Video Codec Scoring Based on Modified Natural and Artificial Video Sequence Processing

Alexander Parshin, Dmitriy Vatolin, *Member, IEEE,* Ilya Brailovskiy and Philip Corriveau

***Abstract*—In this paper we propose a novel approach to video codec evaluation, comparison and testing based on preparation of specific video sequences followed by efficiency evaluation. Several strategies for applying natural video sequence modification as well as fully artificial video clip creation are suggested and studied. Experimental results with measurements and scoring summary for several H.264 and MPEG-4 ASP codecs are presented and discussed.**

Index Terms*—video codec quality analysis, objective quality metrics, video sequences modifications.*

I. Introduction

COMPRESSION of video data is a key processing technology that enables distribution of media for a wide range of applications. The ability to compress video without noticeable or with acceptable subjective quality degradation is one of the main features that aided in the explosion of digital video into the existing consumer base and onto emerging markets.

There are currently more than ten international standards for digital video coding, and several more are under development. These standards mostly specify the syntax and semantics of coded data streams. The decoding and (moreover) encoding algorithms usually are not specified by the standards, thus allowing multiple implementations of one standard to co-exist.

Video codecs can be compared using different criteria, such as the supported coding feature set, codec control options, performance, introduced delay and various other characteristics. In this paper we present a method that allows for automatic evaluation of codec quality from the end-user perspective. We investigate a method for producing a calculated score that allows comparison of different codecs. This evaluation method is designed to be codec-, standard- and vendor-neutral.

There are several characteristics of a codec that can be manipulated to achieve the required trade-offs for the visual quality provided by a specific compression rate. When an acceptable end-user visual quality can be achieved with a lower compression rate, it not only saves needed bandwidth, but it is also more efficient with regard to processing. The methodology that we present allows codec comparison with regard to manipulating this compression feature to achieve lower distortion for a given bitrate budget.

Another important dimension of end-user codec quality perception is performance. It is obvious that the faster a user can get the desired result, the better. In this analysis, however, we assume that all codecs that are to be compared or ranked will have the same level of performance or, at least, will be within the same performance class. For example, we do not compare online and offline codecs, since they usually have very different performance levels (offline codecs are much slower).

Considering the nature of video codecs comparison, it is clear that there is a need to have access to both the original and processed video sequences for effective codec analysis. Our goal is to examine full encoding operations, which requires that we have the original raw video sequences and that we submit all unprocessed data to the encoder. The output of the encoder is the processed data, which leaves us with both the original and processed video sequences. That allows us to apply so-called reference metrics like PSNR and SSIM [8] or to apply VQM methodologies [15], [16] to evaluate the subjective quality of the encoded video. The ability to use number of metrics and methodologies is critical, since one of the main problems of objective analysis methods is the absence of adequate metrics that correlate well with the human visual system. PSNR and other reference metrics do not guarantee a result that coincides with real human visual perception [5], but they do provide a reasonable correlation.

In addition, scoring results for a given codec should be normalized to allow comparison of different codecs. In this context we consider a reference (or model) codec that has the desired characteristics. The parameters of this model codec are used as normalization parameters for the codec under test. For example, the reference codec could have two-times-better quality than another codec for given bitrates. At the same time,

Manuscript received April 29, 2008. This work was supported in part by the Russian Foundation for Basic Research under Grant № 07-01-00759-a.

A. Parshin is with the Moscow State University, Moscow, Russia (phone: +7-495-939-0190; e-mail: aparshin@graphics.cs.msu.ru).

D. Vatolin is with the Moscow State University, Moscow, Russia (phone: +7-495-939-0190; e-mail: dmitriy@graphics.cs.msu.ru).

I. Brailovskiy is a Media Architect with Intel Corporation, Moscow, Russia, (8-495-641-4662; e-mail: ilya.v.brailovskiy@intel.com)

P. Corriveau is a Principal Engineer with Intel Corporation, Hillsboro, USA, (+1-503-799-9284, e-mail: philip.j.corriveau@intel.com)

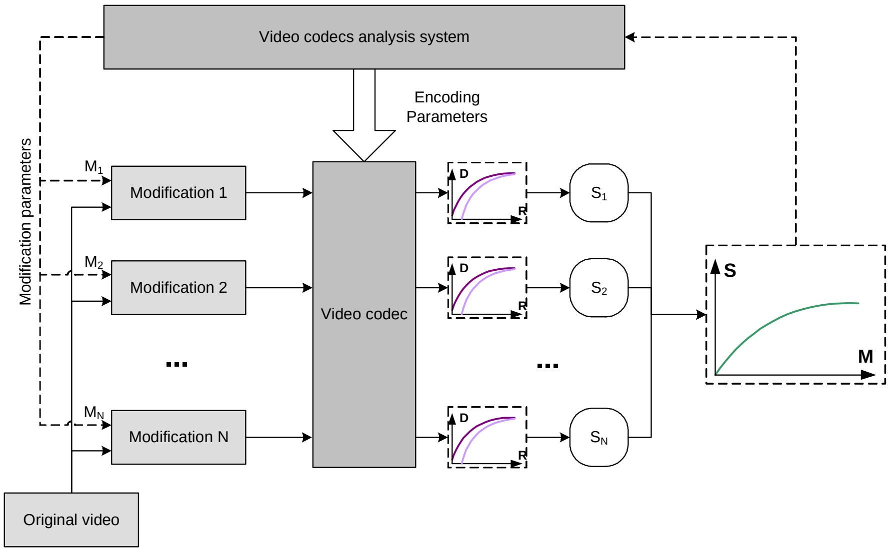

Fig. 1. General scheme of video codec analysis.

the reference codec should be within some valid quality range for current industry-leading codecs—it should not be orders of magnitude better. This is more of a pragmatic consideration, since, otherwise, normalization would not be very useful for visual comparison.

