Overview of AVS-Video: tools, performance and complexity*

Lu Yu**, Feng Yi, Jie Dong, Cixun Zhang
Institute of Information and Communication Engineering, Zhejiang University, Hangzhou 310027, China

ABSTRACT

Audio Video coding Standard (AVS) is established by the Working Group of China in the same name. AVS-video is an application driven coding standard. AVS Part 2 targets to high-definition digital video broadcasting and high-density storage media and AVS Part 7 targets to low complexity, low picture resolution mobility applications. Integer transform, intra and inter-picture prediction, in-loop deblocking filter and context-based two dimensional variable length coding are the major compression tools in AVS-video, which are well-tuned for target applications. It achieves similar performance to H.264/AVC with lower cost.

Keywords: AVS, video coding, standard

1. INTRODUCTION

Audio Video coding Standard (AVS)[1] is established by the Working Group of China in the same name. Up to now, there are two separate parts in this standard targeting to different video compression applications: AVS Part 2 for high-definition digital video broadcasting and high-density storage media and AVS Part 7 for low complexity, low picture resolution mobility applications. The informal name AVS 1.0 and AVS-M are used to represent for AVS Part 2 and Part 7 respectively.

In this paper, the major AVS video coding tools, their performance and complexity are analyzed. System architecture of AVS Part 2 encoder and the major coding tools are given in section II. In section III, the performance and complexity of AVS Part 2 tools are presented. An overview of AVS Part 7 is given in section IV. Section V is the conclusion of the paper.

2. OVERVIEW OF AVS PART 2

2.1 System architecture

AVS Part 2 is hybrid coding based on spatial and temporal prediction, integer transform and entropy coding. The system architecture is illustrated in Fig. 1.

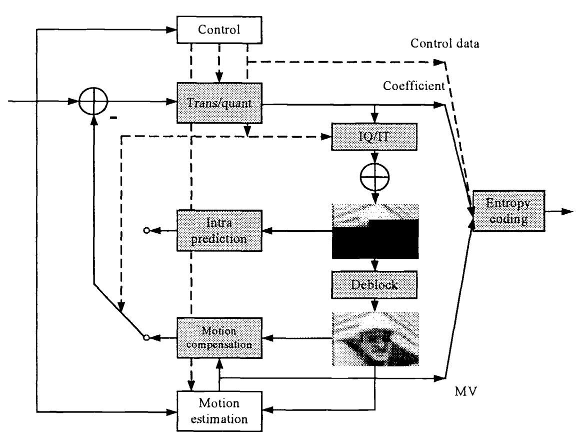

Figure1: AVS Part 2 coding structure.

*This research is supported by NSFC under contract No. 60333020.

** yul@zju.edu.cn phone: 86-571-87953107

2.2 Coding tools

8x8 integer transform, 8x8 spatial prediction, 8x8 motion compensation, in-loop deblocking filter and 2-dimensional variable length coding are the major coding tools of AVS Part 2.

2.2.1 8x8 integer cosine transform (ICT)

AVS Part 2 adopts an 8x8 integer cosine transform with Pre-Scaled Integer Transform (PIT) technique[4].

The forward transform is

$$C_{8\times8} = (ICT_8 \times f_{8\times8} \times ICT_8^T + (1 << 4)) >> 5 \,, \tag{1}$$

where f_{8x8} and C_{8x8} stand for the input pixel matrix and the transformed coefficient matrix respectively, and

$$ICT_8 = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & 8 & 8 \\ 10 & 9 & 6 & 2 & -2 & -6 & -9 & -10 \\ 10 & 4 & -4 & -10 & -10 & -4 & 4 & 10 \\ 9 & -2 & -10 & -6 & 6 & 10 & 2 & -9 \\ 8 & -8 & -8 & 8 & 8 & -8 & -8 & 8 \\ 6 & -10 & 2 & 9 & -9 & -2 & 10 & -6 \\ 4 & -10 & 10 & -4 & -4 & 10 & -10 & 4 \\ 2 & -6 & 9 & -10 & 10 & -9 & 6 & -2 \end{bmatrix}.$$

After forward transform, combined scaling and quantization are performed,

$$CF_{8\times8} = ((C_{8\times8} \otimes CS_{8\times8} + (1 << 18)) >> 19) / Q(QP) \,, \tag{2}$$

where

$$CS_{8\times8} = \begin{bmatrix} 32768 & 37958 & 36158 & 37958 & 32768 & 37958 & 36158 & 37958 \\ 37958 & 43969 & 41884 & 43969 & 37958 & 43969 & 41884 & 43969 \\ 36158 & 41884 & 39898 & 41884 & 36158 & 41884 & 39898 & 41884 \\ 37958 & 43969 & 41884 & 43969 & 37958 & 43969 & 41884 & 43969 \\ 32768 & 37958 & 36158 & 37958 & 32768 & 37958 & 36158 & 37958 \\ 37958 & 43969 & 41884 & 43969 & 37958 & 43969 & 41884 & 43969 \\ 36158 & 41884 & 39898 & 41884 & 36158 & 41884 & 39898 & 41884 \\ 37958 & 43969 & 41884 & 43969 & 37958 & 43969 & 41884 & 43969 \end{bmatrix}.$$

CS_{8x8} is combined scaling matrix, Q(QP) is quantization step size, CF_{8x8} is quantized coefficient matrix.

