Multimed Tools Appl
DOI 10.1007/s11042-007-0121-z

Effective algorithms for fast transcoding of AVS to H.264/AVC in the spatial domain

Zhihang Wang · Xiangyang Ji · Wen Gao · Qingming Huang · Debin Zhao

Abstract This paper proposes a transcoding scheme from AVS to H.264/AVC. As high-compression video coding standards, H.264/AVC jointly developed by MPEG and ITU and AVS developed by the Audio Video Coding Standard Working Group of China will co-exist in the future market. Therefore, it is worthy to transcode the AVS format to the H.264/AVC format or vice versa. After an insight into the inter transcoding from AVS to H.264/AVC, a simple and effective method is proposed by reusing the mode and motion vectors to achieve high-efficient and fast transcoding. The problem in reusing the skip mode is studied and an effective method to eliminate the artifacts is proposed. Furthermore, a fast intra transcoding algorithm based on the distribution of the DCT coefficients is proposed to speed up the transcoding process. Detailed experiment results demonstrate that the proposed algorithm can effectively reduce the transcoding complexity.

Keywords AVS · H.264/AVC · Transcoding · Mode mapping · Mode reuse · AVS to H.264/AVC

Z. Wang (✉) · X. Ji
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
e-mail: zhihangwang@jdl.ac.cn

X. Ji
e-mail: xyji@jdl.ac.cn

W. Gao
Peking University, Beijing, China
e-mail: wgao@jdl.ac.cn

Z. Wang · Q. Huang
Research Center of Digital Media, Graduate School of the Chinese Academy of Sciences, Beijing, China

Q. Huang
e-mail: qmhuang@jdl.ac.cn

D. Zhao
Department of Computer Science, Harbin Institute of Technology, Harbin, China
e-mail: dbzhao@jdl.ac.cn

1 Introduction

Video transcoding is becoming more and more important nowadays. On one hand, there exist several video coding standards such as MPEG-1, MPEG-2, MPEG-4 [10], H.264/AVC [11], AVS [18] and VC-1 [31]. So there will be a lot of video content compressed in different video formats. On the other hand, with the development of hardware, there are a variety of terminal devices such as PDA, Pocket PC, smart phone, mobile phone and many kinds of TV sets. It is important to transcode the video content from one format to another or transcode the video content from one resolution to another to adapt to these terminal devices. AVS (Audio Video Coding Standard) was developed by the Audio Video Coding Standard Working Group of China [18]. It provides almost the same video compression efficiency as that of H.264 [37]. Many companies have produced a lot of devices to support the AVS standard. There will be a lot of compressed video content in AVS format especially in China. Meanwhile, there are also a lot of devices supporting the H.264/AVC standard. So it is important to share the video content between these devices. Transcoding AVS to H.264 or vice versa is becoming a necessity.

The video transcoding technology is emerged with the development of the video coding standard [15, 23, 33]. Due to its importance in the application, many researchers studied this technology deeply. The computation complexity and the quality of the transcoded video are two important criteria for the video transcoder. The video transcoding technology can be mainly classified into two categories: homogeneous transcoding and heterogeneous transcoding [1].

The homogeneous transcoding technology focuses on bitrate reduction [5], spatial resolution downscaling [29, 30], temporal resolution reduction [3, 7], logo insertion [24, 25] and watermark insertion [22] as well as the combination of them within the same video format.

The heterogeneous transcoding technology also focuses on bitrate reduction [26], spatial resolution downscaling [44] and temporal resolution reduction [26]. The main difference from the technology of the homogeneous transcoding is that heterogeneous transcoding must consider the different features of the two video coding standards carefully to optimize the transcoder.

How to use the features of different standards efficiently is very important for heterogeneous transcoding. For example, the MPEG-2 standard is mainly used in the application of broadcast and storage [9], while the MPEG-4 standard is particularly designed to operate on mobile-wireless or satellite networks [10]. Thus transcoding MPEG-2 to MPEG-4 for the purpose of resolution reduction is needed. In [26, 44], the problems in heterogeneous transcoding were further studied.

With the emergence of the up-to-date video coding standard in recent years [11, 18, 31], there are new challenges for the transcoder. In the past a few years, many researchers investigated the problems in MPEG-2 to H.264/AVC transcoding. Issues in MPEG-2 to H.264/AVC transcoding were reviewed in [14]. Because H.264/AVC adopts the new intra prediction technology to improve the intra frame coding performance, improving the intra frame transcoding speed is a new problem. In [32], this problem was studied in the transform domain for MPEG-2 to H.264/AVC transcoding. Since H.264/AVC has adopted 4×4 and 8×8 transform that are different from that of MPEG-2. [27, 41] studied the problem of converting transform coefficients. While macroblocks of 16×16 are still used, motion block types of 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4 are also supported in H.264/AVC. So efficiently deciding the inter mode is very important in order to improve the transcoding speed. In [4, 19], different algorithms were proposed to speed up the mode decision process for MPEG-2 to H.264/AVC transcoding. Because H.264/AVC also supports the application in the mobile phone, low resolution video content is needed in this scenario. In [8], reducing resolution for MPEG-2 to H.264/AVC transcoding in the spatial domain was studied. Other problems, such as rate control, mode mapping, motion vector refinement and rate distortion optimization were also investigated in [6, 28, 42, 43, 47].

Besides these work, many researchers studied the problems in transcoding other standards such as H.263 [2], MPEG-4 [17], VC-1 [16] and AVS [38] to H.264/AVC. Because AVS can provide almost the same coding efficiency as that of H.264/AVC [35], transcoding from AVS to H.264/AVC at almost the same bitrate is needed as a new problem. It is important because many devices such as TV sets can only receive one format signal. When the programs are broadcasted in different format (AVS or H.264/AVC), transcoding can make the consumers choose from the two formats freely.

The basic part of AVS, AVS1-P2, targets at standard definition (SD) and high definition (HD) format video compression. The Fidelity Range Extensions (FRExt) of H.264/AVC [21] is also designed for the high-definition TV/DVD application. So transcoding the SD or HD bitstreams from AVS1-P2 to H.264 FRExt or vice versa is required. In [38], we proposed a fast intra transcoding algorithm. Besides the intra transcoding from AVS to H.264/AVC, inter transcoding is also very important. AVS adopts different motion compensation prediction techniques. The direct mode is different from that of H.264. The symmetry mode in B frames is a unique technology in AVS. The motion compensated interpolation, quantization and transform technologies are also different. These will cause new problems. However, to the best of the author's knowledge, little work has been reported on AVS to H.264/AVC transcoding considering problems caused by these technologies. In this paper, we focus on the application scenario as shown in Fig. 1.

By a thorough analysis of the difference between AVS and H.264, we propose the approaches to deal with these problems and report the corresponding experiment results in detail. The cascaded transcoding architecture [1] as shown in Fig. 2 will be used as the reference architecture. We use the close-loop transcoding architecture [1] as shown in Fig. 3 to evaluate the performance of the transcoder.

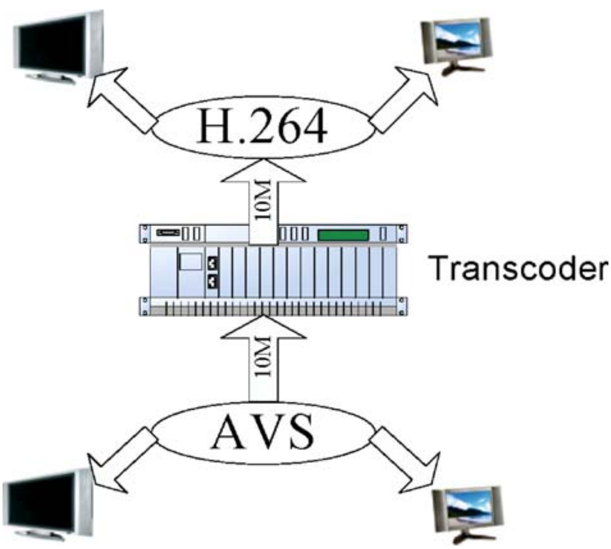


Fig. 1 Transcoding AVS to H.264 in digital TV applications

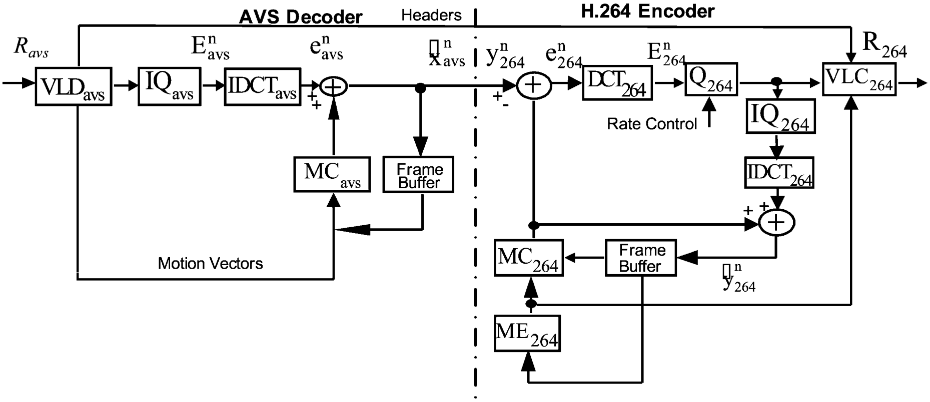


Fig. 2 Cascade transcoding architecture

This paper is organized as follows. The technologies used in AVS are briefly overviewed in Section 2. After analyzing the major different inter-prediction technologies of AVS and H.264/AVC and their impacts on the transcoder, we propose our approaches to deal with this transcoding problem in Section 3. The fast intra transcoding algorithm is provided in Section 4. Section 5 concludes this paper.

