Rate-control algorithms testing by using video source model

Eugeniy Belyaev, Andrey Turlikov and Anna Ukhanova
Saint-Petersburg State University of Aerospace Instrumentation, Russia
Email: {ebelyaev, turlikov, anja}@vu.spb.ru

***Abstract*— In this paper the method of rate control algorithms testing by the use of video source model is suggested. The proposed method allows to significantly improve algorithms testing over the big test set.**

***Index Terms* – rate-control, rate-distortion model, video source model, JPEG2000.**

I. Introduction

Traditional way of rate control algorithms testing is confined by the getting experimental data over the testing set of the moderate size. This concerned with the laboriousness of testing over the large testing set. This paper proposes the method of rate control algorithms testing based on the use of video source model. As a basis for model it is suggested to use the hidden Markov chain. The states of this Markov chain are presented by the type of the video source. In this particular case there are three main types: images with computer graphic, photographic images and mixed type. Each state of Markov chain corresponds to a characteristical distribution of rate-distortion function parameters for each source type.

This paper focuses on the method of source model construction. It also describes the practical efficiency for the rate control algorithm in the memory restricted video compression and transmission systems, proposed in [1], [2], based on the JPEG2000 algorithm.

This paper is organized as follows. Firstly the short description of the rate control algorithm in JPEG2000 [3] standard is given. Then the model of the video source and the way of its construction are presented. Finally, the results of the comparison of rate control algorithm applied to the real video data and data obtained with this model.

II. Real-time video compression and transmission system

Consider the system timing is discrete and slotted. The slot time is a unity of the system time $[t, t+1)$, and time moment t refers to the end of this slot. The video source gives coder a tile at the certain time slots. Coder works in the real-time. After compressing tile into r_t bits, coder places it into the *transmitter buffer* at the $[t, t+1)$ slot end (see fig. 1). Depending on the number of bits in the transmission buffer, rate-distortion controller forms the requirements for the next tile. The number of bits in the buffer $b(t)$ after placing there a new compressed tile and transmitting over the channel with the constant throughput rate c, changes as follows:

$$b(t) = \max\{0, b(t-1) - c\} + r_t. \quad (1)$$

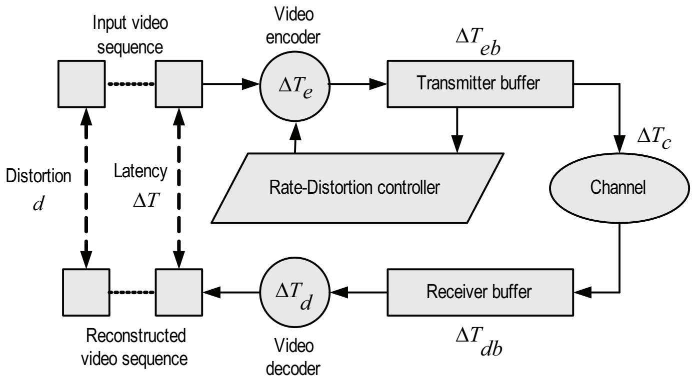

Fig. 1. Real-time video compression and transmission system

Data on the receiver side are accumulated for some time L after which the decoding and playing starts.

The total latency $\triangle T$ between the time when a new frame goes to the encoder and the moment when this frame is displayed at the receiver side consists of the following components:

$$\triangle T = \triangle T_e + \triangle T_{be} + \triangle T_c + \triangle T_{bd} + \triangle T_d, \quad (2)$$

where $\triangle T_e$ and $\triangle T_d$ - time for encoding and decoding correspondingly, $\triangle T_{be}$ - waiting time in the transmitter buffer, $\triangle T_{de}$ - delay time in the receiver buffer, $\triangle T_c$ - transmitting time over the channel.

If in the above described scheme the size of the buffer at the transmitter and receiver side is equal $B^e_{max} = B^d_{max} = L \cdot c$ and rate control works in the way that the number of bits in the transmitter buffer is $b^e(t) \leq B^e_{max}$ (transmitter buffer is not overflowed), then the delay caused by buffering at the transmitter side is constants and equals the initial buffering delay L [5].

