Fast Motion Estimation Algorithm for MPEG-4 to H.264 Transcoder

Jong-Ki Han* *Lifelong Member*, Chan-Won Seo* *Associate Member*

ABSTRACT

In this paper, we propose a fast ME (motion estimation) algorithm for MPEG-4 to H.264 Transcoder. Whereas 2 modes (8x8, 16x16) are used for ME in MPEG-4 simple profile, ME using 7 modes is supported for further enhanced coding efficiency in H.264. The transcoding speed is affected dominantly by the computational complexity of encoder part in transcoder, where ME module of H.264 encoder has high complexity due to using 7 modes. In order to increase the speed of transcoding between MPEG-4 and H.264, we use 3 PMVs (predicted motion vectors) and the mode information of MBs (macroblocks) provided from the decoder part of transcoder. Since the proposed 3 PMVs are very close to an optimal motion vector, and we consider only some restricted modes according to the MB information transferred from decoder part, the proposed scheme can speed up the transcoding procedure without loss of image quality.

We show experimental results which demonstrate the effectiveness of the proposed algorithm, where performance of our scheme is compared with that of the conventional fast algorithm for H.264.

Key Words : Transcoder, MPEG-4, H.264

Ⅰ. INTRODUCTION

In the circumstance of using various multimedia applications and networks whose bandwidths are diverse, video sequences need to be encoded by various encoders which are operating base on various target bit rates, frame rates, and coding standards. Thus, interoperability of multimedia content between different applications and between different bandwidth networks is becoming more important. Transcoding is one of key technologies to convert the video sequence encoded by a particular encoder into those encoded with different target bit rates or frame rates, and spatial resolutions. Video transcoder enables multimedia data to be transferred over heterogeneous applications or transmission channels.

The transcoding speed is affected dominantly by the complexity of ME module in transcoder. To reduce the complexity of transcoder, several efficient algorithms for ME have been proposed [1]-[7]. These algorithms are used for homogeneous video format conversion such as MPEG-2 to MPEG-2[3], H.263 to H.263[4][5], H.264 to H.264 [6][7]. In the homogeneous transcoding, since motion vectors and mode information decoded in the decoder part of transcoder can be used in the encoder part of transcoder, the complexity of system can be reduced significantly. For more adaptive video format conversation, techniques for heterogeneous video transcoding such as MPEG-2 to H.263[8], H.261 to H.263[8], and MPEG-1 to MPEG-4[9] have been proposed. In this case, in contrast to homogeneous transcoding, since syntaxes about motion vector and mode information among heterogeneous coding standards are different with each other, it is difficult to use the decoded MV and mode information in the encoder

※ This work was supported by the faculty research fund of Sejong University in 2007.
* Dept. of Information and Communications Eng., Sejong Univ., Seoul, Korea (hjk@sejong.ac.kr, sachonwon@teramail.com)
논문번호 : KICS2008-03-134, 접수일자 : 2008년 3월 19일, 최종논문접수일자 : 2008년 6월 5일

part of transcoder. Thus, the heterogeneous transcoder requires a heavy processing burden. Various transcoding techniques[8][9] had been proposed to solve this problem.

The purpose of this paper is to derive the fast algorithm for ME module in transcoder between MPEG-4 and H.264. Since ME module of H.264 encoder considers 7 modes, complexity of the transcoder is very high. To reduce the complexity, we use 3 PMVs and consider some limited modes based on the mode information decoded in the decoder part of transcoder. Since the 3 PMVs are very close to optimal motion vector, the proposed scheme can speed up transcoding process. Using the restricted modes increases the speed of transcoder further.

This paper is organized as follows. Section II describes the ME schemes adapted in MPEG-4 and H.264 standard. In Section III, we propose an efficient ME scheme for transcoder. And computational complexity of the proposed ME scheme is analyzed in section IV. Computer simulation results for the proposed algorithm are presented in Section V. A brief conclusion is given in Section VI.

Ⅱ. Motion estimation and mode decision in MPEG-4 and H.264

ME using variable block size modes is used to obtain the enhanced encoding efficiency in MPEG-4 and H.264 standards, where a large block mode is used in homogeneous region of a frame, and the smaller block mode can produce better motion compensation in complex region [10][11][12].

2.1 ME in MPEG-4

In MPEG-4 encoder, ME module considers 2 modes (16x16 and 8x8) to enhance the coding efficiency. An optimal MV (Motion Vector) for a block is obtained by minimizing SAD (Sum of Absolute Difference) between the current and reference block. The SAD is defined as

$$SAD_{N\times M} = \sum_{x=1,y=1}^{x=N,y=M} |c(x,y) - r(x-v_x, y-v_y)| \qquad (1)$$

where NxM is the block size of current mode, and a vector (v_x, v_y) is a MV of NxM mode. The c and r is current block and reference block, respectively. After ME process has estimated MVs for 16x16 and 8x8 mode, $SAD_{16\times16}$ and $SAD_{8\times8}$ have been calculated. Based on those, Mode Decision is performed by

$$\text{mode}_{opt} = \begin{cases} 16\times16, & \text{if } SAD_{16\times16} - N_B \le \sum_{b=1}^{4} SAD_{8\times8}^{b} \\ 8\times8, & \text{if } SAD_{16\times16} - N_B > \sum_{b=1}^{4} SAD_{8\times8}^{b} \end{cases} \qquad (2)$$

where N_B is the number of pixels inside the frame multiplied by bit per pixel value, and mode_{opt} is an optimal mode decided for a current MB.

2.2 ME in H.264

In H.264, 7 modes (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4) are used for ME process. At first, a MB is encoded as one of four MB modes, 16x16, 16x8 , 8x16, and P8x8. After ME process considers the 4 modes, the MB decided to be encoded as the 8×8 mode can be considered as further detailed modes such as 8x4, 4x8, and 4x4. The coding efficiency can be improved by using 7 block size modes whereas other standards consider one or two modes. ME process in H.264 is performed by minimizing a Lagrangian cost function in the form of

$$mcost = SAD_{N\times M} + \lambda_{motion} \times R_{motion} \qquad (3)$$

where λ_{motion} is a Lagrangian multiplier for rate constrained encoder control, and R_{motion} represents the number of bits required to encode the difference between the MV and predictive MV (PMV). Once the MVs for all possible modes have been obtained by minimizing (3), the best mode for current MB is determined by minimizing following cost function.

$$J(mode) = SSD + \lambda_{mode} \times R_{mode} \qquad (4)$$

where SSD is the sum of the squared differences between the reconstructed and the original block.

λ_{mode} is a Lagrangian multiplier used for rate control. And R_{mode} is the number of bits required to encode the header and motion vectors information of a mode. Mode represents one of possible combination of {16x16, 16x8, 8x16, 8x8,..., 4x4}. For example, mode can be {one 16×16 block} for 16x16 mode, {two 16×8 blocks} for 16x8 mode, { two 8×8 blocks, eight 4×4 blocks} for P8x8 mode, and so on.

In video coding standard using multiple block modes, such as MPEG-4 and H.264, ME module requires huge computation time to estimate MV and to decide an optimal mode. To solve this problem, several conventional schemes[13][14] had been proposed.