2 Overview of AVS

AVS is a new video coding standard developed by the Audio Video Coding Standard Working Group of China. Its aim is to provide technologies for the application from mobile TV to HDTV. AVS1-P2 video coding standard is the earliest and most important part of AVS. The mandate of AVS1-P2 is to provide the similar coding efficiency as that of H.264/AVC with lower complexity [35]. Many organizations, such as ISMA (Internet Streaming

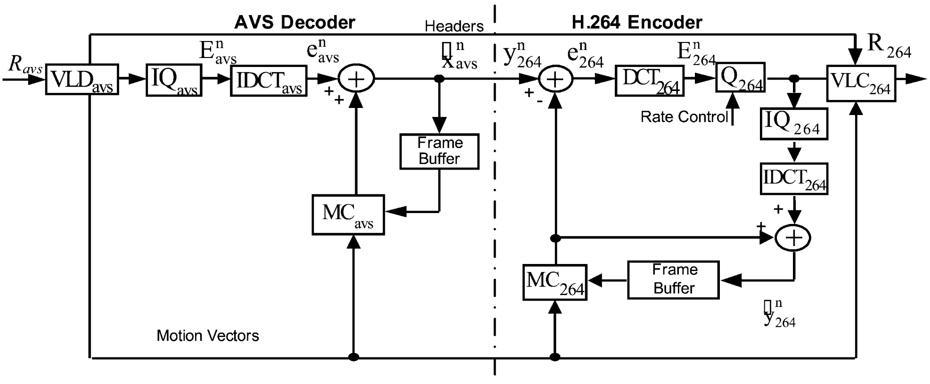


Fig. 3 Close-loop transcoding architecture

Media Alliance), decided to make AVS as an optional video compression standard in addition to H.264/AVC for IPTV. The framework of AVS is show in Fig. 4. We will introduce the AVS technology briefly. In [45], a detail overview of AVS is provided.

It can be seen from Fig. 4 that the AVS framework is similar to that of H.264/AVC [40]. But, because AVS targets at SD and HD application, it adopts many different technologies to reduce the complexity considering the characteristics of SD or HD video sequence.

2.1 Intra prediction technology

There are only five luma intra modes (Fig. 5), vertical, horizontal, down-left (Fig. 6), down-right (Fig. 7) and DC (Fig. 8), in AVS. The vertical and horizontal modes are the same as those of H.264/AVC.

The down-left (Fig. 6) mode uses the weighted average of diagonal-up three pixels and diagonal-left three pixels to predict the current pixel.

The prediction values of the down-right mode (Fig. 7) are obtained by weighting the up or left neighboring pixels. The up-diagonal pixels use the weighted average of diagonal up three pixels to predict the current pixel. The weighted average of the left-up corner three pixels is used to predict the diagonal pixels. The down-diagonal pixels use the weighted average of the diagonal-left neighboring three pixels as the prediction of the current pixel.

The prediction value of a pixel in DC (Fig. 8) mode is corresponding to the average of the weighted average of the up and left three pixels. The chroma intra prediction of AVS is the same as that of H.264/AVC except the DC mode. The chroma DC mode is the same as the luma DC mode. Because the resolution of SD or HD video sequence is very large, the block size used in AVS is only 8×8.

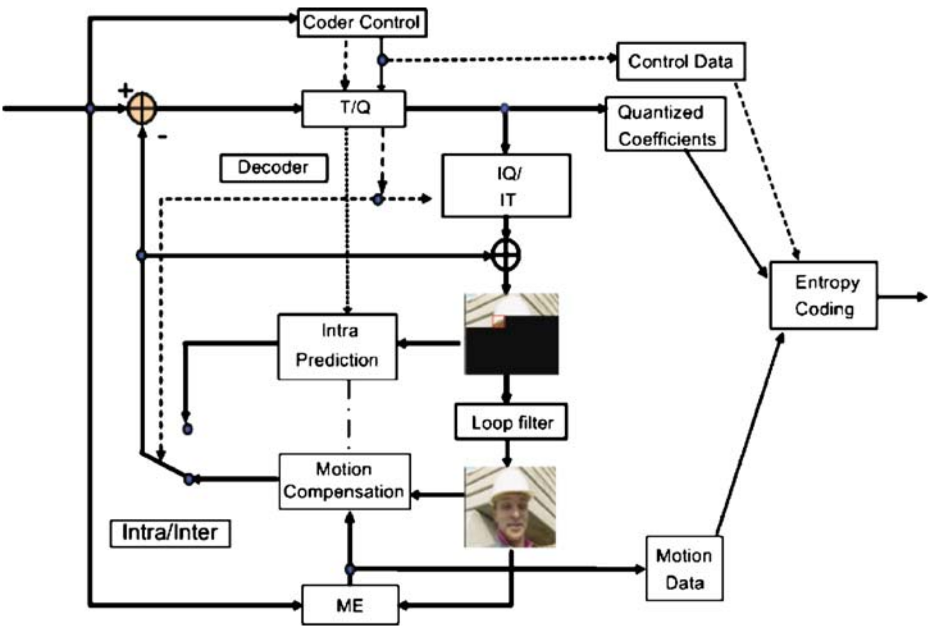


Fig. 4 AVS coding framework

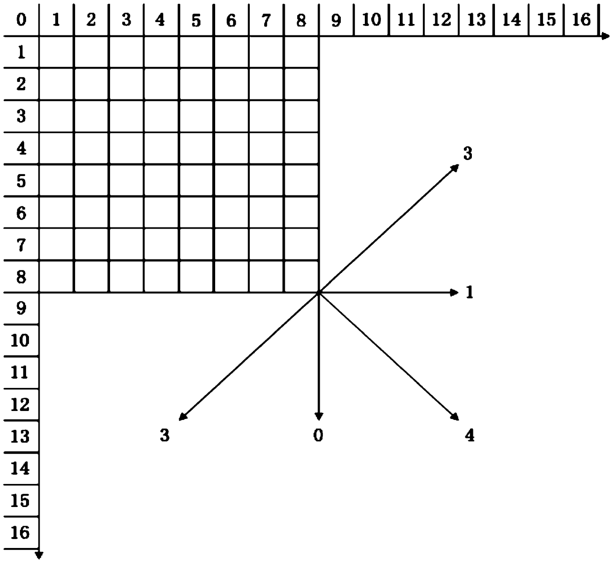


Fig. 5 AVS luma intra prediction modes

2.2 Transform and quantization in AVS

The forward transform matrix of AVS is

$$T8=\begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}.$$

AVS uses the 8x8 integer transform. The pre-scaled technology is used to remove the reverse scaling in the decoder side to reduce the decoder complexity [20, 46]. In order to use the integer transform, the forward and backward scaling technology integrated into the quantization process are used in H.264/AVC as shown in Fig. 9. If the inverse scaling process is shifted to the quantization process, the de-quantization table will become smaller. This is the basic idea of the pre-scaling technology shown in Fig. 10. Detailed description and analysis of this technology can be found in [20, 46].

AVS uses a scalar quantization. A total of 64 values of Quantization step (Qstep) are supported in AVS and they are indexed by a Quantization Parameter, QP, as that of H.264/AVC. Approximately, the Qstep doubles in size for every increment of 8 in QP. Because of the pre-scaling technology, the Qstep size only approximately changes according to this rule.

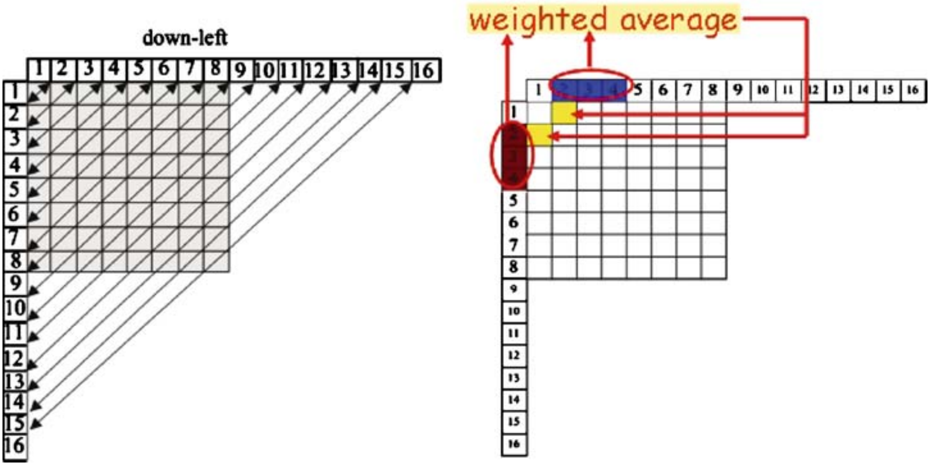


Fig. 6 AVS luma down-left mode

2.3 AVS inter technologies

AVS uses P (prediction) and B (bi-direction) frames. The max reference frame number for P is constrained to 2 for progressive coding. Only one forward and one backward reference frame can be used to predict the B frame. The motion vector prediction is different from that of H.264/AVC [11]. The algorithm of median motion vector prediction is different. The principle of the median motion vector prediction in AVS is to calculate the center of gravity of the triangle formed by the up, up-right and left motion vector. Detailed discussion on the motion vector prediction technology will be given in Section 3.4.

The motion compensated interpolation uses two 4-tap filters to reduce the spatial complexity [36]. The half-sample positions are obtained by applying a one-dimensional 4-tap FIR filter (−1, 5, 5, −1) horizontally and vertically. Prediction values at quarter-sample positions except the pixels at e, g, p, r positions (Fig. 11) are generated by another 4-tap filter (1, 7, 7, 1). The prediction values at e, g, p, r positions are achieved in the same way as that of H.264/AVC.