III. Bitrate control mechanism in JPEG2000

The general scheme of the algorithm JPEG2000 is described below [4]. At first, color space transformation is applied to the tile-component data. Then a discrete wavelet transformation is used to decompose each image tile into a hierarchy of subbands. Two wavelet filters are used for this purpose: low-frequency $h_0(n)$ and high-frequency $h_1(n)$. Firstly, the decomposition is applied to the rows of image, then to the columns. As a result, 4 matrixes HH_0, HL_0, LH_0, LL_0 are obtained, each one accords to the corresponding filtration procedure: filter $h_1(n)$ applied for rows and columns, filter $h_1(n)$ for rows and filter $h_0(n)$ for columns, filter $h_0(n)$ for rows and filter $h_1(n)$ for columns, filter $h_0(n)$ for rows and columns.

Then the decimation of all matrixes with coefficient 2 is made. After it the subsequent wavelet transform is applied to the matrix LL_0. As a result of this transformation four matrixes are obtained : HH_1, HL_1, LH_1, LL_1.The decomposition is made v times in the way described above. As a result of this transformation the set of $3v+1$ matrixes is received.

The following encoding consists of three main steps.

During the first step (Tier-1) wavelet coefficients are lossless compressed. Each matrix of a tile is divided into blocks that are encoded independently. Binary coefficients of the blocks, related to the same order form bit plane. The encoding is made plane by plane starting from the high-order bit. Each bit plane (except most significant non-zero bit plane) is encoded in three passes in the way that the encoding may be truncated after any number of passes.

On the second step (RD-Optimization) for each block the passes to be sent to the decoder should be chosen. The blocks in the tile of the image are numbered as $i = 1, 2, ...$ The block consists of an integer number of passes. Each additional pass from block i contributes some bytes to the block i and, in turn, decreases the overall image distortion. Let r_i^k be the number of bits and d_i^k be the mean squared error of a block while transmitting passes with the numbers $1, 2, ..., k$, $k = 1...p$. Decoder receives the passes with the numbers $1, 2, ..., n_i$. The number of the last pass n_i, that is sent to the decoder is called *truncation point*. Consider $\mathbf{n} = \{n_i\}$ to be *truncation vector*, where n_i signifies that $i-th$ block is truncated in the point with the number n_i. Denote the overall distortion

$$d(\mathbf{n}) = \sum_i d_i^{n_i},$$

where $d_i^{n_i}$ - corresponds to the distortion value of the block i truncated in the point number n_i. Denote the resulting tile rate

$$r(\mathbf{n}) = \sum_i r_i^{n_i},$$

where $r_i^{n_i}$ - number of bits for i block truncated in the point number n_i.

The optimization task in JPEG2000 standard consist in the search of the truncation vector $\mathbf{n}_{opt}$, so that

$$\begin{cases} d(\mathbf{n}_{opt}) = \min\limits_{\{\mathbf{n}\}} d(\mathbf{n}) \\ r(\mathbf{n}_{opt}) \leq r_{in}. \end{cases} \tag{3}$$

IV. Memory constrained MINMAX algorithm of rate and visual quality control

In papers [1], [2] it is proposed to control not only the number of bits of the compressed tile, but also the visual quality of each tile. In other words, it is suggested to compress tiles with the equal acceptable level of visual quality taking into account the channel throughput constraints. To provide

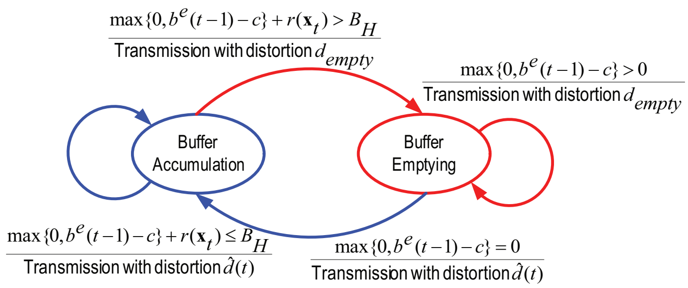

Fig. 2. Memory constrained algorithm of rate and visual quality control

the distortion not more than d it is needed, similarly to (3), to find the truncation vector $\mathbf{x}_t \in \mathbf{N}_t$, so that

$$\begin{cases} r(\mathbf{x}_t) = \min\limits_{\mathbf{n} \in \mathbf{N}_t} r(\mathbf{n}) \\ d(\mathbf{x}_t) \leq d. \end{cases}$$