According to authors with expert knowledge in the area of video compression, and based on various other research [1]–[3], there are three major coding blocks that define the quality of the codec:

- Motion compensation (MC) algorithm;
- Macroblock decision algorithm (macroblock-type decision selection and higher-level frame-type selection);
- Rate control (quantizer selection for both frame and macroblock levels).

These algorithms are critical and, in most cases, fully define codec quality performance. Our goal is then to create a set of tests that assess each aspect of the defining characteristics. We do not use any explicit knowledge or implementation details about specific codecs. Instead, in this paper we make some general assumptions about the codecs: the codecs perform block-based transforms, quantization and motion compensation. The proposed methodology does not assume any knowledge about specific bitstream syntax for a given standard, so the methodology can be used for multiple video coding standards.

The remainder of the paper includes details of the proposed methodology in Section II. In Section II.A attention is focused on the proposed video sequence modifications. RD-characteristic calculation and the algorithm for comparison of RD curves are considered in Sections II.B and II.C, respectively. In Section II.D the algorithm for analyzer scoring calculation is described in detail. Results of the application of the proposed methods are summarized in Section III, based on analysis of motion compensation algorithms (Section II.A) and overall codec efficiency (Section III.B). Combination of analyzer estimates as a final scoring stage is described in Section IV.

II. Video Codec Analysis

The overall high-level scheme of the proposed method is depicted in Fig. 1. The first stage involves taking a video sequence and creating modifications of the video using known modification parameters M_i. Modified sequences can be created using natural video feeds or using totally artificial feeds. The modified sequences are submitted to the codec under test, which encodes and decodes each of modified streams. The encoded results are then compared with the results of the original video sequence encoding. These results are also compared with the results of the coding using the reference video codec to produce baseline coding results S_i. The video codec analysis system takes the array of results, S, and the array of modification parameters, M, as the input and calculates the estimate for a sequence modification. The final step is combination of estimates for several different modifications (or codec analyzers) to produce the final codec score. Each stage of this analysis is discussed in further detail in the following sections.

A. Modified video sequences

When determining codec performance, one is able to investigate the common characteristics of the codecs. Not all codecs are the same, but there are baseline features that have a common effect on the resulting video quality. The motion compensation, quantization parameter selection and macroblock mode decision algorithms ultimately define the coding quality results in most cases.

Although we have not mentioned any other encoding algorithms that might affect visual quality, these other algorithms are considered indirectly in our approach. For example, noise cancellation is also important to the end-user

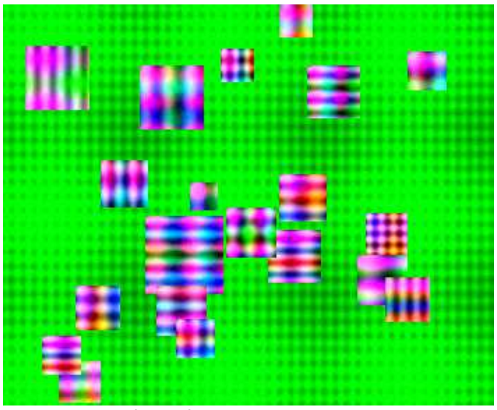

Fig. 2. One frame from a synthetic motion sequence.

perception and can be evaluated separately. At the same time, the quality of the noise cancellation can be calculated using the quality of the motion estimation and the macroblock decisions. Codecs with better noise cancellation algorithms end up having more robust motion estimation.

The contribution of each of these three parts (motion compensation, rate control and mode decisions) to the final quality varies. The contribution is defined not only by the specifics of the codec implementation, but also by the nature of the sequence being encoded. Video sequence modifications of the natural and artificial sequences are designed in such a way that they impact a particular part of the encoding algorithm. By looking at the absolute and relative variations of coding efficiency with respect to the extent of the modification, we can try to draw some conclusions about the efficiency of different codec algorithms.

We implemented several modifications of various natural sequences and one synthetic sequence. The following modifications are discussed further in this paper:

- **Frame decimation.** From the source video sequence containing frames $\{F_i\}$, $i = 1,..,P$, we keep each Nth: $i = 0 \pmod N$ and drop the rest of the frames. The modification parameter in this method is the dropping factor *N*. Decimated video sequences have lower correlation between frames in the time domain. One of the main algorithms where efficiency is highly dependent on the correlation between subsequent frames is motion estimation. So, the coding efficiency of such highly decimated video sequences mostly depends on the motion estimation algorithm efficiency.
- **Noisy frame insertion.** Frames with a uniform noise distribution (all generated values have equal probability) are inserted into the original video sequences. Each noisy frame is inserted after every *N* frames. The modification parameter in this method is the number of noisy frames added, *N*. Insertion of vastly different frames into the source sequence strongly impacts the frame-level or group-of-frames-level (GOP-level) bitrate control algorithms. A similar situation occurs when there are highly dynamic scenes in the video. This method is designed to test the stability of frame-level encoding algorithms.
- **Noisy macroblock insertion.** In each frame, at a random position *N*, noisy 16x16-pixel macroblocks are added. The modification parameter in this method is the number *N* of noisy macroblocks added. Significantly varying macroblocks have a tremendous influence on the frame statistics. Codecs with good macroblock-level algorithms will work mostly effectively with such videostreams modifications. This modification aims to test macroblock-level coding efficiency.
- **Spatially altered noise.** Gaussian-distributed noise $N(0, \sigma)$ is added to each pixel of each frame. The noise variation σ is adjusted linearly from the first to the last pixels in the frame. As a result, the last pixels become much nosier relative to the first pixels. The modification parameter here is the noise parameter for the last pixel, *SIGMA_MAX*. The target codec component of this modification is the macroblock-level rate-control algorithm. By taking into account different levels of noise for different macroblocks, such algorithms can significantly improve encoding results.
- **Synthetic stream with moving objects.** This video sequence consists of a static background and a number of moving objects. The background texture is generated using the following formula:

