PAPER Special Section on Recent Advances in Multimedia Signal Processing Techniques and Applications

Low-Complexity Coarse-Level Mode-Mapping Based H.264/AVC to H.264/SVC Spatial Transcoding for Video Conferencing

Lei SUN[†a)], **Jie LENG**[†], **Jia SU**[†], **Yiqing HUANG**[††], **Hiroomi MOTOHASHI**[††], ***Nonmembers***, ***and*** **Takeshi IKENAGA**[†], ***Member***

SUMMARY Scalable Video Coding (SVC) was standardized as an extension of H.264/AVC with the intention to provide flexible adaptation to heterogeneous networks and different end-user requirements, which provides great scalability in multi-point applications such as video conferencing. However, due to the existence of H.264/AVC-based systems, transcoding between AVC and SVC becomes necessary. Most existing works focus on temporal transcoding, quality transcoding or SVC-to-AVC spatial transcoding while the straightforward re-encoding method requires high computational cost. This paper proposes a low-complexity AVC-to-SVC spatial transcoder based on coarse-level mode mapping for video conferencing scenes. First, to omit unnecessary motion estimations (ME) for layers with reduced resolution, an ME skipping scheme based on AVC mode distribution is proposed with an adaptive search range. Then a probability-profile based scheme is proposed for further mode skipping. After that 3 coarse-level mode-mapping methods are presented for fast mode decision and the adaptive usage of the 3 methods is discussed. Finally, motion vector (MV) refinement is introduced for further lower-layer time reduction. As for the top layer, direct encapsulation is proposed to preserve better quality and another scheme involving inter-layer predictions is also provided for bandwidth-crucial applications. Simulation results show that proposed transcoder achieves up to 92.6% time reduction without significant coding efficiency loss compared to re-encoding method.

key words: *AVC-SVC transcoding, spatial scalability, mode-mapping, low-complexity, video conferencing*

1. Introduction

The Scalable Video Coding extension of the H.264/AVC standard provides the ability to adapt to diverse client capabilities and requirements, which enables transmission of one bitstream containing multiple subset bitstreams [1], [2]. These subset streams are organized in layered structure and can be extracted adaptively according to user requirements. SVC provides 3 kinds of scalabilities: spatial (resolution) scalability, temporal (frame rate) scalability and quality (SNR, Signal-to-Noise Ratio) scalability. It is a good solution for video broadcasting and video conferencing which involve multiple terminals with different processing capabilities and network conditions. Performance evaluations of SVC and its key technologies can be found in [3]–[7].

Though SVC achieves good coding efficiency benefitting from the inter-layer predictions [6], it is impossible for every existing or under-developing system to support SVC codec. There are a lot of legacy systems or terminals which do not support SVC standard, and newly developed ones may choose another coding standard due to the system characteristic limitations. These systems or terminals are potential participants in a future video conferencing application. In order to communicate with such kind of user ends, transcoding is needed for SVC-based systems. Now let's assume a multiparty video conferencing scenario, as shown in Fig. 1. Part A is a new video conferencing system based on SVC standard, and part C is a personal PDA user with limited network bandwidth who supports only AVC codec for processing small size frames. Part B is a legacy multipoint control unit (MCU) [8] based system which also supports only H.264/AVC standard. Assume that a desktop PC in B sends a frame to a mobile in A, or a notebook PC in A sends a frame to a PDA in C, the receivers may be unable to decode or even receive. One straightforward solution is to transmit multiple AVC streams simultaneously. However, this requires the elements within the SVC system to handle not only SVC behaviours but also AVC multi-stream behaviours, and the benefits of SVC standard can not be utilized. Therefore, transcoding between AVC and SVC is needed in order to insure the homogeneous behaviour within SVC system and to utilize the SVC benefits. Note that B probably can not transcode between AVC and SVC since SVC is later standardized than AVC. Therefore, as a solution to provide backward compatibility, the gateway for system B should integrate the functionality of transcoding between AVC and SVC.

A simple and straightforward solution for transcoding is the cascaded re-encoding architecture [9], which fully decodes the input bitstream and then re-encodes. It usu-

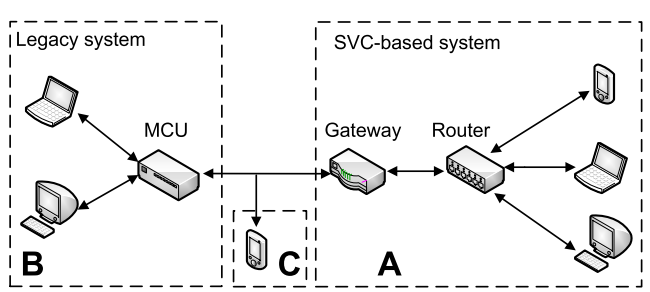


Fig. 1 A hybrid video conferencing scenario.

Manuscript received May 2, 2011.
Manuscript revised August 3, 2011.
[†]The authors are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyusyu-shi, 808–0135 Japan.
[††]The authors are with Ricoh R&D Center, Yokohama-shi, 224–0035 Japan.
a) E-mail: sunlei@ruri.waseda.jp
DOI: 10.1587/transinf.E95.D.1313

ally requires high computational cost. In earlier works on transcoding, the majority of interest focused on 2 directions: homogeneous transcoding (same coding standard for input & output bitstreams) [9] and heterogeneous transcoding (different coding standards for input & output bitstreams) [10], [11]. Though SVC has a layered structure, the AVC/SVC transcoding is more of a homogeneous transcoding due to SVC's AVC-compatible single layer encoding, except for the inter-layer predictions. Most conventional works focus on single layer transcoding for bit-rate reduction [12]–[17], spatial/temporal resolution reduction [17]–[21], CBR (constant bit-rate) and VBR (variable bit-rate) conversion, error-resilience transcoding and so on [9]. Newly developed works include AVC/SVC temporal transcoding [22], quality transcoding [23]–[26], and SVC-to-AVC spatial transcoding [27]. AVC-to-SVC spatial transcoding has not been fully investigated in existing literatures except for [28] which integrates some existing techniques and achieves about 2/3 time reduction. However, the PSNR (Peak Signal-to-Noise Ratio) drops about 1 dB at the same bit-rate compared with re-encoding method for many cases, probably due to the introduced inter-layer prediction, non-optimal mode decision and proportional MV scaling.