The operator $\otimes$ in $A_{n\times m} = B_{n\times m} \otimes C_{n\times m}$ means

$$A_{n\times m}(i,j) = B_{n\times m}(i,j) \times C_{n\times m}(i,j)$$
$$0 \le i < n, 0 \le j < m \,. \tag{3}$$

Note that right shifting 19 bits is needed in the combined forward-inverse scaling and quantization process to guarantee 12bit output, so the transform scheme could harmonize with the entropy coding methods to achieve satisfied performance. On some low-end processors, which only support right shifting less than 16 bits in one cycle, alternative solution can be used.

Equation (2) above shows that the quantization and scaling in AVS are separated. The scheme of periodic QP in H.264/AVC is not employed directly in AVS, but rather a quantization/dequantization scheme with QP range 0-63 and a QP period of approximate 8 is used instead.

On decoder side, the quantized coefficient matrix is first dequantized. Then, horizontal and vertical transform are performed one by one.

$$f'_{8\times8} = (ICT_8^T \times (((CF'_{8\times8} \times DQ(QP)) \times ICT_8 + (1 << 2)) >> 3) + (1 << 6)) >> 7 \quad (4)$$

where DQ(QP) is the dequantization step size and f'_{8x8} is the reconstructed pixel matrix.

2.2.2 Intra prediction

Spatial prediction is used in intra coding in AVS Part 2 to exploit spatial correlation of picture. The intra prediction is based on 8x8 block. There are five luminance intra prediction modes, and four chrominance intra prediction modes. The reconstructed pixels of neighboring blocks before deblocking filtered is used as reference pixels for the current block. (Fig. 2)

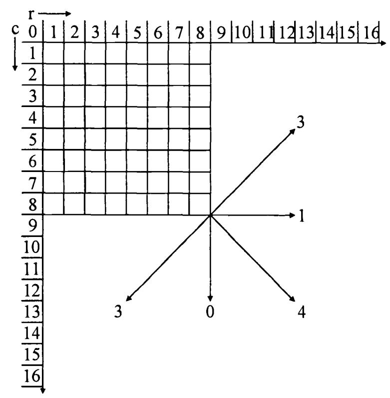

Figure 2: Neighbor Pixels in Luminance Intra Prediction.

Four luminance prediction directions are illustrated in Fig. 2. Five luminance prediction modes are illustrated in Fig. 3.

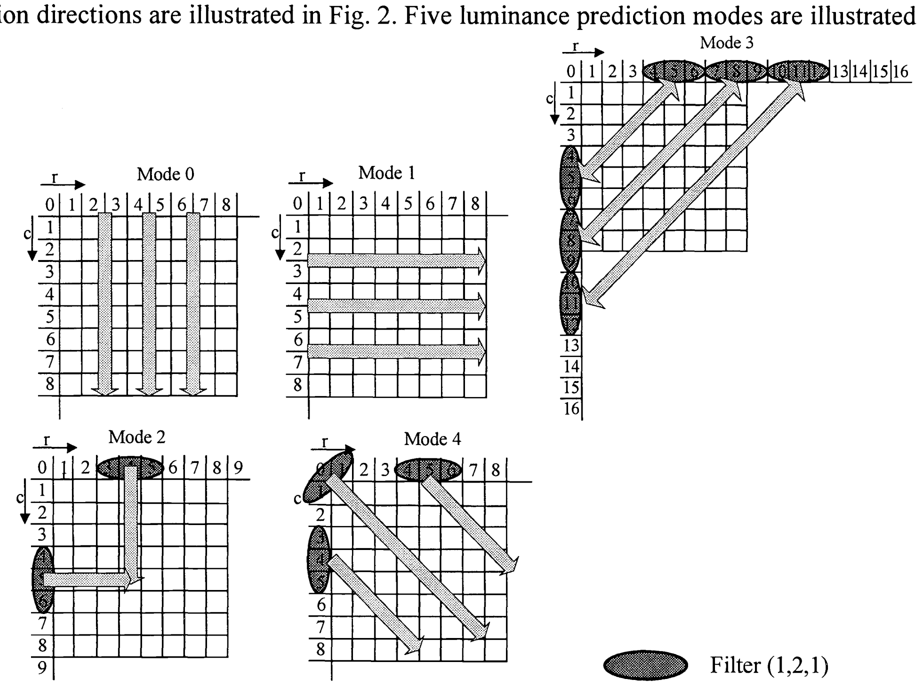

Figure 3: Five luminance intra prediction modes.

