MOTION ESTIMATION FOR HIGH PERFORMANCE TRANSCODING

Jeongnam Youn, Ming-Ting Sun and Chia-Web Lin[1]
Information Processing Laboratory
Department of Electrical Engineering, Box #352500
University of Washington, Seattle, WA 98195
[1]Computer and Communication Research Labs
ITRI, Hsinchu, Taiwan 310, ROC

Abstract

Traditionally, the re-use of motion vectors extracted from incoming video bit-stream during transcoding has been widely accepted. However, this simple re-use scheme introduces significant quality degradation in many applications including the situation when the frame-rate conversion is needed. In this paper, we analyzed the quantization errors that cause the extracted motion vectors to be non-optimal and we performed simulations to show the quality degradation due to the inaccurate motion vectors during transcoding. To improve the video quality, we proposed an adaptive motion vector refinement. With a highly reduced computational complexity, the proposed adaptive motion vector refinement achieves significant quality improvement in comparison to the conventional motion vector re-use scheme. In addition, the adaptive motion vector refinement is almost as good as performing a new full-scale motion estimation.

1. Introduction

Transcoding, as a process of converting a previously compressed video bit-stream into a lower bit-rate video bit-stream has been studied recently in several literatures [1-6] due to its wide range of applications. A typical application of a transcoder is for video services over heterogeneous networks in which end-users require different Quality of Service (QoS) [9]. Because different networks may have different bandwidths, a gateway can include a transcoder to adapt the video bit-rates in order to provide video services to users on different networks.

The simple approach for implementing the transcoder is the use of open-loop transcoding in which the incoming bit-rate is down-scaled by truncating the DCT coefficients, by performing a requantization process or by selecting arbitrarily selecting the DCT coefficients [1,2]. Since the transcoding is done in the coded domain, a very simple and fast transcoder is possible. However, the open-loop transcoding produces an increasing distortion caused by the "drift" due to the mismatched reconstructed pictures in the encoder and the decoder. This results in an unacceptable video quality in many situations. Drift-free transcoding [4] is possible by using a decoder to decode the incoming video and then using an encoder to re-encode the video at the lower rate. When a pre-encoded video stream arrives at the transcoder, it already carries a great deal of useful information such as picture types, motion vectors, quantization step-sizes, bit-allocation statistics, and so forth. This makes it possible to construct transcoders with different performance in terms of complexity and video quality [3,5].

One aspect that has not been fully discussed in the literature is the motion estimation in the transcoder [10]. Traditionally, motion estimation has not been considered in transcoding because of its high computational complexity. Furthermore it was generally thought that using the extracted motion vectors from the incoming video stream for the outgoing video stream would be almost as good as performing a new motion estimation. In this paper, we demonstrate that in many applications this

Manuscript received June 17, 1998

simple re-use scheme introduces significant quality degradation. We analyzed the quantization errors that cause the non-optimized motion vectors and we performed simulations to show the quality degradation due to the inaccurate incoming motion vectors. First, we proposed a motion vector refinement scheme that used the motion vector extracted from the incoming video stream as the base motion vector and then performed a motion estimation in a very small search range around the base motion vector. We discussed the use of the motion vector refinement scheme when a frame-rate conversion occurs during transcoding. Then, we proposed an adaptive motion vector refinement scheme based on the quantization information. Through this method, we showed that this adaptive motion vector refinement scheme achieves significant reduction of the computational complexity and is almost as good as performing a new full-search motion estimation.

The organization of this paper is as follows. In section 2, we analyze the quantization errors that cause the extracted motion vectors to be non-optimal. In section 3, we propose the motion vector refinement that is able to improve the video quality significantly for the transcoder without the computation burden of performing a new full-scale full-search motion estimation. We also discuss the motion vector refinement scheme when the frame-rate is changed during transcoding. In section 4, the adaptive motion vector refinement based on the analysis of quantization errors is introduced. Simulation results are presented in section 5. Finally, a conclusion is provided in section 6.

2. Motion Estimation in Transcoding

2.1 Full Motion Estimation in Transcoding

In most current video coding standards including MPEG, H.261 and H.263, motion estimation is performed on the luminance macroblocks based on the Sum of Absolute Difference (SAD). In order to obtain the motion vector for the current macroblock, the best matching block that results in a minimal SAD is searched within a predefined search area S in the previous reconstructed reference frame. Figure 1 shows the structure of a transcoder constructed by cascading a decoder and an encoder. Since the output bit-rate is lower than the input bit-rate, usually the quantizer step size Q2 in the transcoder is much coarser than the quantizer step size Q1 in the front encoder.

