Overview of AVS Video Standard

Liang Fan¹, Siwei Ma², Feng Wu³

¹) Dept. Elec. & Comm. Engineering, Sun Yat-sen University, Guangzhou 510275, PRC
²) Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, PRC
³) Microsoft Research Asia, Beijing 100080, PRC
E-mail: isslf@zsu.edu.cn, swma@ict.ac.cn, fengwu@microsoft.com

Abstract

This paper overviews the AVS video standard in terms of basic features, adopted major techniques and its performance. The current version of AVS video standard mainly aims at the increasing demand on high definition and high quality video services. It provides a good trade-off solution between complexity and coding efficiency for digital broadcast and digital storage media. Furthermore, similar to the syntax structure in MPEG-2 video standard, it can be easily applied into the extensively existing MPEG-2 systems while significantly improving coding efficiency.

1. Introduction

AVS video standard is developed by the Audio Video Coding Standard Working Group of China (AVS working group in short), which was approved by the Chinese Science and Technology Department of Ministry of Information Industry in June 2002. The mandate of the AVS working group is to establish the China’s national standards for compression, manipulation and digital right management in digital audio and video multimedia equipment and systems.

AVS working group has finished the first version of AVS video standard in December 2003 [1], and expects to be approved as a national standard in 2004. This version mainly targets at high definition and high quality digital broadcasting, digital storage media and other related applications. It can be also applied in existing standard definition applications.

This paper is organized as follows. Section 2 introduces the basic architecture of AVS video codec and its features. Section 3 describes some key techniques adopted in AVS video standard. Section 4 explains the profile and the levels defined in the current AVS video specification. Finally, the experimental results are given in Section 5.

2. Architecture of AVS video codec and its features

Figure 1 depicts the block diagram of AVS video encoder. Due to the same modules, the architecture of AVS looks like that of H.264 [2]. However, considering the target applications, backward compatibility with MPEG-2 and decoding complexity, the technique of the AVS video codec in every module is more and less different from that used in H.264.
mapping tables are defined to map coded symbol into a special codebook and its elements.

The sum of prediction and current reconstructed error image forms the reconstructed reference. AVS uses the de-blocking filter in motion compensation loop. The de-blocking process directly acts on the reconstructed reference first across vertical edges and then across horizontal edges. Obviously, different image regions and different bit rates need different smooths. Therefore, the de-blocking filter is automatically adjusted in AVS depending on activities of blocks and QP parameters.

Since MPEG-2 codec and system are extensively deployed in the existing broadcast systems, the syntax structure of AVS is specially designed to be similar to that of MPEG-2 [3]. The similarity enables AVS to be readily applied to the MPEG-2 system. Some start codes are defined in AVS to indicate the beginning of a sequence, a picture and a slice respectively in a stream; however, there is no group of pictures (GOP) header indicated in AVS. The absolute time information can be contained in the I picture header by setting a flag as 1. Therefore, I pictures enable random access to compressed stream and switches among programs. To distinguish I picture from P and B pictures, the start code of I picture is different from that of P and B pictures.

In addition, AVS defines a new start code of video_edit_code to indicate whether the first consecutive B pictures immediately following the coded I picture can be reconstructed properly in the case of a random access. Currently, AVS supports 4:2:0 and 4:2:2 sampling structure and 8-bit sample precision.

3. The key techniques in AVS video

The main techniques adopted in the AVS codec are briefly discussed in this section.

3.1. Entropy coding

As mentioned above, AVS uses k-th-order Exp-Golomb codebook (k = 0, 1, 2, 3). CBP, macroblock coding mode, motion vectors, and so on, are decoded with 0th-order Exp-Golomb codebook. The quantized transform coefficients are decoded with one of the four codebooks. Firstly, an initial codebook is selected for the first non-zero quantized coefficient. Afterward, the absolute value of the current decoded coefficient decides which codebook is used for the next non-zero quantized coefficient. In order to efficiently map the coded symbols to the elements of Exp-Golomb codebooks, 19 mapping tables are defined in AVS.

Because of the regularization of the Exp-Golomb codebooks, the AVS decoder does not need to store these codebooks. The syntax element can be decoded with a simple parse followed by an optional look-up table. Although the 19 mapping tables only occupy less than 2k bytes, they provide a strong adaptation to different distributions and high coding performance.