2.2.3 Inter prediction

P-picture and B-picture are specified in AVS Part 2. There are four macroblock partition types for inter prediction, 16x16, 16x8, 8x16 and 8x8.

2.2.3.1 P-Prediction

In P-picture, there are 5 inter prediction modes, P_Skip (16x16), P_16x16, P_16x8, P_8x16, and P_8x8. For the latter 4 modes in P-frame, each partition of macroblock is predicted from one of the two candidate reference frames, which are latest decoded I- or P-frame. For the latter 4 modes in P-field, each partition of macroblock is predicted from one of the four latest decoded reference fields.

2.2.3.2 Bi-prediction

There are two kinds of bi-predictions in AVS Part 2, symmetric-prediction and direct-prediction.

In symmetric-prediction, only one forward motion vector is transmitted for each partition. The backward motion vector is conducted from the forward one by a symmetric rule (As shown in Fig. 4) [7, 11]. In direct-prediction, forward and backward motion vectors are all derived from the motion vector of the collocated block in the backward reference picture.

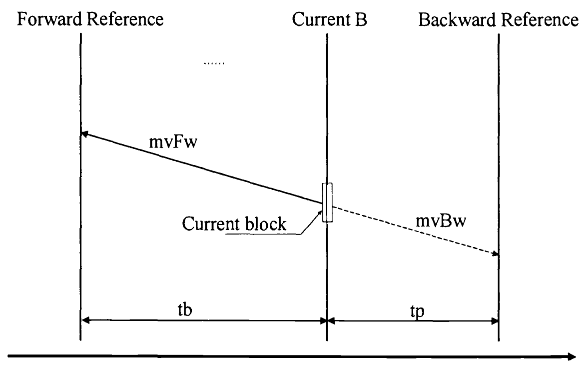

Figure 4: Symmetric mode of AVS Part 2.

2.2.4 Interpolation

A 1/4-pixel interpolation method named as Two Steps Four Taps interpolation (TSFT) is adopted in AVS Part 2. 1/2-pixel samples are interpolated in step 1 and 1/4-pixel samples in step 2. 1/2-pixel interpolation filter is a 4-tap filter H1 (-1/8, 5/8, 5/8, -1/8). For ordinary 1/4-pixel samples, a, c, d, f, I, k, n and q in Fig.6, a 4-tap filter H2 (1/16, 7/16, 7/16, 1/16) is applied, and four special 1/4-pixel samples, e, g, p and r, are filtered by 2-tap bi-linear filter H3(1/2, 1/2) [1, 6, 20, 21]. The positions of the pixels are illustrated in Fig. 5.

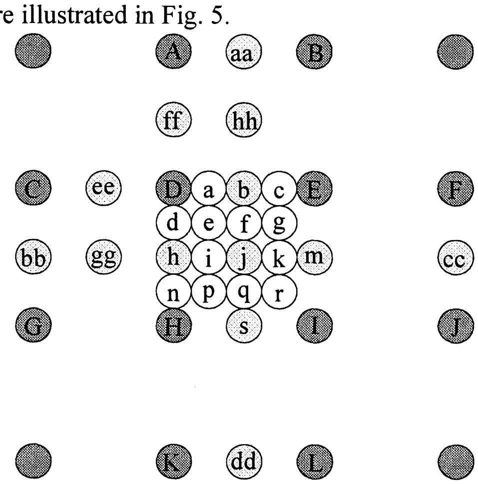

Figure 5: Position of integer pixels, 1/2 pixels and 1/4 pixels.#

The grey pixels are the integer pixel samples, blue ones are 1/2-pixel samples and white ones are 1/4-pixel samples

2.2.5 In-loop deblocking filter

In-loop deblocking filter is applied in AVS Part 2, to reduce the blocking artifacts and enhance both subjective and objective performance. AVS Part 2 deblocking filter first calculates the Boundary-Strength (Bs) of each block boundary, and then applies different filters for different Bs. There are three levels of Bs in AVS Part 2. If one of the two blocks next to the block boundary is intra coding, the Bs of this block boundary is 2. If there reference frames of the two neighboring blocks are the same and the difference of the motion vectors between these two blocks is less than 1 pixel, then the Bs of the block boundary between these two blocks is 0. Otherwise, the Bs is 1.

No filter is applied if Bs=0. When Bs=1 and Bs=2, separate filter process is applied. The details of filters, please refer to [1].

2.2.6 2D_VLC

In AVS Part 2, an efficient context-based 2D-VLC entropy coder is designed for coding 8x8 block-size transform coefficients. 2D-VLC means that a pair of Run-Level is regarded as one event and jointly coded. Context-based is a technique, which uses the coefficient information to switch among different VLC tables. High performance can be achieved with the cost of relatively low complexity.

2.3 Profile and level

Up to now AVS Part 2 has one profile, Jizhun profile. In this profile, there are 4 levels, which are level 4.0 and 4.2 for Standard Definition (SD) video with 4:2:0 and 4:2:2, level 6.0 and 6.2 for HD video with 4:2:0 and 4:2:2, respectively.