$$C(x,y) = \frac{200}{8}(\sin(fr_{x1}\cdot x) + \sin(fr_{x2}\cdot x) + ,$$
$$+\sin(fr_{y1}\cdot y) + \sin(fr_{y2}\cdot y) + 4)$$

where $C(x,y)$ is the color of pixel (x,y) and fr_{x1}, fr_{x2}, fr_{y1} and fr_{y2} are the randomly selected frequencies (same for all frames in the sequence). Moving objects are represented by squares with the following synthetic texture (for each colorplane):

$$C(x,y) = \frac{255}{4}(\sin(fr_x\cdot x) + \sin(fr_y\cdot y) + 2)\cdot$$

Object size is initially defined randomly (the size depends only on target video resolution). Each object in a given frame can be described by its position (x,y) and speed (v_x,v_y). The initial position is random, and the initial speed is selected randomly from the interval *[0, MAX_SPEED]*, where *MAX_SPEED* is a sequence parameter that defines motion complexity. The position of the object in the next frame $(i+1)$ is defined using the current speed of the object:

$$x_{i+1} = x_i + v_x^i$$
$$y_{i+1} = y_i + v_y^i$$

The object speed is updated in two stages:

1. Add a random value from the interval *[-MAX_SPEED/2, MAX_SPPED/2]* to each speed component.

2. Add the correlation component to the speed vector to emulate correlated motion in the scene.

An example of a frame from a generated sequence is depicted in Fig. 2. The use of these synthetic sequences is intended to test a codec's motion compensation algorithm. The constant texture both of the background and objects allows for encoding ideally compensated frames very efficiently. Thus, the main reason for differences in codec results in this case is the motion compensation algorithm.

- **Moving object tail-area analysis.** A synthetic sequence with just a small number of moving objects is generated in a manner similar to that of the previous modification. The modification parameter in this case is again the object speed. The purpose of this modification is to analyze quality differences in the tails of objects (those regions covered by the objects in the previous frame) and the average quality of the entire frame. Quality in the tail areas depends mostly on the mode decision algorithm of codec under test.

Many other modifications of natural and synthetic sequences can be proposed to analyze video codecs. The advantage of natural sequence modification is more adequate results in the target video codec application area; the advantage of synthetic motion modifications is improved sequence parameter control and additional semantic information about sequence structure.

B. RD curves

We consider the dependence of distortion on the compression rate (rate-distortion curve, or RD curve) as one of the main characteristics of the coding results for a video sequence. In rate-distortion theory, the dependence of encoding size (or encoding bitrate) on distortion is defined in following way [6]:

$$R(D) = \min_{p(\hat{x}|x):\sum_{(x,\hat{x})} p(x)\,p(\hat{x},x)\,d(\hat{x},x)\le D} I(X;\hat{X}),$$

where *R(D)* is the RD function, *X* is the source signal, $\hat{X}$ is the decoded signal, *I* is the full information, *p* is the probability density of the signal and *d* is the distance between the source and decoded signals according to the metric.

This formula is used to find the code for a signal (which is explicitly defined by conditional probability $p(\hat{x}\,|\,x)$) such that the average distortion will not exceed a given threshold *D* and the transmission rate ($I(X,\hat{X})$) of the signal is minimized.

There are numerous analytical expressions for this function that are applicable to different types and distributions of data. Some of these expressions are successfully used in codec control algorithms based on RD models [7]. Nevertheless, in the case of evaluation of total quality, such analytical expressions are not common. This situation can be explained by the high correlation of source data and the high complexity of the video codecs.

Approximation of the relationship between compression coefficients and resulting distortion was carried out using piece-wise linear functions. For building RD curves, the codec was run several times with various target bitrates. After decoding, we obtain information about introduced distortion and achieved bitrate (or actual compression rate). It is worth mentioning that the actual bitrate was significantly different from the target bitrate for several of the codecs that were tested.

For quality assessment (as a measure of introduced distortion) we used the PSNR and SSIM metrics [8]. PSNR (peak-to-signal noise ratio) is the classical measure of the difference between two signals:

$$PSNR(X,Y) = 10\cdot\log_{10}\frac{255^2\cdot N\cdot M}{\sum_{i=1}^{n}\sum_{j=1}^{m}\left(X_{i,j}-Y_{i,j}\right)^2}.$$

Here *X* and *Y* are compared images with *MxN* resolution.

