PROXIMATE CONTROL STREAM ASSISTED VIDEO TRANSCODING FOR HETEROGENEOUS CONTENT DELIVERY NETWORK

Xuebin Zhang[⋆]*, Yiran Li*[†]*, Jiangpeng Li*[⋆†]*, Kai Zhao*[⋆]*, and Tong Zhang*[⋆]

[⋆] ECSE Department, Rensselaer Polytechnic Institute, NY, USA
[†] Qualcomm, CA, USA
[⋆†] Shanghai Jiao Tong University, China

ABSTRACT

Video transcoding can be used to facilitate video streaming in content delivery network. A concept of control stream assisted transcoding has been recently proposed aiming to reduce transcoding computational complexity at the cost of data storage/transmission overhead, and the control stream is obtained by directly removing the residual information from the target video bitstream of transcoding. However, it is subject to relatively significant storage/transmission overhead, and more importantly does not well match to the increasingly heterogeneous networking environment with varying computation/storage/transmission resources at different nodes. This work presents a proximate control stream assisted transcoding design strategy that can reduce the control stream size and enable a large storage/transmission vs. computational complexity trade-off design space. Experiments demonstrate its effectiveness and noticeable advantages over other alternatives including simulcast and SVC (scalable video coding).

Index Terms— transcoding, control stream, H.264/AVC

1. INTRODUCTION

As one important technique being used to facilitate video delivery, transcoding [1, 2] aims to convert one video sequence to its another version with a lower spatial and/or temporal resolution or even different compression format. Transcoding often employs the first-decoding-then-encoding procedure to minimize the size (or bitrate) of the output target video sequence, particularly for mobile video data delivery. This however results in a high computational complexity due to the computation-intensive nature of video encoding.

An assisted transcoding design strategy was presented in [3] to reduce the video encoding computational complexity at the cost of data storage and/or transmission overhead. The key is to complement the source video sequence with a *control stream* that can facilitate the encoding process. In [3], the control stream is obtained by removing the residual information from the target video sequence, which is called *full control stream* in this paper. Given the full control stream, the encoding process in transcoding becomes much more computation-efficient (e.g., the computations for determining the modes, reference frames, and motion vectors are eliminated). Nevertheless, as modern video compression standards continue to improve the motion compensation efficiency and hence reduce the residual information volume, the size of full control streams tends to be relatively more significant, leading to noticeable storage and/or transmission overhead. For example, our studies on representative video sequences with H.264/AVC show that full control streams account for on average 38.7% of the entire target video sequences. Meanwhile, as content delivery network infrastructure becomes increasingly heterogeneous, especially with the emergence of heterogeneous wireless networking [4, 5], different nodes (e.g., edge proxy servers) may have different resources in terms of computation, storage, and communication bandwidth. Therefore, it is highly desirable for control stream assisted transcoding to reduce the control stream size and, more importantly, gracefully explore the data storage/transmission vs. encoding computation trade-off space.

To achieve this objective, this paper presents a *proximate control stream assisted transcoding* design strategy. Different from normal video streams that reach the end users and hence must strictly follow the video standard syntax, control streams are only created and consumed within the video storage and delivery infrastructure. Hence, control streams do not have to strictly follow the standard syntax. This flexibility can be leveraged to gracefully and effectively adjust the data storage/transmission vs. encoding computation trade-off. The proposed proximate control stream design strategy has two key aspects: (i) We can modify the definition or context of certain types of syntax elements in order to reduce control stream size at small increase of encoding computational complexity. In particular, we can relax the precision and completeness of syntax elements, e.g., for the syntax element of reference index, instead of conveying the exact index of the best reference, we can use 1-bit true-or-false flag to represent whether the best reference is the most likely one, and if it is false, the encoder has to carry out further computation to search for the best reference. In this paper, we studied the scenarios of relaxing the completeness of motion vector dif-

ference and reference index, referred to MVD-proximate and Ref-proximate control stream. (ii) We can modify the control stream bitstream processing procedure (e.g., not necessarily following the macroblock-by-macroblock order) to improve the compression ratio. In particular, before the entropy encoding process, control stream data of each slice are re-ordered so that the same type of syntax elements are grouped together. Intuitively, it will increase the correlation of the data stream and hence improve the entropy coding efficiency.

