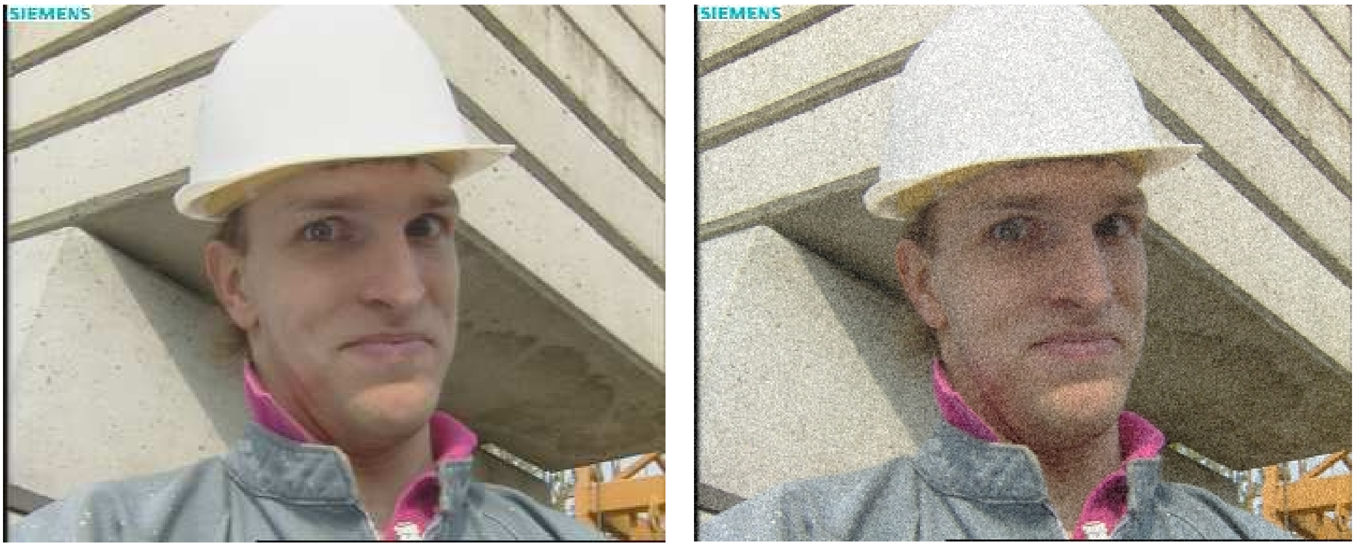
11.1.2 Examples

The following is an 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.

Original Processed SSIM

Figure 60. SSIM example for compressed image

The following are more examples how various types of distortion influence the SSIM value.



Original image Image with added noise

Blurred image

Sharpen image

Figure 61. Original and processed images (for SSIM example)

The SSIM values for the Y-plane for these images are given below.

SSIM for image with itself, value = 1

SSIM for image with noisy image,
value = 0.552119

SSIM for image with blurred image, value = 0.9225

SSIM for image with sharpen image, value = 0.958917

Figure 62. SSIM values for original and processed images

11.2 PSNR (Peak Signal-to-Noise Ratio)

11.2.1 Brief Description

This metric, which is often used in actual practice, is called the peak signal-to-noise ratio, or PSNR.

$$d(X,Y) = 10 \cdot \log_{10} \frac{255^2 \cdot m \cdot n}{\sum_{i=1,j=1}^{m,n} (x_{ij} - y_{ij})^2},$$

Where *d(X,Y)* – PSNR value between *X* and *Y* frames

x_{ij} – the pixel value for *(i,j)* position for the *X* frame

y_{ij} – the pixel value for *(i,j)* position for the *Y* frame

m,n – frame size *mxn*

Generally, this metric has the same form as the mean square error (MSE), but it is more convenient to use because of the logarithmic scale. It still has the same disadvantages as the MSE metric, however.

In MSU Video Quality Measurement Tool the PSNR can be calculated for all YUV and RGB components and for the L component of LUV color space. The PSNR value is quick and easy to calculate, but it is sometimes inappropriate as relates to human visual perception.

