

DYNAMIC REGION OF INTEREST TRANSCODING FOR MULTIPOINT VIDEO CONFERENCING

Chia-Wen Lin

Department of Computer Science and Information Engineering,
National Chung Cheng University, Chiayi, Taiwan, R.O.C.

cwlin@cs.ccu.edu.tw

Yung Chang Chen

Department of Electrical Engineering
National Tsing Hua University, Hsinchu, Taiwan R.O.C.

ycchen@benz.ee.nthu.edu.tw

Ming-Ting Sun

Department of Electrical Engineering
University of Washington, Seattle, Washington, USA

sun@ee.washington.edu

ABSTRACT

This paper presents a region of interest transcoding scheme for multipoint video conferencing to enhance the visual quality. In a multipoint videoconference, usually there are only one or two active conferees at one time which are the regions of interest to the other conferees involved. We propose a Dynamic Sub-Window Skipping (DSWS) scheme to firstly identify the active participants from the multiple incoming encoded video streams by calculating the motion activity of each sub-window, and secondly reduce the frame-rates of the motion inactive participants by skipping these less-important sub-windows. The bits saved by the skipping operation are reallocated to the active sub-windows to enhance the regions of interest. We also propose a low-complexity scheme to compose and trace the unavailable motion vectors with a good accuracy in the dropped inactive sub-windows after performing the DSWS. Simulation results show that the proposed methods not only significantly improve the visual quality on the active sub-windows without introducing serious visual quality degradation in the inactive ones, but also reduce the computational complexity and avoid whole-frame skipping. Moreover, the proposed algorithm is fully compatible with the H.263 video coding standard.

1. INTRODUCTION

Video telephony is an efficient way for businesspersons, engineers, scientists, etc. to exchange information at remote locations. With the rapid growth of video telephony, the need of multipoint video conferencing is also growing. A multipoint videoconference involves three or more conference participants. In continuous presence video conferencing, each conferee can see others in the same window simultaneously [1]. Fig. 1 depicts an application scenario of multiple persons participating in a multipoint videoconference with a

centralized server. In this scenario, multiple conferees are connected to the central server, referred to as the Multipoint Control Unit (MCU), which coordinates and distributes video and audio streams among multiple participants in a video conference according to the channel bandwidth requirement of each conferee. A video transcoder [2-5] is included in the MCU to combine the multiple incoming encoded video streams from the various conferees into a single coded video stream and send the re-encoded bit-stream back to each participant over the same channel with the required bit-rate and format for decoding and presentation. In the case of a multipoint videoconference over PSTN (Public Switch Telephone Network) or ISDN (Integrated Service Digital Network), the channel bandwidth is constant and symmetrical. Assuming each conferee has a channel bandwidth of B kb/s, then MCU receives each conferee's video at B kb/s each, decodes and combines the videos, and re-encodes the combined video at B kbps so as to meet the channel bandwidth requirements for sending back the encoded video to the conferees. Therefore it is required to perform bit-rate conversion/reduction at the video transcoder. Bit-rate conversion from high bit-rate to low bit-rate in video transcoding will, however, introduce video quality degradation. The visual quality, the computational load, and the used bit-rates need to be traded off in video transcoding to achieve a good solution.

The problem of how to efficiently redistribute the limited bit-rates to different parts of a video in video transcoding is critical in providing satisfactory visual quality. In a multipoint videoconference, most of the time only one or two conferees are active at one time. The active conferees need higher bit-rates to produce good quality video while the inactive conferees require lower bit-rates to produce acceptable quality video [4]. Simply uniformly distributing the bit-rates to the conferees will result in non-uniform video quality. To make best use of the available bit-rates, a joint rate-control scheme which takes into account each conferee sub-window's activity

can be used [4-5]. Sun *et al.* [4] proposed to measure the motion activity of each sub-window by calculating the sum of the magnitudes of its corresponding motion vectors, and allocate the bit-rates to each sub-window according to its activity. Thus more bits will be allocated to those sub-windows with higher activities, and this strategy will produce much more uniform quality video. Wu *et al.* [5] extended the work in [4] by allocating the bit-rates to each sub-window according to its spatial-temporal activity which takes into account the motion, the variance of the residual signal, and the number of the encoded macroblocks. Similar work on joint rate-control can also be found in the statistical multiplexing (StatMux) of multiple video programs [6-8] and MPEG-4 joint rate-control of multiple video objects [9]. However the strong correlation among the conferees in a multipoint video conference usually does not exist in the general cases of the StatMux and MPEG-4 object rate control. In addition, we will show that dynamic temporal resolution control [10] for each sub-window, which has not been addressed in the previous work, may bring further coding gain and computation reduction in multipoint video transcoding.