The motion vector *(Ix, Iy)* of the current macroblock in the front encoder (or the first-stage encoder) proceeding to the transcoder is obtained by:

$$(Ix,Iy) = \arg\min_{(m,n)\in S} SAD_f(m,n),$$

$$SAD_f(m,n) = \sum_i \sum_j |P_f^c(i,j) - R_f^p(i+m,j+n)|,$$

where m and n are the horizontal and vertical components of the motion vector. The $P_f^c(i,j)$ and $R_f^p(i+m,j+n)$ represent a pixel in the current frame and a displaced pixel by *(m, n)* in the previous reconstructed reference frame respectively, the superscript "c" or "p" represents the "current" or "previous" frame respectively, and the subscript "f" indicates the first-stage encoder. *(Ix, Iy)* should be within a predefined search area S.

In the transcoder, optimized motion vectors for the outgoing video stream can be obtained by applying the full-scale full-search motion estimation. In this case, the decoded video stream in the transcoder becomes the input video stream for the encoder in the transcoder (or the second-stage encoder). If the pixels of the previously reconstructed frame and the current frame in the second-stage encoder are $R_s^p(i,j)$ and $P_s^c(i,j)$ respectively, then the motion vector *(Ox, Oy)* by a full-scale full-search motion estimation in the second-stage encoder is given by:

$$(Ox,Oy) = \arg\min_{(m,n)\in S} SAD_s(m,n),$$

$$SAD_s(m,n) = \sum_i \sum_j |P_s^c(i,j) - R_s^p(i+m,j+n)|,$$

where the subscript "s" indicates the second-stage encoder.

From Figure 1, since the reconstructed picture of the first-stage decoder R_f is the same as the current picture of the second-stage encoder P_s,

$$SAD_s(m,n) = \sum_i \sum_j |P_s^c(i,j) - R_s^p(i+m,j+n) + (P_f^c(i,j)$$
$$- R_f^p(i+m,j+n)) - (P_f^c(i,j) + R_f^p(i+m,j+n))|$$
$$= \sum_i \sum_j | P_f^c(i,j) - R_f^p(i+m,j+n)$$
$$+ \Delta^c{}_f(i,j) - \Delta_s^p(i+m,j+n)|,$$

where $\Delta^c{}_f(i,j) = R_f^c(i,j) - P_f^c(i,j)$ *and*

$$\Delta_s^p(i,j) = R_s^p(i,j) - P_s^p(i,j).$$

The $\Delta^c{}_f(i,j)$ represents the quantization error of the current frame in the first-stage encoding process, while the $\Delta_s^p(i,j)$ represents the quantization error of the previous frame in the second-stage encoding process. Therefore, the optimal motion vector in the transcoder (or the second-stage encoder) is correlated with the incoming motion vector and the quantization errors occurred in the first and the second-stage encoders.

(Ox, Oy) be the truly optimized motion vector when a full-scale full-search motion estimation is performed in the second-stage encoder. Then, *SAD$_s$(Ox,Oy)* will be the minimal value among all possible *SAD's*. Thus,

$$SAD_s(Ox,Oy) \le SAD_s(Ix,Iy)$$
$$\le SAD_f(Ix,Iy) + \sum_i \sum_j |\Delta^c{}_f(i,j) - \Delta_s^p(i+Ix,j+Iy)|$$
$$= SAD_f(Ix,Iy) + SDQE,$$

where $SDQE = \sum_i \sum_j |\Delta^c{}_f(i,j) - \Delta_s^p(i+Ix,j+Iy)|.$

The SDQE (Sum of Differential Quantization Error) will be used for the adaptive motion vector refinement scheme discussed in section 4. The above relation tells us that the re-use of the incoming motion vectors results in non-optimized, outgoing motion vectors due to the differential quantization errors.