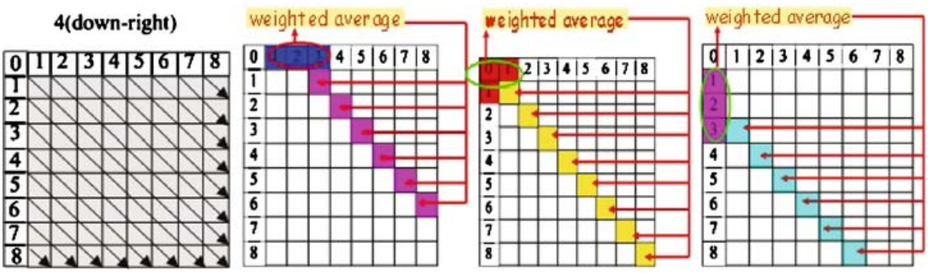
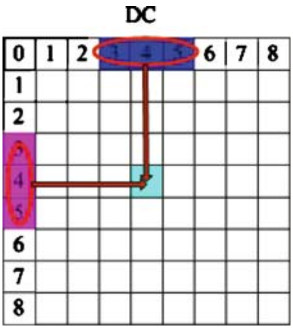


Fig. 7 AVS luma down-right mode

Fig. 8 AVS luma DC mode



The principle of the AVS direct mode in B frame is the same as that of H.264/AVC [34]. But the AVS direct mode adopts a more accurate scaling technique and provides a better coding efficiency [13]. The bi-directional prediction technology in AVS adopts the symmetry mode [12]. The backward motion vector of this mode is calculated by extending the forward motion vector in the backward direction.

The detailed description of other technologies, such as entropy coding and deblocking, can be found in [35, 37, 39]. In the following, we will present a detailed analysis of the transcoding problems encountered in transcoding from AVS to H.264/AVC in the spatial domain.

3 Analysis of inter frame transcoding

The experiment in this paper is done under the following condition. The total encoded frame number is 100 for all the experiment. When the experiment is only about P frame, the skip frame number is 2. If the experiment is about the coding efficiency of B frame, the GOP structure is IBBPBBPBBPIBBPBBPBBP.... If the experiment is only about intra frame, the skip frame number is 0. We test many video sequences. The performances are alike each other. Only the BigShips sequence results are given to reduce the paper length.

3.1 QP mapping

Because there are 64 quantization levels in AVS and only 52 quantization levels in H.264/AVC, if the AVS QP is the same as that of H.264/AVC, the Qstep is not the same. However, we can adjust the H.264/AVC QP so that the Qstep is about the same as that of AVS. The

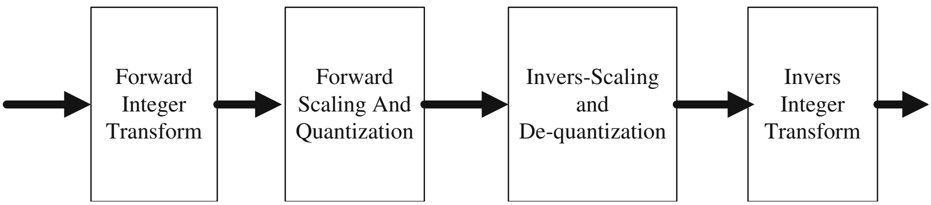


Fig. 9 Block diagram of Integer Transform and Quantization scheme in H.264/AVC

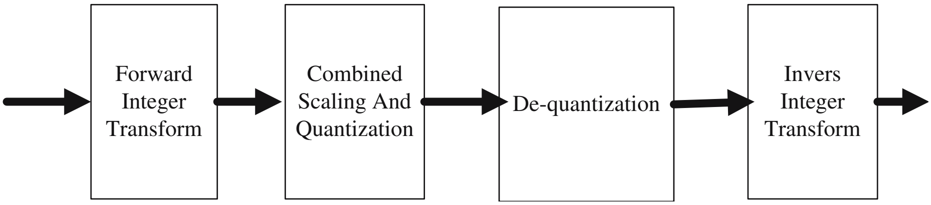


Fig. 10 Block diagram of the pre-scaling integer transform and quantization scheme

AVS Qstep is calculated as follows:

$$Qstep_{avs} \approx \left(2^{(QP_{avs}//8)}\right) * QP2QSTEP_{avs}[QP_{avs}\%8] \tag{1}$$

where $QP_{avs}//8 = \lfloor QP_{avs}/8 \rfloor$, $QP_{avs}\%8$ is the remainder of $QP_{avs}/8$. We use ≈ instead of = because of the pre-scaling technology used in AVS [20]. The variable $QP2QSTEP_{avs}$ contains the initial AVS Qstep values when $QP_{avs}<8$ is true.

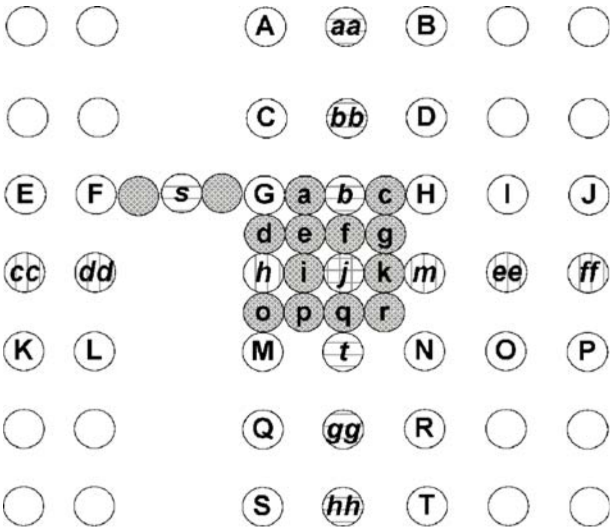
$$QP2QSTEP_{avs} = \{1.0, 1.0905, 1.189, 1.297, 1.414, 1.542, 1.682, 1.834\}$$

Similarly, the Qstep in H.264/AVC is computed as formula (2).

$$Qstep_{264} = \left(2^{(QP_{264}//6)} * QP2QSTEP_{264}[QP_{264}\%6]\right. \tag{2}$$

where $QP2QSTEP_{264}$={0.625,0.6875,08125,0.875,1,1.125}. All the Qsteps of AVS and H.264/AVC can be calculated according to (1) and (2). The Appendix shows all AVS and H.264/AVC Qsteps. In order to transcode AVS to H.264/AVC at about the same bitrate, we can choose the H.264/AVC QP that minimizes the Qstep distance between AVS and H.264/

Fig. 11 Filtering for fractional-sample accurate motion compensation. *Upper-case letters* indicate samples on the full-sample grid, while *lowercase samples* indicate samples in between at fractional-sample positions



AVC so that the H.264 Qstep is almost the same as that of AVS. For example, when the AVS QP is equal to 28, the corresponding H.264/AVC QP should be 25, not 24 or 26.

3.2 Mode mapping

Because the resolution of the SD or HD video sequence is relatively large, the inter mode under 8×8 block size, such as 8×4, 4×8, 4×4 can be disabled to reduce the complexity with little loss in the coding efficiency. Although we can't know how the AVS bitstream is optimized, we can suppose that it is encoded by the rate optimization technology used in the reference software. We also suppose that the full motion estimation algorithm is used. We disabled the skip mode now to make the problem clear because the motion vector of the skip mode is different. Detailed analysis of the impact of the skip mode will be given in Section 3.4.

If $QStep_{avs} \approx Qstep_{264}$, the quantization error of AVS and H.264 will be almost the same. So $R_{avs} \approx R_{264}$ is reasonable in the rate distortion theory. Although this is not always true in the reality because of the technique limitations used in the standard, their impact on the coding efficiency is not very large. The experiment results (Fig. 12) show that the loss in the coding efficiency caused by reusing the mode can be neglected.

3.3 Motion vector mapping

In Section 3.2, we know that if the AVS motion vectors are the same as those of H.264/AVC, mode can be approximately reused. We will analyze the motion vector in this section to show that the motion vector can also be reused. It is straight forward that the integer pixel motion vector can be reused. It is interest to see the impact of the different interpolation filter used in AVS and H.264/AVC. For the half-sample and quarter-sample position pixel values, if the impact of the corresponding filter can also be omitted, the half-pixel or quarter-pixel motion vector will also can be reused. For simplicity, we assume the images that will be filtered by the AVS and H.264/AVC filters are the same.

3.3.1 Half-pixel motion vector

The AVS filter that is used to calculate the half-sample position pixel value is

$$F_{avs} = (-1, 5, 5, -1)/8 \tag{3}$$

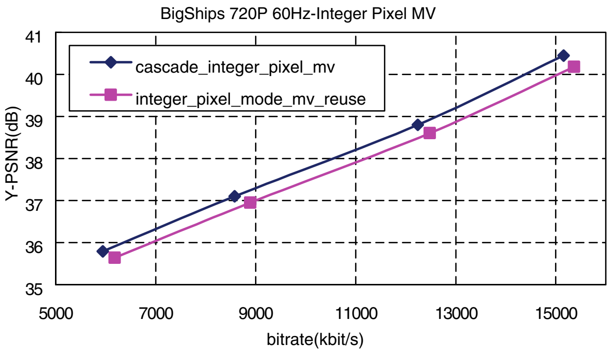


Fig. 12 RD curve for reusing mode and integer pixel motion vector (AVS QP=28)

While the corresponding H.264/AVC filter is

$$F_{264} = (1, -5, 20, 20, -5, 1)/32 \quad (4)$$

For every half-sample position pixel value, the difference between AVS and H.264/AVC can be calculated directly from (3) and (4). For example, as shown in Fig. 11, the AVS half-pixel value at position b is

$$b_{avs} = \frac{-F + 5*G + 5*H - I}{8} \quad (5)$$

while the H.264/AVC half-pixel value at position b is

$$b_{264} = \frac{E - 5*F + 20*G + 20*H - 5*I + J}{32} \quad (6)$$

The difference is

$$\theta = b_{avs} - b_{264} = \frac{-E + F + I - J}{32} \quad (7)$$

It is clear that if $-E + F + I - J$ is less than 32, θ will be equal to zero. This will be true when the texture of the image is relatively smooth. So the half-pixel motion vector can be reused in the smooth area. In the worst case θ will lead a difference of 16, but this will not occur frequently in the real world. So the coding performance will not loss significantly. In the experiment results shown in Fig. 13, it can be seen that the coding performance loss by reusing the half-pixel motion vector is not very large.