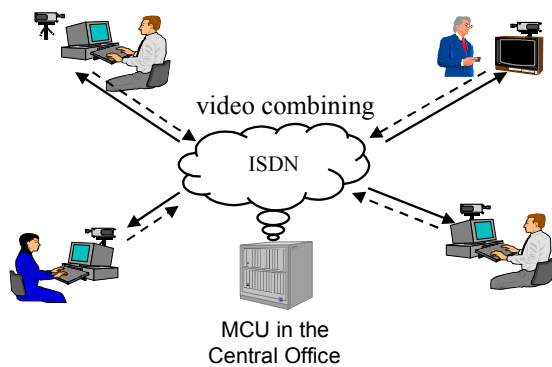


Fig. 1. Application example of multipoint video conferencing

In this paper, we present a DSWS scheme which provides the flexibility that sub-windows can be encoded in different temporal resolutions according to their motion activities. The proposed DSWS scheme classifies the sub-windows into active and inactive classes by calculating the associated motion activities. The inactive sub-windows can then be dropped so that the saved bits can be used to significantly enhance the visual quality of the active ones without introducing serious degradation on the inactive ones. In addition to the performance gain on active sub-windows, the DSWS scheme also presents two other advantages: achieving computation reduction and avoiding the whole-frame skipping. Furthermore, we present a motion-vector composing scheme which can compose and trace the unavailable motion vectors in the dropped sub-windows with good accuracy at a very low computational and memory cost.

The rest of this paper is organized as follows. Section 2 presents the proposed DSWS scheme. In Section 3, a pre-filtered activity-based motion-vector composing scheme is proposed for composing the unavailable motion

vectors in the dropped sub-windows after performing the DSWS scheme. Section 4 reports the experimental results of the proposed algorithms and the comparison with the H.263 [11] TMN8 direct transcoding method. Finally, conclusions are drawn in Section 5.

2. DYNAMIC SUBWINDOW SKIPPING (DSWS)

Fig. 2 depicts the architecture for multipoint video transcoding discussed in this paper. In this architecture, the input data for the video transcoder consist of multiple H.263 encoded video streams from the client-terminals through a heterogeneous network environment. The video streams could be transmitted through PSTN, ISDN, or LAN (Local Area Network) with various bandwidth requirements. For simplicity but without loss of generality, we assume each video stream is encoded in the QCIF (176x144) format and each participant can see 4 participants in a CIF (352x288) frame in a continuous presence fashion. In our experiments we assume the video transmission is over ISDN as shown in the scenario in Fig. 1. As shown in Fig. 2, the input video streams first are buffered at the input to regulate the difference between the input data-rate and the transcoding data-rate for each video frame. Each video stream is decoded through a Variable Length Decoder (VLD) and then transcoded into lower data-rates. To meet various user bit-rate requirements, more than one transcoder may be required for the same video stream to generate multiple video streams with different bit rates. The transcoded bit-streams are subsequently combined into CIF frames through a number of multiplexers and video combiners. The multiplexing unit for the video combiner is the GOB (Group of Block) as specified in the H.263 standard [11].

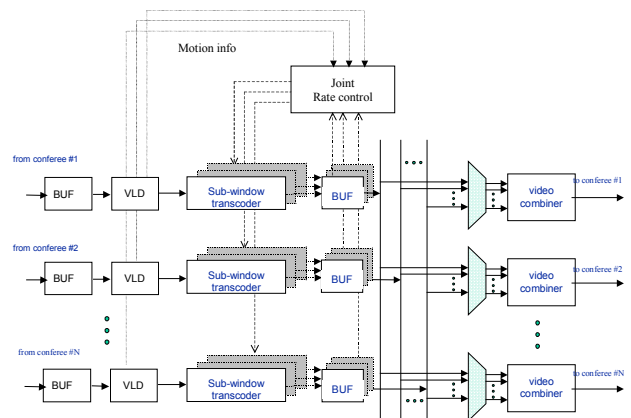


Fig. 2. The proposed multipoint video transcoding architecture

As mentioned above, in a multipoint video conference, usually only one or two persons are motion active at one time. The active conferees (e.g., the conferees who are speaking) are often the center of focus. Therefore, allocating the active sub-windows with relatively higher

bit-rates can provide a much better visual experience to the viewers. The active conferees usually have larger motions than others, thus they can be easily detected by using the incoming motion information.

