Systems and Computers in Japan, Vol. 33, No. 12, 2002
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J84-D-II, No. 6, June 2001, pp. 1084–1093

"Trampeg": An MPEG-2 Real-Time Software Transcoder for Low-Bit-Rate Transmission

Takashi Akutsu,[1] Hideaki Kimata,[1] Atsushi Shimizu,[2] Yoshiyuki Yashima,[1] and Naoki Kobayashi[1]

[1]NTT Cyber Space Laboratories, Yokosuka, 239-0847 Japan

[2]NTT East Corporation, Tokyo, 565-0456 Japan

SUMMARY

In the audiovisual handling of content coded by MPEG-2 transmitted over various kinds of networks such as the Internet or LAN, transformation of coding bit rates or transcoding is necessary. The simplest transcoding method is that of decoding MPEG-2 completely and then reencoding it at a lower bit rate. However, real-time processing of this method on a commonly used PC is difficult since the level of operations or the computational complexity for coding is high. However, if using the before-transformation coding information decreases the computational complexity, real-time transformation (transcoding) by software becomes possible. In this study, the Trampeg software transcoder (transcoder for MPEG-2) capable of real-time transcoding MPEG-2 with high quality has been developed. The applications, design guidelines, and processing algorithms of Trampeg are presented in this paper. The results of evaluation for operation of the system over a real PC have verified that an MPEG-2 system stream input via a network can be transcoded in real time and the transcoded low-rate stream can be transmitted directly to a streaming server. © 2002 Wiley Periodicals, Inc. Syst Comp Jpn, 33(12): 77–86, 2002; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/scj.10062

Key words: transcoder; MPEG-2; real-time transformation; motion vector; image coding.

1. Introduction

With digital broadcasting and dissemination of DVD, large amounts of content are expected to be managed by MPEG-2 itself. In transmitting the contents managed by MPEG-2 using various kinds of networks, it becomes necessary to transform coding bit rates, or to "transcode." Transforming to low bit rates is especially essential in audiovisualizing such content via the Internet or LAN.

The most basic method of transcoding is that of first decoding all frames completely in pixel regions, contracting the sizes of the decoded images depending on need, and then reencoding them at a lower bit rate. However, real-time processing of this method is difficult since the amount of operations or the computational complexity for coding increases enormously. In order to make this method processable in real time, methods for decreasing the computational complexity required for transformations [1–3] by utilizing motion vector information and the like contained in the before-transformation MPEG-2 stream have been proposed. The authors have developed the Trampeg software transcoder (transcoder for MPEG-2) which is capable of real-time transforming (transcoding) an MPEG-2 system stream to a high-quality stream of low bit rate (20 to 512 kbit/s) [4, 5] on a commonly used PC.

In this paper, the applications and requirements of the transcoder are clarified and the design guidelines for a transcoding algorithm satisfying the requirements are presented in Section 2. An algorithm for estimating motion vectors for low-rate use from motion vector information contained in an MPEG-2 stream is presented in Section 3. The motion vector estimation plays the greatest role in reducing the computational complexity of Trampeg. Next, the results of evaluating the performance of the motion vector transformation algorithm from the points of view of picture quality and the conversion speed on a real PC are presented in Section 4, and the results of evaluating the entire system including the audio part are presented in Section 5.

2. System Requirements and Basic Configuration of Software

2.1. System requirements

An example of the application of Trampeg is shown in Fig. 1. A client using an exclusive large-capacity line can directly receive, view, and hear a high-quality MPEG-2 content stream or an accumulated file coded by hardware. In this case, if, for a client using a lower-speed circuit such as the Internet, high-quality MPEG-2 content transmitted over a large-capacity exclusive line input into Trampeg over a network are transformed into a low-bit-rate stream in real time and are streamed, content the same as the MPEG-2 content before transformation can be simultaneously viewed and heard over the low-speed circuit. The following items are system requirements for the above.

i) The coding bit rate of an MPEG-2 stream (SP @ ML and MP @ ML) for a standard television signal must be less than 15 Mbit/s, by the specifications, and Ref. 6 points out that picture quality of the same degree as that of the current television signals is obtained at 4 Mbit/s. Based on this, 4 to 15 Mbit/s is targeted as the input bit rate. In addition, regarding bit rates after transformation, a transformation to 20 to 512 kbit/s is assumed possible, assuming a modem user (28.8 kbit/s) as low-bit-rate network and the most widely distributed ISDN (64 kbit/s, 128 kbit/s) and DSL (384 to 512 kbit/s) as digital circuits.