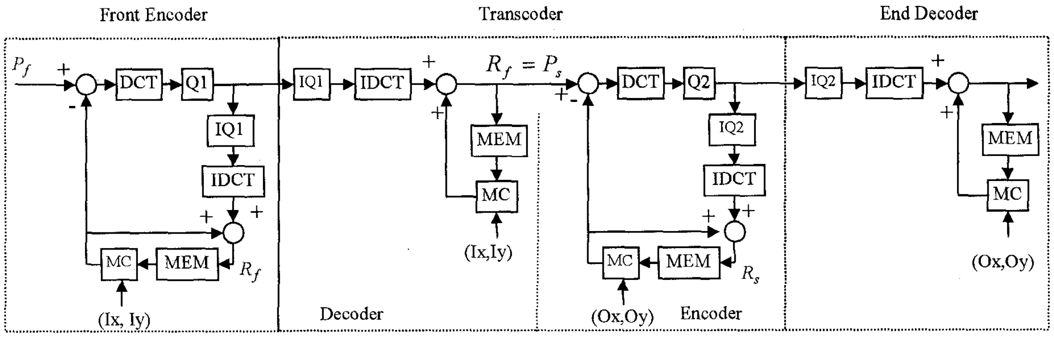


Figure 1. Structure of a Cascaded Transcoder

2.2 Re-use of the Incoming Motion Vector

The full-scale motion estimation for the transcoder described in the previous section requires a high computational complexity. To reduce the computational complexity, and since it is generally considered that using the incoming motion vectors may be as good as performing a new motion estimation, the re-use of the incoming motion vectors for the outgoing video stream has been widely accepted.

Let *(Ix, Iy)* be a motion vector extracted from the incoming video stream during transcoding and

When the down-scaling in the bit-rate is in a reasonably small range, the re-use of the incoming motion vectors may not cause significant quality degradation since the differential quantization errors can be relatively small. However, considerable differential quantization error can cause significant quality degradation. In low-bit rate video coding, such as H.263, our experimental results show that quality degradation is significant and accurate motion vectors are necessary to prevent severe quality degradation when the down-scaling in the bit-rate is large.

2.3 Re-use of the Incoming Motion Vector with Frame-Rate Conversion

For video applications over narrow-band networks, such as Public Switched Telephone Network (PSTN) and wireless networks, a high compression ratio for video coding is required to obtain a low-bit rate. However, the high compression ratio may result in an unacceptable quality when coding the video with the full frame-rate. For example, in a wireless network which normally has a less than 20 kbps bandwidth, the quality degradation due to the low bit-rate is significant with 25 or 30 frames per second. Frame-rate reduction is often used as an efficient scheme to allocate more bits to the remaining frames to maintain an acceptable quality. Frame-rate conversion is also needed when an end system only supports a lower frame-rate capability. In these cases, a transcoder in the gateway will perform a frame-rate conversion by dropping frames.

Figure 2 illustrates a situation when the frame-rate conversion occurs. Here, frames from *n-k* to *n-1* are dropped where *k* is the total number of dropped frames between two consecutive non-dropped frames.

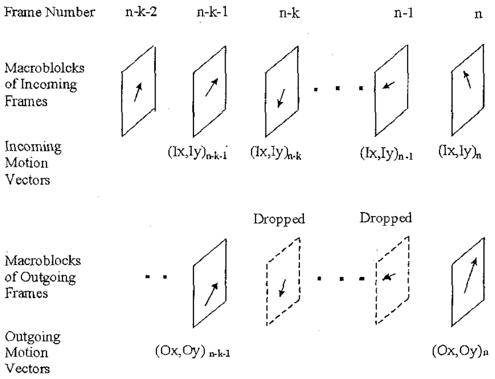


Figure 2. Motion Vectors with Frame-Rate Conversion

From the history of the incoming motion vectors of dropped frames and current frame, the outgoing motion vector $(Ox,Oy)_n$ can be obtained based on the *(n-k-1)-th* frame as the previous reconstructed reference frame. As shown in Figure 2, with the sequence of incoming motion vectors during frame-dropping, $\{(Ix,Iy)_{n-k},$ $(Ix,Iy)_{n-k+1}$, ..., $(Ix,Iy)_{n-1}\}$ and the motion vector $(Ix,Iy)_n$ extracted from the current frame, the outgoing motion vector for the *n-th* frame can be estimated as following:

$$(Ox,Oy)_n = \left(\sum_{d=1}^{k+1} (Ix)_{n-d+1}, \sum_{d=1}^{k+1} (Iy)_{n-d+1} \right).$$

However, this outgoing motion vector may not be optimized due to similar reasons as discussed in the previous section. Our simulation results show that optimal motion vectors are necessary to prevent severe quality degradation with the frame-rate conversion. In section 3, we propose a motion vector refinement scheme to obtain near optimal motion vectors.

3. Motion Vector Refinement

From the analysis in the previous section, we demonstrated that the differential quantization errors during transcoding may cause a perturbation in the position of the optimal motion vector. In most macroblocks, we can expect that the range of deviation will be small and the position of the optimal motion vector will be near that of the incoming motion vector. From these observations, we introduce a motion vector refinement scheme instead of re-using the incoming motion vector that results in quality degradation, or instead of applying the full-scale motion estimation that requires the most computational complexity in transcoding.