3.3.2 Quarter-pixel motion vector

The quarter-sample position values of AVS are calculated in two ways. The first way is to use the filter (8) to calculate the quarter-pixel value. The pixels except those at e, g, p, r positions are generated using the following filter

$$\mathrm{FQ}_{\mathrm{avs}} = (1, 7, 7, 1)/16 \quad (8)$$

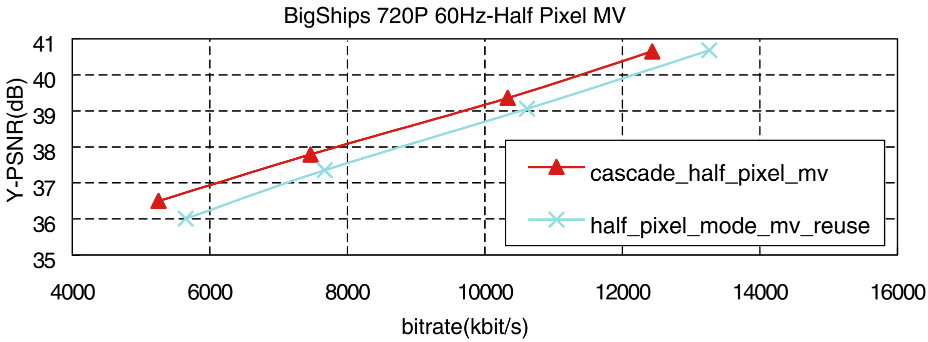


Fig. 13 RD curve for reusing mode and half pixel motion vector (AVS QP=28)

while the corresponding H.264/AVC filter is

$$FQ_{264} = (1,1)/2 \tag{9}$$

Because the AVS half-pixel values are almost the same as those of H.264/AVC, we can prove that the difference at the quarter-sample position can also be omitted. For example, the AVS pixel value at position a is

$$a_{avs} = \frac{s_{avs} + 7^*G + 7^*b_{avs} + H}{16} \tag{10}$$

while the H.264/AVC pixel value at position a is

$$a_{264} = \frac{G + b_{264}}{2} \tag{11}$$

Because $b_{avs} = b_{264} + \theta$, we have

$$\eta_1 = a_{avs} - a_{264} = \frac{s_{avs} - G - b_{avs} + H}{16} + \frac{\theta}{2} \tag{12}$$

The other way is to use the filter (9) to calculate the quarter-pixel values locating at e, g, p, r.

$$e_{avs} = \frac{G + j_{avs}}{2} \tag{13}$$

$$e_{264} = \frac{G + j_{264}}{2} \tag{14}$$

The difference between e_{avs} and e_{264} is

$$\eta_2 = e_{avs} - e_{264} = \frac{j_{avs} - j_{264}}{2} \tag{15}$$

From (12) and (15) we can see that the half-pixel difference is halved. Because the coding performance loss caused by the half-pixel difference is not very large, the loss in the quarter-pixel position will also not very large. Experiment results in Fig. 14 also show that the coding performance loss can be neglected.

The computation complexity can be reduced greatly by reusing the mode and motion vectors. Traditional algorithm needs examine at least four modes (16×16, 16×8, 8×16 and 8×8). Now it is not necessary to do mode decision. The motion estimation process which costs the most encoder time is also bypassed.

3.4 Skip mode transcoding

In Sections 3.2 and 3.3, the skip mode is disabled in order to show that other modes and motion vectors can be reused. In this section, we will discuss the algorithm that disposes the skip mode.

The coding efficiency of the skip mode is very high because no coefficients need to be encoded. The motion vector for the skip mode is equal to the prediction motion vector of the current block. If the motion vector prediction technology is different, the skip mode motion vector will also be different. Thus reusing the skip mode will cause significant loss in the reconstructed video quality, especially when the percentage of the skip mode is large.

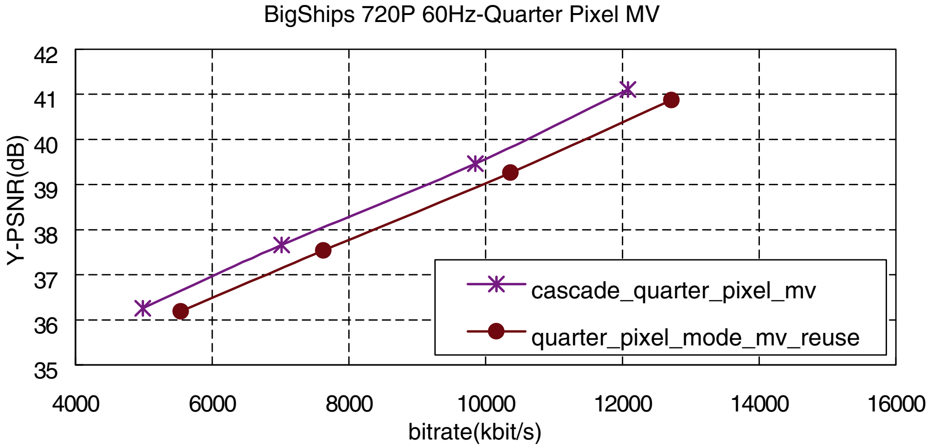


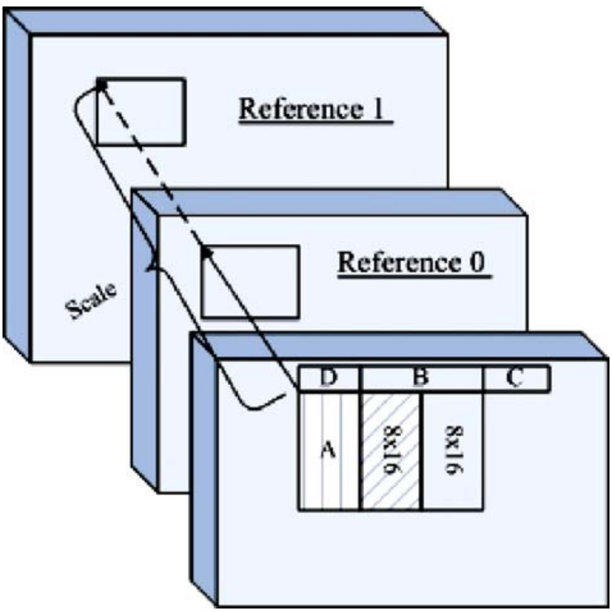
Fig. 14 RD curve for reusing mode and quarter pixel motion vector (AVS QP=28)

The AVS motion vector prediction technology is different from that of H.264/AVC in the following case:

1) The reference frame of the current block is different from the neighboring block reference frame

AVS will scale the neighboring block motion vector according to the picture distance in this case as shown in (Fig. 15). The reference frame of the left block is reference frame 0, while the reference frame of the current block is reference frame 1. Based on the assumption that the motion of the left block is linear, the motion vector of the left block in

Fig. 15 AVS prediction motion vector scaling technology



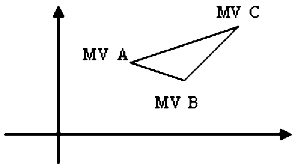


Fig. 16 AVS median motion vector prediction

AVS will be scaled according to the distance between these two reference frames to obtain the motion vector of the left block in reference frame 1. There is no scaling technology in H.264/AVC. So if AVS selects the skip mode in this case, the motion vector will be different from that of H.264/AVC. Simply reusing the skip mode is not enough now.

2) The motion vector prediction type is median

In this case that the AVS median prediction algorithm is different from that of H.264/AVC, the neighboring block motion vectors compose a triangle (Fig. 16). The edge whose length is middle in this triangle will be selected. Then the vertex opposite to this edge is the selected vertex. The motion vector corresponding to this vertex is the prediction motion vector of the skip mode.

Due to the difference of the skip mode motion vector, reusing mode and motion vectors will cause loss in the coding performance as shown in Figs. 17 and 18.

The impact of the skip mode is proportional to the skip mode number. The skip number for every experiment is shown in Table 1. It is clear that because the percentage of the skip mode number in the BigShips sequence is about 30% when the AVS QP is 32, the impact of the skip mode is very severe (Fig. 18). Sometimes there will even be some artifacts if we only reuse the skip mode as shown in Fig. 19. These artifacts will cause the drifting problem. It will cause significant coding efficiency loss. The reason why simply reusing the skip mode will cause some artifacts is illustrated in Fig. 20. When the AVS skip mode

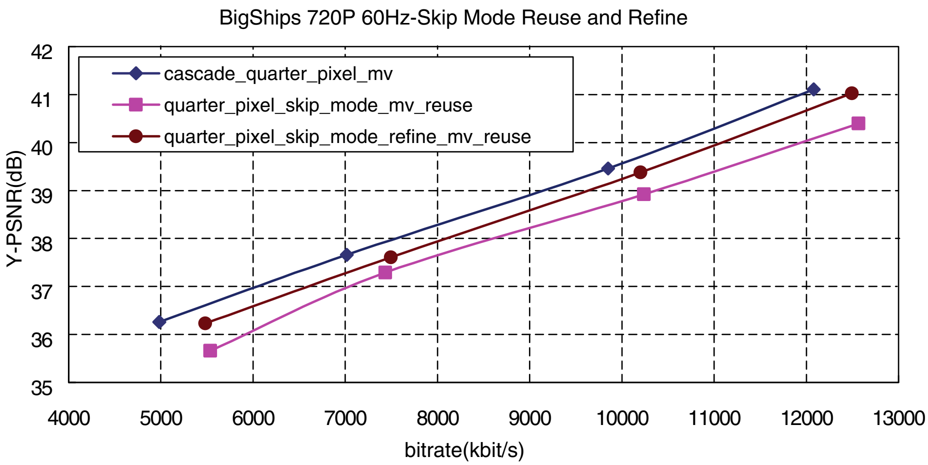


Fig. 17 Reusing all modes and refine the skip mode (AVS QP=28)

