Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Fast mode decision algorithm for spatial resolutions down-scaling transcoding to H.264*

BU Jia-jun, MO Lin-jian†, CHEN Chun, YANG Zhi
(*School of Computer Science, Zhejiang University, Hangzhou 310027, China*)
†E-mail: molin@zju.edu.cn

Received Dec. 15, 2005; revision accepted Feb. 17, 2006

Abstract: A fast mode decision algorithm is proposed in this paper to accelerate the process of transcoding videos into H.264 with arbitrary rate spatial resolution down-scaling. The proposed algorithm consists of three steps. First, an early-stop technique is introduced to determine the 16×16-mode blocks, which take up about 70% of all the macroblocks; then, a bottom-up merging process is performed to determine the mode of rest non-early-stopped blocks; and then, we adopt half-pixel motion estimation to further refine the acquired predictive motion vectors. In order to obtain the predictive motion vectors for early-stop and merging processes, we propose a motion vector composition scheme, which can reuse the information in the input pre-encoded videos to handle the spatial resolution down-scaling. Experimental results showed that our algorithm is about four times faster than the Cascaded-Decoder-Encoder method and has negligible PSNR drop and little bit rate increase.

Key words: Transcoding, H.264, Mode decision, Motion composition
doi:10.1631/jzus.2006.AS0070 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

The forthcoming multimedia telecommunication services are expected to use more efficient pre-encoded video for storage and transmission. H.264, as the emerging video coding standard, achieves much higher coding efficiency than previous standards. Thus, in some of the real-time video communication scenarios, it is desirable to rapidly transcode videos from other standards to H.264 without much video quality reduction.

In many applications, the pre-encoded video needs to be transcoded to lower spatial resolution, such as fitting for various display screen sizes on hand-held devices. Thus, To-H.264 transcoder needs to handle spatial resolution down-scaling. However, different from previous video coding standard, H.264 allows seven block-sizes motion estimation, which makes the transcoding task more challenging.

Many works addressed the spatial resolution reduction transcoding (Xin *et al*., 2002; Shanableh and Ghanbari, 2000; Takahashi *et al*., 2001). The motion information from input pre-encoded video can be reused during H.264 motion estimation processing, to reduce the computational complexity. The works by (Xin *et al*., 2002; Takahashi *et al*., 2001) describe techniques for handling motion vector (MV) re-use in transcoding to previous standards, such as MPEG2 and MPEG4. However, many new features are introduced that make the transcoding to H.264 very different. With the quad-tree structure and various block-sizes, the motion information re-use techniques should be re-designed while transcoding to H.264.

Seven block-sizes mode decision for INTER frame is firstly introduced by H.264. Chang *et al*.(2004) pointed out that variable block size motion estimation and mode decision take up about 70% computational resource in the encoding process. It is

* Project supported by the National Natural Science Foundation of China (No. 60573176), the Key Technologies R & D Program of Zhejiang Province (Nos. 2005C23047 and 2004C11052), China

necessary to explore fast algorithm. Some researchers tried to reduce the search range, especially with the early-stop technique which stops the process early when some conditions are satisfied. For instance, early-stop on 16×16-mode (MODE 0 or 1 of INTER Block in H.264) technique directly sets the block to 16×16-mode according to the texture or sub-block MV information (Kucukgoz and Sun, 2004). Others address this problem through a different approach named information re-use. Top-down splitting (Zhou *et al*., 2004) and bottom-up merging (Zhou *et al*., 2004; Tu *et al*., 2003) methods are introduced to reuse the up/low-layer MV in the quad-tree structure. These techniques can also be applied for To-H.264 transcoder; moreover, reusing the MVs of input pre-encoded videos will achieve better performance, because the input pre-encoded motion information predicts the MV better than that obtained from the spatially or temporally neighboring macroblocks.