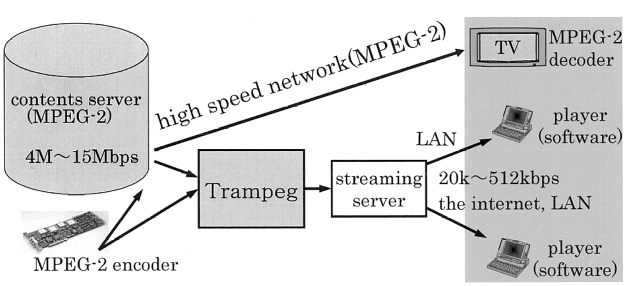


Fig. 1. An example of Trampeg application.

ii) For a constant bit rate, there is a trade-off relationship between the picture quality (including the screen size) and the frame rate and there is an optimal value depending on the contents [7, 8]. On this basis, the system has a configuration which allows a user to select whether to consider the frame number or the picture quality more importantly during transcoding. Regarding the screen size, QSIF (176 × 120, horizontal vertical 1/4) and SIF (352 × 240, horizontal vertical 1/2 of standard TV) used for streaming H.263 or MPEG-4 over the Internet are considered. Attainment of at least 5 to 6 frames/s with a bit rate of about 100 to 120 kbit/s, which is the ISDN band shown in i) is aimed at in software designing, since jerkiness is noticed with a frame number less than 5 frames/s according to Ref. 8.

iii) Real-time processing by software alone should be possible on a commonly used PC (Pentium II, approximately 500 MHz × 2 CPU) up to and in addition to transformations i) and ii), the transmission of the posttransformation low-rate stream to a streaming server (multiplexing or packeting).

iv) Not only should an MPEG-2 stream be input from a file, but also inputs via the network should be handled.

Among the above, i) and ii) are requirements regarding the input–output streams themselves, while iii) is a requirement regarding the interface of the software regarding an output low-bit-rate stream, iv) is a requirement for the interface of the software regarding an MPEG-2 input stream, and iii) and iv) are both functional requirements. By iii) and iv) the use of Trampeg transforming an MPEG-2 stream and network streaming using a streaming server are possible. In addition, for item iii), an easily obtainable 2CPU high-end machine is assumed here. However, no processing unique to 2CPU is specially performed, considering that resource distribution to two CPUs must be optimal under the OS management, as mentioned later, and that the PC should be commonly usable in constructing software.

The basic configuration of the algorithm of the transcoder regarding these requirements is considered in the next section.

2.2. Software basic configuration

Figure 2 shows the configuration of the software module of the transcoder based on complete decoding/reencoding (tandem connections). The computational complexity increases with tandem connections and real-time processing becomes impossible. Thus, we first consider which parts are problematic in terms of the computational

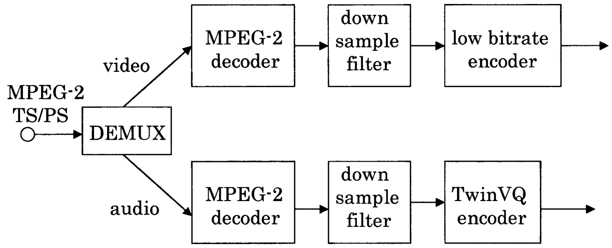


Fig. 2. Diagram of tandem transcoder.

complexity, considering high-speed processing for real-time processing.

In general, the computational complexity of the audio portion is lower than that of the video portion. While more than 400 MOPS is required combining the audio and video portions for complete decoding of MPEG-2 [9], the ratio of the amount of computation required to decode the audio portion is less than 1/10 according to Refs. 9 and 10. In addition, if adopting the Twin VQ scheme [11], which can ensure high quality even at low bit rates, is considered in audio reencoding, the CPU occupancy rate is fully 10% when 11-kHz stereo real-time coding is performed by a 500 MHz × 2 CPU PC. From the above, the amount of computation as a fraction of the entire transcoding process becomes low, even when the audio portion is completely decoded and then reencoded with subsampling. Thus, for audio, the scheme of complete decoding + subsampling + Twin VQ coding is used.

The remaining problem is how to reduce the computational complexity of the video portion. The estimated computational complexity of the video portion based on Refs. 9 and 12 is given in Table 1. The values shown are for the case of completely decoding MPEG-2 video, image contraction by subsampling by 1/2 horizontally and vertically, motion compensation (MC), discrete cosine transformation (DCT) of H.263 type coding and reencoding at SIF, 30 frames/s. Motion vector estimation during reencoding makes use of estimation based on layer searches, which involves an increase by a factor of about 16 in the case of full searches [12].

Table 1. Computational complexity of tandem transcoding from MPEG-2 video to lower-bit-rate stream (MOPS)

MPEG-2 decoder	200	VLD+IQ : 50
		IDCT : 30
		MC : 80
		Misc : 40
Filtering+subsampling	30	
Low-bit-rate encoder	730	ME : 500
		DCT, IDCT : 130
		Misc : 100
Video total	960	

According to Table 1, the ratios of IDCT, DCT, and motion estimation processing (ME) in low-bit-rate coding are high. Thus, the software is designed according to guidelines discussed below for Trampeg. The following two methods are considered in order to reduce the computational complexities of ME, DCT, and IDCT.

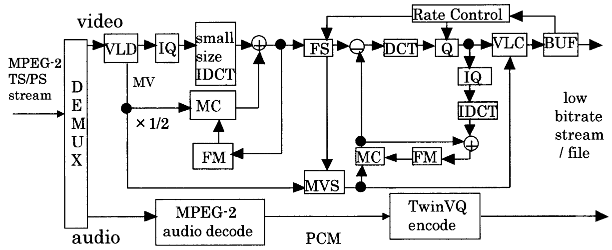
A. Method of decoding each frame completely, while using data of motion vectors etc. during reencoding.

