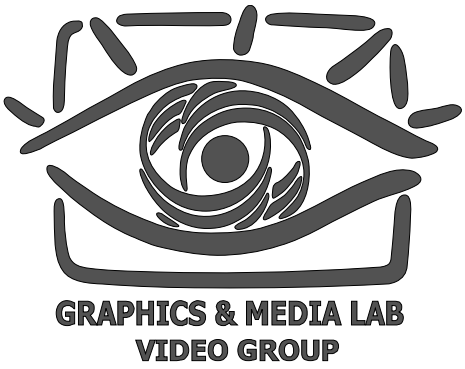


MPEG-4 AVC/H.264 Video Codec Comparison

Project head: Dmitriy Vatolin
Measurements, analysis: Dmitriy Kulikov, Alexander Parshin
Translation: Artem Titarenko
Verification: Stanislav Soldatov

Codecs:

DivX 6.0 (MPEG4 ASP codec)
ArcSoft H.264
Ateme H.264
ATI H.264
Elecard H.264
Fraunhofer IIS H.264
VSS H.264
x264

December 2005

CS MSU Graphics&Media Lab
Video Group

http://www.compression.ru/video/
videocodec-testing@graphics.cs.msu.ru

Contents

Thanks

We would like to express our gratitude to the following companies for providing us with codecs and settings:

- **ArcSoft, Inc.**
- **Ateme**
- **ATI Technologies Inc.**
- **Elecard LTD**
- **Fraunhofer Institute for Integrated Circuits**
- **Vanguard Software Solutions, Inc.**
- **x264 Development Team**

We would also like to thank these companies for their technical support and help during our tests.

Overview

Codecs

Codec	Developer	Version
1. DivX	DivXNetworks, Inc	6.0
2. ArcSoft H.264	ArcSoft, Inc.	dev. version for 19.09.05
3. Ateme H.264	Ateme	1.2.1.6
4. ATI H.264	ATI Technologies Inc.	3.1.2
5. Elecard H.264	Elecard LTD	dev. version for 09.09.05
6. Franhofer IIS H.264	Fraunhofer Institute for Integrated Circuits	dev. version for 20.09.05
7. VSS H.264	Vanguard Software Solutions, Inc.	3.0.2.7
8. x264	Development group x264	revision 293

Note: DivX 6.0 is not H.264 codec. It is MPEG-4 ASP codec, it takes part in this testing only as one of the best representative of previous video codec standard MPEG-4

Sequences

Sequence	Number of frames	Frames per second	Spatial dimension and color space
1. foreman	300	30	352x288(YV12)
2. susi	374	25	704x576(YV12)
3. bbc	374	25	704x576(YV12)
4. battle	1599	24	704x288(YV12)
5. simpsons	365	24	720x480(YV12)
6. matrix	239	25	720x416(YV12)
7. concert	390	25	1664x1088(YV12)

Goal and testing rules

H.264 Codecs Testing Objectives

The main goal of this document is an evaluation of the quality of new H.264 codecs, using objective metrics for comparison. The comparison was done using the settings provided by the developers for each codec.

Testing rules

- Every codecs developers provided settings for both "Best Quality" (high quality compression) and "Best Speed" (better processing speed) presets, except DivX.
- Codec restrictions: Only codecs that supported MPEG-4 AVC/H.264 Main profile and single pass encoding were tested.
- Each codec was tested for speed 3 times, after this the median score (the middle value of the three measurements) was used as the resulting time.
- The following 10 bitrates were used (in Kilobits per second) 100, 225, 340, 460, 700, 938, 1140, 1340, 1840, 2340.
- During the testing three types of video sequences were used:
  - Source sequences (.yuv extension) in the YUV 4:2:0 format.
  - (.avi extension) sequences in the YUV 4:2:0 format."

    These sequences were made in the following way:
    Sequence was opened in VirtualDub using AviSynth script and, after that, it was saved in AVI file format. VirtualDub parameters “Color depth – Decompression Format” and “Color depth – Output format for compressor” were set to “Autoselect” and “Same as decompression format” respectively.
    AviSynth script for sequence opening:

    RawSource("<source_file>.yuv", <wigth>, <height>, "I420")
    AssumeFPS(<source_fps*1000>, 1000)

    After that in obtained YUV file fourcc was manually changed from “DIB “ to “YV12” using HexEditor from VirtualDub. This was required for correct work of Ateme codec.
  - *.yuv sequences with duplicated last frame. These sequences were used for testing x264 codec (see its description). Written for this case, special utility was used for generating such sequences.
- In this testing VirtualDub version 1.6.10 was used.
- In this testing AviSynth version 2.55 was used.
- For all metrics’ measurements in this test special build of MSU Video Quality Measure was used.

- Two computers with following configuration were used for encoding:
  - Processor: Pentium 4, 2.8 GHz with Hyper Threading
  - Operating system: Windows 2000 Pro, SP4
  - Memory: 1Gb
  - Video Accelerator: ATI Radeon 9600 Series
  - Hard Drives: ATA100 2x80Gb

Metrics Used in Comparison

During testing following metrics were measured:

- PSNR (Y, U, V, L, R, G, B components)
- SSIM (Y component)
- VQM (Y component)
- Blurring (Y, U, V components)
- Blocking (Y, U, V components)

Detailed description of these metrics including VQM and SSIM algorithms could be found here:

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

Number of graphs

Graph type	Total number of graph	Inserted into this document
Average metric	**196**	**24 (12%)**
Relative average	**196**	**7 (3.5%)**
Per frame metric	**1414**	**13 (0.9%)**
Bitrate Handling	**14**	**14 (100%)**
Speed/Quality	**1960**	**8 (0.4%)**
Speed	**16**	**16 (100%)**
Total	**3796**	**82 (2%)**

Sequences

Foreman

Sequence title	foreman
Resolution	352x288
Number of frames	300
Color space	YV12
Frames per second	30
Source	Uncompressed (standard sequence), progressive

Picture 1. Frame 77

Picture 2. Frame 258

This is one of the most famous sequences. It represents a face with very rich mimic. Motion is not very intensive here, but on the other hand it is disordered, not forward. Intricate character of motion creates problems for the motion compensation process. In addition camera is shaking that makes the image unsteady. In the end of the sequence camera suddenly turns to the building site and there follows an almost motionless scene. So this sequence also shows codec's behavior on a static scene after intensive motion.

Susi

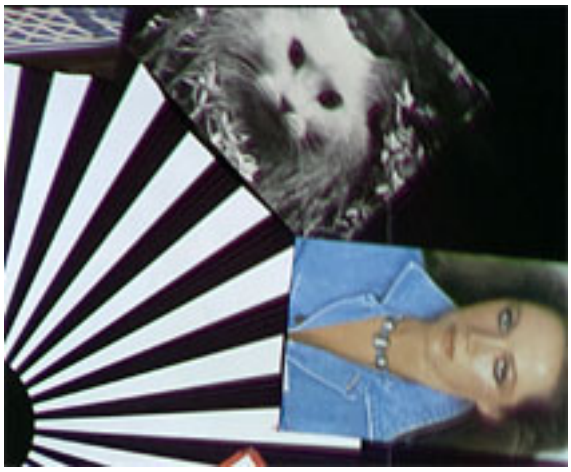
Sequence title	susi
Resolution	704x576
Number of frames	374
Color space	YV12
Frames per second	25
Source	MPEG-2 (40Mbit), Smart Deinterlace

Picture 3. Frame 193

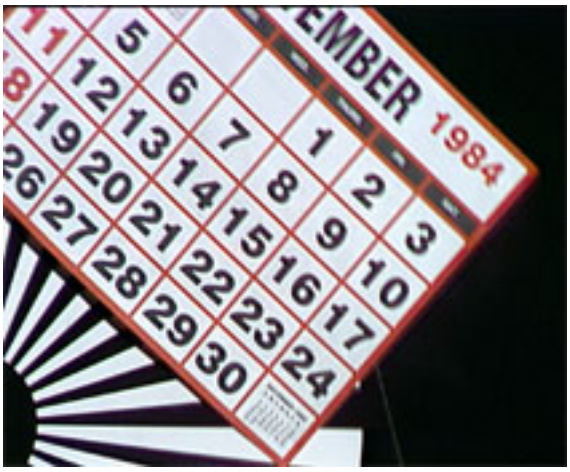
This sequence is characterized by high-level noise and slow motion. In its first part the scene is almost static (the girl only blinks), then there is some motion (she abruptly moves her head) and then the scene becomes almost static again. Noise is suppressed on every second frame due to the B-frames option in MPEG-2 codec.

BBC

Sequence title	bbc
Resolution	704x576
Number of frames	374
Color space	YV12
Frames per second	25
Source	Uncompressed (standard sequence), Smart Deinterlace



Picture 4. Frame 185



Picture 5. Frame 258

This sequence is characterized by pronounced rotary motion. It contains a rotating striped drum with different pictures and photos on it. Quality of the compressed sequence can be evaluated by observing details on these images.

Battle

Sequence title	battle
Resolution	704x288
Number of frames	1599
Color space	YV12
Frames per second	24
Source	MPEG-2 (DVD), FlaskMPEG deinterlace

Picture 6. Frame 839

This sequence is a fragment of the “Terminator-2” movie, which represents its very beginning. In terms of compression this sequence is the most difficult one among all other sequences that took part in the testing. That is because of three main reasons: constant brightness changes (explosions and laser flashes, see the picture above), very quick motion and frequent changes of the scene that make codecs often compress frames as I-frames.

Simpsons

Sequence title	simpsons
Resolution	720x480
Number of frames	365
Color space	YV12
Frames per second	24
Source	MPEG-2 (DVD), progressive

Picture 7. Frame 50

This sequence is a fragment of “Simpsons” cartoon film. This is a classical representative of cartoon films: sketchy movement, great number of monochrome regions with abrupt junctions between them. Previously this sequence was compressed in MPEG-2 with rather low bitrate.

Matrix

Sequence title	matrix
Resolution	720x416
Number of frames	239
Color space	YV12
Frames per second	25
Source	MPEG-2 (DVD), Smart Deinterlace

Picture 8. Frame 226

This sequence is a fragment of ”Matrix” movie. Relatively simple movement, quite dim colors and small resolution allows codec to treat this sequence in rather simple way.

Concert

Sequence title	concert
Resolution	1664x1088
Number of frames	390
Color space	YV12
Frames per second	25
Source	MPEG-2 (DVD), Smart Deinterlace

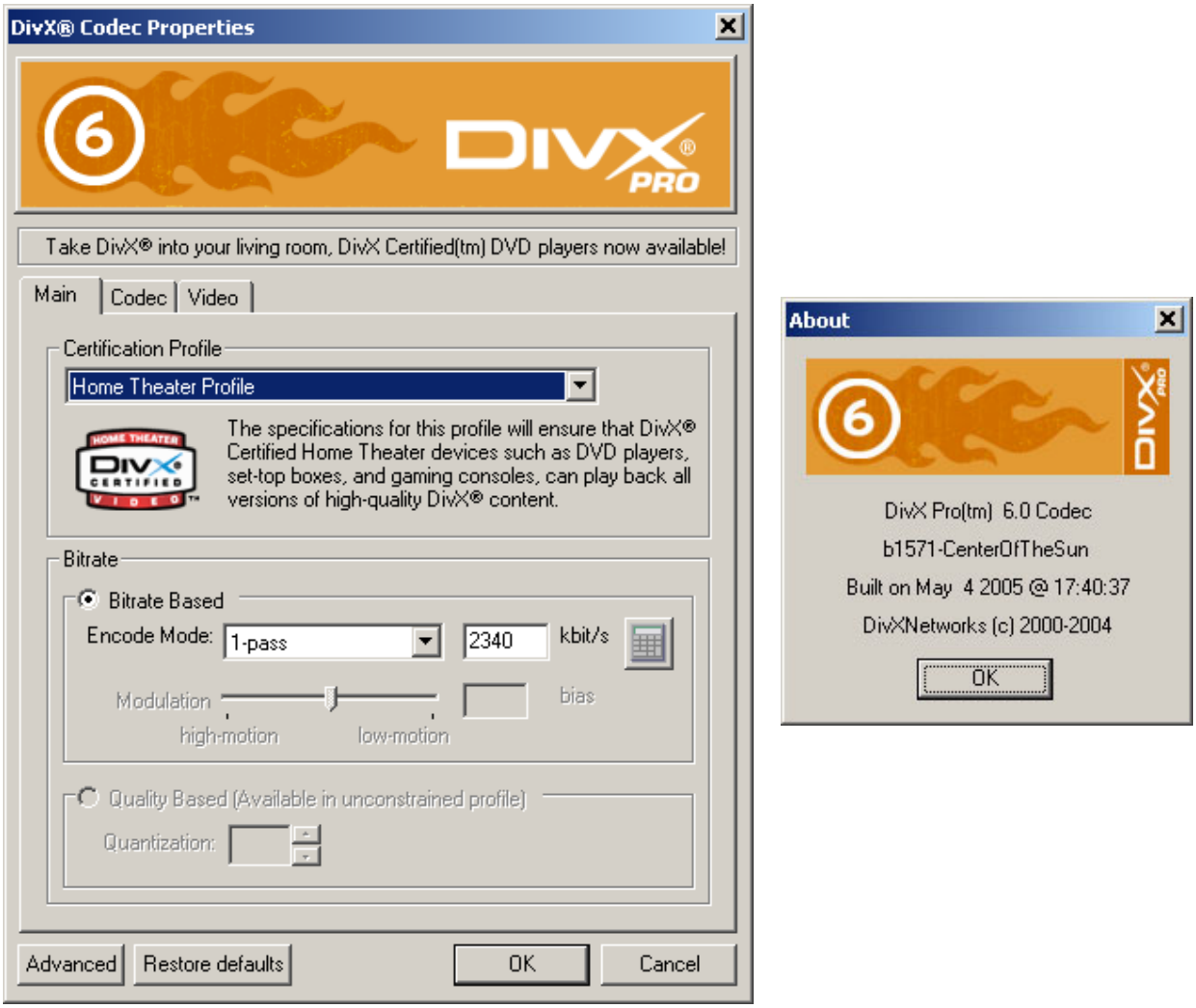
Picture 9. frame 128

This sequence is a part of HDTV broadcast of symphonic orchestra concert. Sequence's spatial resolution is very high. At the same time motion is rather simple and sometimes it completely disappears. There are two scene changes in this sequence.

Codecs

DivX 6.0

- DivX 6.0 is not H.264 codec. It is MPEG-4 ASP codec, it takes part in this testing only as one of the best representative of previous video codec standard MPEG-4
- This is a VfW (Video for Windows) codec.
- Compression was performed using VirtualDub 1.6.10 video processing program.
- Evaluation version of codec works for 6 months.
- There were no presets from developers. All tests were performed using “Home Theater Profile”. For different test presets only Codec Performance settings were changed:
  - For “Best Quality” mode preset “Insane quality” was used
  - For “Best Speed” mode preset “Balanced” was used



Picture 10. DivX 6.0

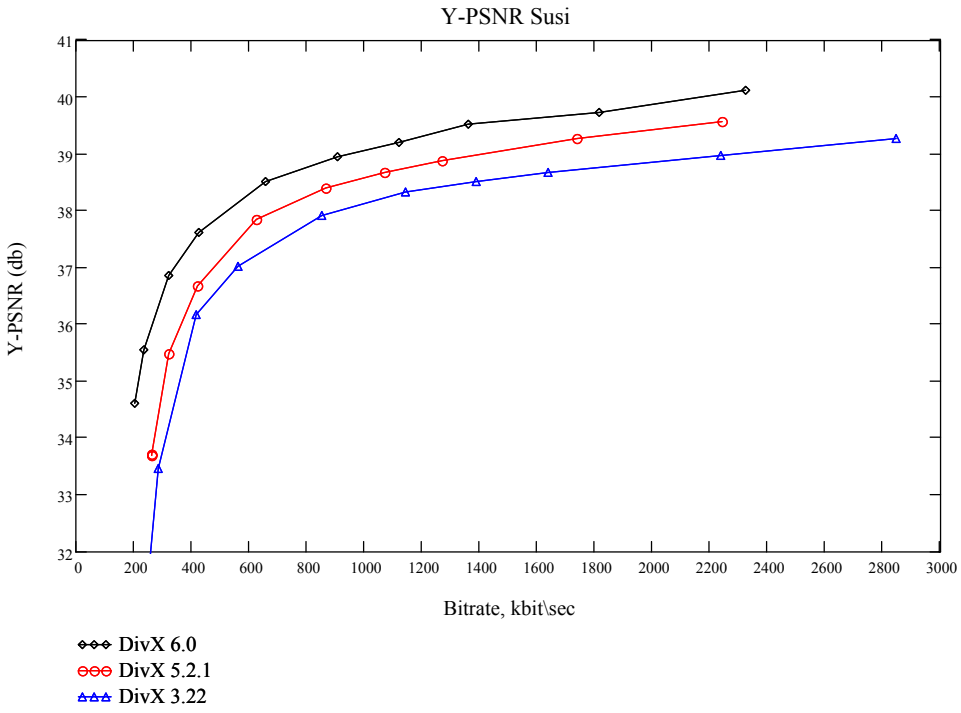
Remarks:

Codec shifts source video sequence by one frame while using its own decoder. With other decoders (specifically, XviD) this effect disappears. For removing such shift next AviSynth script was used:

```
clip = AVISource("source.avi")
clip = clip.DeleteFrame(0)
clip = clip.DuplicateFrame(clip.FrameCount()-1)
return clip
```

Codec generates an error on “concert” sequence’s compression.

However, despite these drawbacks, codec significantly improved compression quality in comparison with previous version. On the next graph there is a comparison between different versions of DivX (metric is Y-PSNR):



Picture 11. DivX 6.0, DivX 5.21 and DivX 3.22 comparison

ArcSoft H.264

- Console encoding program.
- DirectShow filters were used for decoding.
- Codec (encoder and decoder) and presets were provided by ArcSoft, Inc company specially for this test.

Remarks:

Codec works without remarks.

Ateme H.264

Picture 12. Ateme H.264

- Console encoding program.
- Reference decoder JM9.8 was used for decoding.
- Codec and presets were provided by Ateme company specially for testing.

Remarks:

Codec stores compressed sequence in MPEG4 container. Program mp4box version 0.4.0 was used for extracting H.264 stream from it.

ATI H.264

- Console encoding program.
- Reference decoder JM9.8 was used for decoding.
- Codec and presets were provided by ATI Technologies Inc. company specially for this test.

Remarks:

Codec has only “Best Speed” preset.

Elecard H.264

- Console encoding program.
- Console decoder provided by developers was used for decoding.
- Codec (encoder and decoder) and presets were provided by Elecard LTD company specially for this test.

Remarks:

Codec works without remarks.

Fraunhofer IIS H.264

```
H.264/MPEG-4 AVC Baseline, Main and Extended Profile Encoder
(C) 2002-2005 Fraunhofer IIS, All Rights Reserved


 supported options:

  -profile Profile (Baseline: 66, Main: 77, Extended: 88, High: 100)
  -quality Quality Of Encoding [1..6]

  -pf      Parameter File Name

  -if      Input File Name
  -of      Output File Name
  -bf      Bitstream File Name
  -width   Input Frame width [luma samples]
  -height  Input Frame width [luma samples]
  -lf      Loop Filter Idc (0=filter all edges, 1=filter no edges, 2=slice bound
aries)
  -rc      Rate Control Type
  -cr      Target Bit Rate (bps)
  -bd      Buffer Delay (sec)
  -pi      Distance Between I Frames
  -ti      I Frames IDR distance (-1=fist only)
  -pb      Number Of Inserted B Frames
  -pt      Total Number Of Frames
  -ifreq   Input Frequency [Hz]
  -ofreq   Output Frequency [Hz]
  -qpir    I Frame Qp of ref picture
  -qppr    P Frame Qp of ref picture
  -qpbr    B Frame Qp of ref picture
  -qpin    I Frame Qp of non-ref picture
  -qppn    P Frame Qp of non-ref picture
  -qpbn    B Frame Qp of non-ref picture
  -qpall   all Qp
  -qpmin   minimum Qp
  -qpmax   maximum Qp
  -dsf     Direct Spatial Flag (0=Temporal,1=Spatial)
  -sr      Search Range [Full Pel Positions]
  -sm      Search Mode
  -sff     Search Func Full Pel
  -sfs     Search Func Sub Pel
  -sp      Skip Percent
  -swip    SampleWeighting IPSlice (0-off,1-on,2-random)
  -swb     SampleWeighting B Slice (0-off,1-explicit,2-implicit3-random)
  -sym     Symbol Mode
  -cai     CABAC initialization
```

Picture 13. Fraunhofer IIS H.264

- Console encoding program.
- Reference decoder JM9.8 was used for decoding.
- Codec and presets were provided by Fraunhofer Institute for Integrated Circuits IIS specially for this test.

Remarks:

Codec works without remarks.

VSS H.264

```
==> vssh-pro
vssh-pro v3.0.2.7 (20050831) OPT=1 MT PREP=IPP PROTECT=ASP PROFILE=High
Copyright (C) 2002-2005 Vanguard Software Solutions, Inc.
Registered to [compression.ru]
USAGE:
        vssh-pro.exe [-option value]
WHERE options are:
common:
        op      - operation mask (0=all, 1=prep, 2=enc, 4=dec, 8=avi);
        vb      - verbose prints (0/1);
        tt      - test title;
files:
        cf      - common configuration file;
        ec      - encoder configuration file;
        yf      - input YUV file (uncompressed 4:2:0 YUV);
        hf      - output H.264 file (compressed 26l file);
        df      - output YUV file (uncompressed 4:2:0 YUV);
        af      - output AVI file (compressed AVI file);
        rf      - output report file;
frames:
        fw      - input frame width in pixels;
        fh      - input frame height in pixels;
        fs      - number of start frame;
        fn      - number of frames to process;
        fr      - frame rate (frames per 10,000 seconds);
        fk      - number of frames to skip on input after every read;
codec config:
        mp      - multipass number (0/1/2);
        snr     - calc SNR (0/1);
        material- input material (0=progressive, 1=interlaced);
        quality - codec quality (0..5);
        bitrate - output bitrate, kbps;
        preload - enable input file preload in RAM;
        rtp     - size of RTP packets on output (default 1500);
        mt      - enable multi-thread (0/1);
```

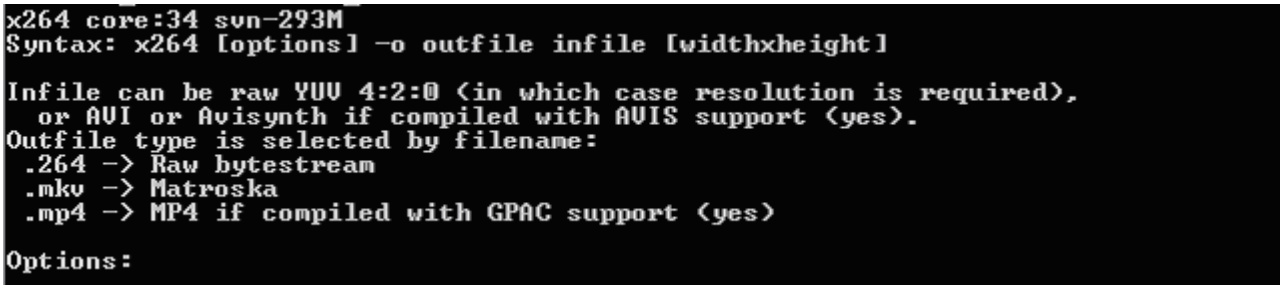
Picture 14. VSS H.264

- Console encoding program.
- Console decoder was used for decoding.
- Codec (encoder and decoder) and presets were provided by Vanguard Software Solutions, Inc. company specially for this test.

Remarks:

Codec generates an error while compressing “foreman” sequence.

x264



Picture 15. x264

- Console encoding program.
- Program ffmpeg 2005.09.09 was used for decoding.
- This codec is an Open Source project.
- Optimal settings were provided by codec's Development Team.

Remarks:

During decoding ffmpeg program does not write last frame of video sequence. Therefore we used sequences with duplicated last frame for this codec.

Preset "Best quality"

Y-PSNR

In this mode codecs tried to show best quality in single-pass mode. Therefore major importance in this section is paid to graphs of quality of compressed sequences in different metrics. All H.264 codecs used main profile of encoding standard.

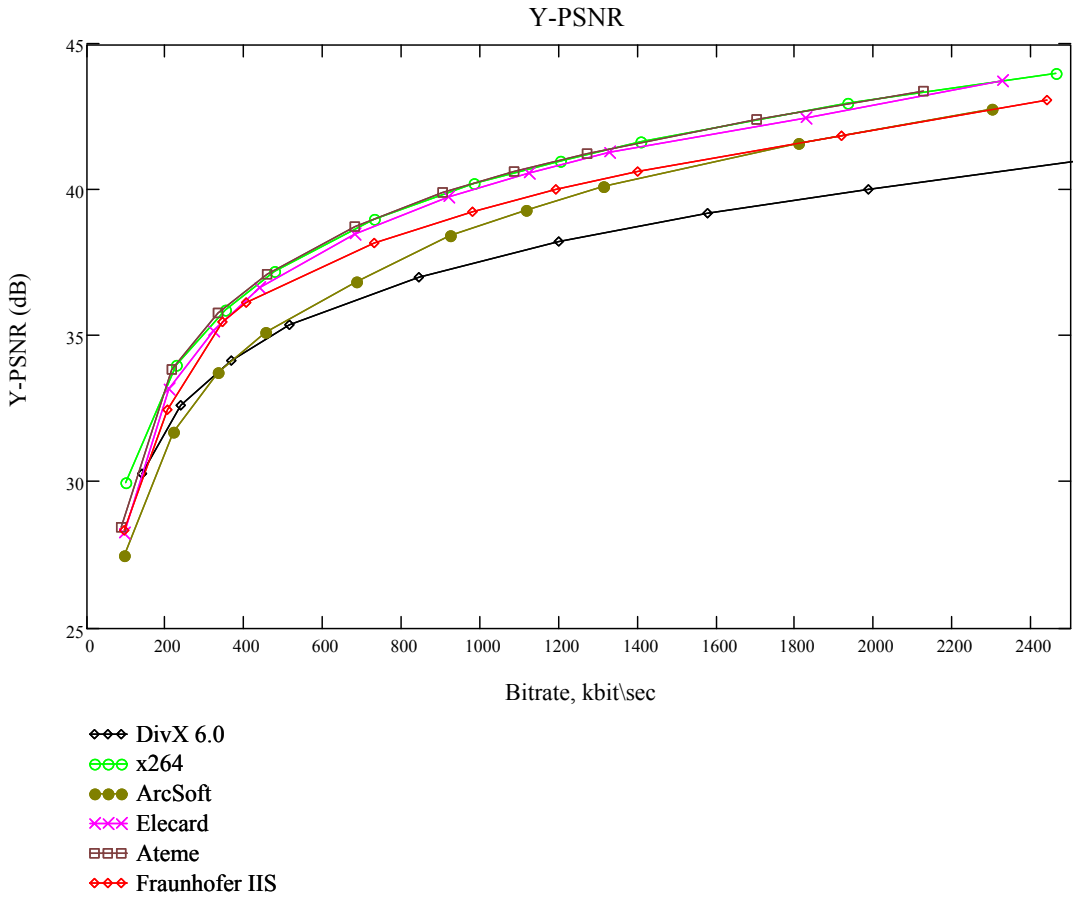
Below are some Y-PSNR/Bitrate and Delta Y-PSNR/Bitrate graphs.

PSNR (peak-to-peak signal-to-noise ratio) is a classic metric for video quality measurement. For two pictures $x_{i,j}$ and $y_{i,j}$ metric's value is defined by the following formula:

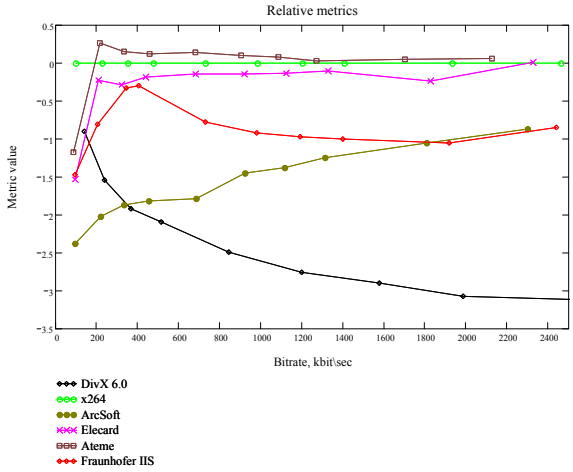
$$d(x,y) = 10 \times \log_{10} \frac{255^2 \times n^2}{\sum_{i=1,j=1}^{n,n} (x_{ij} - y_{ij})^2}$$

Despite this metric does not generally reflect human perception of pictures' distortion, during last decades it is commonly used as main criterion for video codecs' performance evaluation.

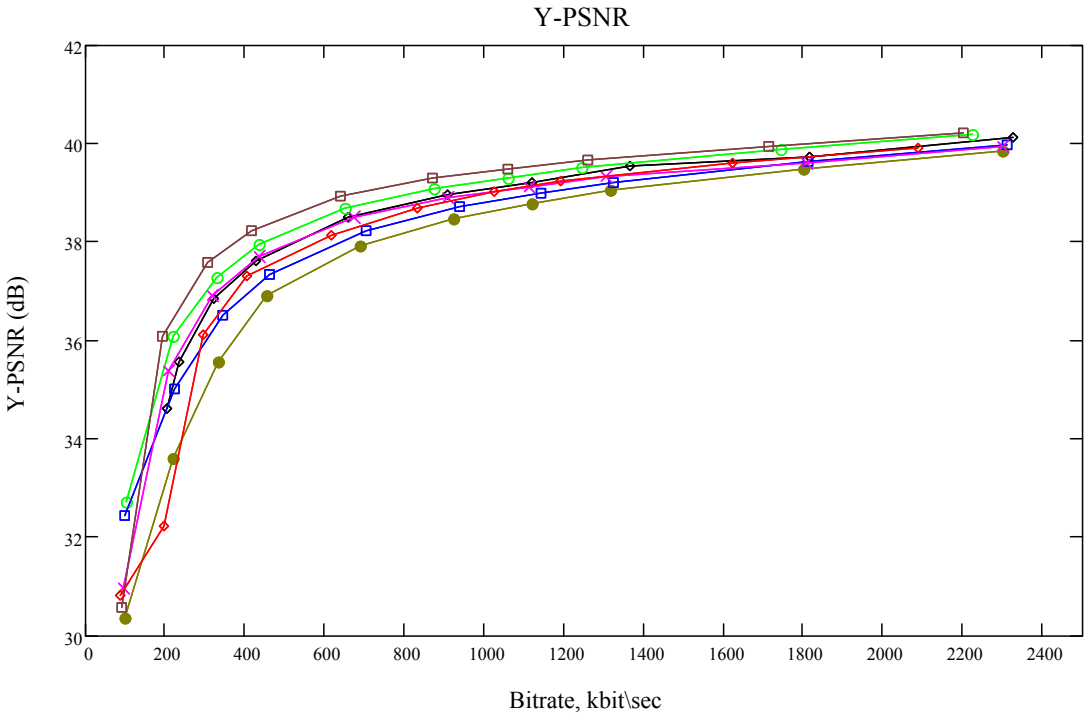
Delta Y-PSNR (or Relative Y-PSNR) are comparative PSNR graphs, where x264 codec was chosen as reference. For every measurement of each codec the difference between this measurement and PSNR value of reference codec with the same bitrate is plotted on these graphs. If corresponding reference value is absent then it is linearly interpolated.



