AN EFFICIENT METHOD FOR NO-REFERENCE H.264/SVC BITSTREAM EXTRACTION

Shengbin Meng, Jun Sun, Yizhou Duan, Zongming Guo[*]

Institute of Computer Science and Technology,
Peking University, Beijing, China, 100080

ABSTRACT

This paper investigates the no-reference SVC bitstream extraction problem and presents an efficient solution to approximate the "optimal" extracted sub-stream. First, we introduce a linear error model to accurately estimate the distortion caused by discarding any combination of packets, even when the original sequence is not available. Then we propose a greedy algorithm to decide each packet's priority according to its R-D impact. The priority value of packets can be stored in the bitstream and used for R-D optimized extraction. Experimental results show that our bitstream extraction method can achieve a significant PSNR gain compared to the extractors of JSVM, without computational complexity increment. Comparison with other methods also demonstrates the advantage of the proposed method.

Index Terms— SVC, bitstream extraction, distortion model, no reference

1. INTRODUCTION

Scalable Video Coding (SVC) [1] is an extension of the H.264/AVC video coding standard and has provided an effective way to deal with the new challenges of modern video applications. SVC allows efficient scalability in terms of video frame rate, spatial resolution, and picture fidelity, with only one single bitstream. The SVC bitstream is composed of Network Abstract Layer Units (NALUs), which are separated into multiple layers: one base layer for basic video quality and several enhancement layers for video quality improvement. By discarding NALUs of the enhancement layers, the bitstream can be extracted as required, providing the most appropriate data rate and video service according to the client capability and network conditions.

SVC bitstream extraction is an essential issue that has attracted many attentions in the scalable video research. Under the constraint of a given target bitrate, SVC bitstream extraction aims at selecting NALUs from the whole stream to form an "optimal" sub-stream, which represents video with best quality, or minimal distortion. It should be noted that, in some scenarios, the bitstream extraction needs to be conducted without access to the original YUV sequence. For example, if the bitstream is not extracted at the encoding side, it's most likely that the original sequence is not available. This makes the bitstream extraction a more difficult problem, since the distortion is harder to obtain for lacking of reference. If the distortion caused by discarding some packets cannot be accurately estimated, the video quality of the extracted stream cannot be well guaranteed. Comparing to the problem of SVC bitstream extraction with reference, this kind of "no-reference bitstream extraction problem" has attracted far less attention. In this paper, we thoroughly investigate the no-reference SVC bitstream extraction problem and propose an efficient method to solve it. For the quality scalable video coding, a linear error model is introduced to estimate the distortion caused by discarding any combination of the enhancement packets. Then based on this distortion estimation technique, we propose a greedy-like method to determine the priority of each packet in the SVC stream and thus achieve RD-optimized bitstream extraction. The proposed algorithm has proved its advantage over other state-of-art methods in the experiments.

The rest of this paper is organized as follows. In section 2, we analyze the bitstream extraction problem and some related works. In section 3, we first introduce the linear error model for accurate distortion estimation; then based on that, we propose our own greedy method for no-reference bitstream extraction. Experimental results are shown in section 4 and conclusions are made in section 5.

2. PROBLEM ANALYSIS AND RELATED WORKS

SVC bitstream extraction is substantially a combinatorial optimization problem. More specifically, it is in the form of "0-1 Knapsack Problem". Each NALU data packet can be treated as an object with certain value (its contribution to the video quality) and weight (its size). The solving process is mainly to decide whether or not a packet should be included in the final extracted sub-stream; or in the term of "0-1 Knapsack Problem", to choose which object to pack up.

Generally, the optimal solution of a typical "0-1 Knapsack Problem" can be found using the dynamic programming algorithm. However, this is not operational for the bitstream extraction problem, mainly due to the dependence between

[*] is the corresponding author. This work was supported by National Natural Science Foundation of China under contract No. 61271020 and National Key Technology R&D Program of China under Grant 2014BAK10B02.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

packets. First, the value of a packet is not obvious and certain, since a packet's contribution to video quality may depend on the other packets. Second, the subset of packets to be extracted cannot be chosen arbitrarily, since if a packet is included, all the packets it depends on must also be included.