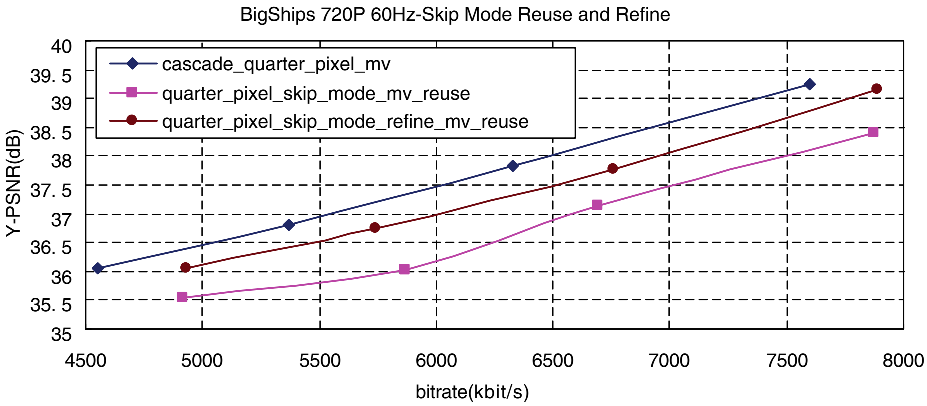


Fig. 18 Reusing all modes and refine the skip mode (AVS QP=32)

motion vector is different from that of H.264/AVC, the prediction macroblock will be different. Because there are not any coefficients in the skip mode, the artifacts will occur.

While for the other sequence such as harbour, there are only about 6% of the total mode is skip mode, simply reusing the skip mode doesn't cause significant loss in the coding performance. But it is also probable to cause the artifacts. Nevertheless, the impact of the skip mode can't be ignored, especially when the value of QP is relatively big because there will be more skip mode.

We propose to use the AVS skip mode vector and the INTER16×16 mode as a candidate mode in addition to the skip mode when the AVS skip mode vector is different from that of H.264/AVC. Because the macroblock uses the AVS skip mode motion vector, the prediction of this macroblock should be relatively well. Thus the coding efficiency of the INTER16×16 mode for this macroblock will not decrease very much. The increase in the complexity can be ignored. Figs. 17 and 18 show the experiment results. It is obvious that refining the skip mode can improve not only the objective video coding performance but also the subjective visual quality as shown in Fig. 19.

For the skip mode, only one additional mode (16×16) needs to be checked. Traditional algorithm needs to check 5 modes (skip mode, 16×16, 16×8, 8×16, 8×8). The mode decision algorithm can speed up about 60%. Because the motion estimation process is omitted, this can further reduce the computation complexity.

3.5 Symmetry mode transcoding

The bi-direction mode in AVS is the symmetry mode. The backward motion vector can be derived from the forward motion vector [12]. Its main advantage is saving backward motion vector bits with little coding efficiency loss when the motion of the object in the sequence is

Table 1 The skip mode number in BigShips

AVS QP	Skip mode number	Total inter mode	Percentage (%)
28	56033	270000	21
32	83065	270000	31

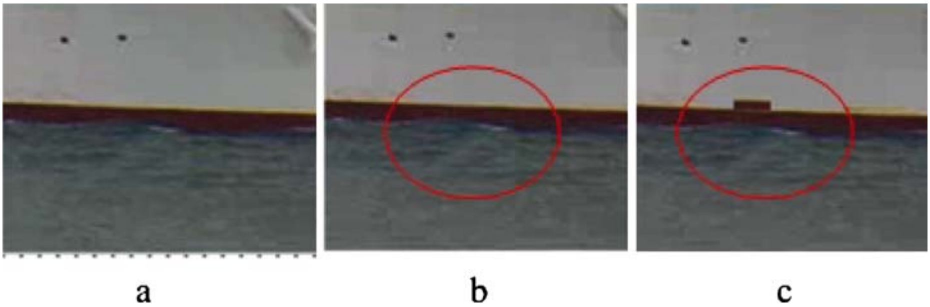


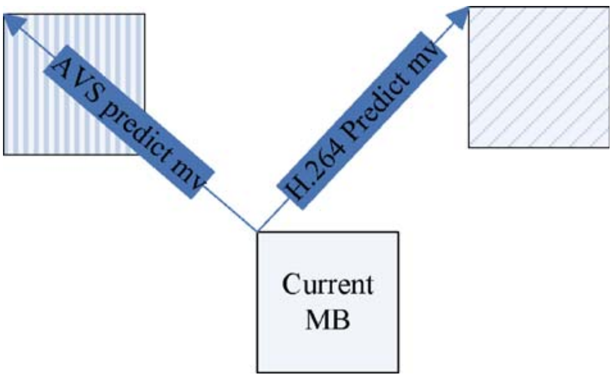
Fig. 19 Subjective quality comparison of the third frame in BigShips when AVS QP is 32 **a** AVS reconstruction **b** skip mode refine **c** only reuse skip mode

linear. Usually the frame rate of the SD or HD video sequence is very high, the motion between two frames is smooth, and thus the symmetry mode can provide a good coding efficiency. In the AVS to H.264 transcoder, when the AVS mode is symmetry mode, we reuse the forward and backward motion vector. Figure 21 shows the experiment results. We disabled the direct mode in the experiment. It is obvious that reusing the motion vector is good enough. The reason why reusing the B frame motion vector will work is the same as that of P frame.

3.6 Direct mode transcoding

The principle of the AVS direct mode is the same as that of H.264/AVC. Because the scaling technology in AVS is more accurate, the coding efficiency of the AVS direct mode is higher than that of the H.264/AVC direct mode [13]. But the direct mode is different from the skip mode. There are not any coefficients in the skip mode, while there may be some coefficients in the direct mode. So, although the skip mode may cause some significant artifacts due to the different motion vector, the direct mode would not cause significant artifacts due to the motion vector difference. Experiment results (Fig. 22) show that reusing

Fig. 20 Comparison of AVS skip mode and H.264 skip mode



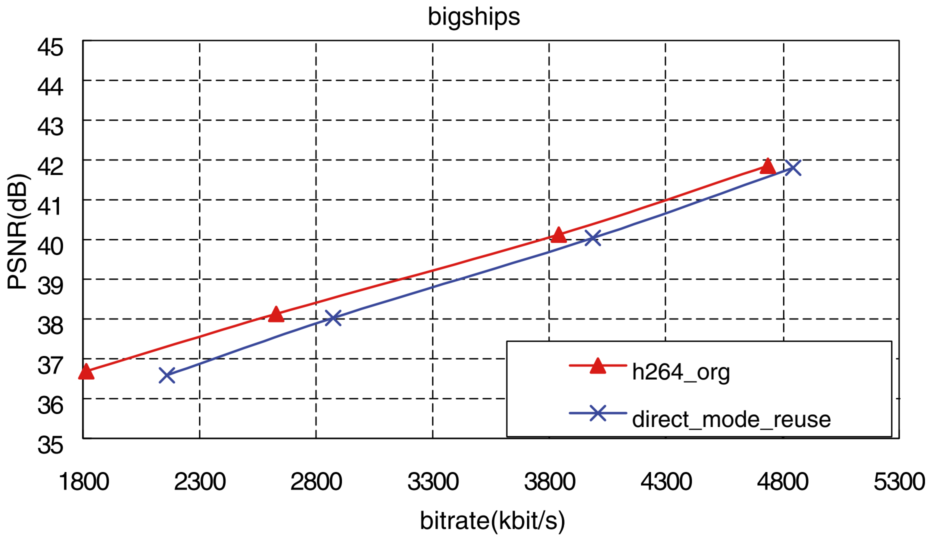


Fig. 21 Reuse AVS symmetry mode and motion vector(AVS QP=28)

the direct mode will cause about 0.2 dB loss in the coding efficiency. In terms of transcoding applications, this can be ignored.

4 Intra transcoding technology

Because the motion estimation and mode decision process for the P and B frame can be omitted, the intra frame becomes more complex than the inter frame. The percentage of the intra frame encoding time in one GOP is shown in Table 2 and Table 3. It is obvious that optimizing the intra frame is important. There are five INTRA8×8 luma modes and four INTRA8×8 chroma modes in AVS, while there are nine INTRA4×4, nine INTRA8×8, four INTRA16×16 luma modes and four INTRA8×8 chroma modes in the H.264/AVC

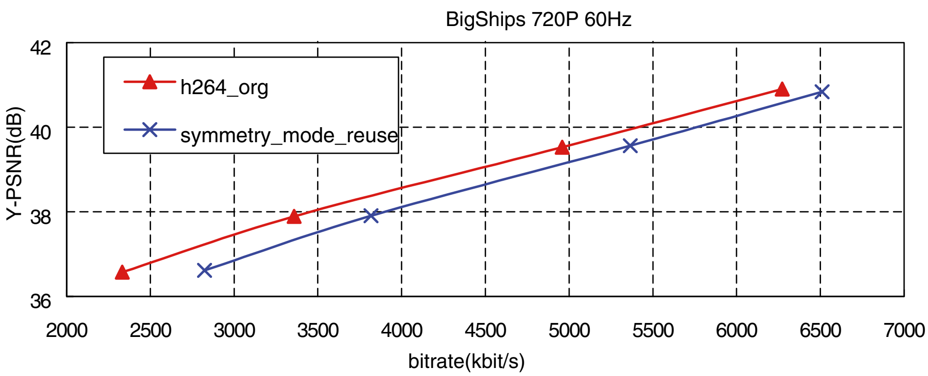


Fig. 22 Reuse AVS direct mode and motion vector (AVS QP=28)

Table 2 Intra frame encoding time when AVS QP is 28 and H.264 QP is 25

Sequence	Intra frame time (ms)	Total time (ms)	Intra/total time (%)
Bigships	22719	31749	72
Harbour	37328	53344	70
Night	34437	50109	69
Sheriff	19250	28186	68

FREx profile. We can use the AVS intra mode information to speed up the H.264 intra mode decision process. The principle for the fast intra mode decision algorithm is mode reuse. The AVS DCT coefficients contain plentiful information for selecting the optimal H.264/AVC intra mode. Here we first use the distribution of the AVS DCT coefficients to decide whether a macroblock should be encoded as intra 16×16 or intra 8×8 or intra 4×4 mode. Then we use the mode and residual information to further speed up the intra mode decision process. The basic scheme of this algorithm is shown in Fig. 23.