3. PERFORMANCE AND COMPLEXITY ANALYSIS OF AVS PART 2

3.1 8x8 block-size

In the block-based video coding, there are transform block-size, inter and intra picture compensation block-size. In MPEG-1, MPEG-2 and MPEG-4 Part 2, the transform block-size is 8x8, and their smallest motion compensation block-size are 16x16, 16x8 and 8x8 respectively. While in H.264/AVC, the smallest transform and intra and inter compensation block-size are 4x4. The smaller the block-size is, the more precise the intra and inter compensation is and the smaller the residuals are. On the other hand, the smaller the block-size is, the more the bits are required for motion vectors and intra prediction modes.

A suitable block-size should be selected for a specific kind of applications. So a series of experiments are designed to compare the coding efficiency with block-size 4x4 and 8x8. The experiments are simulated on JM50c, and the test condition is listed in Table 1. Here the 4x4 transform is the ICT of H.264/AVC and the 8x8 transform is the ICT of AVS Part 2. The results are shown in Table 2. It is obvious that the performance of 8x8 block-size is better than that of 4x4 at high resolution, which is the resolution of the target applications of AVS Part 2.

Table 1: Test condition of 4×4 block-size Vs 8×8 block-size.

Option	4×4 block-size	8×8 block-size
Bitstream structure	IBBPBBP, Only one I frame	IBBPBBP, Only one I frame
Inter block-size	16×16, 16×8, 8×16, 8×8, 8×4, 4×8 & 4×4	16×16, 16×8, 8×16 & 8×8
Intra block-size	16×16 & 4×4	8×8
Entropy coding	CABAC	CABAC
RD optimization	On	On
Reference frame	2 frames	2 frames
Fast ME	On	On
Deblocking filter	4×4 based deblocking filter	8×8 based deblocking filter
QP	24,28,32 & 36	24,28,32 & 36

Table 2: Results of 4×4 block-size Vs 8×8 block-size[##].

Size	QCIF (176×144)					
Sequence	Foreman	Coastguard	Container	Mother&Daughter	News	Average
ΔPSNR (dB)	-0.158	0.022	-0.913	-0.525	-1.026	-0.52
ΔBitrate (%)	3.71	-0.61	20.35	11.81	18.37	10.73

Size	CIF (352×288)					
Sequence	Foreman	Coastguard	Container	Mother&Daughter	News	Average
ΔPSNR (dB)	-0.063	0.386	-0.363	-0.192	-0.528	-0.152
ΔBitrate (%)	1.56	-9.77	9.93	5.05	10.5	3.454
Size	CCIR					
	720×480		720×576			
Sequence	Container	Tempete	Mobile	Flowergarden	Football	Average
ΔPSNR (dB)	-0.388	0.009	-0.442	-0.197	0.147	-0.174
ΔBitrate (%)	12.64	-0.24	14.34	3.82	-2.67	5.578
Size	HD (1280×720)					
Sequence	Crew	Night	City	Spin&Calendar	Harbour	Average
ΔPSNR (dB)	0.27	0.092	0.144	0.046	0.46	0.202
ΔBitrate (%)	-9.12	-2.85	-5	-3.05	-12.8	-6.564
Size	HD (1920×1088)					
Sequence	Flamingo	Fireworks	Kayak			Average
ΔPSNR (dB)	0.286	-0.32	0.519			0.162
ΔBitrate (%)	-5.46	5.19	-8.53			-2.93

##ΔPSNR means at the same Bitrate, PSNR gain of 8×8 block-size verse 4×4 block-size. ΔBitrate means at the same PSNR, Bitrate increase of 8×8 block-size verse 4×4 block-size[9].

The computation complexity of 8x8 ICT is a little bit higher than that of 4x4 ICT, but the complexity of motion compensation, motion estimation and deblocking filter of 8x8 is much lower than that of 4x4. According to the performance and complexity, AVS Part 2 adopts 8×8 as the transform and smallest motion compensation unit.

3.2 Transform and quantization

AVS Part 2 utilizes 8x8 ICT transform, which approximates 8x8 DCT well. It is defined in bit-exact operations, so that inverse-transform mismatch is avoided[3, 4, 10]. This transform can be implemented in 16 bits using adds and shifts. (For 8-point 1-D transform, 32 adds and 24 shifts are needed.)

Simulation on JM9.3 shows that transform of AVS Part 2 (Equation 1) has 0.05dB gain of PSNR over the 8x8 ICT of H.264/AVC High Profile (Table 3).

Table 3: AVS Part 2 ICT Vs H.264/AVC 8x8 ICT

ΔPSNR (dB)	City	Harbour	Night	Raven	Average
IBBP	0.022302	0.067614	0.058222	0.035915	0.046013
IPP	0.032716	0.089913	0.071394	0.041802	0.058957

It's well known that in H.264/AVC, the forward/inverse scaling and the quantization/dequantization are combined in order to reduce the computational complexity. Fig.6 is the block diagram for the conventional ICT coding scheme in H.264/AVC.

