MINMAX Rate control in near-lossless video encoders for real-time data transmission

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

Abstract

This paper is about real-time data transmission over high throughput channels based on near-lossless video encoders. The algorithm of the video source rate control is proposed to guarantee low-latency data transmission according to MINMAX absolute difference criteria. The practical results for video encoder based on JPEG-LS standard are also given as well as the comparison with other algorithms.

I. Introduction

In the last few years there appeared lots of a high throughput video broadcasting systems (such as WirelessHD [1], IEEE 802.15.3c [2] etc.), which require latency of the data transmission significantly lower than the time of one frame playback. For such a system it is enough to use only the small compression ratio of the video data. Thus, near-lossless video compression algorithms like JPEG-LS [3] could be effectively used for such systems. So that the development of the source rate control for these encoders is very actual and important question. There are many papers devoted to the rate control for near-lossless encoders (see [4], [5]). But these approaches based on multi-pass encoding are not suitable for real time compression. Moreover, they do not provide video transmission taking into account any visual quality criteria.

This paper describes a video source rate control algorithm, that is based on MINMAX quality criteria and provide low-latency video transmission over constant throughput channel. For the first time the idea of this algorithm was proposed in [6], [7] for JPEG2000 standard. Now we show that ideas presented in these papers can be easily interpreted for near-lossless encoders that use maximum absolute difference criteria instead of mean squared error.

This paper is organized as follows. In section II latency in video compression and transmission systems is discussed. In section III MINMAX optimization task for the rate control algorithm is introduced. Section IV presents rate control algorithm and proves that it finds the solution of the optimization task. In section V and VI the conception of virtual buffer is introduced for scene changing detection and decreasing algorithm adaptation time. Finally, the practical results for different test video sequences are shown and some conclusions are made.

II. Latency in video compression and transmission systems

To transmit the video sequence with latency less than the time of one frame playback each frame is divided into the sequence of slices. 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 slice at the certain time slots. Coder works in the real-time. After compressing the slice into $r(d_t)$ bits, coder places it into the *transmitter buffer* at the $[t, t+1)$ slot end. Depending on the number of bits in the transmission buffer, rate controller forms the maximum absolute difference value d_{t+1} for the next slice

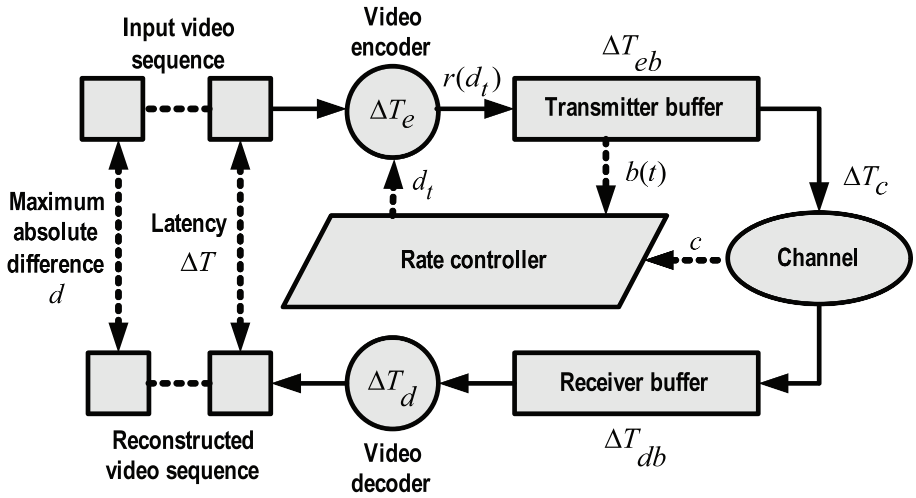

Fig. 1. Real-time video compression and transmission system

(see fig. 1). The number of bits in the buffer $b(t)$ after placing there a new compressed slice and transmitting over the channel with the constant rate c, changes as follows:

$$b(t) = \max\{0, b(t-1) - c\} + r(d_t). \tag{1}$$

The transmitted data is accumulated for time L (initial buffering latency) on the receiver side after that the decoding and playing starts.

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

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

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 sides 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

$$b^e(t) \leq B^e_{max}, \tag{3}$$

then the latency caused by buffering at the transmitter side is constant and equals the initial buffering latency L [8].