In this paper, we observe that in multipoint video conferencing, the temporal resolutions of the low-active sub-windows may be reduced without introducing significant visual degradation from the Human Visual System (HVS) point of view. The main reason is that, since the motions presented by the inactive sub-windows are relatively slow and the conferees usually concentrate their focuses on the active ones, the effect of the temporal resolution reduction by skipping inactive sub-windows can often be masked by the high motions in the active sub-windows thus is not sensitive to viewers' perceptions. To make best use of this property, we propose to drop motion inactive sub-windows by using sub-window repetition to approximate those dropped sub-windows at the end decoder, so that the saved bits can be used to enhance the quality of the remaining non-skipped active ones which are usually the regions of interest. In addition, if a sub-window is decided to be skipped, much computation in transcoding this sub-window can be saved, thus significant computation reduction can be achieved. Sub-window skipping can be implemented in the H.263 syntax [11] by simply setting all the COD bits of the Macroblocks (MBs) belonging to the skipped sub-windows to "1" to get rid of sending the associated DCT coefficients, motion vector information, and MB overhead bits. Only 99 bits (for 99 MB COD bits respectively) are required to represent a skipped QCIF (176x144) sub-window, thus the overhead is relatively negligible. In our proposed DSWS scheme, the motion information is used to calculate the motion activity of each sub-window for dynamic sub-window skipping control. The DSWS scheme is summarized as follows:

if ($S_m^{MV} < TH_{MV}$) && ($SAAD_m < TH_{SAAD}$)

then

Skip the transcoding of the m th sub-window

else

Transcode the m th sub-window

where the sum of the magnitude of accumulated motion vectors of the m th sub-window is defined as

$$S_m^{MV} = \frac{1}{N} \sum_{n=1}^N (|MV_{m,n}^x| + |MV_{m,n}^y|), \quad (1)$$

N is the number of MBs in a sub-window, $(MV_{m,n}^x, MV_{m,n}^y)$ is the motion vector associated with the n th macroblock of the m th sub-window with respect to its corresponding previously encoded sub-window (i.e., $f_m^{\text{prev}}(\cdot)$), and the Sum of Accumulated Absolute Difference (SAAD) of the sub-window is defined as

$$SAAD_m = \frac{1}{N} \sum_{n=1}^N \sum_{x,y \in \text{MB}_{m,n}} |f_m(x,y) - f_m^{\text{prev}}(x + MV_{m,n}^x, y + MV_{m,n}^y)| \quad (2)$$

The sum of the magnitude of motion vectors of a sub-window can be used as a good indication of its motion activity. A sub-window is classified as active if the sum is larger than a predetermined threshold, TH_{MV} , otherwise it is classified as inactive. An inactive sub-window will be skipped if its associated SAAD value defined in (2) is below a threshold, TH_{SAAD} . If an inactive sub-window is skipped, the corresponding latest non-skipped sub-window is repeated to approximate the skipped sub-windows. Human visual perception is relatively insensitive to the little differences between the skipped sub-windows and their reconstructed ones from sub-window repetition if the skipped sub-windows are inactive. The two thresholds, TH_{MV} and TH_{SAAD} , are used for the classification, the larger the thresholds are set, the more the sub-windows will be skipped, and the more the saved bits will be used in other sub-windows (but jerky motions will become more serious). The SAAD value of each sub-window is used to constrain the sub-window skipping. If the current frame is inactive but the SAAD is larger than a threshold, the proposed method enforces that the sub-window, which would otherwise be skipped, be encoded. This measure can prevent the error accumulation caused by slow but steady motions by using only the motion activity measure as in [10]. It should be noted that, since the incoming sub-windows may be dropped in consecutive frames with the DSWS method, the incoming motion vectors may not be valid since they may point to the dropped sub-windows that do not exist in the transcoded bit-stream. To compose and trace the required but unavailable motion vectors along the consecutively skipped sub-windows with respect to the corresponding latest encoded sub-windows, the motion vector composing scheme proposed in Section 3 is used.

The proposed DSWS scheme presents several advantages. First, the quality of the active sub-windows can be effectively enhanced. The quality loss on the inactive sub-windows is relatively small and visually insensitive to the viewer's perception. Second, skipping a sub-window implies saving much computation in transcoding that sub-window (in our simulation, about 2/3 of the computation in transcoding that sub-window can be saved), thus achieving significant computation reduction. Finally, by skipping the motion inactive sub-windows, many whole-frame skipplings due to insufficient bit-allocation can be avoided, so that the temporal resolution of the motion active sub-windows can be kept as high as possible. Moreover, the proposed method can be combined with the dynamic bit-allocation scheme presented in [4] to further improve the visual quality with almost no extra complexity.