This work is aimed at designing a fast mode decision algorithm in transcoding to H.264 with arbitrary-rate spatial resolution down-scaling. The motion estimation and mode decision process in H.264 are combined to avoid unnecessary time consumption. A novel predictive motion vector (PMV) composing scheme is suggested to handle the arbitrary-block-size multiple-to-one macroblocks merging. The proposed algorithm consists of three steps. First, we try to pre-detect the 16×16 mode blocks (SKIP, 16×16 Inter) which almost take up 70% of them all; then, a bottom-up merging processing is performed to handle the non-decided blocks in Step 1; and then, the obtained MV will be refined with half-pixel precision search. Experimental results showed that our algorithm acts about four times faster than the Cascaded-Decoder-Encoder method and has negligible PSNR drop and little bit rate increase.

The rest of the paper is organized as follows. Section 2 describes how to merge multiple MVs into one. The details of the fast mode decision algorithm are described in Section 3, and Section 4 presents the experimental results. We conclude this paper in Section 5.

PROPOSED MV COMPOSITION SCHEME

For transcoding into lower spatial resolution pictures, a new MV is to be calculated from a set of input MVs of the pre-encoded higher spatial resolution sequence. The acquired MV will act as PMV for the encoding process of the lower spatial resolution sequence. For instance, transcoding a bitstream in CIF format into QCIF format requires calculating a new MV from four input MVs. This corresponds to transcoding four macroblocks into one macroblock. The proposed MV composition scheme in this section can predict the PMVs for various size blocks in H.264 with spatial resolution down-scaling.

To efficiently reuse the input motion information for better prediction, various methods have been proposed, such as Random, Mean, Weighted Average, Weighted Median, DCmax, etc. (Ahmad *et al*., 2005). Here Weighted Median method yields good performance. Referring to Weighted Median method, we propose an MV composition scheme, which is based on the block size and covered area corresponding to the input MVs, as shown in Fig.1. The details are as follow:

$$\begin{gathered} MV_{\text{pred}} \in \{MV_i\}, \\ WAMV = \left(\sum_i (CBSize_i \times MV_i)\right) \Big/ \left(\sum_i CBSize_i\right), \\ |MV_{\text{pred}} - WAMV| \le |MV_i - WAMV|,\ i = 1, 2, \ldots, N, \end{gathered} \quad (1)$$

where $WAMV$ is weighted average MV and MV_{pred} is the target PMV for the macroblock in down-scaling picture. N is the number of covered blocks. $CBSize_i$ and MV_i denote the block size in the covered area in the original picture and the corresponding MV, respectively. For instance, in Fig.1, $CBSize_A$ is 8×8=64, but $CBSize_B$ equals 16×8=128.

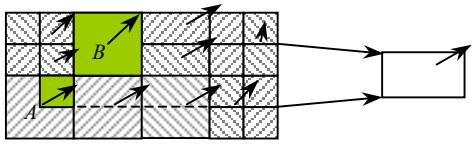

Fig.1 Merging multiple MVs to one. MV on block *B* has bigger impact on the target predictive MV than that of block *A*. Slim continuous lines mean the covered area (The source picture consists of various size-blocks)

It should be mentioned that if the input MVs are interlaced but the target MV is progressive; or the input MVs are progressive but target MV is interlaced,

some adjustments should be applied. This paper focuses on the former which are widely supported by actual applications.

Let (MV_x, MV_y) be the original MV, (MV'_x, MV'_y) be the adjusted MV; *TF* means top field and *BF* means bottom field.