III. MINMAX OPTIMIZATION TASK FOR RATE CONTROL

In this section we propose the optimization task for the rate control, which is based on MINMAX quality criteria. This criteria is used in many papers, dedicated to the video compression (e.g. see [9]). In this paper MINMAX criteria is interpreted as follows. For each slice t the maximum absolute difference value d_t should be chosen, so that:

$$minimize \max_t d_t. \tag{4}$$

Suppose that despite statistical properties of the slices in the frame may be quite different from each other, statistical properties of all frames vary insignificantly. It means that there is only one scene in the input video sequence. This assumption does not hold true generally, because video sequence usually consists of subsequences (scenes) with different statistical properties. However, so that the understanding of the algorithm working with several scenes will be easier, let us initially take into account the case when video sequence has only one scene.

Let us formulate rate control optimization task according to the latency requirements (3) and the MINMAX quality criteria (4). For each slice t it is necessary to choose the maximum absolute difference value d_t, so that

$$\begin{cases} minimize \max_t d_t \\ b(t) \le B^e_{max}. \end{cases} \tag{5}$$

Solution of the task (5) can be found by the following hypothetic algorithm which consists of the following two steps:

Step 0. (*Initialization*)
0.1 Set $\{d_i\} = \{0, 1, ..., d_{max}\}$, $i \leftarrow 0$.
0.2 Go to step 1.

Step 1.
1.1 $\tilde{d} \leftarrow d_i$, $\tilde{b}(0) \leftarrow 0$.
1.2 For slices $t = 0, 1, ...$ calculate $\tilde{b}(t) \leftarrow \max\{0, \tilde{b}(t-1) - c\} + r_t(\tilde{d})$.
1.3 If $\max_t \tilde{b}(t) > B^e_{max}$ then $i \leftarrow i + 1$ and go to Step 1.1
else $\tilde{d}$ is solution of the MINMAX optimization task (5).

The algorithm described above is called the *consecutive search algorithm.*

Theorem 1. *There is no such sequence of maximum absolute differences $y_1, y_2, \ldots$ that does not lead to the transmitter buffer overflow and $\max_t y_t < \tilde{d}$, where $\tilde{d}$ is the value found by the consecutive search algorithm.*

Proof. Let us suppose that *consecutive search algorithm* has stopped at the step i. Then for each step $j < i$ for every slice t maximum absolute difference $x_t = d_j$ was chosen. From consecutive search algorithm description follows that after encoding slice τ buffer $\tilde{b}(\tau)$ is overflowed:

$$\tilde{b}(\tau) > B^e_{max}. \tag{6}$$

Let us choose any sequence of maximum absolute differences $y_1, y_2, \ldots$, where $y_t \le d_j$, and $b(t)$ is the number of bits in transmitter buffer, when slice t is encoded with y_t value. Then

$$y_t \le x_t, \tag{7}$$

consequently,

$$r(y_t) \ge r(x_t) \tag{8}$$

So if $\tilde{b}(0) = b(0) = b_0$, then from (1) and (8) follows that $\tilde{b}(t) \le b(t)$. It means that exists such $\tau' \le \tau$ that

$$b(\tau') > B^e_{max}. \blacksquare \tag{9}$$

IV. PROPOSED RATE CONTROL DESCRIPTION

Consecutive search is hypothetic algorithm that shows the decision of (5), but can not be implemented in real-time system, because it is impossible to restart data transmission after buffer overflowing.

Therefore, this paper proposes algorithm that allows to find the estimation of $\tilde{d}$ for the consecutive search algorithm. Consider $\hat{d}_t$ to be the estimation of $\tilde{d}$ value. It is supposed to estimate $\tilde{d}$ value as follows. All slices are compressed with maximum absolute difference $\hat{d}_t$ until the number of bits in the buffer will not exceed B^e_{max}. Buffer overflowing means that it is impossible to hold the $\hat{d}_t$ value for the given channel throughput. Consequently, slices are not placed into the buffer until it becomes empty (in this case decoder gets corresponding slice from previous frame) and then the estimation of $\hat{d}_t$ increases. Algorithm consists of the following three steps.

Step 0. (*Initialization*)
0.1 Set $\hat{d}_0 \leftarrow d_0$, $t \leftarrow 0$, $b(0) \leftarrow 0$.
0.2 Go to step 1.