The SSIM metric takes into account three components of an image: luminance change, contrast change and variation of the image structure. In its final form, the SSIM metric for signals *X* and *Y* can be represented by the following formula:

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

where $\sigma_{xy} = \sum_{i=1}^{N}\omega_i\left(X_i - \mu_x\right)\left(Y_i - \mu_y\right)$, $\mu_x = \sum_{i=1}^{N}\omega_i X_i$, $\sigma_x = \left(\sum_{i=1}^{N}\omega_i\left(X_i - \mu_x\right)\right)^{\frac{1}{2}}$, C_1 and C_2 are constants and w_i is the set of coefficients of the smoothing filter. More details on the SSIM metric can be found in [8].

Note that the quality metric is a parameter of the proposed methodology. Instead of the quality metric PSNR used in the present analysis, SSIM and VQM [15]–[18] can be later applied for more adequate visual quality estimates. Additionally, subjective testing results can be used in lieu of automated testing results.

After several trials of the video codecs using various bitrates for each modification of the source video sequence, we obtain a set of numbers *(R, D)*, where *R* is the actual bitrate and *D* is some quantitative measure of the introduced distortion. This data is an approximation of the RD curve for the codec using the modification under considered. The RD curve between the obtained RD points is approximated linearly (without any extrapolation).

C. RD curves comparison

The next phase involves comparison of the two RD curves to obtain a single value for each modified sequence. The geometric mean of the bitrate ratio for a specific quality was also used for comparing two RD curves [9]:

$$S_{1,2}^{[a,b]} = \exp\left\{\frac{1}{b-a}\int_a^b \ln\left(\frac{R_1(D)}{R_2(D)}\right)dD\right\}, \tag{1}$$

where $R_1(D)$ and $R_2(D)$ are the RD curves (which relate the dependence of the rate to the amount of distortion) under

TABLE I
FINAL RESULTS OF MOTION COMPENSATION ALGORITHM ANALYSIS USING FRAME DECIMATION MODIFICATION

Algorithm	Bitrate Ratio	Frame Decimation Score
ESA	98.9%	0.588
UMH	99.1%	0.584
HEX	100.0%	0.612
DIA	101.0%	0.621

Lower values correspond to better video coding results.

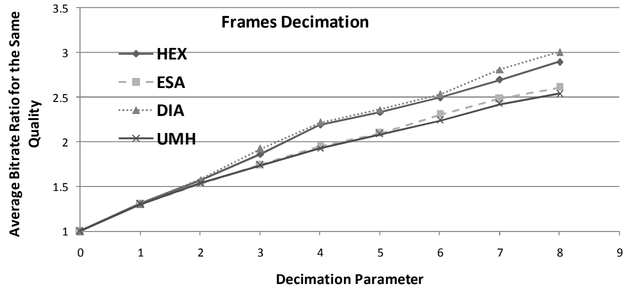

Fig. 3. Results of motion estimation algorithms' analysis using sequence modification with frames decimation.

consideration and *[a,b]* is the range of the quality metric that we use to conduct our comparison. The estimate value $S_{1,2}^{[a,b]}$ characterizes the average ratio of bitrate,for a same quality for a set amount of introduced distortion, that can be achieved by a video codec with corresponding RD characteristics. Note that, in the case of piece-wise linear approximation of the RD-curve functions, it is the formula above can be calculated analytically.

D. Estimate calculation for one modification

The RD-curve comparison phase yields a single value for each modified sequences, with the value characterizing video codec efficiency for a given modified sequence. The main goal of this stage is to analyze these numbers and create a summary video codec estimate. The method proposed in this paper is based on an analysis of the change in codec behavior with respect to the increase in processed video sequence complexity, all in comparison with the reference (or model) codec behavior. The reference codec is also run though the same tests as the codec under examination. As the results we got values *(M, S^r)* for the reference codec and *(M, S^t)* for the codec under test. The overall estimation calculation for the codec is performed using the following comparison function:

$$Q = Q(M, S^r, S^t)$$

Empirical experiments demonstrated that for all tested modifications, the function *S(M)* is very nearly linear. As a result, we can use the difference in slopes for the linear approximation (calculated by a least-mean-squares method) for these functions as an approximation of the difference in the rates of change of the efficiency. The average bitrate ratio $S_{ref,tested}^{[a,b]}$ (see (1)) for the original sequences is used to consider

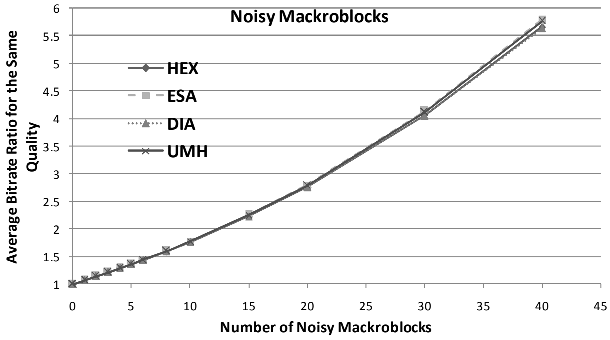

Fig. 4 Analysis of motion estimation algorithms using noisy frame insertion modification.

TABLE II
TESTED CODECS

Encoder	Preset	Description
x264 [10]	default	Default codec parameters
	ref_4	Four reference frames are used for ME
	subme_7	Maximum complexity of rate control algorithms
IPP H.264 [12]	–	GOP structure: IBBPBBP...; 6 reference frames; frame-level CBR
XviD [11]	–	-max_bframes 2 -quality 6 -vhqmode 1 -bvhq -qpel –turbo -single

reference codec quality. Linear combination of the values above is used as the estimate:

$$Q(M, S^r, S^t) = (1-\gamma)\big(\alpha(M, S^t) - \alpha(M, S^r)\big) + \gamma S_{ref,tested}^{[a,b]}, \quad (1)$$

$$\alpha(M, S) = \frac{n\sum_i M_i S_i - \sum_i M_i \cdot \sum_i S_i}{n\sum_i M_i^2 - \left(\sum_i M_i\right)^2},$$

where γ is a constant in the range *[0, 1]* and *α(M,S)* is the slope of the approximating line (the average rate of change of the coding efficiency).