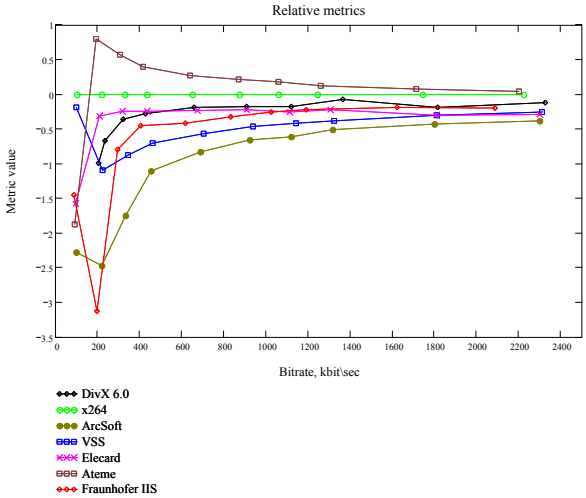
Picture 16. Y-PSNR. Sequence “foreman”



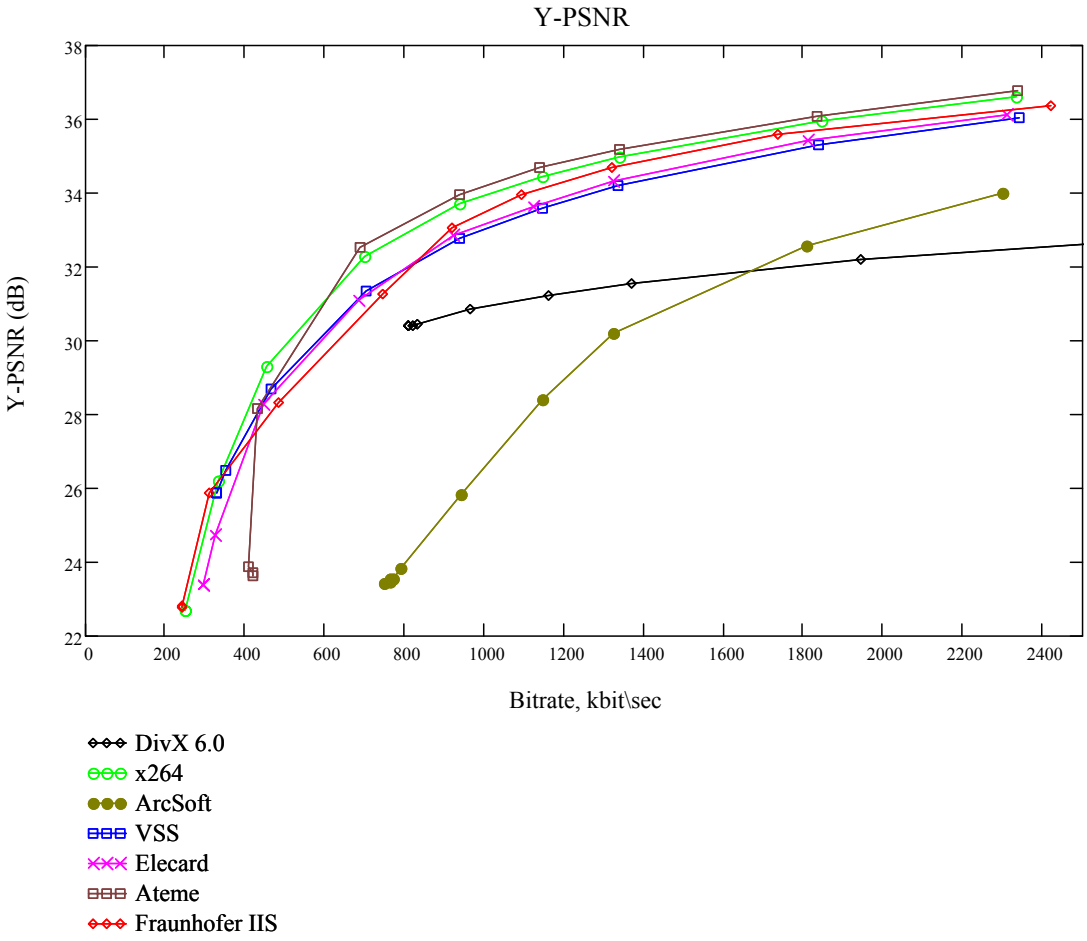
Picture 17. Delta Y-PSNR. Sequence “foreman”



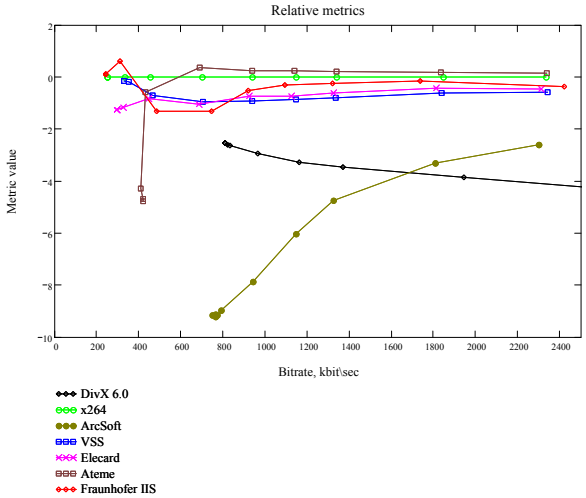
Picture 18. Y-PSNR. Sequence “susi”



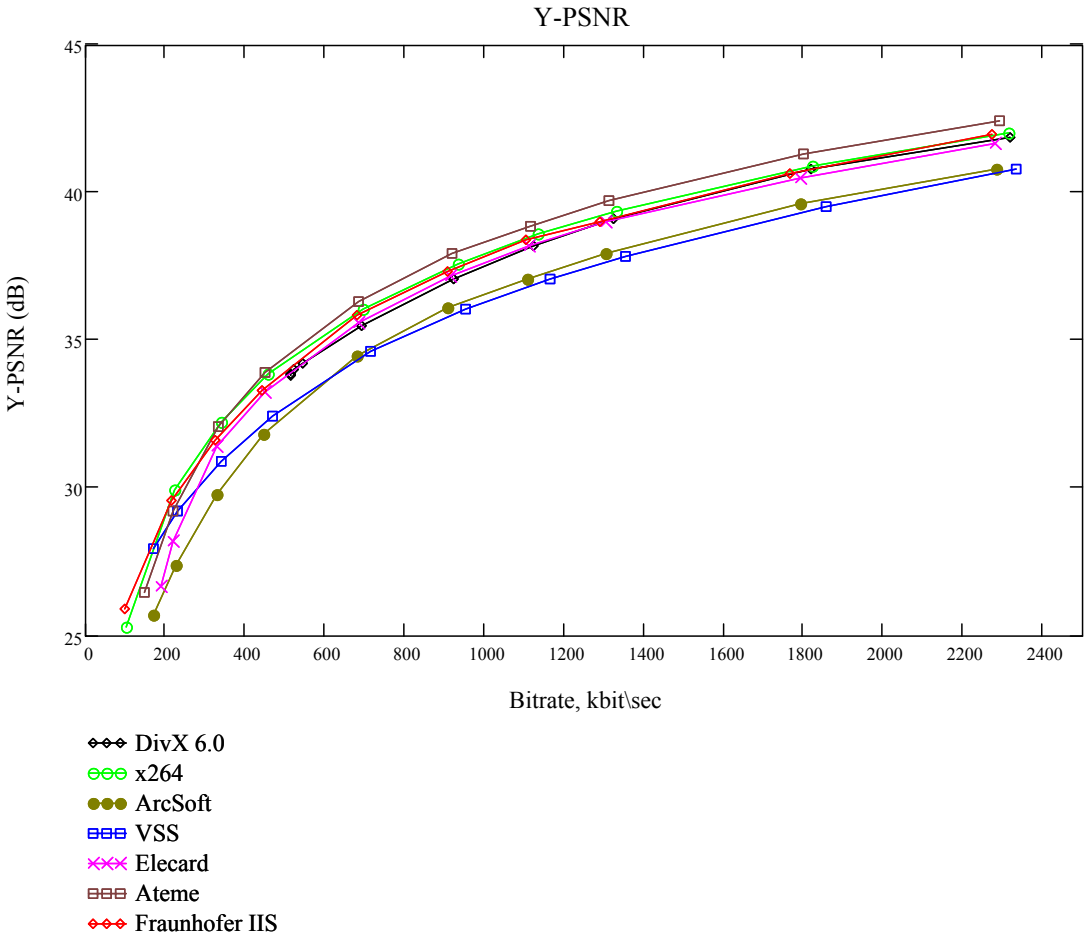
Picture 19. Delta Y-PSNR. Sequence “susi”



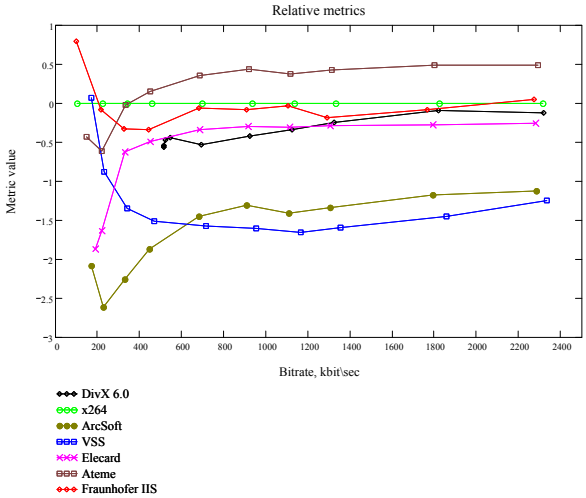
Picture 20. Y-PSNR. Sequence “bbc”



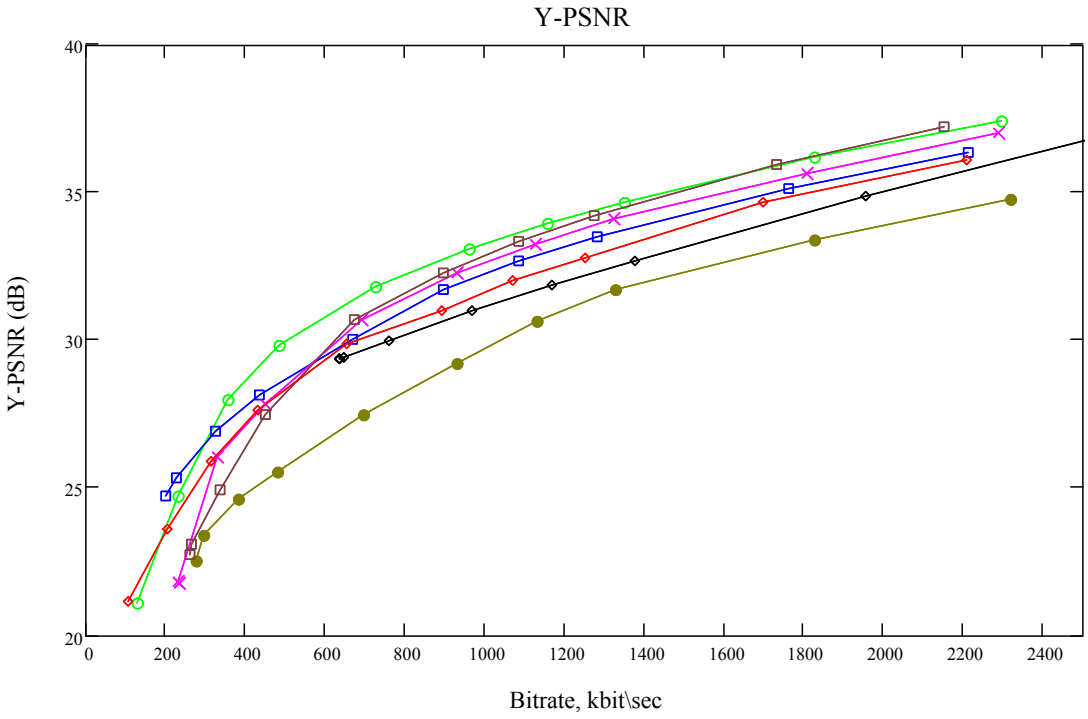
Picture 21. Delta Y-PSNR. Sequence “bbc”



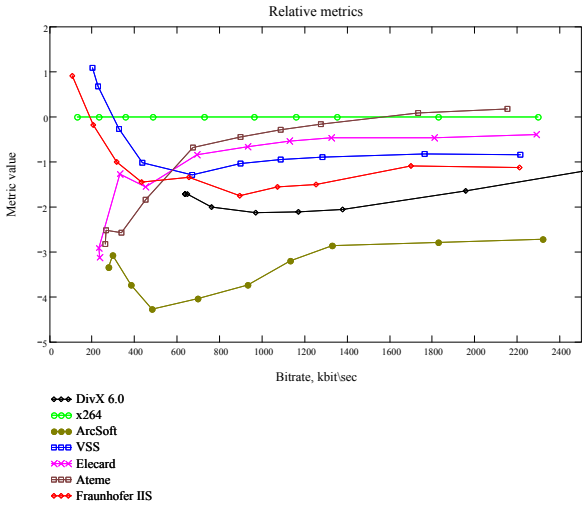
Picture 22. Y-PSNR. Sequence “battle”



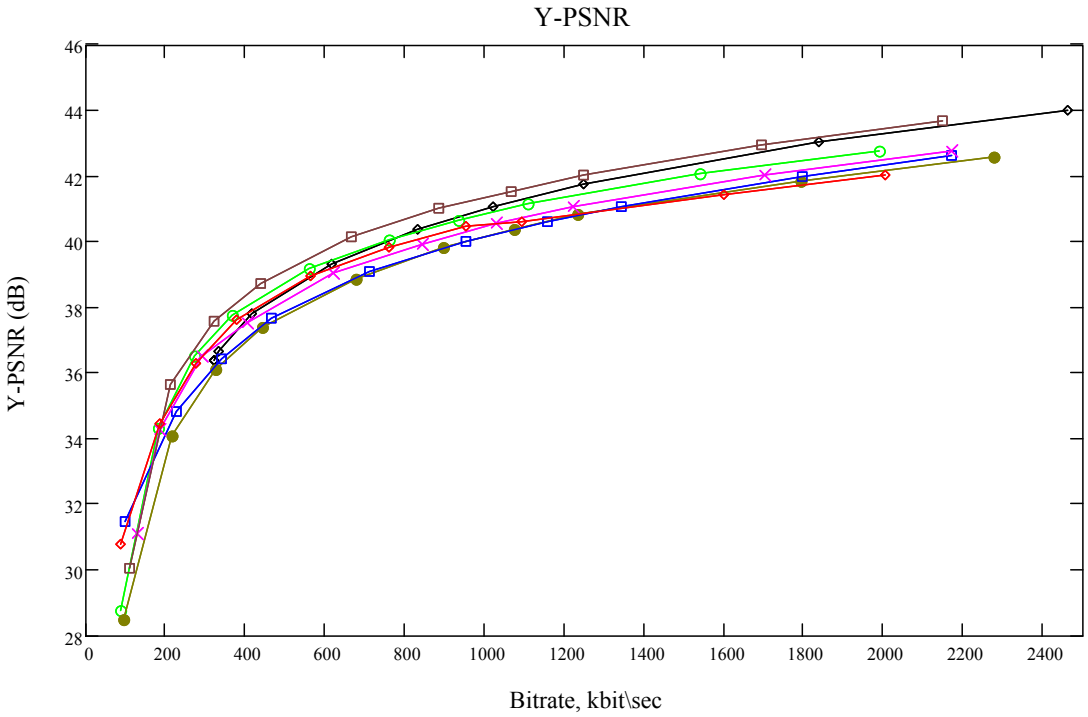
Picture 23. Delta Y-PSNR. Sequence “battle”



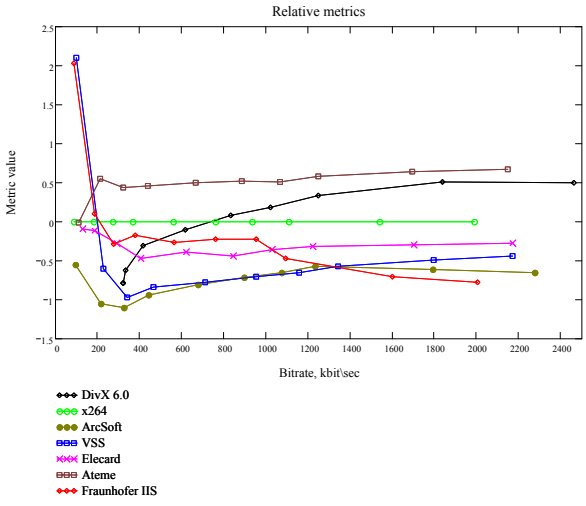
Picture 24. Y-PSNR. Sequence “simpsons”



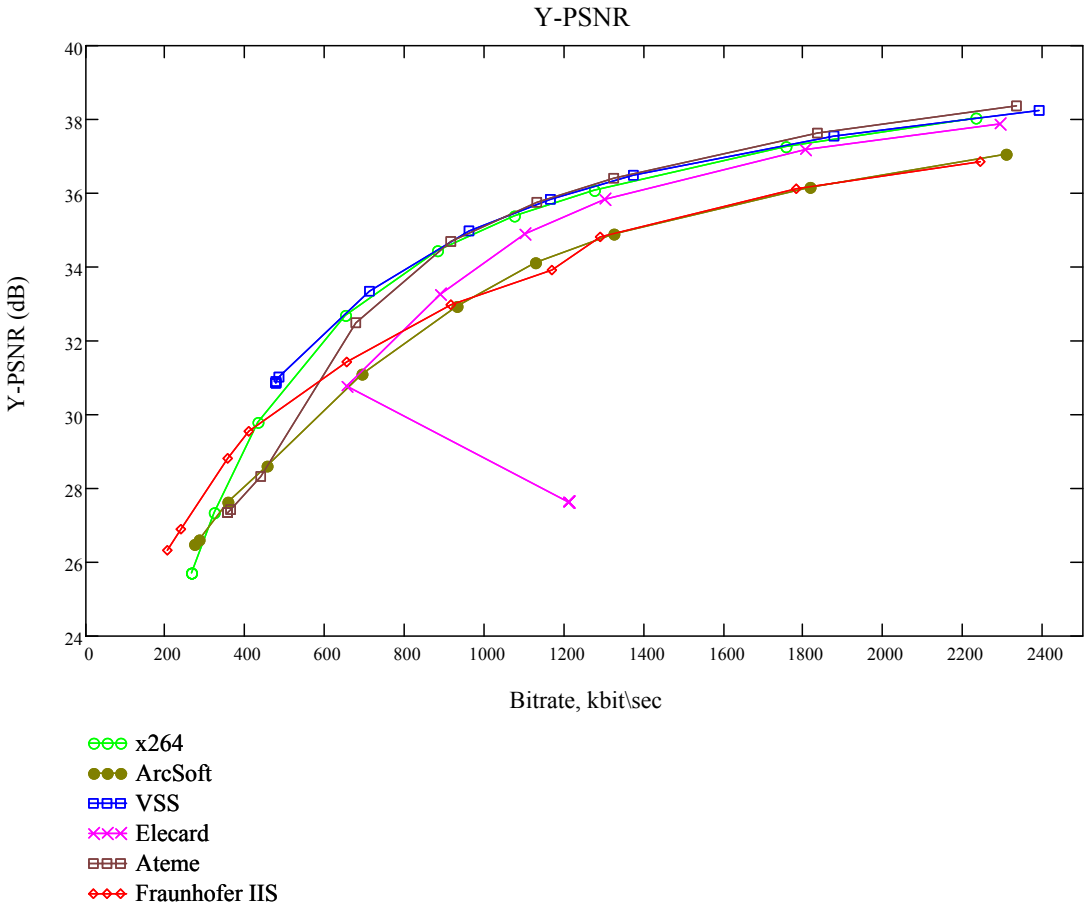
Picture 25. Delta Y-PSNR. Sequence “simpsons”



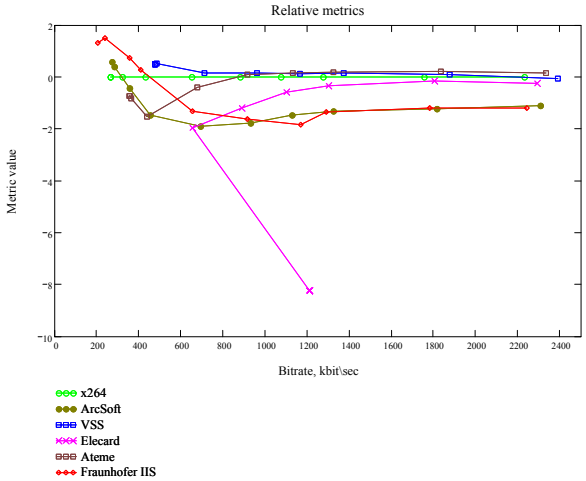
Picture 26. Y-PSNR. Sequence “matrix”



Picture 27. Delta Y-PSNR. Sequence “matrix”



Picture 28. Y-PSNR. Sequence "concert"



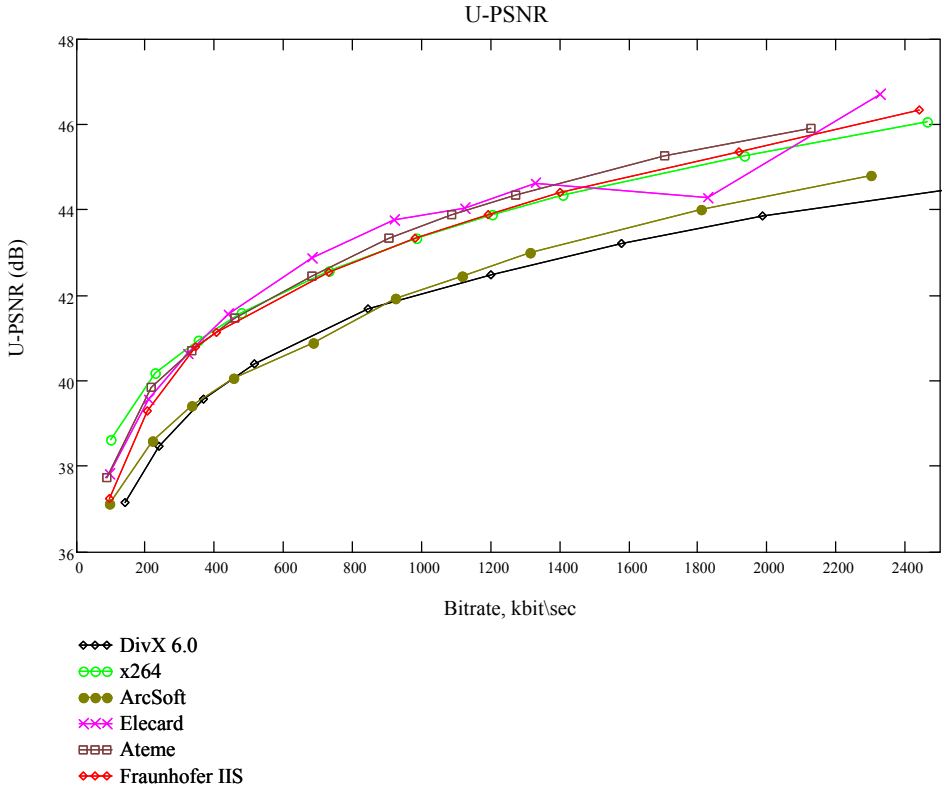
Picture 29. Delta Y-PSNR. Sequence "concert"

Conclusions:

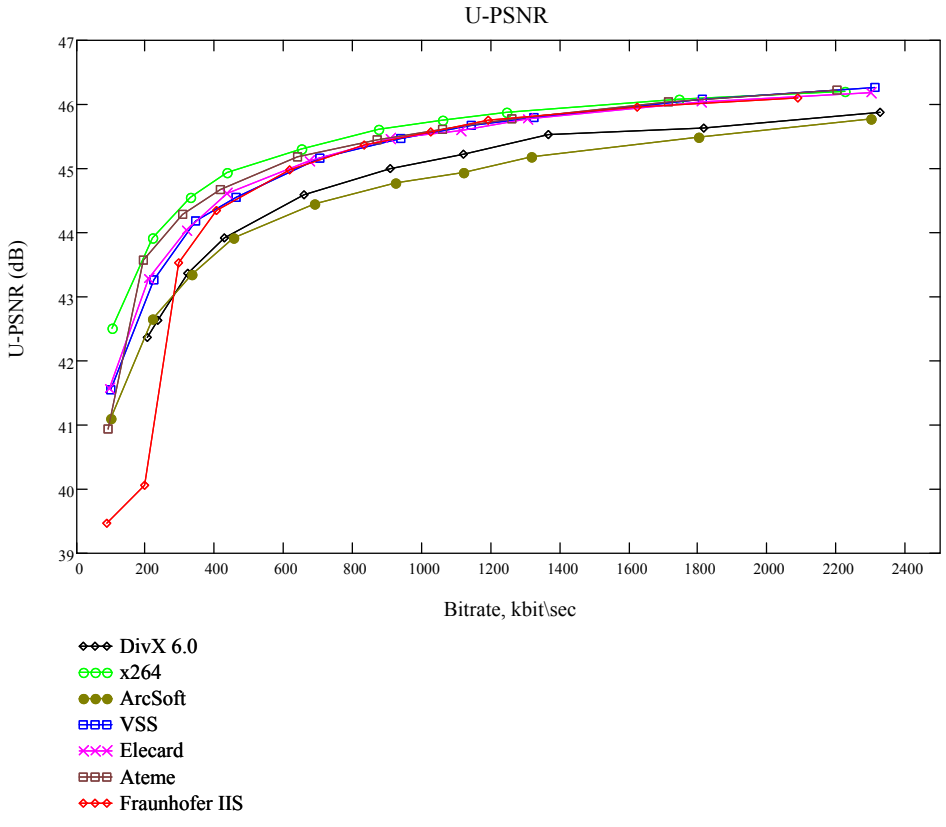
- Despite significant compression quality improvement in DivX, it is not a leader. And even more, if DivX is not far from leaders on “susi”, “battle” and “matrix” sequences, on the rest sequences (“foreman”, “bbc”, “simpsons”) it is most likely to be related to trailing codecs. New version of DivX codec shows worse results on “bbc” sequence than its old version 5.2.1; it was not able compress “concert” sequence at all.
- There is no obvious leader among H.264 codecs. In most cases leaders are x264 and Ateme codecs.
- The bad performance of Ateme H.264 codec on low birates is due to a bug, which was fixed by developers after publication of this comparison.
- Elecard codec could not manage with low bitrates of “concert” sequence. It is clearly seen from Y-PSNR graph for this codec.

U-PSNR, V-PSNR

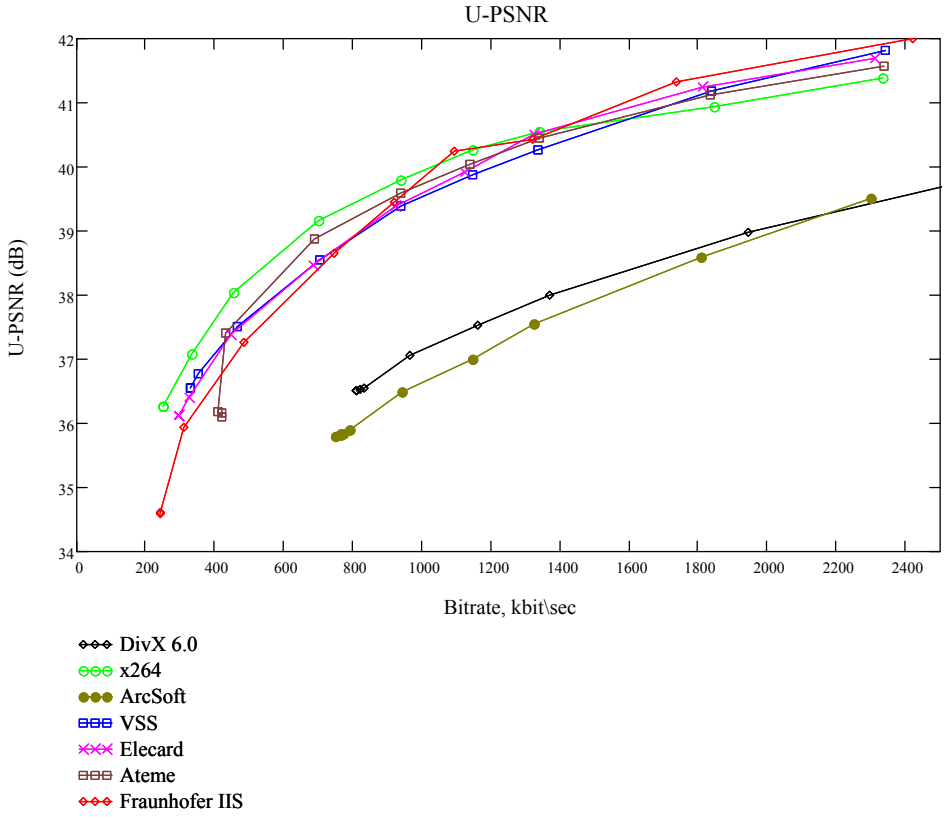
Besides luminance component (Y-plane) sometimes it is interesting to look at codecs’ behavior on color components (U and V planes).



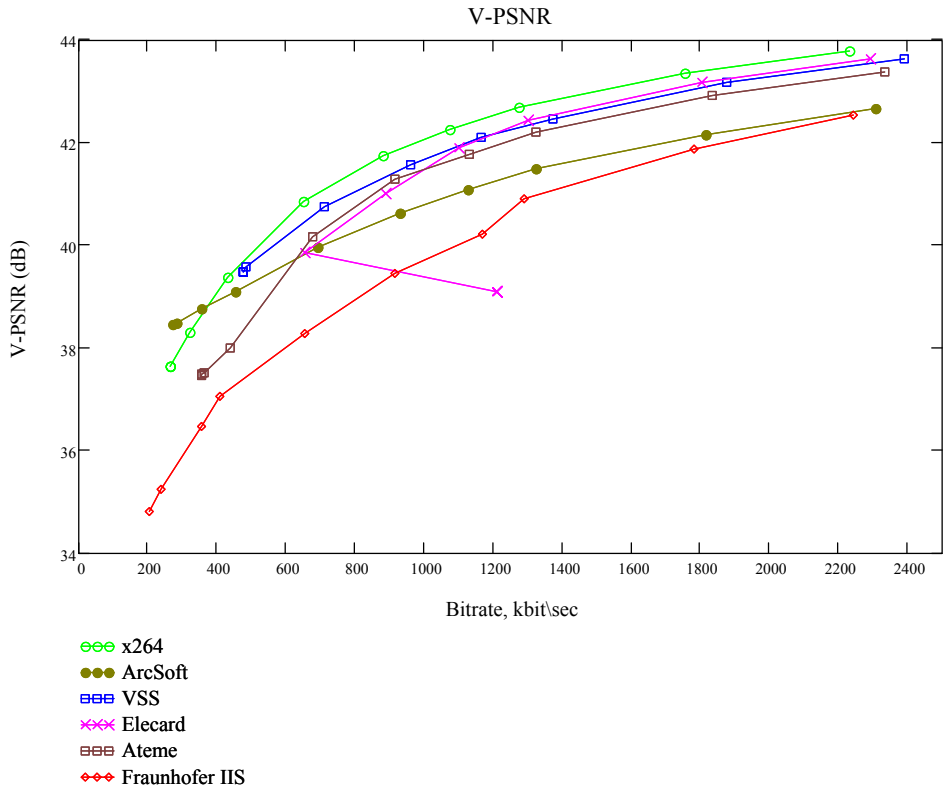
Picture 30. U-PSNR. Preset “Best quality”. Sequence “foreman”



Picture 31. U-PSNR. Preset "Best quality". Sequence "susi"



Picture 32. U-PSNR. Preset "Best quality". Sequence "bbc"



Picture 33. V-PSNR. Preset "Best quality". Sequence "concert"

Conclusions:

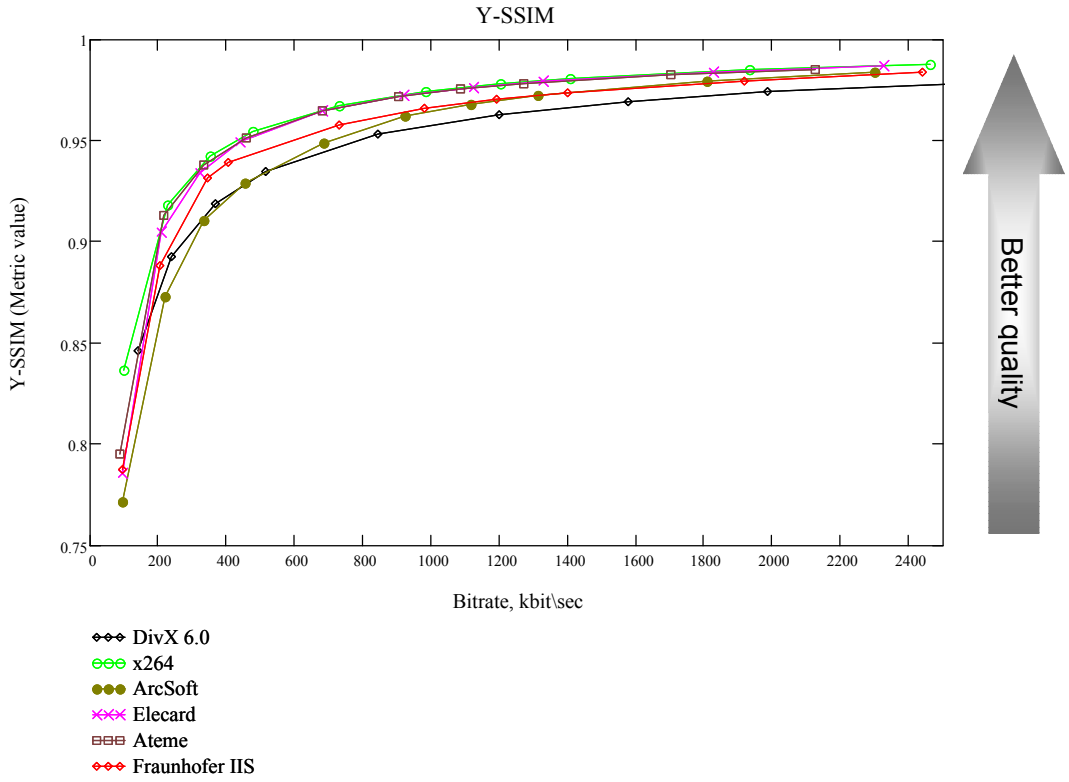
- In color planes x264 codec shows slightly better result than Ateme codec.
- Elecard codec has some problems with bitrate control on "foreman" sequence. It is shown as "fall" of its curve on UV-PSNR graphs. But this codec shows the best results on low bitrates on this sequence.

SSIM, VQM

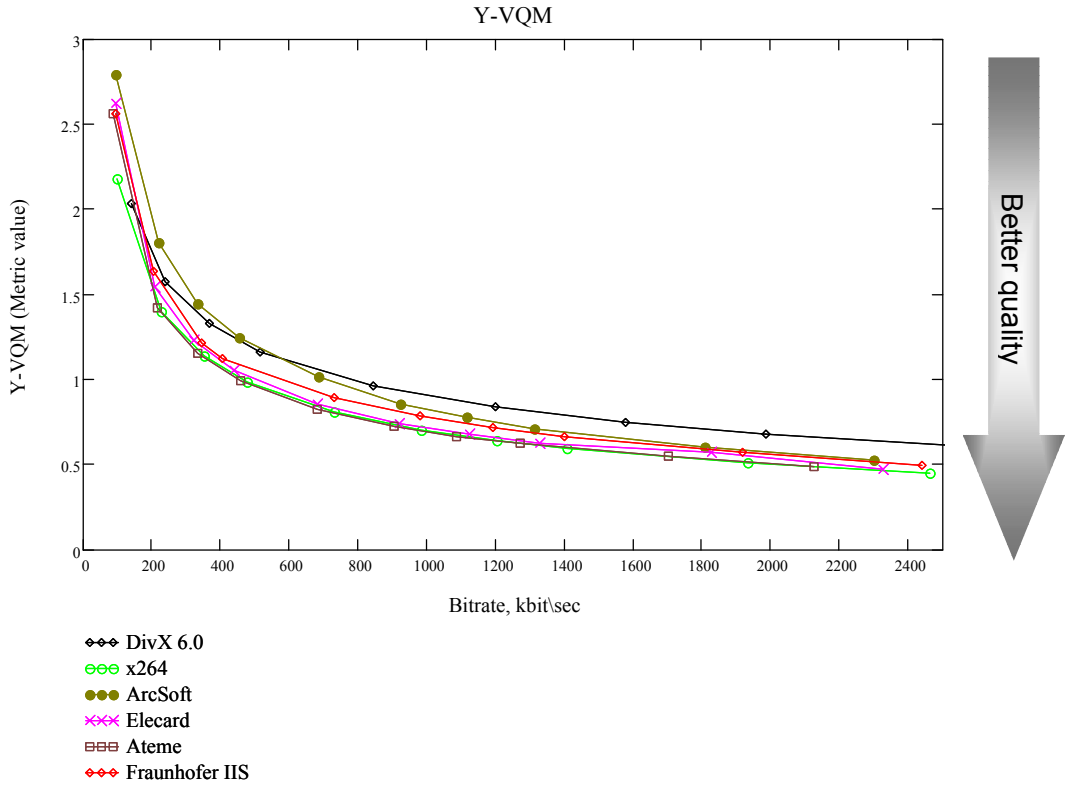
Several metrics recently appeared which aspire to be better approximations of human perception. The most popular among them are SSIM and VQM.