3. COMPOSING MOTION VECTORS IN THE SKIPPED SUB-WINDOWS

After performing the DSWS scheme, sub-windows may be skipped in consecutive frames. Similar to the frame-rate conversion situation stated in [10,12], the motion vectors in the dropped sub-windows are usually unavailable in the incoming bit-stream, thus need to be re-computed. For example, in Fig. 3, a situation where one sub-window is dropped in two consecutive frames in transcoding is illustrated. As shown in Fig. 3, each video stream going into the MCU carries the incoming motion vectors IV_m^n , where n and m represent the frame and MB sequence numbers respectively, for the 16x16 MBs on the segmented grid. In this example, the equivalent outgoing motion vector sent from the MCU to the client-terminals of the block MB_1^n should be $OV_1^n = IV_1^n + MV_1^{n-1} + MV_1^{n-2}$, instead of the incoming motion vector IV_1^n . However, MV_1^{n-1} and MV_1^{n-2} do not exist in the incoming bit-stream since MB_1^{n-1} and MB_1^{n-2} are not aligned with the MB grid. Thus the outgoing motion vector needs to be either re-estimated using motion estimation schemes or composed using the available motion information of the MBs on the grid.

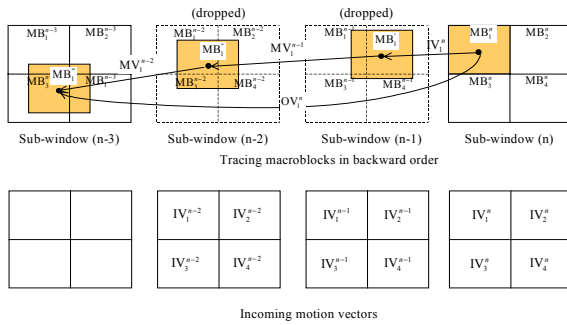


Fig. 3. Motion vector tracing in a MB with 2 sub-windows skipping.

Motion vector re-estimation is undesirable due to the intensive computation. Instead, motion vector composing using the available motion information is a better approach [10,12]. Similar problem has been discussed for video downscaling applications [13-15]. When a MB is not aligned with the MB grid, it will in general overlap with four MBs with the corresponding motion vectors $\{IV_1^{n-k}, IV_2^{n-k}, IV_3^{n-k}, IV_4^{n-k}\}$. Composing the motion vector MV^{n-k} from the neighboring motion vectors $\{IV_1^{n-k}, IV_2^{n-k}, IV_3^{n-k}, IV_4^{n-k}\}$ is to find a mapping function $\widehat{MV}^{n-k} = f(IV_1^{n-1}, IV_2^{n-1}, IV_3^{n-1}, IV_4^{n-1})$ which can approximate MV^{n-k} with a good accuracy. One straightforward mapping function for composing the

motion vector MV^{n-k} is an interpolation with a form [10,13-14]:

$$\widehat{MV}^{n-k} = \frac{\sum_{i=1}^4 IV_i^{n-k} A_i ACT_i}{\sum_{i=1}^4 A_i ACT_i} \quad (3)$$

where A_i and ACT_i represents the corresponding overlapping area with and the activity of the i th neighboring grid MB respectively. In (3), the block activities, ACT_i 's, can be computed in the pixel-domain or the DCT-domain. Since the number of the DCT coefficients of a MB is usually small, computing the activity in the DCT domain is much more efficient. We propose to use the sum of the magnitude of the non-zero DCT AC coefficient as the activity measure as follows:

$$ACT_i = \sum_{j \in DC} |Coef_{i,j}| \quad (4)$$

where $Coef_{i,j}$ is the j th nonzero DCT AC coefficient of the i th anchor block which is decoded and de-quantized from the incoming bit-stream. Since the number of the nonzero de-quantized DCT coefficients in a residual block is usually small, the computational cost is thus quite low.