III. ANALYSIS RESULTS

This section includes examples of an implementation of some of the methods described in the previous section. Two tests were conducted: first, a test for motion estimation algorithm analysis and, second, a test for the overall codec effectiveness analysis.

A. Motion compensation algorithm analysis

The motion compensation (MC) algorithm analysis for the x264 encoder (video coding standard MPEG-4 AVC/H.264 [13]) was performed as the first test of the proposed analysis method. There are four MC algorithms in the x264 encoder: ESA (an exhaustive search), DIA (a simple diamond search), HEX (an adaptive pattern search) and UMH (a combination of different methods over a hexagonal motion search pattern). Frame decimation modifications were used to analyze all of the algorithms. The standard video sequences "Stefan" and "Flower Garden," each with resolution 352x288 (CIF), were

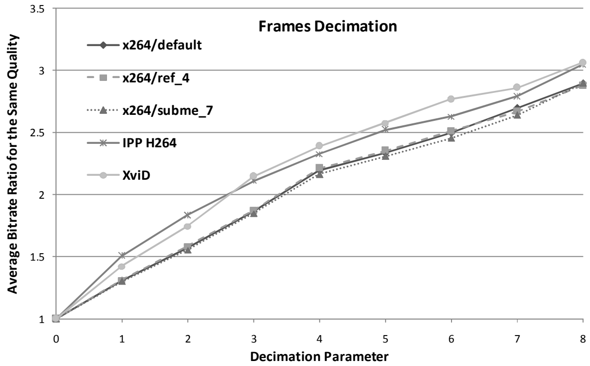

Fig.5 Video codec analysis results using frame decimation modification.

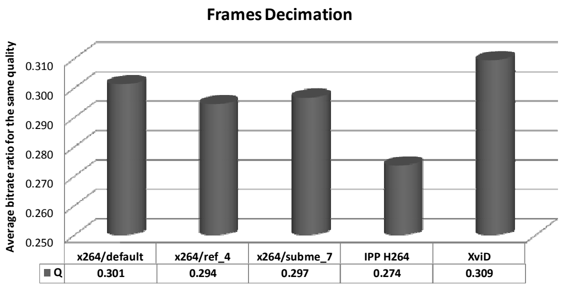

Fig. 6 Video codec scores, obtained using frame decimation modification (γ=*0.5*).

used as the original sequences. These sequences were decimated with parameters ranging from 1 to 8. The 18 resulting sequences (the original plus eight modifications for each of the two sequences) were encoded with the x264 encoder using bitrates from 100 Kbit/sec to 3 Mbit/sec (a total of seven bitrates). The RD curves calculated for each modification were compared with the RD curve for the corresponding original video sequence. Comparison results are depicted in Fig. 3.

It is obvious from Fig. 3 that the DIA and HEX algorithms have a better relative effectiveness trend for an increasing number of decimated frames. Based on this result it is possible to draw a conclusion about the increased quality of these algorithms relative to simpler implementations of DIA and HEX.

Note that the differences between MC algorithms are relatively small for other types of video sequence modifications. This situation allows us to use certain modifications to test only particular parts of a codec. Changes in other codec parts have a weak impact on the analysis results. An example of the same results for noise macroblock insertion modifications are depicted in Fig. 4.

The x264 encoder with default settings was selected as the reference codec (the HEX algorithm is used). The average bitrate ratio for the original sequence relative to the reference encoder, along with final analysis estimates, is listed in Table I (γ=*0.5* is used in (2)). Lower estimates correspond to better results.

The average bitrate ratio for a given quality is nearly equal for all of the MC algorithms. The final results of presets with complex algorithms are much better because of varying rates of codec effectiveness degradation with increasing motion complexity.

TABLE III
AVERAGE BITRATE RATIO FOR A GIVEN QUALITY FOR NATURAL SEQUENCES

Codec/Preset	Bitrate Ratio
x264/default	100.0%
x264/ref_4	97.1%
x264/subme_7	98.0%
IPP H.264	88.1%
Xvid	105.0%

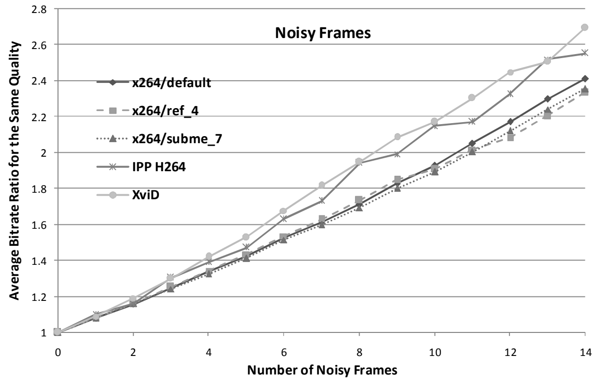

Fig. 7 Video codec analysis results using noisy frame insertion modification.