In this case, we define the base motion vector *(Bx, By)* as a motion vector obtained from the incoming video stream and the delta motion vector *(Dx, Dy)* as a difference vector between the base and the optimal motion vectors. With the delta motion vector, we can refine the base motion vector to the optimal motion vector. In other words, given the base and the delta motion vector, the optimal motion vector *(Ox, Oy)* is:

$$(Ox, Oy) = (Bx, By) + (Dx, Dy).$$

In situations where the frame-rate conversion is not involved during transcoding, the base motion vector is obtained by:

$(Bx, By) = (Ix, Iy)$.

In practice, the delta motion vector (Dx, Dy) can be estimated within a small search area S^R, given the base motion vector (Bx, By) by using the following equation:

$$(Dx, Dy) = \arg \min_{(m,n)\in S^R} SAD_R \,,$$

$$SAD_R = \sum_i \sum_j \left| P_s^c(i,j) - R_s^p(i + B_x + m, j + B_y + n) \right| .$$

In section 5, we will show that the new search area S^R can be set much smaller than the original full search area S to produce nearly the same quality as using the full-scale full-search motion estimation.

When the frame-rate conversion is performed during transcoding, it is also possible to apply the same concept of the motion vector refinement. If the k frames from n-k to n-1 are dropped during transcoding as described in section 2.3, the optimal outgoing motion vector for the n-th frame can be found by using the base and the delta motion vectors. The base motion vector is obtained by applying motion vector addition as described in the previous section:

$$(Bx, By)_n = \left(\sum_{d=1}^{k+1} (Ix)_{n-d+1}, \sum_{d=1}^{k+1} (Iy)_{n-d+1} \right).$$

The delta motion vector is estimated within a small search area around the base motion vector as in the case of non-frame dropping as:

$$(Dx, Dy) = \arg \min_{(m,n)\in S^D} SAD_D \,,$$

$$SAD_D = \sum_i \sum_j \left| P_s^n(i,j) - R_s^{n-k-1}(i + B_x + m, j + B_y + n) \right| .$$

where the previous reconstructed reference frame for (Dx, Dy) is set to the $(n$-k-$1)$-*th* frame which is the frame which comes before the first frame-dropping occurs. Our simulation results shows that the new search area S^D can be as small as S^R.

In comparison to the full-scale full-search motion estimation, the proposed motion vector refinement significantly reduces the computational complexity of motion estimation only by searching the delta motion vector within a much smaller search area. As shown in section 5, the performance of the motion vector refinement is close to that of the full-scale full-search motion estimation.

4. Adaptive Motion Vector Refinement

Based on the discussion in section 3, if the SDQE is small in comparison to the $SAD_f(Ix, Iy)$, it indicates that the quality degradation due to the re-use of incoming motion vectors is insignicant. Thus, it is possible to devise an adaptive motion vector refinement scheme based on the value of the SDQE. If the SDQE of the current macroblock is smaller than a threshold, the motion vector refinement may be skipped and the incoming motion vector can be used for the outgoing motion vector.

Since the mean quantization error of a uniformly distributed random variable with a quantization step-size q can be approximated by $\frac{q^2}{12}$, we can approximate the SDQE by :

$$SDQE = \left| \left(\frac{q_1^2}{q_2^2} - 1 \right) \right| \sum_i \sum_j \left| \Delta_s^p(i + Bx, j + By) \right| ,$$

where q_1 is the quantization step-size extracted from an incoming video bit-stream and q_2 is the quantization step-size used in the transcoder from when the previous frame was encoded. The complexity of the SDQE computation is about that of checking one search position in the motion estimation, so it does not require much new computation.

Using this technique, we implemented the adaptive algorithm for the motion vector refinement. The simulation results show that the computation for the motion estimation can be significantly reduced and kept minimal while achieving nearly same quality improvement as we can achieve by applying the motion vector refinements to all of the incoming macroblocks. Using this adaptive scheme, the percentage of the macroblocks that re-used the incoming motion vectors was about 65% as shown in Figure 6.

5. Simulation Results

In this section, we will explore some experimental results of the proposed scheme. In our experiment, a public domain H.263 software [7,8] was modified to implement the transcoder with the proposed motion estimation. Extensive simulation results demonstrated the effectiveness of the proposed motion vector refinement with and without the application of the frame-rate conversion during transcoding. The frame-rate conversion was performed by dropping one or two frames of the incoming video bit stream. One frame-dropping means that the outgoing video bit stream was transcoded using half of the incoming frame-rate. For example, an incoming frame-rate of 30 frames per second was transcoded into an outgoing frame rate of 15 frames per second.