For reduced resolution transcoding, [18]–[21] proposed approaches in the DCT (Discrete Cosine Transform)-domain, which basically achieve larger time reduction compared with pixel-domain approaches. However, drift problem occurs and should be compensated, which needs additional calculation effort and decreases the overall gain. The resultant PSNR drops a lot, averagely about 0.5–0.9 dB for [18], 0.3–0.4 dB for [19], 0.7–1.6 dB for [20] and 0.5–1.5 dB for [21]. Since in video conferencing systems, the quality is usually expected to remain as much as possible, these DCT-domain approaches are not suitable. [17] and [29] are pixel-domain transcoding methods. In [17] the authors utilize 4-to-1 MV mapping with refinement which involves no sub-macroblocks (H.263), and the complexity reduction is claimed to be 23% averagely with approximately 0.7 dB PSNR loss. [29] presents a mode-mapping based downscaling transcoding method. Though refinement through an MV-based block merging scheme is possible, the PSNR still drops about 1–4 dB due to the underlying proportional mapping.

This paper investigates AVC-to-SVC spatial transcoding and proposes a coarse-level mode-mapping based low-complexity transcoding architecture for video conferencing. For SVC lower-layer encoding, an ME skipping scheme based on the mode distribution of input AVC stream is adopted for saving unnecessary ME calculations. The search range for ME skipping scheme is determined through an adaptive way. A following probability-profile based mode control method is applied for further mode skipping. Then, for non-skippable MBs, 3 coarse-level mode-mapping methods are presented with different tradeoff between coding efficiency and computational complexity, and the adaptive usage of these methods is also explained. Finally, MV refinement is introduced for further time reduction after mode decision. For SVC top-layer encoding, 2 schemes with different focus on quality or bit-rate are discussed.

This paper is organized as follows. Section 2 analyzes the reference model, i.e. re-encoding method. Overall architecture and algorithms are described in Sect. 3, and Sects. 4 & 5 explain the proposed methods in detail for lower layers and top layer respectively. Experimental results are given in Sect. 6, followed by conclusions in Sect. 7.

2. Reference Model Analysis

The cascaded pixel-domain re-encoding architecture for AVC-to-SVC spatial transcoding [28] is selected as the starting point of proposed transcoder and serves as a reference model. 3 procedures are involved: input AVC bitstream decoding, downsampling and SVC encoding (with adaptive inter-layer predictions). Table 1 shows the time cost distribution in re-encoding model for 8 test sequences which will be specified in Sect. 6. As expected, the most time-consuming part is the SVC encoding procedure, which involves motion estimations. AVC decoding and downsampling procedures are trivial in computation cost compared with SVC encoding. Complexity reduction in SVC encoding is necessary on the road towards low-delay transcoding for video conferencing. Reduction in top layer is simple since R-D (Rate-Distortion) optimized information from AVC bitstream is available. The follows paragraph discusses the possible solutions for time reduction in reduced-resolution transcoding.

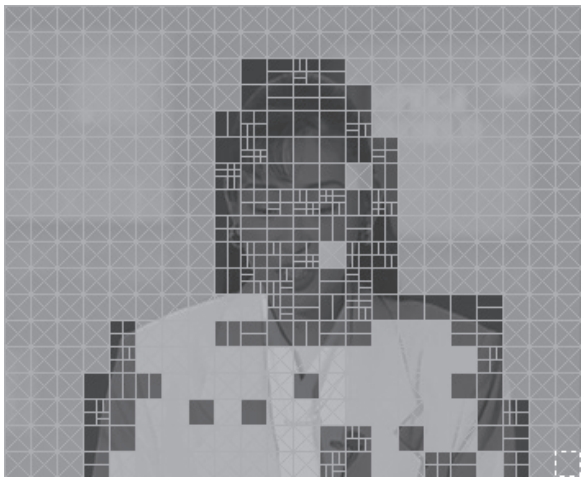
Though more general and statistical analysis is preferred, we only give some representative data to show the trend of motion data correlation between original frame and reduced-resolution frame. Table 2 shows the mode percentage difference for AVC frame and SVC downsized frame at a randomly selected frame in dyadic transcoding. Here the INTER refers to non-SKIP inter modes, and it holds

Table 1 Computational complexity distribution (QP = 20).

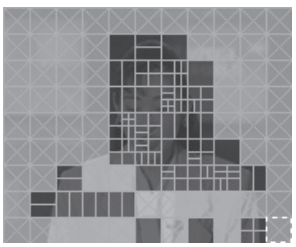
Sequence	AVC decoding	Downsampling	SVC encoding
akiyo	2.29%	1.89%	95.82%
panzoom2	2.45%	1.63%	95.92%
vidyo1	2.16%	1.71%	96.13%
vidyo3	2.17%	1.63%	96.20%
bus	2.42%	1.16%	96.42%
football	2.38%	1.19%	96.43%
flower_garden	2.49%	1.13%	96.38%
cheer_leaders	2.26%	0.92%	96.82%

Table 2 Mode percentage difference (frame 37, QP = 20).

Sequence	INTRA	SKIP	INTER
akiyo	+0.0%	+8.1%	−8.1%
panzoom2	−1.0%	+10.1%	−9.1%
vidyo1	−0.2%	−1.7%	+1.9%
vidyo3	+0.2%	−5.5%	+5.3%
bus	−2.3%	+3.8%	−1.5%
football	−1.5%	+2.8%	−1.3%
flower_garden	−0.6%	+5.2%	−4.6%
cheer_leaders	−0.3%	−4.8%	+5.1%



(a) cif@AVC



(b) qcif@SVC

Fig. 2 Intuitive mode & partition comparison for akiyo sequence, frame 37. (light grey: SKIP, dark grey: INTER (non-SKIP))

for the rest of this paper. It can be inferred from the table that AVC-coded frame and corresponding downsized SVC-coded frame have similar mode distribution. Therefore, mode information reuse is possible. Tables 3 and 4 show the average number of MBs within co-located region (4 MBs in total) in input AVC frame which have the same mode or partition as SVC low-resolution MB. We can see that the mode tends to be similar for co-located positions while the partition tends to be very different. Based on this observation, the mode skipping schemes in Sects. 4.1 and 4.2 are proposed. Besides, although the mode distributions are similar, the MB partitions are usually loosely coupled as shown in Fig. 2 which shows an example for mode and partition comparison. Table 5 shows the percentage difference for INTER partitions which contains large fluctuations for many cases. Therefore, conventional methods based on proportional partition mapping is not suitable. To address this problem, the mode mapping and MV refinement schemes are proposed in Sect. 4.3 and 4.4.

3. Overall Transcoding Architecture

Figure 3 shows the proposed transcoder. Although the proposed methods in following sections are applicable to multiple layers, here for simplicity and clarity we only show the 2-layer structure. Both motion compensation (MC) and downsampling are processed in pixel domain. The marks E, Q, T, E_1, E_2 stand for entropy coding, quantization, transformation, top-layer entropy coding and lower-layer entropy coding respectively.

Table 3 Mode correlation for co-located MBs (frame 37, QP = 20).