3.2. Transform and quantization

Unlike H.264 and MPEG-2, AVS uses an 8×8 integer transform. In order to reduce the rounding errors in the dequantization and the inverse transform, they are considered as a process. The operations can be completed within 16 bits.

For an 8×8 block, the decoded levels \( x'_{ij} \) (i=1..7, j=1..7) are dequantized with the following equation

\[
x_{ij} = (x'_{ij} \times a(QP) + 2^{s(QP)-1}) \gg s(QP)
\]

where \( QP \) is the quantization step, \( a(QP) \) is the inverse quantization table and \( s(QP) \) is the varied shift value for inverse quantization. The range of \( x_{ij} \) is \([-2^{15}, 2^{11} - 1]\). \( x_{ij} \) is the dequantized coefficient.

The horizontal inverse transform is performed as

\[
H' = X \times T^T
\]

where \( T \) is the transform matrix as given in Figure 2, \( T^T \) is the transpose matrix of \( T \), \( X \) is the 8×8 dequantized coefficient matrix and \( H' \) is the intermediate result after the horizontal inverse transform. In order to guarantee 16-bit operation, every element of \( H' \) adds 4 and right-shifts 3 bits to form the new matrix \( H'' \):

\[
T_s = \begin{bmatrix}
8 & 10 & 10 & 9 & 8 & 6 & 4 & 2 \\
9 & 4 & -2 & -8 & -10 & -10 & -6 \\
8 & 6 & -4 & -10 & -8 & 2 & 10 & 9 \\
8 & 2 & -10 & -6 & 8 & 9 & -4 & -10 \\
8 & -2 & -10 & 6 & 8 & -9 & -4 & 10 \\
8 & -6 & -4 & 10 & -8 & -2 & 10 & -9 \\
8 & -9 & 4 & 2 & -8 & 10 & -10 & 6 \\
8 & -10 & 10 & -9 & 8 & -6 & 4 & -2
\end{bmatrix}
\]

Figure 2. The transform matrix in AVS.

The vertical inverse transform is described as follows

\[
Y = T \times H'^T
\]

The element \( y_{ij} \) of \( Y \) should be in the range of \([-2^{15}, 2^{15} - 1]\). The final decoded residue \( r_{ij} \) is calculated as

\[
r_{ij} = (y_{ij} + 2^5) \gg 7
\]

3.3. Intra-frame prediction

AVS video standard uses intra-frame prediction to improve the performance of intra coded macroblocks. The intra-frame prediction in AVS is conducted for each 8×8 luma/chroma block in the spatial domain. Compared with AVC/H.264’s 9 intra
prediction modes for 4×4 luma block, 4 modes for 16×16 luma block and 4 modes for 4×4 chroma block, AVS defines 5 modes for 8×8 luma block and 4 modes for 8×8 chroma block.

3.4. Reference pictures

In previous video coding standards, such as MPEG-1 [4] and MPEG-2 [3], bi-directionally predictively-coded pictures (B pictures) use one previous picture and/or one future picture as references. Although predictively-coded pictures (P pictures) use only one previous picture to predict the current picture, the actual reference buffer size in a decoder has to be twice of the picture size. AVS fully utilizes the reference buffer in P picture coding, where P pictures can use two previous adjacent I/P pictures as references. It improves the coding efficiency, and the reference buffer size is the same as MPEG-1 and MPEG-2 have.

3.5. Symmetrical mode for B picture

In the existing coding standards, a macroblock in B pictures can be coded with one of four modes (Direct, Forward, Backward and Interpolation). Both forward and backward motion vectors are coded in the Interpolation mode. AVS adopts a symmetrical mode to replace it. In the mode only the forward motion vectors are coded and the backward motion vectors are derived from the correlation between backward and forward. Therefore, at most one directional motion vector is coded in B macroblocks of AVS.

3.6. Weighted prediction

Weighted prediction can significantly improve coding efficiency especially for scene transition and illumination change. A simple linear model is employed in weighted prediction. The model parameters are coded in the predictive picture header. But, each macroblock is free to decide whether or not use the weighted prediction.

3.7. De-blocking filter

Block-based video coding often produces blocking artifacts especially at low bit rates. AVS defines an adaptive in-loop de-blocking filter to improve the decoded visual quality. The filtering is applied to the boundaries of luma and chroma blocks except for the boundaries of picture or slice. The filtering strength is dependent on macroblock coding type, quantization step, motion vectors and difference among blocks.