We have used them in our comparison, but they have presented no big differences from PSNR metric.

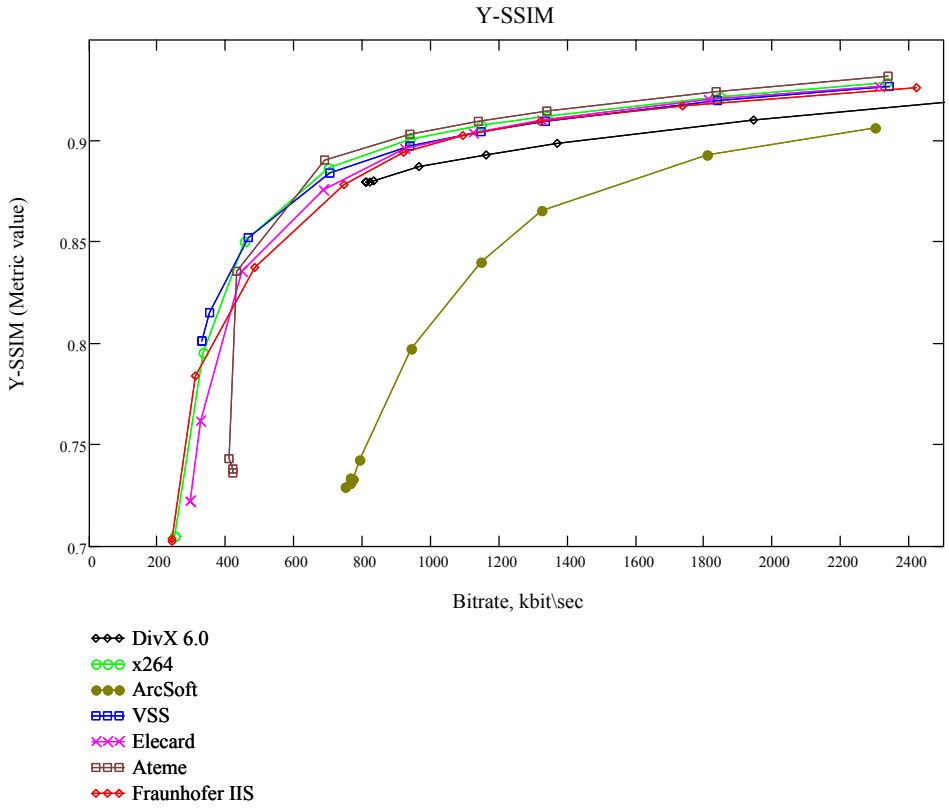
Below are some graphs of these metrics for different sequences.



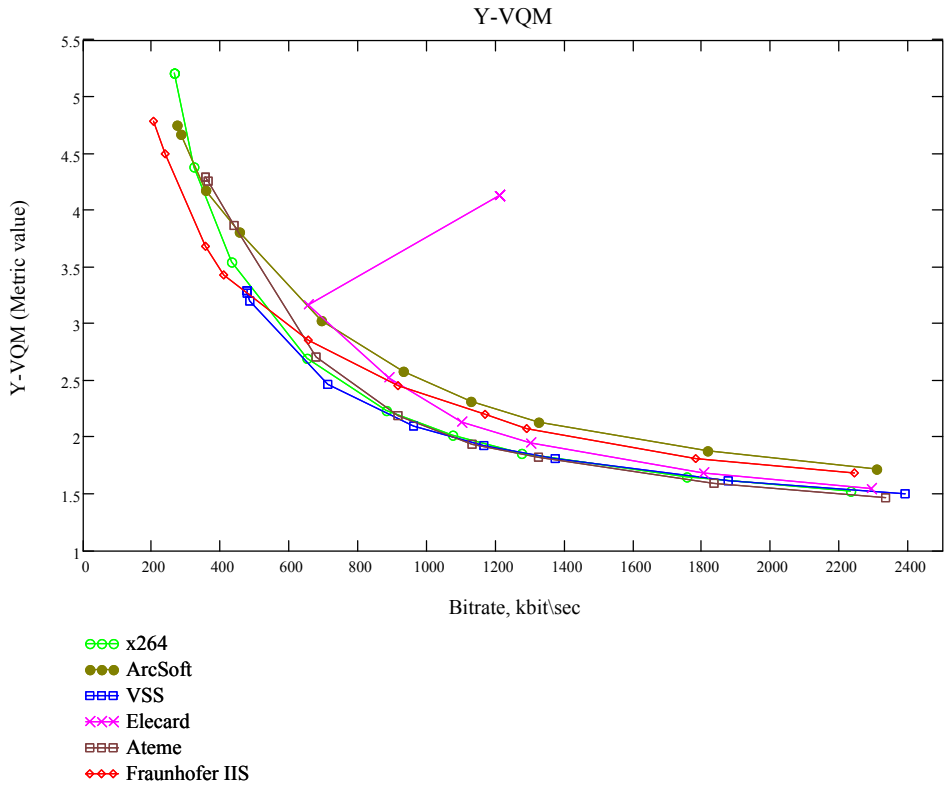
Picture 34. Y-SSIM measure. Preset "Best quality". Sequence "foreman"



Picture 35. Y-VQM measure. Preset "Best quality". Sequence "foreman"



Picture 36. Y-SSIM measure. Preset "Best quality". Sequence "bbc"

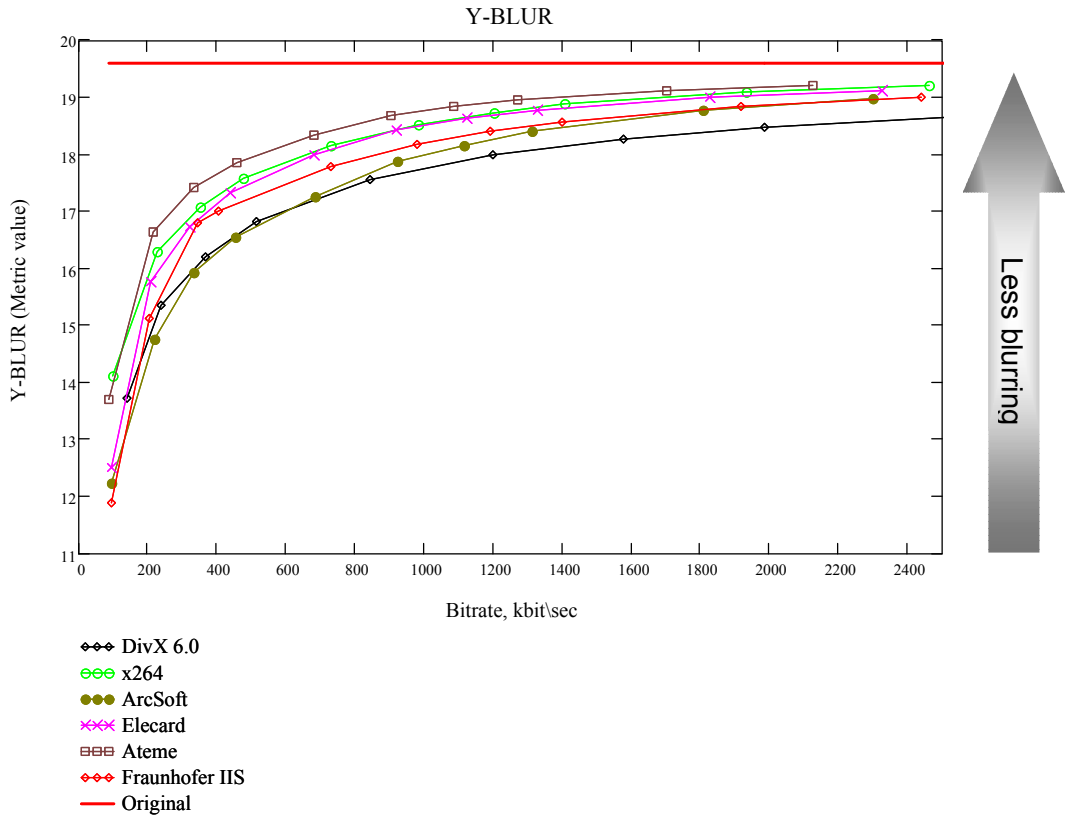


Picture 37. Y-VQM measure. Preset "Best quality". Sequence "concert"

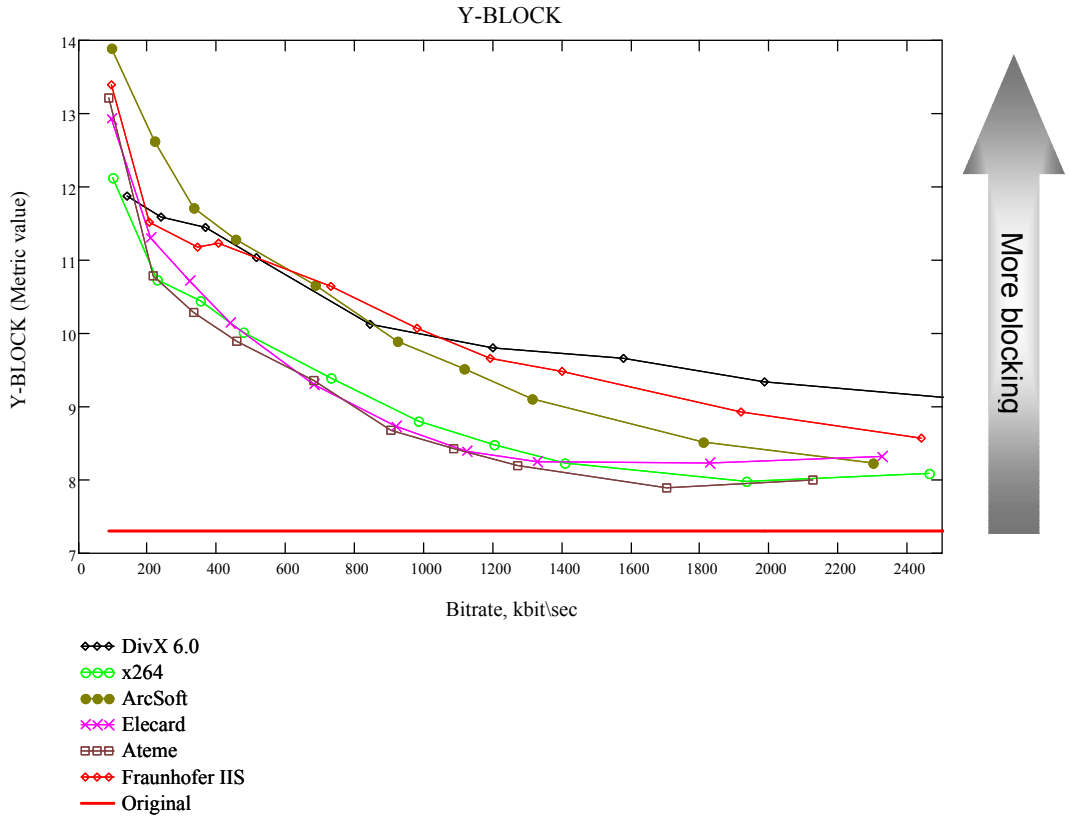
Blurring measure, blocking measure

During some time our laboratory is developing metrics, which allow measuring blocking and blurring of frames in video sequences. Work on these metrics is in progress, therefore we show only several graphs with them.

Red horizontal line is a blocking measure (or blurring measure) of source sequence. For blurring measure: the more is its value, the lesser is the blurring of frame; for blocking measure: the more is its value, the bigger is blocking of frame.



Picture 38. Y-Blurring measure. Preset “Best quality”. Sequence “foreman”

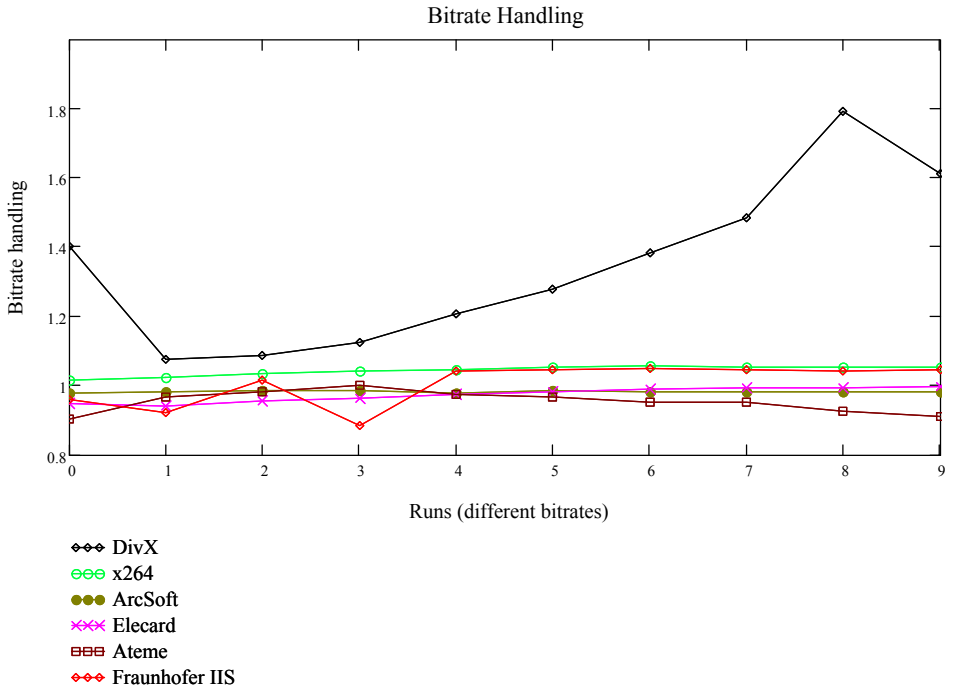


Picture 39. Y-Blocking measure. Preset "Best quality". Sequence "foreman"

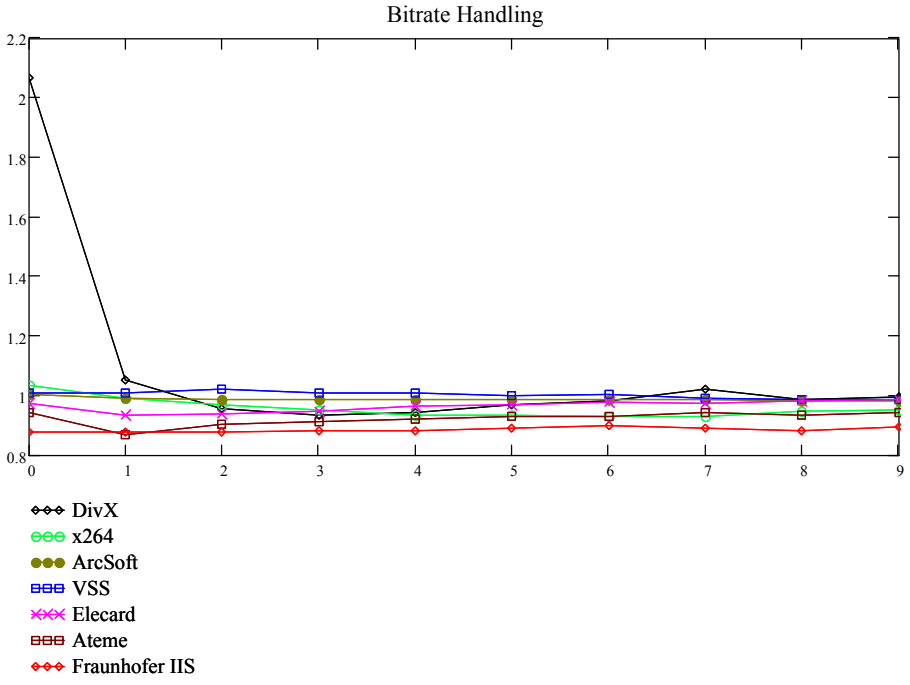
Graph type	Total number of graph	Inserted in this document
Average metric	196	24 (12%)

Bitrate handling

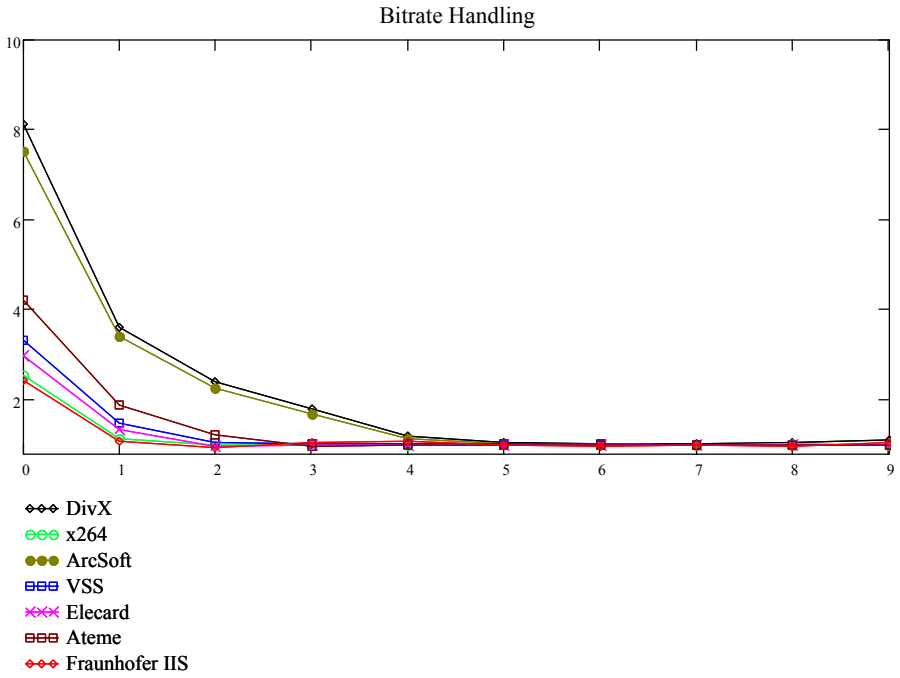
Consider graphs showing how accurately every codec kept requested bit rate. Grades on x-axis show target bitrates (point 0 is 100 kbps, point 9 – 2340 kbps). Grades on y-axis show how many times codec exceeded declared bit rate (real bit rate to requested bit rate ratio).



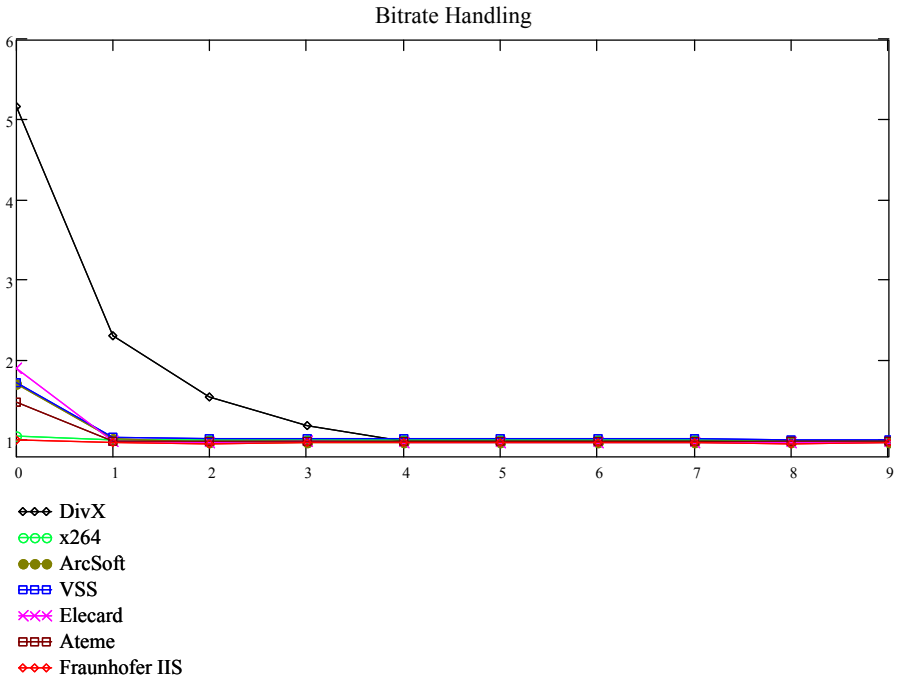
Picture 40. Bitrate handling. Preset "Best quality". Sequence "foreman"



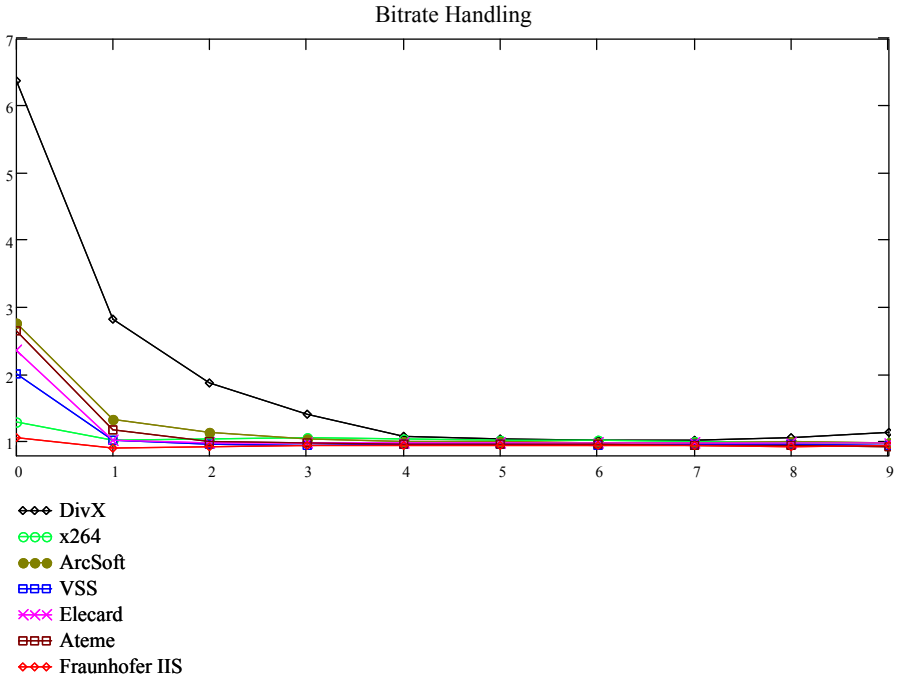
Picture 41. Bitrate handling. Preset "Best quality". Sequence "susi"



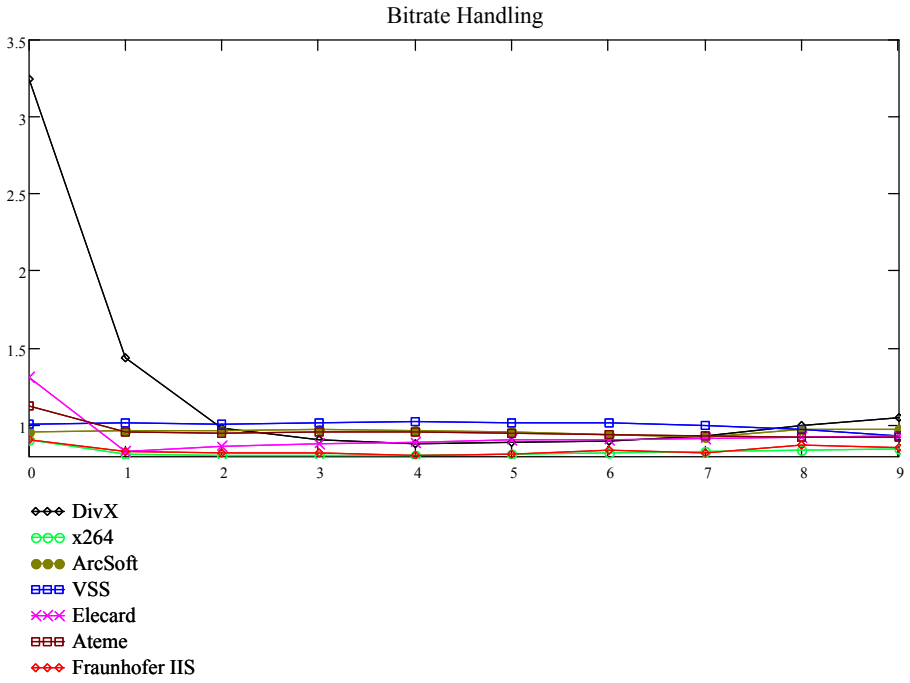
Picture 42. Bitrate handling. Preset “Best quality”. Sequence “bbc”



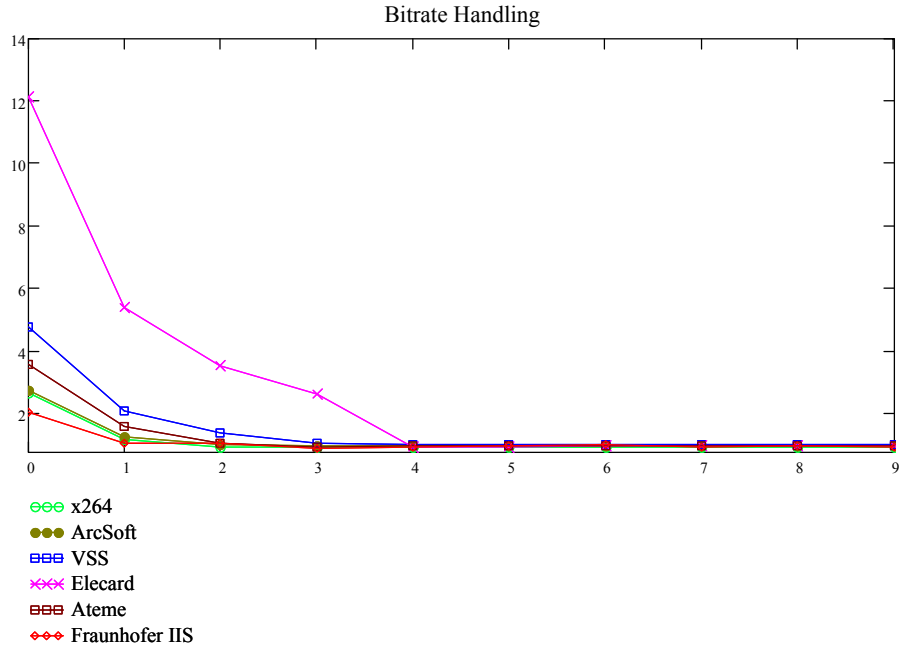
Picture 43. Bitrate handling. Preset “Best quality”. Sequence “battle”



Picture 44. Bitrate handling. Preset "Best quality". Sequence "simpsons"



Picture 45. Bitrate handling. Preset "Best quality". Sequence "matrix"



Picture 46. Bitrate handling. Preset "Best quality". Sequence "concert"

Conclusions:

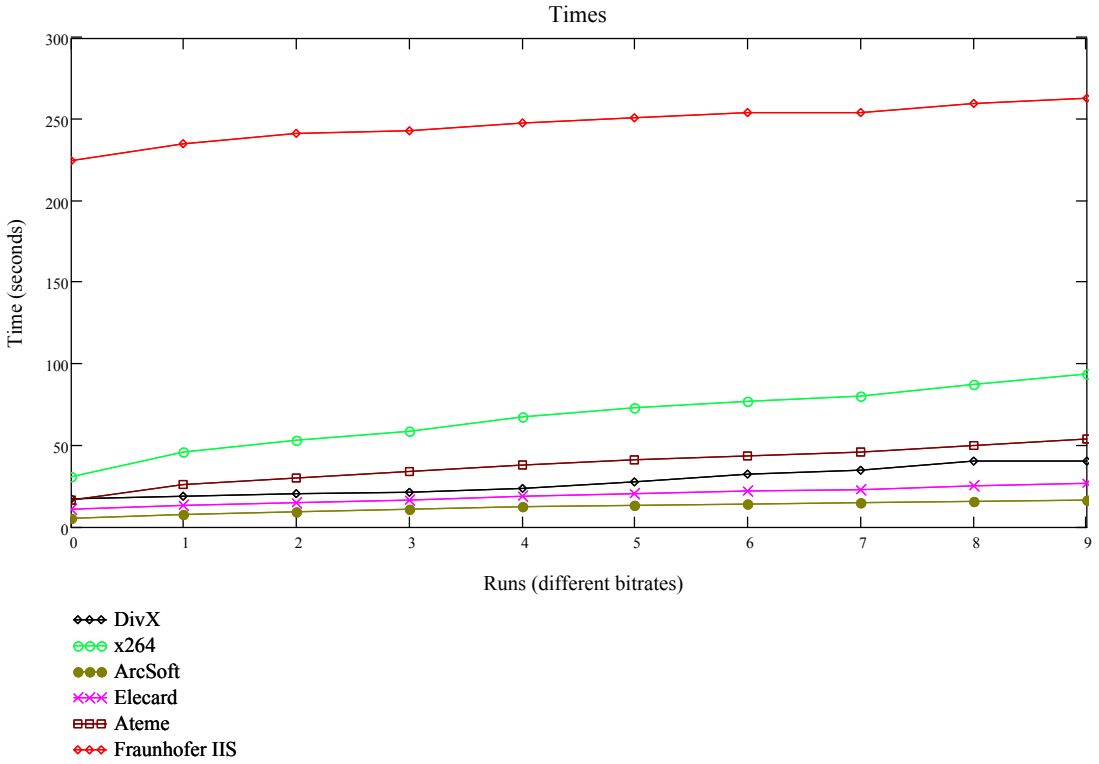
- DivX codec keeps low bit rates badly. Frequently it generates files which are several times bigger than the worst H.264 codec (by bit rate handling).
- "bbc" is the hardest sequence for bit rate handling for all codecs.
- Fraunhofer IIS codec keeps bit rate better than others.

Time

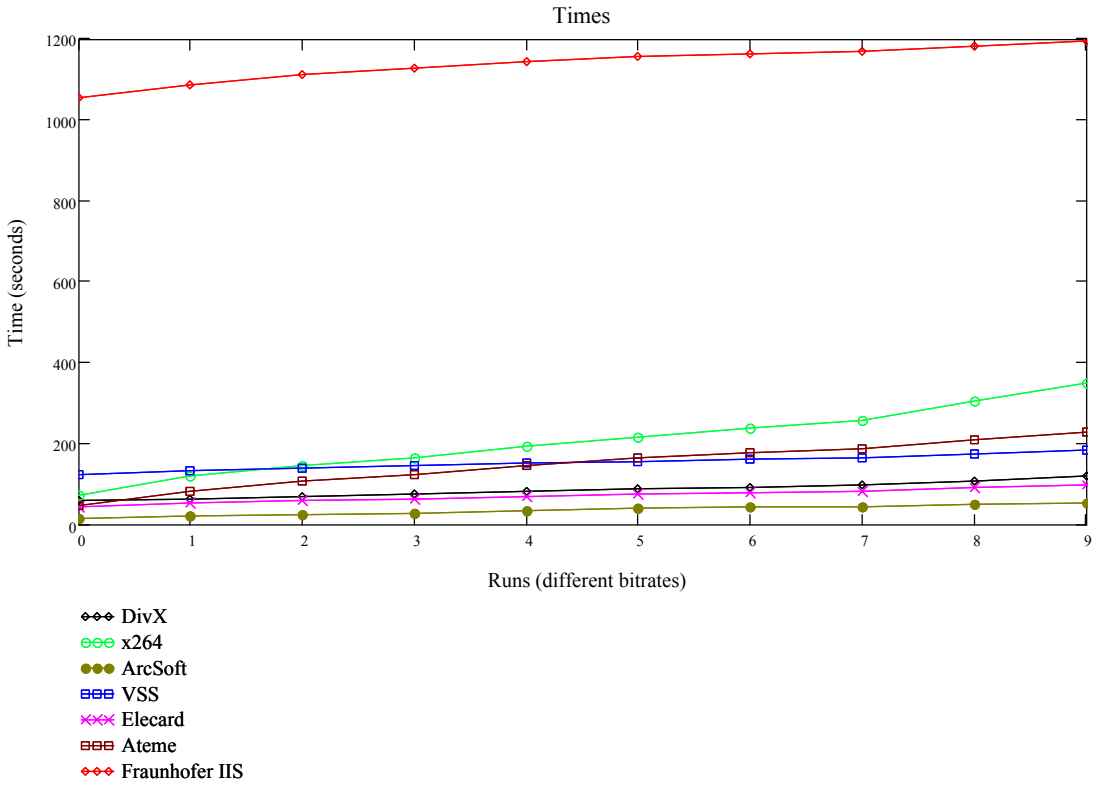
Now let's consider graphs showing encoding time for each sequence using "Best Quality" preset and average normalized time for this preset.

Average normalized time is obtained as follows. At first, for every sequence codec with largest compression time is found. Compression times of other codecs were divided by compression time of this codec. After this step every codec has value from 0 to 1 for each video sequence.

