A Low-Complexity Quantization-Domain H.264/SVC to H.264/AVC Transcoder with Medium-Grain Quality Scalability

Lei Sun[1], Zhenyu Liu[2], and Takeshi Ikenaga[1]

[1] Graduate School of Information, Production and Systems,
Waseda University, Kitakyushu, 808-0135 Japan
[2] Tsinghua National Laboratory for Information Science and Technology,
Tsinghua University, Beijing, 100084 China
sunlei@ruri.waseda.jp, liuzhenyu73@mail.tsinghua.edu.cn,
ikenaga@waseda.jp

Abstract. Scalable Video Coding (SVC) aiming to provide the ability to adapt to heterogeneous requirements. It offers great flexibility for bitstream adaptation in multi-point applications. However, transcoding between SVC and AVC is necessary due to the existence of legacy AVC-based systems. This paper proposes a fast SVC-to-AVC MGS (Medium-Grain quality Scalability) transcoder. A quantization-domain transcoding architecture is proposed for transcoding non-KEY pictures in MGS. KEY pictures are transcoded by drift-free architecture so that error propagation is constrained. Simulation results show that proposed transcoder achieves averagely 37 times speed-up compared with the re-encoding method with acceptable coding efficiency loss.

Keywords: low complexity, quantization domain, SVC/AVC transcoding, MGS.

1 Introduction

With the intention of providing scalability to diverse environments in multi-point applications, SVC enables transmission of a single bitstream containing multiple subset bitstreams. The subset bitstreams are organized in layered structure efficiently and can be extracted according to different requirements [1, 2]. Three main scalabilities are provided, i.e., spatial, temporal and quality. Performance of SVC and key technologies are described in literatures [3–5].

For multi-point applications involving terminals with different network bandwidth, display resolution and processing ability, SVC is expected to be a promising solution. Unlike the traditional transcoding-based systems, SVC-based system only requires lightweight operations for bitstream adaptation. However, as an undeniable fact, the current applications are mostly adopting legacy AVC-based systems. To achieve interoperability, transcoding between SVC format and AVC format is needed. In this paper, we focus on the ultra-low-delay transcoding from SVC to AVC with MGS Scalability (as will be explained in Section 2).

S. Li et al. (Eds.): MMM 2013, Part I, LNCS 7732, pp. 336–346, 2013.

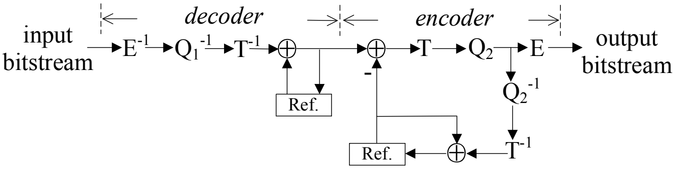


Fig. 1. Re-encoding transcoding architecture. (E: entropy coding, $Q_i (i = 1, 2)$: quantization, T: DCT transform, $Ref.$: reference picture buffer, superscript "-1": inverse process.)

SVC standard provides a special encoder-side configuration for SVC-to-AVC rewriting [6], which requires the encoding process modification at the sender side. It is actually a particular functionality during bitstream generation, rather than a real transcoder. When the sender side uses normal SVC configuration (i.e., without the rewriting functionality), this method does not work.

Transcoding has been an important task for bitstream adaptation or format conversion, due to the continuous progress in video coding standards [7–9]. A straightforward solution is the "re-encoding" method (Figure 1). It fully decodes the input bitstream and then fully re-encodes the decoded pictures, consuming intensive computations. For single layer transcoding, many works have been done. Literatures [10–17] are based on the motion reuse (MR) transcoding architecture. The modes and motion vectors (MVs) of input bitstream are utilized and refined to accelerate the motion estimation (ME) process of encoder. MR based works only accelerates the ME part of the re-encoding method, and the speed-up is restricted by existence of other components such as DCT transforms. In [18] the authors merge the decoder and encoder MCP (motion-compensated prediction) loops under the assumption that motion data are the same for decoder and encoder, referred as Single-Loop (SL) transcoding architecture (Figure 2). This architecture is free of drift (error propagation), and one inverse transform and one picture buffer are reduced. A further accelerated transcoding architecture is proposed by [19], in which transforms are totally removed and motion compensation is directly performed on DCT transform coefficients (denoted as MC-DCT). MC-DCT needs floating-point matrix multiplication which is quite costly and diminishes the speed gain. Another common known architecture is the open-loop (OL) transcoding architecture (Figure 3), for which the drift problem is quite severe. The SL and OL architectures are often referred as

