Optimized Bit Extraction of SVC Exploiting Linear Error Model

Wenyao Zhang, Jun Sun*, Jiaying Liu and Zongming Guo
Inst. of Comput. Sci. & Technol., Peking Univ. Beijing 100871, China
Telephone: (+86) 10 82529523 Email: sunjun@pku.edu.cn

Abstract— **The Scalable Video Coding (SVC) extension of the H.264/AVC video coding standard supports fidelity or quality (SNR) scalability. The quality enhancement packets would be discarded in case of limited network capacity, which calls for an optimized bit extraction strategy. In this paper, we first analyze the linear feature in H.264/AVC video coding. A linear error model is also constructed using this feature in case of SVC quality scalability. Then based on the linear error model, the rate and distortion (R-D) impact of each quality enhancement packet over the whole sequence is obtained. Finally a new priority assigning algorithm is designed for a more efficient extraction, giving high rank to those with great R-D impacts. Extensive experiments are presented to demonstrate the accuracy of the linear error model and the validity of the priority assigning algorithm. Tests on the set of eight standard video sequences show the quality promotion under any bitrate constraint, and a fidelity gain up to 0.4 dB PSNR is achieved by the proposed strategy, compared to the JSVM reference software with Quality Layer information.**

I. Introduction

The scalable extension to H.264/AVC[1], called Scalable Video Coding (SVC)[2], supports three types of scalability: spatial, temporal, and quality scalability. When the coded scalable video signals are transmitted over the Internet, there would be much variation lying in the bandwidth. So we have to discard some SVC data packets in case of bad network traffic, where an optimal strategy is needed. If the basic traversal strategy is adopted, we have to extract from the bitstream at every possible bitrate, decode the substream and compare the decode quality of every possible substream. Considering a K-frame video segment having L_Q quality scalable layers, the number of possible extraction points is ${L_Q}^K$, assuming base layer packets are always retained. The exponential computational complexity in basic traversal and the essential time-consuming characteristic of decoding make this scheme unrealistic, whereas modeling and suboptimal schemes are required.

Amonou[3] proposed an algorithm to assign priorities by calculating the product of rate and distortion increment. To reduce the computational complexity for obtaining rate and distortion information, two simplified patterns – dependent pattern and independent pattern – are designed. In either pattern, for each temporal level and quality layer, there is a corresponding extraction and decoding, leading to the computational complexity of $O(L \log_2 K)$. However, there still exists promotion space of R-D performance for the simplified patterns. Sun[4] and Maani[5] constructed models containing undetermined coefficients, to estimate drift propagation, and measured R-D impact by the derivative of distortion over rate. Their linear-like models aimed at estimating the interference of MSE (Mean Square Error) or PSNR (Peak Signal to Noise Ratio) among neighboring frames, to further promote the R-D optimization performance. However, since the mismatching between quadratic calculations in objective distortion criteria like MSE or PSNR and the essential linear operations in the coding process, a more accurate modeling method is needed to directly estimate pixel value errors before MSE or PSNR is calculated.

The main steps in H.264/AVC coding – prediction, transform and quantization – are proximately linear operations, if manipulations like rounding and clipping are neglected. So the whole coding process can be described in a matrix format[6]. Additionally, quality scalable extension of H.264/AVC follows a layered coding approach with "inter-layer prediction"[1], where the lower layer image samples are employed as a linear predictor for encoding of the higher layer ones. In this paper, we extend the linear coding model of H.264/AVC[6] to construct a linear error model for SVC quality scalability, which can be used to estimate pixel value errors and calculate MSE or PSNR more precisely, rather than directly estimate MSE or PSNR. Then the error brought about by the removal of each enhancement packet, including error pixel values in neighboring frames (drift), can be obtained independently, using this model. Here the distortion impact of an enhancement packet is presented as a vector, of which each element represents what pixel value difference would be generated if the specific packet is removed from the bitstream. The rate and distortion information is then utilized in a packet-discarding simulation process using Greedy Algorithm[7], to achieve a relatively optimal R-D performance. The packet-discarding order in simulation is then assigned to each packet as its priority.

The next section introduces the linear error model and its verification for a quality scalable video sequence. Sec. III describes the optimization algorithm in details, including the acquisition of rate and distortion information, and priority assignment strategy. Experimental verification and conclusion are provided in Sec. IV and Sec. V respectively.