I	Sequence	INTRA	SKIP	INTER
II	akiyo	-	3.6	3.1
	panzoom2	-	1.2	3.5
	vidyo1	-	3.2	2.8
	vidyo3	0.8	3.2	3.2
	bus	2.0	1.0	3.7
	football	3.4	0.8	3.1
	flower_garden	-	3.0	3.8
	cheer_leaders	2.9	1.1	3.4

I: The mode of SVC low-resolution MB
II: The number of same-mode MBs within 4 co-located MBs
("-" means no such a mode in low-resolution frame)

Table 4 Partition correlation for co-located MBs (frame 37, QP = 20).

I	Sequence	16×16	16×8	8×16	8×8
II	akiyo	1.3	0.3	0.3	0.3
	panzoom2	1.6	0.5	0.6	0.0
	vidyo1	1.2	0.7	0.5	0.4
	vidyo3	1.5	0.6	0.6	0.7
	bus	1.8	0.6	0.6	0.0
	football	1.8	0.5	0.9	0.0
	flower_garden	1.2	0.8	0.4	1.4
	cheer_leaders	0.9	0.5	0.7	0.9

I, II: same definition as Table 3

Table 5 INTER partition percentage difference (frame 37, QP = 20).

Sequence	16×16	16×8	8×16	8×8
akiyo	−7.7%	−10.9%	+10.2%	+8.4%
panzoom2	−3.8%	−7.6%	−7.8%	+19.2%
vidyo1	−8.8%	+4.0%	−1.6%	+6.4%
vidyo3	−15.1%	−1.7%	−1.4%	+18.2%
bus	+0.1%	−0.1%	+4.5%	−4.5%
football	−12.9%	−14.8%	−11.4%	+39.1%
flower_garden	−5.3%	−2.2%	−5.6%	+13.1%
cheer_leaders	−8.8%	−7.7%	−4.2%	+20.8%

For the lower-layer ME, 4 proposed schemes are applied - 2 mode skipping schemes, 1 mode mapping scheme (incl. 3 sub-schemes) and 1 MV mapping scheme. First, the ME skipping scheme is applied with the intention to skip unlikely mode types which is described in Sect. 4.1. The following profile-based mode control method (Sect. 4.2) is utilized if more than 2 candidate mode types remain after ME skipping. Then, the coarse-level mode-mapping method is applied for INTER MBs which are not skipped by previous steps. Section 4.3 presents 3 such kind of methods and explains the adaptive usage of the 3 methods. And at last, MV refinement is applied for further time reduction.

A is a switch which changes the top-layer strategy according to the network condition. If the bandwidth is enough, the upper routine described in Sect. 5.1 with well-preserved top-layer quality will be selected. Otherwise, the lower routine described in Sect. 5.2 will be adopted for lower bit-rate with degraded top-layer quality.

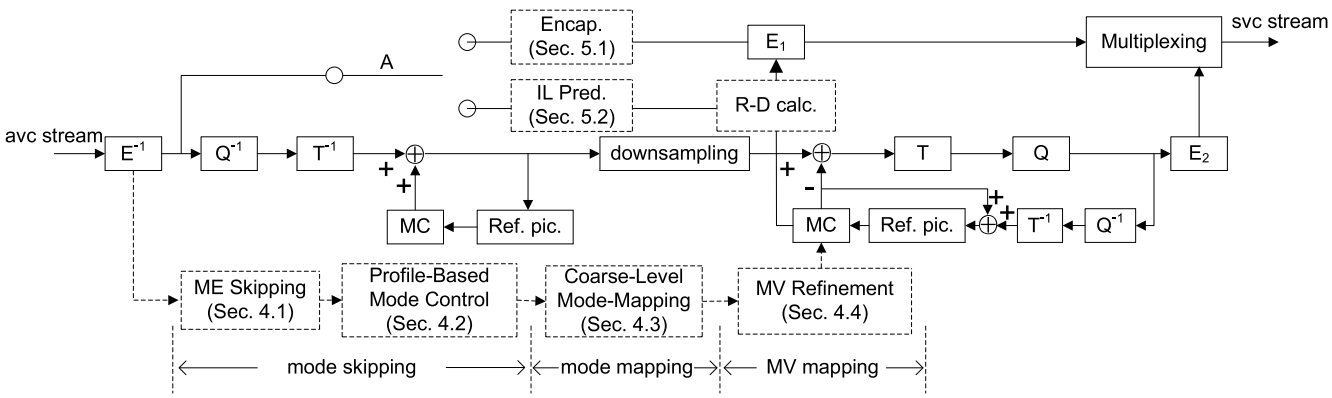


Fig. 3 Proposed transcoder. (Encap.: Encapsulation, IL Pred.: Inter-Layer Prediction)

4. Proposed Lower-Layer Transcoding Schemes

4.1 ME Skipping Scheme

Although downsampling methods cannot completely construct the relation between high-resolution and low-resolution frames, we notice that there is a rough rule between them, that is the similar mode distribution. If the lower-layer MB is mode X (INTER, INTRA or SKIP; no concern about sub-partitions), then the input AVC frame co-located region (consisting of β^2 MBs and β is the scaling factor) is probably also mode X. This is usually the case except for some irregular MBs caused by downsampling losses. Based on this rule, an ME skipping scheme is proposed.

First an adaptive search range in high-resolution frame is defined. The lower bound for the search range is $\beta \times \beta$ (co-located MBs). Assume that the top-layer resolution is $M \times N$, then the upper bound is set to $U \times U$ according to Eqs. (1) & (2).

$$SR\_Width = round\left(\sqrt{\frac{M}{16} \times \frac{N}{16} \times \alpha}\right) \quad (1)$$

$$U = \beta \times round\left(\frac{SR\_Width}{\beta}\right) \quad (2)$$

The round(.) operator calculates the nearest integer for the parameter. α is the percentage of MBs in search range over entire frame. Equation (1) calculates the search range width measured in terms of MBs, and Eq. (2) maps the value to multiples of scaling factor in order to make the search range symmetrical to co-located region. Too large α decreases overall time reduction and too small value will lead to non-statistical result. Through vast experiments over different sequences, we found that generally 0.04 gave good performance. In dyadic transcoding ($\beta = 2$), the upper bound is 4×4 for CIF size, 6×6 for VGA size and 12×12 for 720p size when α is set to 0.04. Figure 4 shows an example for VGA sequence. The grey region shows the co-located MBs and the numbers mean the search order (from 1 to 36 with increasing distance to center - decreasing relevance to lower-layer MB).

26	25	24	23	22	21
27	10	9	8	7	20
28	11	1	2	6	19
29	12	3	4	5	18
30	13	14	15	16	17
31	32	33	34	35	36

Fig. 4 Search range and search order in VGA sequence.

The search range is adapted MB by MB between lower bound and upper bound according to the homogeneity of previous MB. If the search range of previous MB contains only one type of mode, it is considered smoothed and the current MB will decrease the search range by $[-2, -2]$ (e.g. 6×6 -> 4×4). Otherwise, it is considered detailed and the search range will be enlarged by $[+2, +2]$. Of course, the search range will not across the boundaries.