Using representative video sequences, we carried out studies to evaluate the proposed proximate control stream design strategy. Results show that MVD-proximate, Ref-proximate, and combined MVD/Ref-proximate control streams can reduce the size by 40.1%, 5.2%, and 44.8% compared with full control streams. This translates into noticeable savings on storage and/or transmission cost. Meanwhile, compared with stand-alone encoding, the use of full, MVD-proximate, Ref-proximate, and combined MVD/Ref-proximate control streams can reduce the computational complexity by 96.4%, 89.2%, 94.4%, and 88.4%, respectively. When further integrating with syntax element based reordering, MVD-proximate and combined MVD/Ref-proximate control streams can reduce the size by 45.1% and 48.5%, compared with original full control streams.

2. PROPOSED DESIGN SOLUTIONS

This section presents the proposed proximate control stream design strategy, which consists of two key components described below. We note that we use H.264/AVC as the baseline compression standard for all the discussions, and the same design strategy can be straightforwardly applied to other standards such as HEVC (high efficiency video coding).

2.1. Relaxing the Completeness of Syntax Elements

To reduce data storage/transmission overhead, the straightforward option is to remove one or few types of syntax elements from the control stream. Different types of syntax elements not only have different size but also have different importance from computational complexity reduction perspective. Among the major types of syntax elements in control streams, the skip flag and modes data are the most important and should not be removed or modified. Hence, we can only consider the removal of motion vector difference and reference index. The control streams, from which the motion vector difference and reference index are removed, are referred to as MVD-less and Ref-less control streams. Our study shows that motion vector difference is the most dominant syntax element, and MVD-less, Ref-less, and combined MVD/Ref-less control streams can reduce the size by 51.5%, 10.1%, and 59.4%, compared with full control streams. Nevertheless, such direct removal of syntax elements apparently results in significant increase of computational complexity.

Aiming to seek more graceful storage/transmission vs. encoding computation trade-offs, we propose to appropriately *relax the completeness* of the two types of syntax elements (i.e., motion vector difference and reference index) instead of simply removing them. First, we note that, in normal video compression, each motion vector difference and reference index element is represented with variable-length binary string to better match the unequal probability of all the possible values. Therefore, if we intentionally make each syntax element incomplete and only indicate whether the most likely scenarios occur or not, we can effectively reduce the control stream size and meanwhile largely maintain the reduction of encoding computational complexity. In particular, we propose to relax the completeness of motion vector difference and reference index as described below:

- *MVD-proximate control stream*: We replace each motion vector difference (consisting the difference on both X and Y dimensions) with a 1-bit motion vector flag. As illustrated in Fig. 1, we define a motion vector mini-search window with a very small size (e.g., 3×3 or 5×5) around the predicted motion vector. If the best-matching motion vector falls into the mini-search window around the searching center, we set the 1-bit motion vector flag as '1', otherwise we set it as '0'. During the control stream assisted transcoding, if the 1-bit motion vector flag is '1', the video encoder searches for the best-matching motion vector within the mini-search window around the searching center, otherwise the video encoder searches within the much larger normal search window.

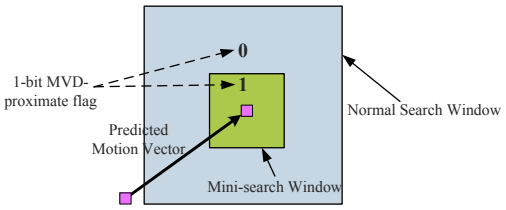

Fig. 1. Illustration of the use of mini-search window in the proposed MVD-proximate control stream.

- *Ref-proximate control stream*: We replace each reference index element with a 1-bit flag as well. If the best-matching reference frame is the most likely reference frame, which is typically set as the reference frame closest to the current frame, we set the flag as '1', otherwise we set it as '0'. During control stream assisted transcoding, if the 1-bit reference frame flag is '1', the reference index is immediately available, otherwise the encoder needs to exam all the possible reference frames to search for the best-matching reference frame.

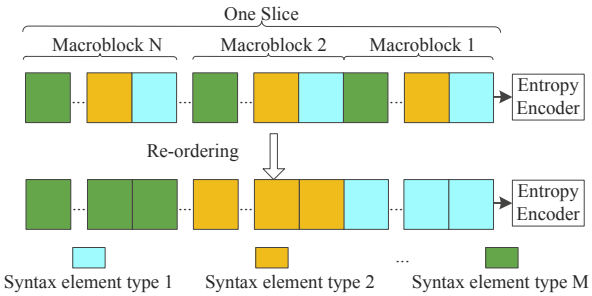

Fig. 2. Illustration of syntax element based re-ordering.