In Figure 3, the video qualities obtained from the proposed schemes for outgoing motion vectors are compared to that of the full-search full-scale motion estimation when the frame-rate of the incoming video bit stream is preserved. In these simulations, the "carphone" test sequence was encoded at 128 kbps and then transcoded into 32 kbps. As shown in the Figure 3, the re-use of the incoming motion vectors can introduce about 0.8 dB quality degradation between frame number 20 to 80, for example. The simulation results of various test sequences are shown in Table 1. As indicated in Table 1, the quality achieved by applying the motion vector refinement schemes is almost identical to the quality achieved by using the full-scale full-search motion estimation. These results demonstrate that the proposed scheme is very effective because the search area for delta motion vectors was fixed to only ±2 integer pixels. This is significantly smaller than the computational complexity of the full-scale motion estimation.

From Figure 4, one can see the comparison in the video quality obtained by different schemes for outgoing motion vectors in which a one frame-dropping is applied to the "suzie" test sequence. The simulation results show that the quality achieved from application of the proposed motion vector refinement scheme is nearly identical to the quality achieved from the use of the full-scale full-search motion estimation. Table 2 shows the experimental results when the frame-rate conversion is applied to different test sequences. Similar to those cases without the frame-rate conversion, we encode the original test sequences at 128 kbps and then transcode them into several lower-rate sequences. In each one of these cases, the macroblock coding modes were re-computed. In a one-frame dropping, we used a fixed search area of ±2 integer pixels. In a two-frame dropping, a search area of ±4 integer pixels is applied. The simulation results show that the proposed motion estimation scheme also performs well with the frame-rate conversion.

In Figure 5, the performance of the proposed adaptive motion vector refinement is shown. The original test sequence 'foreman' is encoded at 128 kbps in the first-stage encoder, and transcoded into 32 kbps with half of the incoming frame-rate. In this instance, the incoming frame-rate is 30 frames per second and the transcoded frame-rate is now 15 frames per second. For both the motion vector refinement and the adaptive motion vector refinement, we utilized a search area of ±2 integer pixels. In comparison, the proposed adaptive motion vector refinement is similar in performance to the motion vector refinement. However, in the adaptive motion refinement case, the motion vector refinements were performed on only about 35 % (6937 from total 19701 macroblocks) of total number of the incoming macroblocks. This computational saving is significant and demonstrates the effectiveness of the proposed adaptive motion vector refinement scheme. Figure 6 shows the distribution of the number of macroblocks in which the motion vector refinements are carried out. In comparison of two graphs in Figure 5 and 6, the proposed motion refinement scheme performs superbly in tracking the relative PSNR degradation.

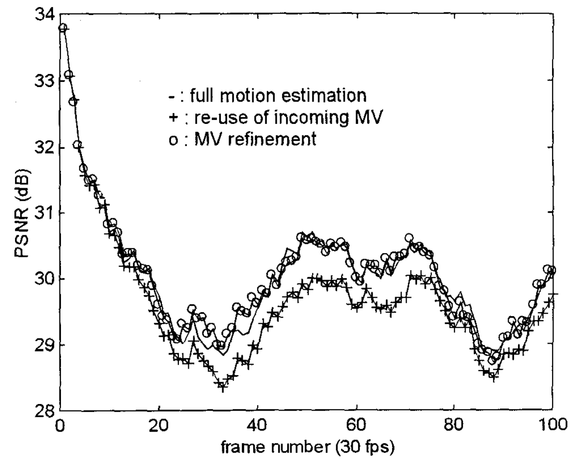
6. Conclusion

Traditionally, motion estimation was not considered in transcoding because of its computational complexity. Furthermore it was generally thought that using incoming motion vectors extracted from an incoming video stream would be almost as productive as performing a new motion estimation. However, this simple re-use scheme introduces significant quality degradation in many applications, including the situations in which the frame-rate conversion is needed.

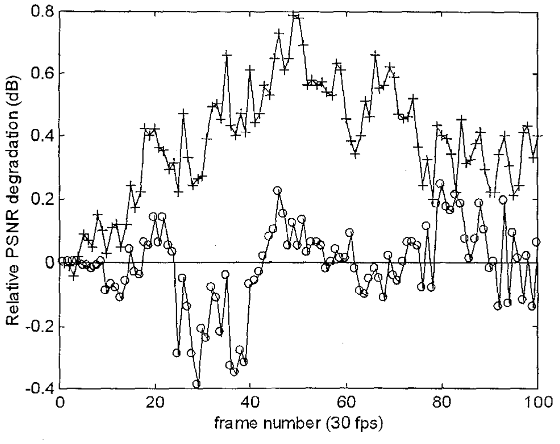
In this paper, several schemes for motion estimation in the transcoder are discussed. Based on the analysis of the quantization errors that cause the extracted motion vectors to be non-optimal, we presented a motion vector refinement scheme for high performance transcoding. With the motion vector refinement within a much reduced search area, it is possible to achieve fast motion estimation for near-optimal outgoing motion vectors with a quality close to the full-scale motion estimation. In addition, we also proposed an adaptive scheme based on the sum of the differential quantization errors to further reduce the computational complexity. Through extensive simulations, we have shown that the proposed scheme improves the video quality to such a degree that it rivals the application of a full-scale full-search motion estimation, with a minimal increase in computational complexity.