A maximum deviation of 255 is used for the PSNR for the RGB and YUV color components because, in YUV files, there is 1 byte for each color component. The maximum possible difference, therefore, is 255. For the LUV color space, the maximum deviation is 100.

The values of the PSNR in the LUV color space are in the range [0, 100]; the value 100 means that the frames are identical.

11.2.2 Examples

PSNR visualization uses different colors for better visual representation:

- Black – value is very small (99 – 100)
- Blue – value is small (35 – 99)
- Green – value is moderate (20 – 35)
- Yellow –value is high (17 – 20)
- Red –value is very high (0 – 17)

The following is an example of the PSNR metric:

Original Processed PSNR

Figure 63. PSNR example for two frames

The following are further examples demonstrating how various distortions can influence the PSNR value.

Original image

Image with added noise

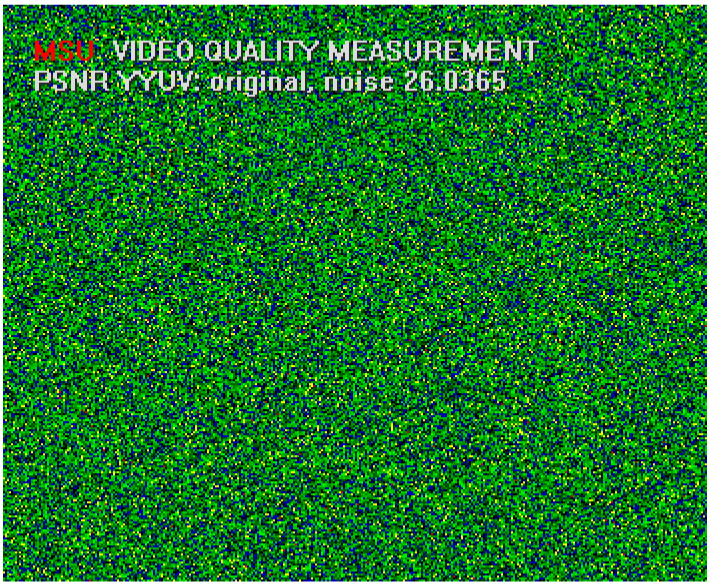
Blurred image

Sharpen image

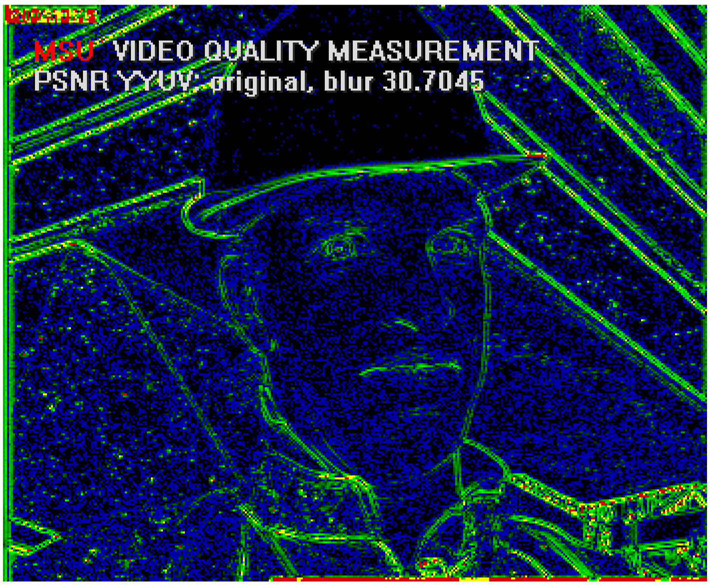
Figure 64. Original and processed images (for PSNR example)