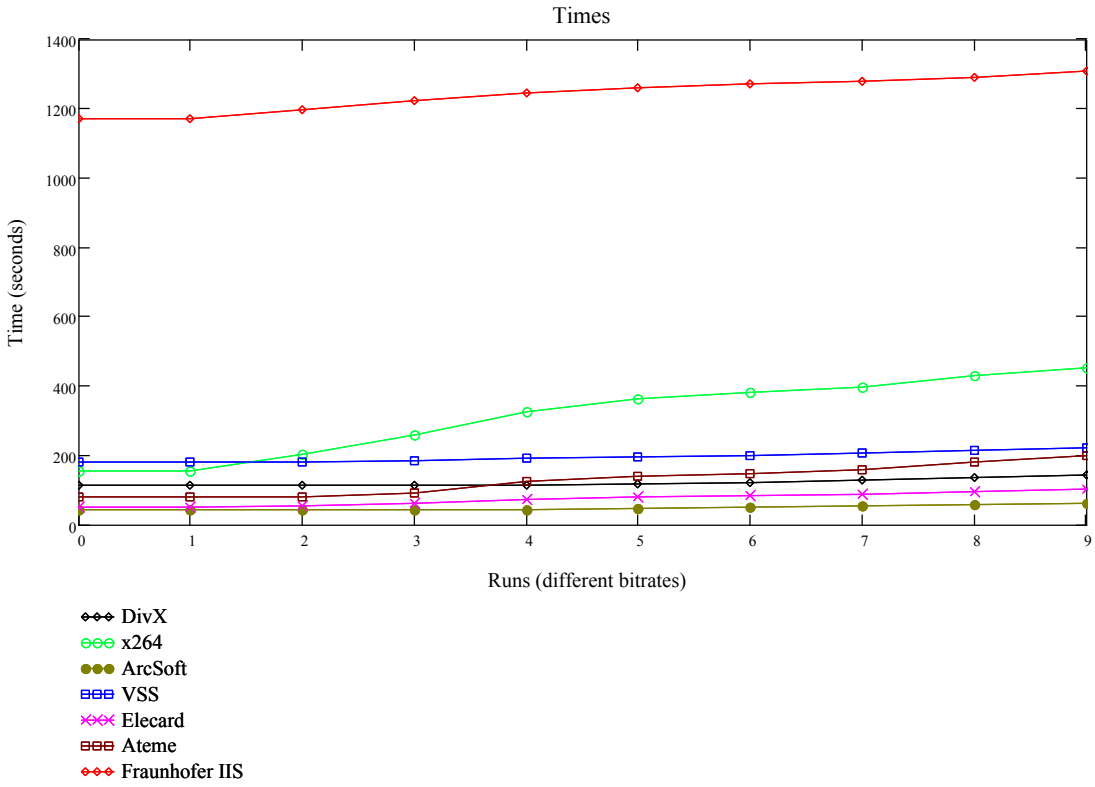
On the second step every codec gets evaluation calculated as arithmetic average of obtained values for all sequences. And these evaluations are shown on average normalized time graph.



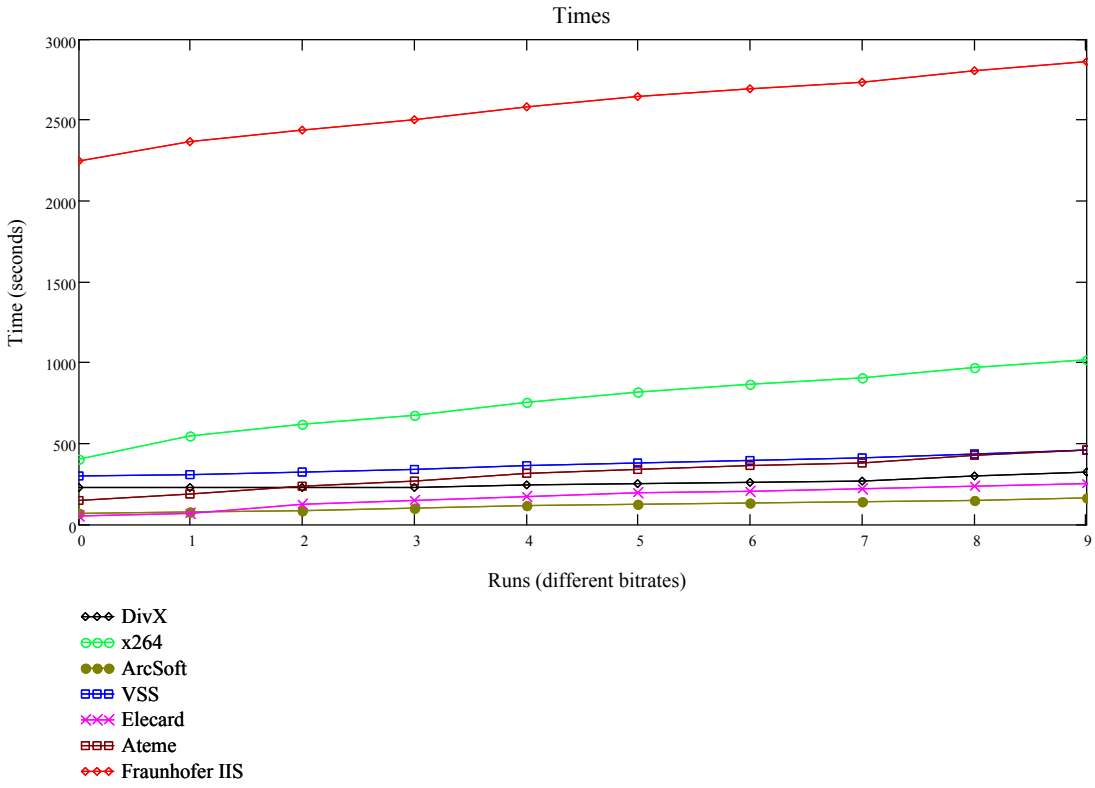
Picture 47. Encoding time. Preset "Best quality". Sequence "foreman"



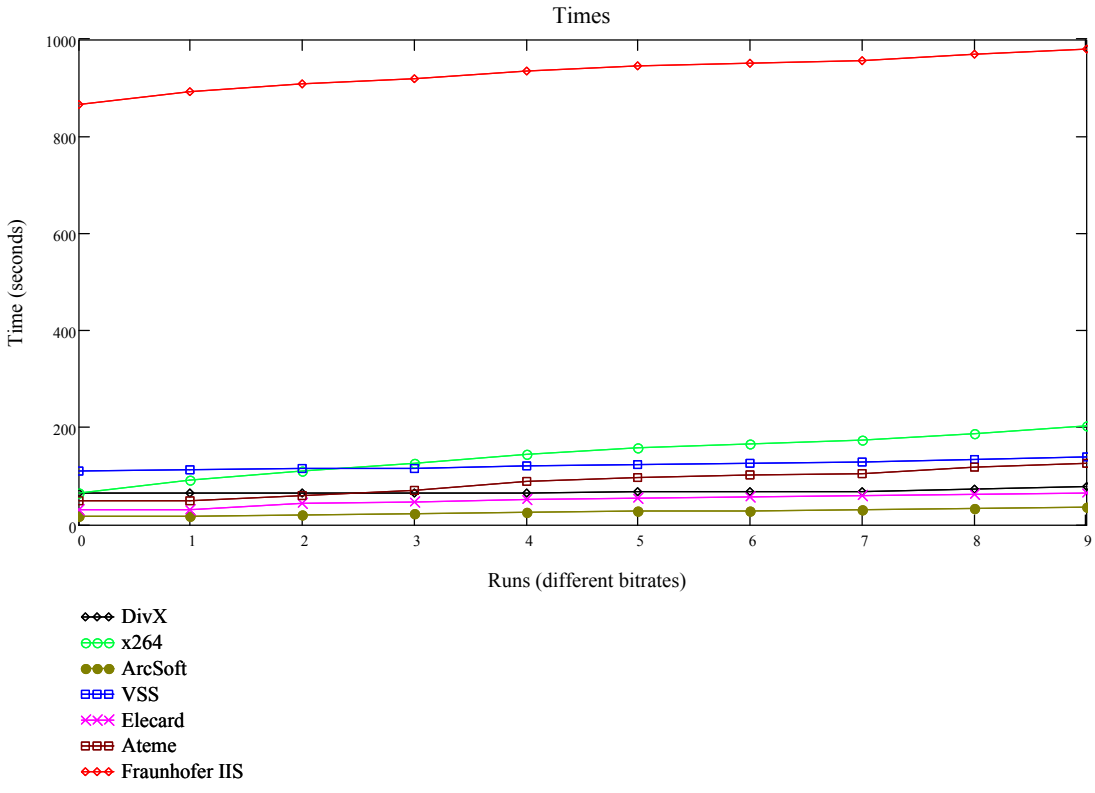
Picture 48. Encoding time. Preset “Best quality”. Sequence “susi”



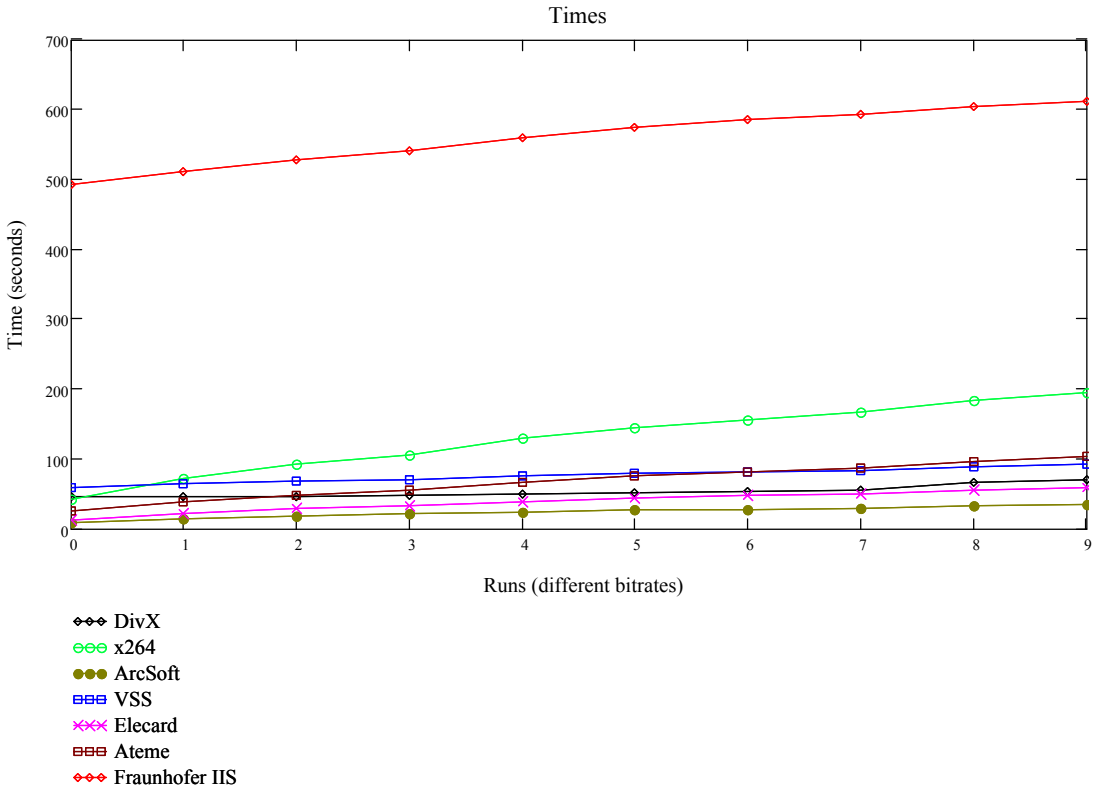
Picture 49. Encoding time. Preset “Best quality”. Sequence “bbc”



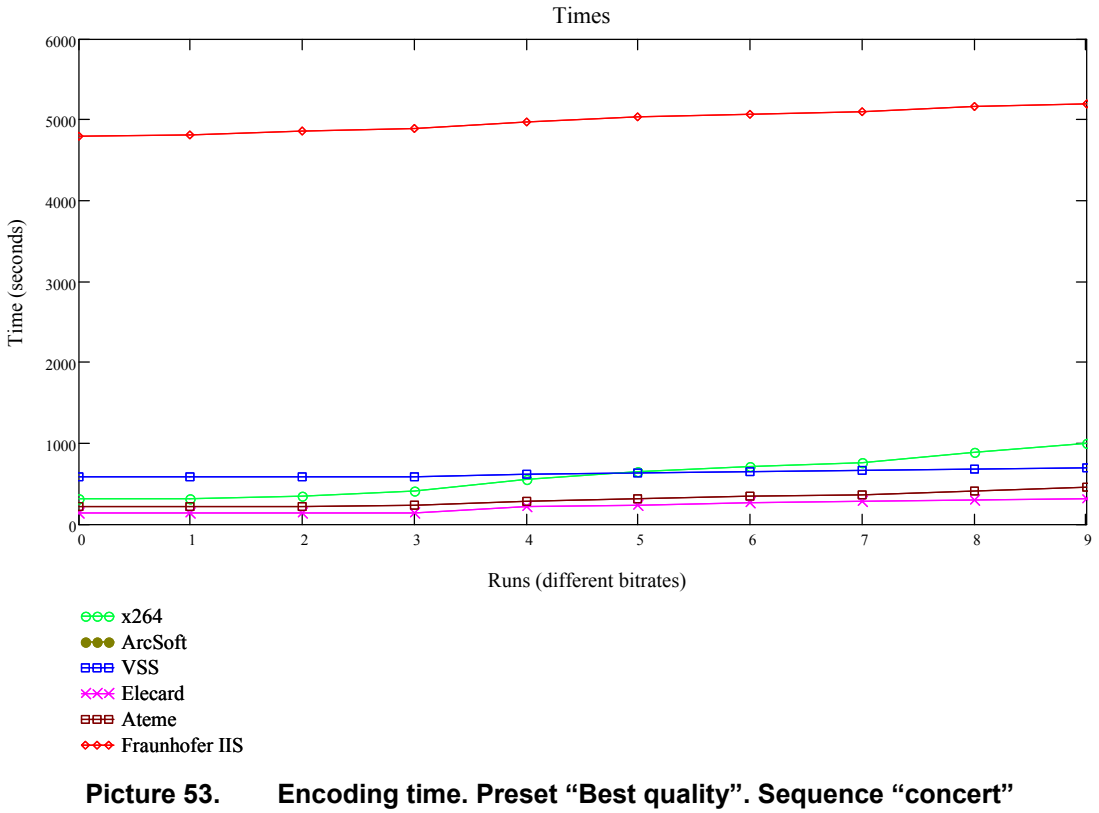
Picture 50. Encoding time. Preset “Best quality”. Sequence “battle”



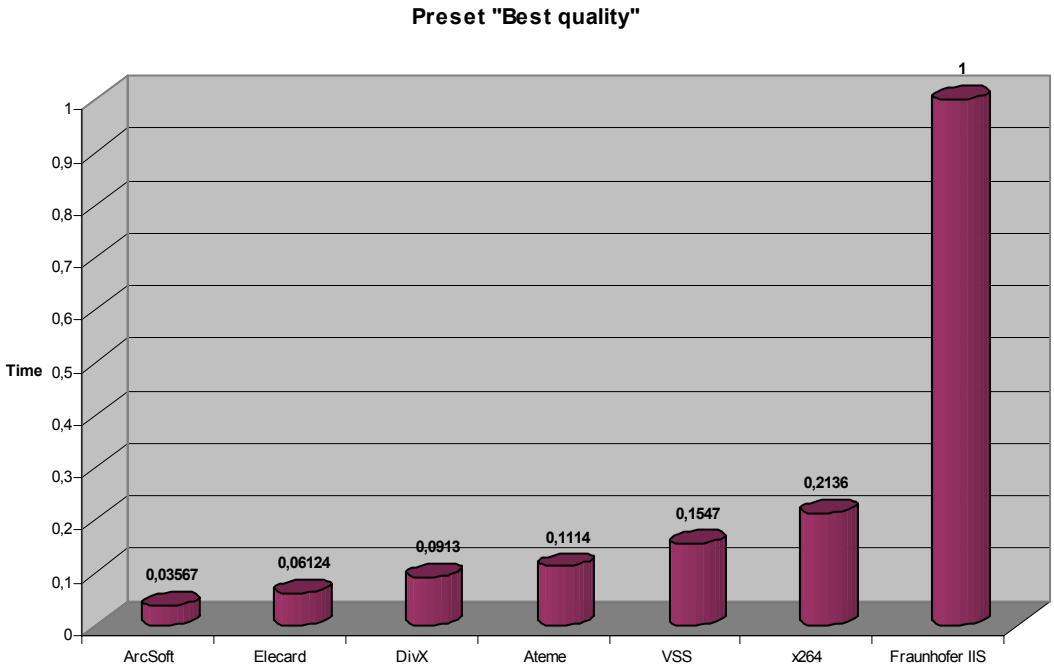
Picture 51. Encoding time. Preset “Best quality”. Sequence “simpsons”



Picture 52. Encoding time. Preset "Best quality". Sequence "matrix"



Picture 53. Encoding time. Preset "Best quality". Sequence "concert"



Picture 54. Normalized average encoding time. Preset "Best quality"

Conclusions:

- The fastest codec is ArcSoft codec, the slowest – Fraunhofer IIS codec.
- There is a noticeable work slowdown of Fraunhofer IIS and x264 codecs during bit rate increase.
- Fraunhofer IIS codec is approximately five times slower than all the rest codecs.

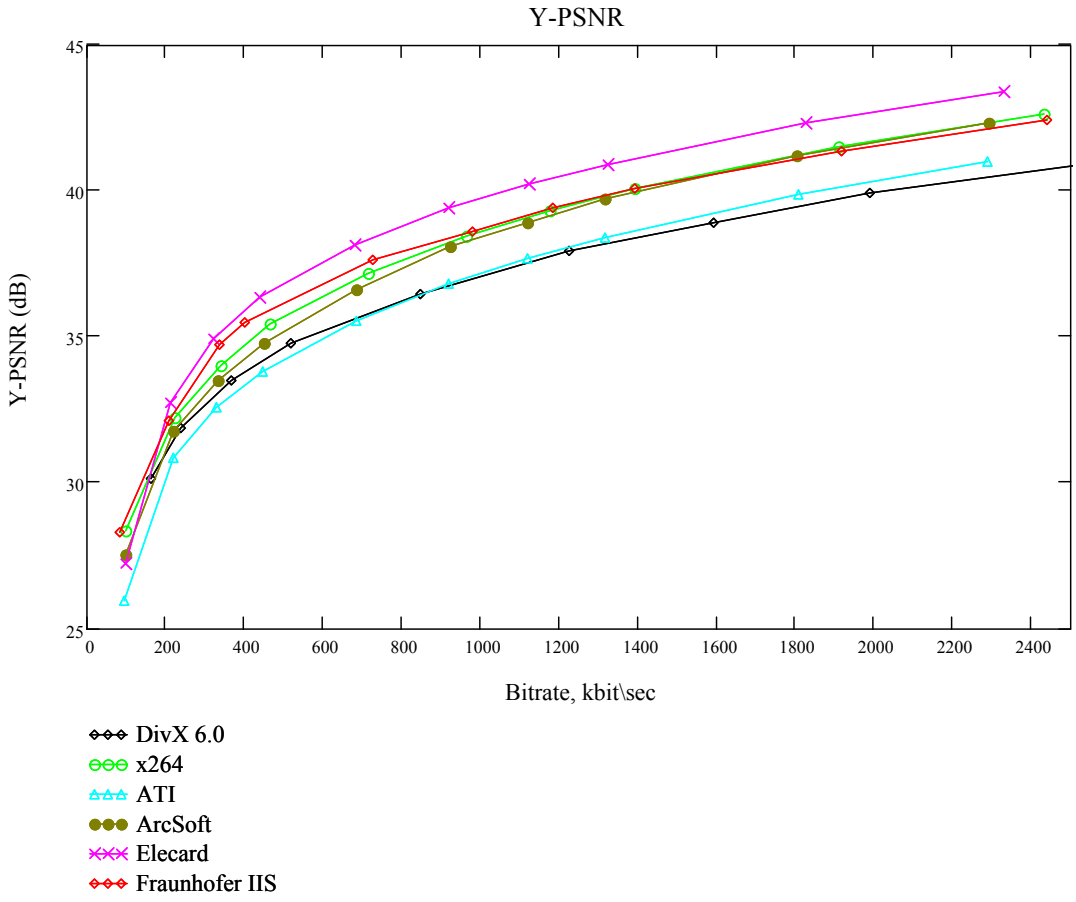
Preset “Best speed”

Main codecs’ objective in this mode is the fastest video compression with satisfactory quality. Therefore main graphs in this section are graphs that show compression speed.

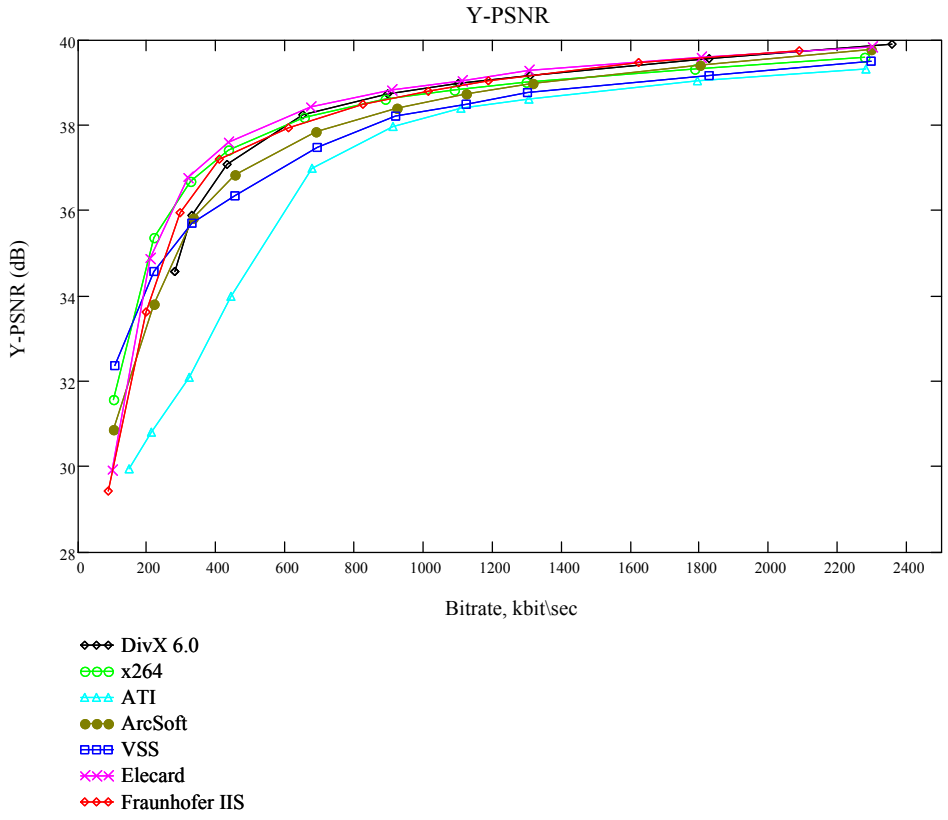
Y-PSNR

However, let’s consider Y-PSNR graphs at first.

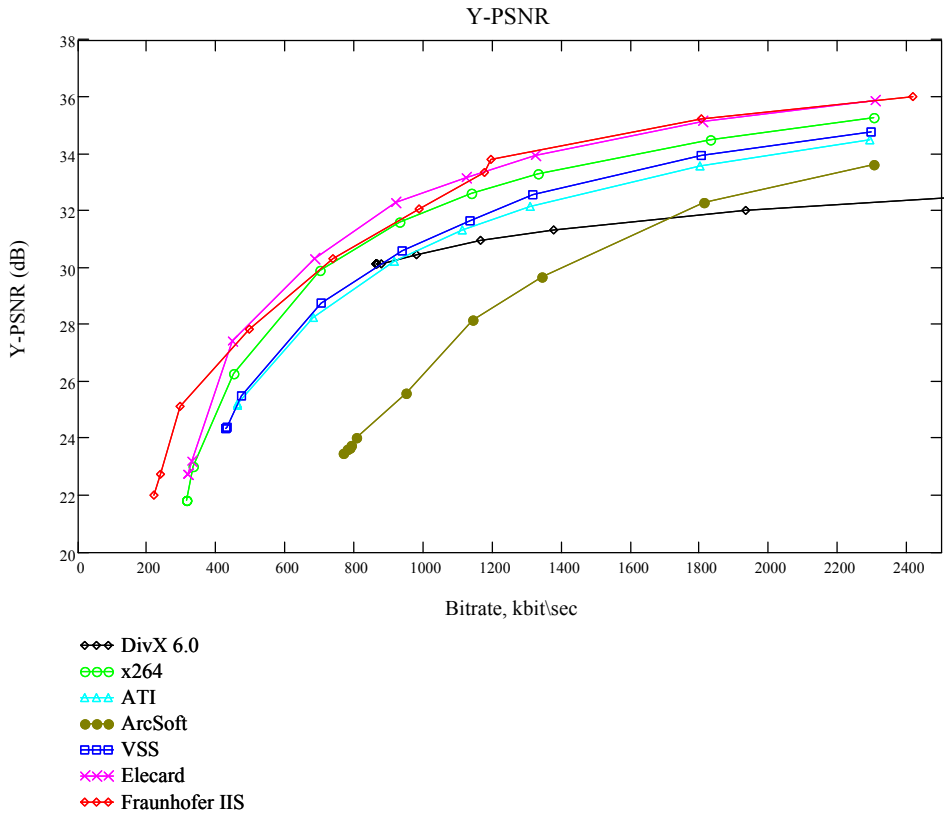
Graph type	Total number of graph	Inserted in this document
Average metric	**196**	**7 (3.5%)**



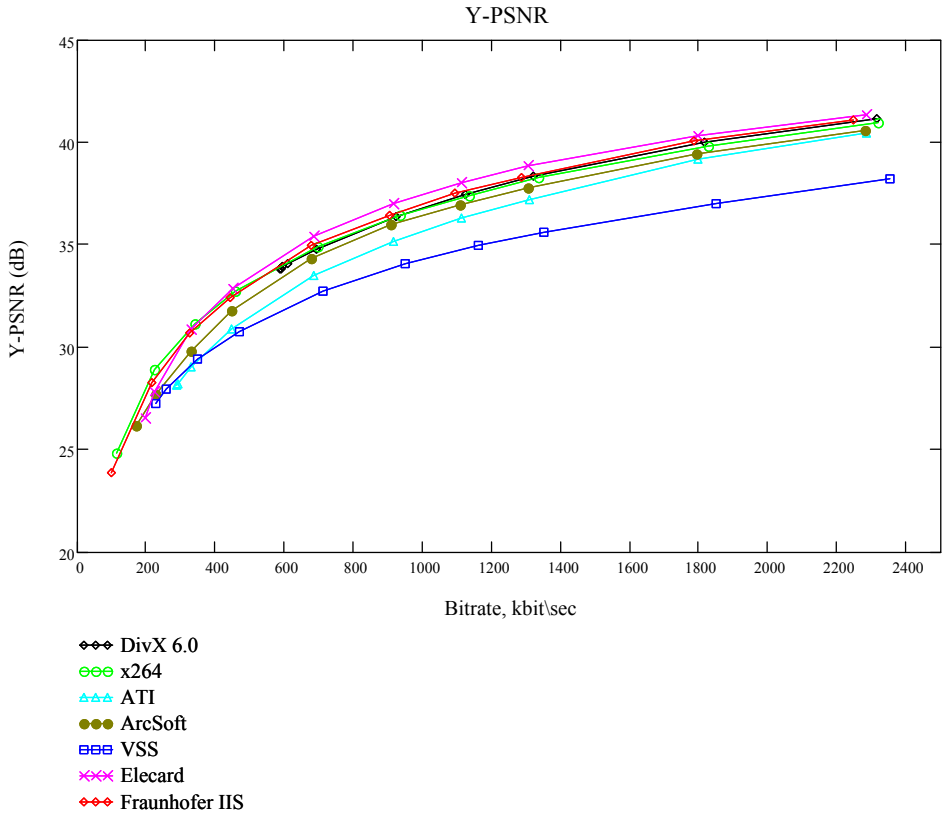
Picture 55. Y-PSNR. Sequence “foreman”



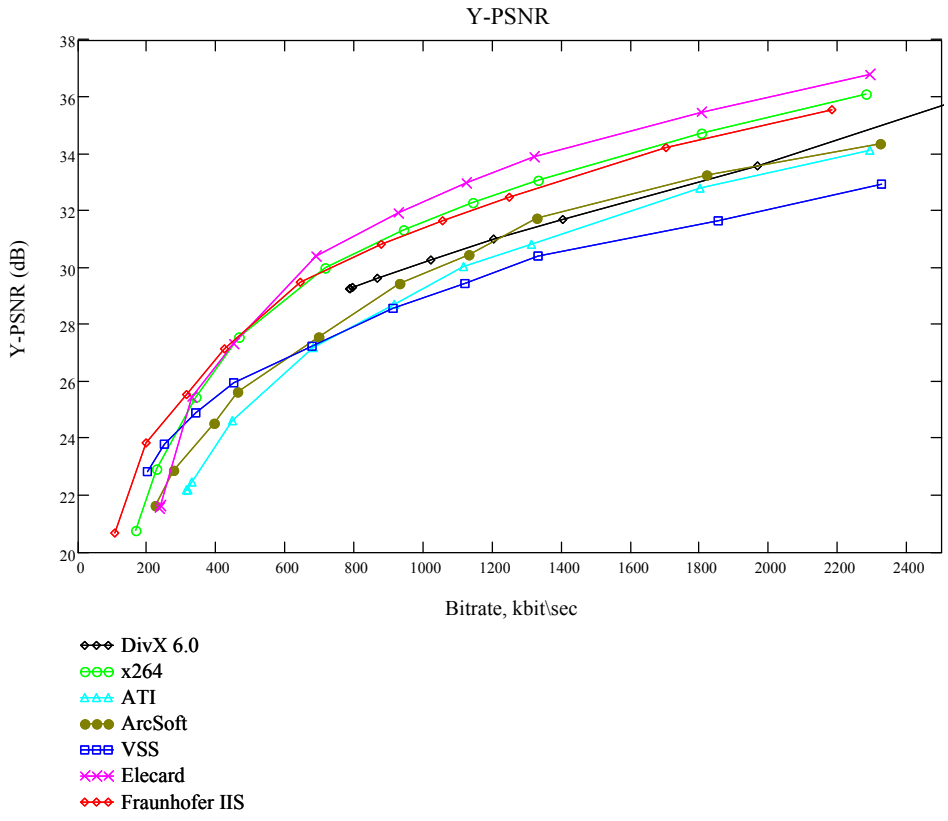
Picture 56. Y-PSNR. Sequence “susi”



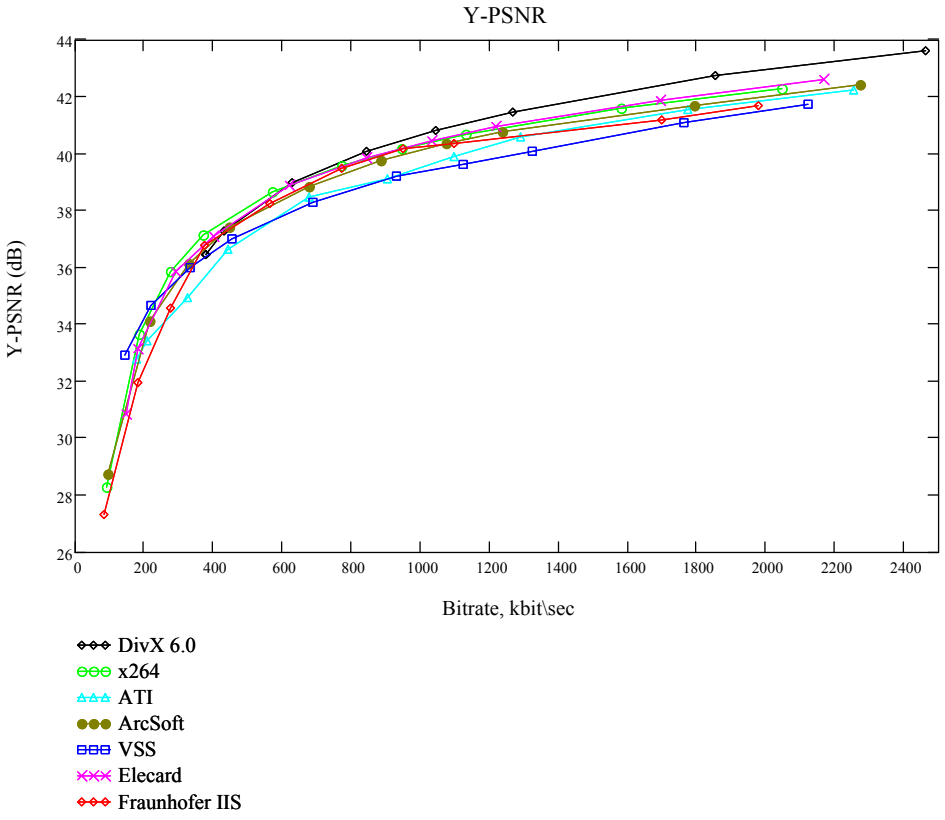
Picture 57. Y-PSNR. Sequence “bbc”



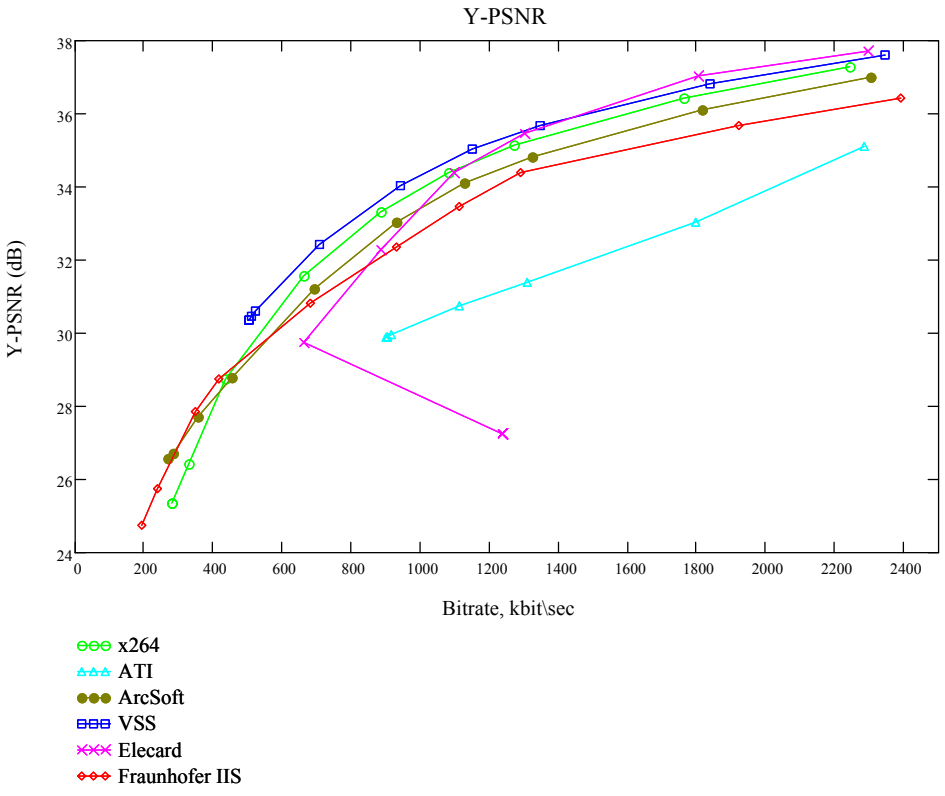
Picture 58. Y-PSNR. Sequence “battle”



Picture 59. Y-PSNR. Sequence “simpsons”



Picture 60. Y-PSNR. Sequence "matrix"



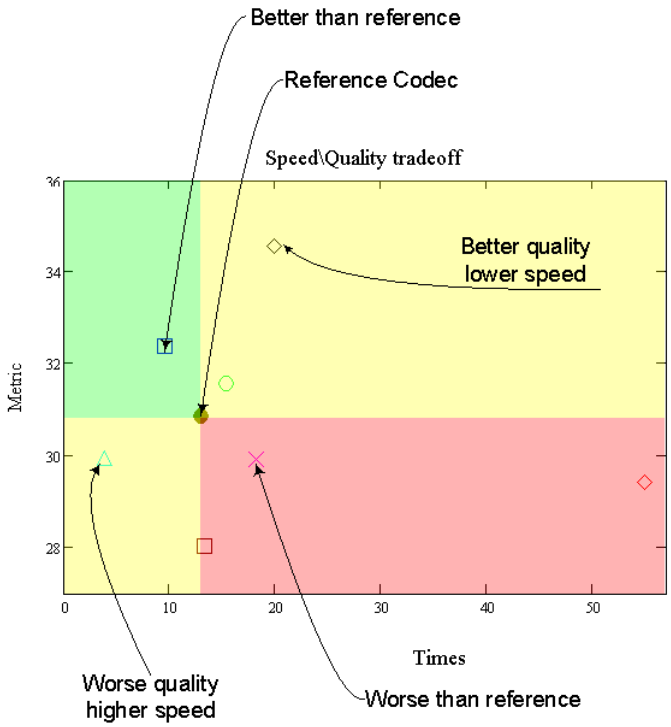
Picture 61. Y-PSNR. Sequence "concert"

Speed/Quality Tradeoff

Here are the most important graphs for this preset. They show codec's work time (abscissa axis) and quality of sequence, compressed with it (axis of ordinates). If codec A is to the left of codec B and above it then codec A is better than codec B; if to the right and below it then on the contrary, codec B is better than codec A. In other cases nothing can be said about definite advantage of one's codec compared to another's.