7. References

1. A. Eleftheriadis and D. Anastassiou, “Constrained and general dynamic rate shaping of compressed digital video,” ICIP ’95, pp396-399, 1995.
2. H. Sun, W. Kwok and J.W. Zdepski, “Architecture for MPEG compressed Bitstream Scaling”, IEEE Trans. on CSVT, vol. 6, No. 2, pp191-199, April 1996.
3. G. Keesman, et al., “Transcoding of MPEG bitstreams,” Signal Processing Image Comm., vol. 8, pp. 481-500, 1996.
4. P.N. Tudor and O. H. Werner, “Real-Time Transcoding of MPEG-2 Video Bit Streams”, Int. Broadcasting Convention, Amsterdam, Sept. pp 286-301, 1997.
5. P. Assuncao and M. Ghanbari, “Post-processing of MPEG2 coded video for transmission at lower bit rates,” ICASSP’96, vol. 4, pp. 1998-2001, May 1996.
6. P. Assuncao and M. Ghanbari, “Optimal Transcoding of Compressed Video,” ICIP’97, 1997.
7. ITU-T SG-15 Recommendation H.263, “Video Coding for Low Bit Rate Communications,” May 1996.
8. ITU-T SG-15, “Video Codec Test Model, TMN5”, January 1995.
9. N. Yeadon, F. Garcia, D. Hutchison. and D. Shepherd, "Filters: QoS Support Mechanisms for Mulipeer Communications", IEEE J. on Sel. Areas in Computing (JSAC), Vol. 14, No. 7, pp 1245-1262, Sept. 1996
10. J. Youn and M.T. Sun, "Motion Estimation for High Performance Transcoding," Int. Conference on Consumer Electronics, Los Angels, June 1998.



(a)



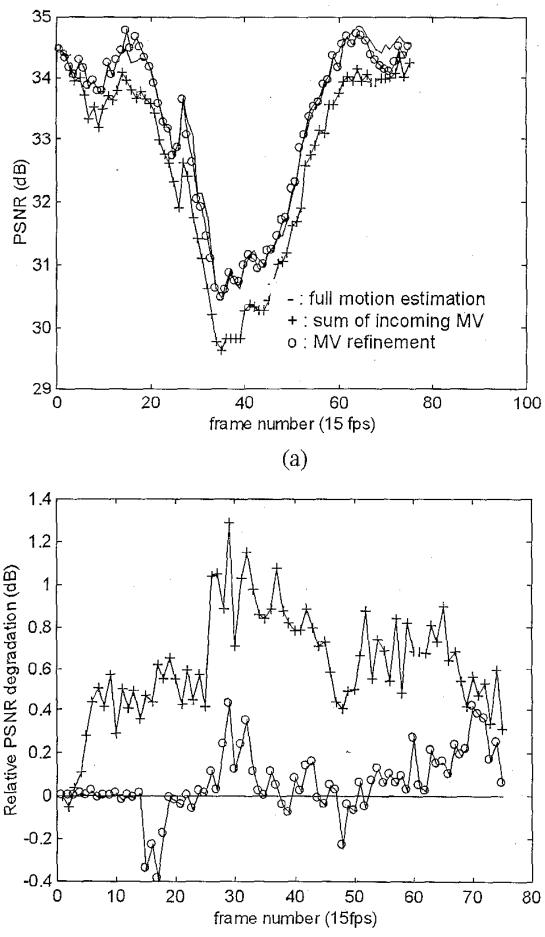
(b)

Figure 3. (a) Quality comparison of different motion estimation schemes without frame-rate conversion ("carphone" test sequence). Test sequence encoded at 128 kbps is transcoded into 32 kbps. A fixed search area for delta motion vectors (± 2 integer pixels) was used. (b) Relative PSNR degradation compared to the full-scale full-search motion estimation.