2.2. Syntax Element Based Re-ordering

The last step of video encoding is lossless compression of all the syntax elements using entropy coding, e.g., context-adaptive variable-length coding (CAVLC) and context-based adaptive binary arithmetic coding (CABAC). As illustrated in Fig. 2, the input bitstream to the entropy encoder is organized with the unit of macroblock (in H.264/AVC), e.g., all the syntax elements associated with the same macroblock are consecutive in the bitstream. Regardless to the specific entropy coding algorithm, the compression efficiency will increase as the correlation among adjacent data becomes stronger. As mentioned above, control streams are generated and consumed within the video storage and delivery infrastructure, hence we may not have to strictly follow the standard syntax. Intuitively, the same type of syntax element among adjacent macroblocks tends to have a stronger correlation. Therefore, we propose to re-order the input bitstream to the entropy encoder, as shown in Fig. 2, so that the same type of syntax element in each slice is grouped together. This can increase the bitstream data correlation and hence improve the entropy coding efficiency, leading to reduced control stream size.

3. EXPERIMENTAL RESULTS

To evaluate the effectiveness of the proposed design techniques, we carried out experiments using the H.264/AVC JM Reference Model version 18.2 with the following configurations: The search range is set as 32, the number of reference frames is 5, *rate-distortion optimization* is turned on, CABAC is chosen to perform entropy encoding, the integer-pixel motion search algorithm is UMHexagonS [6], and fractional pixel motion estimation is enabled. We studied the transcoding from 4CIF (704×576) to 480p (640×480) for four representative video sequences, including "Foreman" and "Soccer", which contain fast motion, and "News" and "Akiyo", which contain less and slow motion.

3.1. MVD-proximate and Ref-proximate Control Stream

We considered seven different scenarios, including MVD-less, MVD-proximate, Ref-less, Ref-proximate, MVD/Ref-less (i.e., both motion vector difference and reference index are removed from the full control stream), MVD/Ref-proximate (i.e., motion vector difference and reference index are replaced by 1-bit motion vector difference flag and 1-bit reference index flag, respectively), and the full control stream. The size of the mini-search window is set as 5×5 in the case of MVD-proximate. Fig. 3(a) shows the control stream size under these seven different scenarios, normalized against the complete target 480p video sequences. Fig. 3(b) shows the computation time as the measurement of computational complexity under these seven difference scenarios, normalized against stand-alone transcoding without any control streams.

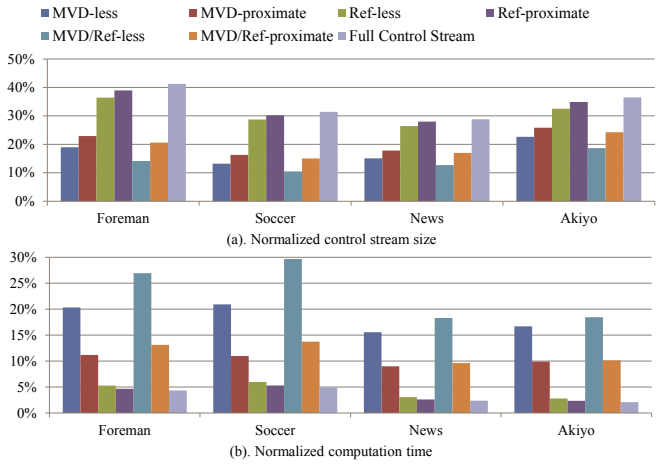

Fig. 3. (a) Control stream size normalized against to complete video sequences, and (b) computation time normalized to stand-alone transcoding.

The results in Fig. 3 show a large space of trade-offs between control stream size (hence storage/transmission overhead) and computational complexity. Although full control streams assisted transcoding can reduce the computation time by 94%, their size is on average 37% of complete video sequences. MVD-less control streams can largely reduce the size by about 50%, but the computation time increases by 15%. Compared with MVD-less scenario, the use of Ref-less control streams has less computation time but larger size. Compared with MVD-less control streams, MVD-proximate control streams can noticeably reduce the computation time (by 81% on average) and meanwhile modestly increase the size (by 4% on average). The difference between Ref-less and Ref-proximate is not very significant, mainly due to the small size of reference index in control streams.

3.2. Syntax Element Based Re-ordering

We further evaluated the effectiveness of the proposed syntax element based re-ordering that aims to reduce the control stream size. The key is to increase the adjacent data correlation in the bitstream to be processed by the entropy encoding.

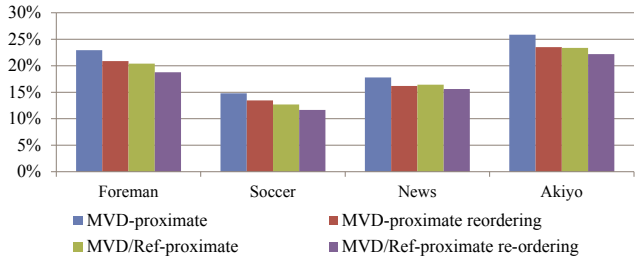

Fig. 4. Normalized control stream size after re-ordering.