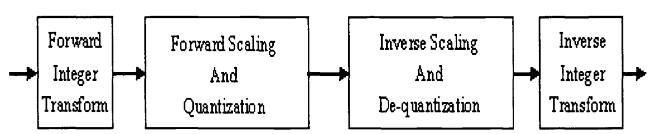

Figure 6: Block diagram of conventional ICT scheme.

The main problem with this conventional ICT scheme is that when 8x8 ICT is used, a memory of 6×8×8=384 bytes (the QP period is 6 in H.264/AVC.) should be allocated to store the dequantization matrix in decoder if we fully expand it to avoid lookup operations in order to facilitate parallel processing and take advantage of the efficient multiply/accumulate architecture of many processors. (Otherwise, every transformed coefficient requires a 3-D lookup operation, which uses QP and the coordinates of the coefficient in the block, to search for the corresponding dequantization coefficient. In this situation, the computational burden will increase a lot.) In fact, this memory size can be reduced without increasing computational complexity if we move the inverse scaling to encoder side and combine it with forward scaling and quantization as one single process. This scheme is named as Pre-Scaled Integer Transform (PIT). The block diagram for the PIT coding scheme is shown below.

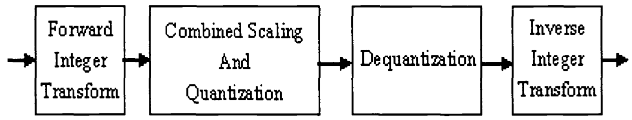

Figure 7: Block diagram of proposed PIT scheme.

Compared with conventional ICT scheme, when PIT is used, implementation complexity can be reduced on decoder side. A memory of only 6 bytes is needed instead of $6\times8\times8=384$ bytes. On encoder side, since forward scaling, inverse scaling and quantization are combined as one single process, both the computational and storage complexity remains unchanged. In AVS Part 2, since the scheme of periodic QP is not used, the saving of memory is significantly larger. Only a memory of 64 bytes is needed instead of a memory of $64\times8\times8=4096$ bytes. (The QP range is 0~63. In order to save memory, the scaling and quantization/dequantization are separated. Even in this case, a memory of $8\times8=64$ bytes can be saved and at the same time the computational complexity is also reduced when PIT is used because no scaling is needed any more on decoder side.) From the analysis above, it is clear that the scheme of PIT is superior to the conventional ICT scheme for its significant complexity reduction, especially for low-end processors while no performance penalty is observed but rather a slightly gain in PSNR can be obtained[4].

3.3 Intra and inter prediction

3.3.1 Intra prediction

A larger intra block-size lengthens the distance between the samples to be predicted and the predicting samples, hence weaken the interrelationship and lower the accuracy of the prediction[5]. So a 3-tap low-pass filter (1, 2, 1) is applied on predicting samples before using DC mode, Diagonal Down Left mode and Diagonal Down Right mode.

Additionally, in the DC mode, each pixel is predicted from the average of the vertically and horizontally corresponding reference pixels, so that the predicted value might be different from pixel to pixel. This makes the DC prediction more accurate than that of H.264/AVC, especially for 8x8 block-size[5].

Table 4 gives an experiment result simulated on RM4.0 (RM is the reference software of AVS Part 2) [5]. Using 5 modes of AVS Part 2 instead of 9 modes of H.264/AVC introduces 0.05 dB lose in PSNR.

Table 4: AVS Part 2 5 modes Vs H.264/AVC 9 modes on 8x8 [5]

	City	Crew	Harbour	Night	Average
ΔPSNR (dB)	-0.09701	-0.06686	0.00399	-0.05718	-0.05427

3.3.2 Bi- Prediction

Symmetric mode is introduced in AVS Part 2 to exploit motion continuity in sequential pictures. On the other hand, this mode can efficiently save bits of coding the second motion vector. For traditional bi-prediction picture, although the coding performance can be improved by joint estimation, independently estimating forward and backward prediction block is practical because joint estimation of forward and backward motion vectors is computationally too expensive to be implemented in reality. However, for symmetric mode, joint estimation algorithm can be easily implemented with the same process as conventional forward or backward independent search solution. Accordingly, symmetric mode can efficiently predict the motion compensation [7,11].

An experiment result to compare the symmetric mode and conventional bi-predictive mode is given in Table 5.

Table 5: Performance comparisons between symmetric mode and conventional bi-predictive mode in HD sequence[7].

	Spincalendar	Crew	Night	Harbour	Average
ΔPSNR (dB)	0.121	0.006	0.044	-0.044	0.03175
ΔBitrate (%)	3.76	0.19	1.2	-1.1	1.0125

3.3.3 Interpolation

Table 6 gives the data fetch bandwidth and computation operations of sub-pixel interpolation of AVS Part 2 and H.264/AVC. Compared with the sub-pixel interpolation of H.264/AVC, the bandwidth of data fetch in AVS Part 2 is reduced by 11 percent while the computation complexity remains similar.