(1) Field-to-frame, as shown in Fig.2:

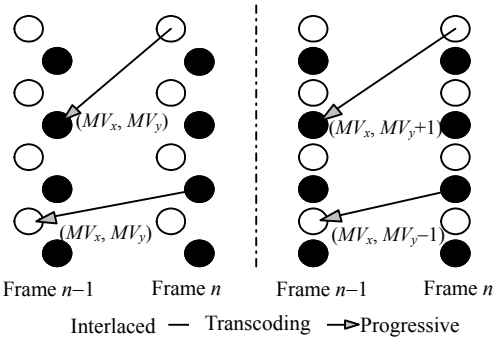

Fig.2 MV adjustment for field-to-frame transcoding

$$(MV'_x, MV'_y) = \begin{cases} (MV_x, MV_y), & MB_{\text{cur}} \in TF,\ MB_{\text{ref}} \in TF, \\ (MV_x, MV_y + 1), & MB_{\text{cur}} \in TF,\ MB_{\text{ref}} \in BF, \\ (MV_x, MV_y - 1), & MB_{\text{cur}} \in BF,\ MB_{\text{ref}} \in TF, \\ (MV_x, MV_y), & MB_{\text{cur}} \in BF,\ MB_{\text{ref}} \in BF. \end{cases} \quad (2)$$

(2) Frame-to-field:

$$(MV'_x, MV'_y) = \begin{cases} (MV_x, MV_y), & MB_{\text{cur}} \in TF,\ MB_{\text{ref}} \in TF, \\ (MV_x, MV_y - 1), & MB_{\text{cur}} \in TF,\ MB_{\text{ref}} \in BF, \\ (MV_x, MV_y + 1), & MB_{\text{cur}} \in BF,\ MB_{\text{ref}} \in TF, \\ (MV_x, MV_y), & MB_{\text{cur}} \in BF,\ MB_{\text{ref}} \in BF. \end{cases} \quad (3)$$

PROPOSED MODE DECISION ALGORITHM

There is high correlativity between the up-layer and low-layer blocks in the quad-tree structure of the seven block types in H.264. Top-down splitting and bottom-up merging methods are introduced to reuse the up/low-layer MV. Meanwhile, as in (Kucukgoz and Sun, 2004), 16×16-mode blocks usually take up about 70% of them all in H.264 encoding. If these 16×16-mode blocks are captured in the beginning of the mode decision, remaining procedures will be skipped and significant computation reduction will be achieved. Employing both the bottom-up merging and early-stop techniques, we proposed a novel fast predictive mode decision algorithm consisting of four steps:

Step 1: Prepare the PMVs

According to the scheme proposed in the previous section, we can get the PMVs for all size blocks in a macroblock, which are one 16×16, two 16×8 and two 8×16, four 8×8, eight 8×4 and eight 4×8, sixteen 4×4 size block MVs.

It should be mentioned that these PMVs are on-demand-calculated. For example, if the process is early-stop at 16×16 mode (proposed in Step 2), only one 16×16 size MV will be considered.

Step 2: Early-stop detection on 16×16-mode blocks

For the 16×16 size block, compute the *SAD* with the PMV. If the *SAD* value is less than a threshold, *T*, the current macroblock will be decided to be 16×16-mode and the corresponding PMV acts as final MV without performing the following steps. Otherwise, the current macroblock will be regarded as a non-decided block, and go to Step 3.

According to the simulation results based on many sequences with *QP*=28, 32, 36, 40, we find that usually threshold *T* is in [450, 1000]. However, different sequences, even different frames within the same sequence lead to different thresholds. As a result, an adaptive threshold for each frame is introduced to handle this problem. It can be described as follow:

(1) Initialization: $T=AMAD=0$, $N=0$, where N denotes accumulated number of macroblock used mode-16×16.

(2) Repeat: For each macroblock, find the minimal *SAD*, $SAD_{\min}$, with the PMV. If $SAD_{\min}<T$, choose mode-16×16 as final block type and update *T* by the following steps:

$$AMAD=AMAD+SAD_{\min},\ N=N+1,$$
$$T=w[(AMAD+\alpha)/(N+\beta)], \quad (4)$$

where w is a weight to adjust the threshold while different *QP* is used, α and β are introduced to prevent a sharp alteration of *T* which may be caused by the sudden appearing of a small or large *SAD*. Usually, smaller *QP* causes smaller block types, such as 4×4

mode. Therefore, we set w to be 1.1, 1.2, 1.3, 1.4 for QP=28, 32, 36, 40 respectively. w should be larger than 1, otherwise T will never increase. α/β can be regarded as the initial threshold because the initial $AMAD$, N are 0. Fig.3 shows a trade-off between quality and efficiency. We set α=10000 and β=10.