As explained in [12], there are several drawbacks with the above interpolation scheme. First, for consecutively dropped sub-windows, the interpolation should be processed in backward order starting from the last dropped sub-window to the first dropped sub-window. This backward processing requires all motion vectors of the dropped sub-windows be stored, which requires much extra memory. Another drawback of the interpolation scheme is the inaccuracy of the resultant motion vector. In spite of the proper weighting of each neighboring motion-vector based on the associated overlapping area and activity, unreliable motion vectors can be produced because the area covered by four neighboring MBs may be too divergent and too large to be described by a single motion. The interpolation of these diverse motion flows thus may not produce an accurate motion vector. To improve the results, a Forward Dominant Vector Selection (FDVS) scheme was proposed in [12] to compose and trace the un-available motion vectors in the dropped frames. Instead of interpolating the target motion vector from its four neighboring motion vectors, the FDVS method selects one out of four neighboring motion vectors as a dominant one to approximate the target motion-vector, i.e., $\widehat{MV}^{n-k} = \text{dominant}(IV_1^{n-1}, IV_2^{n-1}, IV_3^{n-1}, IV_4^{n-1})$, where the dominant motion-vector is defined as the motion-vector carried by the neighboring MB which overlaps the target MB the most. Fig. 4 illustrates the FDVS scheme for the 2-frame skipping case. The FDVS scheme can compose the motion vectors with good accuracy and low complexity in the forward order [12].

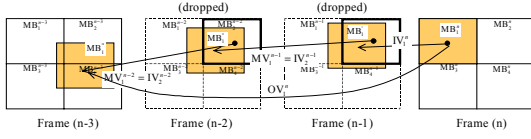


Fig. 4. Forward dominant vector selection schemes [12].

The performance of the FDVS method can be further improved with a little extra computational complexity. If there is no strongly dominant MB which overlaps the reference block with a significantly large area (e.g., the overlapping area is larger than a predefined threshold, say 80% of the block area), selecting a dominant vector which diverges largely from the other neighboring motion vectors may degrade the quality significantly since the motion vector chosen may be unreliable. To solve this problem, we propose to remove the unreliable motion vectors from the candidate list before selecting the dominant motion vector if no strongly dominant MB is found. Furthermore, in the dominant vector selection, the “largest overlapping area” may not be the best criterion when the overlapping areas of some of the other neighboring anchor blocks are similar. In this case, we propose to select the neighboring MB with the largest overlapping energy/activity as the dominant MB and use the activity measure as defined in (4).

The proposed Pre-filtered Activity-based FDVS scheme (PA-FDVS) is summarized as follows:

- (1) For each MB, calculate the largest overlapping area, and if the largest overlapping area is greater than a predetermined threshold (e.g., 80% in our simulation) then select the motion vector of the neighboring MB with the largest overlapping area as the dominant vector and process the next MB, otherwise go to step (2).
- (2) Perform the following motion vector pre-filtering procedure:
Set the initial candidate list as the four neighboring vectors $\{IV_1, IV_2, IV_3, IV_4\}$

Calculate the mean and the standard deviation of the four neighboring motion vectors as follows:

$$IV_{\text{mean}} = \frac{1}{4} \sum_{i=1}^4 IV_i$$

$$IV_{\text{std}} = \sqrt{\frac{1}{4} \sum_{i=1}^4 (IV_i - IV_{\text{mean}})^2}$$

for $i = 1$ to 4

if $|IV_i - IV_{\text{mean}}| > k_{\text{std}} \cdot IV_{\text{std}}$

IV_i is unreliable, remove it from the dominant vector candidate list

else

IV_i is reliable, keep it in the dominant vector candidate list

where K_{std} is a predefined constant.

- (3) Calculate the area-activity products $A_i \cdot \text{ACT}_i$ for the MBs with the motion vector in the dominant vector candidate list, where A_i is the overlapping area with the neighboring block (i) and ACT_i is the activity measure as defined in (5). Then select the motion vector of the neighboring MB with the largest area-activity product as the dominant vector.

4. EXPERIMENTAL RESULTS

In our experiments, two 200-frame standard QCIF (176x144) test sequences “foreman” and “carphone” with a frame rate of 30 frames/s are used to verify the performance of the proposed PA-FDVS scheme. The performance comparisons of the full-search motion estimation method and the motion-vector composing methods (interpolation, FDVS, and PA-FDVS methods) using the two test sequences are shown in Table 1 and Figure 5. Table 1 shows the average PSNR comparisons for the two test-sequences which were first encoded with 128 kb/s and 30 frames/s and then transcoded with 32 kb/s and 7.5 frames/s. The result in Table 1 indicates that PA-FDVS performs better than FDVS and significantly outperforms the interpolation scheme. The frame-by-frame PSNR comparison of the PA-FDVS and FDVS schemes with the same test condition used in Table 1 are shown in Figure 5. Although the average PSNR values of the PA-FDVS and FDVS in Table 1 are close, Figure 5 suggests that the PA-FDVS scheme achieves significant PSNR improvement (up to 1.6 dB and 2.9 dB for the “foreman” and “carphone” sequences respectively) over the FDVS scheme on several frames with many divergent object-motions.