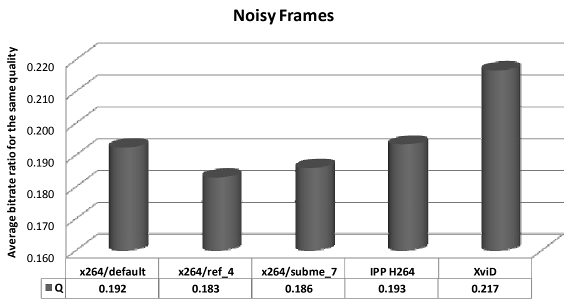

Fig. 8 Video codec scores, obtained using noisy frame insertion modification (γ=*0.5*).

B. Overall codec effectiveness analysis

This section describes the overall codec effectiveness analysis using the proposed analysis methods. The codecs under test and their respective parameters are listed in Table II. There are two MPEG-4 AVC/H.264 codecs (for a total of four presets) and one MPEG-4 ASP [14] codec. The standard video sequences “Stefan” and “Flower Garden” (in CIF resolution) were used for all of the tests.

The average bit-rate ratio for a fixed quality for the original sequence in comparison with the reference codec is presented in Table III. Lower values correspond to better results. The x264 codec with the default parameters was selected as the reference encoder. The maximum difference between the best and worst codecs was 17% of bit-rate. The best result is demonstrated by the IPP H.264 encoder, and the worst result by the Xvid encoder.

All of the following results were obtained with $\gamma=0.5$. Target bitrates were varied from 100 Kbit/sec to 3 Mbit/sec. The actual bitrate for some codecs differed significantly from target bitrate, but this did not adversely influence the stability of the analysis. Codec performance was not strictly calibrated for this test. The difference in encoding speed between the fastest codec (Xvid) and the slowest codec (IPP H.264) was as much as a factor of five. Nevertheless, all of the codecs fit in the same category of medium complexity single-pass offline encoders.

The video codecs analysis results for various video sequences are presented below.

1) Frame decimation

The frame decimation parameter was changed from $N=1$ to $N=8$ (only 1/9th of all the frames in the sequence were left in the last case). The actual effectiveness of the degradation depended on the number of decimated frames, as depicted in Fig. 5. The x264 encoder demonstrates better results compared to both XviD and IPP H.264. Therefore, it can be concluded that the x264 encoder yields better performance in cases of complex dynamic video sequences with rapid scene changes and active motion. The final analysis results, including the comparison with the reference codec, are depicted in Fig. 6.

2) Adding noisy frames

Sequences with 1 to 14 inserted frames with uniformly distributed noise were used as basic modifications. Noise frames were inserted uniformly in the following positions:

$$F_i = \left\lfloor (i+0.5)\frac{M}{N} \right\rfloor, i = 0,\dots,\mathrm{N}-1,$$

where M is the number of frames in the video sequence and N is number of frames with noise. Each pixel for each color-plane was set to a uniformly distributed random value.

The trend of video encoder effectiveness depended on the noise frame count depicted in Fig. 7. The best results are rendered by the x264 encoder with preset “ref_4” (four reference frames). The worst-performing codec is XviD, likely because of its frame-level rate-control algorithm, which is too stable to effectively handle fast scene changes. The final analysis results, which include comparison with the reference codec, are depicted in Fig. 8.

3) Adding noisy macroblocks

Blocks of pixels with uniform noise (all colors have equal probability) were inserted into the original sequence in this test. Exactly N noisy blocks, each with a size of 16x16 pixels, were added to each frame. Block positions were aligned to a 16-pixel grid:

$$(x, y) = (0,0) \bmod 16 .$$

The alignment is necessary to create stricter conditions for video codec rate control algorithms. As in the case of noisy frames, each macroblock pixel of each color-plane has a uniformly distributed random color value.

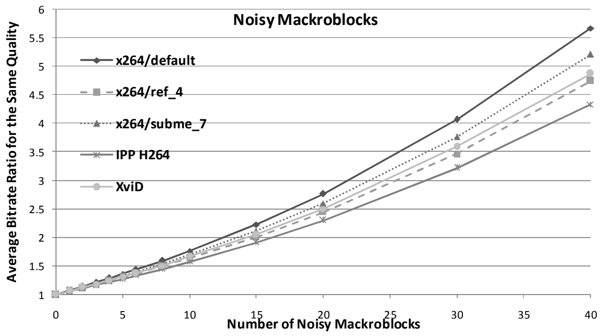

Fig. 9 Video codec analysis results using noisy macroblock insertion modification.

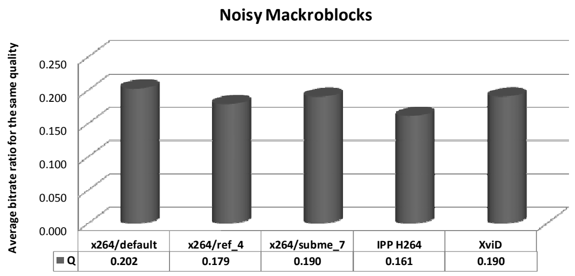

Fig. 10 Video codec scores, obtained using noisy macroblock insertion modification ($\gamma=0.5$).

TABLE IV
AVERAGE BITRATE RATION FOR A GIVEN QUALITY, SYNTHETIC SEQUENCE FOR MOTION ANALYSIS

Preset	Bitrate Ratio
x264/default	100.0%
x264/ref_4	94,9%
x264/subme_7	91,4%
IPP H.264	72,0%
Xvid	109,7%

The encoding effectiveness depended on the number of noisy blocks is depicted in Fig. 9. The best trend is demonstrated by the IPP H.264 encoder. There is no quantizer selection algorithm implemented in the x264 encoder, but the x264 results change depending on the particular settings used. The best results are demonstrated by the preset “ref_4.” The analysis estimates, including original video encoding effectiveness, are shown in Fig. 10.