B. Method of reencoding in the DCT domain using coded data input without DCT, IDCT, etc.

Method A can omit motion vector search processing etc., and there are a number of studies on this method [1–3]. In addition, method B is effective with low computational complexity when the stream structures before and after transcoding are the same as in bit rate transformations [15] of H.263, H.261, or bit rate transformations of MPEG-2 [13, 14]. However, when transforming coded data having the interlaced configuration of MPEG-2 into low bit streams of a progressive configuration accompanying size contraction and frame rate reduction, complicated computations are needed for operation in the DCT domain.

From such a point of view, an algorithm is designed on the basis of method A. In method A, the amount of computation of motion estimation processing inside the low-bit-rate encoder of Table 1 can be zero. Although an algorithm for estimating a motion vector to be used at low bit rates with high efficiency, using motion vectors before transformations, etc., is needed, in this study the problem is resolved by using an algorithm based on motion vector tracking [16, 17]. The algorithm of this portion is explained in detail in the following section. This algorithm can further reduce the amount of computation of the low-bit-rate encoder, since an MPEG-2 stream of 30 frames/s is suitable for transformation to a frame number of about 6 frames/s.

The MPEG-2 decoding processing load is high when method A is used, while the amount of computation is reduced by not requiring motion vector searches. Using the method of size contraction only in the DCT domain reduces the MPEG-2 decoding processing load. Specifically, the image size is contracted by half in the horizontal direction (and by half in both the horizontal and vertical directions in the case of the QSIF size) by the half-decoder scheme in which only the low-frequency components of the DCT coefficient data are reverse-transformed without decoding an MPEG-2 stream completely. Thus, only a partial field portion in the decoding result is used in reencoding. This can nearly halve the amounts of operations of IDCT and MC of MPEG-2 decoding of Table 1.



FS	:	frame selection to be transcoded
MVS	:	motion vector scaling
VLD	:	variable length decoding
VLC	:	variable length coding
IQ	:	inverse quantization
Q	:	quantization
IDCT	:	inverse discrete cosine transform
DCT	:	discrete cosine transform
MC	:	motion compensation
FM	:	frame memory
BUF	:	buffer
MV	:	motion vectors

Fig. 3. Configuration of Trampeg.

In addition, processing by decoding only I, P frame images in the case of MP @ ML including a B frame of the transformation target, with a rate of less than 10 frames/s, which is the target of real-time transformation to the SIF size, is effective. For example, transforming streams of M = 3 (including two B frames between the I, P frames), which is used frequently as MP @ ML in equal intervals such as 10 or 5 frames/s, is possible by using only I, P frame decoded images. If reencoded target frames become B frames at a frame rate other than the above, the decoding processing of B frames can be reduced by substituting the closest I, P frames before and after the above frames. From the point of view of realizing high-speed transcoding, the scheme of substituting I, P frames is adopted in this study.

Based on the above, the basic configuration of Trampeg is shown in Fig. 3. Trampeg inputs system streams of MPEG-2 (TS or PS), and transforms the video data to a low-rate video stream of MC–DCT base and the audio data to Twin VQ. Here, low-rate video streams are processed with high efficiency by incorporating global motion compensation [18], optimal coding allocation of motion vectors [19], motion-adaptive frame rate control [20], etc.

3. Motion Vector Transformation Algorithm

An algorithm for calculating motion vectors which allows omission of motion searches etc. while allowing high-efficiency compression is obtained by combining the following features i) to iv) and using an algorithm based on the motion vector tracking [16, 17] discussed in the preceding section (Fig. 4):

i) size contraction and vector length transformation (MV scale)

ii) IP transcoding transforming an I frame to a P frame (IP trans) [21]

iii) motion vector tracking and P frame thinning (P drop) [16, 17]

iv) integration of vectors calculated for each contracted block unit

The algorithm is explained in order below and an example of transforming to the SIF size is explained.

3.1. Image size contraction and vector length transformation

First, from the 720 × 480 coding information of the interlace structure of the before-transformation MPEG-2, the SIF size is obtained by contracting the horizontal direction by half by a half-decode using only the horizontal-direction low-frequency component of the DCT coefficient, and contracting also the vertical direction by half by extracting only the second field of the decoded image.

At this time, the motion vector length of MPEG-2 is transformed as shown in Fig. 5, corresponding to the size contraction. The length transformation coefficient is determined by the space scaling coefficient α and the time scaling coefficient β. There are three types of motion vector of MPEG-2: frame MC, field MC (from same parity), field MC (from different parity), and as shown in Table 2, the space scaling coefficient α, determined by the image size, becomes 1/2 for all cases. In addition, the time scaling coefficient β becomes 2/3 only for the case of field MC

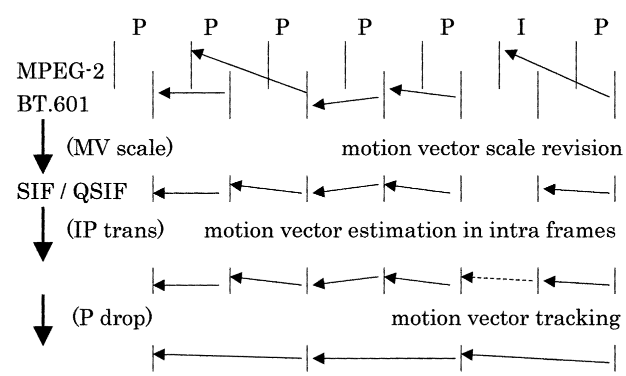


Fig. 4. Process outline of motion vector translation.