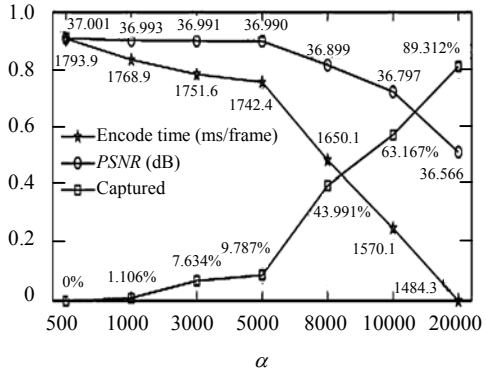

Fig.3 Performance of transcoding from 4CIF to CIF format under different α with N=10 and QP=28, the three components of y-axis are normalized

Experimental results (Fig.4) show that with our proposed pre-detection of 16×16-mode block algorithm, about 2/3 of the 16×16-mode blocks are captured.

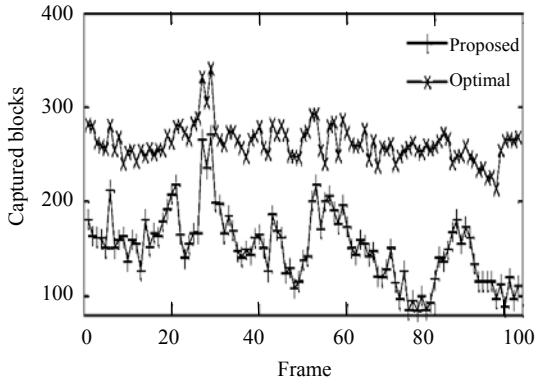

Fig.4 Average performance of 16×16-mode capture algorithm

Step 3: Bottom-up merging

Based on the achieved 4×4-block MVs, a merging procedure is taken to finish the mode decision procedure. The principle is simple but efficient: If the distance of neighboring MVs, pred1, pred2, is less than a threshold, T_{pred}, which means they are well aligned, merging will happen.

First of all, let us define the distance between two MVs as follows:

$$Dist_{v1,v2}=\max\{|MV1_x, MV2_x|, |MV1_y, MV2_y|\}, \quad (5)$$

where $MV1_x$, $MV1_y$, $MV2_x$ and $MV2_y$ denote the x and y components of vectors $\boldsymbol{MV}1$ and $\boldsymbol{MV}2$ respectively.

Fig.6 depicts the flowchart of the merging from 4×4-blocks to 8×8-blocks. With the PMVs for all size blocks estimated by Step 1, the bottom-up merging process can be described as follows. For the vertical and horizontal merging in Fig.5, while $Dist_{\text{pred1,pred2}}\leq T_{\text{pred}}$, merge the corresponding two blocks to an up-layer block; in particular, if $Dist_{\text{pred1,pred2}}=0$, let the MV of merged block be MV_{pred1} instead of the PMV taken from the proposed scheme in Step 1. If $Dist_{\text{pred1,pred2}}>T_{\text{pred}}$ in both of the vertical and horizontal merging processes, stop the merging and choose the current modes as the final block types (Status 1 in Fig.6); if both vertical and horizontal merging happen, these four 4×4 blocks will be merged into an 8×8 block without any additional checking (Status 3 in Fig.6).