Then check the modes of these MBs in the search range in predefined order. If SKIP, INTER or INTRA exists, estimate this mode respectively. On the contrary, if some mode does not exist in the search range, then skip the estimation for this mode. This scheme works efficiently when some mode is concentrated in limited areas, which means the other modes may be skipped by proposed scheme.

4.2 Probability-Profile Based Mode Decision Control

If more than 2 modes out of SKIP, INTER and INTRA are not skipped by the method in Sect. 4.1, a scheme for further mode selection is adopted by maintaining a profile of mode percentages for high-resolution frame. This method tries to "catch up with" the percentage profile of high-resolution frame.

Assume that we are processing an MB in lower layer. Let N'_X be the number of mode X in high-resolution frame up to current co-located MBs, N''_X be the lower-layer number of mode X over all previous MBs. ΔN is the maximum al-

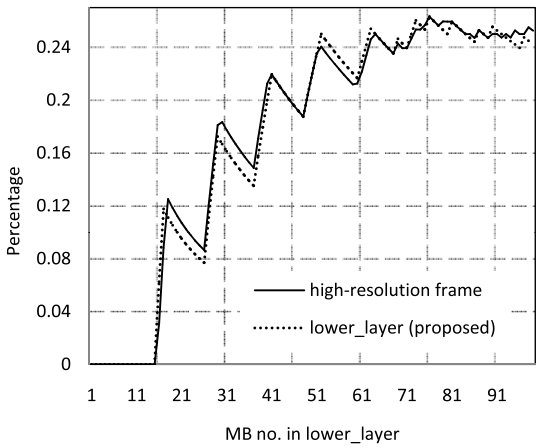


Fig. 5 INTER (non-SKIP) mode profile for akiyo sequence, frame 37.

lowed difference between high-resolution frame and scaled lower layer in terms of MBs and $\Delta\hat{N}$ is the actual difference as shown in Eq. (3). β is the scaling factor and M_X calculated by Eq. (4) is a signed measurement for the lower-layer deviation from top layer, ranged from -1 to 1. P''_X denotes the probability of mode X for current MB in lower layer and Eq. (5) maps its range to $[0, 1]$. ΔN is typically set to 20 in our dyadic experiments which means the maximum allowed deviation for lower-layer is 5 MBs ($20/2^2$).

$$\Delta\hat{N} = N'_X - \beta^2 \times N''_X \tag{3}$$

$$M_X = \begin{cases} -1 & , \Delta\hat{N} \le -\Delta N \\ 1 & , \Delta\hat{N} \ge \Delta N \\ \Delta\hat{N}/\Delta N & , \text{otherwise} \end{cases} \tag{4}$$

$$P''_X = \frac{M_X + 1}{2} \tag{5}$$

For each existing mode, a lottery function based on the estimated probability P''_X is applied to judge whether to skip that mode. If all the modes are skipped, the mode with largest probability will be selected.

Figure 5 shows an example for INTER mode. The horizontal axis denotes the lower-layer MB number and the vertical axis is the percentage of INTER-coded MBs. The straight line shows the profile for high-resolution frame and the dotted line shows the profile for lower layer by proposed method. It can be seen that the mode distribution is well mirrored.

4.3 Coarse-Level Mode-Mapping Methods

Although there is a similar mode distribution rule between high-resolution and low-resolution frames, this is not the case for partitions and MVs of INTER MBs. They have rarely proportional relations. Therefore, lower layer should not be mapped by scaling partition and MV directly [29]. Instead, we find another coarse-level rule that if lower-layer MB has few details, AVC co-located MBs usually also have few details. On the contrary, if lower-layer MB has many details, AVC co-located MBs probably also have many details (but not have to be proportional). Based on this rule, 3 mode mapping methods for INTER estimation with different tradeoff between coding efficiency and complexity are explained in the following paragraphs, and an adaptive usage is proposed to achieve an optimal combination of the 3 methods.

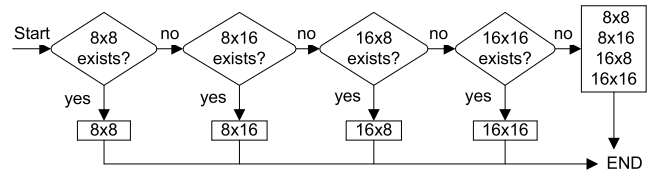


Fig. 6 Direct mapping method.

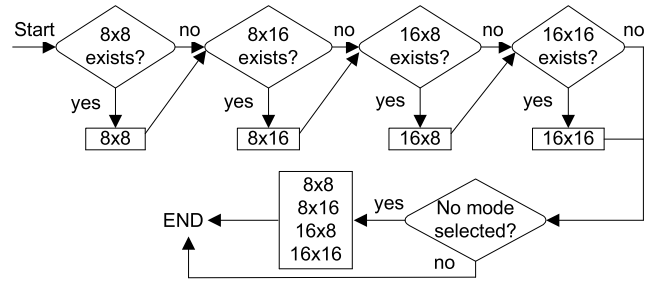


Fig. 7 Candidate mapping method.

The first method is the direct-mapping method, which is a 4-to-1 mapping as shown in Fig. 6. This method first checks the co-located MBs to see if 8×8 mode (no concern about sub-partitions) exists. If it exists, stop the procedure and estimate 8×8 (incl. sub-partitions) only. Otherwise, continue to check 8×16, 16×8 and 16×16 similarly. If no mode is selected at last, all INTER modes will be estimated.

Candidate-mapping method is the second approach which performs ME for candidate modes only. This method selects co-located MBs' modes as candidates for lower-layer encoding, as shown in Fig. 7. It checks 8×8, 8×16, 16×8 and 16×16 sequentially in co-located MBs. If some mode exists, it will be added to estimation list. Otherwise, it will not be added. When the procedure finishes, only the modes in estimation list will be estimated. If no mode exists in the estimation list at last, all INTER modes will be estimated.

Another method is the priority-mapping method, which performs ME based on priority. The priory is defined as: $8 \times 8 > \{8 \times 16,\ 16 \times 8\} > 16 \times 16$. This is because the complexity and uncertainty of detailed MB is larger than smooth one. This method checks from 8×8 to 16×16 as shown in Fig. 8. If some mode exists, estimate all modes with larger (or equal) priority. Similarly, if no INTER mode exists at last, all modes will be estimated.

These methods reuse the AVC stream information at a coarser level which involves no sub-partitions due to the irregularity of sub-partitions. Table 6 in Sect. 6 shows that the direct mapping method has the largest complexity reduction while priority mapping method achieves the best coding efficiency, and candidate mapping method performs moderately.