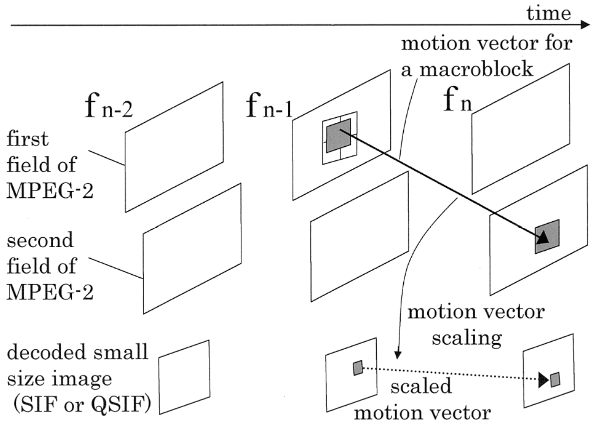


Fig. 5. Method of motion vector scaling.

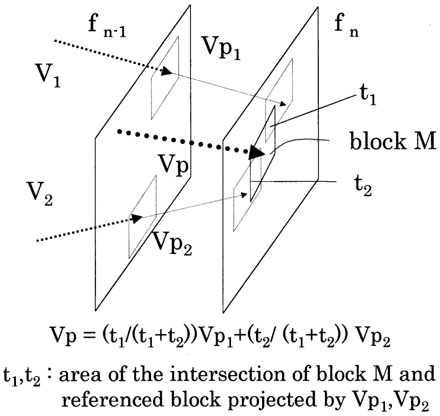


Fig. 6. Method of motion vector estimation in intraframes (1).

(from different parity), since only the second field is extracted. The final length transformation coefficient is given by the product of α and β. In the contracted decoded image, a macro block of 16 × 16 pixel size of MPEG-2 becomes a block of 8 × 8 pixel size, and the length transformed vector corresponds to each of these blocks. These vectors are used in the following procedures.

3.2. Motion vector estimation for an I frame (IP transcoding processing)

Although MPEG-2 is commonly coded in a form in which an I frame is inserted once each 0.5 to 1 s, the I frame interval is frequently widened from the point of view of coding efficiency at rates lower than 100 kbit/s. Thus, an I frame must be transformed to a P frame in the picture type. Since there is no motion vector information in an I frame, this is estimated by using the motion vectors of the P frames before and after it in IP transformation (transcoding) processing. This processing becomes necessary when an I frame is contained between a reencoded frame and the reference frame in an MPEG-2 stream that has been transformed.

As shown in Fig. 6, let the I frame to be IP-transformed be f_n and let the P frame immediately before this be f_{n-1}. Then, the blocks on f_{n-1} are projected onto f_n using motion vectors (V_{Pi}) that are the same as the motion vectors of this block. For block M on f_n, all motion vectors V_{Pi} (i = 1 to k_p) having common parts with the blocks projected from f_{n-1} are extracted. Next, the average motion vector V_P is calculated by weighting the projected motion vectors V_{Pi}, corresponding to the ratio between block M and blocks projecting V_{Pi}. Figure 6 shows an example of $k_p = 2$.

In addition, from the immediately following P frame f_{n+1}, the motion vector of each block is projected onto f_n as shown in Fig. 7. Then, for block M, all motion vectors V_{Si} (i = 1 to k_s) having common parts with the blocks projected from f_{n+1} are extracted, and the average motion vector V_S is obtained similarly to the case of V_P.

Table 2. Scaling coefficients of motion vectors

MC type	scaling coefficients α	β	$\alpha \times \beta$
frame MC	1/2	1	1/2
field MC (from same parity)	1/2	1	1/2
field MC (from different parity)	1/2	2/3	1/3

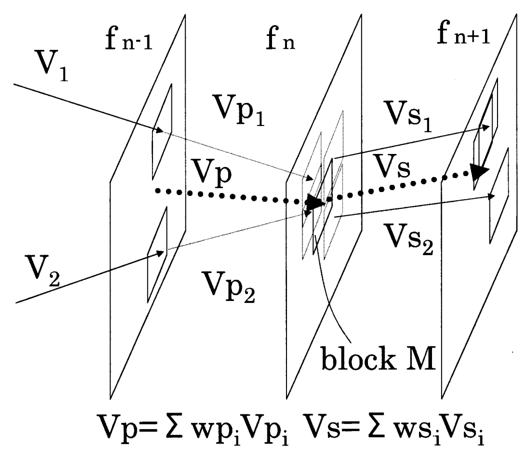


Fig. 7. Method of motion vector estimation in intraframes (2).

Finally, whether the directions of the calculated motion vectors V_P and V_S are uniform is determined. If the directions are uniform, or when

$$|\, Vp - Vs \,| < T \quad (1)$$

is satisfied for a certain value T, block M is determined to be an interblock and a motion vector is calculated (Fig. 7). The final motion vector for M is calculated by averaging V_P and V_S. If Eq. (1) does not hold, processing as an intrablock is done.