Additionally, AVS Part 2 interpolation method achieves an average 0.0375dB PSNR gain in HD sequences [6,12,13].

3.5 In -loop deblocking filter

One particular characteristic of block-based coding is the production of visible block structures. Block boundaries are typically reconstructed with less accuracy than interior pixels and "blocking" is generally considered to be one of the most visible artifacts[10]. For this reason, deblocking filter is used in AVS Part 2. It is an adaptive deblocking filter. The filter is classified into 3 classes with filter strength Bs. Strongest filter is used for Intra block boundary, and weaker filter for Inter block without continuous motion compensation block.

The block boundary and pixels involved in deblocking filter is illustrated in Fig. 8. For each boundary, at most 6 pixels are involved and no more than 4 pixels are changed in deblocking filter.

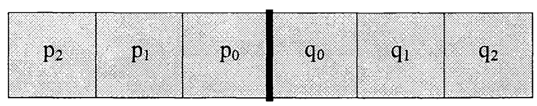

Figure 8: Block boundary and pixels involved in deblocking filter.

Since 8x8 is the smallest prediction and transform block-size, the block boundaries of deblocking filter are 8x8 aligned. The number of block boundaries, the number of Bs levels and the number of pixels changed in the deblocking filter are all much fewer than that of H.264/AVC.

3.4 Entropy coding

In AVS Part 2, the codewords for all syntax elements are constructed based on Exponential-Colomb codes or fixed-length codes. The fixed-length codes are used to code the uniform distributed syntax elements while the Exp-Golomb codes can approximate variable distributions with proper order[19]. In AVS, the value of syntax element is firstly mapped to a non-negative integer, defined as CodeNum. The Exp-Golomb codes have regular structures, which means that any non-negative CodeNum can be mapped to a unique binary codeword using the regular code-constructing rule. Due to the regular codeword structure, for a given CodeNum, its binary code can be constructed in coding process without involving high computational complexity. So CodeNum can be stored in VLC tables instead of real codewords like Huffman codes in MPEG-2/4. It is a valuable feature that resolves the problem of high memory requirement for multiple VLC tables. The memory requirement is kept relatively low while 21 VLC tables are used in AVS Part 2.

In this standard, an efficient context-based 2D-VLC entropy coder is designed for coding 8x8 block-size transform coefficients, where 2D-VLC means that a pair of Run-Level is regarded as one event and jointly coded. In former video coding standards, such as MPEG-2/4, 2D-VLC is already used. They use one single VLC table to code a certain type of transform blocks, e.g. one table for Intra blocks, one table for Inter blocks, etc. However, when transform coefficients in one block are mapped into one-dimensional (Level, Run) sequence by the zigzag scan, the sequence exhibits decreasing tendency for Level's magnitude and increasing tendency for Run. This tendency is the useful context information. To fully exploit the context information and obtain further compression, multiple 2D-VLC tables may be used while coding one transform block. The table for coding the current (Level, Run) is selected depending on the last coded Level's magnitude. Since reverse zigzag scan order is used here, the magnitude of the first level to be scanned is generally equal to 1. So the first (Level, Run) to be coded in a transform block use a fixed code table.

This coding method introduces 19 2D-VLC tables for coding of residual coefficients and the memory requirement is about 1k bytes. This method gain up to 0.23 dB compared to one-table-for-one-type-of-block coding method[8].

3.6 Performance

Experiments results (Fig. 9) show that AVS Part 2 has similar performance with H.264/AVC Main profile and is much better than MPEG-2, which is widely used in SD and HD video compression up to now. The test conditions are listed in Table 6.

Table 6: Test condition of comparison of AVS Part 2 H.264/AVC and MPEG2.

Option	Flamingo (1920 x 1088 interlace)			Harbour (1280 x 720 progressive)		
	AVS Part 2	H.264/AVC	MPEG2	AVS Part 2	H.264/AVC	MPEG2
Bit stream structure	IBBPBBP…	IBBPBBP…	IBBPBBP…	IBBPBBP…	IBBPBBP…	IBBPBBP…
Field/Frame type	PAFF	MBAFF	Field coding	Frame coding	Frame coding	Frame coding
Entropy coding	2D-VLC	CABAC	VLC	2D-VLC	CABAC	VLC
RD optimization	On	On	N/A	On	On	N/A
Reference frame	2 frames	2 frames	1 frame	2 frames	2 frames	1 frame
Fast ME	On	On	On	On	On	On
Deblocking filter	On	On	N/A	On	On	N/A
QP	27,29,31 33,36,39	22,24,26 28,30,32 34	8,10,11,12 14,16,20 24,28,31	27,29,31 33,36,39	22,24,26 28,30,32 34	8,9,10 11,12,16 20,22