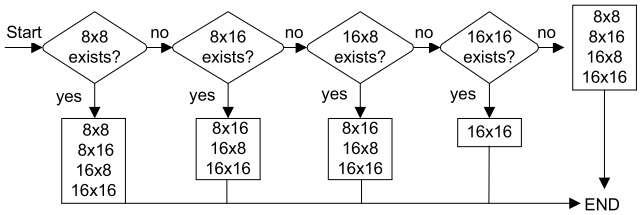


Fig. 8 Priority mapping method.

In proposed transcoder, they are combined adaptively according to the homogeneity of current search range. 3 levels are defined: level 1 with less than 1/3 SKIP or INTER_16×16 modes in current search range, level 2 with less than 2/3 but more than 1/3 SKIP or INTER_16×16 modes, level 3 with more than 2/3 SKIP or INTER_16×16 modes. Level 1 is the most detailed, so the most accurate priority-mapping method is adopted. Level 2 is moderately detailed and candidate-mapping method is selected. Direct-mapping method is used in level 3 which is the least detailed.

4.4 MV Refinement Scheme

Some conventional works focused on MV refinement based on nearly whole-frame MV mapping [18], [30], which turned out to be inaccurate and caused efficiency loss. However, MV refinement is expected to be more efficient in homogeneous area compared with detailed area since less MVs are involved. Detailed area which leads to more MVs will increase the complexity and uncertainty for MV mapping. In proposed transcoder, MV refinement is only applied for MBs whose co-located MBs are all SKIP or INTER_16×16. Before applying the MV refinement another check is executed - the MV diversity of co-located MBs. Equation (6) calculates the arithmetic average MV among co-located MBs. β is the scaling factor and MV_i_x & MV_i_y represent the horizontal and vertical components for i_{th} MB respectively. Equation (7) calculates the diversities of horizontal and vertical MV components by summing the absolute difference (SAD) between the MVs of co-located MBs and the average MV.

$$\begin{cases} \overline{MV\_x} = \frac{1}{\beta^2} \sum_{i=0}^{\beta^2-1} MV_i\_x \\ \overline{MV\_y} = \frac{1}{\beta^2} \sum_{i=0}^{\beta^2-1} MV_i\_y \end{cases} \tag{6}$$

$$\begin{cases} SAD_x = \sum_{i=0}^{\beta^2-1} |MV_i\_x - \overline{MV\_x}| \\ SAD_y = \sum_{i=0}^{\beta^2-1} |MV_i\_y - \overline{MV\_y}| \end{cases} \tag{7}$$

Then the SAD values are compared with pre-defined thresholds, as shown in Eq. (8). If Eq. (8) holds, the MV refinement is applicable. Otherwise, it will not be performed. Smaller thresholds will constrict the applicable rate while larger thresholds will result in worse coding efficiency. In our experiments, the thresholds Th_x and Th_y are both set to 4 since small thresholds give no harm anyhow.

$$\begin{cases} SAD_x \le Th_x \\ SAD_y \le Th_y \end{cases} \tag{8}$$

$$\begin{cases} MV\_scaled\_x = \frac{1}{\beta}\overline{MV\_x} \\ MV\_scaled\_y = \frac{1}{\beta}\overline{MV\_y} \end{cases} \tag{9}$$

If the MV refinement is feasible, INTER_16×16 is chosen as the lower-layer MB mode. The scaled average MV calculated by Eq. (9) will be used for lower-layer ME, and it is used as the starting point for motion search with a much smaller search range compared to original search window. In our experiments, the refinement search range is selected as [−2, +2] for both horizontal and vertical components.

5. Proposed Top-Layer Transcoding Schemes

5.1 Direct Encapsulation

One approach for top-layer transcoding is to transmit the top-layer bit-rates directly without full decoding and re-encoding. The top-layer AVC bitstream is first VLC (Variable Length Coding) decoded and then VLC re-coded using SVC encoder. There is no quality loss since no quantization is needed. After the VLC re-coding, the top-layer bitstream is multiplexed with lower-layer bitstream which is generated using proposed lower-layer schemes. This behaviour is actually very similar to simulcast transmission except that the final bitstream formats are different. In simulcast, the bitstreams are transmitted as 2 AVC streams while in direct encapsulation method the bitstreams are multiplexed into SVC format. [6] points out that the SVC coding tools are less efficient for spatial scalability especially for simple and slow-motion scenes, which is often the case in video conferencing applications. Therefore, direct encapsulation is recommended for video conferencing if the bandwidth is sufficient.

5.2 Inter-Layer Prediction Utilization

As another choice, the inter-layer predictions can be utilized when the bandwidth is crucial. In direct encapsulation method, R-D costs for different modes which the inter-layer predictions need can not be obtained since there is no ME performed in top layer. As shown in Fig. 3, we only re-calculate the R-D cost according to the mode and MV got from the AVC bitstream, which have been R-D optimized already. It should be noticed that the source sequence can not be obtained on the transcoder side, so we use the decoded sequence instead as the input for R-D cost calculation in SVC encoder, just like the re-encoding method. The inter-layer motion and intra predictions act the same as original SVC encoder while residual prediction is only performed for the corresponding mode and MV got from AVC frame.

Although the transcoder processing speed decreases a little, the overall complexity is still kept very low since ME is not performed. The drawback for this scheme is the degraded quality due to a second-time quantization loss. It is recommended for bandwidth-crucial applications.

Table 6 Performance comparison of proposed transcoder (lower-layer).

Sequence	direct			candidate			priority			adaptive		
	C1	C2	C3	C1	C2	C3	C1	C2	C3	C1	C2	C3
akiyo	+6.45	−0.34	−71.4	+5.21	−0.28	−70.2	+2.38	−0.13	−68.3	+3.50	−0.17	−70.0
panzoom2	+8.54	−0.42	−66.9	+6.82	−0.33	−63.7	+3.34	−0.16	−62.0	+3.40	−0.16	−63.7
vidyo1	+11.50	−0.46	−73.6	+7.13	−0.29	−72.2	+4.46	−0.19	−70.6	+5.14	−0.21	−72.1
vidyo3	+8.51	−0.39	−73.6	+4.45	−0.21	−72.0	+2.92	−0.14	−70.3	+3.28	−0.17	−72.7
bus	+6.75	−0.43	−44.5	+3.28	−0.22	−37.2	+2.46	−0.16	−32.8	+2.71	−0.18	−34.7
football	+5.72	−0.34	−37.4	+4.56	−0.27	−33.4	+3.69	−0.22	−29.1	+3.89	−0.23	−32.4
flower_garden	+5.70	−0.36	−56.3	+4.05	−0.26	−50.7	+3.55	−0.24	−46.9	+3.75	−0.25	−49.4
cheer_leaders	+4.35	−0.35	−35.1	+3.42	−0.27	−29.5	+2.58	−0.20	−22.1	+2.85	−0.22	−24.0

Criterions: C1: BDBR (%), C2: BDPSNR (dB), C3: $\Delta time$ (%)

Table 7 Performance comparison of proposed transcoder (top-layer).