In [2], Amonou et al. proposed a bitstream extraction framework based on the concept of Quality Layers. This framework has been adopted by the SVC reference software, i.e., JSVM [1], and achieves better R-D performance compared with the basic extractor of JSVM. In Amonou's framework, a Quality Layer value, which reflects a packet's rate-distortion (R-D) impact, is calculated and used for R-D optimized bitstream extraction. The process of obtaining distortion impact of every packet is extremely time consuming, and the distortion estimation accuracy can also be further improved. So other distortion/error models were proposed to either reduce the computational complexity or improve the estimation accuracy. Sun [3] and Maani [4] constructed models to calculate distortion based on the drift propagation between frames. Sun's method is almost as good as JSVM for the performance, but with a much reduced complexity. Maani's model can achieve better estimation accuracy than JSVM; however, in order to get robust and accurate model parameters, more computation is needed for training.

In our previous work [5], we proposed a linear error model which exploits the linear feature of pixel value errors and achieves good performance in distortion estimation and bitstream extraction with reference. In this paper, we adapt this model to the no-reference bitstream extraction problem. A new efficient algorithm is designed to meet the special characteristic of the no-reference case, and thus better performance is achieved.

3. NO-REFERENCE BITSTREAM EXTRACTION BASED ON LINEAR ERROR MODEL

In this section, we first introduce the linear error model which can be used to accurately estimate distortion, and then propose the new bitstream extraction method based on that.

3.1. The Linear Error Model

In H.264/AVC, the decoding process involves a linear feature described in [6], neglecting any rounding, clipping, and deblocking filtering operation. This feature is expressed as the following equation:

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

It means, the reconstructed samples of a GOP (Group of Pictures), denoted by s, can be viewed as a linear combination of previously reconstructed samples in the GOP, the residual samples, and a static predictor. In (1), c refers to the quantized transform coefficient values and p is a static predictor; $\mathbf{M}$ and $\mathbf{T}$ are square matrices such that the product $\mathbf{M}s$ gives the MCP (Motion Compensated Prediction) signal and $\mathbf{T}c$ gives the residual 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.

In our previous work [5], we applied the above linear decoding model to the case of quality scalability of SVC and proposed a Linear Error Model (LEM). It's concluded that the pixel error caused by discarding an enhancement packet j can be expressed as:

$$e_j = (\mathbf{I} - \mathbf{M})^{-1}\mathbf{T}_j c_j \, , \tag{2}$$

which is only determined by packet j, and has no relation with other enhancement packets. This value is called the "error vector" of 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 $I_2 \setminus I_1 = \{j\}$. Once all packets' error vectors are obtained, the distortion caused by discarding a group of packets can be estimated by adding up the error vector of each packet in the group.

Details about the deduction of (2) and the correctness of the above model can be found in [5]. We have done the verification by removing some random sets of enhancement packets from a video stream and comparing the actual MSE with the MSE estimated with this model. A relative estimation error of 0.6 % demonstrates the accuracy of the model. For an SVC stream with N_Q quality layers and N_T temporal layers, to acquire the error vectors of all packets will need $(N_Q - 1) \cdot (N_T + 1)$ times of extraction and decoding, approximately the same as the Quality Layer information acquisition in JSVM. In another word, this new model will not increase computational complexity comparing with JSVM when used in our bitstream extraction method.

3.2. Priority Assignment and Bitstrean Extraction

In this sub-section, we utilize the Linear Error Model to enable R-D optimized bitstream extraction. Since finding the global optimal solution of the bitstream extraction problem is impractical, our bitstream extraction method uses a greedy algorithm [7] to get a sub-optimal solution. Similar to the greedy solution of the "0-1 Knapsack Problem", every time a packet is to be discarded, we choose the packet that has the minimal R-D impact. A packet's R-D impact is mathematically measured by:

$$\Phi_i = \frac{\partial D}{\partial R} \, , \tag{3}$$

where ∂D represents the distortion change brought by discarding this packet, and ∂R is the corresponding rate change. ∂R is simply equivalent to the size of this packet, while ∂D needs to be calculated by subtraction of distortions before and after the packet is discarded.

Typically, distortion is measured relative to the original sequence. However, since the original sequence is not available in the no-reference bitstream extraction scenario, we choose to calculate distortion with respect to the fully reconstructed signal, i.e., the signal reconstructed from the whole bitstream with all enhancement packets. This signal represents the best video quality available and thus is suitable to serve as the reference.

As for the distortion criteria, we use MSE instead of PSNR. The PSNR values would be too large in this case, since discarding some enhancement packets will not bring much significant error relative to the fully reconstructed signal. And MSE proves to be a more effective distortion criteria in the proposed algorithm.

Using the above greedy strategy, we simulate a process of discarding all the enhancement packets from a stream. During this process, the discarding order of a packet is regarded as its priority and recorded. Once the priority values of all the packets are obtained, they can be stored in the bitstream (either in the priority id field of the NALU header or in separate SEI messages) and used for real extraction in the future. The detailed discarding and priority assignment process is described below:

1) Initially, the sequence error vector, e_{seq}, is set to zero.