3.3. P frame thinning procedure (motion vector tracking)

A need to reduce the number of frames to be reencoded arises in order to maintain the picture quality even at a low rate. As shown in Fig. 8, the case of reencoding in a form in which P frames are referred to by skipping one frame is considered. Although motion vectors must be obtained from new reference frames when P frames are thinned, this processing is realized by tracking motion vectors used in Sections 3.1 and 3.2.

The motion vector of a small block of the reencoding target frame f_n is calculated as

$$V = \sum wp_i Vp_i + Vs \quad (2)$$

[17]. Here, the reference vectors to f_{n-1} are denoted by V_S, vectors to f_{n-2} from four blocks overlapping the area of the reference vectors Vs to the skipped frame are denoted by Vp_1, Vp_2, Vp_3, and Vp_4, respectively. In addition, wp_i is a coefficient expressing the ratio of the common parts between each block and the referred destination of V_S as shown in Fig. 9. In weighting in accordance with the area referred to by Eq. (2), wp_i for a block corresponding to an intrablock of the corresponding MPEG-2 is set to 0, and reencoding with intrablocks is performed without calculating the vectors of small blocks of the reencoding target frame f_n when the percentage of intrablocks is greater than half. The above operations are repeated if more than two frames are skipped.

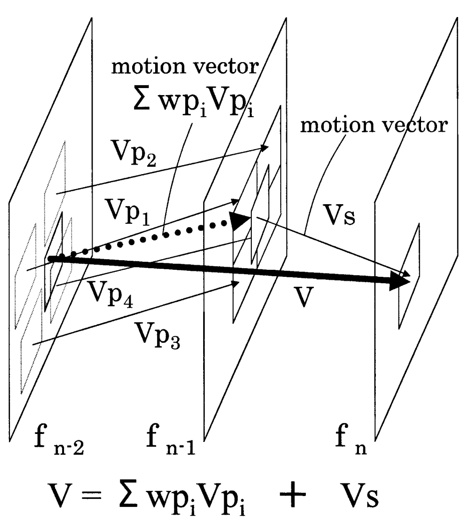


Fig. 8. Method of motion vector tracking (1).

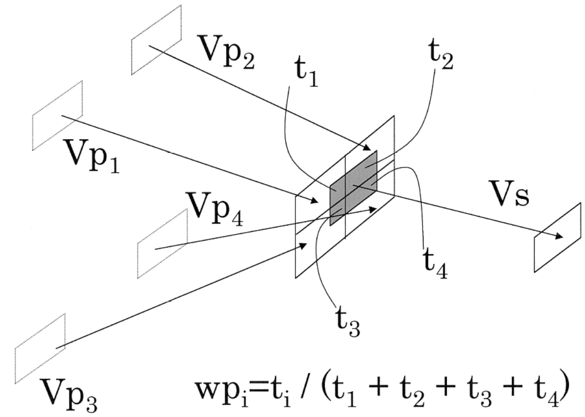


Fig. 9. Method of motion vector tracking (2).

3.4. Calculated vector integrating processing

A vector is calculated for each small block of 8 × 8 pixels on a contracted image by the processing described up to the preceding section. Since motion compensation is done for each block of 16 × 16 pixels during reencoding, the vectors of the calculated 8 × 8 pixel blocks are averaged for four vectors for each block unit to be used for reencoding. If more than three among the integrated 8 × 8 pixel small blocks do not have vectors as a result of being determined to be intrablocks, 16 × 16 pixel blocks are determined to be intrablocks. In addition, for the 16 × 16 pixel blocks in which fewer than two 8 × 8 pixel small blocks determined to be intrablocks, the average of only the 8 × 8 pixel small blocks for which vectors have been calculated is obtained and used for reencoding as a postintegration vector.

4. Performance Evaluation of Motion Vector Transformation Algorithm

The following experiments have been conducted to evaluate the effects of the motion vector transforming algorithm described in Section 3. First, 120 frames (4-second portions) of the standard Flower Garden and Mobile and Calendar images were coded into MPEG-2 video streams (SP @ ML, $N = 15$, 8 Mbit/s). The search range of motion vectors was ±63 in half-pixel units in both the horizontal and vertical directions, using the TM5 scheme [22]. This MPEG-2 stream was transcoded by methods 1 to 5 described below. The transcoding parameters include the image size SIF, a bit rate of 92 kbit/s, and a frame rate of 6 frames/s. Figures 10 and 11 compare the SNR of the decoded images obtained by methods 1 to 5, with the partial field images decoded after size contraction of the MPEG-2 stream (i.e., low-bit-rate encoder input image, data after FS of Fig. 3) as references.

[Method 1] Encoding by motion vectors obtained by full searches in half-pixel units during reencoding

[Method 2] Encoding by motion vectors obtained by full searches in integer pixel units during reencoding

[Method 3] Encoding using motion vectors calculated by the Trampeg transformation algorithm described in Section 3

[Method 4] Encoding without using IP transformations in Trampeg transformations, using adaptive prediction of [intra/inter (0 vector)] of portions for which motion vectors cannot be calculated

[Method 5] Encoding by adaptive prediction only of [intra/inter (0 vector)], without motion vector transformation during reencoding for all frames

The following two points are understood from Figs. 10 and 11:

(1) The difference between the case of full searches and Trampeg during reencoding is about 1 to 2 dB.