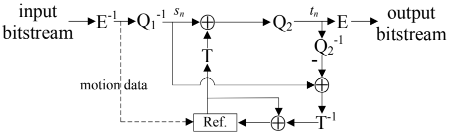


Fig. 2. Single-loop (SL) transcoding architecture

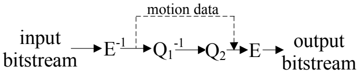


Fig. 3. Open-loop (OL) transcoding architecture

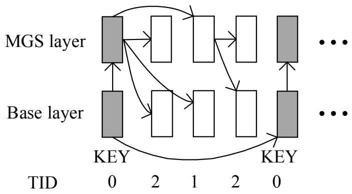


Fig. 4. Hierarchical-P with KEY pictures. (GOPSize = 4)

frequency-domain transcoding methods since there is no transform operations on the main route. Relatively, the re-encoding and MR architectures are usually mentioned as pixel-domain transcoding methods.

In this paper, a fast SVC-to-AVC MGS transcoding architecture is proposed. Significant speed-up is achieved by proposed quantization-domain transcoding for non-KEY pictures. KEY pictures are transcoded with drift-free SL architecture such that drift will be restricted between KEY pictures.

The rest of this paper is organized as follows. Section 2 describes SVC features. Section 3 shows proposed quantization-domain transcoding architecture for non-KEY pictures, and Section 4 explains KEY pictures transcoding. Simulation results are given in Section 5 and conclusions are drawn in Section 6.

2 Scalable Video Coding

2.1 Medium-Grain Quality Scalability

MGS includes two main features, i.e., coefficients partitioning and KEY picture concept. Coefficients partitioning allows to distribute the transform coefficients among several NALUs (Network Abstraction Layer Units). The coefficients are divided into groups, and each group corresponds to one NALU. Up to 16 NALUs are possible, and by discarding several of them flexible packet-based quality scalability is provided. In transcoding, it is very easy to parse the input NALUs and reassemble the transform coefficients.

Another feature is the KEY picture concept, which is based on hierarchical prediction structure. Figure 4 shows a hierarchical-P prediction structure for MGS encoding [20] (we target for low-delay applications where highly delayed B pictures are rarely used). TID represents the temporal layer ID. Grey-colored pictures are KEY pictures ($TID = 0$), which only use other KEY pictures for prediction. Base layer KEY picture is predicted from previous base layer KEY picture and MGS layer KEY picture is predicted from current base layer picture. Non-KEY picture ($TID > 0$) is predicted by the MGS layer of nearest previous picture with smaller TID. Such prediction structure can constrain the drift due to discarded packets within a GOP (Group Of Pictures).

2.2 Coding Modes in SVC

SVC introduces inter-layer prediction (ILP) schemes while inheriting the AVC coding modes (INTER and INTRA). Three kinds of ILPs are introduced to explore the correlation between base layer (BL) and enhancement layer (EL). They are inter-layer residual, inter-layer intra and inter-layer motion predictions (denoted as IL_Residual, IL_Intra and IL_Motion predictions hereafter).

IL_Intra prediction predicts the original enhancement layer input picture using the upsampled base layer reconstructed picture. IL_Residual prediction tries to predict the residual data generated by INTER prediction. The residual generated by normal INTER prediction is predicted by the upsampled base layer reconstructed residual signal. IL_Motion prediction tries to reduce the size of motion data for INTER coded MBs. The upsampled base layer mode and MV information is utilized to predict the enhancement layer motion data. More descriptions about ILPs can be found in [1].

The IL_Intra prediction is totally independent from the AVC INTRA or INTER modes, while the IL_Residual and IL_Motion predictions are additional refinements based on AVC INTER mode. Thus the coding modes in SVC are shown in Table 1. It is also possible that IL_Residual and IL_Motion both exist for an INTER MB. In such case, it is considered as IL_Residual. For short, "INTER with IL_Residual" and "INTER with IL_Motion" will be denoted as IL_Residual and IL_Motion hereafter.