2.3 Fast ME for H.264

In H.264 video coding standard, since ME module uses 7 modes, the full search ME scheme is very expensive in the respect of computational complexity. In[14], instead of full search, FME (Fast Motion Estimation) scheme using motion vector prediction and early termination techniques is proposed. The predictor for MV gives four PMVs, $PMV_{adjacent}$, $PMV_{collocated}$, PMV_{zero}, and PMV_{mode} where $PMV_{adjacent}$ is a median vector of MVs of adjacent blocks at the left, top, and top-right. $PMV_{collocated}$ is the MV of a collocated block in previous frame, and PMV_{zero} = (0,0). The MV of a block mode among 7 modes (16×16,16×8,8×16, 8×8,8×4,4×4) is highly correlated with those of other modes. For example, a MV of 16x16 block can be used as a PMV for a 16x8 block. In [14], a MV estimated for larger block is used as a PMV for smaller block. The PMV generated from different modes is denoted by PMV_{mode}. The MVs estimated by searching around $PMV_{adjacent}$, PMV_{mode}, $PMV_{collocated}$, and PMV_{zero} are denoted as $MV_{adjacent}$, MV_{mode}, $MV_{collocated}$, and MV_{zero}, respectively. Since they are very close to optimal MVs generated from full-search ME, using PMVs increases the coding speed of ME module without loss of image quality.

In addition to using PMV, an early termination scheme[14] has been proposed to reduce the complexity of ME process, where the coding efficiency is further improved by using adaptive threshold technique. Since *mcost* value of a mode is highly correlated with those of other modes, the *mcost* value of a mode is used as a threshold for other mode. If the $mcost(MV^*)$ of MV^* estimated for a mode is smaller than the threshold value calculated from the *mcost*'s of the other size modes or neighbor blocks, then the estimated MV^* is decided as an optimal MV for the considered mode, and the ME process is skipped to the next mode.

Ⅲ. Proposed ME algorithms for transcoder

3.1 ME using 3 PMVs and thresholds

In homogeneous transcoders such as MPEG-2 to MPEG-2 and H.264 to H.264, MVs decoded in the decoder part of transcoder have been used as reused MVs. Using the reused MV and MV refinement process reduces the complexity of ME module in transcoder significantly. However, in heterogeneous video transcoding such as MPEG-4 to H.264 transcoder, since block size for ME in MPEG-4 is different from that in H.264, we can not use the decoded MVs to estimate an optimal MV.

To overcome the mismatch problem between block sizes considered in the decoder and encoder parts, we propose an efficient ME scheme which uses 3 PMVs, PMV_{reused}, $PMV_{adjacent}$, and PMV_{mode}. Though MVs provided from the decoder part have been generated from different size blocks, they can be used to make predictive MVs for different size blocks. Fig 1 shows a process to make PMV_{reused} from MVs transferred from the decoder part. If the MB decoded in MPEG-4 decoder is a 16x16 mode, PMV_{reused}'s for all modes in H.264 encoder are set to the decoded MV. On the other

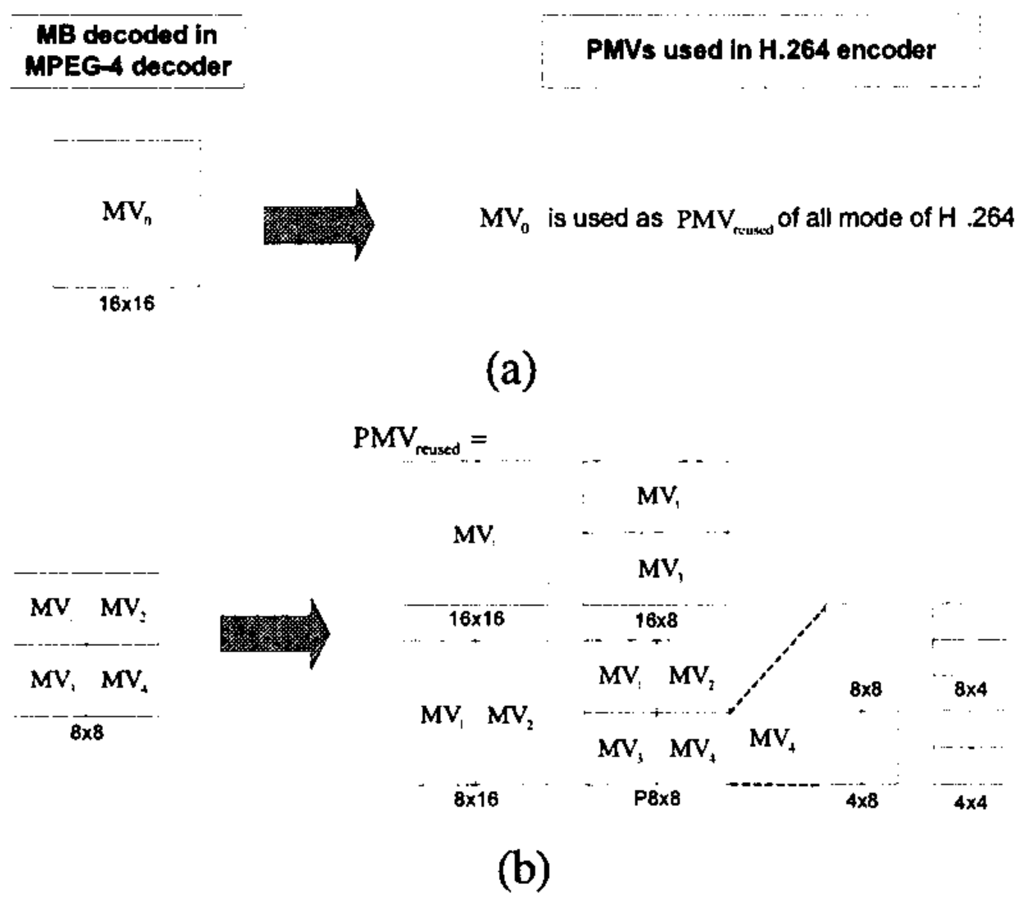


Fig. 1. PMV_{reused} for H.264 encoder is set to one of MVs decoded in MPEG 4 decoder part. : (a) When decoded MB is 16x16 mode, PMV_{reused}'s of all modes are set to the decoded MV (i.e, MV_0), (b) When decoded MB is 8x8 mode, PMV_{reused} of each mode is selected from one of the decoded 4 MVs (i.e, MV_1, MV_2, MV_3, and MV_4).

hand, when the decoded MB represents 8x8 mode, four decoded MVs are used to make PMV_{reused}'s for blocks at same location.

Since modes of the decoded blocks may be different from those of blocks considered in the encoder, the decoded MVs may be not always appropriated for making PMV_{reused}. Thus, in addition to the PMV_{reused}, we use $PMV_{adjacent}$ and PMV_{mode} used in FME scheme. Using $PMV_{adjacent}$ and PMV_{mode} can further improve the performance of transcoder with a little extra computational complexity. Since using PMV_{zero} and $PMV_{collocated}$ produces worse performance than utilization of $PMV_{adjacent}$ and PMV_{mode} in the viewpoint of estimating an optimal MV, we do not use information about PMV_{zero} and $PMV_{collocated}$.

After we have PMV_{reused}, $PMV_{adjacent}$, and PMV_{mode}, a diamond search method [14] is applied over a narrow search range around 3 PMVs. Since PMVs are very similar with an optimal MV, the computational complexity of ME process can be reduced significantly.