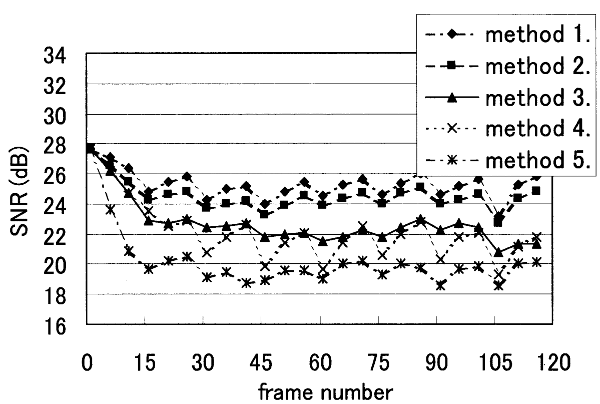


Fig. 10. SNR of transcoded images in each method (Flower Garden).

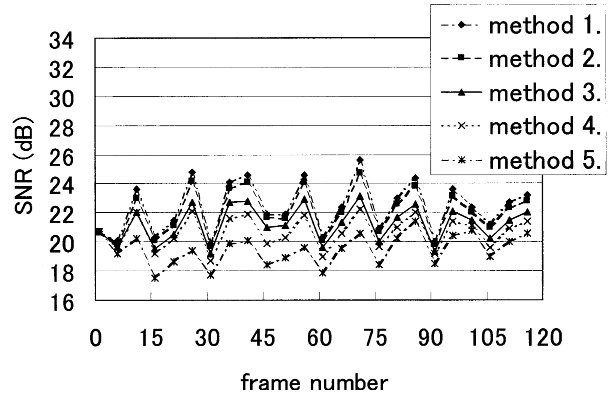


Fig. 11. SNR of transcoded images in each method (Mobile and Calendar).

(2) There is a picture quality improvement of 1 to 2 dB by the introduction of IP transformation.

In addition, the SNR tends to decrease every 15 frames in Figs. 10 and 11. The reasons for this tendency are given below. While an MPEG-2 decoded image (= low-bit-rate encoder input image) is encoded for every 15 frames as an I frame, an I frame is encoded with higher quality than a P frame or a B frame by rate control beginning with TM5 [22], since the property of an I frame is different from those of P and B frames. Since an I frame at intervals of 15 frames is encoded as a P frame by a low-bit-rate encoder, the I frame is predicted from the local decoder image of the P frame preceding in time, while the coding efficiency decreases as a result of the prediction from a P frame different in properties. This is because control is performed by assigning the same amount of information to each P frame by the low-rate encoder. Although visual degradation due to this degradation in SNR has not been observed in this study, the method of allocating information to a frame by IP transformation must be optimized in the future.

Table 3 shows the time required for transcoding by methods 1 to 5. A PC consisting of a 450-MHz dual-CPU

Table 3. Processing time for transcoding

transition rate	460 kbit/s, 10 fr/s	92 kbit/s, 6 fr/s
method 1	1761	1043
method 2	361	214
method 3	37	34
method 4	37	34
method 5	36	34

Pentium II is used for measurements. While real-time processing is impossible due to computational complexities in methods 1 and 2, real-time processing is possible by methods 3 to 5. Since full searches are used for estimating motion in methods 1 and 2, the ratio of the amount of processing of motion estimation to the entire amount of transcoder processing is about 95% [12]. Since motion estimation is performed between frames targeted for encoding, the amount of processing is approximately proportional to the number of frames. On the other hand, since the variable length coding processing level is influenced by the bit rate after transformation, but its ratio is small with respect to the entire processing, the processing time for the case of 460 kbit/s, 10 fr/s and the case of 92 kbit/s, 6 fr/s depends significantly on the frame rates after transformation.

On the other hand, since motion estimation is not done in methods 3, 4, and 5, the processing of the entire transcoder does not depend on the bit rate or frame rate after transformation, and the relative amount of processing such as MPEG-2 decoding increases, while DCT, IDCT, etc. are influenced by the frame rate or the bit rate after transformation, and the effects on the entire processing amount are small, amounting to about 10%.

In addition, it can be understood that the motion vector transformation processing proposed in Section 3 that is used by methods 3 and 4 requires very little time with respect to the entire transcoding time by virtue of the fact that the levels of processing of methods 3, 4, and 5 are almost unchanged for the same bit rate and the same frame rate, and that method 5 does not perform vector estimation at all.

The results of Figs. 10 and 11 and Table 3 show that the transcoding algorithm of method 3 used by Trampeg is slightly lower in picture quality than the case of performing full searches in half-pixel units or integer pixel units during reencoding, and that it performs best under conditions that allow real-time processing.