Table 1. Coding modes in SVC

Inherited modes	Newly introduced modes
INTRA INTER without ILP	IL_Intra INTER with IL_Residual INTER with IL_Motion

3 Non-KEY Pictures Transcoding

3.1 Quantization-Domain Single-Loop Transcoding for IL_Residual MBs

In this subsection, a special frequency-domain transcoding architecture is derived for IL_Residual transcoding, namely quantization-domain single-loop (QDSL) transcoding. Let's start from the drift-free SL architecture as shown in Figure 2. Two signals s_n and t_n are shown and the relation between them is shown in (1).

$$t_n = Q_2(s_n + T(MC(T^{-1}(s_{n-1} - Q_2^{-1}(t_{n-1}))))) \qquad (1)$$

Here the $MC(.)$ represents the motion compensation operation corresponding to the bottom addition symbol in Figure 3. By assuming a distributive property for quantization (which is actually not true; same implication for following "assuming"), Equation (1) is modified to Equation (2).

$$t_n = Q_2(s_n) + Q_2(T(MC(T^{-1}(s_{n-1} - Q_2^{-1}(t_{n-1}))))) \qquad (2)$$

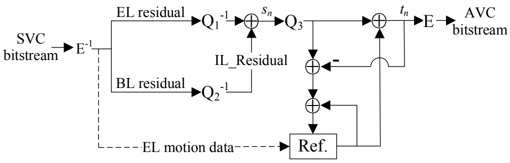


Fig. 5. Quantization-domain single-loop (QDSL) transcoding architecture

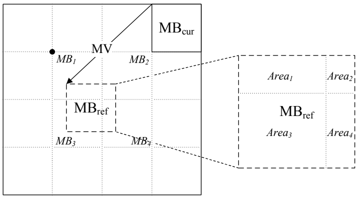


Fig. 6. MC in quantization domain

Then by assuming a commutative property between DCT transform and motion compensation, Equation (2) is further modified to Equation (3), also based on the fact that DCT transform is a lossless operation.

$$t_n = Q_2(s_n) + Q_2(MC(s_{n-1} - Q_2^{-1}(t_{n-1}))) \tag{3}$$

By assuming the commutative property between quantization and motion compensation, Equation (3) is changed to Equation (4).

$$t_n = Q_2(s_n) + MC(Q_2(s_{n-1} - Q_2^{-1}(t_{n-1}))) \tag{4}$$

Finally, by applying the previously assumed distributive property of quantization operation, Equation (5) is obtained.

$$t_n = Q_2(s_n) + MC(Q_2(s_{n-1}) - t_{n-1}) \tag{5}$$

The second term of Equation (5) implies a motion compensation on quantized transform coefficients. The first term is easily obtained by quantizing the input signal s_n. A corresponding architecture based on Equation (5) for IL_Residual is shown in Figure 5. Proposed QDSL architecture eliminates DCT transforms and further reduces one inverse quantization component comparing with reference [19].

Figure 6 illustrates the motion compensation method in quantization domain. MB_{cur} is the current MB to be coded and MB_{ref} is the reference MB. Dotted lines are aligned MB boundaries. If the motion vector points to an intersection of dotted lines, e.g. the intersection near the word "MB_1", then the prediction signal can be easily decided as the quantized transform coefficients of MB_1. When the MV does not point to an intersection, the prediction is composed by weighted sum of several related MBs. $MB_i(i = 1..4)$ are MBs overrode by MB_{ref}. The right sub-figure in Figure 10 enlarges the overrode area consisting of 4 regions. The areas of these regions are denoted by $Area_i(i = 1..4)$. Let $Coeff(MB_i)$ be the quantized coefficients matrix of MB_i. The prediction signal is generated by Equation (6). Partition or sub-partition motion compensation is done in a similar way.

$$PRED = \frac{\sum_{i=1}^{4}[Area_i \times Coeff(MB_i)]}{Area_1 + Area_2 + Area_3 + Area_4} \tag{6}$$

3.2 Quantization-Domain Intra Prediction for IL_Intra MBs

Similar to the derivation in previous subsection, Equation (7) can be obtained in the context of intra prediction. Here the $I\_PRED(.)$ is the intra prediction operation. The second term implies quantization-domain intra prediction (QDIP).