Step 1. (*Buffer accumulation*)
1.1 $t \leftarrow t + 1$, $\hat{d}_t \leftarrow \hat{d}_{t-1}$.
1.2 $b(t) \leftarrow \max\{0, b(t-1) - c\}$.
1.3 Compress slice t with maximum absolute difference $\hat{d}_t$.
1.4 If $b(t) + r(\hat{d}_t) > B^e_{max}$ go to Step 2.1
else $b(t) \leftarrow b(t) + r(\hat{d}_t)$ and go to Step 1.1

Step 2. (*Buffer emptying*)
2.1 $t \leftarrow t + 1$, $\hat{d}_t \leftarrow \hat{d}_{t-1}$.
2.2 $b(t) \leftarrow \max\{0, b(t-1) - c\}$.
2.3 If $b(t) = 0$ then $\hat{d}_t \leftarrow \hat{d}_t + \Delta d$ and go to Step 1.1
else go to Step 2.1

Theorem 2. *Consider that consecutive search algorithm with transmitter buffer of size B^e_{max} finds the maximum absolute difference value $\tilde{d}$. Then for the proposed algorithm with initial value $\hat{d}_0 \leq \tilde{d}$, the inequality $\hat{d}_t \leq \tilde{d} + \Delta d$ holds true for any moment of time t.*

Proof. Let $\tilde{b}(t)$ be the buffer size for the consecutive search algorithm. From its description

$$\tilde{b}(t) \leq B^e_{max}. \tag{10}$$

Let us suppose that $\hat{d}_0 \leq \tilde{d}$ and at the moment of time τ this inequality holds true firstly:

$$\tilde{d} \leq \hat{d}_\tau < \tilde{d} + \Delta d. \tag{11}$$

So at this moment the number of bits in the buffer (see step 2.3) is:

$$b(\tau) = 0. \tag{12}$$

From (11) for $t \geq \tau$ following inequality holds true:

$$r(\hat{d}_t) \leq r_t(\tilde{d}), \tag{13}$$

so that from (1), (10), (12) and (13) follows that at the moment of time $t \geq \tau$ the number of bits in the buffer is:

$$b(t) \leq \tilde{b}(t) \leq B^e_{max}. \tag{14}$$

Thereby, from the moment of time τ the statement of the step 1.4 of this algorithm fails. Consequently, algorithm will not reach the step 2.3 and parameter $\hat{d}_t$ will not be increased. ■

V. Scene changing detection and virtual buffer conception

Now let us take a look at the video sequences that consist of several scenes $s_0, s_1, \ldots, s_n$. Then MINMAX optimization task (5) should be applied for each scene. Let $\tilde{d}(s_i)$ be a solution provided by consecutive search for scene s_i. If $\tilde{d}(s_{i+1}) \geq \tilde{d}(s_i)$, then algorithm proposed above will adapt to a new scene. However, if $\tilde{d}(s_{i+1}) < \tilde{d}(s_i)$, then algorithm will not decrease $\hat{d}_t$, that means that the quality will not be improved.

Therefore, to overcome this problem we introduce an approach based on *virtual buffer* conception. For each slice t the following value is calculated:

$$b^-_{virt}(t) \leftarrow \begin{cases} b(t), \ if \ t = t^*, \\ \max\{0, b^-_{virt}(t-1) - c\} + r_{virt}(\hat{d}_t - \triangle d^-_{virt}), \ if \ t \neq t^*, \end{cases} \tag{15}$$

where t^* is a number of the first slice in the current frame.

In addition, the difference between the number of bits that is placed into the buffer and maximum number of bits that could be transmitted is accumulated:

$$\triangle r^-_{virt} \leftarrow \sum_{i=t^*}^{t^*+N-1} r_{virt}(\hat{d}_t - \triangle d^-_{virt}) - N \cdot c, \tag{16}$$

where N is a number of slices in the frame.

To predict the bit size of slice $r_{virt}(\hat{d}_t - \triangle d^-_{virt})$ compressed with the maximum absolute difference $\hat{d}_t - \triangle d^-_{virt}$ the lengthes of the codewords without forming the output bit stream are accumulated.