5. Software Construction and Performance Evaluation

On the basis of the above discussion, Trampeg software to be operated on a real PC has been constructed. The software specifications are shown in Table 4. In order to evaluate the real-time properties of the Trampeg system, the software was operated on a real PC and its processing time is measured. A PC of Pentium II Dual CPU of 450 MHz was used for measurements. In addition, the software was divided into multiple processing units, including the MPEG-2 system DEMUX portion, MPEG-2 video decoder portion, low-rate video encoding portion, MPEG-2 audio decoder portion, audio encoder portion, etc. Figure 12 shows the results of measuring the transcoding time when transcoding MPEG-2 system streams of MP @ ML with input bit rates of 5.5, 12.0 Mbit/s and SP @ ML with rates of 4.3, 6.0, and 10.8 Mbits into video at 92 kbit/s and audio at 20 kbit/s. The vertical axis shows the frame rate after transcoding and the horizontal axis shows the ratio of the time required for Trampeg processing to the length of the transcoded MPEG-2 stream. Real-time processing becomes possible at less than 100%. It can be seen from this that high-quality real-time transcoding of inputs of SP@ ML 6.0 Mbit/s to SIF, 92 kbit/s, 6 frames/s is possible. In addition, since processing of B frames can be omitted in the case of MP @ ML, the processing speed is faster than the case of SP @ ML, and real-time processing is possible even with MPEG-2 of 12 Mbit/s. In addition, it is confirmed that by real-time transcoding of network-input MPEG-2 system

Table 4. Specifications of Trampeg

input (file / stream from network)	MPEG-2 TS/PS	
	video	SP@ML, MP@ML, frame structure, M ≦ 3
	audeo	MPEG-2 Layer1, 2, MPEG-1 Layer2
decoding method	video	half decode
	audio	full decode
output (file / stream to network)	video	size：SIF(352 × 240), QSIF(176 × 112)
		frame rate：2～15 frame/s
		bitrate：20 kbit/s～512 kbit/s
	audio	format：TwinVQ
		bitrate：8 kbit/s～80 kbit/s

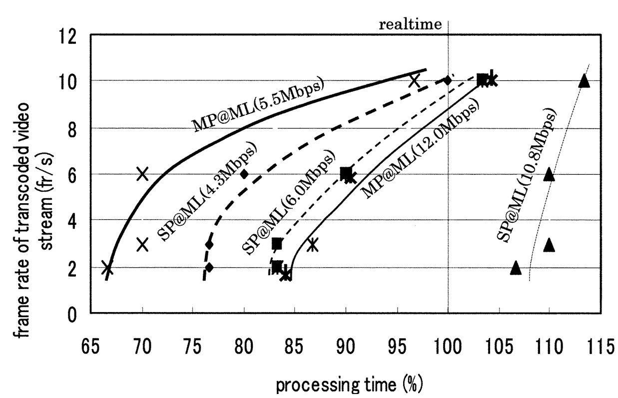


Fig. 12. Processing time of Trampeg for MPEG-2 system stream (SP @ ML and MP @ ML).

streams (TS) by Trampeg, network streaming can be achieved by sending video portion streams transcoded into SIF, 92 kbit/s, 6 frames/s, and audio portion streams transcoded into 20 kbit/s directly to a server.

6. Conclusions

The Trampeg software transcoder, capable of real-time transformation of an MPEG-2 stream to a low-bit-rate stream, has been developed. In this paper, the applications and requirements of the transcoder have been clarified and the design guidelines for the transcoding algorithm have been presented. In addition, an algorithm for estimating motion vectors during reencoding using the motion vector information of MPEG-2, which has a significant effect in reducing computational complexity among the transforming algorithms, is explained in detail. In addition, the Trampeg algorithm of the video portion and the entire system including the audio portion are evaluated on a real PC. It is confirmed from the evaluation results that real-time processing which allows maintenance of picture quality and high-speed processing is realized by a Trampeg algorithm which combines adaptive scaling of motion vectors in the MPEG-2 stream and downsampling in the DCT domain. Using this algorithm, MPEG-2 content managed by one source can be simply seen and heard via a LAN or the narrowband Internet, and a preview system etc. in VOD or simultaneous streaming broadband and narrowband systems can be easily constructed. The transcoding algorithm presented in Section 3 can also be applied to transcoding in H.263 or MPEG-4 etc. or coding based on DCT and motion compensation.

Expansion of the proposed system to MPEG-2 streams for HDTV, which is likely to be widely disseminated in the near future, and addition of a high-frame-rate transmission function using B frame decoded images etc., are being considered for future studies.

Acknowledgments. The authors are deeply indebted to Project Manager M. Hase and Director K. Kogure of NTT Cyberspace Laboratories for the opportunity to conduct this study. They also extend their gratitude to various persons, including Lead Technical Member S. Ishibashi and Professor H. Watanabe at Waseda University for daily guidance and helpful suggestions.