Bit rates in cutlines inside this section are target bit rates (not real codec bit rates).

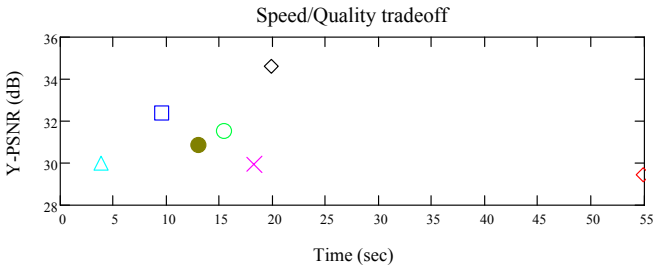
Graph type	Total number of graph	Inserted in this document
Speed/quality tradeoff	**1960**	**8 (0.4%)**



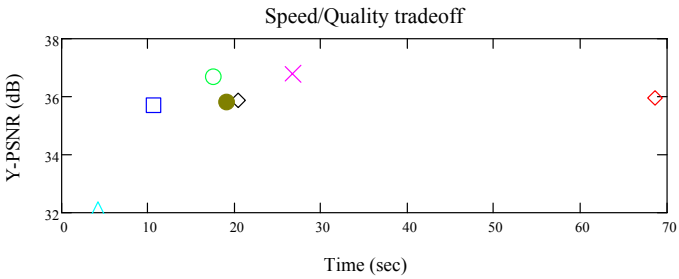
Picture 62. Codec comparison with at speed/quality tradeoff graphs

Picture 63. Legend for Speed/quality tradeoff graphs

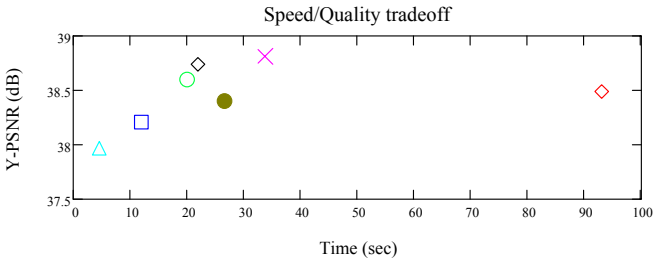
Sequence “susi”, Y-PSNR



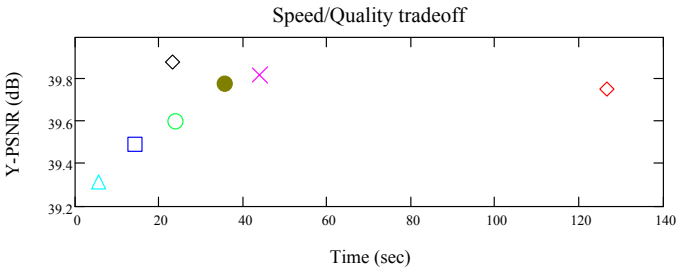
Picture 64. Bitrate 100 kbps



Picture 65. Bitrate 340 kbps

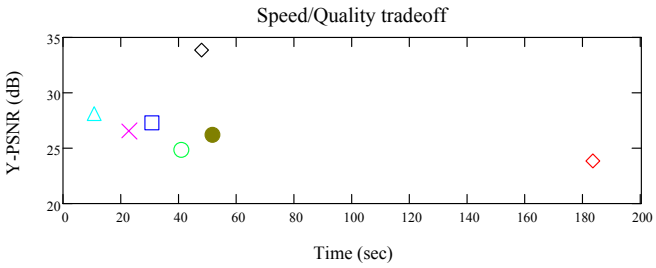


Picture 66. Bitrate 938 kbps

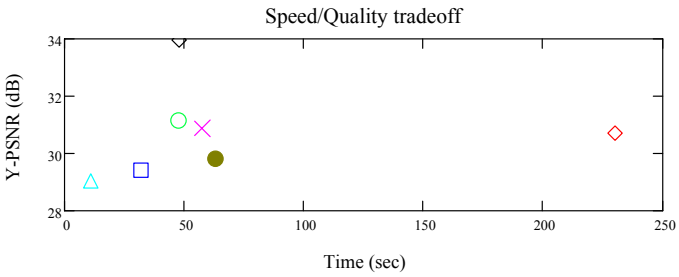


Picture 67. Bitrate 2340 kbps

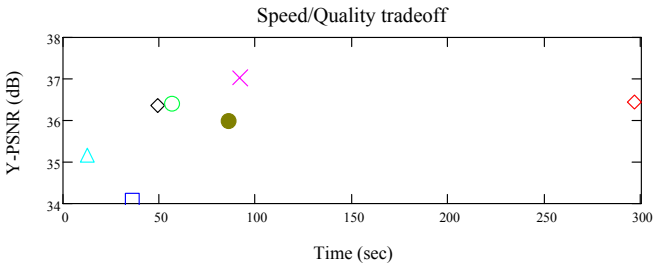
Sequence “battle”, Y-PSNR



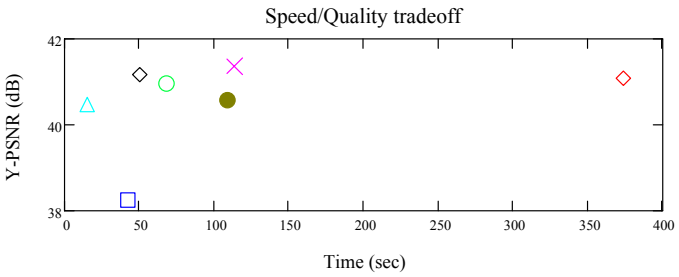
Picture 68. Bitrate 100 kbps



Picture 69. Bitrate 340 kbps



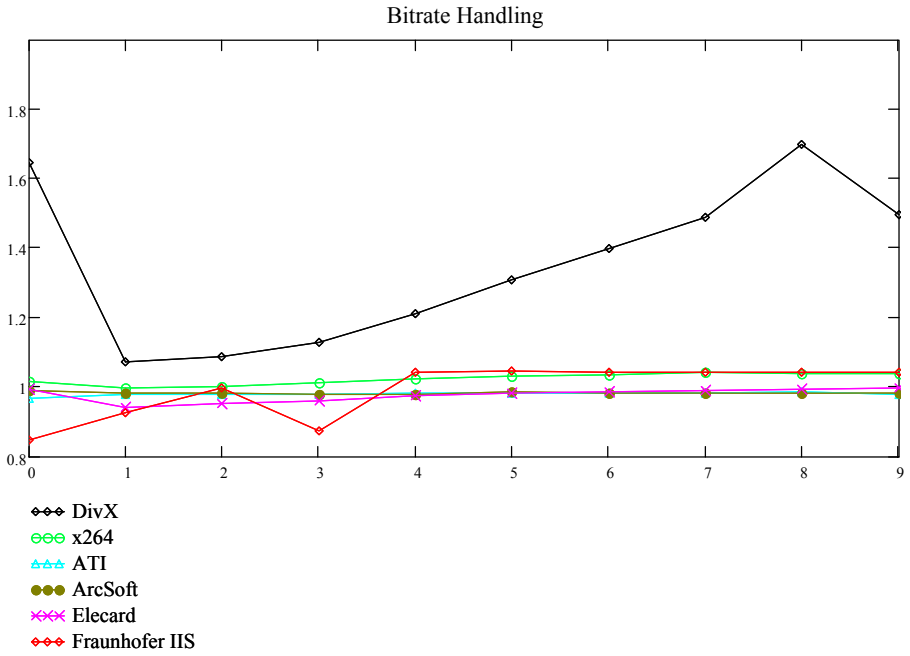
Picture 70. Bitrate 938 kbps



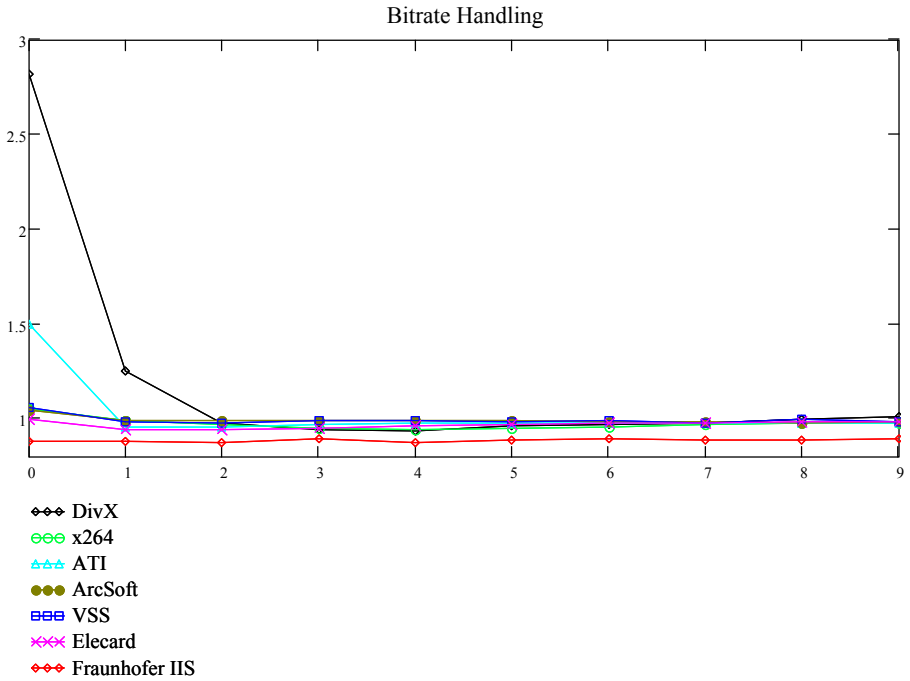
Picture 71. Bitrate 2340 kbps

Bitrate handling

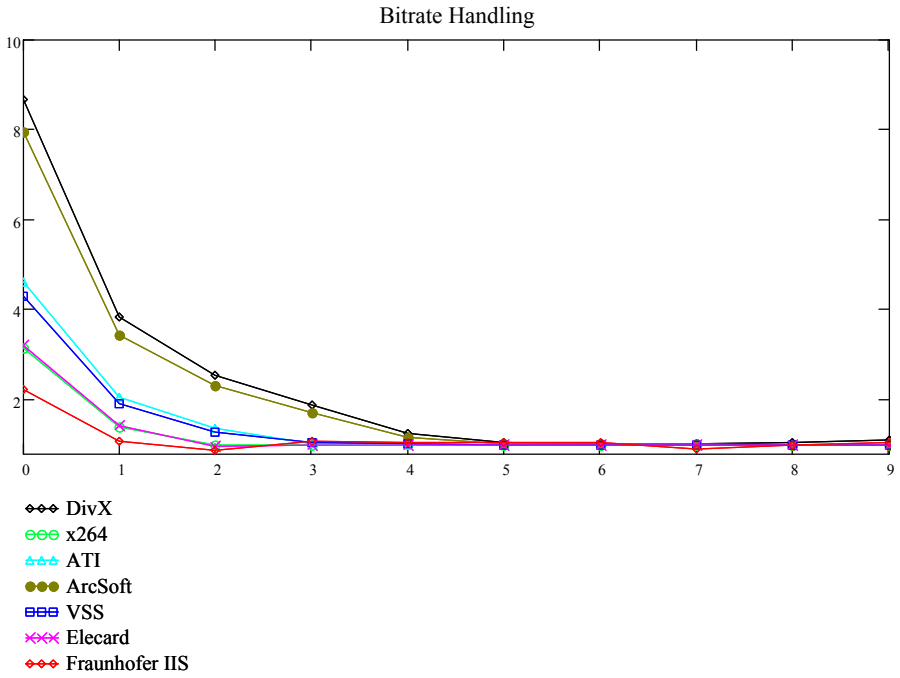
Consider graphs showing how accurately every codec kept requested bit rate. Grades on x-axis show target bitrates (point 0 is 100 kbps, point 9 – 2340 kbps). Grades on y-axis show how many times codec exceeded declared bit rate (real bit rate to predetermined bit rate ratio).



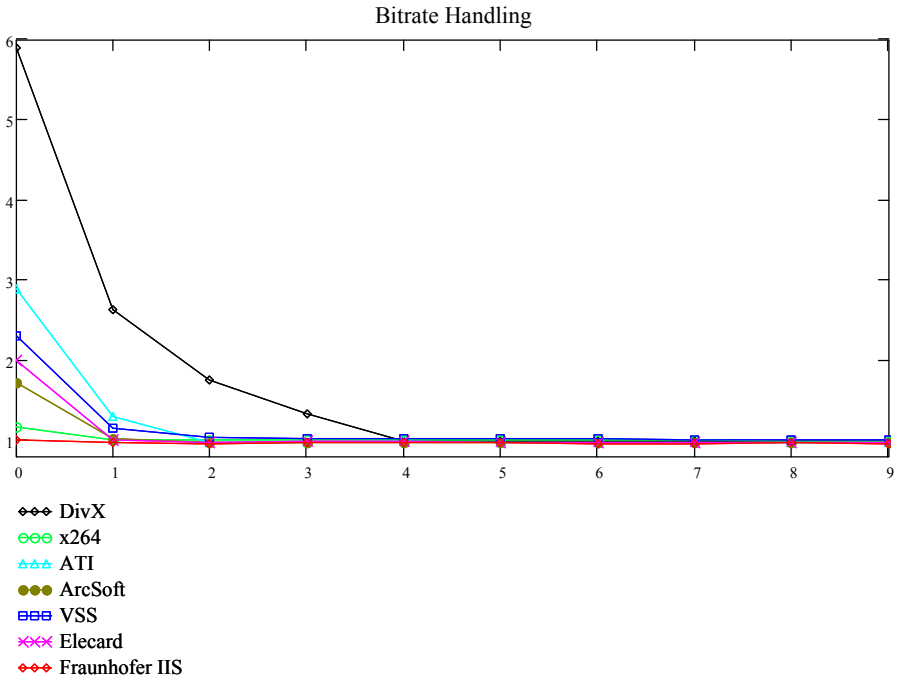
Picture 72. Bitrate handling. Preset "Best speed". Sequence "foreman"



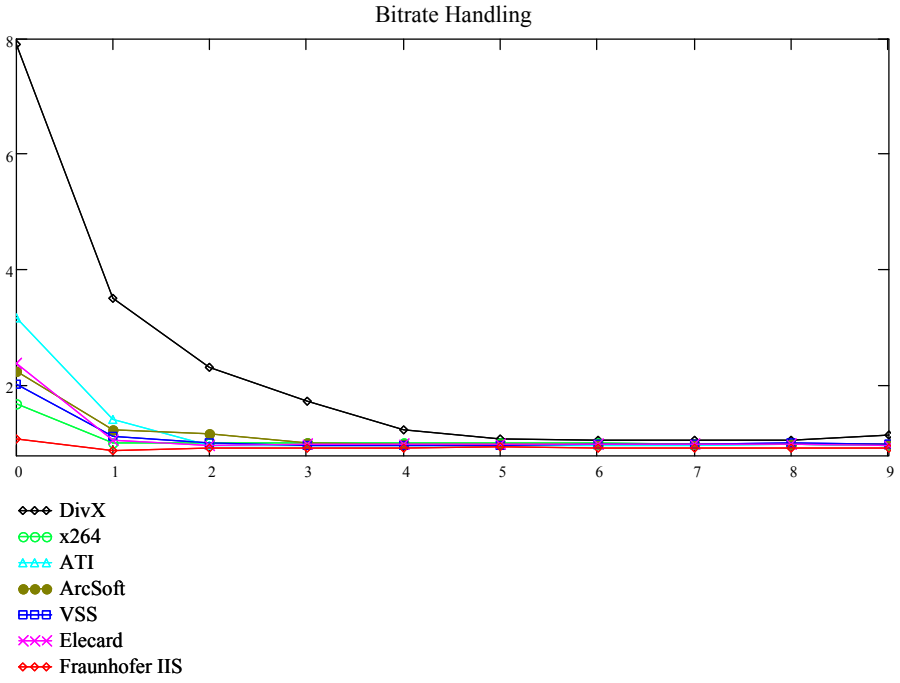
Picture 73. Bitrate handling. Preset "Best speed". Sequence "susi"



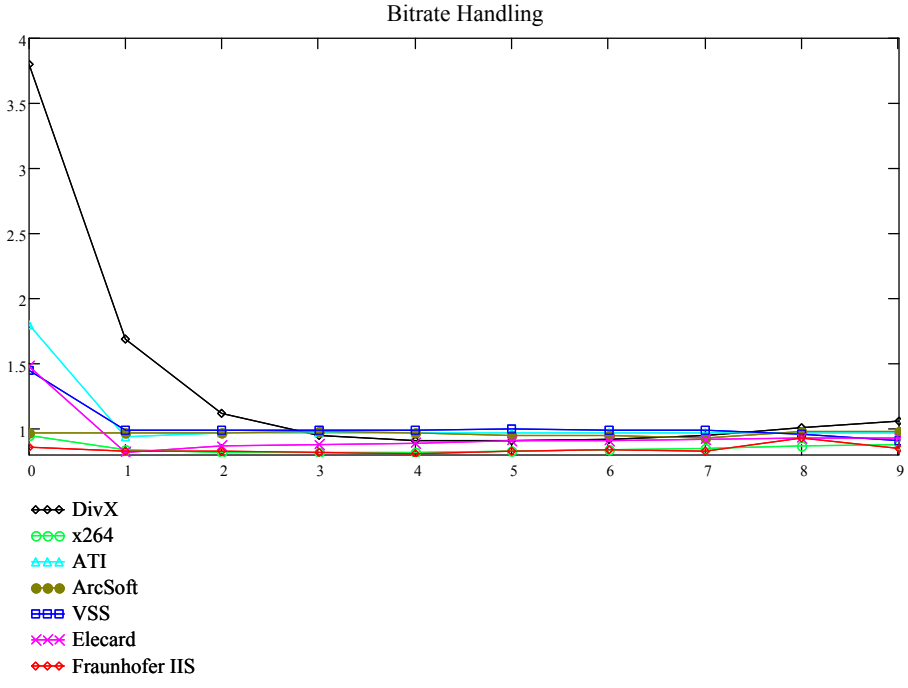
Picture 74. Bitrate handling. Preset “Best speed”. Sequence “bbc”



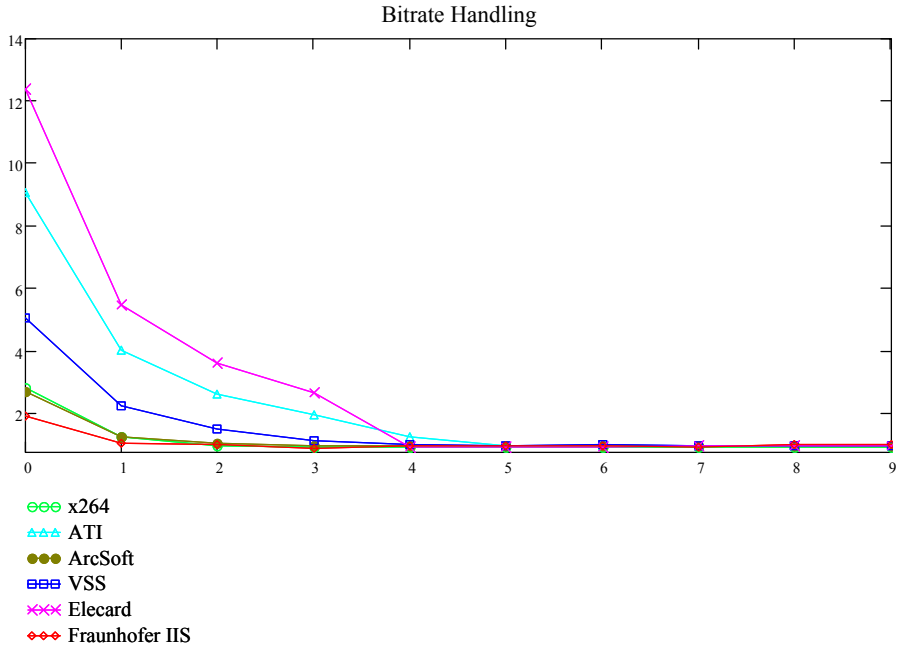
Picture 75. Bitrate handling. Preset “Best speed”. Sequence “battle”



Picture 76. Bitrate handling. Preset “Best speed”. Sequence “simpsons”



Picture 77. Bitrate handling. Preset “Best speed”. Sequence “matrix”



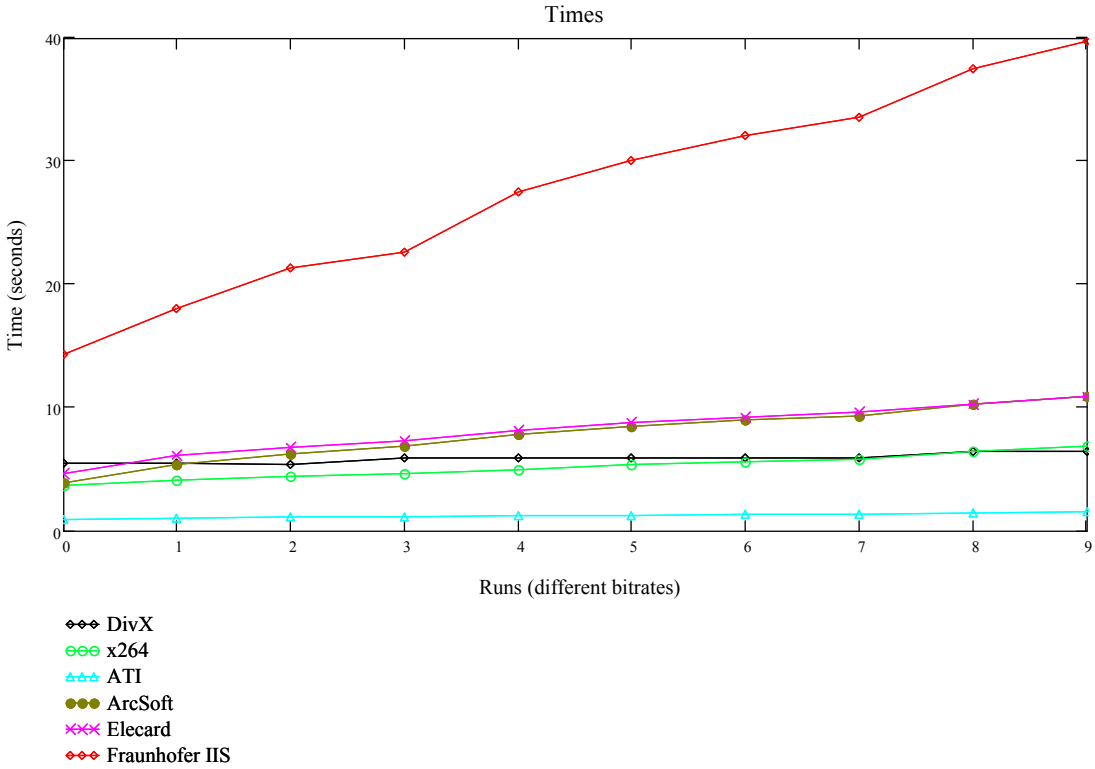
Picture 78. Bitrate handling. Preset "Best speed". Sequence "concert"

Conclusions:

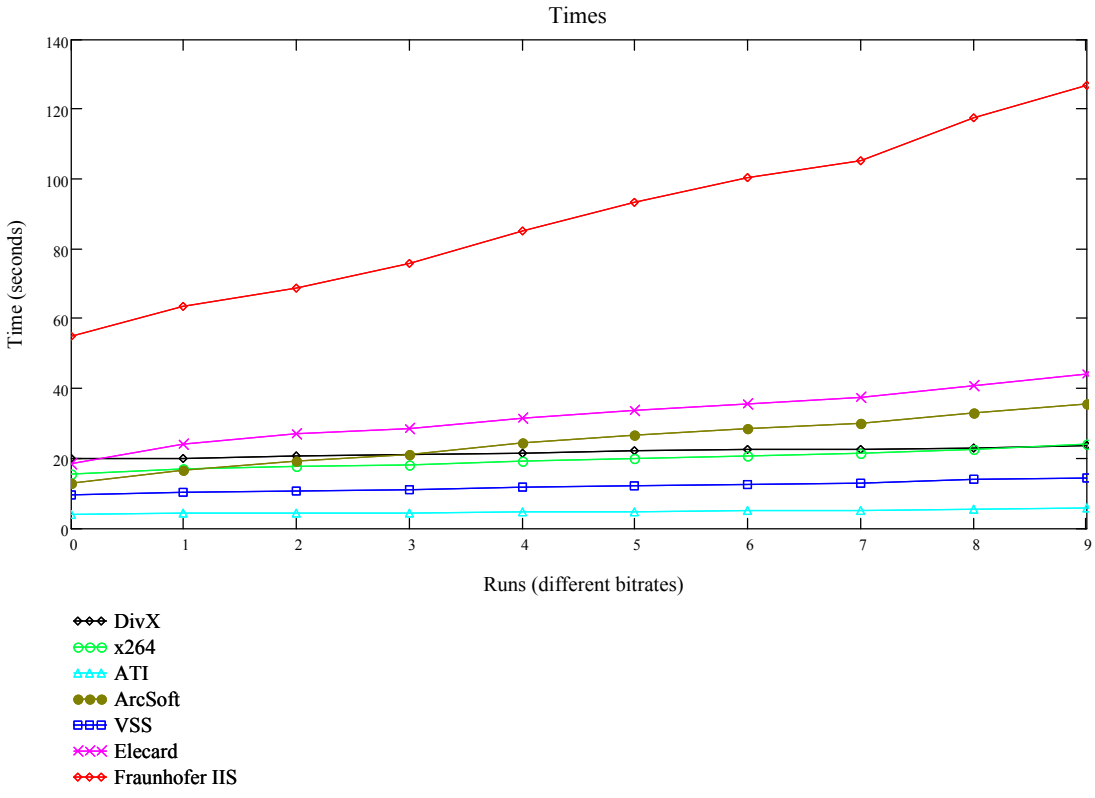
- Situation here is similar to "Best Quality" preset – DivX keeps bit rate very bad, raising it too high on low bit rates.
- The best among H.264 codecs in bitrate handling is Fraunhofer IIS, as before.

Time

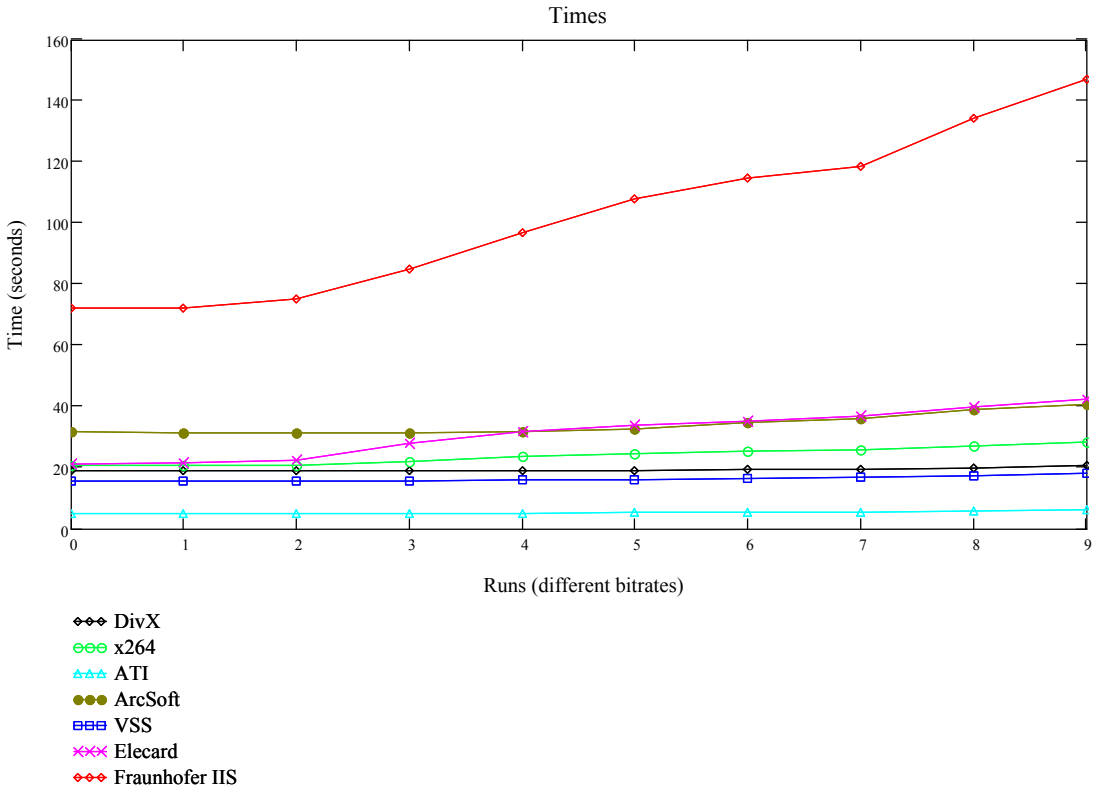
Now let's consider graphs that show time spent for each video sequence encoding in "Best Speed" preset and average normalized time for this preset.