(Unit : dB)

Test Sequence		64 kbps	32 kbps	16 kbps
Trevor	A	31.81	29.72	28.36
	B	31.69 (-0.12)	29.23 (-0.49)	27.78 (-0.58)
	C	31.79 (-0.02)	29.63 (-0.09)	28.20 (-0.16)
Carphone	A	30.36	28.27	27.35
	B	30.26 (-0.1)	27.90 (-0.37)	26.81 (-0.54)
	C	30.35 (-0.01)	28.26 (-0.01)	27.29 (-0.06)
Claire	A	39.60	37.07	34.49
	B	39.50 (-0.1)	36.75 (-0.32)	33.63 (-0.86)
	C	39.55 (-0.05)	37.07 (0)	34.35 (-0.14)
Miss_am	A	39.65	37.84	35.54
	B	39.43 (-0.12)	37.54 (-0.30)	34.76 (-0.78)
	C	39.68 (+0.3)	37.83 (-0.01)	35.53 (-0.01)
Suzie	A	34.17	32.20	30.62
	B	34.15 (-0.02)	31.78 (-0.42)	29.91 (-0.71)
	C	34.18 (+0.01)	32.21 (+0.01)	30.57 (-0.05)

Table 1. Quality comparison without frame-rate conversion. Incoming video streams at 128 kbps are transcoded into different lower rates. In scheme A, a full-scale full-search motion estimation is used. In scheme B, motion vectors of the incoming video stream are used for the outgoing video stream. In scheme C, the proposed motion vector refinement scheme is applied. Numbers in () indicate the average PSNR degradation from the full-scale full-search motion estimation.



(a)

(b)

Figure 4. (a) Quality comparison of different motion estimation schemes with the frame-rate conversion ("suzie" test sequence). Test sequence encoded at 128 kbps is transcoded into 32 kbps with one-frame dropping. A fixed search area for delta motion vectors (± 2 integer pixels) is used. (b) Relative PSNR degradation compared to the full-scale full-search motion estimation.

(Unit : dB)

Test Sequence		64 kbps	32 kbps	16 kbps
Trevor	A	31.81	29.72	28.36
	B	31.69 (-0.12)	29.23 (-0.49)	27.78 (-0.58)
	C	31.79 (-0.02)	29.63 (-0.09)	28.20 (-0.16)
Carphone	A	30.36	28.27	27.35
	B	30.26 (-0.1)	27.90 (-0.37)	26.81 (-0.54)
	C	30.35 (-0.01)	28.26 (-0.01)	27.29 (-0.06)
Claire	A	39.60	37.07	34.49
	B	39.50 (-0.1)	36.75 (-0.32)	33.63 (-0.86)
	C	39.55 (-0.05)	37.07 (0)	34.35 (-0.14)
	A	39.65	37.84	35.54

Miss_am	B	39.43 (-0.12)	37.54 (-0.30)	34.76 (-0.78)
	C	39.68 (+0.3)	37.83 (-0.01)	35.53 (-0.01)
Suzie	A	34.17	32.20	30.62
	B	34.15 (-0.02)	31.78 (-0.42)	29.91 (-0.71)
	C	34.18 (+0.01)	32.21 (+0.01)	30.57 (-0.05)

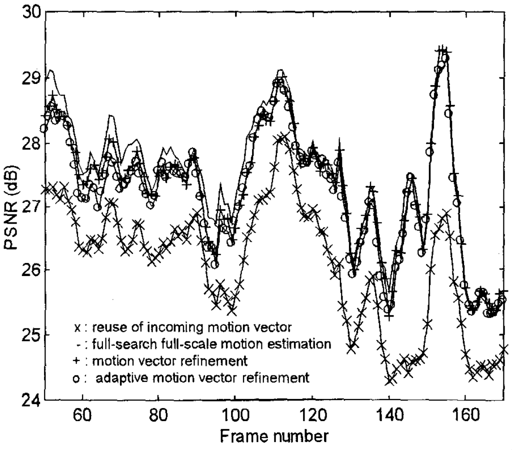
(a) One-frame dropping

(Unit : dB)