*Corresponding author. This work was supported by National Nature Science Foundation of China under contract No. 60902004, Beijing Municipal Natural Science Foundation under contract No. 4102025, Beijing Nova program under contract No. 2010B001.

978-1-4673-0219-7/12/$31.00 ©2012 IEEE

II. Linear Error Model for Quality Scalable Video

In H.264/AVC, the decoding process involves a linear feature described in [6], neglecting any rounding, clipping, and deblocking filtering operation. In this model, the reconstructed samples are obtained as a linear combination of previously reconstructed samples, the residual samples, and a static predictor. Considering a group of K pictures, each of width W and height H, we obtain the following relationship:

$$s = \mathbf{M}s + \mathbf{T}c + p\,, \tag{1}$$

where the vectors s, c, and p are $N \times 1$ column vectors with $N = K \times W \times H$. s refers to the reconstructed sample values, c the transform coefficient values and p a static predictor. M and T are $N \times N$ square matrices such that the product $\mathbf{M}s$ gives the MCP (Motion Compensated Prediction) signal vector and $\mathbf{T}c$ gives the residual sample values. The actual values of $\mathbf{M}$ depend on the selected macroblock types, reference indices and motion vectors, whereas the actual values of $\mathbf{T}$ depend on the chosen QP (Quantization Parameter) values.

A. Extension for Quality Scalability

We apply the linear decoding model for the case of quality scalability. For a bitstream just containing the base layer,

$$s_B = \mathbf{M}s_B + \mathbf{T}_B c_B + p\,, \tag{2}$$

where the subscript B refers to base layer variables.

Considering one bitstream containing an enhancement packet subset I_1 out of the universal enhancement packet set U, as shown in Fig.1, we obtain the reconstruction relationship

$$s_{I_1} = \mathbf{M}s_{I_1} + \mathbf{T}_B c_B + \Big(\sum_{i \in I_1} \mathbf{T}_i c_i\Big) + p\,, \tag{3}$$

where i refers to an enhancement packet as an element of I_1. c_i is a vector containing coefficients within the enhancement packet i. $\mathbf{T_i}$ is the transform matrix depending on the QP value of i. $\mathbf{T}_i c_i$ gives the residual sample values decoded from i.

For another bitstream containing a subset I_2, we have

$$s_{I_2} = \mathbf{M}s_{I_2} + \mathbf{T}_B c_B + \Big(\sum_{i \in I_2} \mathbf{T}_i c_i\Big) + p\,. \tag{4}$$

Their reconstruction difference $e = s_2 - s_1$ is obtained by (4) - (3):

$$(s_{I_2} - s_{I_1}) = \mathbf{M}(s_{I_2} - s_{I_1}) + \Big(\sum_{i \in I_2} \mathbf{T}_i c_i - \sum_{i \in I_1} \mathbf{T}_i c_i\Big)\,, \tag{5}$$

$$\Rightarrow e = s_{I_2} - s_{I_1} = (\mathbf{I} - \mathbf{M})^{-1}\Big(\sum_{i \in I_2} \mathbf{T}_i c_i - \sum_{i \in I_1} \mathbf{T}_i c_i\Big)\,. \tag{6}$$

If I_1 is a subset of I_2, i.e. $I_1 \subseteq I_2$, then

$$e = (\mathbf{I} - \mathbf{M})^{-1} \sum_{i \in I_2 \setminus I_1} \mathbf{T}_i c_i\,, \tag{7}$$

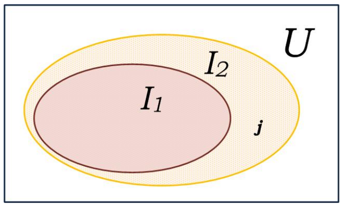

Fig. 1. Enhancement Packet Set Diagram for e_j Acquisition

Particularly, if the difference of I_2 and I_1 is a single enhancement packet j, i.e. $\{j\} = I_2 \setminus I_1$ as shown in Fig. 1, the difference of two reconstruction vectors corresponding to packet j can be written as

$$e_j = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{T}_j c_j\,. \tag{8}$$

Obviously, e_j has relation only with $\mathbf{T}_j$ and c_j, and no relation with all other enhancement packets. So the subtraction of pixel values decoded from I_2 and those from I_1 can be regarded as the "error vector" of enhancement packet j. We can obtain the error vector of each enhancement packet by subtraction between video sequences reconstructed from two enhancement packet subsets I_1 and I_2, with $I_1 \subseteq I_2$ and $\{j\} = I_2 \setminus I_1$.