Sequence	Direct Encapsulation			Inter-Layer Prediction		
	Δ bit-rate (%)	Δ Y-PSNR (dB)	Δ time (%)	Δ bit-rate (%)	Δ Y-PSNR (dB)	Δ time(%)
akiyo	+7.49	+1.510	−96.4	+1.11	−0.078	−85.9
panzoom2	+11.60	+1.488	−97.9	+0.98	−0.063	−84.0
vidyo1	+8.97	+1.364	−97.4	+0.64	−0.160	−86.9
vidyo3	+5.41	+1.419	−97.6	+1.52	−0.133	−87.1
bus	+7.94	+1.577	−96.5	+3.60	−0.087	−77.1
football	+4.10	+1.555	−96.6	+0.65	−0.115	−81.5
flower_garden	+8.05	+1.653	−94.6	+1.13	−0.085	−77.8
cheer_leaders	+4.34	+1.477	−97.5	+0.38	−0.170	−80.3

Table 8 Overall performance of proposed transcoder (top-layer + lower-layer).

Sequence	Direct Encapsulation		Inter-Layer Prediction	
	Δ bit-rate (%)	Δ time (%)	Δ bit-rate (%)	Δ time (%)
akiyo	+6.69	−91.1	+1.59	−82.7
panzoom2	+9.96	−91.1	+1.46	−79.9
vidyo1	+8.20	−92.3	+1.54	−84.0
vidyo3	+4.98	−92.6	+1.87	−84.1
bus	+6.89	−90.3	+3.42	−72.9
football	+4.06	−90.2	+1.30	−76.6
flower_garden	+7.19	−90.1	+1.65	−75.0
cheer_leaders	+4.04	−90.2	+0.87	−74.7

6. Experimental Results

In this section, proposed methods are applied to some sequences and the results are shown. All experiments are performed on an Intel Core 2 (2.67 GHz) computer with 2.0 GB RAM and software implementation is based on JM (Joint Model) 17.2 and JSVM (Joint Scalable Video Model) 9.18. JSVM's AVC compatible decoder and down-converter are used for AVC decoding and downsampling processes respectively. 8 sequences are examined with 2-layer dyadic spatial scalability. Akiyo, panzoom2, football and bus are CIF to QCIF transcoding at 30 fps; flower_garden and cheer_leaders are VGA to QVGA transcoding at 30 fps; vidyo1 and vidyo3 are 720p to 360p (640 × 360) transcoding at 60 fps. Akiyo, panzoom2, vidyo1 and vidyo3 are sequences similar to video-conferencing scenes with still background, slow motions or camera motions, while the other 4 are complex and detailed ones. For each sequence 150 frames are tested with the GOP structure of IPPP. The search range is 16 for CIF-to-QCIF transcoding and 32 for the rest. In experiments the QPs (Quantization Parameter) for AVC encoder and transcoder are set to same values, and QPs are selected as 20, 24, 28 and 32. Other parameters are carefully selected to insure the comparability between proposed transcoder and the reference model.

Table 6 shows the lower layer gains for the 3 methods as well as the adaptive usage, compared with re-encoding model. The methods in Sects. 4.1, 4.2 and 4.4 are applied and mode-mapping method is switched between the 4 methods described in Sect. 4.3. It can be seen that adaptive method achieves almost the same time reduction as candidate method, while obtaining comparable coding efficiency to priority method. It should be noticed that proposed adaptive method achieves 69.6% time reduction averagely for the top 4 sequences, which are video-conferencing similar scenes. For the following 4 sequences, the time reduction is only 35.1% averagely.

Table 7 shows the top-layer results for the 2 top-layer methods. The adaptive method in Sect. 4.3 is selected as lower-layer transcoding method along with other schemes (Sects. 4.1, 4.2 & 4.3). The bit-rate data contains only top-layer bit-rate and the lower-layer bit-rate is not included. To fairly compare the top layer quality, the Y-PSNR between original sequence (user side, encoded by AVC and sent to transcoder) and the reconstructed sequence after transcoding are calculated, since it's meaningless to calculate the Y-PSNR between decoded sequence (transcoder side, decoded