It is obvious that bit sizes of each tile can vary a lot and exceed the channel throughput c. Therefore, the size of the transmitter buffer $B_{max}^e > c$. Consider that statistical properties of all frames in video sequence vary insignificantly. Formulating the optimization task more exact, it is needed to select the truncation vector $\mathbf{x}_t$ for each tile t, so that

$$\begin{cases} minimize \max\limits_t d(\mathbf{x}_t) \\ b(t) \leq B_{max}^e. \end{cases} \tag{4}$$

A. Consecutive search algorithm

If there aren't any memory or computational resources constraints, this task can be solved as follows. Consider $\mathbf{D}$ to be the set of the possible distortion values for all tiles in ascending order, complemented with 0 and ∞. Assume d_i to be the element of this set. The algorithm is staged as follows. At the i-th stage the threshold $\tilde{d} = d_i$ is chosen. For each tile t it is needed to choose truncation vector $\mathbf{x}_t \in \mathbf{N}_t$, so that

$$\mathbf{x}_t = \arg\max_{\mathbf{n} \in \mathbf{N}_t} \{d(\mathbf{n}) : d(\mathbf{n}) \leq \tilde{d}\},$$

and compute the number of bits in the buffer $\tilde{b}(t)$ according to (1). If the buffer of size B_{con} is overflowed at the i-th stage, the same operations will be applied at the stage $i+1$, otherwise the solution is found and the threshold $\tilde{d}$ is the required solution for the optimization task. The algorithm described above is called the *consecutive search algorithm*.

Lemma 1: There is no sequence of truncation vectors, that does not lead to the buffer overflow and has the maximum distortion value less than $\tilde{d}$ [2], where $\tilde{d}$ is the maximum distortion value, found by the consecutive search algorithm.

It is evident from this lemma, that the consecutive search algorithm can be used for solving the optimization task (4).

B. Memory constrained algorithm of rate and visual quality control

The method described above implies no memory constraints. In [1], [2] was proposed an algorithm that allows to find the estimation of $\tilde{d}$ for the memory constrained consecutive search algorithm. Consider $\hat{d}(t)$ to be the estimation of $\tilde{d}$ value and $\hat{d}(0) < \tilde{d}$. It is supposed to estimate $\tilde{d}$ value as follows (see fig. 2). All tiles are compressed with the distortion not more than $\hat{d}(t)$ until the number of bits in the buffer will not exceed some threshold B_H. Threshold B_H crossing means that it is impossible to hold the $\hat{d}(t)$ level for the distortion value for the given channel throughput. Consequently, the buffer is firstly emptied and then the estimation of distortion $\hat{d}(t)$ is changed, that peak signal-to-noise ratio is decreased to $\Delta\mathtt{PSNR}$ value. Peak signal-to-noise ratio value depend on distortion by following:

$$\mathtt{PSNR}(d) = 10 \cdot \log(\frac{255^2 \cdot 3 \cdot t_w \cdot t_h}{d}), \tag{5}$$

where t_w and t_h are tile width and height. From (5) it is follow, that

$$d(\mathtt{PSNR}) = 255^2 \cdot 3 \cdot t_w \cdot t_h \cdot 10^{\frac{-\mathtt{PSNR}}{10}},$$

Algorithm consists of the following three steps.

Step 0. (*Initialization*)

0.1 Set initial value $\hat{d}(0) = d(\mathtt{PSNR}(0))$, $t = 0$, $b(0) = 0$.

0.2 Go to step 1.

Step 1. (*Buffer accumulation*)

1.1 $t = t + 1$, $\min\{b(t-1), c\}$ bits are transmitted.

1.2 t-th tile is lossless compressed, the set of truncation vectors $\mathbf{N}_t$ is found.

1.3 Search for $\mathbf{x}_t \in \mathbf{N}_t$, so that

$$\begin{cases} r(\mathbf{x}_t) = \min\limits_{\mathbf{n}\in\mathbf{N}_t} r(\mathbf{n}) \\ d(\mathbf{x}_t) \leq \hat{d}(t). \end{cases}$$

1.4 If $\max\{0, b(t-1) - c\} + r(\mathbf{x}_t) > B_H$ then go to step 2.3.

1.5 Append compressed tile to the transmitter buffer according to the vector $\mathbf{x}_t$ and go to step 1.1.

Step 2. (*Buffer emptying*)

2.1 $t = t + 1$, $\min\{b(t-1), c\}$ bits are transmitted.

