Signal Processing: *Image Communication* 25 (2010) 633–647

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Algorithm analysis and architecture design for rate distortion optimized mode decision in high definition AVS video encoder

Hai bing Yin [a,b,*], Honggang Qi [c], Hui Zhu Jia [b], Chuang Zhu [b], Xiao Dong Xie [b]

[a] *China Jiliang University, Department of Information Engineering, No. 258, Xue yuan street, Xiasha, Hangzhou, Zhejiang 310018, China*
[b] *National Engineering Laboratory for Video Technology, Peking University, Beijing 100871, China*
[c] *Graduate University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 11 December 2009
Accepted 11 August 2010

Keywords:
AVS
Mode decision
Rate distortion optimization
VLSI architecture

ABSTRACT

There are abundant intra and inter prediction modes in the AVS video coding standard. Rate distortion optimized mode decision can fully utilize this flexibility to improve the spatio-temporal prediction efficiency and maximize the coding efficiency. However, the implementation complexity is dramatically high due to huge throughput burden. Hardware oriented mode decision algorithm is tailored for VLSI implementation in this work for high definition video coding. Mode preselection is employed to alleviate the dramatic throughout burden. Also, intelligent pipeline scheduling mechanism is proposed to break the intrinsic data dependency in intra prediction, which is directly related with mode decision. The proposed simplified algorithm is well-suited for hardware implementation with small performance penalty. Finally, the VLSI architecture is proposed with good trade off between circuit consumption and rate distortion performance.

1. Introduction

Chinese audio and video coding standard (AVS) is a new national standard built for digital media applications [1]. Its video part (AVS-P2) had been formally accepted as the national standard of China in 2006. AVS video standard achieves equivalent coding performance compared with H.264/AVC in high definition (HD) cases with lower complexity. The industrialization for the AVS standard is being on and leaded by the AVS industry alliance.

Although the implementation complexity of AVS is relatively lower than that of H.264, real-time HD AVS video encoder implementation is still a huge challenge. Being different from H.264 baseline profile, bidirectional B frame prediction is supported in AVS Jizhun profile, which is equivalent to H.264 main profile to some extent. The powerful multicore processors available also cannot satisfy this high throughput desired in HD AVS video coding with B frame supporting. Currently dedicated HD AVS video encoder chip is still vacant, and hardware implementation for HD AVS video encoder is highly desired.

AVS achieves superior performance than MPEG-2 partly due to adopting abundant intra and inter coding modes selected by the rate distortion optimization (RDO) technique [2]. Although RDO based mode decision in AVS improves the performance greatly, the cost function (RDcost) calculations for all modes are computationally intensive. It is challenging to implement genuine RDO based mode decision with moderate circuit resource and system power consumption. Feature and rate distortion models based quick mode decision algorithms for H.264 had attracted intensive research recently [2,13–21], and these algorithms can also be used as reference for AVS

[*] Corresponding author at: China Jiliang University, Department of Information Engineering, No. 258, Xue yuan street, Xiasha, Hangzhou, Zhejiang 310018, China. Tel.: +86 0571 81320806;
fax: +86 0571 86914573.
E-mail addresses: haibingyin@163.com (Hai bing Yin), hgqi@jdl.ac.cn (H. Qi), hzjia@pku.edu.cn, hzjia@polito.it (H.Z. Jia), czhu@jdl.ac.cn (C. Zhu), xdxie@pku.edu.cn (X. Dong Xie).

0923-5965/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2010.08.001

video coding. However, they generally suffer from obvious performance degradation, or illsuited for hardware implementation due to algorithm irregularity. Also, RDO based mode decision was turned off in typical H.264 video encoder VLSI architectures [3–5] and prevailing intelligent property cores such as CAS [6], Imagination [7], 4i2i [8], Faraday [9], Global Unichip [10], etc. In these architectures, simplified mode decision criterion is used instead of RDcost. The literature on mode decision architecture is unavailable due to technology secrecy perhaps in H.264 codec chip, such as MB86H50 of Fujitu [11], MG1264 of Mobilygen [12], etc.

Dramatic data processing throughput in genuine RDO based mode decision is the largest challenge. Also, the block level intra prediction in AVS standard results in data dependency, which is very harmful for regular hardware pipeline rhythm. Thus, joint design and optimization between hardware oriented algorithm and VLSI architecture is very important.

In this paper, we analyze the challenges of pipeline rhythm and throughput in mode decision in Section 2. Mode preselection and data dependency immune pipeline scheduling mechanism are proposed in Section 3. The proposed VLSI architecture for RDO based mode decision is given in Section 4. Finally, simulations and conclusion are given in Section 5.