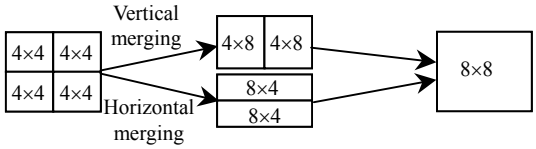

Fig.5 Merging process of 4×4 blocks to 4×8, 8×4 or 8×8 blocks

The merging from 8×8-blocks to 16×16-blocks is similar, only the decide-to-be-merged 8×8 blocks will participate in the following merging.

In order to determine the threshold T_{pred}, statistical analysis is conducted. Statistical analysis (Table 1) showed that if $Dist_{\text{pred1,pred2}}<3$, it is highly-probable that $MD<1$ pixel (MD is defined in Table 1), otherwise the probability is very low. Hence, we let T_{pred}=3, the unit is a pixel.

Table 1 Cumulative probability distribution of $MD=Dist_{\text{predictor,actualMV}}$ for merged blocks by full search with QP=28 and search range=[−16, 15] where $Dist_{\text{pred1,pred2}}$ denotes the distance of two MV predictors

Sequence	$Dist_{\text{pred1,pred2}}\leq 3$		$Dist_{\text{pred1,pred2}}>3$	
	$MD=0$	$MD\leq 1$	$MD=0$	$MD\leq 1$
News	89.38%	97.30%	1.46%	36.20%
M & D	69.73%	89.34%	1.24%	29.53%
Foreman	63.71%	91.47%	1.40%	32.21%
M & C	77.87%	97.01%	1.34%	38.76%
Coastguard	79.18%	97.01%	1.01%	34.90%

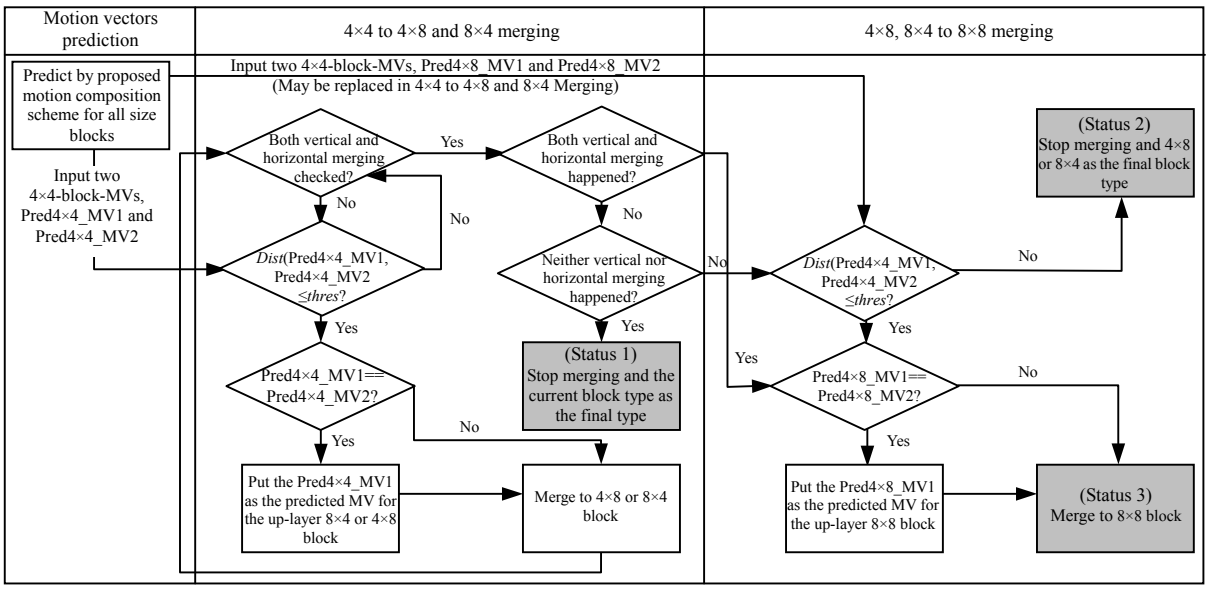

Fig.6 4×4 to 8×8 merging process flowchart, gray parts mean the end states

Step 4: Motion refinement

A half-pixel refinement is done on the merging results in Step 3. As in (Xin *et al*., 2002), it is enough to achieve a good quality.

EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, we calculated the PSNR between each down-sampled sequence and corresponding original decoded video down-scaled to the same resolution. Moreover, the transcoding speed performance of Cascaded-Decoder-Encoder method and our proposed algorithm will also be tabulated in Table 2.

As the sequences to be transcoded, the test sequences are pre-encoded with an open source MPEG-4 encoder (XVID) with 4CIF (704×576) resolution using the simple profile on 30 fps frame rate, only the first 50 frames are encoded. These encoded sequences will be transcoded into H.264 baseline profile by JM8.6 with different resolutions, SQCIF (128×96), QCIF (176×144) and CIF (352×288). The JM8.6 encoder setting is as follow: frame rate is 30 fps, search range is 16 and *QP*=28.

In Table 2, the testing sequences can be classified into three categories: video-conferencing-like sequences such as News and Akiyo which contain smooth motions with static backgrounds; sequences with global motions such as Coastguard; and sequences with zooming, distorted large motion such as Foreman. We achieved the performance about four times faster than the Cascaded-Decoder-Encoder transcoder (Fast Full Search) with average 0.09 *PSNR* drop and 2% bit rate increase.

Table 2 Comparison of PSNR, bitrate and transcoding efficiency performance between Cascaded and proposed algorithm. The source sequences are in 4CIF (704×576) format (the units of *PSNR* and Bit rate are dB, kbps respectively)

Sequence	Down to	Cascaded		Proposed		Speed up
		PSNR	Bit rate	*PSNR*	Bit rate	
Akiyo	128×96	37.15	27.93	37.08	27.63	4.23
	176×144	38.36	34.59	38.29	34.20	4.54
	352×288	39.79	102.95	39.71	102.93	5.01
News	128×96	35.75	63.03	35.64	63.41	3.71
	176×144	36.72	71.52	36.71	71.38	3.97
	352×288	38.18	202.99	38.09	206.99	4.41
Foreman	128×96	36.26	44.54	36.14	45.13	3.84
	176×144	35.47	112.98	35.39	114.50	3.62
	352×288	36.61	381.55	36.55	399.76	3.89
Coastguard	128×96	33.05	119.83	33.01	121.04	2.79
	176×144	33.82	481.21	33.72	483.82	3.17
	352×288	34.33	1335.21	34.30	1337.84	3.39

Fig.7 shows the bit rate comparison between proposed and cascaded algorithm on Foreman with the first 30 frames and *QP*=28. Figs.8 and 9 give the *PSNR* comparison. As Fig.9 shows, we find that the proposed works well on big *QP* values, such as *QP*=40.

Therefore, it can be concluded that our proposed algorithm can adapt to transcode with most kinds of down-sampled resolutions and motion scenarios.

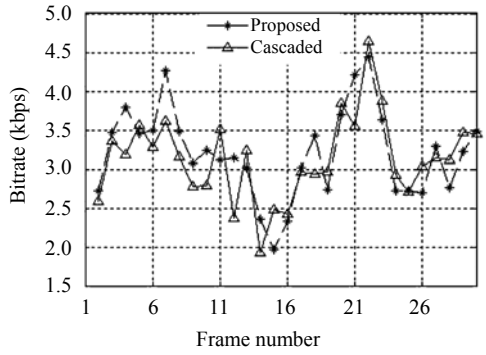

Fig.7 Bitrate comparison between Cascaded and proposed transcoder on the first 29 P-frames of Foreman