$$t_n = Q_2(s_n) + I\_PRED(Q_2(s_{n-1}) - t_{n-1}) \qquad (7)$$

In the pixel domain, neighboring pixels are used to form an intra prediction. But in quantization-domain where coefficients are concentrated in upper-left corner, extracting corresponding coefficients for those neighboring pixels is difficult. In proposed transcoder, QDIP is accomplished by approximating the prediction signal using neighboring 4x4 blocks. Figure 7 shows the intra 16x16 prediction in quantization domain. In the leftmost sub-figure, $B_i (i = 1..8)$ are the neighboring 4x4 blocks containing quantized coefficients. When intra 16x16 prediction mode is vertical or horizontal, the prediction is formed by extending neighboring blocks along the prediction direction. For other modes (DC/plane), the prediction is formed by averaging the vertical and horizontal predictions. $B_{ij} (i = 1..4, j = 5..8)$ is the average of B_i and B_j.

Intra 4x4 predictions are processed as Figure 8. B_{cur} is the current 4x4 block to be predicted and $B_i (i = 1..4)$ are neighboring blocks. $(X, A..L)$ are positions used for intra prediction in pixel domain. For mode 0, 1 or 8, the pixels used for intra 4x4 prediction belong to one particular neighboring block. This block is selected to approximate the prediction signal. For mode 2 which uses the mean value of $(A..B, I..L)$, the average of B_2 and B_4 is selected as the prediction. For the rest modes, a weighted average of neighboring blocks is formed as the prediction. The weight depends on how much one block contributes to the prediction signal. For example, Figure 9 shows the mode 3 prediction (each square corresponds to one pixel position), where there are totally 7 predictor values. Table 2 shows the corresponding positions using these predictors. Function $Con(.)$ represents the contribution of block B_i to each predictor. It equals the sum of weights of B_i pixels used for the predictor, multiplied by the number of blocks using this predictor. For example, in mode 3 the predictor $(C + 2D + E)/4$ uses C, D in B_2 and their weights are 1/4 & 2/4. Number of blocks using this predictor is 3 and thus the contribution of B_2 for this predictor is $(1/4 + 2/4) \times 3 = 9/4$. The total contributions for B_2 and B_3 are 25/4 & 39/4, and thus the prediction is

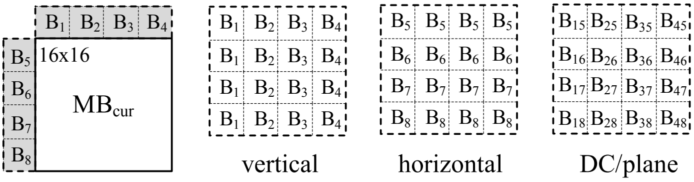


Fig. 7. Intra 16x16 prediction

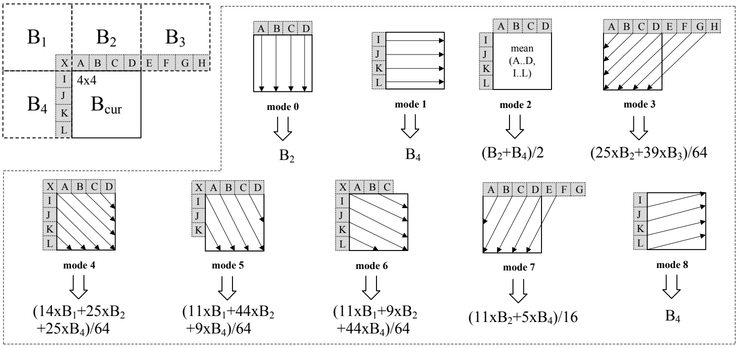


Fig. 8. Intra 4x4 prediction

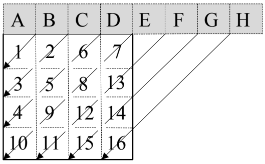


Fig. 9. Intra 4x4 mode 3 prediction

Table 2. Mode 3 prediction

Pixel no.	Predictor	Con(B_2)	Con(B_3)
1	(A+2B+C)/4	1	0
2,3	(B+2C+D)/4	2	0
4,5,6	(C+2D+E)/4	9/4	3/4
7,8,9,10	(D+2E+F)/4	1	3
11,12,13	(E+2F+G)/4	0	3
14,15	(F+2G+H)/4	0	2
16	(G+2H+H)/4	0	1

formed by $(25/4 \times B_2 + 39/4 \times B_3)/(25/4 + 39/4) = (25 \times B_2 + 39 \times B_3)/64$. The predictions for other modes are calculated similarly, and the final results are shown in Figure 13.