2. AVS video coding algorithm

2.1. Algorithm framework review

AVS-P2 is a MPEG-like video coding standard similar with H.264. The block diagram of AVS video encoder is shown in Fig. 1. The basic coding unit is also macroblock (MB). Motion estimation and compensation give a prediction version of the current MB using video temporal correlation. Variable block size motion estimation, fractional pixel motion refinement, and multiple reference frame techniques are used in AVS motion estimation. Intra-frame prediction gives another prediction version of the current MB using video spatial correlation. As a result, multiple intra and inter prediction modes are supported in AVS. Mode decision algorithm selects an optimal mode in the terms of rate distortion optimization for genuine coding. At the same time, the predicted MB is inputted into the residue coding loop including residue generation, transform, quantization, inverse transform, and inverse quantization. The distorted image is reconstructed with in-loop 6-tap deblocking filter with four boundaries of 8 × 8 blocks and stored as the new reference frame for the following frame coding. Entropy coding adopts run length coding and Exp-Golomb variable length coding (VLC) to exploit the symbol statistical correlation.

There are also some differences between AVS and H.264. There are five luminance and four chrominance intra prediction modes on the basis of 8 × 8 blocks in AVS. Also, owing to targeting for high definition applications mainly, only 16 × 16, 16 × 8, 8 × 16, and 8 × 8 MB inter partition modes are used in AVS, in which 1/4 pixel motion compensation with 4-tap fractional pixel interpolation is adopted. Being different from H.264 baseline profile, the Jizhun profile in AVS supports bidirectional prediction using a novel "symmetric" mode [1]. Combined with forward, backward, symmetric, and direct temporal prediction modes, there are totally 50 MB inter prediction modes.

2.2. Mode decision and performance analysis

Similar with H.264, AVS does not stipulate a specific mode decision implementation algorithm. RDO based mode decision is the most direct solution with the optimal coding efficiency and the highest complexity. There are some fast mode decision algorithms based on simplified rate distortion models or local image features in the literature [13–21]. Some of them are illsuited for VLSI implementation due to algorithm irregularity. Others

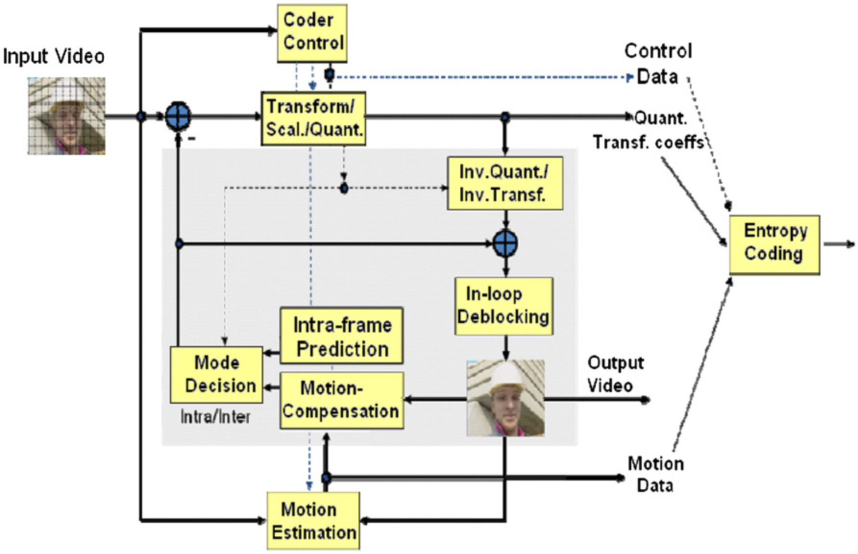


Fig. 1. Block diagram of AVS video encoder.

may also be well-suited for VLSI implementation using simplified decision criterion, however they suffer from obvious performance degradation. Three typical simplified criterions are sum of absolute difference (SAD), sum of absolute transformed difference (SATD), and weighted SAD (WSAD) [3,4]. By employing the Lagrangian optimization technique, the WSAD criterion achieves superior performance than SAD or SATD. Nevertheless, its coding performance degradation compared with genuine RDO based method is still obvious with unnegligible image quality degradation.