3.8. Interlacing coding

For an input interlacing sequence, a picture can be coded as either one frame or two fields (a top field and a bottom field). Only the picture-level adaptation between frame and field is allowed in the current version. When it is coded as two fields, the first field is predicted from the previous decoded fields and the second field is predicted from the first field and the previous decoded fields. These two fields share a picture header. But they should belong to different slices.

3.9. Start code emulation

The entropy coding in AVS can not prevent start code emulation. Therefore, a technique is proposed to solve this problem. When the encoder generates a stream, it will check successive 22 bits before the second least significant bit of a byte. If these bits are all zero, 2 bits “10” are inserted before the second least significant bit. When the decoder reads a byte from the stream, it will check it with two preceding bytes. If they are “0x000003”, the last two bits are discarded.

4. Profile and level

The purpose of defining profiles and levels is to facilitate interoperability among streams from various applications. A profile is a subset of syntax, semantics and algorithms defined by AVS video standard, where a level puts constraints on the parameters of the stream. In AVS video standard, a Jizhun Profile has been defined to target at SD/HD broadcast and storage. The Jizhun Profile contains all techniques defined in AVS video standard except for the Advanced Prediction Mode. There are 4 levels defined in AVS video standard. The maximum picture size varies from 720×576 to 1920×1080 and the maximum bit rate varies from 10 Mbit/s to 30 Mbit/s.

5. Experimental results

This section briefly compares the coding efficiency of H.264 and AVS at 720p and 1080i sequences. H.264 JM 6.1e and AVS RM 5.0 are used in the experiment. The testing conditions are listed in Table 1. Table 2 shows the testing results for both progressive and interlace sequences. For interlace coding, the macroblock adaptive frame/field (MBAFF) coding is used in H.264, while AVS only uses picture adaptive frame/field (PAFF) coding. These experimental results show that AVS achieves performance comparable to that of H.264. Figure 3
to Figure 7 show some PSNR curves for comparisons.

**Table 1. The testing conditions for H.264 and AVS.**

<table>
<thead>
<tr>
<th></th>
<th>JM 6.1e</th>
<th>RM 5.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy coding</td>
<td>CABAC</td>
<td>2D-VLC</td>
</tr>
<tr>
<td>RDO</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>Reference</td>
<td>2 references</td>
<td>2 references</td>
</tr>
<tr>
<td>B frame</td>
<td>2 B frames (IBBP)</td>
<td>2 B frames (IBBP)</td>
</tr>
<tr>
<td>Interface coding</td>
<td>MBAFF</td>
<td>PAFF</td>
</tr>
<tr>
<td>Size of MC</td>
<td>16×16 to 4×4</td>
<td>16×16 to 8×8</td>
</tr>
<tr>
<td>Loop filter</td>
<td>On</td>
<td>On</td>
</tr>
</tbody>
</table>

**Table 2. The testing results on HD sequences.**

<table>
<thead>
<tr>
<th>Progressive</th>
<th>Sequence</th>
<th>Gain over JM6.1e (dB)</th>
<th>Interlace</th>
<th>Sequence</th>
<th>Gain over JM6.1e (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>City</td>
<td>-0.23482</td>
<td></td>
<td>Flamingo</td>
<td>-0.11382</td>
</tr>
<tr>
<td></td>
<td>Night</td>
<td>0.029496</td>
<td></td>
<td>Fireworks</td>
<td>-1.27704</td>
</tr>
<tr>
<td></td>
<td>Harbour</td>
<td>0.283998</td>
<td></td>
<td>Kayaka</td>
<td>-0.37579</td>
</tr>
<tr>
<td></td>
<td>Spincalendar</td>
<td>-0.39819</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Crew</td>
<td>-0.00078</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>-0.06406</td>
<td></td>
<td>Average</td>
<td>-0.58894</td>
</tr>
</tbody>
</table>

**Figure 3. PSNR curve for City.**

**Figure 4. PSNR curve for Harbour.**

**Figure 5. PSNR curve for Spincalendar.**

**Figure 6. PSNR curve for Flamingo.**

**Figure 7. PSNR curve for Kayaka.**

**Acknowledgment**

The authors thank the experts of Audio Video Coding Standard Working Group of China for their contributions.

**References**