Let us take a look on the virtual buffer conception. If $\triangle r^-_{virt} > 0$, the number of bits sent to the transmission buffer is more then the number of bits sent to the channel and this can bring to the effect of buffer overflowing during the transmission of the next frames. On the other side, the bit size distribution for slices in each frame may be so, that buffer may overflow even if $\triangle r^-_{virt} \leq 0$. Therefore, in addition $b^-_{virt}(t)$ is calculated. Thus, if before the encoding of the slice t^* the following statements are fulfilled:

$$\begin{cases} \max\limits_i b^-_{virt}(i) \leq B^e_{max}, \ i \in \{t^* - N, \ldots, t^* - 1\} \\ \triangle r^-_{virt} \leq 0 \end{cases} \tag{17}$$

and rate control was not in the buffer emptying mode during coding of previous frame, the maximum absolute difference value is modified as follows:

$$\hat{d}_t \leftarrow \max\{0, \hat{d}_t - \triangle d^-_{virt}\}. \tag{18}$$

VI. Algorithm adaptation time decrease

To adaptation time decrease of the proposed rate control algorithm to the statistical properties of a new scene, the second virtual buffer is used. For each slice t the following value is calculated:

$$b^+_{virt}(t) \leftarrow \begin{cases} 0, \ if \ t = t^*, \\ \max\{0, b^+_{virt}(t-1) - c\} + r_{virt}(\hat{d}_t + \triangle d^+_{virt}), \ if \ t \neq t^*. \end{cases} \tag{19}$$

As was described in the previous section the following value is calculated:

$$\triangle r^+_{virt} \leftarrow \sum_{i=t^*}^{t^*+N-1} r_{virt}(\hat{d}_t + \triangle d^+_{virt}) - N \cdot c. \tag{20}$$

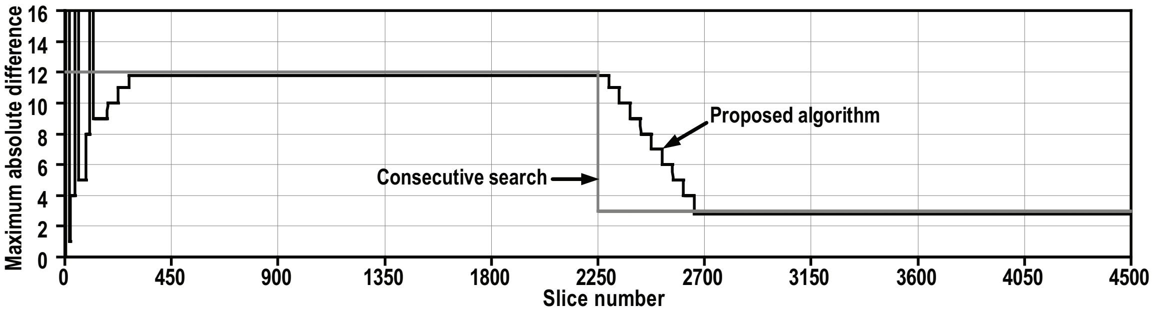

Fig. 2. The dependence of maximum absolute difference from slice number for the consecutive search algorithm and the proposed algorithm of rate control for JPEG-LS.

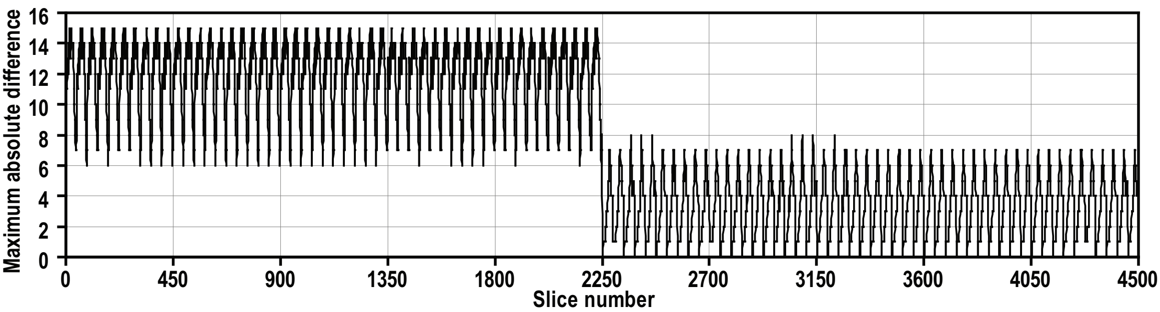

Fig. 3. The dependence of maximum absolute difference from slice number for the algorithm of rate control from [4], [5].