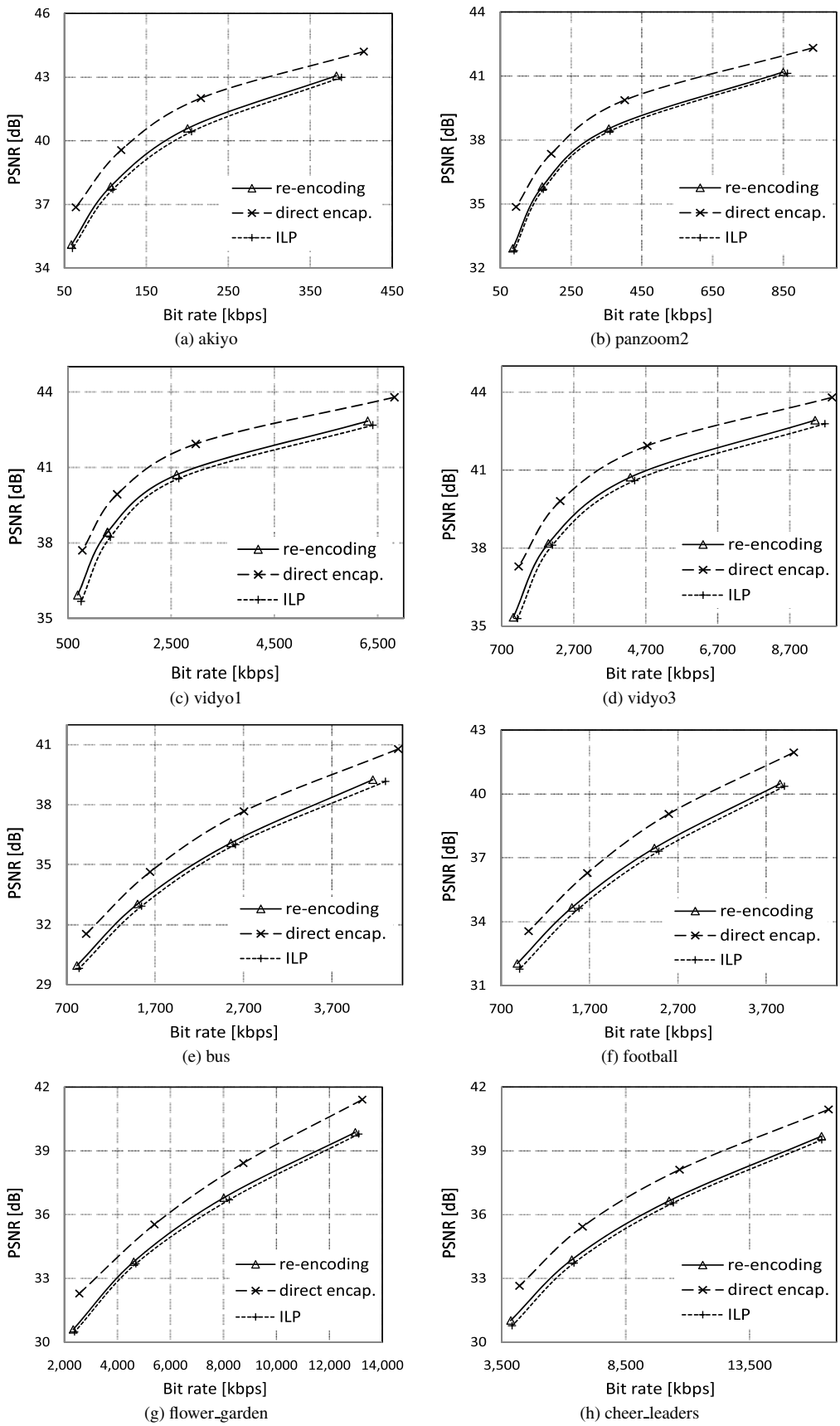


Fig. 9 R-D curves comparison for top layer.

and used as SVC encoder input) and reconstructed sequence which is identical to the original one in case of direct encapsulation.

Table 8 shows the overall results. The lower-layer scheme is fixed - Sects. 4.1, 4.2, 4.4 and adaptive method in Sect. 4.3. Both the overall bit-rate increments and overall time reductions are shown for the 2 top-layer methods.

The direct encapsulation method gains averagely 91% overall time reduction for tested sequences with 6.69% overall bit-rate increase and 1.34 dB top-layer quality increment. The time reduction for top 4 sequences is 1.6% larger than the lower 4 sequences. The merit for this method is the significant time reduction and the well-preserved top-layer quality since there is no second-time encoding.

By contrast, the ILP approach keeps the overall bit-rate low while still obtaining 78.7% overall time reduction averagely. It decreases the bit-rate increment to 1.71%. The time reduction for lower 4 sequences decreases by 7.9% compared with top 4 sequences. It is suitable for applications with limited network bandwidth. The main drawback is the degraded top-layer quality due to re-quantization loss which is a little worse than re-encoding method.

Figure 9 shows the top-layer R-D curves for re-encoding method and proposed transcoder with 2 different top-layer methods (Sects. 5.1 & 5.2). The X-axis shows the overall bit-rate since in ILP the lower-layer bit-rate is required for top-layer decoding. It would be unfair if we only compare the top-layer bit-rates. The Y-axis shows the Y-PSNR for top layer. We can see that the direct encapsulation method achieves best coding efficiency while ILP method is slightly worse than re-encoding method. Direct encapsulation method achieves higher quality and lower complexity compared with re-encoding method, while the overall bit-rate increases. The degree of quality increment is much higher than bit-rate increase, resulting in higher coding efficiency. ILP method achieves lower complexity compared with re-encoding method, however, the quality degrades a little and the bit-rate increases a little, causing the coding efficiency to decrease.

7. Conclusions

This paper proposes a low-complexity AVC to SVC spatial transcoder based on coarse-level mode mapping for video conferencing systems. For lower layer (with reduced picture size) transcoding, 2 mode skipping methods are first applied as described in Sects. 4.1 and 4.2. Then a coarse level mode-mapping method is applied which adaptively selects different sub-schemes described in Sect. 4.3, followed by an MV refinement scheme for special cases for further time reduction. And for the top layer (with identical picture size to AVC frame), 2 schemes are possible according to the network condition. Section 5.1 depicts the direct encapsulation method which is suitable when the bandwidth is sufficient, and Sect. 5.2 shows another approach which utilizes the inter-layer predictions of SVC for reducing the bit-rate. Simulation results show that direct encapsulation method achieves significant time reduction with much higher coding efficiency than re-encoding method, since no second-time quantization is involved. The ILP method achieves lower bit-rate than direction encapsulation when the QP is the same, while the time reduction reduces by 12.3% averagely. The coding performance of ILP method is slightly worse than re-encoding method.

Acknowledgments

This work was supported by KAKENHI (23300018) and Ambient SoC Global COE (GCOE) Program.

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] H. Schwarz and M. Wien, "The scalable video coding extension of the H.264/AVC standard [standards in a nutshell]," IEEE Signal Process. Mag., vol.25, no.2, pp.135–141, 2008.
[3] H. Choi, K. Lee, S.J. Bae, J.W. Kang, and J.J. Yoo, "Performance evaluation of the emerging scalable video coding," IEEE International Conference on Consumer Electronics (ICCE), pp.1–2, Las Vegas, NV, 2008.
[4] T. Oelbaum, H. Schwarz, M. Wien, and T. Wiegand, "Subjective performance evaluation of the SVC extension of H.264/AVC," 15th IEEE International Conference on Image Processing (ICIP), pp.2772–2775, San Diego, CA, 2008.
[5] E.D. Jang, J.G. Kim, T.C. Thang, and J.W. Kang, "Adaptation of scalable video coding to packet loss and its performance analysis," 12th International Conference on Advanced Communication Technology (ICACT), vol.1, pp.696–700, Phoenix Park, 2010.
[6] X. Li, P. Amon, A. Hutter, and A. Kaup, "Performance analysis of inter-layer prediction in scalable video coding extension of H.264/AVC," IEEE Trans. Broadcast., vol.57, no.1, pp.66–74, 2011.
[7] C.A. Segall and G.J. Sullivan, "Spatial scalability within the H.264/AVC scalable video coding extension," IEEE Trans. Circuits Syst. Video Technol., vol.17, no.9, pp.1121–1135, 2007.
[8] M.H. Willebeek-LaMair, D.D. Kandlur, and Z.Y. Shae, "On multipoint control units for videoconferencing," 19th Conference on Local Computer Networks (LCN), pp.356–364, Minneapolis, MN, 1994.
[9] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: An overview," IEEE Signal Process. Mag., vol.20, no.2, pp.18–29, 2003.
[10] T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats," IEEE Trans. Multimed., vol.2, no.2, pp.101–110, 2000.
[11] S. Li, L. Li, T. Ikenaga, S. Ishiwata, M. Matsui, and S. Goto, "Content-based complexity reduction methods for MPEG-2 to H.264 transcoding," IEICE Trans. Inf. & Syst., vol.E90-D, no.1, pp.90–98, Jan. 2007.
[12] H. Sun, W. Kwok, and J.W. Zdepski, "Architectures for MPEG compressed bitstream scaling," IEEE Trans. Circuits Syst. Video Technol., vol.6, no.2, pp.191–199, 1996.
[13] P.A.A. Assuncao and M. Ghanbari, "Post-processing of MPEG2 coded video for transmission at lower bit rates," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol.4, pp.1998–2001, Atlanta, GA, 1996.
[14] D.G. Morrison, M.E. Nilsson, and M. Ghanbari, "Reduction of the bit-rate of compressed video while in its coded form," 6th International Workshop on Packet Video, pp.392–406, Portland, OR, 1994.
[15] P.A.A. Assuncao and M. Ghanbari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams,"

IEEE Trans. Circuits Syst. Video Technol., vol.8, no.8, pp.953–967, 1994.