Table 1

Performance comparison of different motion vector estimation and composition methods. Incoming bit-streams of 128 kb/s and 30 fps were transcoded into 32 kb/s and 7.5 fps

Test sequence	MV composition method	Average PSNR
Foreman	Full-scale ME	27.39
	Interpolation	23.72
	FDVS	25.51
	FA-FDVS	25.67
Carphone	Full-scale ME	29.47
	Interpolation	27.07
	FDVS	28.16
	FA-FDVS	28.27

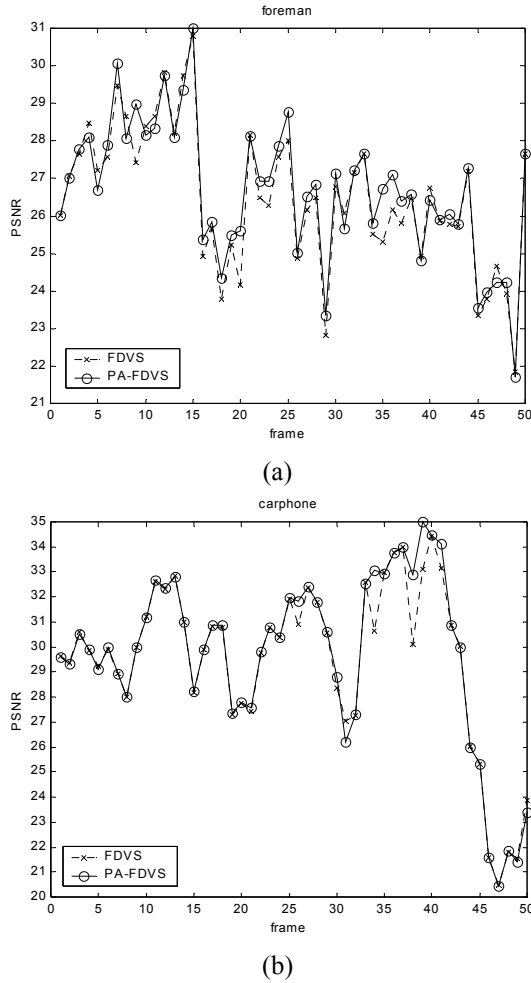


Fig. 5. Performance comparison of the FDVS and the FA-FDVS schemes. Incoming bit-streams of 128 Kb/s and 30 fps are transcoded with 32 Kb/s and 7.5 fps: (a) “foreman” sequence; (b) “carphone” sequence.

To verify the effectiveness of the proposed DSWS scheme, four 400-frame QCIF video sequences captured from a four-point video-conference with a frame-rate of 15 frames/s are used for experiments. We firstly encoded the four QCIF video sequences with 128 kb/s and 15 frames/s using the public-domain H.263 TMN8 software [16]. The four input bit-streams are then jointly transcoded into a single CIF (352x288) video with an output bit-rate of 128 kb/s and an output frame-rate of 15 frames/s using the proposed DSWS scheme. The compression ratio performed by the transcoder is thus four in our experiments.

Fig. 6 depicts the motion-activity of each sub-window. In the simulated video conference session, most of the time only one or two sub-windows are motion active. Figure 7 compares the frame-by-frame PSNR performance of the proposed DSWS method and the direct transcoding using TMN8 [17]. The PSNR of a skipped sub-window is computed from the incoming QCIF sub-window images and the latest previously reconstructed non-skipped one, since the sub-window repetitions will occur for the

skipped sub-windows at the video decoders. The thresholds, TH_{MV} and TH_{SAD} , are empirically set at 0.2 and 10 respectively. Fig. 7 illustrates that the proposed DSWS method achieves PSNR gain on the sub-windows with relatively high activities, while the low-activity sub-windows are degraded. Table 2 shows the comparison of the average PSNR of all the sub-windows using the two methods. As shown in Fig. 7 and Table 2, the proposed DSWS scheme achieves 0.2 and 0.39 dB average PSNR improvements on the overall and the non-skipped sub-windows respectively. In sub-window 4, the performance is degraded by 0.4 dB because of many long intervals with relatively low motion-activity. The degradation is caused by the temporal resolution reduction by repeating the previously decoded sub-window for the period of sub-window skipping, the temporal resolution reduction is visually insignificant, since the motions of these sub-windows are very small. In this simulation case, 418 out of 1600 sub-windows are skipped, thus achieving about 17% computation reduction since the computation required for sub-window skipping decision and motion-vector composing is about 1/3 of the computation for sub-window transcoding. The computation reduction ratio depends on the two threshold values: TH_{MV} and TH_{SAD} . The higher the threshold values, the lower the computation demand, however the lower the video quality. It is thus possible to achieve better trade-offs between computational cost and video quality by adjusting the threshold values adaptively.