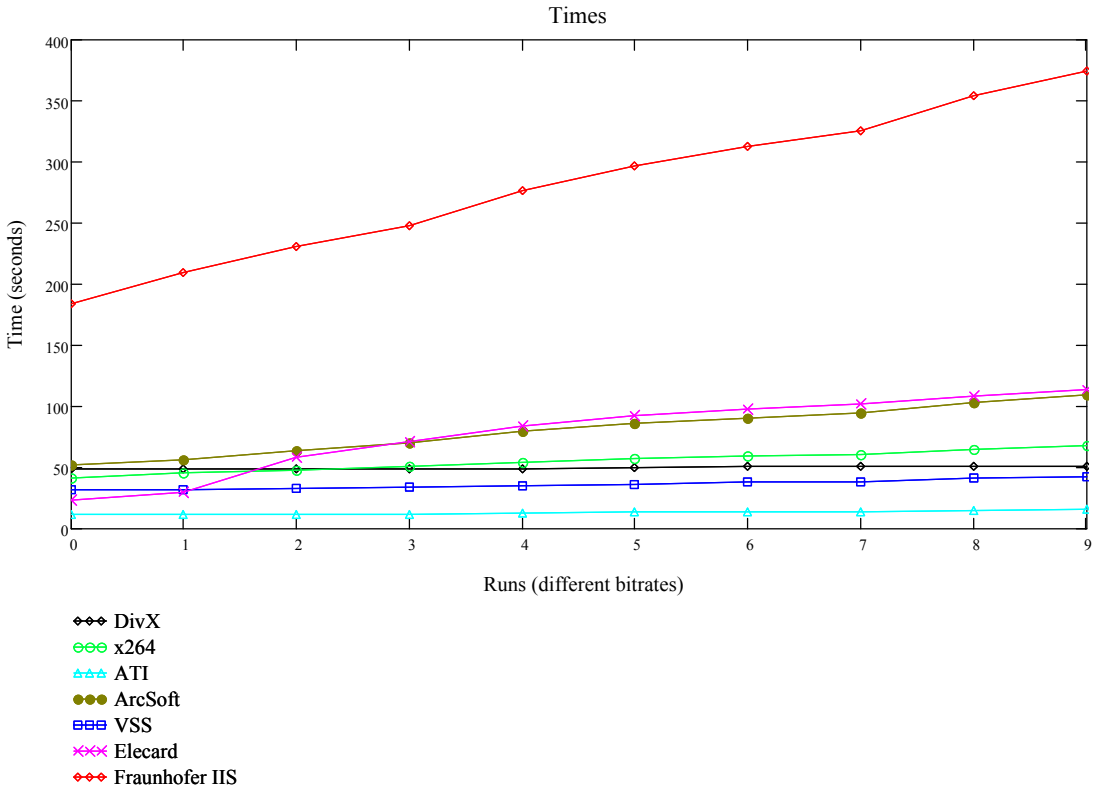
Picture 79. Encoding time. Preset “Best speed”. Sequence “foreman”



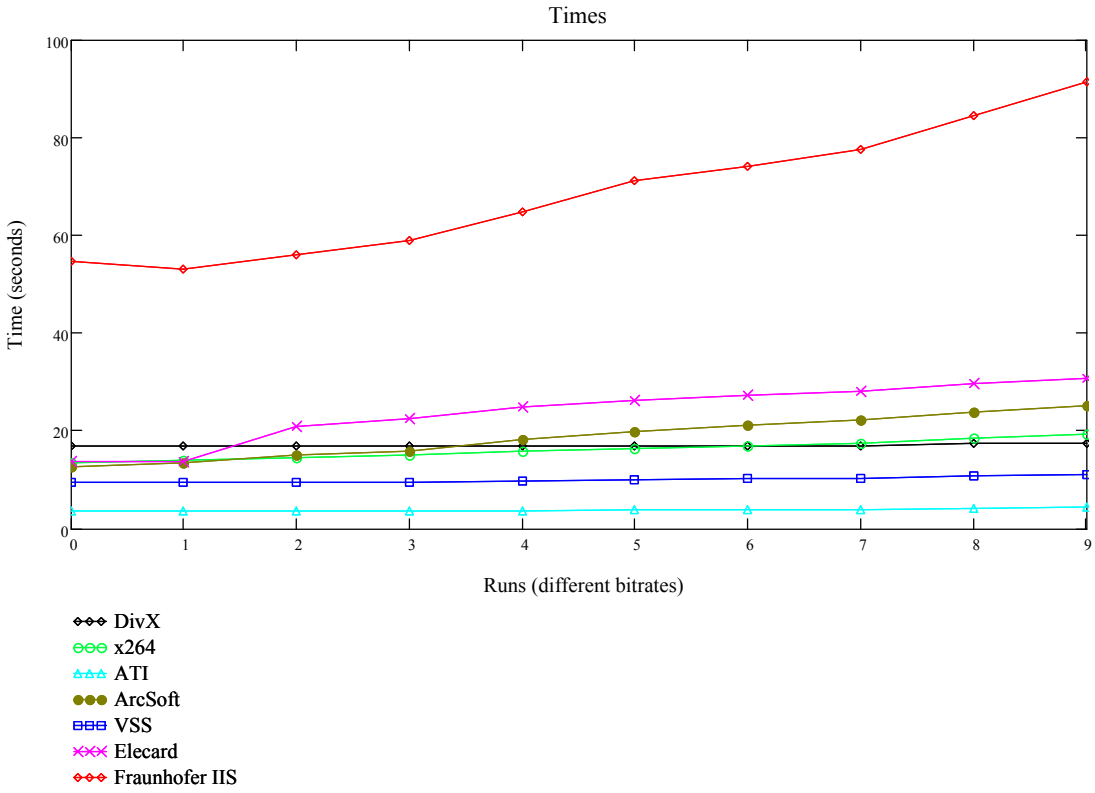
Picture 80. Encoding time. Preset “Best speed”. Sequence “susi”



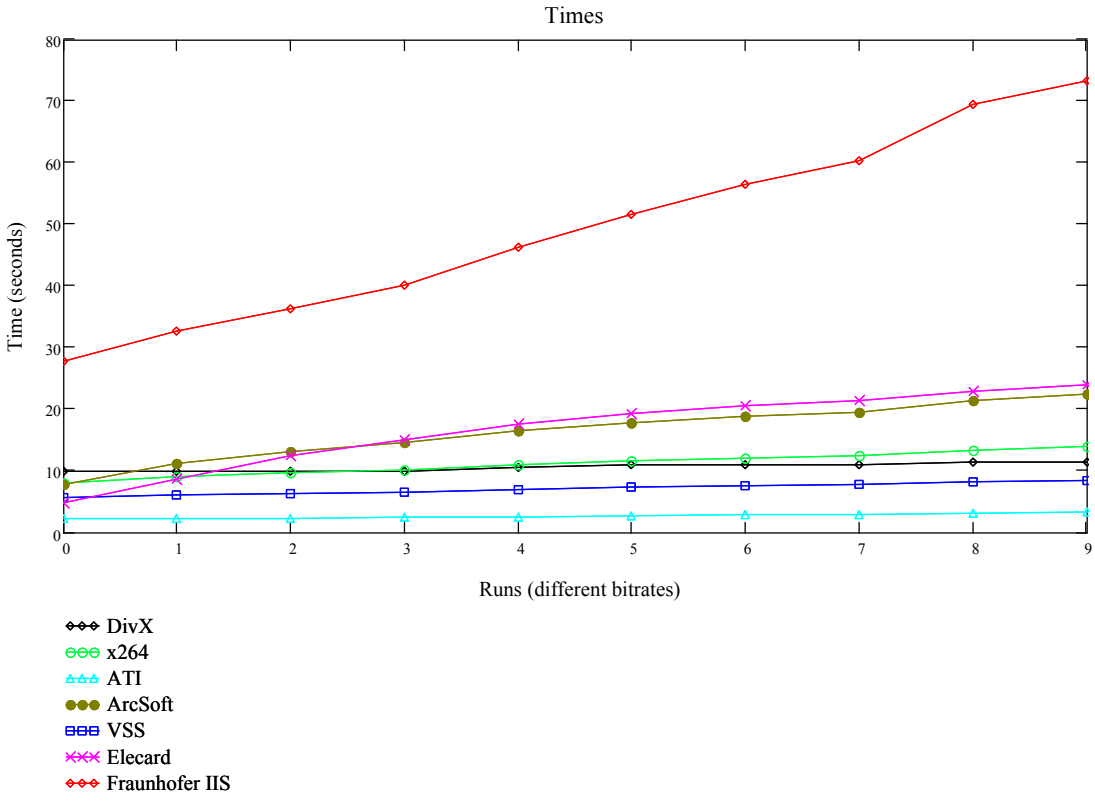
Picture 81. Encoding time. Preset "Best speed". Sequence "bbc"



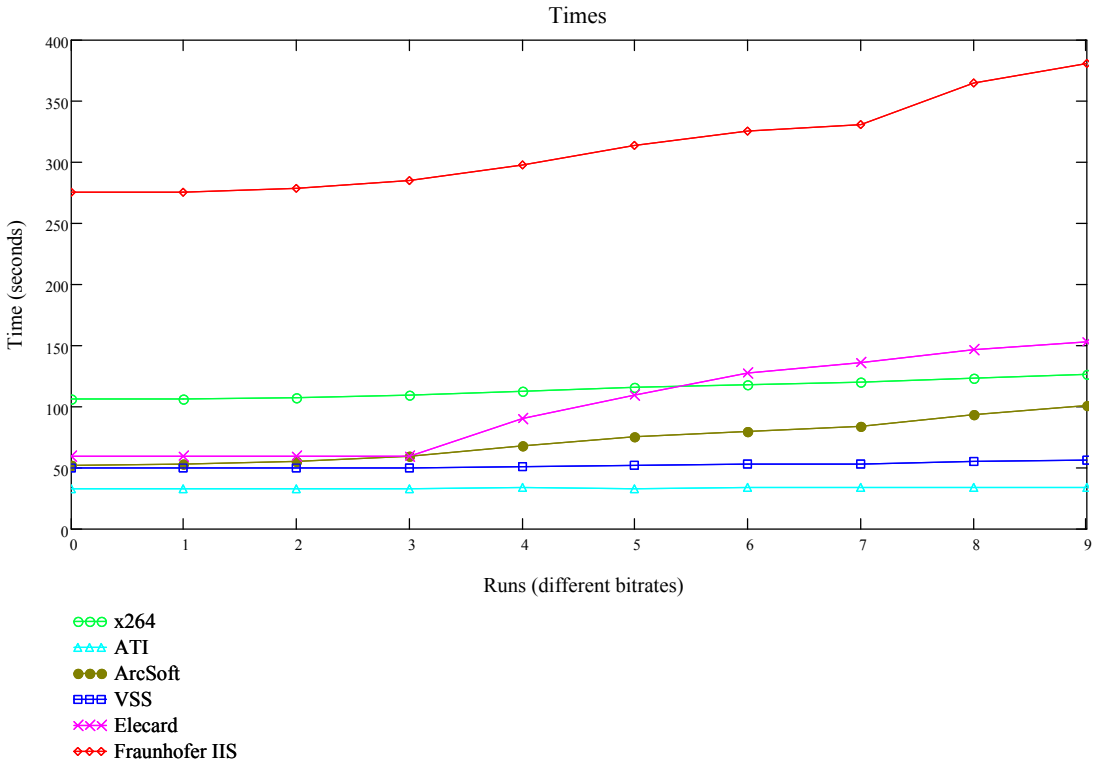
Picture 82. Encoding time. Preset "Best speed". Sequence "battle"



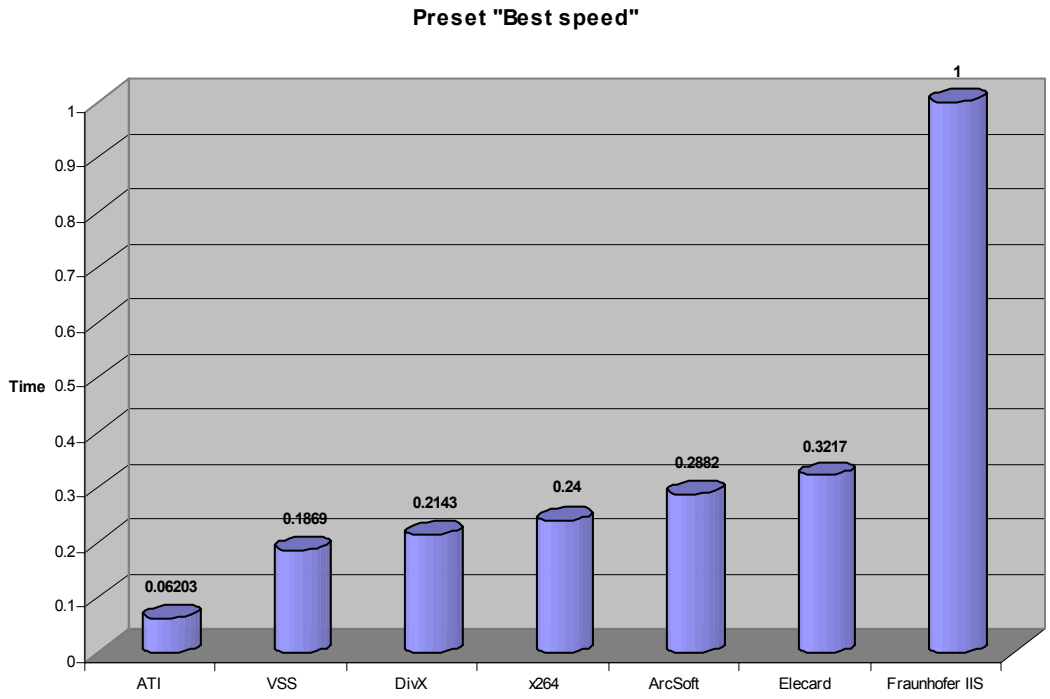
Picture 83. Encoding time. Preset "Best speed". Sequence "simpsons"



Picture 84. Encoding time. Preset "Best speed". Sequence "matrix "



Picture 85. Encoding time. Preset “Best speed”. Sequence “concert”



Picture 86. Normalized average encoding time. Preset “Best speed”

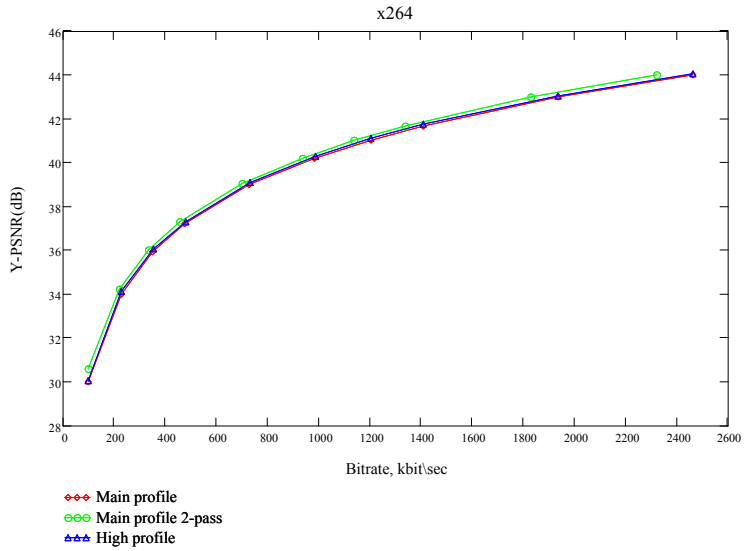
Conclusions:

- Absolute leader by speed is codec from ATI company. At the average, it worked several times faster than its nearest competitor – VSS codec. And at the same time it is not worst one by the quality, though it is one of trailing codecs.
- The slowest codec in this mode again is Fraunhofer IIS. It lags behind its nearest competitor more than 3 times.
- Encoding time for Fraunhofer IIS codec increases noticeably when bit rate is increased.

2-pass mode and High profile

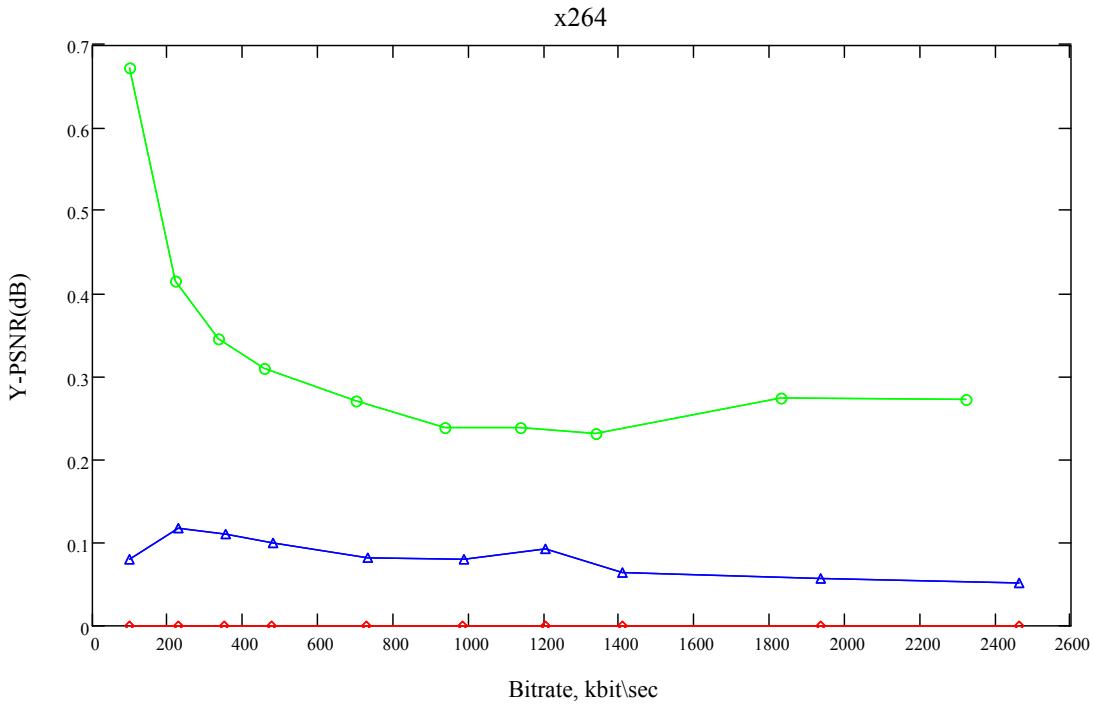
Additionally to one-pass mode in Main profile of standard H.264 we tested two-pass mode and compression using possibilities of High profile. Only x264, Elecard, Ateme and Fraunhofer IIS codecs took part in this test. This test was held on “foreman” and “battle” sequences and without speed measuring.

As it is seen on picture below, it is difficult to analyze results using PSNR/Bitrate graphs. That is why we use Delta Y-PSNR graphs instead of absolute Y-PSNR graphs in this part. In all graphs reference is Main Profile one pass results for current codec.

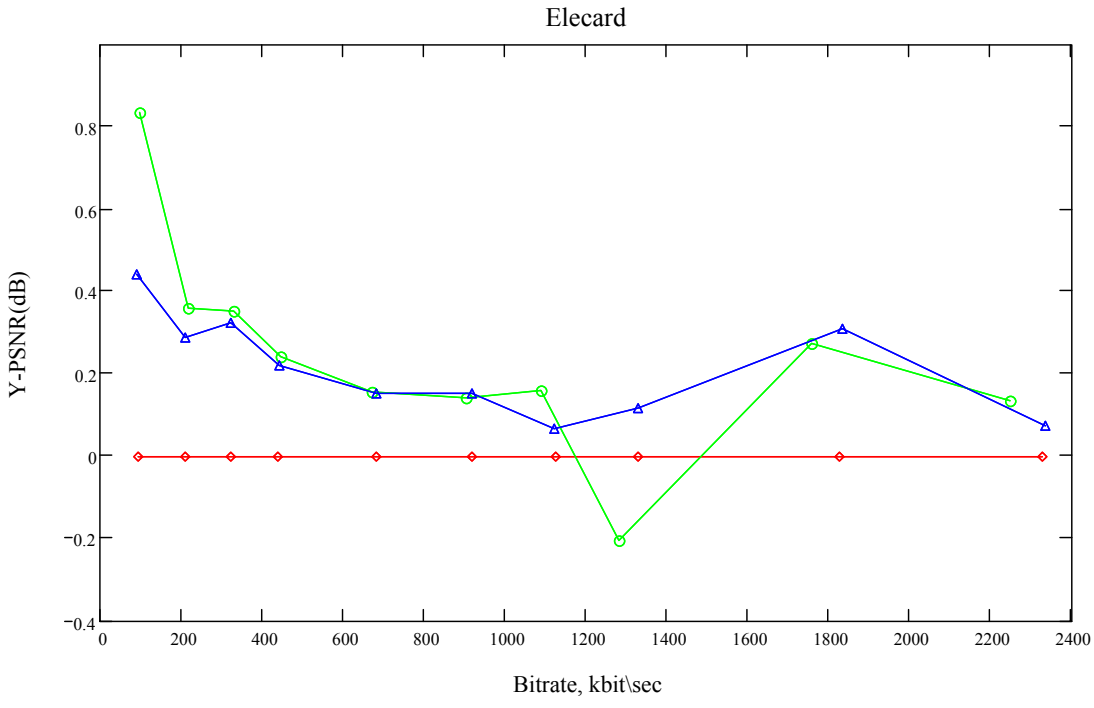


Picture 87. Different presets of codec x264, Absolute Y-PSNR

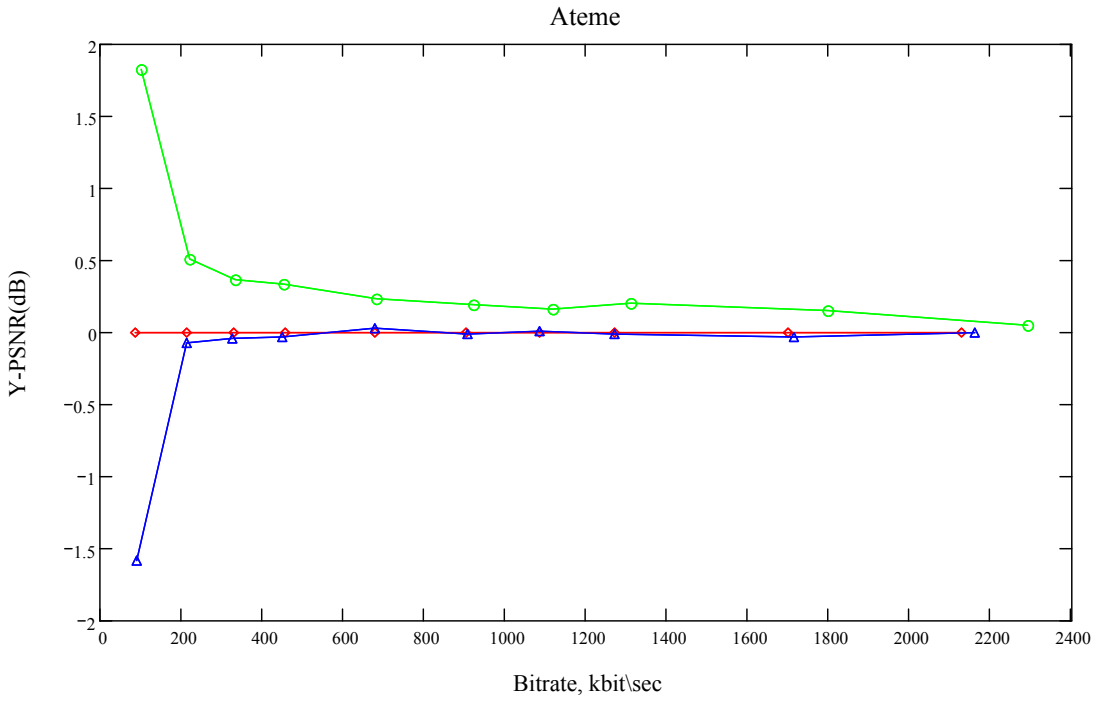
Sequence “foreman”, preset “Best quality”



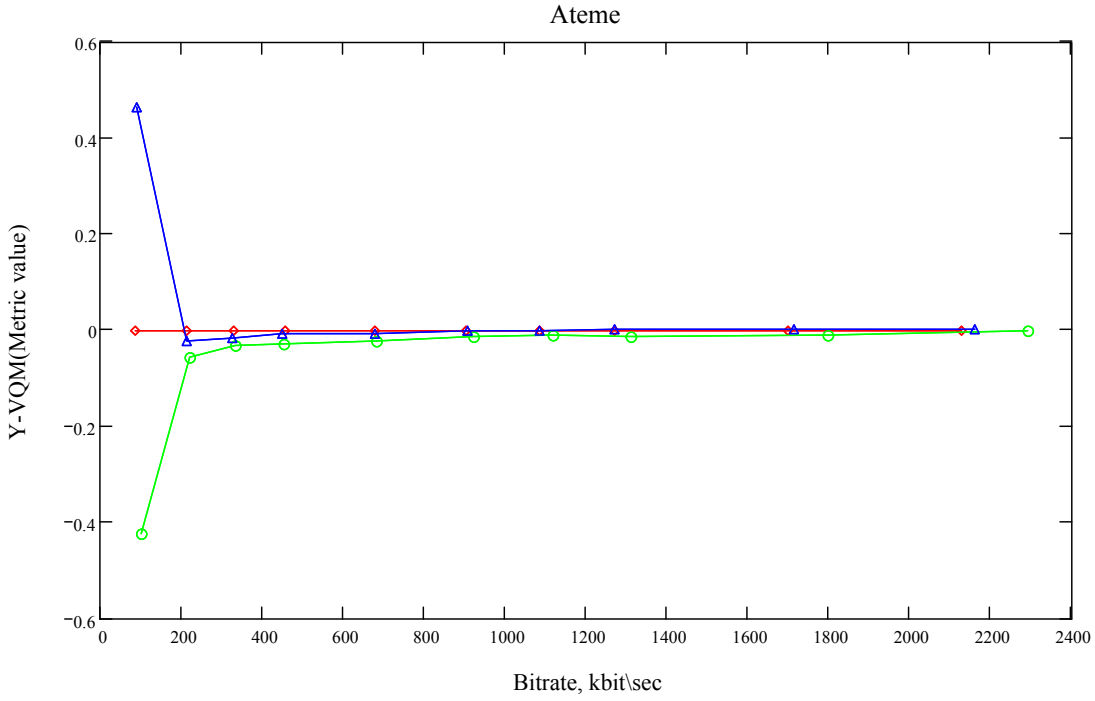
Picture 88. Different presets of codec x264, Delta Y-PSNR



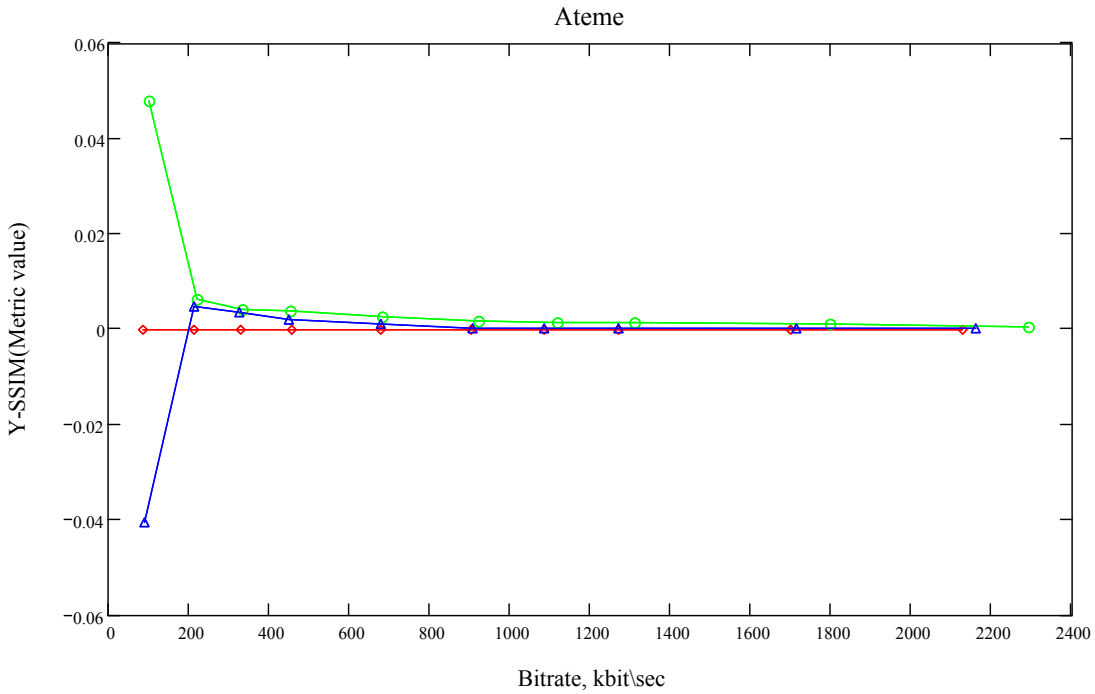
Picture 89. Different presets of codec Elecard H.264, Delta Y-PSNR



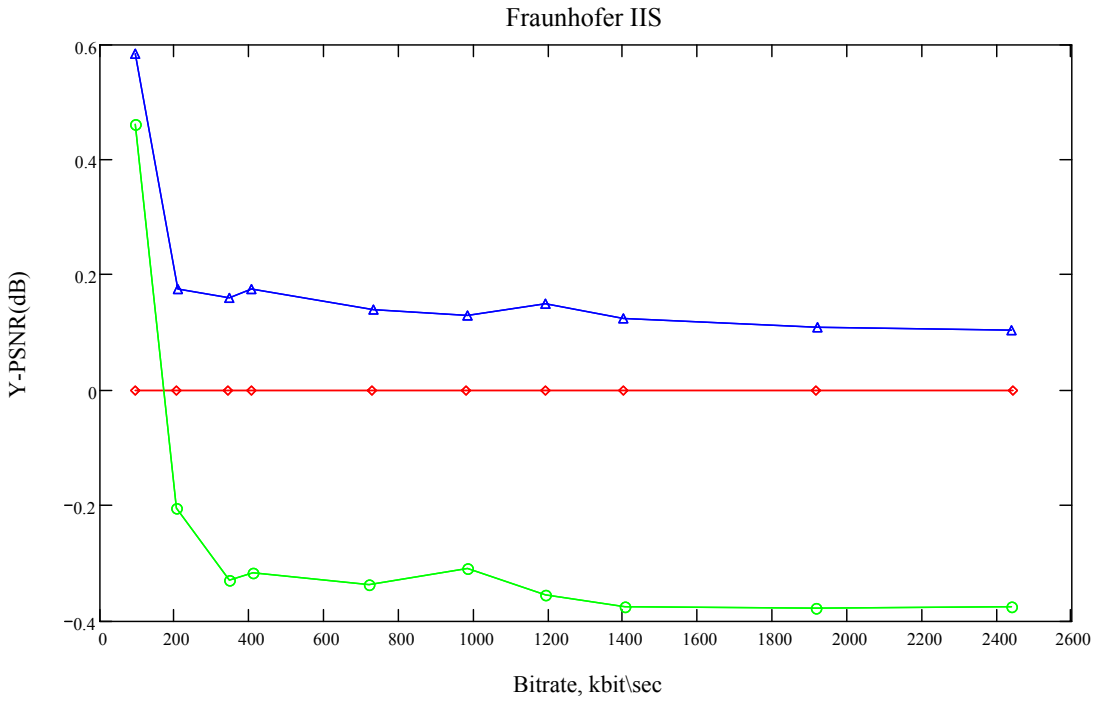
Picture 90. Different presets of codec Ateme H.264, Delta Y-PSNR



Picture 91. Different presets of codec Ateme H.264, Delta Y-VQM

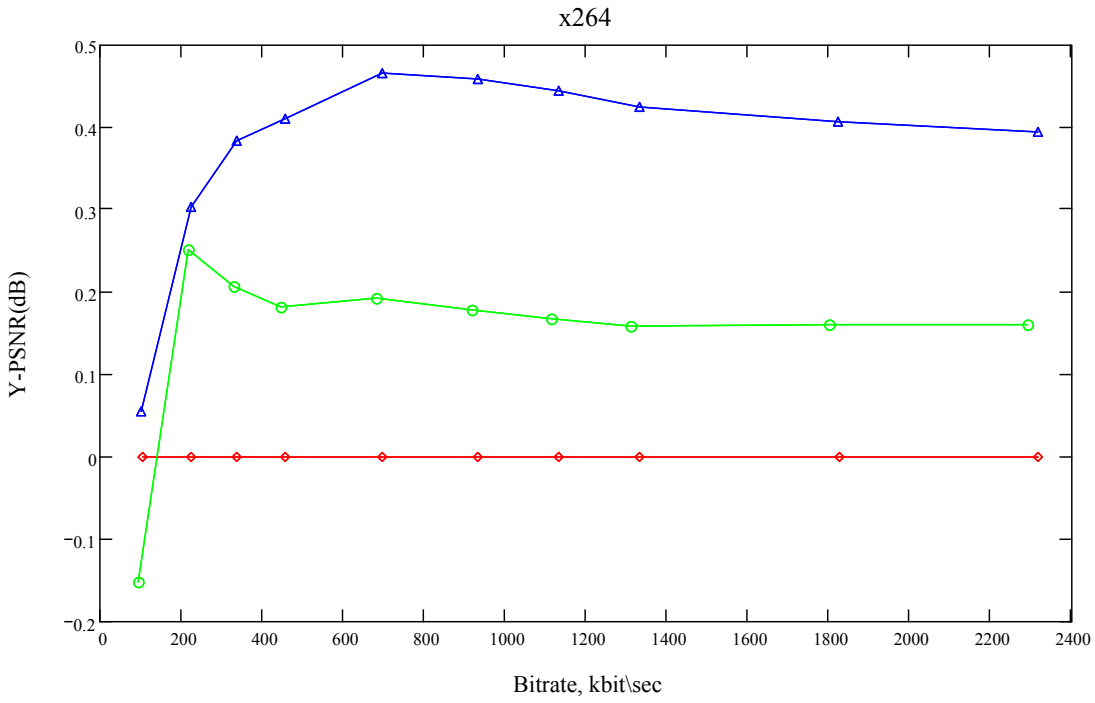


Picture 92. Different presets of codec Ateme H.264, Delta-Y-SSIM

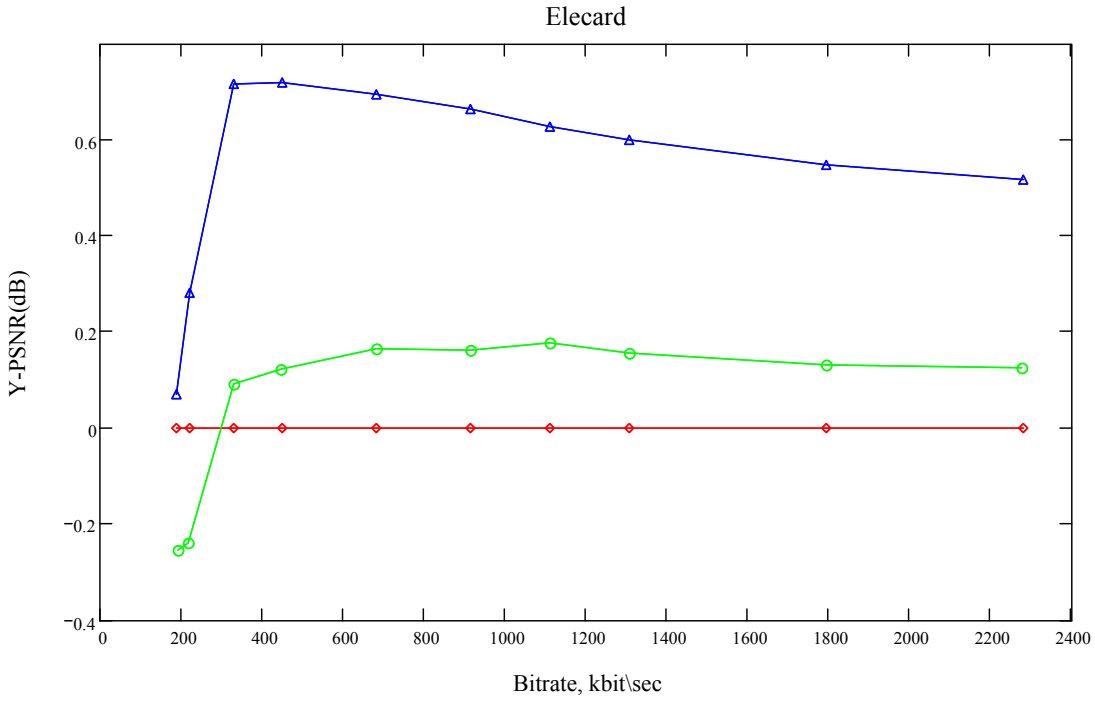


Picture 93. Different presets of codec Fraunhofer IIS H.264, Delta Y-PSNR

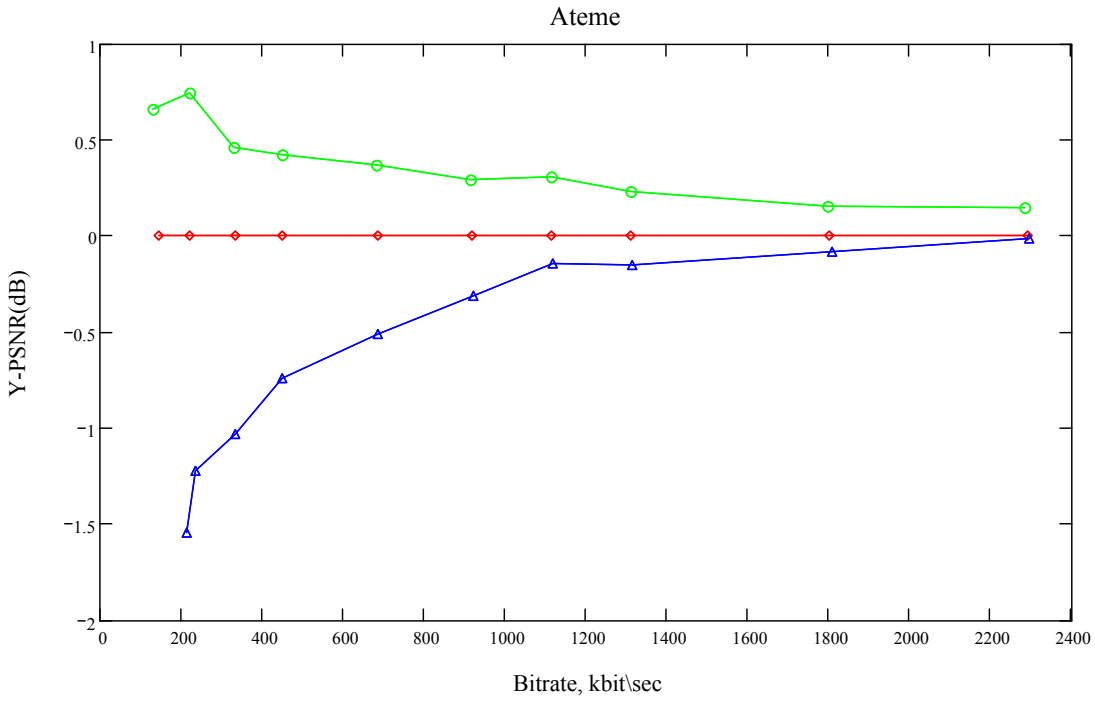
Sequence “battle”, preset “Best quality”