To increase the speed of ME further, we use thresholds to allow early termination of ME. Since PMV_{reused} is more correlated with the optimal MV than $PMV_{adjacent}$ and PMV_{mode}, the $PMV_{adjacent}$ and PMV_{mode} are employed after ME process uses PMV_{reused}. If the mcost value of the MV estimated by using PMV_{reused} is smaller than the threshold in Table I, the ME process is skipped to next mode without considering $PMV_{adjacent}$ and PMV_{mode}. In Table I, $T(i)$'s are threshold values for early termination scheme, where i is a mode number, $1 \le i \le 7$. In this technique, thresholds were calculated from $mcost$'s of spatially adjacent blocks or other mode. When a motion vector is estimated for a 16x16 block (i.e., $mode(1)$), the threshold value $T(1)$ is set to $mcost(PMV_{adjacent})$. And, for other modes, the threshold values $T(i)$'s are calculated from $mcost$'s of the larger size mode. The proposed scheme using 3 PMVs and the early termination scheme can reduce the complexity of ME module in transcoder without loss of image quality.

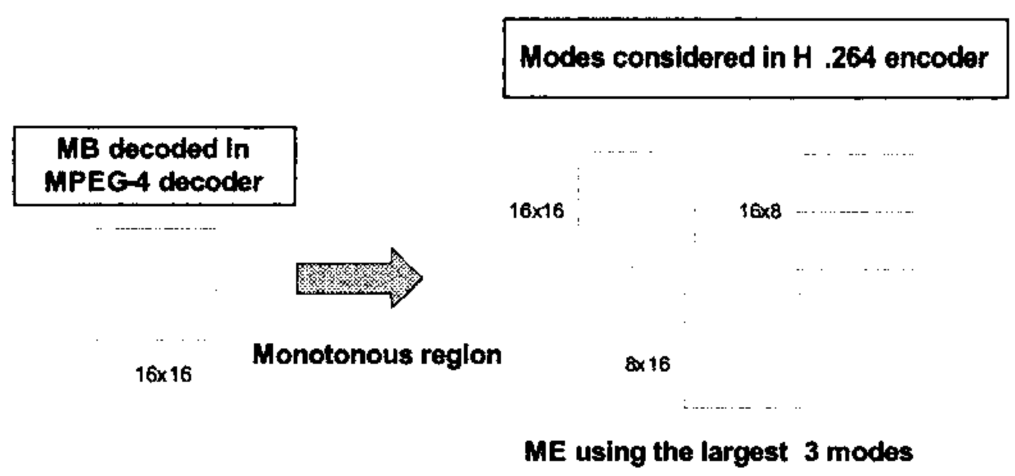
Table 1. Various threshold values

Mode Number i	Block Size of $mode(i)$	Threshold $T(i)$	$mcost$ calculated from the estimated MV
1	16×16	$mcost_{16\times16}$ $(PMV_{adjacent})$	$mcost_{16\times16}$
2	16×8	$mcost_{16\times16}/2$	$mcost_{16\times8}$
3	8×16	$mcost_{16\times16}/2$	$mcost_{8\times16}$
4	8×8	$mcost_{16\times8}/2$	$mcost_{8\times8}$
5	8×4	$mcost_{8\times8}/2$	$mcost_{8\times4}$
6	4×8	$mcost_{8\times8}/2$	$mcost_{4\times8}$
7	4×4	$mcost_{8\times4}/2$	$mcost_{4\times4}$

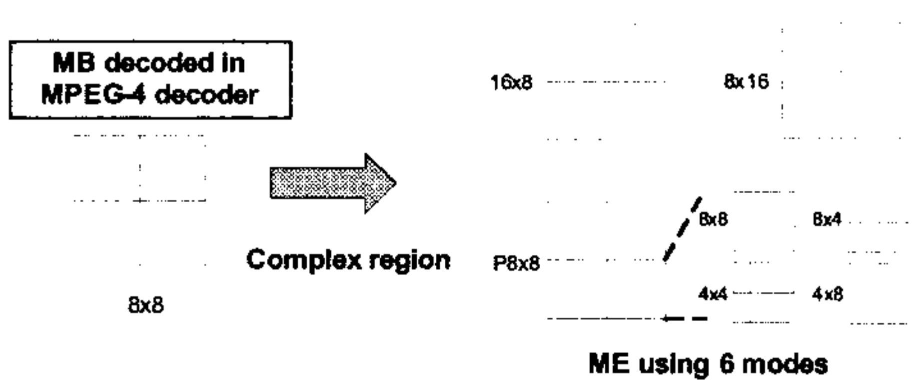
3.2 Early Determined Mode Decision (EDMD)

When a MB is encoded with one of large block modes (e.g. 16x16, 16x8, 8x16) in ME module of H.264, a few bits are required to encode the motion vector and mode information, and the coding efficiency increases. But, if the large block modes are used for regions containing some detail data, such as edge or text, the motion compensated residual may have a significant amount of energy. Thus, the large block mode is not appropriate for the detailed regions. On the other hand, small block modes (e.g. 8x8, 8x4, 4x8, 4x4) may give a lower energy residual for the detail image parts. Thus, the small block mode is more appropriate for complex region than large block mode. But they require many bits to encode the motion vector and mode information of the block. Consequently, whereas the large block modes are appropriate for homogeneous regions, small block modes may be proper for complex region. In this section, we propose an efficient mode decision scheme which utilizes mode information transferred from MPEG-4 decoder of transcoder.

The proposed scheme is described in Fig 2. If the decoded mode information is 16x16, the content of block is assumed as monotonous, the H.264 encoder of the transcoder considers only 3 modes (16x16, 16x8, and 8x16). On the other hand, if the MPEG-4 decoder produces a 8x8 mode, the corresponding block is in complex region, and 6 modes excluding a 16x16 mode are considered in H.264 encoder. When the proposed scheme is used for ME module of transcoder, some modes can be skipped. Due to skipping, the proposed EDMD can reduce the complexity of encoder in transcoder without loss of image quality.



(a)



(b)

Fig. 2. Efficient Mode Decision: (a) When the mode of MB decoded in MPEG 4 is 16x16, (b) When the mode of MB decoded in MPEG 4 is 8x8.

3.3 Fast ME algorithm using 3PMVs and EDMD

In this section, we describe a fast ME algorithm using 3 PMV and EDMD scheme. Table II and Table III show the threshold values and PMV_{mode} used for the proposed fast ME algorithm. Since 16x16 mode is not considered for ME when the decoded MB is a 8x8 mode, the T(i=1) is not required. Thus we propose the thresholds for 16x8 and 8x16 mode as in Table II, where the threshold values are calculated from $mcost_{16\times16}(PMV_{adjacent})$. The PMV_{mode} used in the proposed algorithm is represented in Table III, where some modes are not considered due to skipping. The proposed fast ME algorithm using the threshold values and PMVs of Table II and Table III is described in Figure 3.

In the first step of Figure 3, we select some modes ($i^{start} \sim i^{last}$) that are considered for ME in H.264 encoder. Then, in the next steps, a diamond search method is applied for each PMV to obtain MV_{reused}, MV_{mode}, and $MV_{adjacent}$. After ME process estimates a MV_{reused}, we use the threshold values of Table II to increase the transcoding speed without loss in quality. If mcost of MV_{reused} is smaller than the threshold T(i), we stop the ME process for a current mode, and skip to the next mode. On the other hand, when mcost of MV_{reused} is larger than the threshold, we use PMV_{mode} and $PMV_{adjacent}$ to find an optimal MV.