The performance degradation is mainly derived from measure simplification of rate and distortion. Suppose S and S' are the original MB and the reconstructed one, and P is the predicted version of the current MB of a certain mode. Q_p and λ are the MB quantization step and the Lagrange multiplier for mode decision. Two mode decision criterions RDcost and WSAD are described in the following equations:

$$\mathrm{RDcost}(S,S',\mathrm{mode}|Q_p,\lambda) = \mathrm{SSD}(S,S',\mathrm{mode},Q_p)+\lambda \times R_{MB}(S,S',\mathrm{mode},Q_p) \quad (1)$$

$$\mathrm{WSAD}(S,P,\mathrm{mod}|Q_p,\lambda) = SAD(S,P,\mathrm{mod},Q_p)+\lambda' \times R_{MBheader}(S,P,\mathrm{mod},Q_p) \quad (2)$$

Here, SSD (S,S',mode,Q_p) is the sum of the squared difference between S and S' in the case of Q_p and λ, while SAD (S,P,mode,Q_p) is the sum of the absolute difference between S and P. $\mathrm{R_{MB}}$ (S,S',mode,Q_p) is the coding bit of all syntax elements in the MB in the case of Q_p and λ'. $\mathrm{R_{MBheader}}(S,P,\mathrm{mode},Q_p)$ is the coding bit of the syntax elements in the MB header.

RDO based mode decision achieves superior coding performance contributed by Lagrangian optimization. Genuine distortion is measured with SSD (S,S',mode,Q_p) in the case of RDcost criterion, genuine rate is also used in the case of RDcost measured with $\mathrm{R_{MB}}$ (S,S',mode,Q_p) with all syntax elements considered. Comparatively, only rate factor is considered for mode decision in the case of WSAD criterion, in which rate is estimated with SAD (S,P,mode,Q_p) and $R_{MBheader}$ (S,P,mode,Q_p). The prediction residue SAD (S,P,mode,Q_p) is approximately used as the rate measure for quantized DCT coefficients.

It is the measure simplifications of rate and distortion in WSAD that result in the obvious performance degradation compared with RDcost. In order to sustain the superiority of AVS, we will focus on RDO based mode decision for hardware implementation in this work.

It is very computationally intensive due to the abundant modes adopted in H.264. However, almost all H.264 video encoder architectures adopt simplified mode decision, and WSAD, SATD, or SAD criterion was used instead. Relatively, challenges of RDO based mode decision in AVS video encoder is relatively lower than H.264. On the one hand, the numbers of inter and intra modes in AVS are smaller than that of H.264, and the processing throughput burden is also lower than that of H.264. On the other hand, the processing unit granularity in AVS such as DCT, quantization, inverse DCT, inverse quantization is 8 × 8 block in the rate and distortion calculation loop, while that is 4 × 4 block in H.264 smaller than AVS. This means that the circuit consumption for the basic processing unit of AVS is higher than that of H.264. Thus, it is possible to implement RDO based mode decision with reasonable mode preselection to alleviate the throughput burden without too much additional circuit consumption.

2.3. Computation analysis for RDcost estimation

Fig. 2 shows the framework of RDcost calculation for one 8 × 8 block of a certain MB coding mode. First, the difference between the input block S and its prediction P is calculated by residue generation. Then, the residue r is transformed by DCT and followed by quantization (Quant, Q), and then the coding bit rate R is computed by entropy coding (EC). The quantized coefficients are also fed into inverse quantization (Inver Quant, IQ), inverse transform (IDCT), and compensation to reconstruct the block S'. SSD between the original residue (r) and the reconstructed residue (r') is computed for distortion measure. In the end, RDcost is obtained according to R and SSD. RDcost is used to evaluate the RD performance of all candidate modes, and the mode with the smallest RDcost is selected for bitstream generation.

RDO based mode decision for hardware implementation is challenged by the following two factors. On the one hand, intrinsic data dependencies exist in video coding algorithms. At the MB level, integer pixel motion estimation (IME), fractional motion estimation (FME), mode decision (MD), and intra prediction (IP), deblocking filter