4) Synthetic motion sequence

Nine different synthetic sequences with 300 frames (30 fps) and a resolution of 352x288 pixels were created for this test. The difference between these sequences was the *MAX_SPEED* parameter, which defines the complexity of object motion. This parameter was varied from 1 to 24. The trends in effectiveness degradation depending on speed complexity are depicted in Fig. 11. The best trend is demonstrated by the IPP H.264 encoder, and the worst trend by the default preset of x264. The relative bitrate ratio for the synthetic sequence with *MAX_MOTION=1* is shown in Table IV. Note that IPP H.264

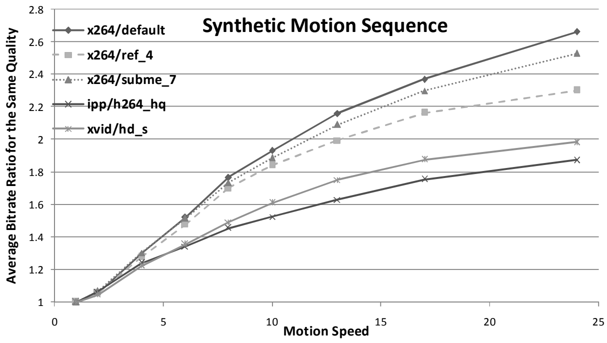

Fig.11 Video codec analysis results using synthetic motion sequences.

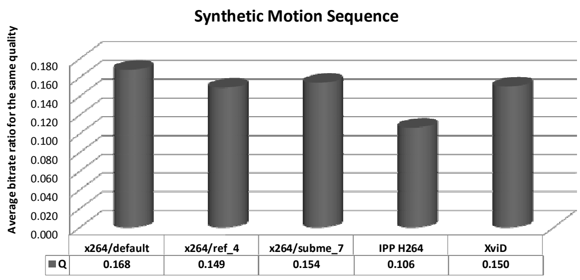

Fig. 12 Video codec analysis results using synthetic motion sequences.

shows an advantage of 28% in bitrate for the same quality as compared to x264 with default parameters. As a result, IPP H.264 shows the best estimate for this type of analysis. All the estimates are shown in Fig. 12.

IV. Final Score Calculation

A number of analyzers were described in the previous section. Each analyzer is intended to analyze a particular part of a video codec and procures one estimate for each codec. This section describes a method for combining all of the estimates into a final codec score. There are two stages of the proposed estimates combination process: 1) normalization and 2) weighted averaging.

A. Estimate normalization

Each analyzer has its own range of estimates, which depends on the range of slopes $\alpha(M,S)$ and analyzer parameter γ. Moreover, the sequences (natural or synthetic) used for the analysis also influence the analyzer estimate via the value $S_{ref,tested}^{[a,b]}$. As a result, normalization is required prior to combination of the estimates.

The proposed normalization process is based on the entire range of results for the codecs under test. The best estimate S_{best} and the worst estimate S_{worst} from among all the analyzer estimates are determined. Next, all of the estimates are linearly rearranged within the interval *[0, 100]* using the following formula:

$$\bar{S} = 100\frac{S_{worst} - S}{S_{worst} - S_{best}},$$

TABLE V
Analyzer Weight

Analyzer	Weight
Frames Decimation	1
Noise Frames	2
Noise Blocks	1.5
Synthetic Motion	1

TABLE VI
Normalized Scores for each Analyzer and Final Codec Score

Codec/Preset	Frames Decim.	Noise Frames	Noise Blocks	Synth. Motion	Final Score
x264/default	23	72	0	0	**167**
x264/ref_4	42	**100**	57	31	**358**
x264/subme_7	36	91	30	23	**285**
IPP H.264	**100**	69	**100**	**100**	**488**
Xvid	0	0	29	29	**73**

Larger values correspond to better results. The best result for each analyzer and the final scores are in boldface.

where $\bar{S}$ is the normalized estimate and S is the original estimate.

This normalization process indicates that the normalized codec estimate depends not only on the codec's results, but also on the results of the other tested codecs. An alternative normalization process can use a static normalization equation; for example, the process could fix setting S_{best} and S_{worst} for each analyzer. Unfortunately, selecting normalization formula parameters requires knowledge about typical estimates for commonly used codecs; in fact, this process is rather subjective. Moreover, static normalization is suitable only for some "typical" set of codecs, and it can produce strange values for other codecs (for example, codecs with perfect or relatively higher efficiency).

B. Estimate weighted averaging

The result of the normalization process is a set of estimates in the range *[0, 100]* for each analyzer. The last step of the proposed scoring method is combination of these estimates to produce a final score for each analyzed codec. A weighted average is used to this end:

$$S_{final} = \sum_{i=1}^{N} w_i \bar{S}_i ,$$

where S_{final} is the final score, $\bar{S}_i$ is the normalized estimate of the i^{th} analyzer, w_i is the weight of i^{th} analyzer and N is the total number of analyzers.

Weights w_i correspond to the significance of the analyzers. The weights can be selected based on the importance of different parts of the codec for a target application area or based on some expert's subjective score for the codecs under test. The weights used in this paper are listed in Table V. The normalized estimates for all of the proposed analyzers, along with the final scores, are presented in Table VI. The best final score (488) was achieved by the IPP H.264 codec, and the worst score (72) by Xvid. This result meets expectations, since the performance of the IPP H.264 encoder is the slowest and that of the Xvid MPEG4 is the fastest within this analysis.