2.2 t-th tile is lossless compressed, the set of truncation vectors $\mathbf{N}_t$ is found.

2.3 Compute the number of bits that are needed to transmit the tile with distortion d_{empty} by searching for the vector for $\mathbf{x}_t \in \mathbf{N}_t$, so that

$$\begin{cases} r(\mathbf{x}_t) = \min\limits_{\mathbf{n}\in\mathbf{N}_t} r(\mathbf{n}) \\ d(\mathbf{x}_t) \leq d_{empty}. \end{cases}$$

2.4 Search for $\mathbf{y}_t \in \mathbf{N}_t$, so that

$$\begin{cases} d(\mathbf{y}_t) = \min\limits_{\mathbf{n}\in\mathbf{N}_t} d(\mathbf{n}) \\ r(\mathbf{y}_t) \leq \min\{B^e_{max} - b(t), r(\mathbf{x}_t)\}. \end{cases}$$

2.5 If $\max\{0, b(t-1) - c\} = 0$ then $\mathtt{PSNR}(t) = \gamma\mathtt{PSNR}(t)$ $\hat{d}(t+1) = d(\mathtt{PSNR}(t))$ and go to step 1.3.

2.6 Append compressed tile to the transmitter buffer according to the vector $\mathbf{y}_t$ and go to step 2.1.

Lemma 2: Consider that consecutive search algorithm with transmitter buffer of size B_{con} finds the maximum distortion value $\tilde{d}$. Then for the proposed algorithm [2] with threshold $B_H = B_{con}$ and initial value of distortion threshold estimation $\hat{d}(0) < \tilde{d}$, the following inequality $\mathtt{PSNR}(\hat{d}(t)) \geq \mathtt{PSNR}(\tilde{d}) - \Delta\mathtt{PSNR}$ holds true for any time moment t.

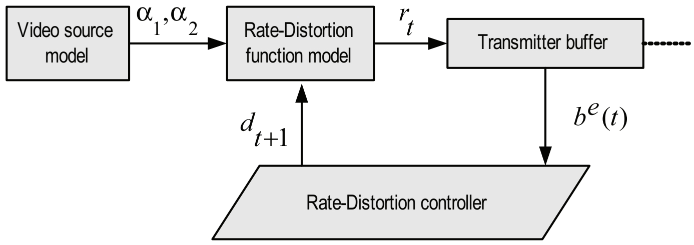

Fig. 3. Rate-control algorithms testing by using video source model

V. Rate-control algorithms testing by using video source model

This paper proposes the method of rate control algorithms testing by using the video source model (see Fig.3). The proposed method is significantly faster that the traditional one, as instead of tile encoding the rate distortion function model is used. Besides the proposed model allows to significantly expand the variety of the test sets, that is much more difficult for traditional method.

A. Image tile types classification

For classification of image tile types it is proposed to use lossless tile compression with different number of wavelet decomposition levels [6]. According to the number of decomposition values, each tile is classified as one of the following types: 1 - tile with computer graphic, 2 - tile of mixed type, 3 - natural (photographic) tile.

B. Rate-distortion function model in JPEG2000 standard

Each state of the Markov chain corresponds to the rate-distortion function parameters distribution. The most frequently used rate-distortion function [7] is presented below:

$$R(D) = \ln(\frac{1}{\alpha D}). \tag{6}$$

From (6) it is easy to come to the next function [8]:

$$\mathtt{PSNR}(D) = \alpha_1 \cdot \log R + \alpha_2. \tag{7}$$

To define parameters α_1 and α_2 for each image tile it is possible to derive the rate-distortion function. To deduce this function experimentally it is proposed similarly to (3) solve the next optimization task. It is needed to find the truncation vector $\mathbf{n}_{opt}$, so that

$$\begin{cases} r(\mathbf{n}_{opt}) = \min\limits_{\{\mathbf{n}\}} r(\mathbf{n}) \\ d(\mathbf{n}_{opt}) \leq d_{in}. \end{cases} \tag{8}$$

Each point on the rate-distortion curve $\{r(n_{opt}), d(n_{opt})\}$ is determined by the input value d_{in}. Value range $\{d_{in}\}$ is chosen in the way that $\mathtt{PSNR}$ value changes from 30 to 50 dB.