2) Within the packets that can be discarded, find the packet m having the least RD impact on the sequence, i.e.

$$\Phi_m = \min_{i \in I_{top}} \Phi_i \; , \tag{4}$$

where I_{top} stands for the top layer packets, those currently able to be discarded. The RD impact of a packet is calculated based on (3), as follows:

$$\Phi_i = \frac{\partial D}{\partial R} = \frac{MSE(e_{seq} + e_i) - MSE(e_{seq})}{SIZE(i)}, \tag{5}$$

where $MSE(*)$ means to obtain MSE from the error vector, i.e., to calculate mean square of the error values, and $SIZE(i)$ is the size of packet i. Note that e_i is the error vector corresponding to packet i.

3) The packet, m, is then removed from the bitstream, and its error vector is added to the sequence's error vector, i.e.

$$e_{seq} = e_{seq} + e_m \; . \tag{6}$$

4) The process from 2) to 3) is repeated until all enhancement packets are discarded, and every packet is assigned a priority value according to the removing order.

For long video sequences, an optimization window is usually needed, and the priority assignment and bitstream extraction is conducted independently in each window. The computational complexity of this priority assignment algorithm is $O(K^2)$, with K being the number of frames involved in the priority assignment process, i.e. the size of the optimization window. Note that this computation complexity is negligible compared with the decoding process needed in the error vector acquisition.

4. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed bitstream extraction algorithm, with JSVM 9.16 as the baseline.

In our experiments, each bitstream contains one base layer and a quality enhancement layer, with quantization parameters QP = 33 and QP = 27, respectively. The enhancement layer is further divided into 2 MGS(Medium Grain quality Scalable) layers. For each test bitstream, ten bitrate points are chosen for extraction. For each bitrate point, the extraction is conducted using the bitstream extractors in JSVM, with and without quality layer (denoted as JSVM QL and JSVM Basic, respectively), and the proposed method. The size of the optimization window in the proposed method is set to 4 GOPs, with a GOP size of 8. All of the eight SVC standard sequences with CIF resolution are considered at 30FPS and tested in the experiments.

The performance comparisons for City, Crew, Foreman, and Mobile are shown in Fig. 1. It can be seen that the R-D curves of the proposed method are always on the top. Note that, the first and last point on the curve represent bitstreams with only the base layer packets and with all enhancement packets, respectively, so the results of different extraction methods are the same. For each sequence, the maximum and average PSNR gain through all 10 bitrate points are calculated and listed in Table 1, which shows that our extraction method can achieve a significant performance gain compared with JSVM. Note that, the average PSNR gain over JSVM QL is up to 0.56 for the sequence City.

We also conduct experiments comparing the result of our method with that of the method described in [4] and [5]. While the method in [5] achieves good performance in bitstream extraction with reference, it does not apply well in the no-reference case, and may even get worse extraction than JSVM QL at some points, as shown in Table 2. The data in Table 2 shows that the proposed method brings remarkable improvement in solving the no-reference bitstream extraction problem. The result of the method in [4] depends on its model parameters. For fair comparison, we trained the parameters at the same computation complexity level with our method. It can be seen that the performance of the method in [4] is no better than ours. Although the method in [4] has the potential to achieve better performance, it may also need more computation in order to train more robust and more accurate model parameters for different sequences, while the complexity of our method is always the same with JSVM QL.