Ⅳ. Computational Complexities

In this section, we quantitatively evaluate the complexities of ME modules in transcoder using full search, FME, and proposed ME scheme. Since the transcoder using the full search scheme has same complexity as re-encoding process, ME module using the full search has the highest computational complexity. In MPEG-4 to H.264 transcoder, when ME is performed by full search scheme within search range of $\pm N$, the number of block matching processes (NBM) for mode(i) becomes

$$NBM_{FULL}(i) = (2 \times N + 1)^2 \qquad , 1 \le i \le 7, \tag{5}$$

where i is a mode number. In the case using FME scheme, 4 PMVs and threshold values are used to reduce the number of block matching processes. Note that the NBM of FME scheme is not fixed since ME process is performed until thresholding condition is satisfied[14]. Prior to the first thresholding assessment in FME scheme, ME using 3 PMVs is performed. And then 4 positions per each PMV are searched by diamond search method. Thus, total number of the required block matching processes can be expressed by

$$12 \le NBM_{FME}(i) \le G \tag{6}$$

where G is the number of block matchings when ME process using 3 PMVs is not satisfied with the first thresholding assessment. If the search range is set to $\pm N$, then the maximum number of G can be $(2 \times N + 1)^2$.

4.1 Complexity of ME module using 3 PMVs and Thresholds

In ME using 3 PMVs proposed in Section III, PMV_{reused}, PMV_{mode}, $PMV_{adjacent}$ are used in Figure 3. In the proposed scheme, diamond search method is applied to each PMVs. If mcost of MV_{reused} estimated in the first step of Figure 3 is smaller than a threshold value, only 4 block matching processes are required. Otherwise, if the condition of the first step is not satisfied, 12 block matching processes are required since ME using PMV_{mode} and $PMV_{adjacent}$ is performed to find a further optimal MV. Thus, the required number of block matchings in the transcoder using 3 PMVs scheme becomes

$$4 \le NBM_{3PMV}(i) \le 12. \tag{7}$$

As we can see from (6) and (7), the maximum number of the block matching processes required in the proposed ME using 3 PMVs is 12 whereas the minimum number required by FME scheme is 12. Therefore, the complexity of the proposed ME using 3 PMVs is less than that of FME scheme as follows

$$NBM_{3PMV}(i) \le NBM_{FME}(i). \tag{8}$$

4.2 Complexity reduced by the proposed EDMD

When ME modules using FME and 3 PMVs consider 7 block size modes, the total numbers of block matchings become

$$\sum_{i=1}^{i=7} NBM_{\mathrm{FME}}(i) \text{ and } \sum_{i=1}^{i=7} NBM_{\mathrm{3PMV}}(i) \tag{9}$$

respectively. In the proposed algorithms of Figure 3, the EDMD scheme reduces the number of modes considered in ME according to mode information decoded in MPEG-4 decoder part. The complexity of ME using EDMD scheme can be represented as follows

$$\sum_{i=i^{start}}^{i^{last}} NBM_{\mathrm{3PMV}}(i) \le \sum_{i=1}^{7} NBM_{\mathrm{3PMV}}(i). \tag{10}$$

As we can see from (10), when EDMD scheme is applied for ME in MPEG-4 to H.264 transcoder, the number of the considered modes is smaller than that of other schemes. That is, EDMD can be used to effectively reduce the computational complexity of ME module in transcoder.

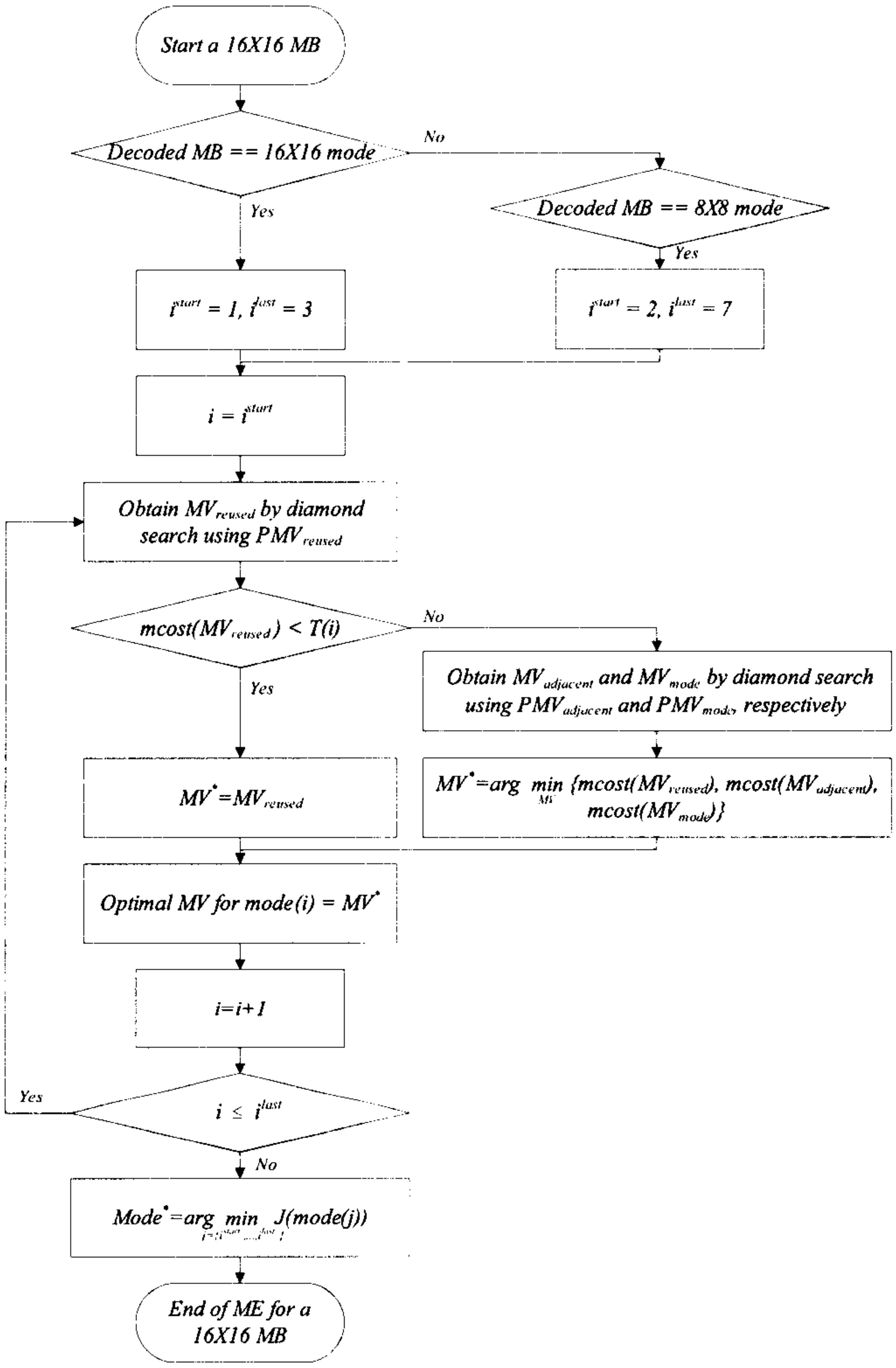


Fig. 3. The proposed ME procedure for a macrblock.

Table 2. Threshold values used in the proposed fast ME algorithm for transcoder.