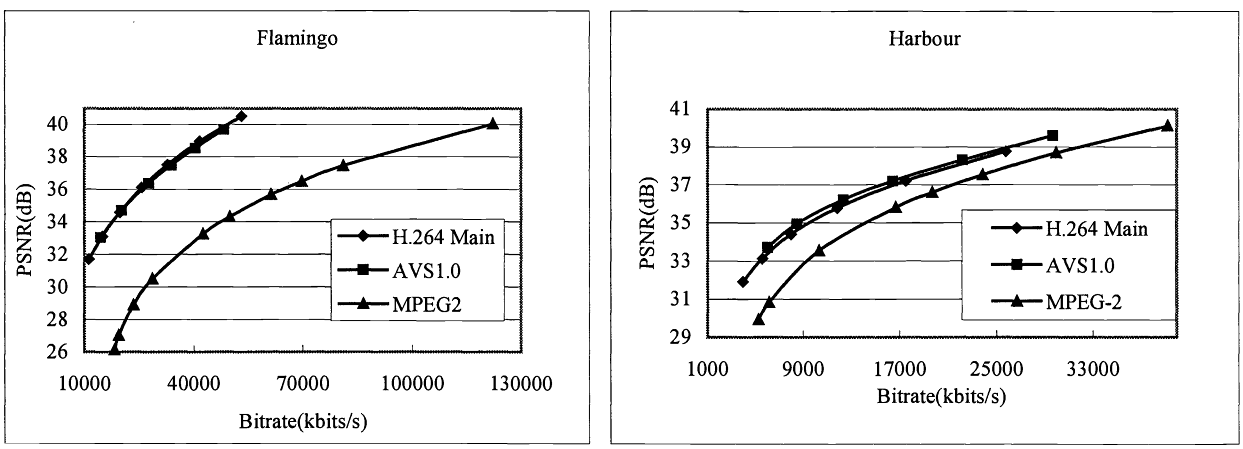

Figure 9: Comparison of AVS Part 2, H.264/AVC and MPEG2 on HD sequence.

4. OVERVIEW OF AVS PART 7

4.1 System Structure

AVS Part 7 is also a prediction, transform and entropy coding based hybrid coder, whose framework is the same as AVS Part 2 (Fig. 1). AVS Part 7 aims at the applications of video communications on mobile devices with limited processing and memory resources.

4.2 Coding tools

In AVS Part 7, only I- and P-picture are applied. The major tools is similar to AVS Part 2: transform, quantization, Inter and Intra prediction, in-loop deblocking filter, 2D-VLC entropy coding etc.

4.2.1 4x4 block-size and 4x4 transform

The experiment in section 3.1 shows that small block-size performs better than big one for lower image resolution. 4x4 is the unit of transform, intra prediction and smallest motion compensation in AVS Part 7.
The following 4x4 transform matrix is applied,

$$T_4 = \begin{bmatrix} 2 & 3 & 2 & 1 \\ 2 & 1 & -2 & -3 \\ 2 & -1 & -2 & 3 \\ 2 & -3 & 2 & -1 \end{bmatrix}.$$

PIT is also used in AVS Part 7 to reduce the complexity[2, 22].

4.2.2 Simplified intra prediction

There are three major low complexity intra prediction tools: Direct Intra Prediction (DIP), padding before prediction and simplified chrominance intra prediction.

In I slice, each macroblock has a bit to indicate the current macroblock coded with DIP-Mode or usual mode. If it is DIP-Mode, the modes of all 16 4x4 blocks in this macroblock are Most_Probably_Mode. If it is usual mode, mode information is coded in bitstream to indicate which intra_4x4 mode is used for each 4x4 block[2, 14]. Only 0.02dB average PSNR lose is introduced by replacing Intra_16x16 with DIP. The advantage of DIP replacing Intra_16x16 is that no DC Hardmard transform, coefficients reordering and relative entropy coding is required.

Padding before prediction is a technique to generate some of reference pixels. r_5, r_6, r_7, r_8 is always padding from r_4, and c_5, c_6, c_7, c_8 is always padding from c_4. Conditional test on available of up-right and down-left reference pixels are no longer required[15].

Only three modes are used in chrominance intra prediction, DC mode, Vertical mode and Horizontal mode. (Fig. 10) All 8 4x4 blocks of U and V components use the same mode[2, 16].

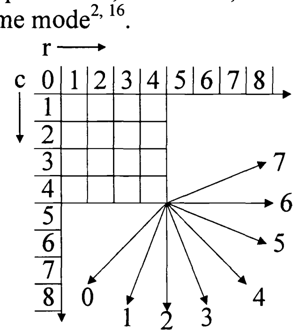

Figure 10: 4x4 intra prediction.

4.2.3 Simplified deblocking filter [2,17,18]

AVS Part 7 adopts a simplified deblocking filter, in which boundary strength is decided at macroblock level instead of block level. Additionally, a simpler filter process, which is suitable for parallel processing, is applied.