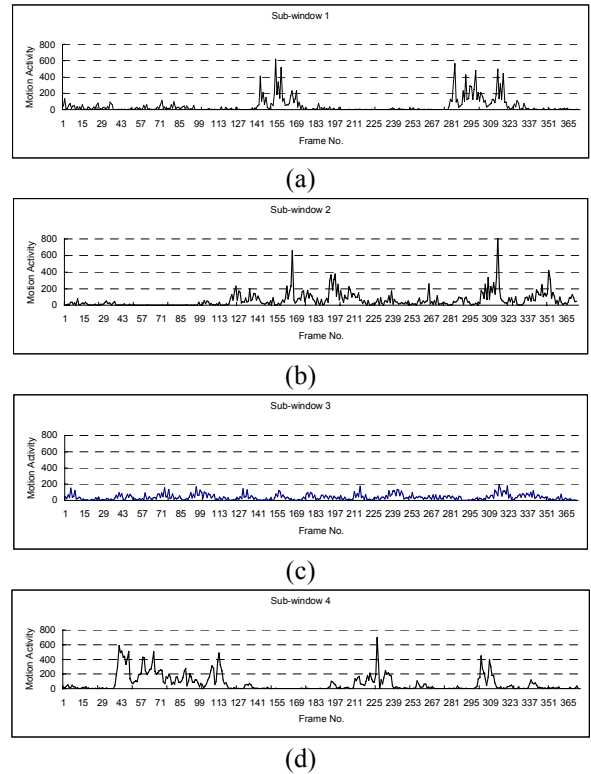
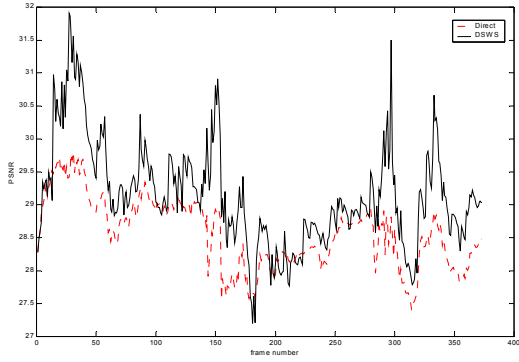
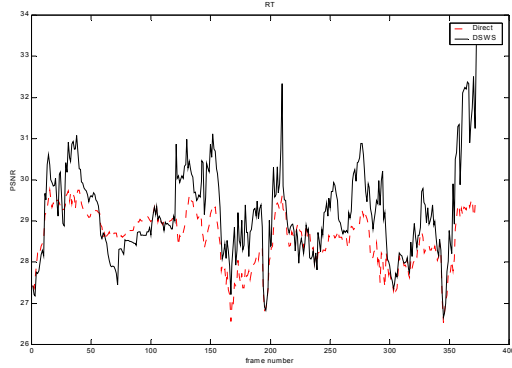


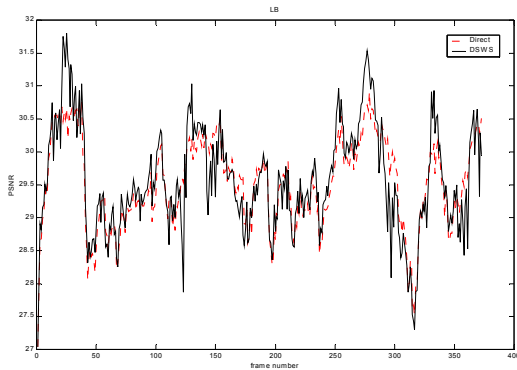
Fig. 6. Motion-activity of each sub-window



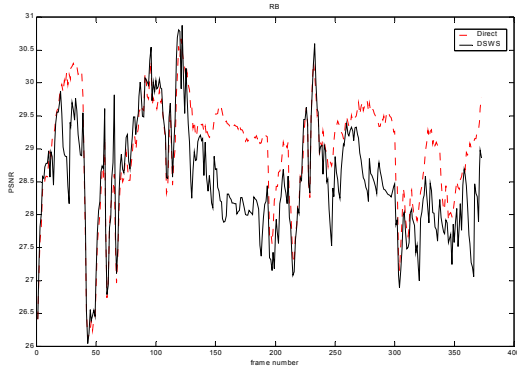
(a)