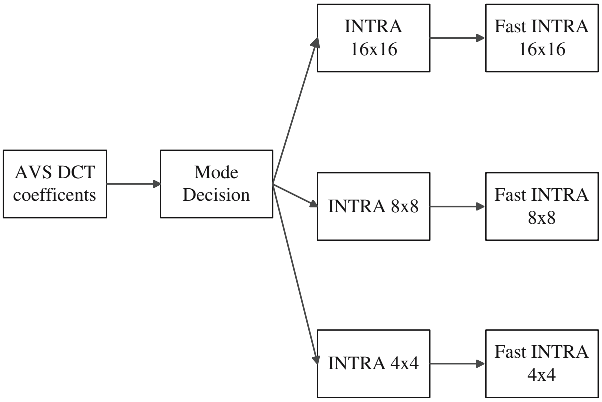
4.1 AVS DCT coefficients distribution

The AVS DCT coefficients contain a lot of information. We can decide the mode depending on the distribution of the DCT coefficients. If the macroblock is smooth, perhaps only the DC coefficient is not zero. Its mode is preferred to intra 16×16. While if there are a lot of high frequency components in this macroblock, intra 4×4 mode maybe provide a good coding performance. We use the coefficients distribution character as shown in Fig. 24 to decide what mode this macroblock should be encoded. Because of quantization, more high frequency components will be quantized to zero especially when the QP is relatively large. So there will be many 8x8 blocks whose DCT coefficients will like Fig. 24. We classify the DCT coefficient distribution to 4 categories: UP_ROW_NOT_ZERO, LEFT_COLUMN_NOT_ZERO, DC_ALL_ZERO and NOT_REGULAR_COEFFICIENTS. If only the coefficients of the up row of one 8x8 block are not zero, this block is called UP_ROW_NOT_ZERO block. Similarly, the block whose coefficients are all zero except the left column coefficients is called LEFT_COLUMN_NOT_ZERO. If all the coefficients are zero or only the DC coefficient is not zero, the block is called DC_ALL_ZERO. Other blocks fall into the NOT_REGULAR_COEFFICIENTS category. Table 4 shows the coefficients distribution proportion. Because the *shuttelstart* sequence is relatively smooth, about 70% of the 8x8 block can be used to help us to speed up the intra mode decision process.

Table 3 Intra frame encoding time when AVS QP is 32 and H.264 QP is 28

Sequence	Intra frame time (ms)	Total time (ms)	Intra/total time (%)
Bigships	30344	42469	71
Harbour	33703	48359	70
Night	33125	48359	68
Sheriff	30406	44251	69

Fig. 23 Block diagram of the fast intra mode decision algorithm



If the four 8×8 blocks of a macroblock have the same DCT coefficient distribution, it means that this macroblock is smooth, so this macroblock will be encoded as intra 16×16 mode. While if every 8×8 block of a macroblock is NOT_REGULAR_COEFFICIENTS block, this macroblock should be encoded as intra 4×4 mode. If not all but at least one 8×8 block is UP_ROW_NOT_ZERO or LEFT_COLUMN_NOT_ZERO or DC_ALL_ZERO block, this macroblock should be encoded as an intra 8×8 block.

Now we will use the AVS mode and the residual energy information to further speed up the intra mode decision process.

Fig. 24 The AVS 8×8 block DCT coefficients distribution

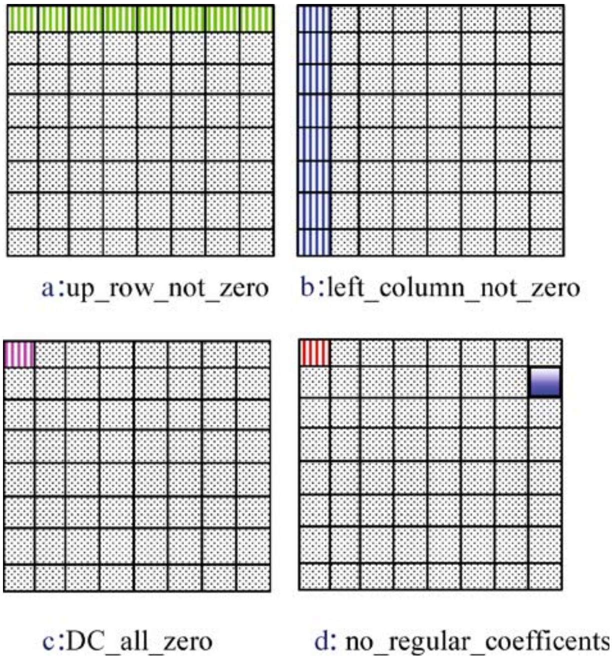


Table 4 The AVS DCT coefficients distribution

Sequence	Qp	Up_row_not_zero	Left_column_not_zero	DC_all_zero	Total8×8num	Percentage (%)
BigShips	20	214	205	76	57600	1
	28	2497	2791	3878	57600	16
	32	3138	5500	8173	57600	29

4.2 Fast luma intra 16×16 mode decision

We use the AVS 8×8 block mode information to predict the intra 16×16 mode. First, if the four 8×8 block modes of AVS are different from each other, it means that this macroblock is difficult to predict. Thus we will choose the four intra 16×16 modes exhaustively to decide which mode should be used. Second, if the four 8×8 block modes are the same, the intra 16×16 mode depends on the intra 8×8 modes. If the intra 8×8 mode is vertical or horizontal or DC mode, it shows the high correlation relationship in that direction. So we can predict that this macroblock should choose this intra prediction mode. If the four 8×8 block modes are all down-left, we will only choose the plane prediction mode for H.264/AVC because it means it is effective to predict the macroblock using the filtered up and left pixels. If the four 8x8 block modes are all down-right, we will select the DC prediction mode for H.264/AVC. Finally, if only some of the 8×8 block mode are the same, we will choose all the modes selected by the four 8×8 blocks for this macroblock as the candidate mode. It should be noted that if the 8×8 block selects the down-left mode, we will choose the plane mode. If all the four 8×8 blocks select the down-right mode, we will choose the DC mode for the macroblock.

4.3 Fast luma intra 8×8 mode decision

If the AVS 8×8 mode is vertical, horizontal or down-right mode, we reuse this mode. If the AVS 8×8 mode is down-left mode, there is not a corresponding mode in H.264/AVC. But the diagonal-down-left, horizontal-up and vertical-left mode is similar to the AVS down-left mode. So we choose these three modes as the candidate mode.

4.4 Fast luma intra 4×4 mode decision

The intra 4×4 mode provides good coding efficiency when the high frequency component of the macroblock is relatively much more because there will only have high correlation in a small

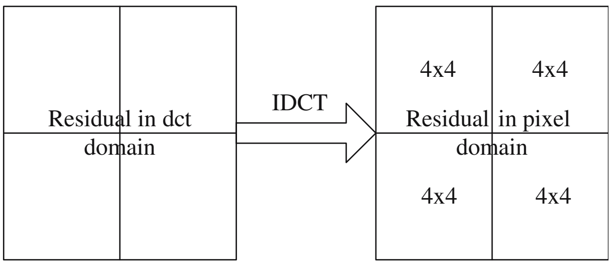
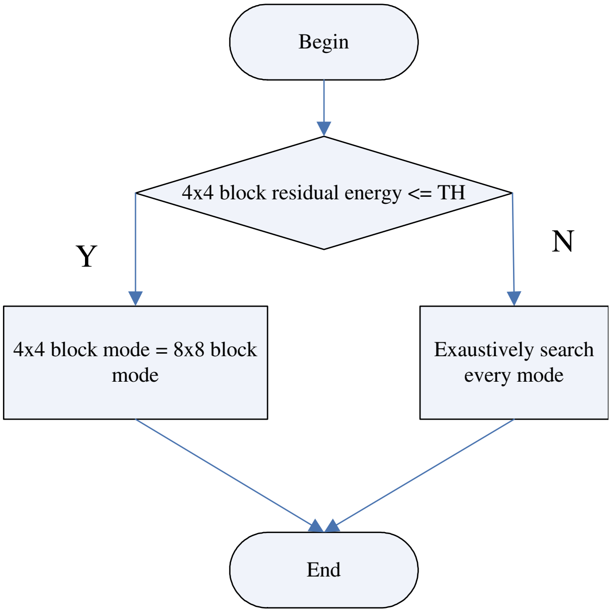


Fig. 25 Get the 4×4 block SAD

Fig. 26 Choose 4×4 block mode when the 8×8 block selected vertical or horizontal mode



region. Because there is not intra 4×4 mode in AVS, the energy of the residual of the 8×8 block will be relatively large when there is much high frequency. But the energy of one or two 4×4 block maybe be very small and the other 4×4 block energy is very big. We can get the residual energy of every 4×4 block as shown in Fig. 25. If the energy of one 4×4 block is very small, it means that the coding efficiency of that mode is very well. In this case we can reuse the AVS block mode. We still use the exhaustively mode decision algorithm to decide the other 4×4 block intra modes. The flow chart of this algorithm is shown in Fig. 26. One problem is how to decide the threshold so that the loss in the coding efficiency is little. The quality of the intra frame is very important because it will serve as the reference frame for P and B frame. So we use another technology to deal with this problem, i.e. using the coefficients of this 4×4 block to decide whether or not to reuse the AVS intra mode. If the coefficients of this 4×4 block are all zero, we will reuse the AVS intra mode. Otherwise, we will search all modes to decide the final mode.