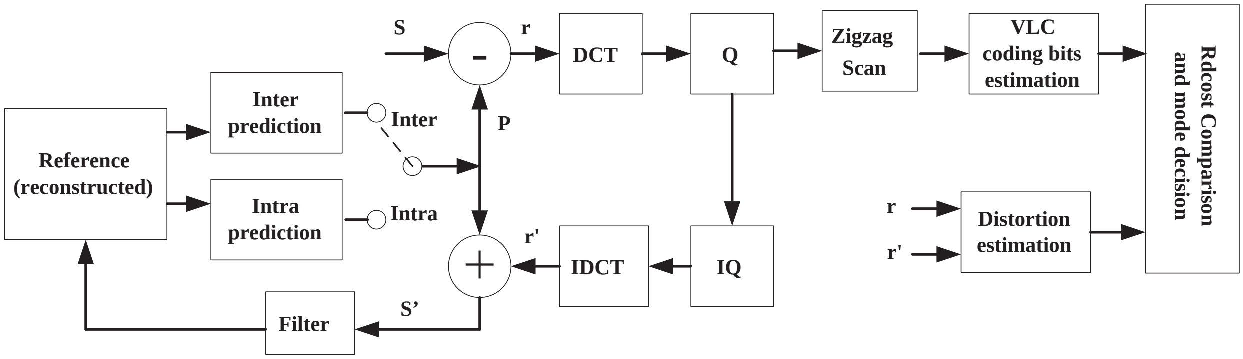


Fig. 2. Framework of cost function calculation and mode decision.

and entropy coding must be processed in turn. At the block level, one block intra prediction cannot initiate until its left, up blocks have finished mode decision and reconstruction; in RDO based mode decision algorithm, DCT, Quant, Inver Quant, and IDCT should be processed in turn. All these dependencies result in harmful hardware latency and inefficient pipelining. On the other hand, throughput is still high although the coding modes in AVS are relatively smaller than that in H.264. In RDO based mode decision for AVS video coding, mode in equation (1) is chosen from the set of all candidate MB partition modes as follows:

For the MBs in P frames:

$$mode \in \{\mathrm{P\_skip,p\_16} \times 16,\mathrm{p\_16} \times 8,\mathrm{p\_8} \times 16,\mathrm{p\_8} \times 8,\mathrm{I\_8} \times 8\} \quad (3)$$

For the MBs in *B* frames:

$$mode \in \{\mathrm{B\_skip,B\_direct,B\_16} \times 16,\mathrm{B\_16} \times 8,\mathrm{B\_8} \times 16,\mathrm{B\_8} \times 8,\mathrm{I\_8} \times 8\} \quad (4)$$

Here, the temporal prediction directions of the blocks in each MB partition mode, forward, backward, or bidirectional/skip, are not considered. Due to the MB partition mode and temporal prediction direction combination, the candidate inter coding mode number is large. In the mode of B_8 × 8 in B frames, each block may be B_direct_8 × 8 or normal B_8 × 8 mode. The motion vectors and reference frame in the direct mode and the skip mode are identical, and no motion information is coded. The difference is that residual information will be encoded in the direct mode, while no residue is encoded in the skip mode.

For the luminance block I_8 × 8 mode:

$$\mathrm{mode}_{I\_8\times 8} \in \{DC,\mathrm{Vertical,Horizontal,Down\_Lef}t,\mathrm{Down\_Right}\} \quad (5)$$

For the chrominance block *I*_8 × 8 mode:

$$\mathrm{mode}_{\mathrm{I\_8\times 8}} \in \{DC,\mathrm{Vertical,Horizontal,Plane}\} \quad (6)$$

In real-time HD AVS video coding, there are so many MBs are to be processed in each picture. Also, there are approximately fifty inter modes and seventy intra/inter modes at the worst case in B frames. Although there are less coding modes in AVS compared with H.264, RDO based mode decision is still highly challenged due to this high throughput.

3. RDO based mode decision algorithm modification

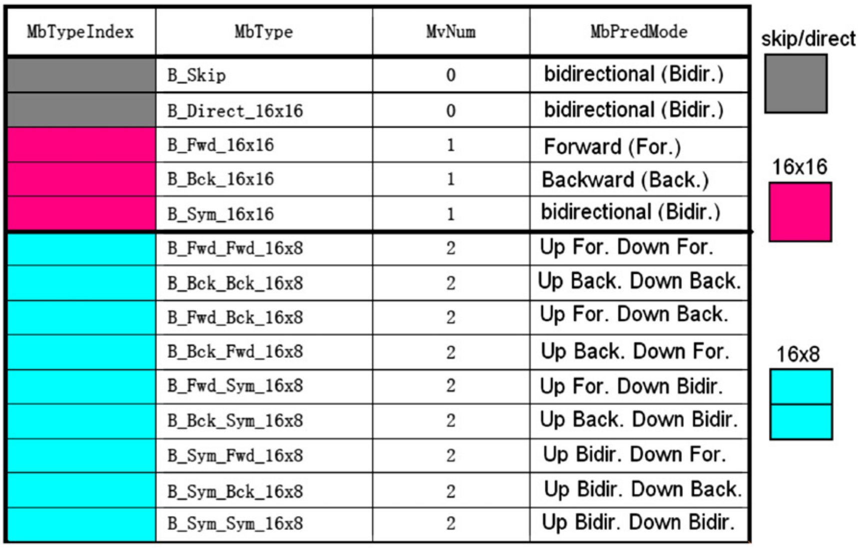
3.1. Data processing throughput analysis

In AVS Jizhun profile, five luminance modes and four chrominance modes are adopted for 8 × 8 block intra prediction. There are four luminance and two chrominance blocks in each MB in the case of 4:2:0 format video. Thus, there are totally 5 × 4+4 × 2=28 blocks to be processed and checked for RDO based intra mode decision.