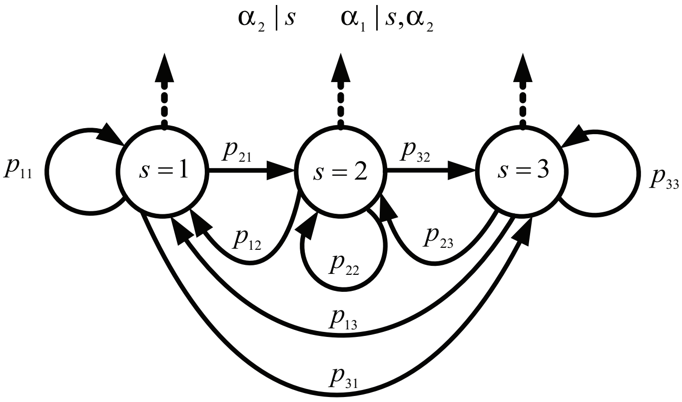

Fig. 4. Video source model based on hidden markov chain

After construction of rate-distortion function, technique of least squares is used to define the model parameters (7) α_1 and α_2.

C. Video source model based on hidden Markov chain

This paper presents video source model, based on the hidden Markov chain (see fig. 4). As a state of this chain tile type s is chosen: $s = 1$ - computer graphic, $s = 2$ - mixed type, $s = 3$ - natural(photographic) type. At the same time each state of the Markov chain corresponds to the conditional distribution of the rate-distortion function parameters $\alpha_2|s$ and $\alpha_1|s, \alpha_2$.

To determine model parameters, by observing the parameters of the real video source it is necessary to define the matrix of transition probabilities for the Markov process s and conditional distribution of the rate-distortion function $\alpha_2|s$ und $\alpha_1|s, \alpha_2$.

It should be also noted that Markov chain is a hidden Markov chain only for the rate-control algorithm. For model construction the algorithm of video sources type classification (see above), and this process s is observed. Therefore, it is not needed to use any specifical mathematical apparat [9].

Values α_1 and α_2 are real, so it is needed to quantize these values at first. The number of the quantizing levels should be big enough. From the other hand, as the number of the experiments is limited, each quantizing interval should have enough values in it. On the grounds of it for conditional distribution construction $\alpha_2|s$ and $\alpha_1|s, \alpha_2$ the non-uniform quantizing should be made, like in [10], [11].

VI. RESULTS

For practical results was used the reference JPEG2000 codec implementation Jasper, v. 1.701.0 with tile width $t_w = 256$ and tile height $t_h = 16$. The source model was built to get the experimental results over the video test set. On its basis α_1 and α_2 were generated. As an input for the rate control described above firstly the real test set was used, and then the parameters of rate-distortion function generated by the video source model were used also. Figures 5,6 show the dependencies of the values $\texttt{PSNR}(\hat{d}(t))$ and the number of bits int the buffer $b^e(t)$ for real and modulated data. Figure 7