V. Conclusion

This paper has presented a novel method of analysis that is suitable both for overall video codec quality and for separate aspects of a video codec, such as rate control, motion estimation and mode decision algorithms. This method can be applied to a variety of codecs and video coding standards because it does not use any specific knowledge of the encoded stream structure or any details of the encoding algorithm implementation. The proposed method considers a video codec as an abstract, lossy compression system with a rate-distortion function that can be approximated. The use of natural sequences increases the adequacy of the analysis for the target video codec implementation area. Selection of a more specific test set can increase the analysis quality in specific usage areas, such as videoconferences and dynamic actions. Generation of sequence modifications is the basis of the proposed methods. Each modification is oriented toward a specific part of the rate control algorithms. Specific video codec algorithms can be analyzed, with only common assumptions about the encoding process being required. Access to information about video coding standards or additional information from encoded streams can improve the quality of the analysis using additional dependencies between encoded syntax elements. Other future opportunities for the proposed methodology exist in the area of combining automatic scoring with subjective human assessment to produce more accurate final scoring.

References

[1] Mohammed Ghanbari. *Standard Codecs: Image Compression to Advanced Video Coding. Institution Electrical Engineers*, 2003, ISBN 0-85296-710-1.

[2] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini and G. J. Sullivan. „Rate-constrained coder control and comparison of video coding standards". *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, Jul. 2003.

[3] Moscow State University, Laboratory of Computer Graphics and Multimedia, *Fourth annual MPEG-4 AVC/H.264 videocodec comparison. 2007*. Available: http://compression.ru/video/codec_comparison/mpeg-4_avc_h264_2007_en.html.

[4] *Methodology for the subjective assessment of the quality of* television *pictures.* ITU-T BT 500.11, 2002.

[5] M. J. Nadenau, S. Winkler, D. Alleysson and M. Kunt. "Human Vision Models for Perceptually Optimized Image Processing – A Review," *Submitted to Proceedings of IEEE*, September 2000.

[6] T. M. Cover and J. A. Thomas. *Elements of information theory. 2nd Edition.* New York: Wiley-Interscience, 2006. ISBN 0-471-24195-4.

[7] B. Tao, B. W. Dickinson and H. A. Peterson, "Adaptive Model-Driven Bit Allocation for MPEG Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, February 2000.

[8] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, VOL. 13, NO. 4, April 2004.

[9] D. Vatolin and A. Parshin. *MPEG-4 AVC/H.264 videocodec comparison using objective quality metrics. GraphiCon'2006, pp. 447-454*.

[10] Videocodec x264 of videocoding standard H.264 Available: http://www.videolan.org/developers/x264.html

[11] Library Intel Integrated Performance Primitives. Available: http://www.intel.com/cd/software/products/asmo-na/eng/perflib/ipp/302910.htm

[12] Videocodec Xvid of videocoding standard MPEG-4 ASP Available: http://www.xvid.org/

[13] *Draft ITU-T recommendation and final draft international standard of joint video specification* (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC). Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[14] *Coding of Moving Pictures and Associated Audio. ISO/IEC 14496-2*, MPEG-4 Committee Draft ISO/IEC JTC1/SC29/WG11, Oct. 1998

[15] ITU-T Recommendation J.144, "Objective perceptual video quality measurement techniques for digital cable television in the presence of a full reference."

[16] ITU-R Recommendation BT.1683, "Objective perceptual video quality measurement techniques for standard definition digital broadcast television in the presence of a full reference."

[17] Philip Corriveau "Chapter 4 Video Quality Testing, Chapter 11 Video Quality Experts Group, Appendix A VQM Performance Metrics." Digital Video Image Quality and Perceptual Coding Text Book, Edited by H.R. Wu and K.R. Rao – Taylor and Francis publishing 2006

[18] A.M. Rohaly P. Corriveau, A. Webster, J. Libert & al., "Video Quality Experts Group: Current Results and Future Directions", Visual Communications and Image Processing 2000 (Proceedings of SPIE Volume 4067, Perth Australia 20-23 June 2000

Alexander Parshin received his M.S. degree from Moscow State University, Moscow, Russia in 2005.Currently he is an Engineer in the Laboratory of Graphics and Multimedia at the university. His research interests include video compression, video codecs analysis, video quality analysis and video reconstruction.

Dmitriy S. Vatolin (M'06) received his M.S. degree in 1996 and his Ph.D. in 2000, both from Moscow State University. Currently he is Head of the Video Group at the CS MSU Graphics & Media Lab. He is author of the book *Methods of Image Compression* (Moscow, VMK MGU, 1999), co-author of the book *Methods of Data Compression* (Moscow, Dialog-MIFI, 2002) and co-founder of www.compression.ru and www.compression-links.info. His research interests include compression techniques and video analysis and filtering methods.

Ilya Brailovskiy received his M.S. degree in 1995 from Moscow State University and his Ph.D. in 2003 from the Russia Academy of Science. He is a Media Architect for Intel Corporation. His interests include a wide range of visual computing technologies.

Philip Corriveau received his B.Sc, degree in 1990 from Carleton University, Ottawa, Canada. He is currently a Principal Engineer with Intel Corporation and manages a group of human factors engineers conducting user experience research and assessment. He is a founding member of the Video Quality Experts Group and an active member in the IEEE, ITU and ANSI.