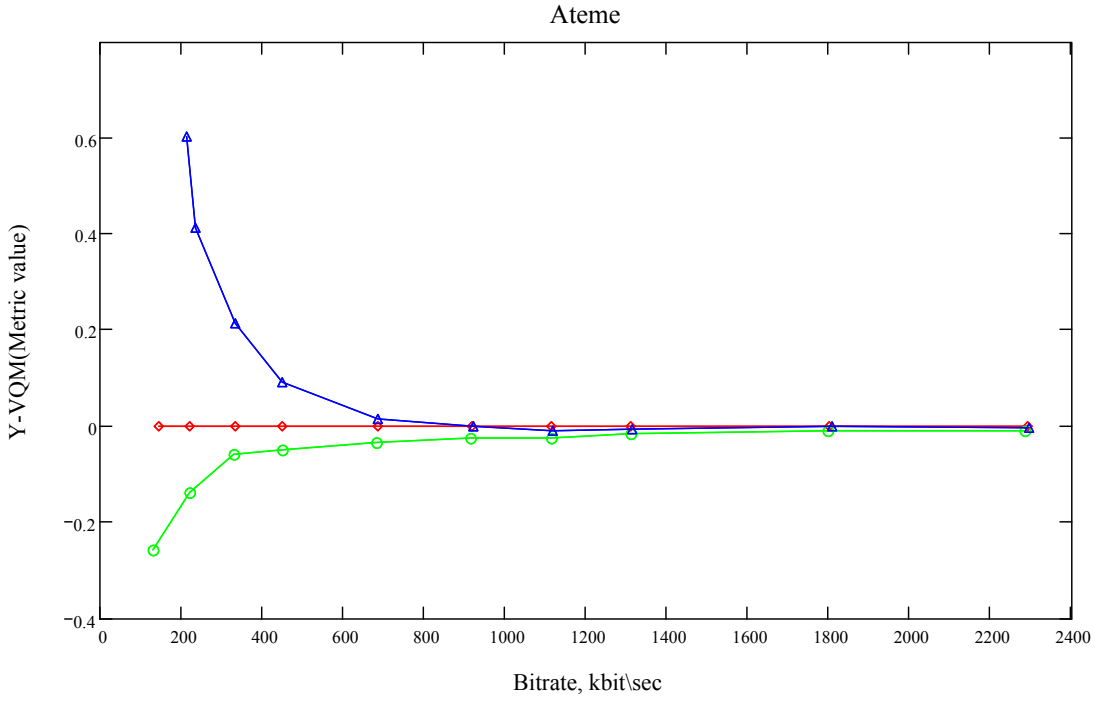
Picture 94. Different presets of codec x264, Delta Y-PSNR



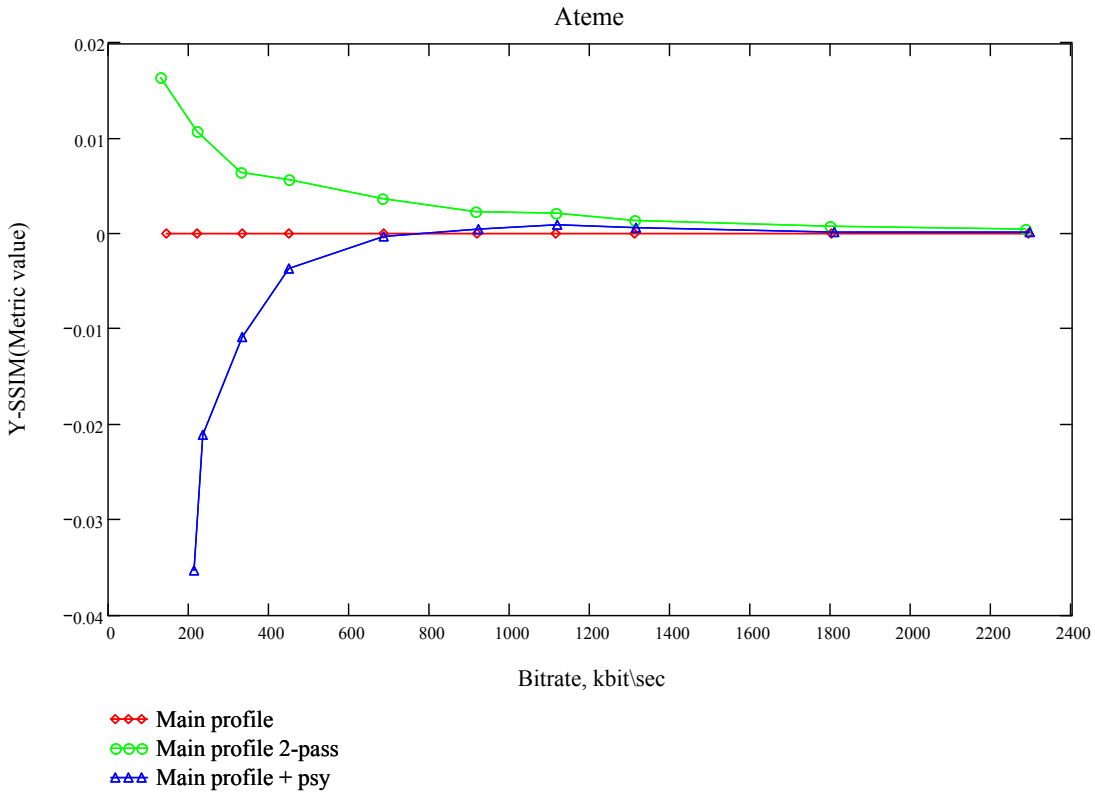
Picture 95. Different presets of codec Elecard H.264, Delta Y-PSNR



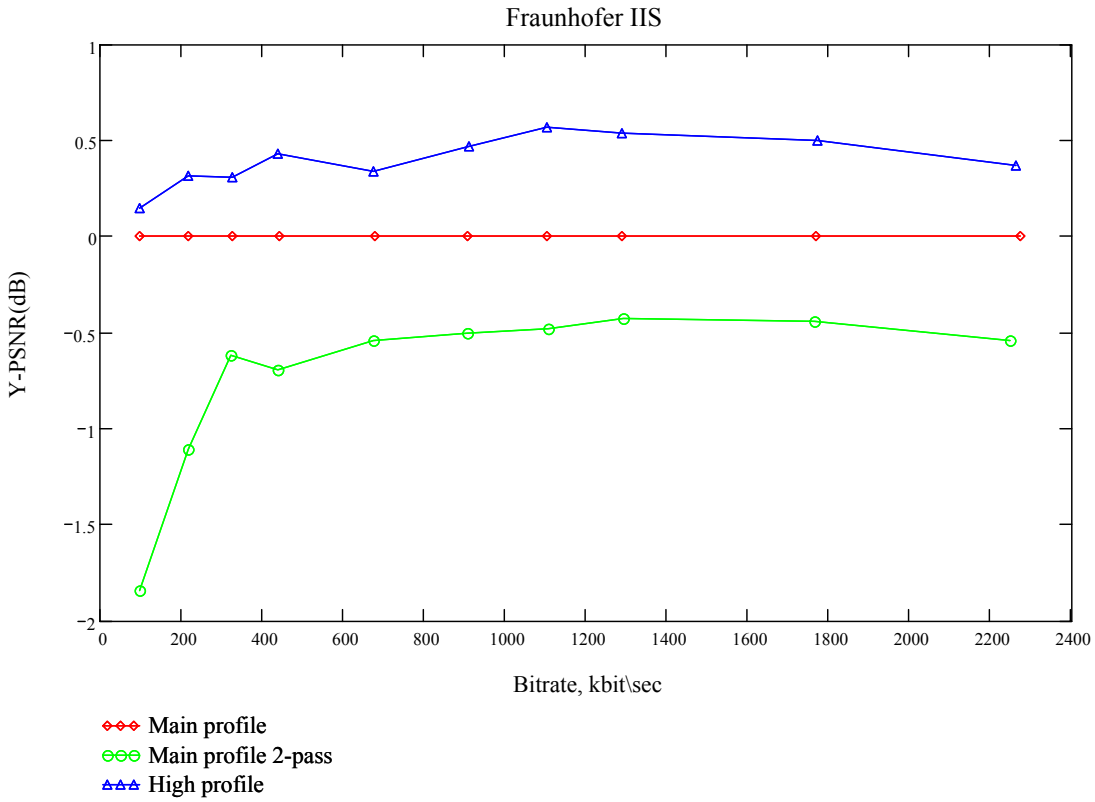
Picture 96. Different presets of codec Ateme H.264, Delta Y-PSNR



Picture 97. Different presets of codec Ateme H.264, Delta Y-VQM



Picture 98. Different presets of codec Ateme H.264, Delta Y-SSIM



Picture 99. Different presets of codec Fraunhofer IIS H.264, Delta Y-PSNR

Note:

- Fraunhofer IIS H.264 codec showed some not very adequate results during this part of testing because of presets that were provided for us:
  - Main profile with two passes has quality parameter equal to 4 (good quality)
  - Main profile with one pass and High profile have quality parameter equal to 6 (best quality)

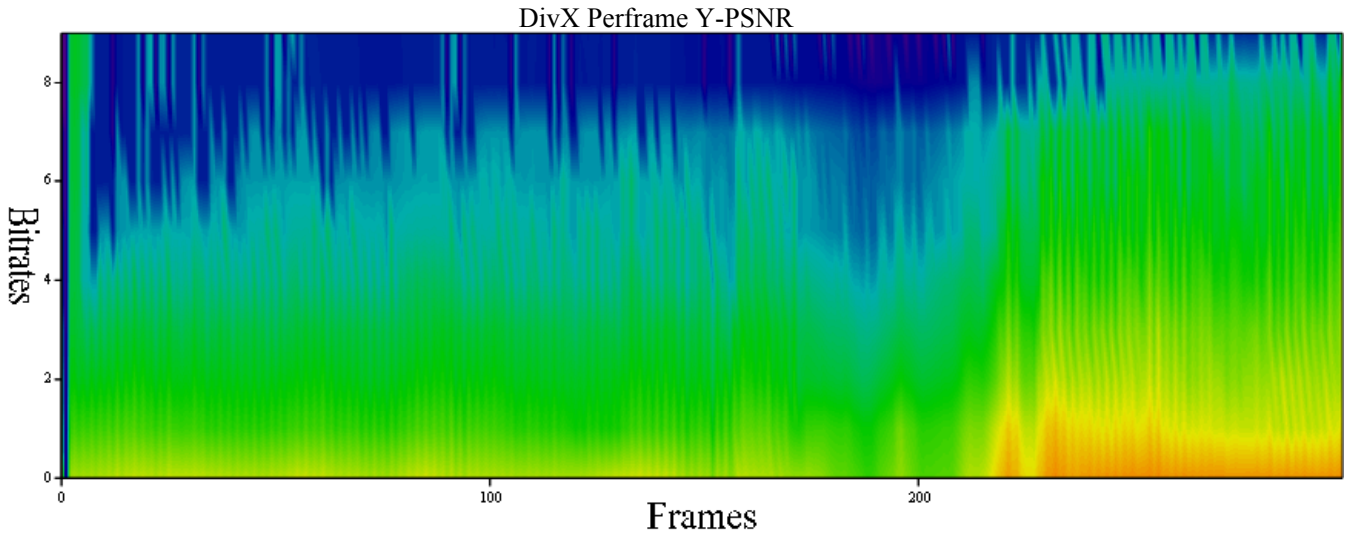
Per frame comparison

In addition to average metrics' values we measured metrics' values for separate frames. Basing on this data, Rate Control of each codec could be analyzed. As a rule, on such graphs it is clearly seen whether codec uses B-frames or not ("striation" of graph), location of I-frames (abrupt peaks or collapses on graph), scene changing reaction, quality of R-D models, etc.

On the next graphs there are frames on abscissa axis and bit rates on axis of ordinates (low bit rates are below). Color shows metric's value: the more red, the lesser is value. Thus, for Y-PSNR metric red color means bad quality, blue – good quality. All graphs of one sequence have the same colors (every color corresponds to only one value of metric on all graphs).

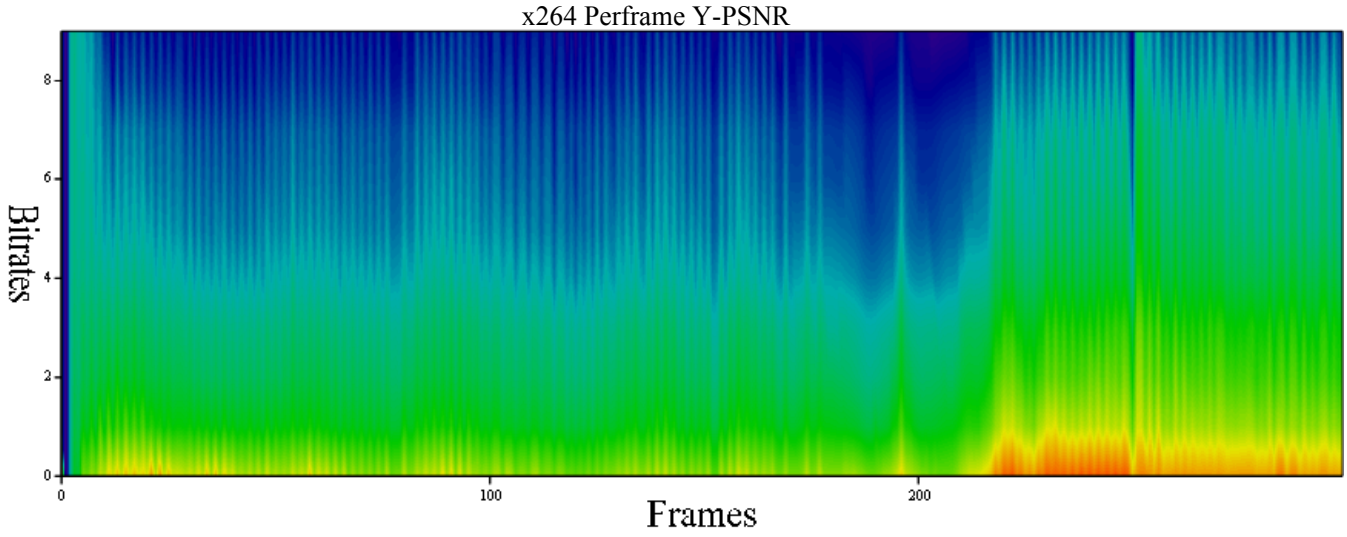
Analyzing these graphics we should take into account bit rate handling of codecs since these graphs do not consider it.

Sequence "foreman", Y-PSNR, Preset "Best quality"



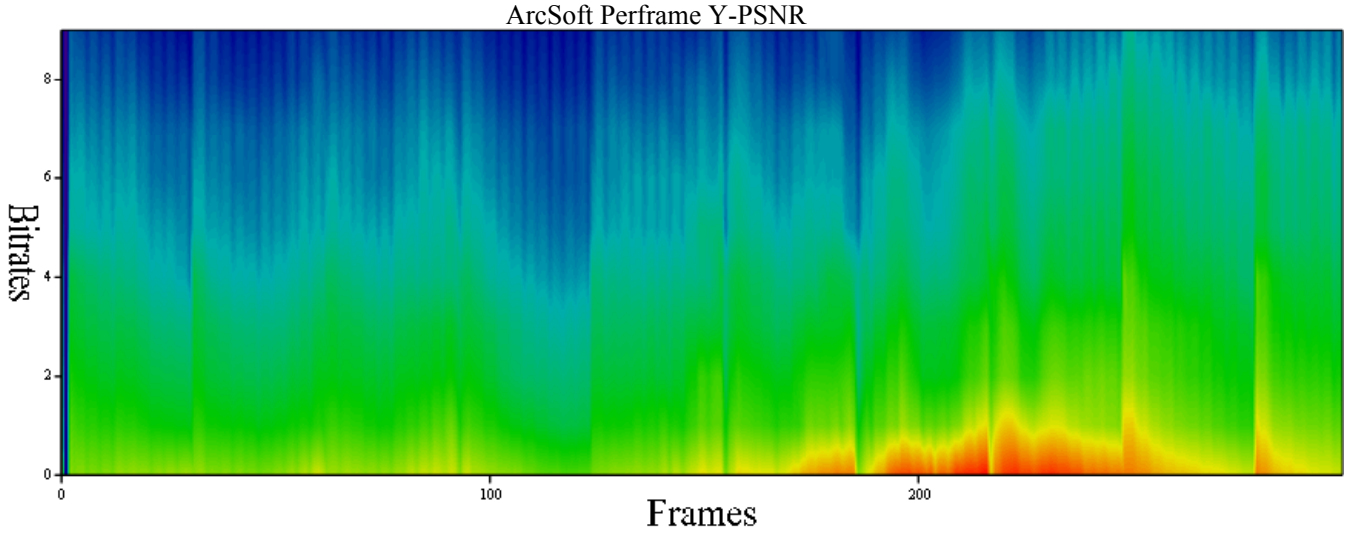
M

Picture 100. Codec DivX. Sequence "foreman".



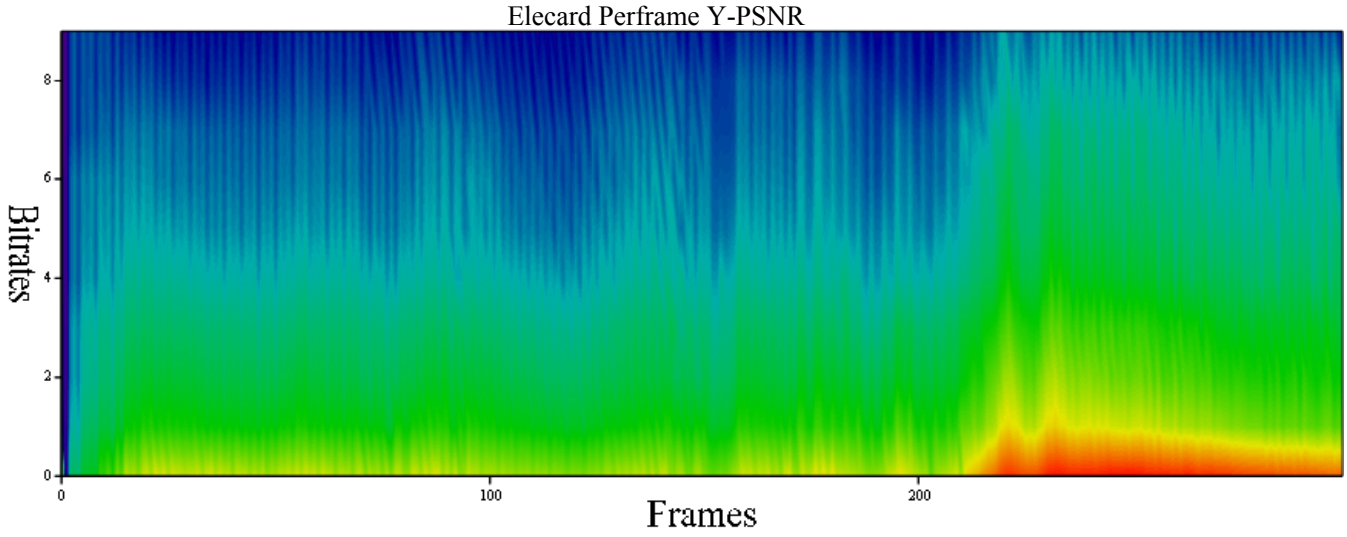
M

Picture 101. Codec x264. Sequence “foreman”.



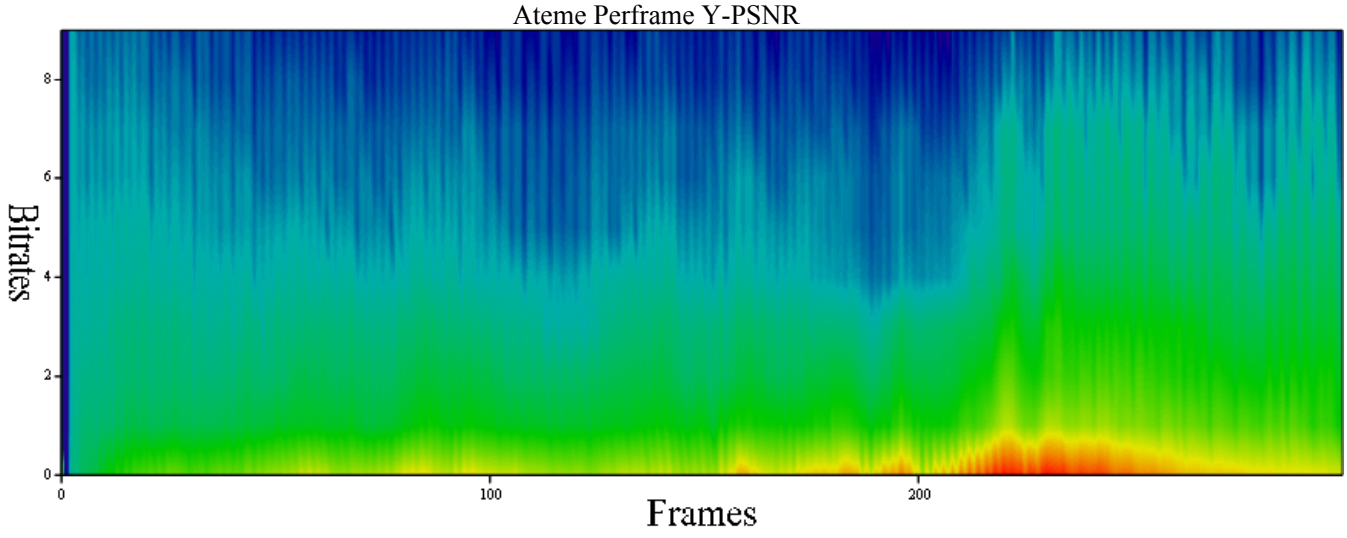
M

Picture 102. Codec ArcSoft H.264. Sequence “foreman”.



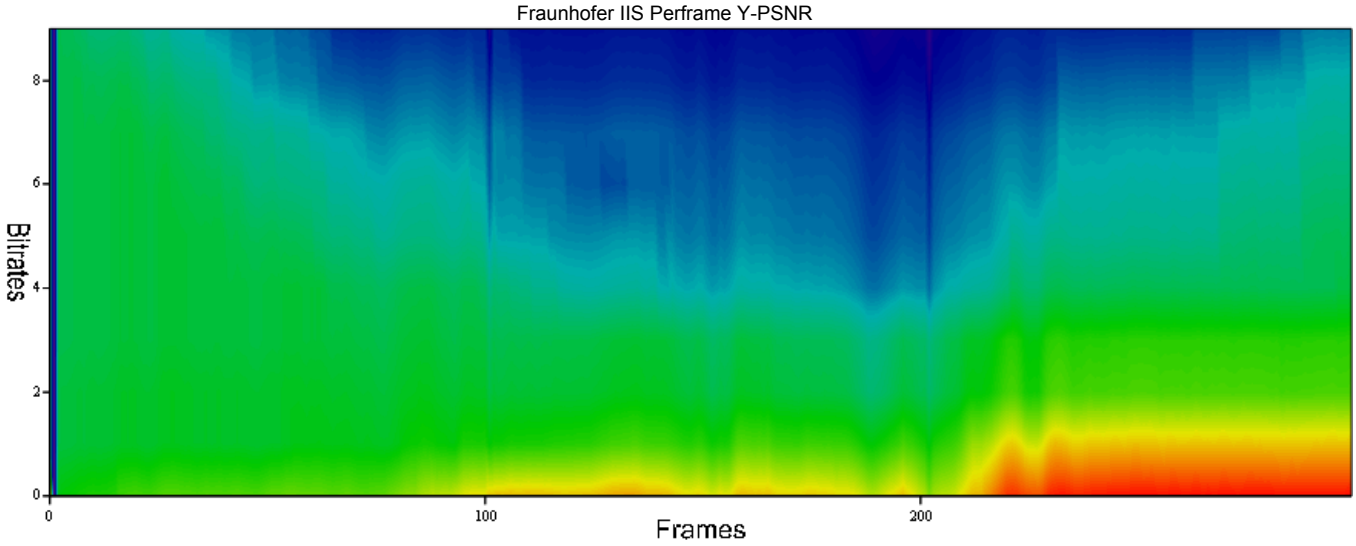
M

Picture 103. Codec Elecard H.264. Sequence “foreman”.



M

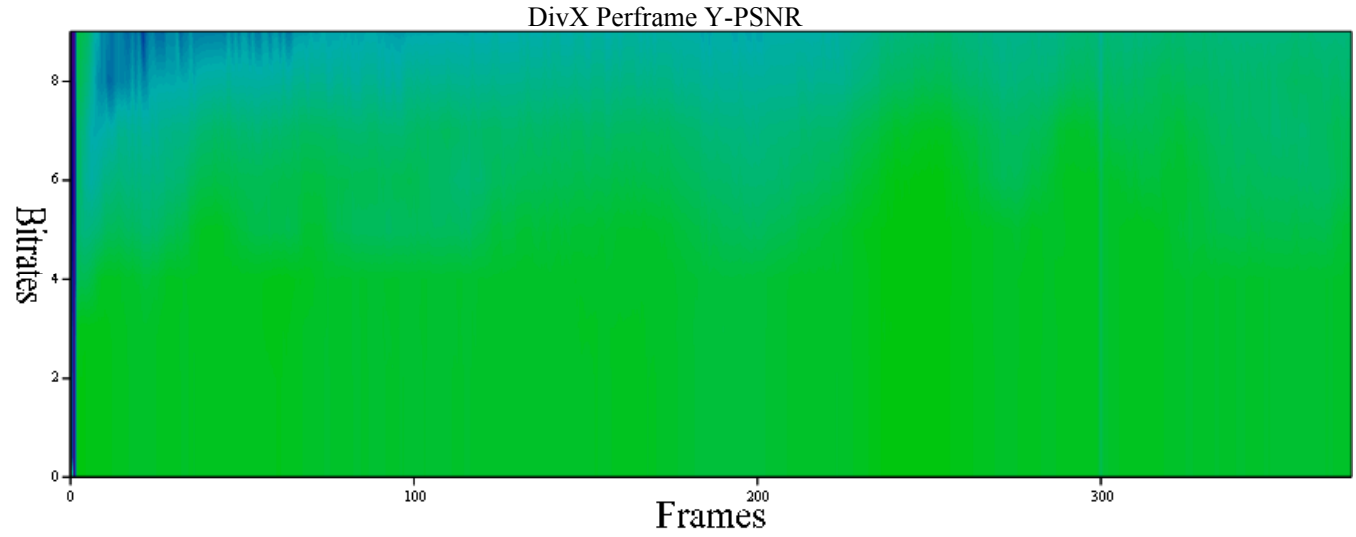
Picture 104. Codec Ateme H.264. Sequence “foreman”.



M

Picture 105. Codec Fraunhofer IIS H.264. Sequence “foreman”.

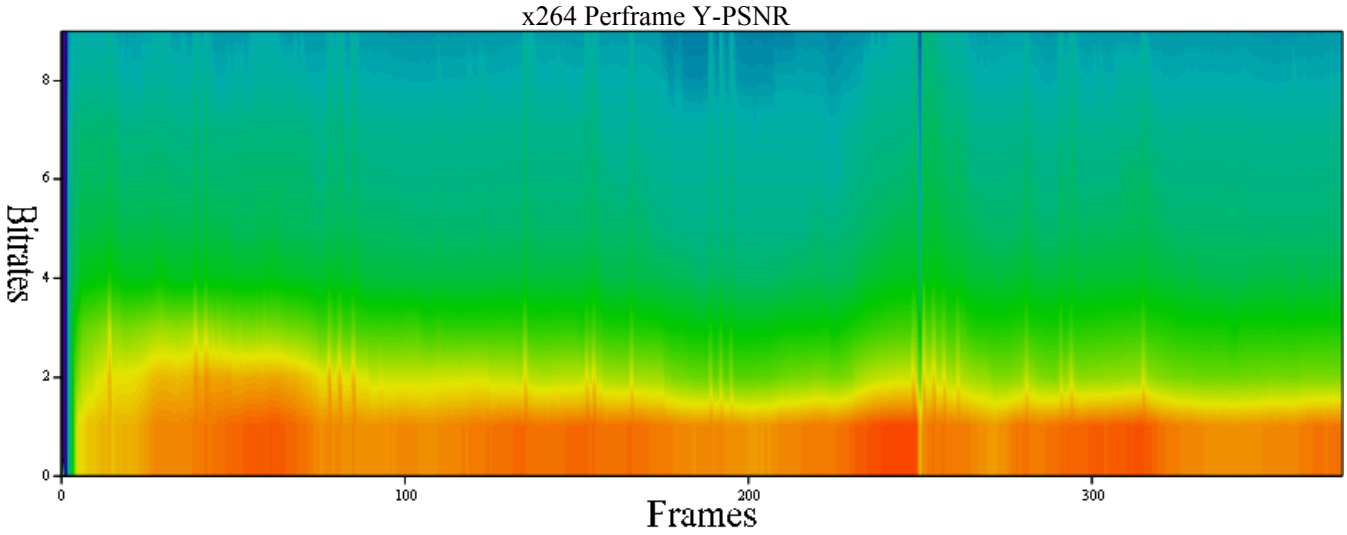
Sequence “bbc”, Y-PSNR, Preset “Best quality”



M

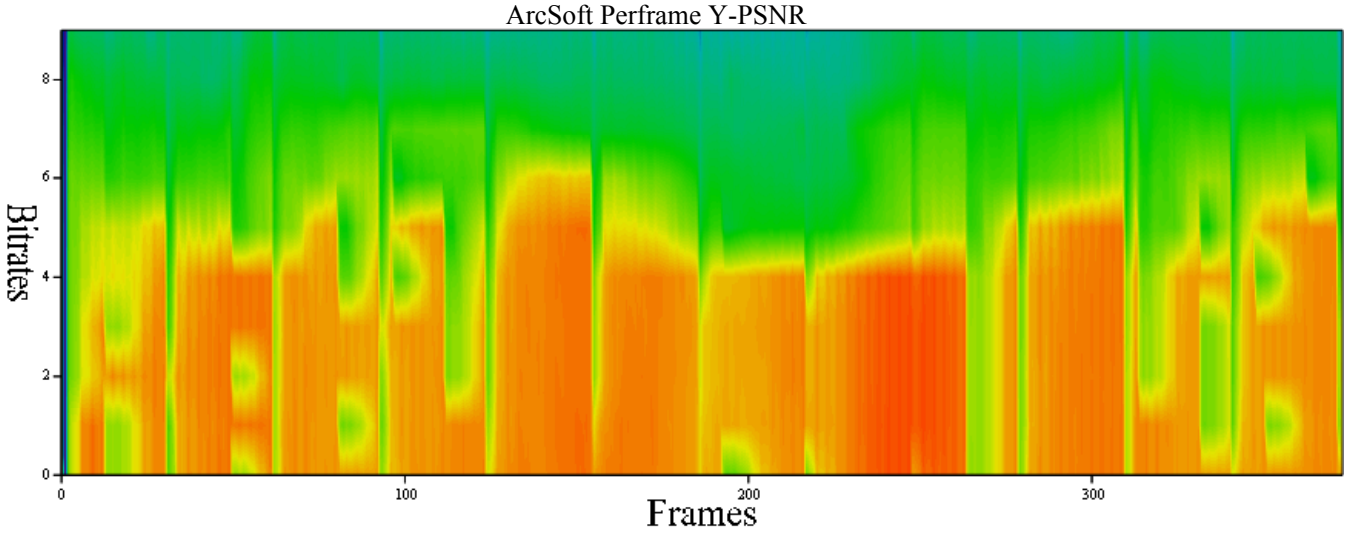
Picture 106. Codec DivX. Sequence “bbc”.