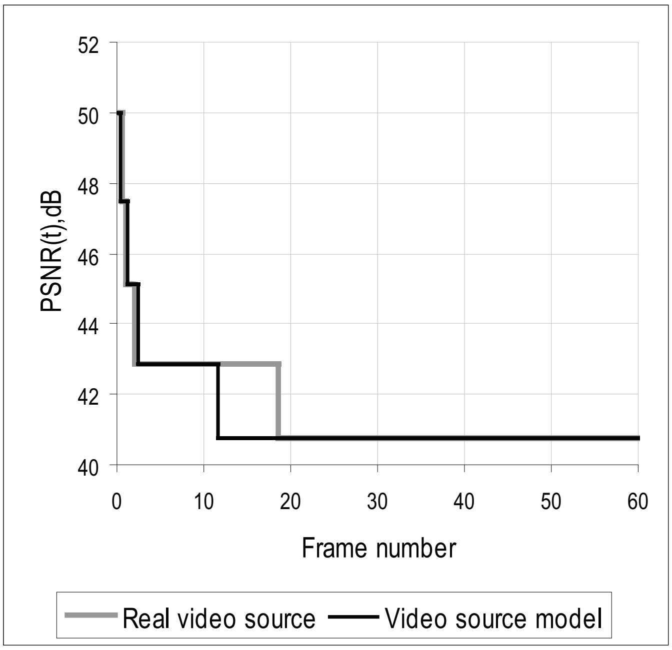

Fig. 5. Experimental comparison of rate-control over real and modulated data, $\texttt{PSNR}(0) = 50$dB, $\gamma = 0.95$, $B^e_{max} = 35$kB, rate=0.1.

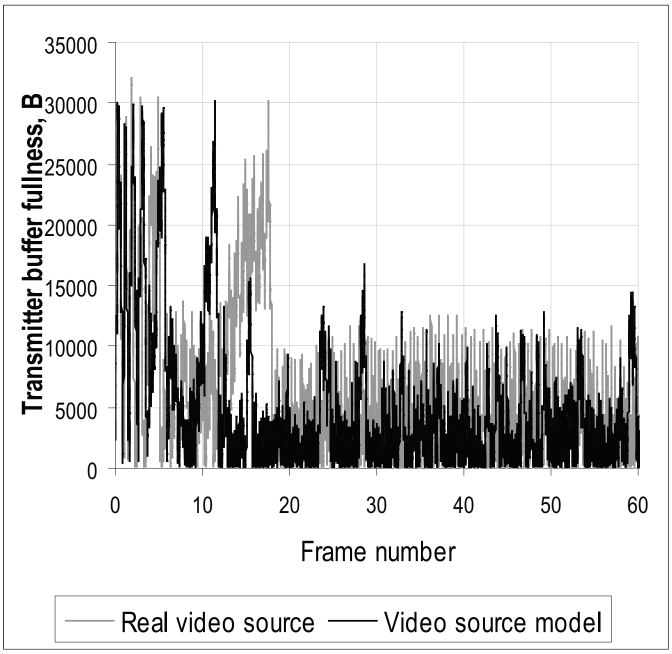

Fig. 6. Experimental comparison of rate-control over real and modulated data, $\texttt{PSNR}(0) = 50$dB, $\gamma = 0.95$, $B^e_{max} = 35$kB, rate=0.1.

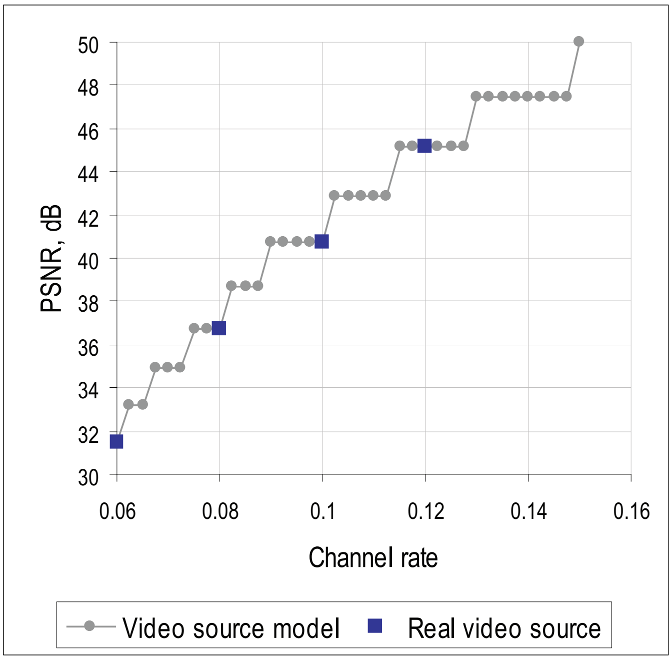

Fig. 7. Experimental comparison of rate-control over real and modulated data, $\texttt{PSNR}(0) = 50$dB, $\gamma = 0.95$, $B^e_{max} = 35$kB.