[16] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams," Signal Process.: Image Commun., vol.8, pp.481–500, 1996.

[17] N. Bjork and C. Christopoulos, "Transcoder architectures for video coding," IEEE Trans. Consum. Electron., vol.44, no.1, pp.88–98, 1998.

[18] B. Shen, I.K. Sethi, and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," IEEE Trans. Circuits Syst. Video Technol., vol.9, no.6, pp.929–936, 1999.

[19] W. Zhu, K.H. Yang, and M.J. Beacken, "CIF-to-QCIF video bitstream down-conversion in the DCT domain," Bell Labs Tech. J., vol.3, no.3, pp.21–29, 1998.

[20] P. Yin, A. Vetro, B. Liu, and H. Sun, "Drift compensation for reduced spatial resolution transcoding," IEEE Trans. Circuits Syst. Video Technol., vol.12, no.11, pp.1009–1020, 2002.

[21] J. De Cock, S. Notebaert, K. Vermeirsch, P. Lambert, and R. Van de Walle, "Efficient spatial resolution reduction transcoding for H.264/AVC," IEEE International Conference on Image Processing (ICIP), pp.1208–1211, San Diego, CA, 2008.

[22] R. Garrido-Cantos, J. De Cock, J.L. Martinez, S. Van Leuven, and P. Cuenca, "Motion-based temporal transcoding from H.264/AVC-to-SVC in baseline profile," IEEE Trans. Consum. Electron., vol.57, no.1, pp.239–246, 2011.

[23] J. De Cock, S. Notebaert, and R. Van de Walle, "Transcoding from H.264/AVC to SVC with CGS layers," IEEE International Conference on Image Processing (ICIP), vol.4, pp.IV-73–IV-76, San Antonio, TX, 2007.

[24] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Advanced bitstream rewriting from H.264/AVC to SVC," IEEE International Conference on Image Processing (ICIP), vol.1, pp.2472–2475, San Diego, CA, 2008.

[25] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Transcoding of H.264/AVC to SVC with motion data refinement," IEEE International Conference on Image Processing (ICIP), vol.1, pp.3673–3676, Cairo, 2009.

[26] J. De Cock, S. Notebaert, P. Lambert, and R. Van de Walle, "Architectures for fast transcoding of H.264/AVC to quality-scalable SVC streams," IEEE Trans. Multimed., vol.11, no.7, pp.1209–1224, 2009.

[27] H. Liu, Y. Wang, Y. Chen, and H. Li, "Spatial transcoding from scalable video coding to H.264/AVC," IEEE International Conference on Multimedia and Expo (ICME), pp.29–32, New York, NY, 2009.

[28] R. Sachdeva, S. Johar, and E. Piccinelli, "Adding SVC spatial scalability to existing H.264/AVC video," IEEE/ACIS International Conference on Computer and Information Science (ICIS), pp.1090–1095, Shanghai, 2009.

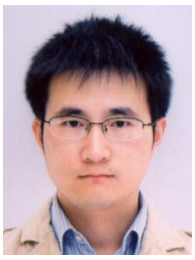
[29] P. Zhang, Y. Liu, Q. Huang, and W. Gao, "Mode mapping method for H.264/AVC spatial downscaling transcoding," IEEE International Conference on Image Processing (ICIP), vol.4, pp.2781–2784, Singapore, 2004.

[30] J. Youn, M. Sun, and C. Lin "Motion vector refinement for high-performance transcoding," IEEE Trans. Multimed., vol.1, no.1, pp.30–40, 1999.

Lei Sun received the B.E. degree in software engineering from Nanjing University, China, in 2008. And he received the M.E. degree from Graduate School of Information, Production and Systems of Waseda University, Japan, in 2010. He is currently working towards a Ph.D. degree in Waseda University. His research interests include image processing, video compression, network communication, and computer vision.

Jie Leng was born in 1987. He received the B.E. degree in information engineering from South China University of Technology (SCUT), China, in 2009. He is currently a master candidate in Graduate school of Information, Production and Systems (IPS), Waseda University, Japan. His research interest includes video compression algorithm and its VLSI hardware architecture design.

Jia Su received the B.E. degree in Telecommunications Engineering school of Xidian University, China, in 2006; and received M.E. degree both in Graduate School of Information, Production and Systems, Waseda University and School of Microelectronics in Xidian University in 2008 and 2009, respectively. She is currently working towards her Ph.D. degree at Waseda University. Her research interests include video compression and computer vision.



Yiqing Huang was born in 1982. He received the B.E. and M.E. degrees in communication and information engineering from Shanghai University, China in 2004 and 2007 respectively. In 2007, he received the M.E. and Ph.D. degrees from Graduate School of Information, Production and Systems, Waseda University, Japan. He is currently working at Ricoh R&D center, Japan. His research interest includes hardware friendly video coding algorithm and related very large scale integration (VLSI) architecture.

Hiroomi Motohashi received a B.E. degree of electrical engineering in 1987 from Waseda University, Tokyo, Japan. In 1987, he joined Ricoh Company where he was engaged in development of multifunction peripheral (MFP) product. In 2005, he joined Ricoh R&D center in Yokohama. He is now a senior specialist of 1st development department of office solution technology development center. His current interests are embedded system design, videoconferencing system, and high-dynamic range camera system for videoconferencing.

Takeshi Ikenaga received his B.E. and M.E. degrees in electrical engineering and Ph.D. degree in information & computer science from Waseda University, Tokyo, Japan, in 1988, 1990, and 2002, respectively. He joined LSI Laboratories, Nippon Telegraph and Telephone Corporation (NTT) in 1990, where he had been undertaking research on the design and test methodologies for high performance ASICs, a real-time MPEG2 encoder chip set, and a highly parallel LSI & system design for image-understanding processing. He is presently a professor in the system LSI field of the Graduate School of Information, Production and Systems, Waseda University. His current interests are application SoCs for image, security and network processing. Especially, he engages in the research on H.264 encoder LSI, JPEG2000 codec LSI, LDPC decoder LSI, UWB wireless communication LSI, public key encryption LSI, object recognition LSI, etc. Dr. Ikenaga is a member of IEEE, IPSJ and IIEEJ.