On this graph compression quality does not significantly depend from bit rate, because DivX codec does not maintain low bit rates well, and works badly on high ones.



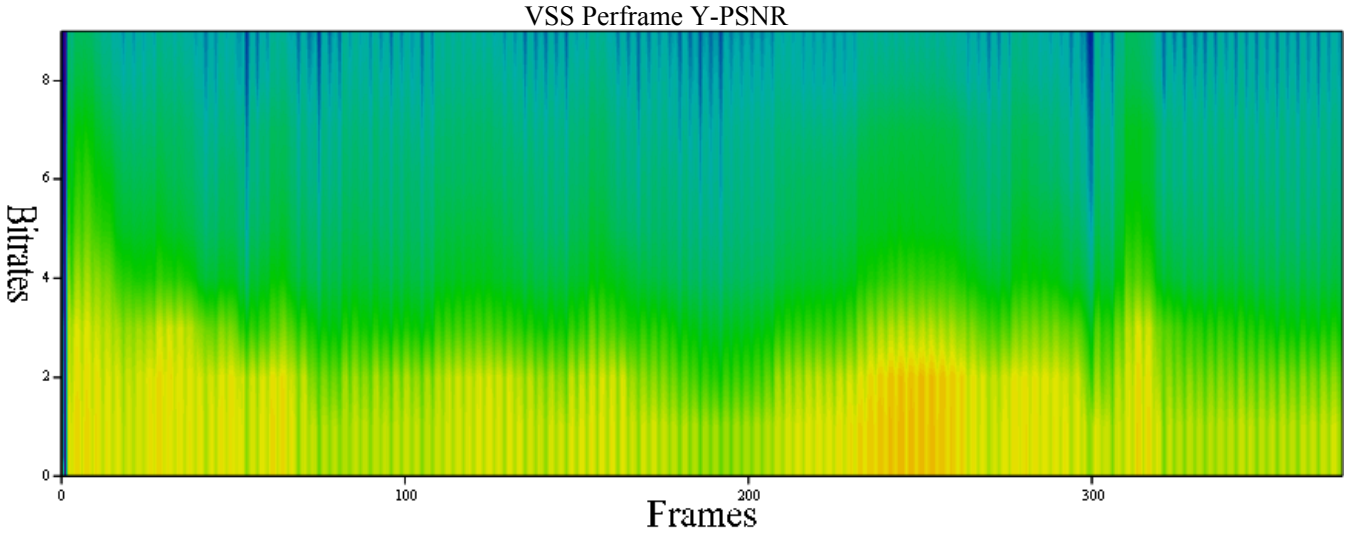
M

Picture 107. Codec x264. Sequence “bbc”.



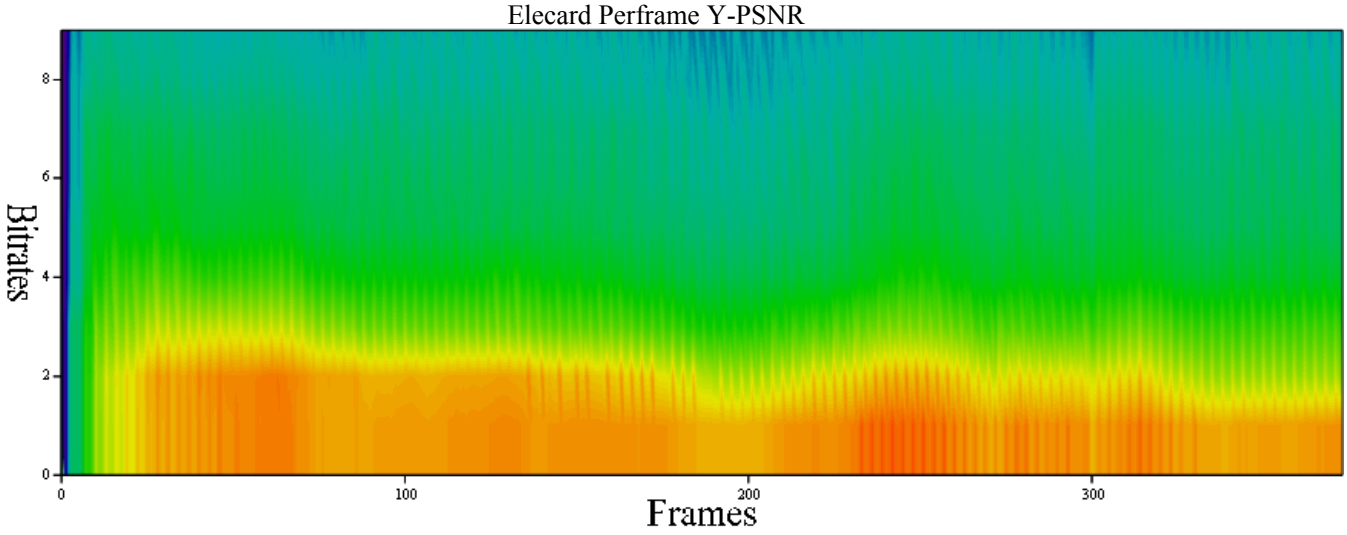
M

Picture 108. Codec ArcSoft H.264. Sequence “bbc”.



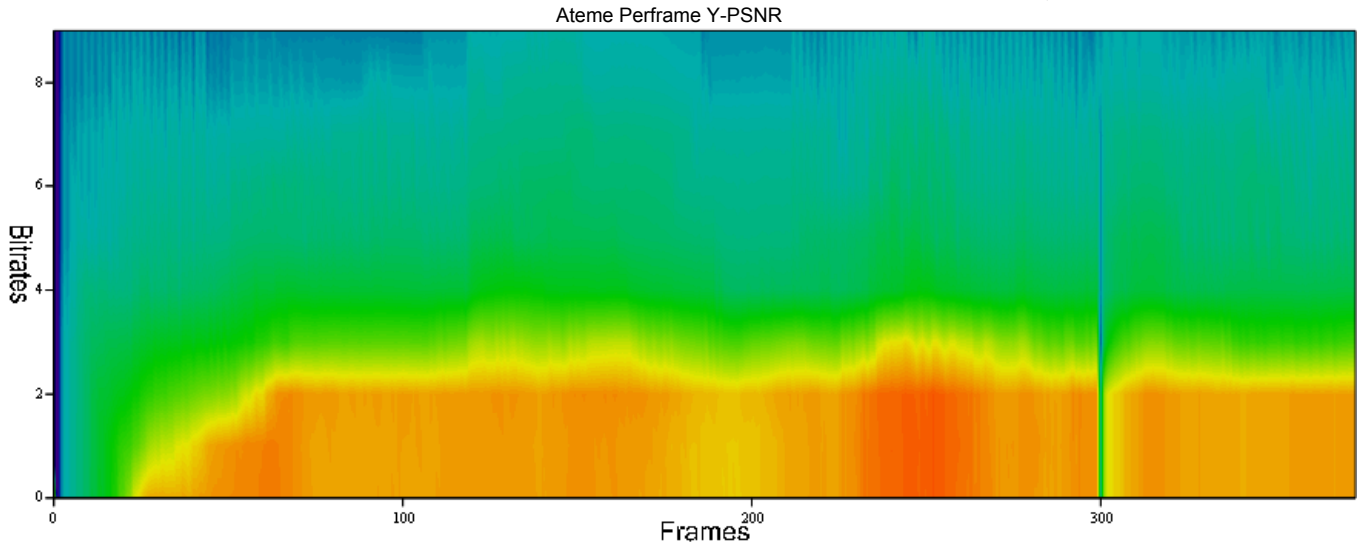
M

Picture 109. Codec VSS H.264. Sequence “bbc”.



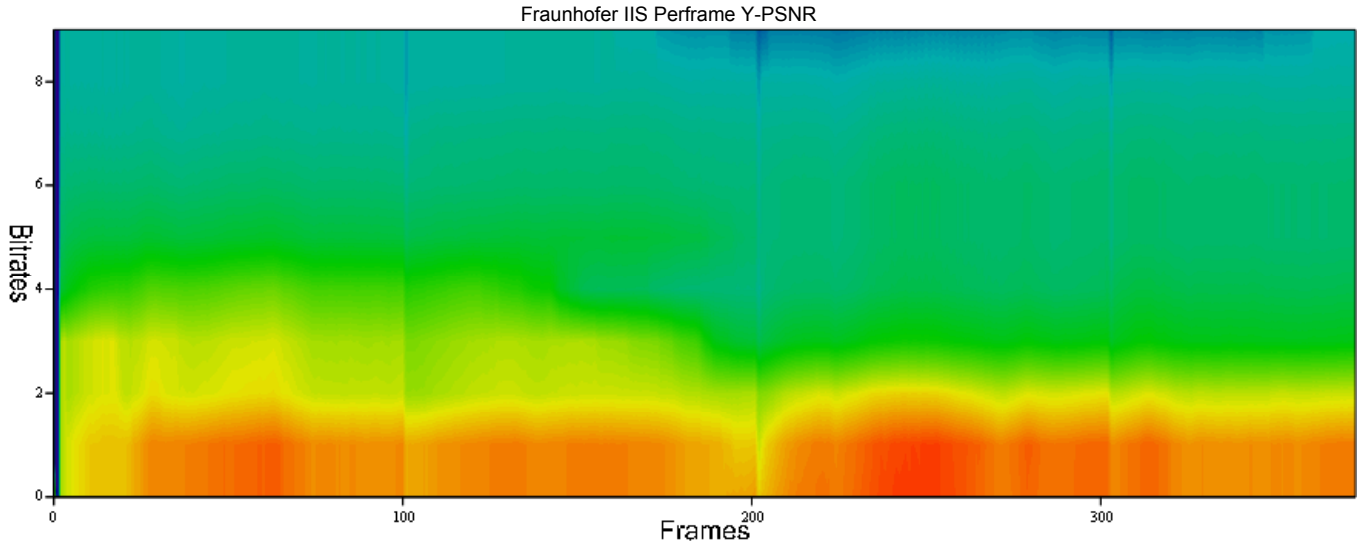
M

Picture 110. Codec Elecard H.264. Sequence “bbc”.



M

Picture 111. Codec Ateme H.264. Sequence “bbc”.



M

Picture 112. Codec Fraunhofer IIS H.264. Sequence “bbc”.

Remarks:

- These graphs show that x264, Elecard, Ateme and VSS codecs use B-frames.
- DivX and Elecard codecs have very interesting B-frames placement algorithm. They change positions of each type of frames depending on bit rate value. This is shown on graphs as vertical wavy lines.
- “bbc” sequence clearly shows problems in bit rate control of ArcSoft codec. It is represented as presence of “holes” on average bit rates. Considering complex circular motion in this sequence, one could suppose presence of problems with motion estimation in this codec.

- Large quality overstating of I-frames in x264 codec (it is also typical for Ateme and Fraunhofer IIS codecs, but at smaller degree) may be caused by inaccurate prediction of current frame type. It is represented as distinct blue lines that appear several times in sequence.

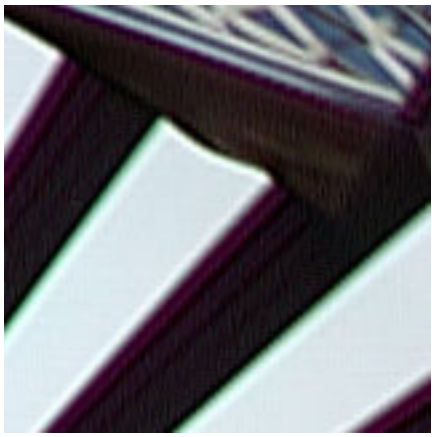
Graph type	Total number of graph	Inserted in this document
Per frame metrics	**1414**	**13 (0.9%)**

Visual comparison

Even in spite of appearance of new metrics, that more adequately reflect human perception of video, subjective video tests still do not lose their importance.

We set ourselves a problem to compare video codecs using only existing objective metrics. But in addition to various graphs that impartially evaluate quality of obtained results, below we give several frames from “bbc” and “battle” sequences, compressed with different codecs.

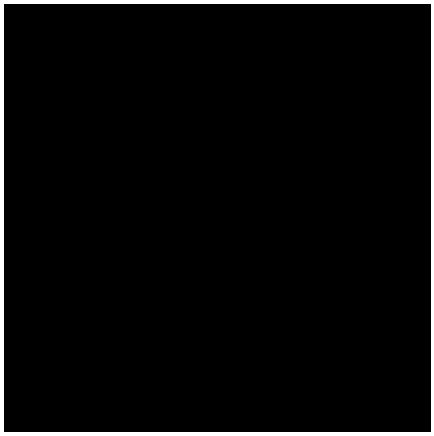
Sequence “bbc”, frame 170, bitrate 1140 kbps



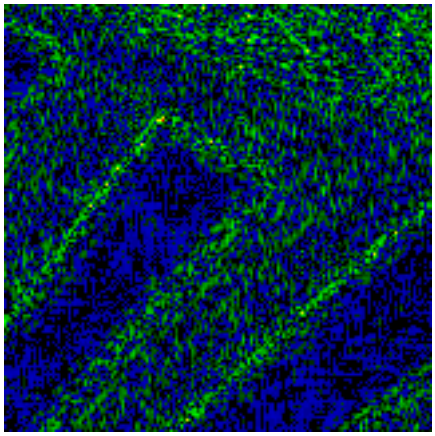
Picture 113. Original

Picture 114. DivX

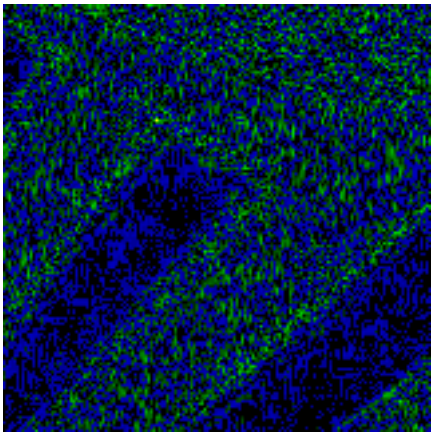
Picture 115. x264



Picture 116. Original. Image differences. Y-PSNR



Picture 117. DivX. Image differences. Y-PSNR

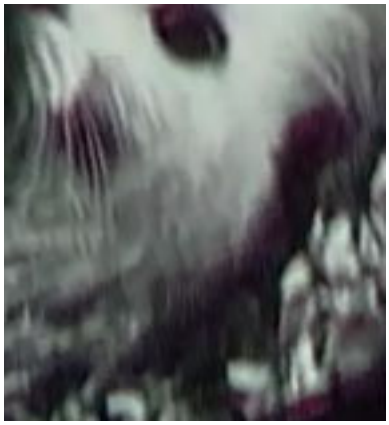


Picture 118. x264. Image differences. Y-PSNR

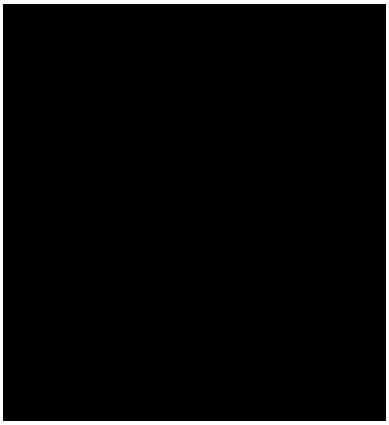
Sequence “bbc”, frame 250, bitrate 1140 kbps

Picture 119. Original

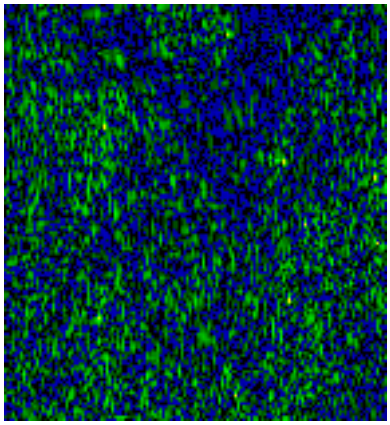
Picture 120. DivX



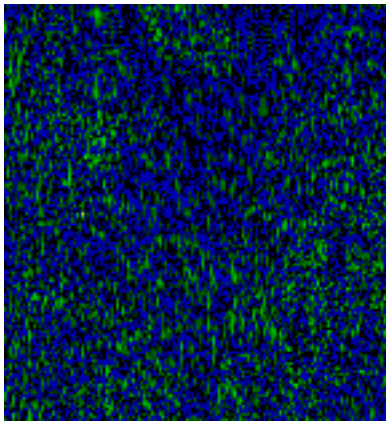
Picture 121. x264



Picture 122. Original. Image differences. Y-PSNR



Picture 123. DivX. Image differences. Y-PSNR



Picture 124. x264. Image differences. Y-PSNR

Sequence “battle”, frame 527, bitrate 700 kbps

Picture 125. Original

Picture 126. DivX

Picture 127. x264

Picture 128. ArcSoft

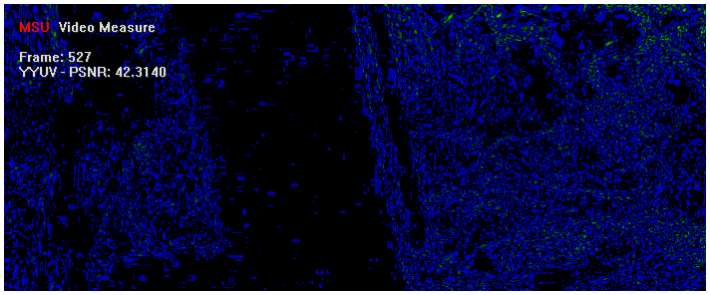
Picture 129. VSS

Picture 130. Elecard

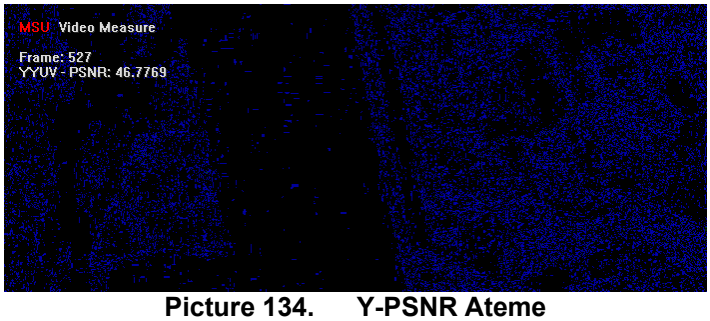
Picture 131. Ateme

Picture 132. Fraunhofer IIS

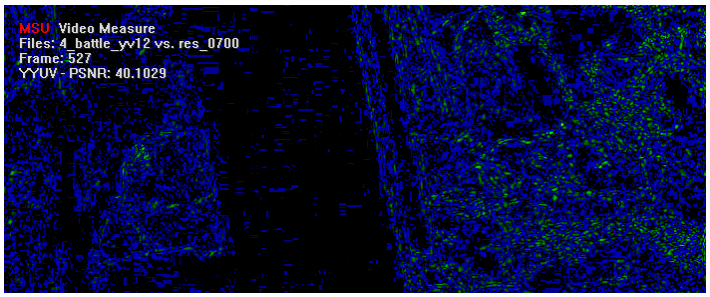
Sequence “battle”, frame 527, bitrate 700 kbps, Y-PSNR



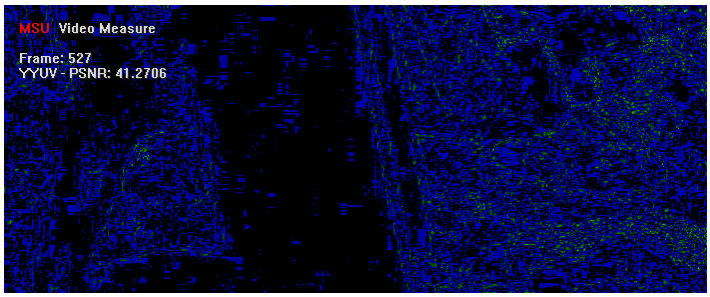
Picture 133. Y-PSNR ArcSoft



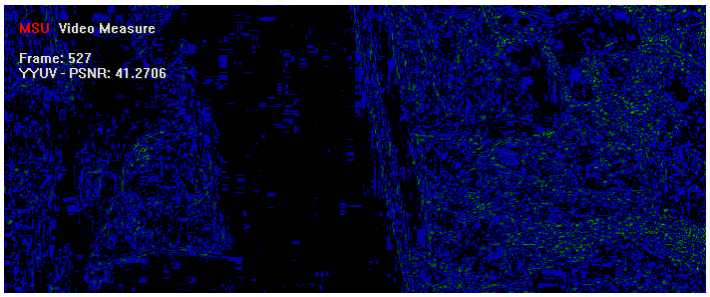
Picture 134. Y-PSNR Ateme



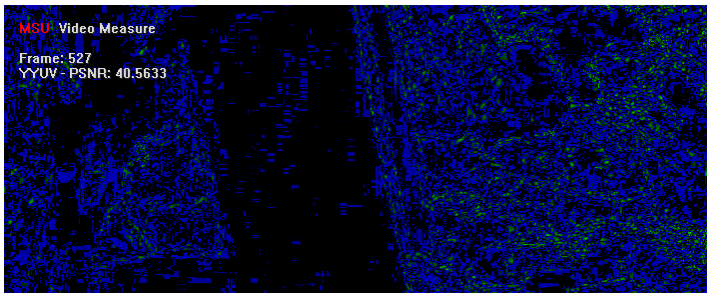
Picture 135. Y-PSNR DivX



Picture 136. Y-PSNR Elecard

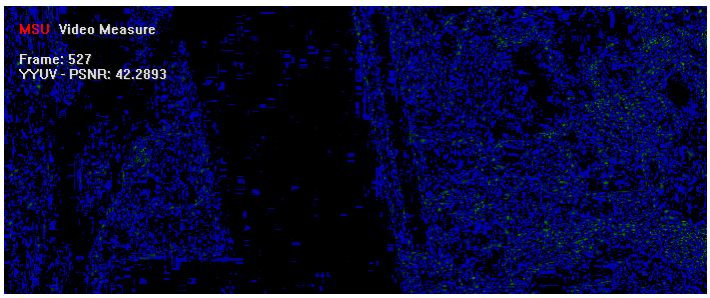


Picture 137. Y-PSNR Fraunhofer IIS



Picture 138. Y-PSNR VSS

Picture 139. Y-PSNR Original



Picture 140. Y-PSNR x264

Informal codec comparison

As it is clearly seen on the graphs, codecs show significantly different results on different sequences. Selected sequences have different types in terms of motion and noise, and this allows checking which codecs are tuned to all types of sequences, and which one - to only one. But also it is very interesting to evaluate codec behavior on all test data.

We considered averaging of results between different sequences as incorrect, and used a method of assigning points to codecs for different measurements. Points are united in arbitrary informal rating in the end.

We tried to reflect real situation on test data as much as possible. We are still developing methodology of our informal comparison. Therefore it is better to consider following results as subjective opinion of comparison authors.

Informal comparison rules

Separate comparisons were carried out for “Best Quality” and “Best Speed” presets. Several graphs were chosen for each preset, on which comparison was held. Obtained marks were summed with some predefined weights.

Codecs received points in a following way:

- If codec is one of the leaders on current graph then it received 3 points.
- If codec is trailing on current graph then it received 1 point.
- Otherwise it is received 2 points.

Following graphs were used for “Best Quality” preset with corresponding coefficients:

- Y-PSNR graphs, coefficient 4.
- U-PSNR graphs, coefficient 1.
- V-PSNR graphs, coefficient 1.
- SSIM graphs, coefficient 1.
- VQM graphs, coefficient 1.
- Bit rate handling graphs, coefficient 3 (denoted in tables as “BH”).
- Graphs of absolute speed, coefficient 1.

Following graphs were used for “Best Speed” preset with corresponding coefficients:

- Y-PSNR graphs, coefficient 1.
- Bit rate handling graphs, coefficient 2 (denoted in tables as “BH”).
- Graphs of absolute speed, coefficient 4.

Average values were computed as arithmetic average of all successfully encoded sequences (i.e. codecs' errors were not taken into account).

Informal comparison results

Preset "Best quality"

Sequence "foreman"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	1	2	1	0	2	3	2
U-PSNR	1	3	1	0	3	3	2
V-PSNR	1	3	2	0	3	3	2
SSIM	1	3	1	0	2	2	2
VQM	1	2	1	0	2	3	2
BH	1	2	2	0	2	2	2
Time	2	2	3	0	2	3	1
Total	13	27	18	0	26	**32**	23

Sequence "susi"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	2	2	1	2	2	3	2
U-PSNR	2	3	1	2	2	2	1
V-PSNR	2	3	1	3	2	2	1
SSIM	2	2	1	2	2	3	2
VQM	2	2	1	2	1	3	2
BH	1	2	3	2	2	2	1
Time	2	2	3	2	2	2	1
Total	21	26	20	25	23	**30**	18

Sequence "bbc"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	1	2	1	2	2	3	3
U-PSNR	1	3	1	2	2	2	3
V-PSNR	2	3	1	2	2	3	3
SSIM	2	2	1	3	2	3	2
VQM	2	2	1	2	2	3	3
BH	1	3	1	2	2	2	3
Time	2	2	3	2	2	2	1
Total	16	29	14	25	24	31	**33**

Sequence "battle"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	2	2	1	1	2	3	2
U-PSNR	1	3	1	2	3	2	3
V-PSNR	1	3	1	2	3	2	3
SSIM	2	2	1	1	2	3	2
VQM	2	2	1	2	2	3	3
BH	1	3	2	2	2	2	3
Time	2	2	3	2	2	2	1
Total	19	29	17	19	26	**30**	29

Sequence "simpsons"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	2	3	1	3	2	3	2
U-PSNR	1	3	1	3	2	2	2
V-PSNR	1	3	1	3	3	2	2
SSIM	2	3	1	3	2	2	2
VQM	2	3	1	3	2	2	2
BH	1	2	2	2	2	2	3
Time	2	2	3	2	2	2	1
Total	19	**32**	17	**32**	25	28	26

Sequence "matrix"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	2	2	1	2	2	3	1
U-PSNR	2	3	1	3	2	2	1
V-PSNR	2	3	1	2	3	2	1
SSIM	2	2	1	2	2	3	1
VQM	2	2	1	2	2	3	2
BH	1	1	2	3	2	2	2
Time	2	2	3	2	2	2	1
Total	21	23	17	28	25	**30**	16

Sequence "concert"

	DivX	x264	ArcSoft	VSS	Elecard	Ateme	Fraunhofer IIS
Y-PSNR	0	2	1	3	1	3	3
U-PSNR	0	3	3	2	2	2	1
V-PSNR	0	3	3	2	2	2	1
SSIM	0	2	1	3	2	1	3
VQM	0	2	1	3	1	2	3
BH	0	2	2	2	1	2	3
Time	0	2	2	2	3	2	1
Total	0	26	20	**30**	17	27	**30**

Results

Codec	Average points	Place
Ateme	29.71	1
x264	27.43	2
VSS	26.5	3
Fraunhofer IIS	25	4
Elecard	23.71	5
DivX	18.17	6
ArcSoft	17.57	7

Preset "Best speed"

Sequence "foreman"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	1	2	1	2	0	3	2
BH	1	2	3	2	0	2	1
Time	2	2	3	2	0	2	1
Total	11	14	**19**	14	0	15	8

Sequence "susi"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	2	2	1	2	2	3	2
BH	1	2	2	3	3	2	1
Time	2	2	3	2	2	2	1
Total	12	14	**17**	16	16	15	8

Sequence "bbc"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	1	2	2	1	2	3	3
BH	1	2	2	1	2	2	2
Time	2	2	3	2	2	2	1
Total	11	14	**18**	11	14	15	11

Sequence "battle"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	2	3	1	2	1	3	2
BH	1	2	2	2	2	2	3
Time	2	2	3	2	2	2	1
Total	12	15	**17**	14	13	15	12

Sequence "simpsons'

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	2	2	1	2	1	3	3
BH	1	2	2	2	2	2	3
Time	2	2	3	2	2	2	1
Total	12	14	**17**	14	13	15	13

Sequence "matrix"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	3	2	2	2	2	2	2
BH	1	2	2	3	3	2	2
Time	2	2	3	2	2	2	1
Total	13	14	**18**	16	16	14	10

Sequence "concert"

	DivX	x264	ATI	ArcSoft	VSS	Elecard	Fraunhofer IIS
Y-PSNR	0	3	1	2	2	3	3
BH	0	2	1	2	2	1	3
Time	0	2	3	2	2	2	1
Total	0	**15**	**15**	14	14	13	13

Results

Codec	Average points	Place
ATI	17.29	1
Elecard	14.57	2
VSS	14.33	3
x264	14.29	4
ArcSoft	14.14	5
DivX	11.83	6
Fraunhofer IIS	10.71	7

General conclusions

- Comparing to previous year comparison, the level of H.264 standard codecs has increased significantly. One year ago best codecs of old MPEG4-ASP standard were comparable to best codecs of H.264 standard, now this situation greatly changed. DivX codec is inferior to most codecs of new standard, even despite considerable growth of its quality comparing to previous version.
- It is rather difficult to choose absolute leader by quality among codecs of new standard. x264 and Ateme codecs showed approximately equal best results.
- Obvious leader by speed is codec from ATI company. It was significantly faster than the others while not always showing worst results.

If you are interested in your codecs' testing or tuning, please write to us at videocodec-testing@graphics.cs.msu.ru

About us (Graphics & Media Lab Video Group)

Graphics & Media Lab Video Group is a part of Graphics & Media Lab of Computer Science Department in Moscow State University. The history of Graphics Group began at the end of 1980's. Graphics & Media Lab was officially founded in 1998. Main research directions of the lab lie in different areas of Computer Graphics, Computer Vision and Media Processing (audio, image and video processing). Some of research results were patented, other results were presented in a number of publications.

Main research directions of Graphics & Media Lab Video Group are video processing (pre-, post- and video analysis filters) and video compression (codecs' testing and tuning, quality metrics research, development of codecs).

Our main achievements in **video processing:**

- High quality industrial filters for format conversion including high quality deinterlacing, high quality frame rate conversion, new fast practical super resolution, etc.
- Methods for modern TV-sets: big family of up-sampling methods, smart brightness and contrast control, smart sharpening, etc.
- Artifacts' removal methods: family of denoising methods, flicking removal, video stabilization with frame edges restoration, scratches, spots, drop-outs removal, etc.
- Specific methods like: subtitles removal, construction of panorama image from video, video to high quality photo, video watermarking, video segmentation, practical fast video deblur, etc.

Our main achievements in **video compression:**

- Well-known public comparisons of JPEG, JPEG-2000, MPEG-2 decoders, MPEG-4 and annual H.264 codec's testing; also we provide tests for "weak and strong points of codec X" for companies with bugreports and codec tuning recommendations.
- Our own video quality metrics research, public part is MSU Video Quality Measurement Tool and MSU Perceptual Video Quality Tool.
- We have internal research and contracts on modern video compression and publish our MSU Lossless Video Codec and MSU Screen Capture Video Codec – codecs with ones of the highest compression ratios.

We are really glad to work many years with companies like Intel, Samsung, RealNetworks and others.

A mutual collaboration in areas of video processing and video compression is always interesting for us.
E-mail: video@graphics.cs.msu.ru

MSU Video Quality Measurement Tool

MSU Graphics & Media Lab. Video Group.

GRAPHICS & MEDIA LAB VIDEO GROUP

Main Features

1. 12 Objective Metric + 5 Plugins

PSNR several versions,
MSAD,
Delta,
MSE,
SSIM Fast,
SSIM Precise,
VQM,
MSU Blurring Metric,
MSU Brightness Flicking Metric,
MSU Brightness Independent PSNR,
MSU Drop Frame Metric,
MSU Noise Estimation Metric,
MSU Scene Change Detector,
MSU Blocking Metric.

2. More Than 30 Supported Formats, Extended Color Depth Support

*.AVI,
*. YUV:
YUV,
YV12,
IYUV,
UYVY,
Y,
YUY2,
*.BMP,
*.AVS:
*.MOV,
*.VOB,
*.WMV,
*.MP4,
*.MPG,
*.MKV,
*.FLV,
etc.,

Extended Color Depth:
P010, P014,
P016, P210,
P214, P216,
P410, P414,
P416,
P410_RGB,
P414_RGB,
P416_RGB.

3. Multi-core Processors Support

MMX, SSE and OpenMP Optimizations

4. Comparative Analysis

Comparison of 3 files at a time

5. ROI Support

Metric calculation for ROI (Region of Interest)

6. GUI & Batch Processing

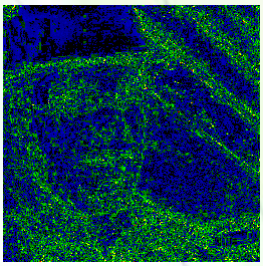
GUI and command line tools

7. Plugins Interface

You can easily develop your own metric

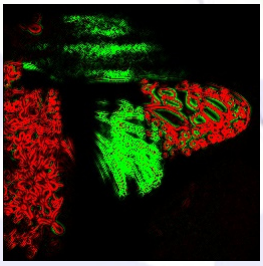
Visualization Examples

Allows easily detect where codec/filter fails

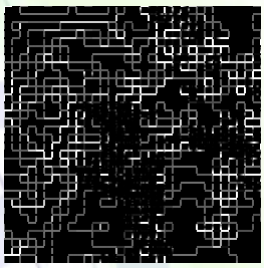


Y-YUV PSNR

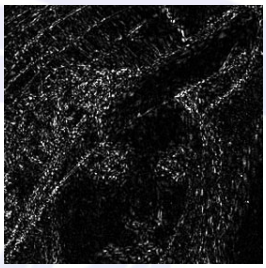
Y-YUV Delta



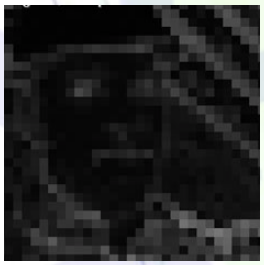
MSU Blurring Metric



MSU Blocking Metric



Y-YUV MSE



VQM

8. Universal Format of Results

Results are saved in *.csv files

9. HDTV Support

10. Open-Source Plugins Available

11. Metric Visualization

Fast problem analysis, see examples above.

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

Tool was downloaded more than 100 000 times!

Free and Professional versions are available

Big thanks to our contributors:

Apple Inc.
intel
skype
NVIDIA.
NASA
DOLBY
CISCO
Sun microsystems
NBC UNIVERSAL
DivX®