Mode Number i	Block Size of $mode(i)$	Threshold $T(i)$		$mcost$ calculated from the estimated MV
		When 16x16 mode is decoded in MPEG-4	When 8x8 mode is decoded in MPEG-4	
1	16×16	$mcost_{16\times16}$ $(PMV_{adjacent})$	Not required	$mcost_{16\times16}$
2	16×8	$mcost_{16\times16}/2$	$mcost_{16\times16}$ $(PMV_{adjacent})/2$	$mcost_{16\times8}$
3	8×16	$mcost_{16\times16}/2$	$mcost_{16\times16}$ $(PMV_{adjacent})/2$	$mcost_{8\times16}$
4	8×8	Not required	$mcost_{16\times8}/2$	$mcost_{8\times8}$
5	8×4	Not required	$mcost_{8\times8}/2$	$mcost_{8\times4}$
6	4×8	Not required	$mcost_{8\times8}/2$	$mcost_{4\times8}$
7	4×4	Not required	$mcost_{8\times4}/2$	$mcost_{4\times4}$

Table 3. PMV_{mode}'s used in the proposed fast ME algorithm.

Mode Number i	Block Size of $mode(i)$	PMV_{mode}		Estimated MV
		When 16x16 mode is decoded in MPEG-4	When 8x8 mode is decoded in MPEG-4	
1	16×16	Not required	Not required	MV_1
2	16×8	MV_1	Not required	MV_2
3	8×16	MV_1	MV_2	MV_3
4	8×8	Not required	MV_3	MV_4
5	8×4	Not required	MV_4	MV_5
6	4 × 8	Not required	MV_4	MV_6
7	4×4	Not required	MV_6	MV_7

V. Experimental Results

Computer simulations using video sequences were performed to evaluate the performance of the proposed fast ME algorithm. They are pre-encoded by MPEG-4 using "IPPP.." of GOP structure with 128kbps and 10Hz for Foreman and News, 305kbps and 15Hz for Mobile and calendar, 502kbps and 30Hz for Stefan. The pre-encoded test sequences are transcoded by a MPEG-4 to H.264 transcoder with same frame rate and GOP structure. In this simulations, "Full search", "FME", and "3PMV+EDMD" schemes are used in ME module of transcoder. In transcoding using full search ME, the size of search range is set to ±16, i.e., N=16.

5.1 Performances of transcoder using 3 PMVs

To prove the performance of scheme using 3 PMVs, a sequence "Mobile & Calendar" pre-encoded by MPEG-4 is transcoded into H.264 sequence by transcoders using "Full search", "FME", and "3PMV" schemes, respectively. Fig. 4 and Fig. 5 show the CPU times and the PSNR's resulted by ME modules using the three schemes, respectively. As we can see these figures, the complexity of ME module using 3 PMVs is smaller than that of FME whereas the qualities of images transcoded by various schemes (Full search, FME, 3PMV) are same. These results imply that the scheme using 3 PMVs exhibits a significant improvement in the reduction of computational complexity without loss of image quality when compared with the conventional schemes.

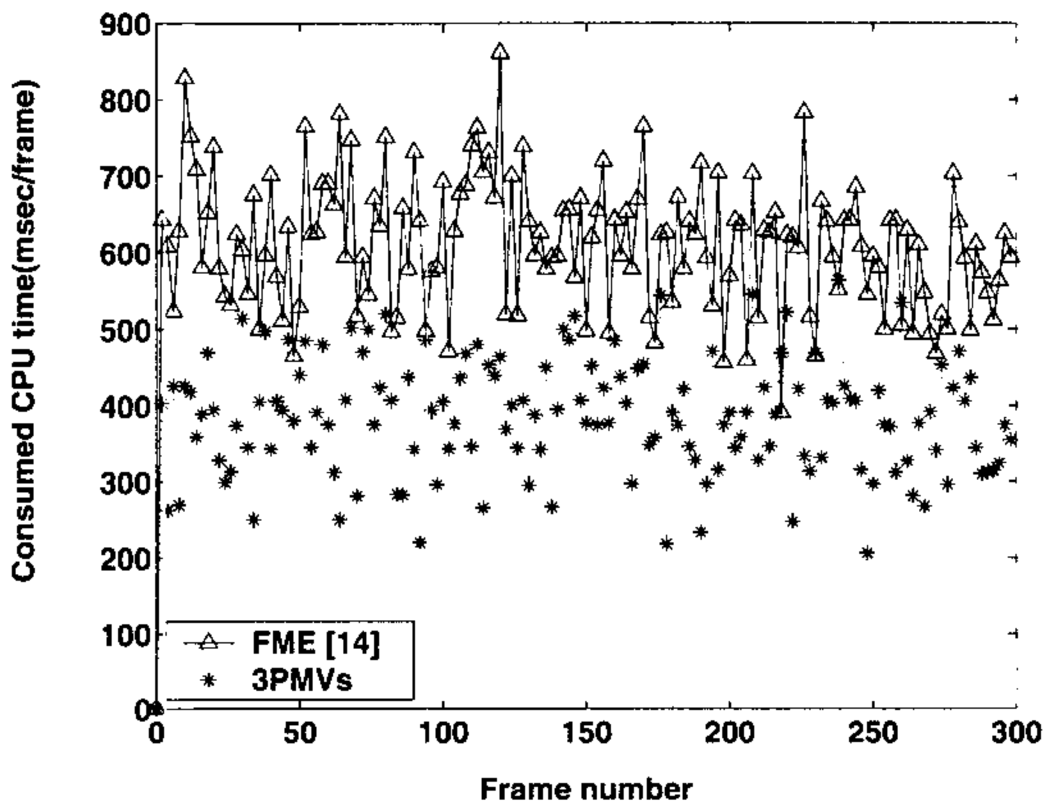


Fig. 4. CPU times consumed by ME module of transcoder (358 kbps →148kbps, 15Hz →15Hz) for a CIF 'Mobile & Calendar' sequence.

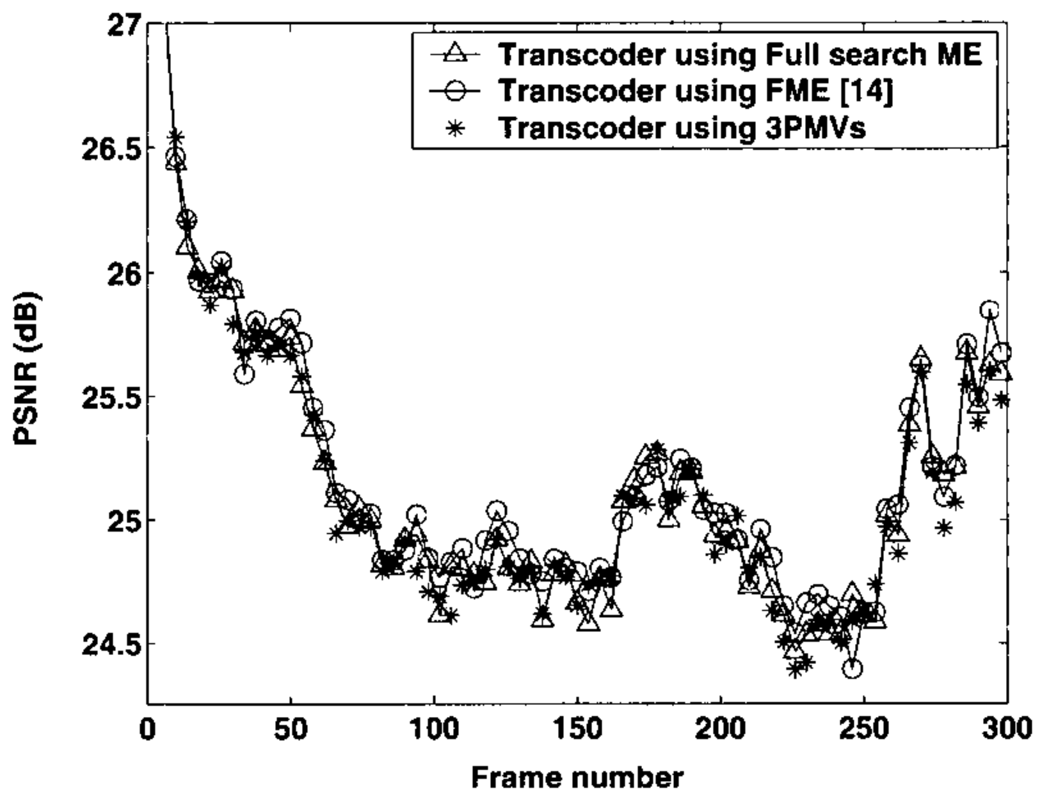


Fig. 5. PSNR's resulted from transcoder (358 kbps → 148kbps, 15Hz →15Hz) for a CIF 'Mobile & Calendar' sequence.