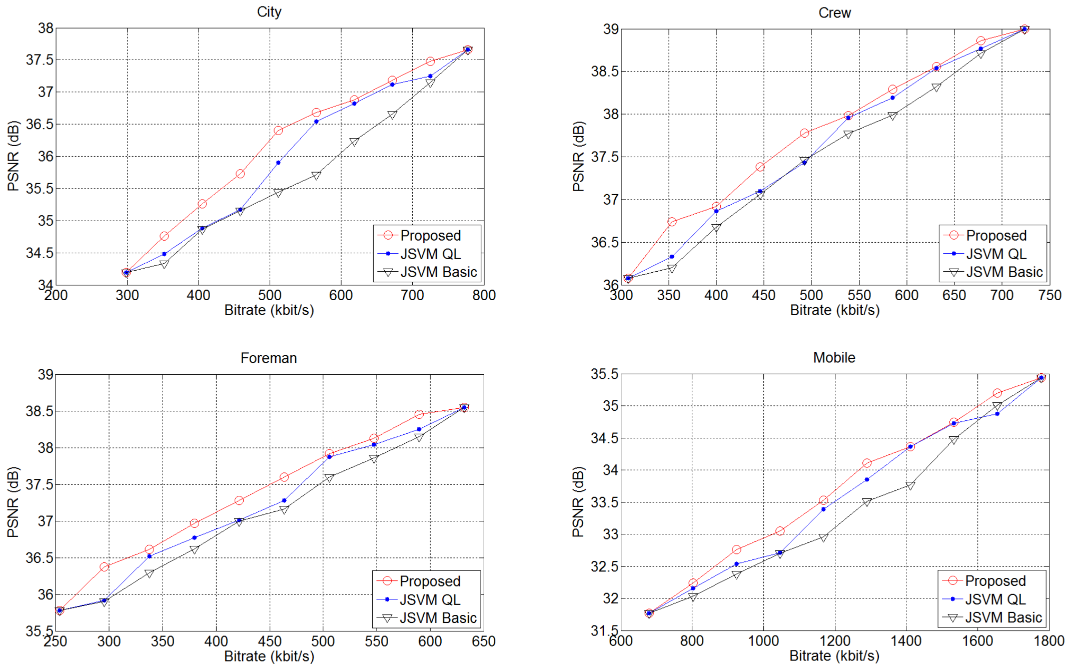

Fig. 1: Performance of three bitstream extraction methods for various CIF sequences

Table 1: Algorithm Performance: PSNR Gain Over JSVM(dB)

Sequence		*Bus*	*City*	*Crew*	*Football*	*Foreman*	*Harbour*	*Mobile*	*Soccer*
JSVM	Max[b]	0.36	0.56	0.41	0.31	0.45	0.30	0.34	0.30
QL[a]	Ave[c]	0.14	0.22	0.13	0.12	0.17	0.11	0.14	0.11
JSVM	Max	0.48	0.97	0.54	0.25	0.47	0.38	0.60	0.53
no QL	Ave	0.23	0.48	0.23	0.12	0.28	0.21	0.31	0.26

[a] Comparing the proposed algorithm with JSVM QL
[b] Maximum PSNR gain through all bitrate constraints
[c] Average PSNR gain through all bitrate constraints

Table 2: Comparing with Method in [4] and [5]: Average PSNR gain over JSVM QL (dB)

Sequence	*Bus*	*City*	*Crew*	*Football*	*Foreman*	*Harbour*	*Mobile*	*Soccer*
Proposed	0.14	0.22	0.13	0.12	0.17	0.11	0.14	0.11
Method in [4]	0.13	0.22	0.13	0.11	0.16	0.02	0.08	0.12
Method in [5]	0.06	0.07	0.01	0.10	0.05	0.04	-0.03	-0.01

5. CONCLUSION

We have presented an efficient method to address the no-reference bitstream extraction issue in the SVC applications. Using the linear error model as distortion estimation technique, a greedy algorithm is applied to simulate the discarding process of all packets, during which priority is assigned to every packet according to its R-D impact. The packets' priority values can be stored in the bitstream and used for R-D optimized extraction. Experimental results show that our proposed scheme can achieve PSNR gain of up to 0.56 compared with the JSVM QL, without computational complexity increment.

We will continue this paper's work in the following two directions. First, since the next generation video coding standard High Efficiency Video Coding (HEVC) [8] adopts the similar coding framework as its predecessor H.264/AVC, we expect to apply this paper's approach to the emerging HEVC standard and its scalable extension. Second, as mentioned in this paper, the current linear model is only applicable in the case of quality scalability, so further investigation and modification is to be done for the spatial or temporal scalability.

6. REFERENCES

[1] H. Schwarz, D. Marpe and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Trans. Circuits. Syst. Video Technol*, vol. 17, no. 9, pp. 1103–1120, 2007.

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

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

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

[5] W. Zhang, J. Sun, J. Liu and Z. Guo, “Optimized bit extraction of SVC exploiting linear error model,” *2012 IEEE International Symposium on Circuits and Systems*, pp. 1887–1890, Seoul, Korea, May 2012.

[6] M. Winken, H. Schwarz, T. Wiegand, “Joint Rate-distortion Optimization of Transform Coefficients for Spatial Scalable Video Coding Using SVC,” *15th IEEE International Conference on Image Processing*, pp. 1220–1223, San Diego, California, USA, Oct. 2008.

[7] P. E. Black, “Greedy Algorithm,” *Dictionary of Algorithms and Data Structures*[online], Vreda Pieterse and Paul E. Black, eds. 2 February 2005. Available from: http://www.nist.gov/dads/HTML/greedyalgo.html

[8] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1648–1667, 2012.