Next are the PSNR values for the Y–plane for these images

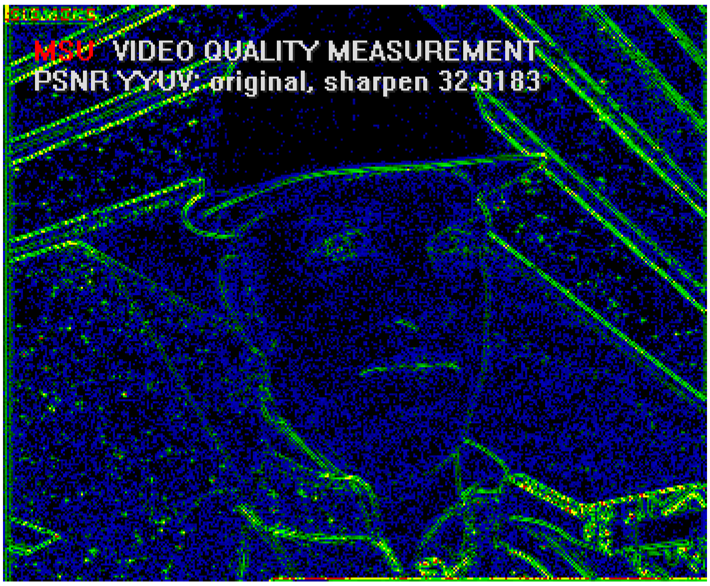
PSNR for image with itself, value = 0



PSNR for image with noisy image, value = 26.0365



PSNR for image with blurred image, value = 30.7045



PSNR for image with sharpen image, value = 32.9183

Figure 65. PSNR values for original and processed images

12 List of Figures

13 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.

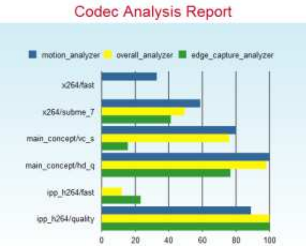
E-mail: video@graphics.cs.msu.ru

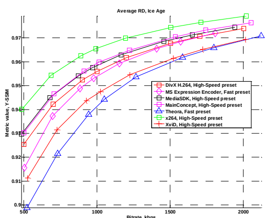
Codec Analysis and Tuning

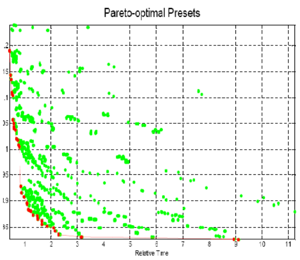
GRAPHICS & MEDIA LAB
VIDEO GROUP

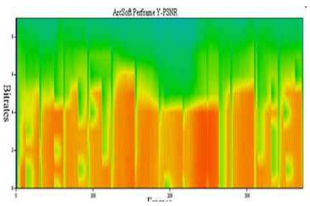
Computer Graphics and Multimedia Laboratory of Moscow State University:

- 8 years working in the area of video codec analysis and tuning using objective quality metrics and subjective comparisons.
- 16 reports of video codec comparisons and analysis (H.264, MPEG-4, MPEG-2, decoders' error recovery).
- Methods and algorithms for codec comparison and analysis development, separate codec's features and codec's options analysis.
- We have helped many companies with private independent codec analysis.









Strong and Weak Points of Your Codec

Deep encoder parts analysis (ME, RC on GOP, mode decision, etc). Weak and strong points for your encoder and complete information about encoding quality on different content types. Encoding Quality improvement by the pre and post filtering (including technologies licensing).

Independent Encoding Quality Estimation for Different Use-cases

Comparative analysis of your encoder and other encoders. You will know encoding quality, speed, bitrate handling and other results for different use-cases (movies, HDTV, broadcasting, transcoding, etc.).

Encoder Features Implementation Efficiency Analysis

We perform encoder features efficiency (speed/quality trade-off) analysis that could lead up to 30% speed/quality characteristics of your codec increase. We can help you to tune your codec and find best encoding parameters.

If you have any questions – do not hesitate to ask!

videocodec-testing@graphics.cs.msu.ru

Special thanks to our contributors:

intel Adobe AMD ATI dicas DOLBY
Google KDDI KDDI R&D LABS NVIDIA QUALCOMM real TATA ELXSI Engineering Creativity