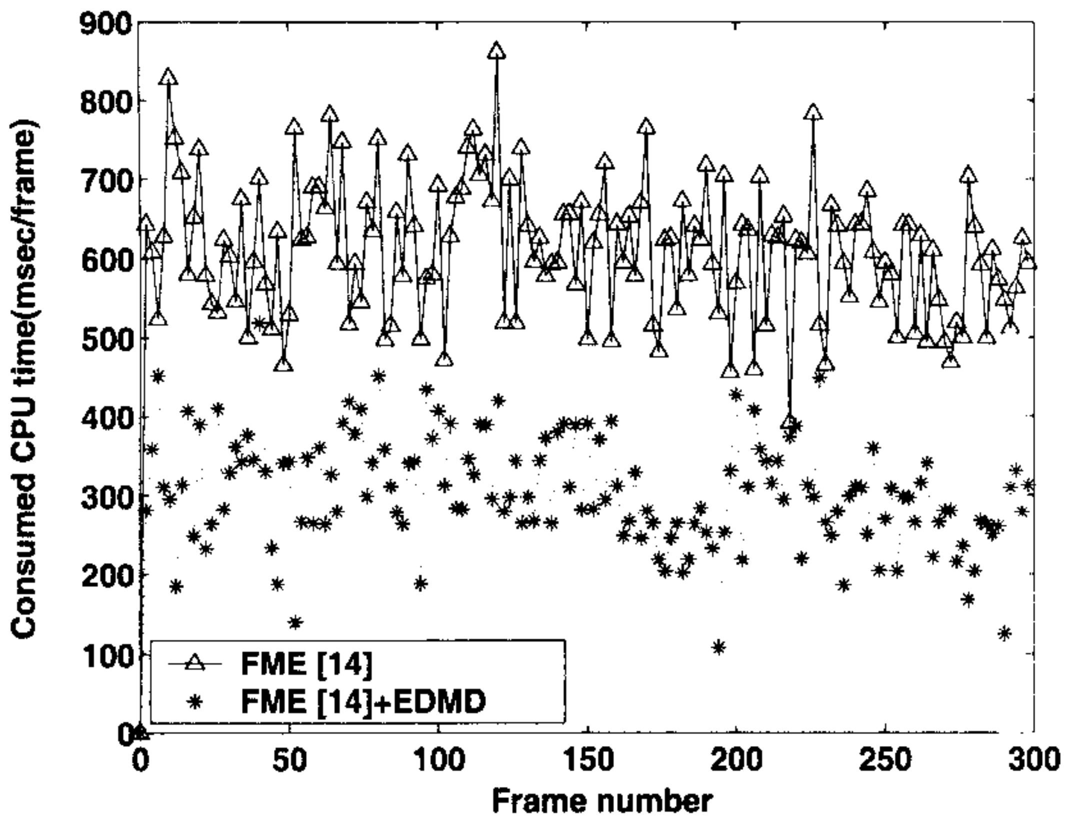


Fig 6. CPU times consumed by ME module of transcoder (358 kbps →148kbps, 15Hz →15Hz) for a CIF 'Mobile & Calendar' sequence

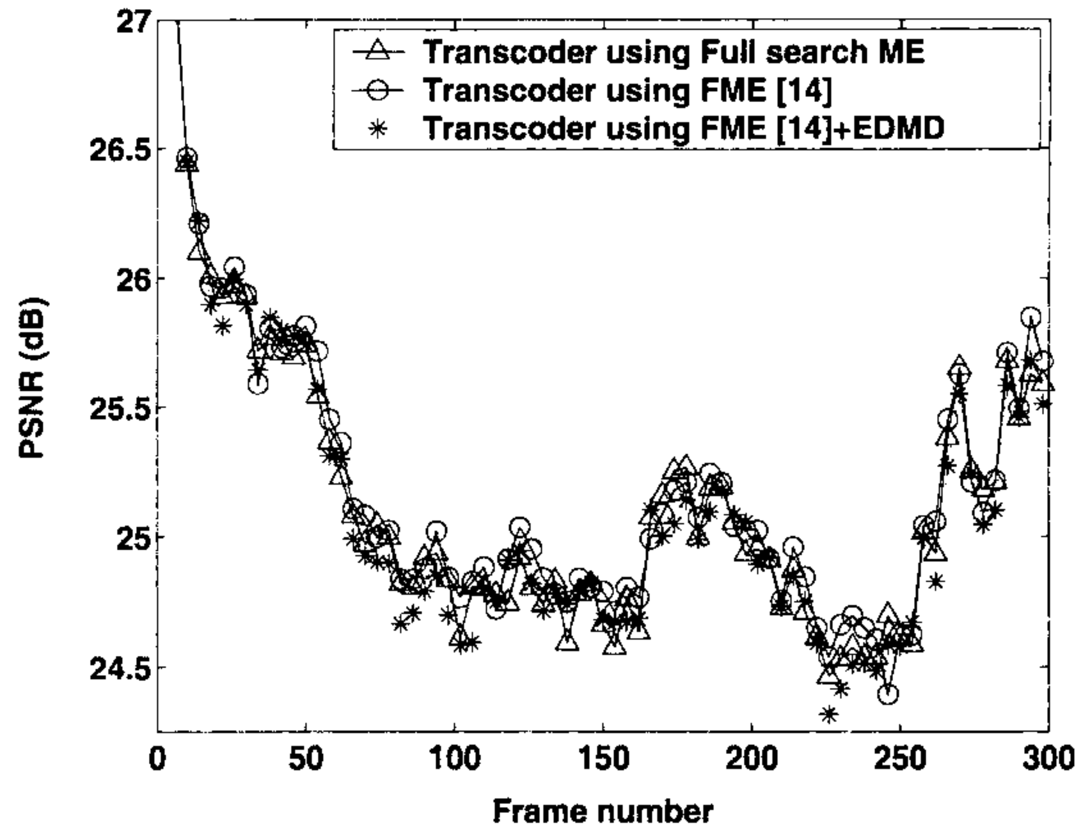


Fig 7. PSNR's resulted from transcoder (358 kbps → 148kbps, 15Hz →15Hz) for a CIF 'Mobile & Calendar' sequence.

5.2 Performances of transcoder using EDMD

The pre-encoded test sequence "Mobile & Calendar" is used to prove

the performance of transcoder using EDMD scheme. Fig. 6 and Fig. 7 show the CPU times and the PSNR's resulted by transcoders using "Full search", "FME", and "FME+EDMD" scheme, respectively. As we can see the figures, PSNR's of conventional and "FME+EDMD" scheme are almost same whereas CPU times consumed by module using the proposed mode decision scheme are much smaller than that of conventional scheme. From these results, we can see that EDMD scheme is a highly effective mode decision scheme in MPEG-4 to H.264 transcoder.