3.3 Quantization-Domain Copy for Other MBs

For MBs with other coding modes (INTRA/INTER without ILP/IL_Motion), a quantization-domain copy (QDC) method is applied. Different from IL_Residual or IL_Intra, for these modes the residual generation process in SVC is identical to AVC encoding. The input MB is entropy decoded only, and the quantized residual coefficients along with the motion data are copied into AVC bitstream directly. In case of IL_Motion, the motion data need to be reconstructed first, which is very easy. Note that no re-quantization is performed here, and thus the residual is kept accurate.

4 KEY Picture Transcoding

To constrain the propagation of errors caused by the false assumptions in QDSL deduction and MC/intra prediction approximations , KEY pictures are transcoded based on drift-free single-loop architecture. In hierarchical-P prediction structure, MGS layer KEY pictures are predicted by base layer KEY

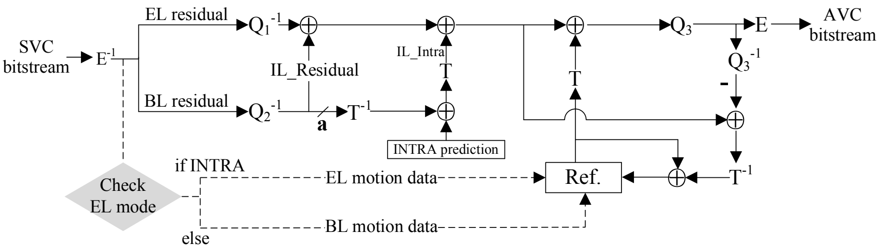


Fig. 10. KEY picture transcoding architecture

pictures. However, after transcoding the base layer pictures will be all discarded. Thus motion re-estimation is necessary for these KEY pictures since a new reference picture must be selected. In proposed transcoder, the previous MGS layer KEY picture is selected as the reference picture. Two merits can be obtained by such selection. Firstly, the prediction structure will remain as a single-layer hierarchical-P structure, by which the temporal scalability is kept. Secondly, due to the high correlation between MGS layer and base layer, the motion data from base layer prediction can be reused for MGS layer.

The above discussion solves INTER MBs in MGS layer (INTER, IL_Residual, IL_Motion). For IL_Intra MB, the base layer prediction mode of current frame is reused since there is no mode information transmitted in MGS layer. For INTRA MB, the MGS layer mode is directly reused. The proposed single-loop based BL-copy architecture is shown in Figure 10. The EL mode is checked to decide which motion data to be utilized. Note that EL does not need inverse transform or motion compensation. Partial decoding is also performed for BL decoding. Only MBs used for IL_Residual and IL_Intra predictions will be decoded. Besides, decoding of MBs used for IL_Residual prediction stops at position **a** since BL reconstruction signal is not needed.

5 Simulation Results

In this section, the proposed transcoder is applied to several publicly available sequences and the results are shown. 6 sequences are encoded with 3-layer MGS scalability, and then the encoded bitstreams are transcoded into AVC format with highest MGS layer quality. *Akiyo* and *bus* are CIF (352x288) sequences. *cheer_leaders* and *flower_garden* are VGA (640x480) sequences. *vidyo*1 and *parkrun* are 720p (1280x720) sequences. The main configuration parameters are shown in Table 3. All experiments are performed on an Intel Core 2 (2.67GHz) computer with 2.0GB RAM.

Besides the proposed method, 3 methods are used for comparison - re-encoding, single loop (SL) and open loop (OL). The implementation of re-encoding and OL methods are straightforward. The SL method is implemented based on reference [18]. Table 4 shows the computational time comparisons. Three criteria are shown, i.e., total transcoding time (C1), time saving (C2) and time per frame

Table 3. Experimental configurations

Parameters	SVC encoding	AVC encoding
Software Version	JSVM 9.18	JSVM 9.18
AVCMode	0	1
FramesToBeEncoded	150	150
SymbolMode	CABAC	CABAC
Enable8x8Transform	disabled	disabled
CodingStructure	Hierarchical-P	Hierarchical-P
NumRefFrames	5	5
SearchMode	4 (FastSearch)	4 (FastSearch)
SearchRange	16 for CIF/VGA, 32 for 720p	16 for CIF/VGA 32 for 720p
Quantization Parameter	28 for BL, 24 for EL	20/24/28/32
Loop Filter	enabled	enabled
DisableBSlices	1 (B-slice disabled)	1 (B-slice disabled)
GOPSize	4	4
MGSVectorX(X=0,1,2)	3,3,10	-
InterLayerPred	2 (adaptive)	-
AVCRewriteFlag	0 (disabled)	-