REFERENCES

1. Wee SJ, Apostolopoulos JG, Feamster N. Field-to-frame transcoding with spatial and temporal downsampling. ICIP 1999;4:271–275.
2. Shen B, Sethi IK, Bhaskaran V. Adaptive motion vector resampling for compressed video downscaling. ICIP 1997;1:771–774.
3. Youn J, Sun MT, Lin CW. Motion vector refinement for high-performance transcoding. IEEE Trans Multimedia 1999;1:30–40.
4. Akutsu T, Kimata H, Shimizu A, Yashima Y, Kobayashi N. Development of low bit rate MPEG-2 realtime software transcoder ‘Trampeg.’ Johogaku Kenpo 2000;AVM29-5:25–30.
5. Tsuji H, Akutsu T, Shimizu A, Yashima Y, Kobayashi N. Development and performance evaluation of low rate MPEG-2 transcoder ‘Trampeg.’ Tech Rep IEICE 2000;IE17-2.
6. Fujiwara H. Textbook on most recent MPEG. ASCII Publishers; 1994.
7. Takishima Y, Wada H, Murakami M. A study on the optimal frame rate in low rate dynamic image coding. Tech Rep IEICE IE56-7, 1991.
8. Hayashi T, Iai T. Effects of changes in frame rate on dynamic image subjective quality. NTT R&D 1997;46:67–74.
9. Ikekawa M, Ishii D, Murata E, Numata K, Takamizawa Y, Tanaka M. A real-time software MPEG-2 decoder for multimedia PCs. ICCE, WAM1.1, p 2–3, 1997.
10. Tsai TH, Chen LG, Huang SC, Chang HC. A low-cost architecture design with efficient data arrangement and memory configuration for MPEG-2 audio decoder. ISCAS 1998;4:65–68.
11. Moriya T, Iwakami N, Jin A, Ikeda K, Miki S. A design of transform coder for both speech and audio signals at bit/sample. ICASSP 1997;II:1371–1374.
12. Osako F, Yashima Y, Kodera H, Watanabe H, Shimamura K. Software image coding by a dynamic computation complexity scalable algorithm. Trans IEICE 1997;J80-D-II:444–458.
13. Kasai H, Tominaga H, Hanamura T, Kameyama S. A scheme for controlling the coding level of a low delay MPEG-2 video transcoder. Trans IEICE 2000;J83-B:151–164.
14. Assuncao PAA, Ghanbari M. A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams. IEEE Trans Circuits Syst Video Technol 1998;8:953–967.
15. Zhu W, Yang KH, Beacken MJ. CIF-to-QCIF video bitstream down-conversion in the DCT domain. Bell Labs Tech J p 21–29, July–Sept 1998.
16. Shimizu A, Sagata T, Watanabe H, Kodera H. Bit stream scaling by frame skipping. Symposium on Image Coding, PCSJ’96, p 25–26.
17. Shimizu A, Sagata T, Watanabe H, Kobayashi N. Bit stream scaling by changing reference frames. Trans IEICE 2000;J83-D-II:967–976.

18. Sagata T, Shimizu A, Watanabe H, Kodera H. Improving the efficiency of video coding considering internet video transmission. 1997 Shingaku So Tai (Information Systems 2), D-11-87.
19. Shimizu A, Sagata T, Watanabe H, Kobayashi N. A study on a coding method based on motion vector norm and direction component expressions. PCSJ'97, p 67–68.
20. Shimizu A, Sagata T, Kamikura H, Kobayashi N. Improving coding performance by reducing quantized parameter information. 1998 So Tai (Information Systems 2), D-11-6.
21. Akutsu T, Shimizu A, Yashima Y, Kobayashi N. A study on MPEG-2 transcoding. Documents of the 27th Annual Meeting of the Image Electronics Society, p 29–30, 1999.
22. ISO-IEC JTC1, MPEG 93/N225b. Test model 5/draft, 1993.

AUTHORS (from left to right)





Takashi Akutsu (member) received his B.S. degree from the Department of Mathematics of the University of Tokyo in 1992 and his M.S. degree in 1994. He then joined NTT, where his research areas have included robot vision and image coding. He received a 1998 IEICE Science Promotional Award. He is currently a leader of the Media Communications Project, NTT Cyber Space Laboratories. He is a member of the Information Processing Society and the Japan Robotics Society.

Hideaki Kimata received his B.S. and M.S. degrees from the Department of Applied Physics of Nagoya University in 1993 and 1995 and joined NTT. His research areas include image coding and error immunity. He is a member of the Video Information Media Society.

Atsushi Shimizu (member) received his B.S. and M.S. degrees from the Department of Electrical Engineering of Nihon University in 1990 and 1992 and joined NTT. His research areas include image coding. He received a 1997 Image Electronics Society Technology Award and a 1998 IEICE Science Promotional Award. He is currently a supervisor at NTT East Corporate Business Headquarters. He is a member of the Video Information Media Society and the Image Electronics Society.

Yoshiyuki Yashima (member) received his B.S. and M.S. degrees from the Department of Electronic Engineering of Nagoya University in 1981 and 1983 and joined NTT. He has conducted research in the areas of high-efficiency coding of image signals and dynamic image processing. He is currently a lead technical member of the Media Communications Project, NTT Cyber Space Laboratories. He holds a D.Eng. degree, and is a member of IEEE, the Video Image Information Media Society, and the Information Processing Society.

Naoki Kobayashi (member) received his B.S. and M.S. degrees from the Department of Applied Physics of Tokyo Institute of Technology in 1979 and 1981 and joined NTT. He has conducted research in the areas of facsimile, image inputting, and multimedia systems, and is currently engaged in research on image coding at the NTT Cyber Space Laboratories. He is group leader of the Media Communications Project Image Coding Technology. He holds a D.Eng. degree. He was chairman of the TTC Fifth Division Second Special Committee.