5.3 Tests for transcoder using 3 PMVs and EDMD

4 test sequences ("Foreman", "News", "Mobile & Calendar", and "Stefan") are used to prove the performance of transcoder using 3 PMVs and EDMD scheme.

To compare the computational complexities of the "3PMV+ EDMD" scheme with those of the "FME" schemes, the CPU times consumed by the ME module for various sequences are checked in Fig. 8, where the consumed time is displayed in the resolution of msec/frame. As shown in this figure, the"3PMVs+ EDMD" scheme requires much smaller CPU time than the FME schemes. From (8) and (10), complexity of "3PMVs+EDMD" scheme can be compared with that of "FME" as follows 1

$$\sum_{i=i^{start}}^{i^{last}} NBM_{3PMV}(i) \le \sum_{i=1}^{7} NBM_{FME}(i). \quad (11)$$

From the results in Fig. 8, we can see that "3PMV+ EDMD" scheme has a significant benefit in the view point of computational complexity when compared with FME scheme.

In order to check the usefulness of the proposed scheme, the rate distortion curves of various schemes ("full search", "FME", and "3PMV+EDMD") are evaluated in Fig. 9. As we can see this figure, the average PSNR's resulting from the "3PMV+EDMD" scheme is close to those from the other schemes. It implies that the proposed "3PMV+EDMD" scheme not degrade the image quality in various bit-rate reduction condition.

The simulation results shown in this section indicate that the PSNR's of the bitstreams generated by the proposed ME are almost equal to those of the conventional ME's, whereas the transcoding system using the proposed ME scheme reduces computational complexity significantly.

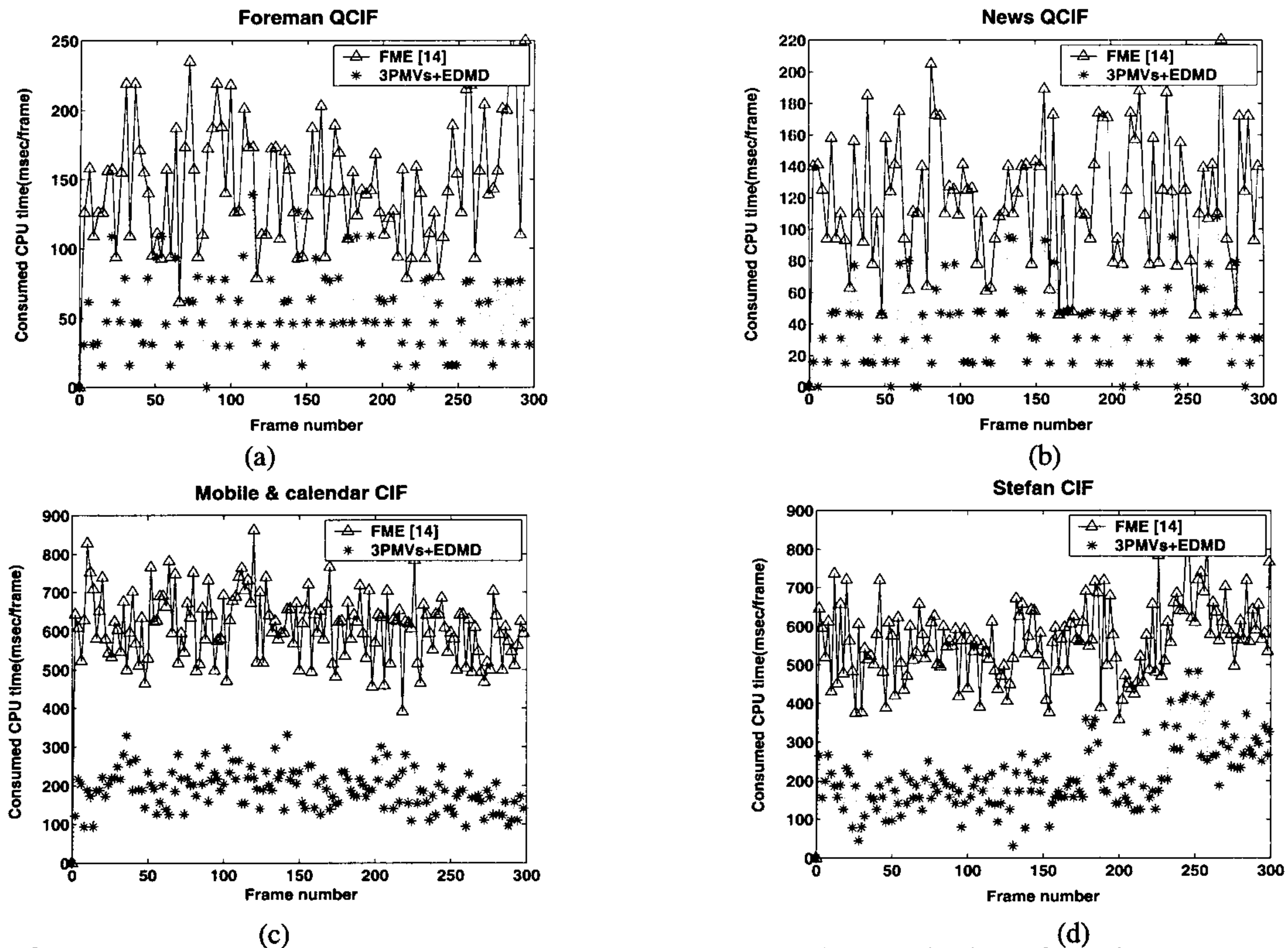


Fig 8. CPU times consumed by ME module in transcoder using FME and proposed scheme for various sequences, (a) Foreman, (b) News, (c) Mobile & calendar, (d) Stefan.