Macroblock type (Intra, Skip or non-skip Inter) and QP of current macroblock is used to decide the boundary strength of this macroblock. There are three different levels of boundary strength. Different filtering processes are applied for different boundary strengths. Two pixels on each side of the boundary are used to modify at most two pixels each side in the filtering process. So filtering of all vertical (horizontal) boundaries is independent to each other, which facilitate parallel processing.

4.2.3 2D_VLC

In AVS Part 7, context-based 2D-VLC is used to code transform coefficients, which is similar to that of AVS Part 2. However, the 2D-VLC tables and the way of table switch are well designed to adapt to the distribution of (Level, Run) of 4x4 transform blocks. Last coded Level is used to switch among different (Level, Run) 2D-VLC tables. And last coded Run is used to switch among tables when the magnitude of last coded Level equals to 1.

5. CONCLUSION

AVS-video is an application driven coding standard with well-optimized techniques. It achieves similar performance to H.264/AVC with lower cost. AVS Part 2 targets to high-definition digital video broadcasting and high-density storage media and AVS Part 7 targets to low complexity, low picture resolution mobility applications.

REFERENCES

1. AVS Video Expert Group, *Information Technology – Advanced Audio Video Coding Standard Part 2: Video, in Audio Video Coding Standard Group of China (AVS)*, Doc. AVS-N1063, Dec.2003.
2. AVS Video Expert Group, *Information Technology – Advanced Audio Video Coding Standard Part 7: Mobility Video, in Audio Video Coding Standard Group of China (AVS),* Doc. AVS-N1151, Dec.2004.
3. H. Malvar, A. Hallapuro, M. Karczwicz, and L. Kerofsky, *Low-complexity transform and quantization in H.264/AVC*, IEEE Trans. circuits Syst. Video Technol., vol. 13, pp. 598-603, July 2003.
4. Ci-Xun Zhang, Jian Lou, Lu Yu, Jie Dong, and Wai-Kuen Cham, *The Technique of Pre-Scaled Integer Transform*, IEEE International Symposium on Circuits and Systems, May 2005.
5. Zhang Nan, Yin Baocai, Kong Dehui, and Yue Wenying, *Spatial Prediction Based Intra-Coding,* IEEE International Conference on Mutimedia and Expro, May 2004.
6. Ronggang Wang, Chao Huang, Jintao Li, Yanfei Shen, *Sub-piexl Motion Compensation Interpolation Filter in AVS*, IEEE International Conference on Mutimedia and Expro, May 2004.
7. Xiangyang Ji, Debin Zhao, Wen Gao, Qingmin Huang, Siwei Ma, Yan Lu, *New Bi-Prediction techniques for B pictures coding*, IEEE International Conference on Mutimedia and Expro, May 2004.
8. Qiang Wang, Debin Zhao, Siwei Ma, Yan Lu, Qingming Huang, Wen Gao, *Context-Based 2D-VLC for Video Coding*, IEEE International Conference on Mutimedia and Expro, May 2004.
9. G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves,* ITU-T SG16 Doc. VCEG-M33, 2001.
10. ThomasWiegand, Gary J. Sullivan, Gisle Bjøntegaard & Ajay Luthra, *Overview of the H.264/AVC Video Coding Standard,* IEEE Trans. circuits Syst. Video Technol., vol. 13, pp. 560-576, July 2003.
11. Xiangyang Ji, Wen Gao & Feng Wu, *A new P frame prediction method*, Doc. AVS M1074 2003.
12. Chao Hang, Ronggang Wang, Jintang Li and Yanfei Shen, *A method of sub pixel interpolation*, AVS Doc. AVS_M1119, 2003.
13. Chao Hang, Ronggang Wang, Jintang Li and Yanfei Shen, *Improvement of sub pixel interpolation*, AVS Doc. AVS_M1162, 2003
14. Feng Yi, Jianpeng Wang, Jie Dong, Lu Yu, *An improvement of intra prediction mode coding,* Doc. AVS_M1456 , 2004.
15. Yanfei Shen, Dongmin Zhang, Lejun Yu, Chao Huang and Shouxun Lin, *A simplified intra prediction method,* Doc. AVS-M1419, 2004.
16. Lianhuan Xiong, *Improvement of chroma intra prediction,* Doc. AVS-M1379, 2004.

17. Feng Yi, Jie Dong, Lu Yu, *An improvement of intra frame loop filter,* Doc. AVS-M1517 2004.
18. Feng Yi, Jie Dong, Lu Yu, *An improvement of inter frame loop filter,* Doc. AVS-M1518 2004.
19. Teuhola, J., *A compression method for clustered bit-vectors*, Information Processing Letters, 1978, 10, (7), pp. 308–311.
20. Yu Zhen, Wen Gao, Debin Zhao, Siwei Ma, Hanqing Lu, *A low-complexity 16-bis 4x4 ICT and quantization*, Doc. AVS_M1331, 2004.