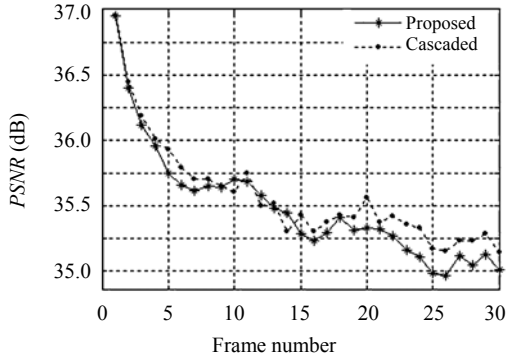

Fig.8 *PSNR* comparison between Cascaded and proposed transcoder on the first 30 frames of Foreman

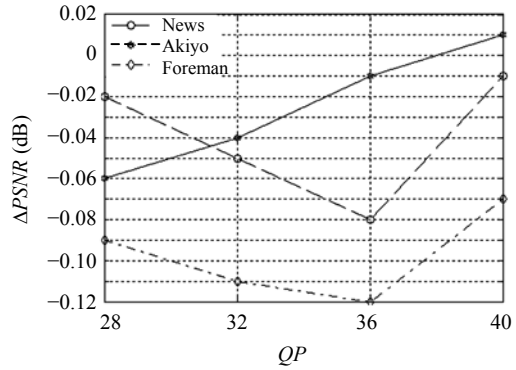

Fig.9 *PSNR* drop of the proposed algorithm compared to the Cascaded transcoder under different *QP* (First 30 frames are used)

CONCLUSION

In this paper, we proposed a fast mode decision algorithm for transcoding to H.264 with arbitrary rate spatial resolution down-scaling. With a PMV composition scheme to reuse the motion information in the input pre-encoded sequences, an early-stop on 16×16-block technique with adaptive threshold and a bottom-up merging processing are introduced to speed up the mode decision. Experimental results showed that the proposed algorithm can achieve a good trade-off between computational complexity and transcoding quality. The computational complexity can be reduced to about 25% of Cascaded-Decoder-Encoder method with Fast Full Search algorithm. *PSNR* drop and bit rate increase are 0.09 and 2%, respectively.

References

Ahmad, I., Wei, X.H., Sun, Y., Zhang, Y.Q., 2005. Video transcoding: An overview of various techniques and research issues. *IEEE Trans. on Multimedia*, **7**(5):793-804. [doi:10.1109/TMM.2005.854472]

Chang, A., Wong, P.H.W., Yeung, Y.M., Au, O.C., 2004. Fast multi-block selection for H.264 video coding. *Inter. Sym. Circuits and Systems*, **3**:817-820.

Kucukgoz, M., Sun, M.T., 2004. Early-stop and Motion Vector Re-using for MPEG-2 to H.264 Transcoding. Inter. Conf. SPIE-IS&T Electronic Imaging.

Shanableh, T., Ghanbari, M., 2000. Heterogeneous video transcoding to lower spatial-temporal resolutions and different encoding formats. *IEEE Trans. on Multimedia*, **2**(2):101-110. [doi:10.1109/6046.845014]

Takahashi, K., Satoch, K., Suzuki, T., Yagasaki, Y., 2001. Motion Vector Synthesis Algorithm for MPEG2-to-MPEG4 Transcoder. Visual Communications and Image Processing. San Jose, CA, p.872-882.

Tu, Y.K., Yang, J.F., Shen, Y.N., Sun, M.T., 2003. Fast variable size block motion estimation using merging procedure with an adaptive threshold. *Inter. Conf. Multimedia and Expo.*, **2**:789-792.

Xin, J., Sun, M.T., Choi, B.S., Chun, K.W., 2002. An HDTV-to-SDTV spatial transcoder. *IEEE Trans. on Circuits and Systems for Video Technology*, **12**(11): 998-1008. [doi:10.1109/TCSVT.2002.805508]

Zhou, Z., Sun, M.T., Hsu, S., 2004. Fast variable block-size motion estimation algorithms based on merge and split procedures for H.264/MPEG-4 AVC. *Inter. Sym. Circuits and Systems*, **3**:725-728.