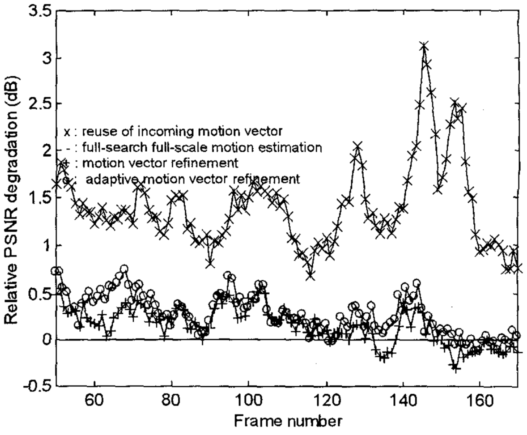
Test Sequence		64 kbps	32 kbps	16 kbps
Trevor	A	32.65	30.54	28.77
	B	32.14 (-0.51)	30.02 (-0.52)	27.97 (-0.8)
	C	31.59 (-0.06)	30.53 (-0.01)	28.71 (-0.06)
Carphone	A	31.39	29.36	27.77
	B	30.97 (-0.42)	28.64 (-0.72)	26.94 (-0.83)
	C	31.34 (-0.05)	29.32 (-0.04)	27.74 (-0.03)
Claire	A	40.41	38.13	35.66
	B	40.17 (-0.24)	37.62 (-0.51)	34.82 (-0.84)
	C	40.39 (-0.02)	38.06 (-0.07)	35.58 (-0.08)
Miss_am	A	40.27	38.56	36.74
	B	40.08 (-0.19)	38.18 (-0.38)	36.08 (0.66)
	C	40.25 (-0.02)	38.51 (-0.05)	36.76 (+0.02)
Suzie	A	34.88	33.25	31.32
	B	34.47 (-0.41)	32.63 (-0.62)	30.33 (-0.88)
	C	34.86 (-0.02)	33.20 (-0.05)	31.29 (-0.03)

(b) Two-frames dropping

Table 2. Quality comparison with the frame-rate conversion. Incoming video streams at 128 kbps are transcoded into different lower rates with the frame-rate conversion. In scheme A, a full-scale full-search motion estimation is used. In scheme B, the motion vectors for the outgoing video stream are the vector-sums of the motion vectors of the incoming video stream during the frame-rate conversion. In scheme C, the proposed motion vector refinement scheme is applied. Numbers in () indicate the average PSNR degradation from the full-scale full-search motion estimation.



(a)



(b)

Figure 5. Performance using the adaptive motion vector refinement. (a) Quality comparison of different motion estimation schemes with the frame-rate conversion ("foreman" test sequence). Test sequence encoded at 128 kbps is transcoded into 32 kbps with one-frame dropping. A fixed search area for delta motion vectors (± 2 integer pixels) is used. (b) Relative PSNR degradation compared to the full-scale full-search motion estimation.

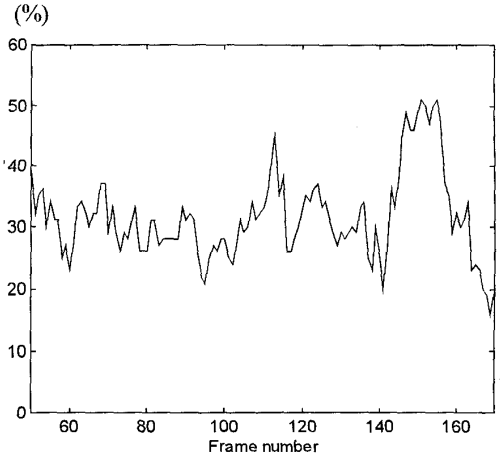


Figure 6. Percentage of computation of new motion vector using the proposed adaptive motion vector refinement scheme.

Jeongnam Youn received the B.S. degree in electronic engineering from Hanyang University, Korea in 1988, the M.S. degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea in 1990. Since 1996, Mr. Youn is working toward the Ph.D. degree in electrical engineering of University of Washington. From 1990 to 1995, he was a member of technical staff in the Korea Telecom. His research interests are video coding, video transcoding, and networked multimedia applications.

Chia-Wen Lin received the M.S. degree from Tsing Hua University, Taiwan in 1992. Since 1992, Mr. Lin has been joining the Computer and Communications Research Institute. He served as a project manager of SDH filter optics loop access systems in 1997, and is interested in videophone/video conference systems design.

Ming-Ting Sun received the B.S. degree from National Taiwan University in 1976, the M.S. degree from University of Texas at Arlington in 1981, and the Ph.D. degree from University of California, Los Angeles in 1985, all in electrical engineering. Dr. Sun joined the faculty of the University of Washington in September 1996. Before that, he was the Director of the Video Signal Processing Research Group at Bellcore. His research interests include video coding, multimedia networking, and VLSI architecture and implementation for real-time video signal processing. Dr. Sun has been awarded 6 patents and has published more than 80 technical papers including several book chapters in the area of video technology. He developed new VLSI architectures for several critical functions in video compression. He was actively involved in the development of H.261, MPEG-1, and MPEG-2 video coding and systems standards. He was the Editor of IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT) from 1995–1997, and the Express Letter Editor of T-CSVT from 1993 to 1994. He was a co-recipient of the T-CSVT Best Paper Award in 1993. From 1988 to 1991 he served as the Chairman of the IEEE CAS Standards Committee and established an IEEE Inverse Discrete Cosine Transform Standard. He received an Award of Excellence from Bellcore in 1987 for the work on Digital Subscriber Line. He is a Fellow of IEEE.