Additionally, the difference between the video sequence decoded from full enhancement packets and the original sequence is marked as e_{full}. It is mainly generated from the quantization process of encoding, and it would be used as the initial value of global error vector e_{global} in Sec. III-B.

B. Linear Error Model Verification

Based on the linear error model described above, we can obtain error vectors of all enhancement packets. In Eq. (7), we let I_2 be U, the universal set of all enhancement packets, and let I_1 be a set I_x. Then the error vector of a video sequence decoded from I_x can be written as

$$e(I_x) = e_{full} + \sum_{i \in \overline{I_x}} e_i\,, \tag{9}$$

where $\overline{I_x}$ is the complement of the subset I_x within U. Eq. (9) has taken into consideration the error between the video sequence decoded from full enhancement packets and the original sequence, written as e_{full}.

To verify this linear error model, we conduct an experiment where some random sets of enhancement packets are removed from a video bitstream, for further comparing the actual distortion and distortion estimated by the linear error model. Experimental results of the sequence *Foreman* is partly shown in Fig. 2. The horizontal axis stands for sample point numbers of those random selections, and the vertical axis for MSE measurement. The solid line represents the actual MSE of a randomly extracted substream, and the dot line shows the estimated MSE for each enhancement packet combination, using our Linear Error Model. A 0.6% maximum relative error of MSE demonstrates the accuracy of the proposed model.

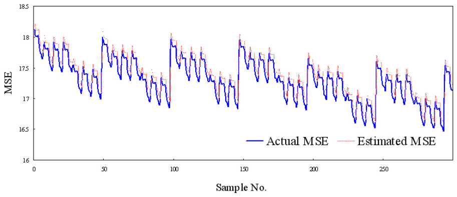

Fig. 2. Result of SVC Linear Error Model Verification

III. Priority-Assigning Algorithm

The experiment of the proposed algorithm can be roughly separated into two parts: error vector acquisition, and priority assignment. The experimental results are shown in the following section.

A. Error Vector Acquisition

In order to acquire error vectors corresponding to all enhancement packets efficiently, we separate all enhancement packets into several groups, so that in one single group multiple pixel value error vectors are obtained independently.

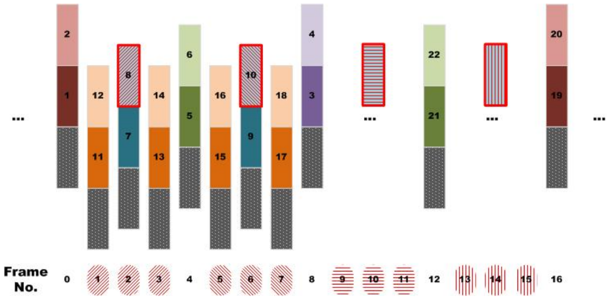

Fig. 3. SVC GOP Structure

Considering a video bitstream with 8-frame GOPs (Group of Pictures) shown in Figure 3, base layer packets are colored black, and enhancement packets with the same color are allocated in one group. For example, the loss of Packet 8 would bring about distortion in Frame 1, 2 and 3, while Packet 10 in Frame 5, 6 and 7, respectively (Frame number starts from 0). So we can do subtraction for this group, for one time, to gain error vectors e_8 and e_{10}, for both packets 8 and 10, as well as peer packets in all other GOPs. Note that these error vectors are highly sparse ones, because the discarding of a single enhancement packet can bring distortion only to a limited number of frames. To acquire all error vectors, the number of required extraction and decoding is approximately the number of enhancement layers $(L_Q - 1)$ times the temporal layer number L_T. For the example in Fig. 3, $L_Q = 3$ and $L_T = 4$. The theoretical extraction-and-decoding number is $(3-1) \times 4 = 8$. However, we have to do the same manipulation for $L_Q - 1 = 2$ more times, to dissociate the distortion impact of "key frame"[1] (e.g. Frame 0, 8, 16, etc.) packets. The loss of one enhancement packet in a key frame would bring distortion to two neighboring GOPs. As an instance, the loss of Packet 4 would propagate drift into Frame 1 to 15, which would interfere with the distortion impact of packets in Frame 0 and 16 (Packet 1, 2 and 19, 20). So key frame enhancement packets within the same quality layer (e.g. Packet 2, 4 and 20) should be allocated in two groups – one for key frame packets of even number GOPs (Packet 2 and 20), the other for those of odd number GOPs (Packet 4). So the actual number of extraction and decoding for error vector acquisition is $(L_Q - 1) \cdot (L_T + 1)$, approximately the same with JSVM reference software[1]. Since computation is mainly consumed in bitstream decoding in the proposed algorithm, the computation complexity has no obvious promotion compared with JSVM.