(b)



(c)



(d)

Fig. 7. PSNR comparison of the proposed method and TMN8

Table 2
Average PSNR Comparison of the proposed DSWS and direct transcoding schemes

	Average PSNR of all frames(dB)			Average PSNR of non-skipped frames (dB)		
	Ori.	DSWS		Ori.	DSWS	
Sub-window 1 (151 skipped)	28.54	29.14	+0.60	28.55	29.31	+0.76
Sub-window 2 (75 skipped)	28.59	29.16	+0.57	28.52	29.25	+0.73
Sub-window 3 (54 skipped)	29.54	29.56	+0.02	29.48	29.59	+0.11
Sub-window 4 (139 skipped)	28.99	28.59	-0.40	28.73	28.68	-0.05
Average	28.91	29.11	+0.20	28.82	29.21	+0.39

5. CONCLUSIONS

In this paper, we proposed a dynamic sub-window skipping scheme for multipoint video conferencing. The proposed scheme can enhance the visual quality of the active sub-windows by saving bits from skipping the inactive ones without introducing significant quality degradation. We also presented an efficient motion-vector composing scheme to compose and trace the motion vectors in the skipped sub-windows

The proposed method is particularly useful in multi-point video conferencing applications since the focuses in such applications are mainly on the active conferees. Simulation results verify the effectiveness of the proposed method. Significant computation reduction is also achieved with the proposed method. Furthermore, the proposed algorithm is fully compatible with the H.263 standard, thus can be integrated into current commercial products. The proposed method can also be further extended to enhance the quality of specific regions/objects in region/object-based coding standards such as MPEG-4 [5].

6. REFERENCES

- [1] M.-T. Sun and I.-M. Pao, "Multipoint video conferencing," *Visual Commun. and Image Proc.*, Marcel Dekker, 1997.
- [2] G. Keesman et al., "Transcoding of MPEG bitstream," *Signal Proc. Image Commun.*, pp. 481-500, 1996.
- [3] 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. 953-567, Dec. 1998.
- [4] M.-T. Sun, T.-D. Wu, and J.-N. Hwang, "Dynamic bit allocation in video combing for multipoint video

conferencing," *IEEE Trans. on Circuit and Systems.*, vol. 45, no. 5, pp. 644-648, May. 1998.

- [5] G. Keesman, "Multi-program video compression using joint bit-rate control," *Philips Journal of Research*, vol. 50, pp. 21-45, 1996.
- [6] L. Wang and A. Vincent, "Bit allocation and Constraints for Joint Coding of Multiple Video programs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 949-959, Sep. 1999.
- [7] A. Vetro, H. Sun and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 186-199, Feb. 1999.
- [8] M. E. Lukacs and D. G. Boyer, , and M. Mills, "The personal presence system experimental research prototype," *IEEE Int. Conf. Comm.*, vol. 2, pp. 1112-1116, Jun. 1996.
- [9] M. E. Lukac and D. G. Boyer, "A universal broadband multipoint teleconferencing service for the 21st century," *IEEE Comm. Magazine*, vol. 33, no. 11, pp. 36-43, Nov. 1995.
- [10] J.-N. Hwang, T.-D. Wu, and C.-W. Lin, "Dynamic frame skipping in video transcoding," *Proc. IEEE Workshop on Multimedia Signal Proc.*, Dec. 1998, Redondo Beach, USA.
- [11] ITU-T Recommendation H.263, "Video codec for low bit-rate communication," May. 1996.
- [12] J. Youn, M.-T. Sun and C.-W. Lin, "Adaptive motion vector refinement for high performance transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30-40 Mar. 1999.
- [13] N. Bjorkand and C. Christopoulos, "Transcoder architecture for video coding," *IEEE Trans. Consumer Electron.*, vol.44, pp.88-98, Feb.1998.
- [14] B. Shen, I.K. Sethi, and V. Bhaskaran, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 929-936, Sep. 1999.
- [15] M. R. Hashemi, L. Winger, and S. Panchanathan, "Compressed domain vector resampling for downscaling of MPEG video," *Proc. IEEE Int. Conf. Image Proc.*, pp. 276-279, Japan, Oct. 1999.
- [16] Image Proccsing Lab, University of British Columbia, "H.263+ encoder/decoder," TMN(H.263) codec, Feb. 1998.
- [17] ITU-T/SG16, "Video codec test model, TMN8," Portland, June 1997.