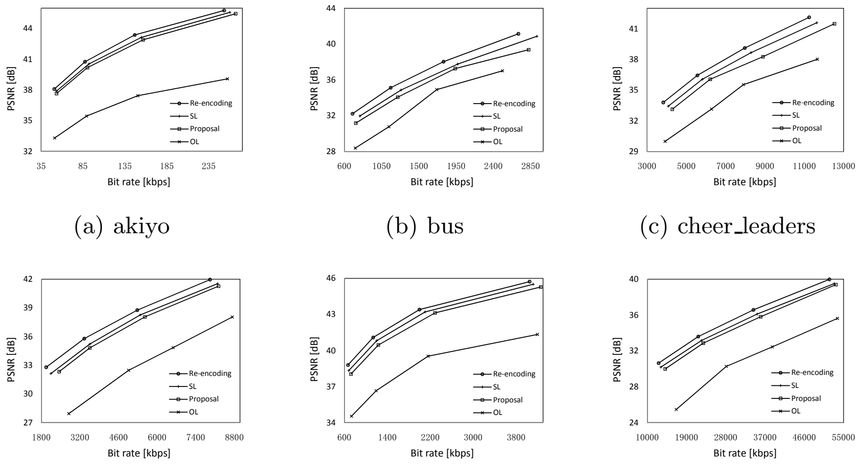
Table 4. Computational time comparisons

Sequence	Re-encoding	Single Loop			Open Loop			Proposal		
	C1	C1	C2	C3	C1	C2	C3	C1	C2	C3
akiyo	194.1	23.5	87.9	157	1.5	99.2	10	3.1	**98.4**	**21**
bus	230.3	26.2	88.6	175	1.9	99.1	13	6.9	**97.0**	**46**
cheer_leaders	738.0	97.6	86.8	651	6.9	99.1	46	25.2	**96.6**	**168**
flower_garden	668.8	80.3	88.0	535	5.7	99.1	38	20.3	**97.0**	**135**
vidyo1	1942.4	242.3	87.5	1615	14.2	99.3	94	3.9	**97.9**	**266**
parkrun	2209.3	281.2	87.3	1875	18.3	99.2	122	73.5	**96.7**	**490**
average	-	-	87.7	-	-	99.2	-	-	**97.3**	-

Criteria: C1: time(s), C2: time saving(%), C3: time/frame(ms)

(C3). The re-encoding method is selected as the comparison base, and the time saving for other methods is calculated by comparing with re-encoding method. The bolded figures in Table 4 show the time saving for our proposal relative to the re-encoding method, as well as the processing time per frame. Time saving ranges from 96.6% up to 98.4%, and the average time saving is 97.3% corresponding to a 37 times speed-up. Processing time per frame ranges from 490 ms down to 21 ms. Comparing with SL methods, proposed transcoder achieves averagely 4.6 times speed-up. OL method is about 3.4 times faster than proposed method.

To give intuitive coding efficiency comparisons, Figure 11 is provided which shows the R-D (rate-distortion) curves for tested sequences. It is obvious that for all sequences the re-encoding method performs best (topmost curve), following by SL and proposal curves sequentially with similar small gaps. OL method is much worse than the other 4 methods, mostly 3 to 4 dB lower.



(a) akiyo (b) bus (c) cheer_leaders

(d) flower_garden (e) vidyo1 (f) parkrun

Fig. 11. R-D curves comparison

6 Conclusions

This paper proposes a fast SVC-to-AVC MGS transcoder. Non-KEY frames of input MGS bitstream are transcoded using a quantization-domain transcoding architecture. KEY frames are transcoded using a drift-free transcoding method to constrain the error propagation. Simulation results show that proposed method gains 37 times speed-up comparing with the re-encoding method with acceptable coding efficiency loss.

Acknowledgement. This work was supported by KAKENHI (23300018) and KDDI foundation, Japan.