If $\triangle r^+_{virt} > 0$, the number of bits sent to the transmission buffer is more then the number of bits sent to the channel and this can bring to the effect of buffer overflowing during the transmission of the next frames, so it is needed to increase maximum absolute difference $\hat{d}_t$ in that case. Moreover, the bit size distribution for slices in each frame may be so, that buffer may overflow even if $\triangle r^+_{virt} \leq 0$. Therefore, in addition $b^+_{virt}(t)$ is calculated. Thus, if before the encoding of the slice t^* any of the following statements hold true:

$$\left[\begin{array}{l} \max\limits_i b^+_{virt}(i) > B^e_{max}, i \in \{t^* - N, \ldots, t^* - 1\}, \\ \triangle r^+_{virt} > 0, \end{array}\right. \tag{21}$$

the maximum absolute difference value is modified as follows:

$$\hat{d}_t \leftarrow \min\{d_{max}, \hat{d}_t + \triangle d^+_{virt}\}. \tag{22}$$

To decrease the adaptation time for the case when maximum absolute difference value is close to consecutive search the following value is calculated:

$$\triangle r \leftarrow \sum_{i=t^*}^{t^*+N-1} r(\hat{d}_t) - N \cdot c. \tag{23}$$

Thus, if $\triangle r > 0$ before the encoding of the slice t^* the maximum absolute difference value is modified as follows:

$$\hat{d}_t \leftarrow \min\{d_{max}, \hat{d}_t + \triangle d\}. \quad (24)$$

VII. Experimental results and conclusion

For practical results was used the JPEG-LS codec software implementation [10]. The performance of discussed algorithm was tested on the video sequence which have 1280×720 frame resolution, 30 fps frame rate and 1280×16 slices. This video was combined from two different videos to make abrupt scene changing in the sequence at the frame number 50 (slice number 2250). Latency value was set to 10 ms, channel throughput is equal to the compression ratio in 7 times, parameters $\Delta d = 1$, $\Delta d^{-}_{virt} = 1$, $\Delta d^{+}_{virt} = 4$. Parameter $d_0 = 0$ was used to introduce algorithm work in particular conditions. Figure 2 shows the dependence between maximum absolute difference from slice number for the consecutive search algorithm and proposed rate control.

Figure 3 shows the dependence between maximum absolute difference from slice number for algorithm described in [4], [5]. These algorithms are based on multi-pass encoding and compress each slice close to the channel rate c. Therefore, maximum absolute difference value varies from slice to slice and brings to the following effect: some compressed slices could have high visual quality, but the quality of other slices could be low. In contrast to the described above algorithms, the proposed algorithm is based on the single-pass compression and provides with the constant visual quality for all slices. All this makes it more preferable for real-time latency-constrained video transmission over high throughput channels.

References

[1] WirelessHD Consortium, http://wirelesshd.org/

[2] IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANS), Task Group 3c, Millimeter Wave Alternative PHY, http://www.ieee802.org/15/pub/TG3c.html

[3] FCD 14495, Lossless and near-lossless coding of continuous tone still images (JPEG-LS) *ISO/IEC JTC1/SC29 WG1 (JPEG/JBIG)*, 1997.

[4] T. Tsai, S. Kao and Y. Lee "The segment-based rate control algorithm in JPEG-LS for bandwidth-efficiency applications" *Multimedia and Expo, 2008 IEEE International Conference on*, pp. 793–796, 2008.

[5] J. Jiang, C. Grecos, "A low cost design of rate controlled JPEG-LS near lossless image compression" *Image and Vision Computing*, no. 19, pp. 153 – 164, 2001.

[6] 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 Redundancy07*, St.-Petersburg, Russia, pp. 13–17., 2007.

[7] E. Belyaev, A. Turlikov and A. Ukhanova. "Rate-control algorithms testing by using video source model" *The 15-th International Conference on Communications*, St.-Petersburg, Russia, 2008.

[8] 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.

[9] N. Cherniavsky, G. Shavit, M. F. Ringenburg, R. E. Ladner, "Multistage: A MINMAX Bit Allocation Algorithm for Video Coders" *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 1, pp. 59 – 67, 2007.

[10] http://www.hpl.hp.com/loco/software.htm