B. Priority Assignment

We simulate the packet removal process of one bitstream, from with full packets to with just base layer ones, using hill-climbing strategy, to assign priorities to all enhancement packets. Once an enhancement packet is assumed to be discarded, mathematically we measure its impact on rate and distortion by PSNR decrement per bitrate $\phi_i = \left|\frac{\partial(\mathrm{PSNR})}{\partial R}\right|$. Here we utilize error vectors previously calculated to simulate the rate and distortion impact.

Initially the global error vector e_{global} is set to e_{full}. Each time a packet is to be discarded, we find the packet m having the least R-D impact on the whole sequence, i.e. $\phi_m = \min_{i \in I_{top}} \phi_i$, where I_{top} stands for the set of top-layer packets, those able to be discarded in the following step. This packet, m, would then be removed from the bitstream, and its error vector e_m be added to the current global error vector e_{global}, after which the global PSNR is calculated. We repeat the same process until all enhancement packets are discarded. The packet removal order is then regarded as priorities. Note that the computation complexity is $O(K^2)$, with K being the number of frames involved in the priority assignment process. So an optimization window is needed, especially for long video sequences, to tradeoff between optimization performance and computational complexity. Note that computation complexity of priority assignment is negligible compared with decoding of error vector acquisition.

IV. Experimental Results

In this section, we evaluate the performance of our proposed bit extraction strategy for SVC. The experiments are conducted using the SVC reference software JSVM_9_16. We used a standard hierarchical B coding structure with three quality layers (including base layer) at CIF resolution and CABAC entropy coding. The difference between enhancement layers and base layer quantization parameter (QP) was set equal to 3. Experimental results are shown in Table. I and Fig. 4 to 7. It can be seen from Table. I that additional gain up to 0.4 dB luma PSNR can be achieved over the JSVM reference software QL (Quality Layer) performance (the blue curve in Fig. 4 to 7), and up to 1.2 dB over JSVM without QL (black curve), under various bitrate constraints. Clearly the proposed algorithm performance curve (red curve) is on top of the other two, without exception.

TABLE I

ALGORITHM PERFORMANCE: LUMA PSNR GAIN ON SAMPLE SEQUENCES COMPARED WITH JSVM

Sequence		*Bus*	*City*	*Crew*	*Football*	*Foreman*	*Harbour*	*Mobile*	*Soccer*
JSVM QL[a] (dB)	Max[b]	0.23	0.17	0.20	0.36	0.20	0.41	0.30	0.29
	Ave[c]	0.11	0.06	0.10	0.19	0.08	0.14	0.16	0.15
JSVM no QL (dB)	Max	1.23	0.19	0.53	0.64	0.42	0.37	0.68	0.52
	Ave	0.42	0.09	0.22	0.32	0.18	0.12	0.42	0.21