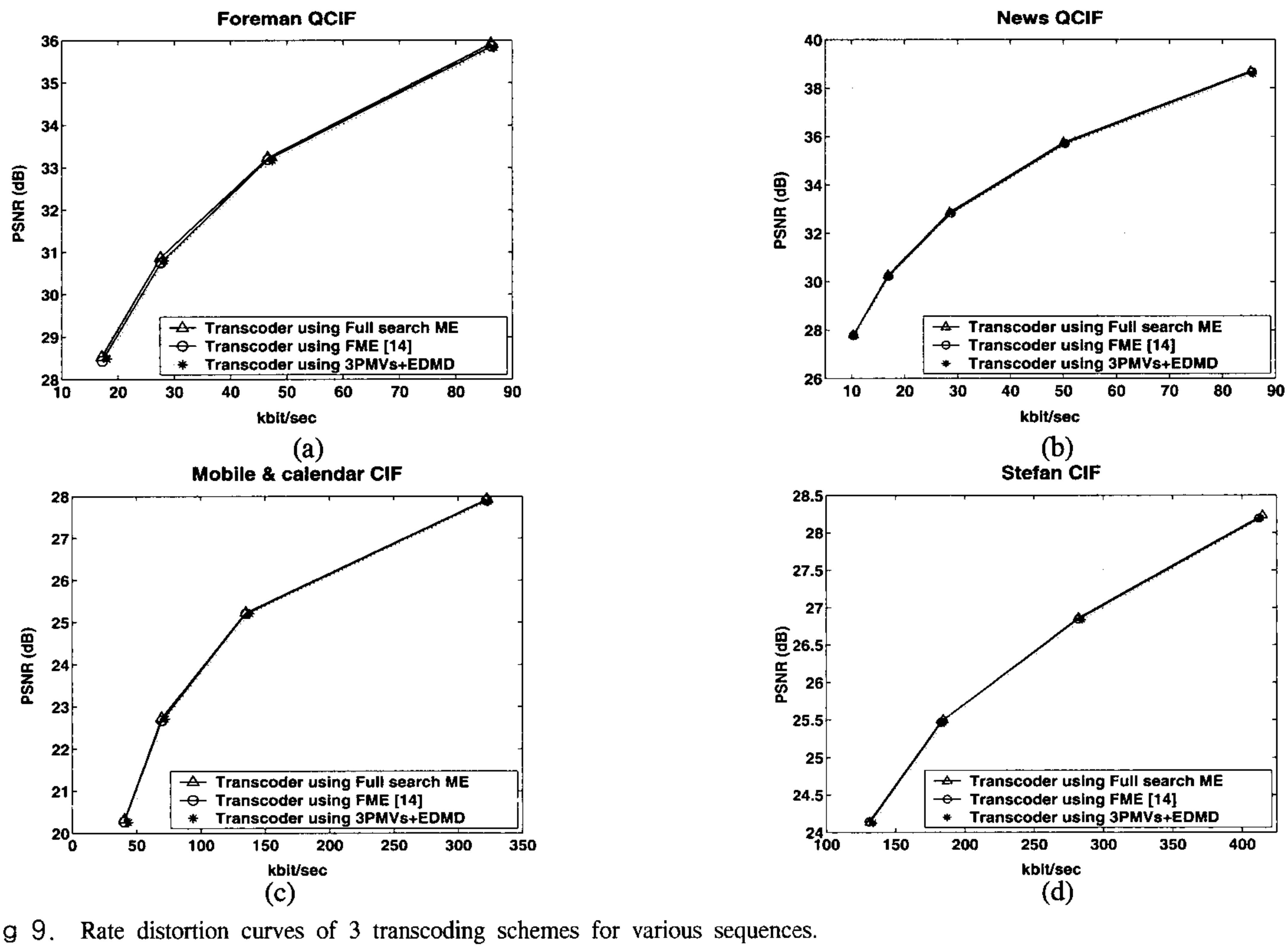


Fig 9. Rate distortion curves of 3 transcoding schemes for various sequences.

Ⅵ. Conclusion

In this paper, we have proposed a fast ME scheme to reduce the complexity of transcoder between MPEG-4 and H.264. To reduce the complexity, we use 3 PMVs and the mode information decoded in the decoder part of transcoder. Since the 3 PMVs are very close to optimal motion vector, the proposed scheme can speed up transcoding process. Furthermore, the proposed EDMD scheme can be used for efficient mode decision of transcoder. In this scheme, the mode of a MB in a frame is expected from the decoded mode information. Instead of considering all modes, using the candidate modes reduce the complexity of ME module in transcoder. Finally, we propose the "3 PMV+EDMD" scheme that use 3 PMVs and mode information decoded in MPEG-4 decoder. The scheme provides a fast algorithm for MPEG-4 to H.264 transcoder.

References

〔1〕 T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video technol*, Vol.13, pp.560 – 576, July 2003.

〔2〕 Vetro, A., Christopoulos, C., Huifang Sun, "Video transcoding architectures and techniques: an overview", *IEEE Signal Processing Magazine*, Vol.20, pp.18 – 29, March 2003.

〔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, pp.953 - 967, Dec. 1998.

〔4〕 J. Youn, M. T. Sun, and Chia-Web Lin, "Motion estimation for high performance transcoding", *IEEE Trans. Consumer Electronics,* Vol.44, pp.649 - 658, Aug. 1998.

〔5〕 C. W. Lin, Y. C. Chen, and M. T Sun, "Dynamic region of interest transcoding for multipoint video conferencing", *IEEE Trans. Circuits Syst. Video technol.*, Vol.13, pp.982 - 992, Oct. 2003.

〔6〕 Y. P. Tan and H. Sun, "Fast motion re-estimation for arbitrary downsizing video transcoding using H.264/AVC standard", *IEEE Trans. Consumer Electronics,* Vol.50, pp.887 - 894, Aug. 2004.

〔7〕 I. H Shin, Y. L Lee, and H. W. Park, "Motion estimation for frame-rate reduction in H.264 transcoding", *in proc., 2nd IEEE workshop, Software Technol. for Future Embedded and Ubiquitous Syst,* pp 63 – 67, May 2004.

〔8〕 T. Shanableh and M. Ghanbari, "Heterogeneous video transcoding to lower spatio-temporal resolutions and for Different Encoding Formats", *IEEE Trans. Multimedia,* Vol.2. No.2, June2000, pp.101-110.

〔9〕 K.D. Seo and J. K. Kim, "Fast motion vector refinement for MPEG-1 to MPEG-4 transcoding with spatial down-sampling in DCT domain", *in proc., Int. Conf., Image Processing, V*ol.1, Oct. 2001, pp.469 – 472.

〔10〕 S.C. Dabner, "Motion estimation for HDTV", *Image Processing for HDTV, IEE Colloquium,* Oct 1989, pp.2/1 - 2/4

〔11〕 H. Gharavi, and Mills, M., "Block matching motion estimation algorithms-new results", *IEEE Trans. Circuits Syst. Video technol.*, Vol.37, pp.649 - 651 May. 1990.

〔12〕 A.M. Tourapis, O. C.Au, and M. L.Liou,"Highly efficient predictive zonal algorithms for fast block-matching motion estimation," *IEEE Trans. Circuits Syst. Video technol.*, Vol.12, No.10, pp.934-947, Oct.2002.

〔13〕 D.S. Turaga, and C. Tsuhan, "Estimation and mode decision for spatially correlated motion sequences", *IEEE Trans. Circuits Syst. Video technol.*, Vol.11, pp.1098 – 1107, Oct. 2001

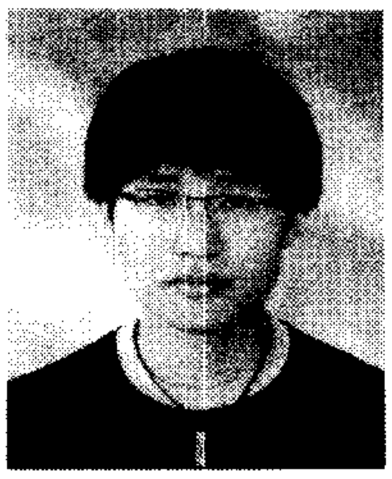
〔14〕 J. Xu, Z. Chen, and Y. He "Efficient Fast ME Predictions and Early-termination Strategy Based on H.264 Statistical Characters," *In proc., Int.Conf., Inform.,Communication and Signal Processing* vol.1, pp.218-222, Dec. 2003.

〔15〕 JVT codec reference software, http://bs.hhi.de/~suehring/tml/download/jm82.

Jong-Ki Han Lifelong Member

He was born in Seoul, Korea, on September 5, 1968. He received the B.S., M.S., and Ph. D degree in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1992, 1994, and 1999, respectively. From 1999 to 2001 he was a member of technical staff at corporate R & D Center, Samsung Electronics Co., Suwon, South Korea. He is currently an associate Professor at the department of information and communications engineering, Sejong university, Seoul, Korea. His research interests include image and audio signal compression, transcoding, and VLSI signal processing.

Chan-Won Seo Associate Member



He was born in Incheon, Korea, on March 23, 1982. He received the B.S. degree in 2007 from the Department of Information and Communication Engineering, Sejong University, Seoul, Korea, where he is currently pursuing the M.S. degree. His research interests include video coding, scalable video coding, and transcoding.