References

1. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the Scalable Video Coding Extension of the H.264/AVC Standard. IEEE Transactions on Circuits and Systems for Video Technology 17(9), 1103–1120 (2007)
2. Schwarz, H., Wien, M.: The Scalable Video Coding Extension of the H.264/AVC Standard [Standards in a Nutshell]. IEEE Signal Processing Magazine 25(2), 135–141 (2008)
3. Oelbaum, T., Schwarz, H., Wien, M., Wiegand, T.: Subjective Performance Evaluation of the SVC Extension of H.264/AVC. In: IEEE International Conference on Image Processing (ICIP), San Diego, pp. 2772–2775 (2008)
4. Jang, E.D., Kim, J.G., Thang, T.C., Kang, J.W.: Adaptation of Scalable Video Coding to packet loss and its performance analysis. In: International Conference on Advanced Communication Technology (ICACT), Phoenix Park, vol. 1, pp. 696–700 (2010)

5. Li, X., Amon, P., Hutter, A., Kaup, A.: Performance Analysis of Inter-Layer Prediction in Scalable Video Coding Extension of H.264/AVC. IEEE Transactions on Broadcasting 57(1), 66–74 (2011)
6. Segall, A., Zhao, J.: Bit-stream rewriting for SVC-to-AVC conversion. In: IEEE International Conference on Image Processing (ICIP), San Diego, pp. 2776–2779 (2008)
7. Vetro, A., Christopoulos, C., Sun, H.: Video Transcoding Architectures and Techniques: An Overview. IEEE Signal Processing Magazine 20(2), 18–29 (2003)
8. Ahmad, I., Wei, X., Sun, Y., Zhang, Y.Q.: Video Transcoding: An Overview of Various Techniques and Research Issues. IEEE Transactions on Multimedia 7(5), 793–804 (2005)
9. Xin, J., Lin, C., Sun, M.: Digital Video Transcoding. Proceedings of the IEEE 93(1), 84–97 (2005)
10. Sun, H., Kwok, W., Zdepski, J.W.: Architectures for MPEG Compressed Bitstream Scaling. IEEE Transactions on Circuits and Systems for Video Technology 6(2), 191–199 (1996)
11. Bjork, N., Christopoulos, C.: Transcoder Architectures for Video Coding. IEEE Transactions on Consumer Electronics 44(1), 88–98 (1998)
12. Shen, B., Sethi, I.K., Vasudev, B.: Adaptive Motion-Vector Resampling for Compressed Video Downscaling. IEEE Transactions on Circuits and Systems for Video Technology 9(6), 929–936 (1999)
13. Zhang, P., Liu, Y., Huang, Q., Gao, W.: Mode Mapping Method for H.264/AVC Spatial Downscaling Transcoding. In: IEEE International Conference on Image Processing (ICIP), Singapore, vol. 4, pp. 2781–2784 (2004)
14. Youn, J., Sun, M., Lin, C.: Motion Vector Refinement for High-Performance Transcoding. IEEE Transactions on Multimedia 1(1), 30–40 (1999)
15. Shanableh, T., Ghanbari, M.: Heterogenesous Video Transcoding to Lower Spatio-Temporal Rsolutions and Different Encoding Formats. IEEE Transactions on Multimedia 2(2), 101–110 (2000)
16. Escribano, G., Kalva, H., Cuenca, P., Barbosa, L., Garrido, A.: A Fast MB Mode Decision Algorithm for MPEG-2 to H.264 P-Frame Transcoding. IEEE Transactions on Circuits and Systems for Video Technology 18(2), 172–185 (2008)
17. Tang, Q., Nasiopoulos, P.: Efficient Motion Re-Estimation with Rate-Distortion Optimization for MPEG-2 to H.264/AVc Transcoding. IEEE Transactions on Circuits and Systems for Video Technology 20(2), 172–185 (2010)
18. Assuncao, P., Ghanbari, M.: Post-Processing of MPEG2 Coded Video for Transmission at Lower Bit Rates. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 4, pp. 1998–2001 (1996)
19. Assuncao, P., Ghanbari, M.: A Frequency-Domain Video Transcoder for Dynamic Bit-Rate Reduction of MPEG-2 Bit Streams. IEEE Transactions on Circuits and Systems for Video Technology 8(8), 953–967 (1998)
20. Hong, D., Horowitz, M., Eleftheriadis, A., Wiegand, T.: H.264 Hierarchical P Coding in the Context of Ultra-Low Delay, Low Complexity Applications. In: Picture Coding Symposium (PCS), Nagoya, pp. 146–149 (2010)