Table 5 Intra 4×4 candidate mode

8×8 block mode	Intra 4×4 mode index			
	Block0	Block1	Block2	Block3
Down-left	3,2	3,7	3,8	3,2
Down-right	2,4	2,4	2,4	2,4

Table 6 Chroma intra mode mapping

AVS chroma mode	H.264 chroma mode
Vertical	Vertical,DC
Horizontal	Horizontal,DC
DC	Plane,DC
Plane	Plane,DC

To reduce the complexity, we calculate the threshold according to the quantized DC coefficient (Q(DC)) of a 4×4 block. If Q(DC) is equal to zero, usually the energy of the AC coefficients is relatively small and can be ignored. So

$$\mathrm{TH} = \sum_{i}^{4}\sum_{j}^{4} r(i,j) \tag{16}$$

$$\mathrm{DC} \approx \frac{\sum_{i}^{4}\sum_{j}^{4} r(i,j)}{16} \tag{17}$$

$$Q(\mathrm{DC}) = \mathrm{DC}/\mathrm{QStep}_{264} = 0 \tag{18}$$

We can get

$$\mathrm{TH} \leq 16 \times \mathrm{QStep}_{264} \tag{19}$$

If the 8×8 block select the down-left or down-right mode, we will deal with the four 4× 4 blocks with different strategy because there is not a corresponding mode in H.264/AVC.

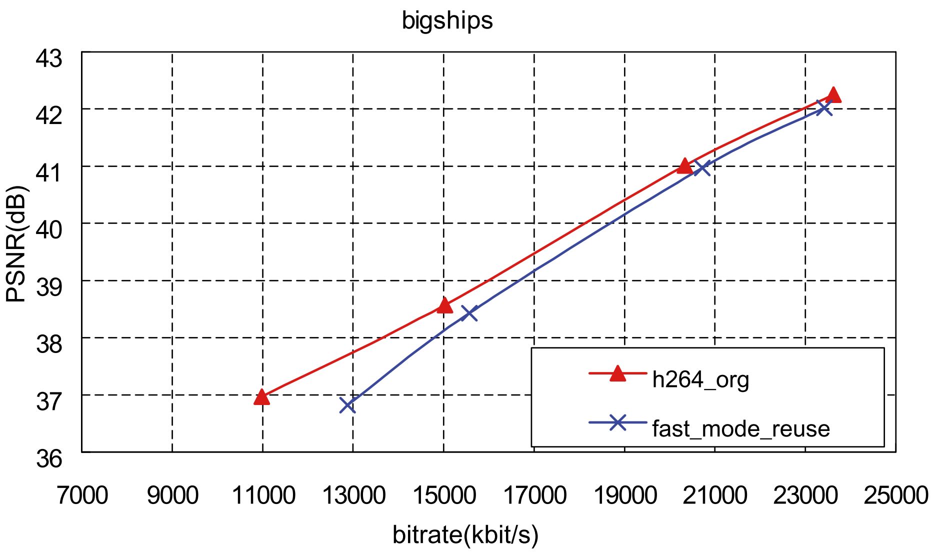


Fig. 27 The performance of the fast intra mode decision algorithm (AVS QP=20)

Table 7 Complexity reduction of the fast intra transcoding algorithm

Sequence	Qp	Org_time(ms)	Fast_time(ms)	Speedup (%)
Bigships	20	99.421	42.626	57
		95.516	40.469	58
		92.219	39.203	57
		87.234	36.937	58
	28	108.67	26.813	75
		104	26.11	75
		102.734	25.719	75
		99.782	24.36	76

Table 6 shows the candidate modes when the 4×4 block residual energy is smaller than or equal to the threshold. The intra 4×4 mode index in Table 5 is identical to [11].

If the 8×8 block selects the DC mode, we will exhaustively search all the candidate modes because there is not any direction information.

4.5 Fast chroma intra 8×8 mode decision

The chroma signal is smooth compared with the luma signal. The chroma modes in these two video standards are almost the same except the DC mode. So it is not necessary to search the entire chroma mode. Simply reusing the chroma mode will cause loss in the coding efficiency because the luma rate distortion character will change. We propose to refine the chroma mode according to Table 6.

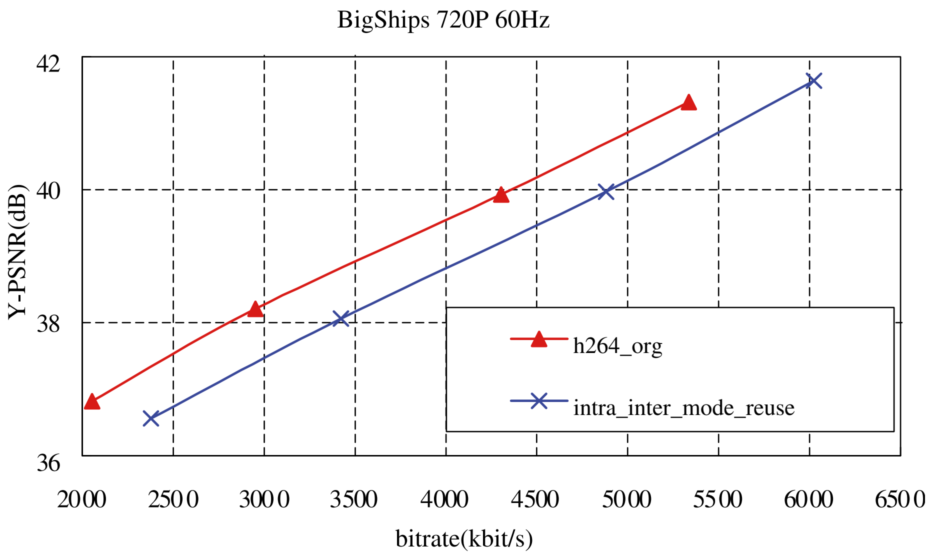


Fig. 28 The performance of the fast intra and inter mode decision algorithm (AVS QP=28)

The performance of the proposed fast intra transcoding algorithm is shown in Fig. 27. Table 7 shows the complexity reduction of the fast intra transcoding algorithm. It is obvious that the complexity can reduce to about 60% or higher.

The overall performance of the proposed algorithm is shown in Fig. 28. The loss in the coding performance is relatively large compared to the other figures. The reason is that the coding efficiency loss of intra and p frame will affect the coding performance of B frame. Especially when there are many skip mode in the P frame and many direct mode in the B frame, the coding performance loss may up to 0.5 dB.

5 Conclusions

In this paper, some problems in inter and intra transcoding from AVS to H.264/AVC are studied. We provide a deep analysis of the feasibility of reusing the AVS mode and motion vectors during transcoding. Because of the different characteristics of the skip mode, it will cause some artifacts. We analyze the reason and provide an effective method to overcome this problem. Because reusing the mode and motion vectors can provide a fairly good coding efficiency, the complexity of intra transcoding is becoming the bottleneck. We propose a fast intra transcoding algorithm to reduce the computational cost of intra transcoding. Detailed experiments show that this algorithm can significantly reduce the complexity during transcoding.

Acknowledgment This work has been supported by Key Technologies for High Efficient Video Coding under the contact no. 60333020, Key Technologies for Dealing with Streaming Media under the contact no. 4041003 and Special Foundation of President of The Chinese Academy of Sciences under Grant no. 20064020

Appendix

All AVS Qsteps

QP	0	1	2	3	4	5	6	7
Qstep	1	1.0905	1.189	1.297	1.414	1.542	1.682	1.834
QP	8	9	10	11	12	13	14	15
Qstep	2	2.181	2.378	2.594	2.828	3.084	3.364	3.668
QP	16	17	18	19	20	21	22	23
Qstep	4	4.362	4.756	5.188	5.656	6.168	6.728	7.336
QP	24	25	26	27	28	29	30	31
Qstep	8	8.724	9.512	10.376	11.312	12.336	13.456	14.672
QP	32	33	34	35	36	37	38	39
Qstep	16	17.448	19.024	20.752	22.624	24.672	26.912	29.344
QP	40	41	42	43	44	45	46	47
Qstep	32	34.896	38.048	41.504	45.248	49.344	53.824	58.688
QP	48	49	50	51	52	53	54	55
Qstep	64	69.792	76.096	83.008	90.496	98.688	107.648	117.376
QP	56	57	58	59	60	61	62	63
Qstep	128	139.584	152.192	166.016	180.992	197.376	215.296	234.752

All H.264 Qsteps

QP	0	1	2	3	4	5	6	7
Qstep	0.625	0.687	0.812	0.875	1	1.125	1.25	1.375
QP	8	9	10	11	12	13	14	15
Qstep	1.625	1.75	2	2.25	2.5	2.75	3.25	3.5
QP	16	17	18	19	20	21	22	23
Qstep	4	4.5	5	5.5	6.5	7	8	9
QP	24	25	26	27	28	29	30	31
Qstep	10	11	13	14	16	18	20	22
QP	32	33	34	35	36	37	38	39
Qstep	26	28	32	36	40	44	52	56
QP	40	41	42	43	44	45	46	47
Qstep	64	72	80	88	104	112	128	144
QP	48	49	50	51				
Qstep	160	176	208	224				