Fig. 4 shows the size of MVD-proximate and MVD/Ref-proximate control streams with and without using syntax element based re-ordering, normalized against the size of complete video sequences. The results suggest that the re-ordering techniques can further reduce the control stream size by over 5%. Compared with the size of complete video sequences, the size of re-ordered MVD-proximate and the MVD/Ref-proximate control streams reduces to 19% and 17%, respectively.

3.3. Comparison with Simulcast and SVC

As pointed out in [3], simulcast and SVC are two alternative schemes for providing different representations of the same video. We carried out further studies to compare with these two alternatives in terms of rate-distortion (R-D) characteristics. Using the 'Foreman' sequence as the test vehicle, Fig. 5 shows the comparison among four different scenarios, including simulcast, SVC, full control stream assisted transcoding, and MVD/Ref-proximate control stream assisted transcoding. We note that the y-axis of the figure is the PSNR of the high-resolution sequence (i.e., 704×576 in this study), and the x-axis is the total bitrate of both high- and low-resolution sequences (i.e., both 704×576 and 640×480 sequences). To make a fair comparison among the four different scenarios, we adjust the coding parameters so that, for the high-resolution bitstreams that have the same PSNR under different design scenarios, the corresponding low-resolution bitstreams have the same PSNR as well. The results show that the MVD/Ref-proximate design strategy has the best R-D performance compared with the other three scenarios, and can save up to 17.5% of bitrate than the use of full control stream. Both types of control stream assisted transcoding can noticeably outperform the other two alternatives, e.g., at PSNR of 39.79 dB, the bitrate of MVD/Ref-proximate design is 28.0% lower than SVC and 34.2% lower than simulcast. Their advantages becomes more and more significant as we increase the PSNR (i.e., reduce QP), because the control stream occupies a less percentage of the complete video sequence under a lower QP. In addition, this study only considers two different resolutions, and the advantage of the proposed design strategy will become more significant as a larger number of different resolutions should be supported.

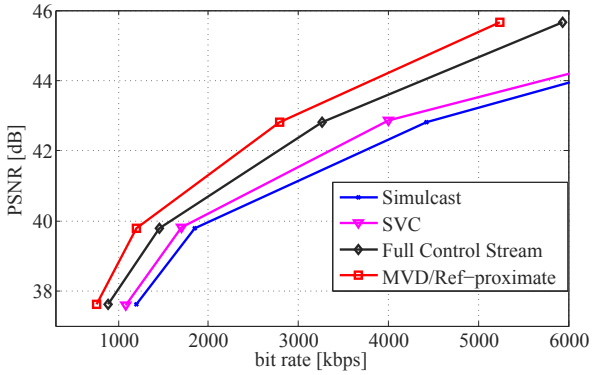

Fig. 5. R-D performance comparison among four different design scenarios.

4. CONCLUSIONS

This paper presents a proximate control stream assisted transcoding design strategy that enables a large and graceful data storage/transmission vs. computational complexity trade-off space for the increasingly heterogeneous networking environment. This large design space is essentially achieved by modifying the definition of certain types of syntax elements in bitstreams, and this paper presents two particular schemes for motion vector difference and reference index, leading to MVD-proximate and Ref-proximate control streams. A syntax element based re-ordering techniques is proposed to further reduce the control stream size. The effectiveness and advantages of this proposed design strategy has been well demonstrated through quantitative experiments with H.264/AVC and representative video sequences.

5. REFERENCES

[1] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing Magazine*, vol. 20, no. 2, pp. 18–29, 2003.

[2] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang, "Video transcoding: an overview of various techniques and research issues," *IEEE Transactions on Multimedia*, vol. 7, no. 5, pp. 793–804, 2005.

[3] G. Van Wallendael, J. De Cock, and R. Van De Walle, "Fast transcoding for video delivery by means of a control stream," in *IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 733–736.

[4] A. Ghosh et al., "Heterogeneous cellular networks: From theory to practice," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 54–64, 2012.

[5] Y. Li, Z. Pi, and L. Liu, "Distributed heterogeneous traffic delivery over heterogeneous wireless networks," in *IEEE International Conference on Communications (ICC)*, 2012, pp. 5332–5337.

[6] Z. Chen, J. Xu, Y. He, and J. Zheng, "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264–290, 2006.