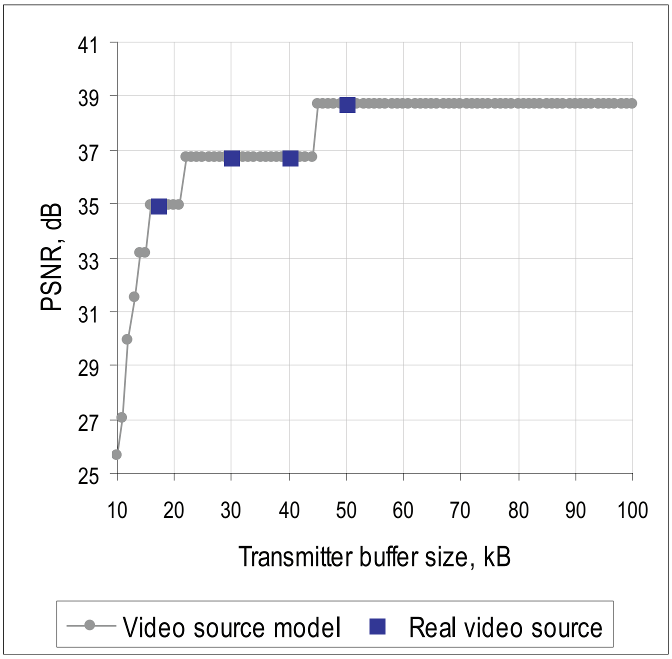

Fig. 8. Experimental comparison of rate-control over real and modulated data, $\texttt{PSNR}(0) = 50$dB, $\gamma = 0.95$, rate=0.08.

shows $\texttt{PSNR}$ dependence for frame after algorithm adaptation to the channel rate and video source parameters from channel rate for real and modulated data with the fixed transmitter buffer size. Figure 8 shows $\texttt{PSNR}$ dependence for frame after rate control algorithm adaptation to the channel rate and video source parameters from transmitter buffer size for real and modulated data for the fixed channel rate.

Experimental results show that the rate-control algorithm works similarly on the real and modulated data, so the proposed model could be used for testing and defining parameters of the rate control algorithm.

The main advantage of this model is that the rate-control testing could be done over the big video sequence data that is difficult to make with the real video data. This approach could also be used for other video compression algorithms.

References

[1] E. Belyaev, A. Turlikov and A. Ukhanova, “Rate-distortion control in wavelet-based video compression systems with memory restriction“, *XI International Symposium on Problems of Redundancy in Information and Control Systems*, St.-Petersburg, Russia, pp. 13–17, 2007.

[2] E. Belyaev and A. Turlikov, “Rate-distortion control in video compression and transmission system with memory restriction on transmitter and receiver side“, *Computer optics journal*, vol.31, NO. 2, pp. 69–76, 2007 (in Russian).

[3] ITU-T and ISO/IEC JTC 1, “JPEG 2000 Image Coding System: Core Coding System, ITU-T Recommendation T.800 and ISO/IEC 15444-1“, *JPEG 2000 Part 1*, 2000.

[4] D. Taubman and M. W. Marcellin, “JPEG2000: Image Compression Fundamentals, Practice and Standards“, *Massachusetts: Kluwer Academic Publishers*, 2002.

[5] A.R. Reibman, B.G Haskell, “Constraints on variable bit-rate video for ATM networks“, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, Issue 4, pp. 361 – 372, 1992.

[6] R. R. Coifman and M. V. Wickerhauser, “Entropy-based algorithms for best basis selection“, *IEEE Transactions on Information Theory*, vol. 38, pp. 713–718, 1992.

[7] A. Viteribi and J. Omura, “A new rate control scheme using a new rate-distortion model“, *Principle of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

[8] M. Dai, D. Loguinov, H. Radha, “Rate-distortion modeling of scalable video coders“, *International Conference on Image Processing*, vol. 2, pp. 1093–1096, 2004

[9] L.R. Rabiner, “A tutorial on hidden Markov models and selected applications inspeech recognition“, *Proceedings of the IEEE* vol. 77, Issue 2, pp. 257–286, 1989.

[10] J. Max, “Quantizing for minimum distortion“, *IEEE Transactions on Information Theory*, vol.6, pp. 16–21, 1960.

[11] S.P. Lloyd, “Least square quantization in PCM“, *IEEE Transactions on Information Theory*, vol.28, pp.129–137, 1982.