References

1. Ahmad I, Wei X, Sun Y, Zhang Y-Q (2005) Video transcoding: an overview of various techniques and research issues. IEEE Trans Multimedia 7:793–804
2. Bialkowski J, Barkowsky M, Kaup A (2006) Overview of low-complexity video transcoding from H.263 to H.264. In: International Conference on Multimedia & Expo
3. Chen M-J, Chu M-C, Pan C-W (2002) Efficient motion-estimation algorithm for reduced frame-rate video transcoder. IEEE Trans Circuits Syst Video Technol 12:269–275
4. Chen G, Zhang Y-d, Lin S-x, Dai F (2004) Efficient block size selection for MPEG-2 to H.264 transcoding. In: Proceedings of the 12th Annual ACM International Conference on Multimedia, pp 300–303
5. Eleftheriadis A, Anastassiou D (1995) Constrained and general dynamic rate shaping of compressed digital video. In: IEEE International Conference on Image Processing, Washington, DC, pp 396–399
6. Fernandez-Escribano G, Kalva H, Cuenca P, Orozco-Barbosa L (2006) RD-optimization for MPEG-2 to H.264 transcoding. In: International Conference on Multimedia & Expo, pp 309–312
7. Fung K-T, Chan Y-L, Siu W-C (2002) New architecture for dynamic frame-skipping transcoder. IEEE Trans Image Process 11:886–900
8. Hu B, Zhang P, Huang Q, Gao W (2005) Reducing spatial resolution for MPEG-2 to H.264/AVC transcoding. In: IEEE Pacific Rim Conference on Multimedia, pp 830–840
9. Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) (2000) Information technology-generic coding of moving pictures and associated audio information: video (MPEG-2 video), ISO/IEC 13818-2, 2nd ed
10. Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) (2001) Coding of audio-visual objects-Part 2: visual (MPEG-4 video), ISO/IEC 14 496-2:2001, 2nd ed
11. Int. Telecommun. Union-Telecommun. (ITU-T) (2003) Draft text of final draft international standard for advanced video coding. Geneva, Switzerland, Recommendation H.264 (draft)
12. Ji X, Zhao D, Gao W, Huang Q, Ma S, Lu Y (2004) New bi-prediction techniques for B pictures coding. In: IEEE International Conference on Multimedia and Expo, 1:101–104
13. Ji X, Zhao D, Gao W, Lu Y, Ma S (2004) New scaling technique for direct mode coding in B pictures. In: International Conference on Image Processing, pp 469–472
14. Kalva H (2004) Issues in H.264/MPEG-2 video transcoding. In: Consumer Communications and Networking Conference (CCNC), pp 657–659
15. Keesman G, Hellinghuizen R, Hoeksema F, Heideman G (1996) Transcoding of MPEG bitstreams. Signal Process, Image Commun 8:481–500
16. Lee J-B, Kalva H (2006) An efficient algorithm for VC-1 to H.264 video transcoding in progressive compression. In: International Conference on Multimedia & Expo, pp 53–56
17. Lee Y-K, Lee S-S, Lee Y-L (2006) MPEG-4 to H.264 transcoding using macroblock statistics. In: International Conference on Multimedia & Expo, pp 57–60
18. Liang F, Ma S, Wu F (2004) Overview of AVS video standard. In: IEEE International Conference on Multimedia and Expo, 1:423–426

19. Lu X, Tourapis AM, Yin P, Boyce J (2005) Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding. In: IEEE International Symposium on Circuits and Systems, pp 1246–1249
20. Ma SW, Gao W (2006) Low complexity integer transform and adaptive quantization optimization. J Comput Sci Technol 21:354–359
21. Marpe D, Wiegand T, Gordon S (2005) H.264/MPEG4-AVC fidelity range extensions: tools, profiles, performance, and application areas. In: IEEE International Conference on Image Processing, pp I-593–6
22. Meng J, Chang S-F (1998) Embedding visible video watermarks in the compressed domain. In: International Conference on Image Processing (ICIP'98), pp 474–477
23. Nakajima Y, Hori H, Kanoh T (1995) Rate conversion of MPEG coded video by re-quantization process. In: International Conference on Image Processing (ICIP'95) p 3408
24. Panusopone K, Chen X, Ling F (2001) Logo insertion in MPEG transcoder. In: Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing, Salt Lake City, UT, pp 981–984
25. Roma N, Sousa L (2002) Insertion of irregular-shaped logos in the compressed DCT domain. In: Proc. 14th Int. Conf. Digital Signal Processing (DSP), pp 125–128
26. Shanableh T, Ghanbari M (2000) Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats. IEEE Trans Multimedia 2:101–110
27. Shen B (2004) From 8-tap DCT to 4-tap integer transform for MPEG to H.264/AVC transcoding. In: International Conference on Image Processing, pp 115–118
28. Shen B (2006) Optimal requantization-based rate adaptation for H.264. In: International Conference on Multimedia & Expo, pp 317–320
29. Shen B, Sethi IK, Vasudev B (1999) Adaptive motion-vector resampling for compressed video downscaling. IEEE Trans Circuits Syst Video Technol 9:929–936
30. Song J, Yeo B-L (1999) Fast extraction of spatially reduced image sequences from MPEG-2 compressed video. IEEE Trans Circuits Syst Video Technol 9:1100–1114
31. Srinivasan S, Hsu PJ, Holcomb T, Mukerjee K, Regunathan SL, Lin B, Liang J, Lee M-C, Ribas-Corbera J (2004) Windows Media Video 9: overview and applications. Signal Process, Image Commun 19:851–875
32. Su Y, Xin J, Vetro A, Sun H (2005) Efficient MPEG-2 to H.264/AVC intra transcoding in transform-domain. IEEE Int Symp Circuits and Syst (ISCAS) 2:1234–1237
33. Sun H, Kwok W, Zdepski JW (1996) Architectures for MPEG compressed bitstream scaling. IEEE Trans Circuits Syst Video Technol 6:191–199
34. Tourapis AM, Wu F, Li S (2005) Direct mode coding for bipredictive slices in the H.264 standard. IEEE Trans Circuits Syst Video Technol 15:119–126
35. Wang X-F, Zhao D-B (2006) Performance comparison of AVS and H.264/AVC video coding standards. J Comput Sci Technol 21:310–314
36. Wang R, Huang C, Li J, Shen Y (2004) Sub-pixel motion compensation interpolation filter in AVS. In: IEEE International Conference on Multimedia and Expo, 1:93–96
37. Wang Q, Zhao D, Ma S, Lu Y, Huang Q, Gao W (2004) Context-based 2D-VLC for video coding. In: IEEE International Conference on Multimedia and Expo, 1:89–92
38. Wang Z, Gao W, Zhao D, Huang Q (2006) A fast intra mode decision algorithm for AVS to H.264 transcoding. In: International Conference on Multimedia & Expo, pp 61–64
39. Wang Q, Zhao D, Gao W (2006) Context-based 2D-VLC Entropy Coder in AVS Video Coding Standard. J Comput Sci Technol 21:315–322
40. Wiegand T, Sullivan GJ, Bjntegaard G, Luthra A (2003) Overview of the H.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13:560–576
41. Xin J, Vetro A, Sun H (2004) Converting DCT coefficients to H.264/AVC transform coefficients. In: Pacific-Rim Conference on Multimedia (PCM), pp 939–946
42. Xin J, Vetro A, Sekiguchi S-i, Sugimoto K (2006) Motion and mode mapping for MPEG-2 to H.264/AVC transcoding. In: International Conference on Multimedia & Expo, pp 313–316
43. Yang J, Dai Q, Xu W, Ding R (2005) A rate control algorithm for MPEG-2 to H.264 real-time transcoding. In: Proc. SPIE., pp 1995–2003
44. Yin P, Vetro A, Liu B, Sun H (2002) Drift compensation for reduced spatial resolution transcoding. IEEE Trans Circuits Syst Video Technol 12:1009–1020
45. Yu L, Yi F, Dong J, Zhang C (2005) Overview of AVS-video: tools, performance and complexity. In: Proc. SPIE., pp 679–690
46. Zhang CX, Lou J, Yu L, Dong J, Cham WK (2005) The technique of pre-scaled integer transform. IEEE Int Symp Circuits Syst 1:316–319
47. Zhou Z, Sun S, Lei S, Sun M-T (2005) Motion information and coding mode reuse for MPEG-2 to H.264 transcoding. IEEE Int Symp Circuits Syst 2:1230–1233

Zhihang Wang Received the M.S. degree in the Ocean University of Qingdao, 2001. He is a PhD candidate at the Institute of Computing technology, Chinese Academy of Science. His research interests include video coding and video transmission and open source project.

Xiangyang Ji Received the MS degree in Harbin Institute of Technology, China, 2001. He is a PhD candidate at the Institute of Computing technology, Chinese Academy of Science. His research interests include video coding and video transmission.

Wen Gao (M'99) received the PhD degree in computer science from Harbin Institute of Technology, China, in 1988 and the PhD degree in electronics engineering, University of Tokyo, Japan, in 1991. He was a research fellow at the Institute of Medical Electronics Engineering, University of Tokyo, in 1992; a visiting scientist at the Robotics Institute, Carnegie Mellon University, in 1993; a visiting scientist at the MIT AI Lab, from May 1994 to December 1995. Now he is a professor of Peking University and the Institute of Computing Technology, Chinese Academy of Sciences and an adjunct professor in Hong Kong University of Science and Technology, China. He is the head of the Chinese National Delegation to MPEG working group (ISO/SC29/WG11) and chairs the Audio Video coding Standard (AVS) workgroup of China. His research interests include pattern recognition, artificial intelligence, image understanding, data compression, hand gesture recognition, multimodal interface, and computer vision. He has published four books and over 400 scientific papers.

Qingming Huang became a Member (M) of IEEE in 2004. He obtained his BS in computer science in 1988 and PhD in computer engineering in 1994, both from Harbin Institute of Technology, China. He was a postdotoral fellow in the National University of Singapore from 1995 to 1996, and a member of research staff in Institute for Infocomm Research, Singapore from 1996 to 2002. He joined the Chinese Academy of Sciences (CAS) in 2003 under the Science100 Plan, and is now a professor in the Graduate School of CAS. His research interests include multimedia computing, video transcoding, digital video analysis, image processing, pattern recognition and computer vision. He has published over 80 academic papers in various journals and conference proceedings, and holds/files six patents in US, Singapore and China.

Debin Zhao received the BS, MS and PhD degrees in computer science from the Harbin Institute of Technology, China, in 1985, 1988, and 1998, respectively. He was a research fellow in the Department of Computer Science, City University of Hong Kong, from 1989 to 1993. And now he is with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include data compression, image processing, and human–machine interface. He has published two books and over 60 scientific papers.