[a] Comparing the proposed algorithm performance with JSVM with QL information

[b] Maximum PSNR gain through all bitrate constraints

[c] Average PSNR gain through all bitrate constraints

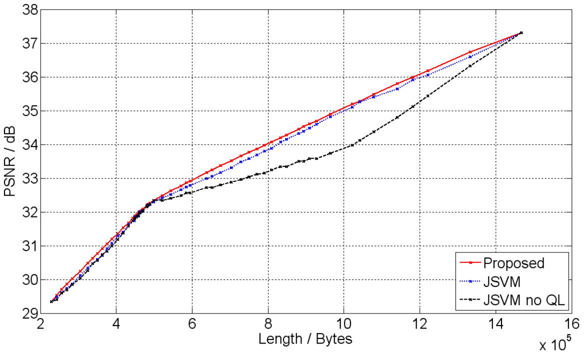

Fig. 4. Bus

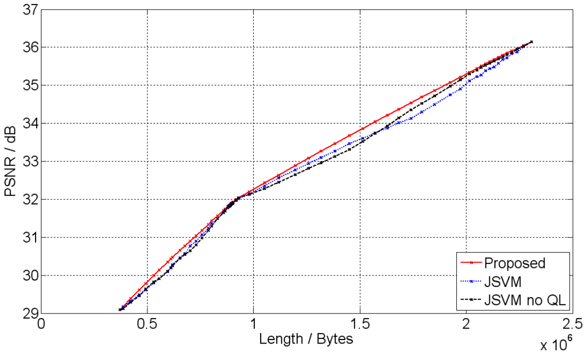

Fig. 5. Harbour

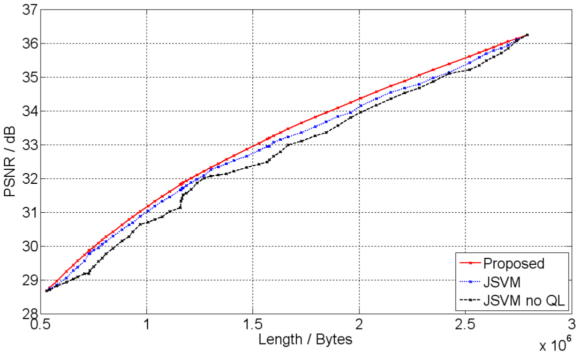

Fig. 6. Mobile

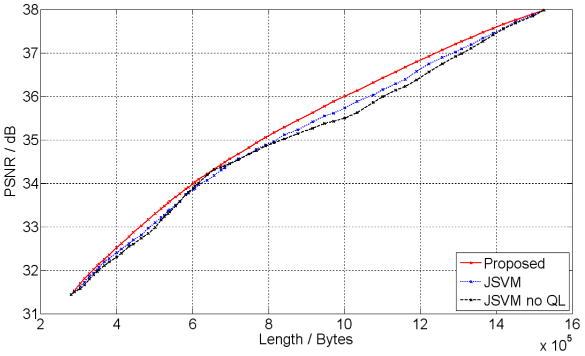

Fig. 7. Soccer

V. CONCLUSION

We have presented an optimized bit extraction approach of SVC in quality scalability with inter-layer prediction, based on the new linear error model. A new error-modeling method concerning pixel values is proposed in this paper. Experimental results demonstrate R-D performance promotion using this algorithm, without exception, in case of eight reference video sequences. Additional gain of up to 0.4 dB luma PSNR under the same bitrate constraints can be observed, compared with the JSVM reference software with QL information.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, A. Luthra, "Overview of the H.264/AVC Video Coding Standard", in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[2] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz, M. Wien, eds., "Joint Draft ITU-T Rec. H.264—ISO/IEC 14496-10/Amd.3 Scalable Video Coding", Joint Video Team, Doc. JVT-X201 Jul. 2007.

[3] I. Amonou, N. Cammas, S. Kervadec, S. Pateux, "Optimized Rate-distortion Extraction with Quality Layers in the Scalable Extension of H.264/AVC" in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1186–1193, Sep. 2007.

[4] J. Sun, W. Gao, D. Zhao, W. Li, "On Rate-distortion Modeling and Extraction of H.264/SVC Fine-Granular Scalable Video" in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 3, pp. 323–336, Mar. 2009.

[5] E. Maani, A. K. Katsaggelos, "Optimized Bit Extraction Using Distortion Modeling in the Scalable Extension of H.264/AVC" in *IEEE Trans. Image Processing*, vol. 18, no. 9, pp. 2022–2029, Sep. 2009.

[6] M. Winken, H. Schwarz, T. Wiegand, "Joint Rate-distortion Optimization of Transform Coefficients for Spatial Scalable Video Coding Using SVC" in *Proc. International Conference on Image Processing 2008*.

[7] P. E. Black, "Greedy Algorithm" in *Dictionary of Algorithms and Data Structures* [online], U.S. National Institute of Standards and Technology, Feb. 2005. http://www.nist.gov/dads/HTML/greedyalgo.html