There are five inter modes including P_skip, 16 × 16, 16 × 8, 8 × 16, and 8 × 8 in P frames. Comparatively, the inter prediction modes of B frames are more complex. An inter prediction mode of B frame is the combination result of two factors. One is the temporal prediction direction such as forward, backward, and bidirectional (symmetric or skip/direct). The other factor is the MB partition mode such as 16 × 16, 16 × 8, 8 × 16, and 8 × 8. The temporal prediction direction and MB partition mode combination results in abundant inter modes in B frames.

Fig. 3 gives an example for inter mode combination between temporal prediction direction and MB partition in B frame. For example, in the 16 × 8 MB partition mode, there are nine possible temporal prediction direction combinations, such as up block forward (For.) and down block forward (For.), up block backward (Back.) down block backward (Back.), etc.

If both the temporal prediction direction and the MB partition of all inter modes are selected by RDO based mode decision, there may be nearly fifty inter prediction



MbTypeIndex	MbType	MvNum	MbPredMode
■	B_Skip	0	bidirectional (Bidir.)
■	B_Direct_16x16	0	bidirectional (Bidir.)
🟪	B_Fwd_16x16	1	Forward (For.)
🟪	B_Bck_16x16	1	Backward (Back.)
🟪	B_Sym_16x16	1	bidirectional (Bidir.)
🟦	B_Fwd_Fwd_16x8	2	Up For. Down For.
🟦	B_Bck_Bck_16x8	2	Up Back. Down Back.
🟦	B_Fwd_Bck_16x8	2	Up For. Down Back.
🟦	B_Bck_Fwd_16x8	2	Up Back. Down For.
🟦	B_Fwd_Sym_16x8	2	Up For. Down Bidir.
🟦	B_Bck_Sym_16x8	2	Up Back. Down Bidir.
🟦	B_Sym_Fwd_16x8	2	Up Bidir. Down For.
🟦	B_Sym_Bck_16x8	2	Up Bidir. Down Back.
🟦	B_Sym_Sym_16x8	2	Up Bidir. Down Bidir.

skip/direct ■

16x16 🟪

16x8 🟦

Fig. 3. Inter mode combination example in B frame.

modes in B frames. If intra modes are also selected by RDO based mode decision, there are almost seventy candidate modes. As a result, the throughput burden is too high and unacceptable for genuine RDO based mode decision. Necessary algorithm simplification is highly desired.

We had made simple investigation and research on RDO based mode decision architecture in [2]. Simplified algorithm for SSD estimation was used, and iterative table lookup based quantization and inverse quantization was used for bypass division calculation. Inverse DCT was not needed for SSD estimation in the mode decision (MD) pipeline. But additional multiplication was necessary. More important, the SSD and bit rate estimation was not accurate enough with error drift. Thus, the whole coding modules such as quantization, inverse quantization, inverse DCT, and entropy were necessary for the final bitstream coding for the selected mode. Otherwise, the error drift between encoder and decoder would be very high resulting in obvious performance degradation. Also, the throughout burden of B frames was not considered.

As far as RDO based mode decision is considered in this work, mode preselection mechanism will be used for throughout burden alleviation on the one hand. On the other hand, hardware implementation of the inverse quantization, inverse DCT, and entropy coding for RDcost estimation and final coding will be jointly considered for hardware efficient reuse.

3.2. Mode preselection and algorithm simplification

We take two approaches to alleviate the serious throughput burden. On the one hand, genuine RDO based mode decision is adopted for intra mode selection in I frames to sustain the fidelity of anchor frame of the whole GOP, while WSAD based mode decision is used for intra mode selection in P, B frames based on two considerations. One is that there are too many candidate modes to be checked, so candidate mode elimination is highly expected. Another is that the percentage of intra modes is largely lower than that of inter modes in P and B frames, hence simplified WSAD based intra mode decision in P and B frames results in negligible performance degradation.

On the other hand, two factors in MB inter prediction modes are separately selected in mode decision algorithm. Temporal prediction direction measures the temporal correlation between the current block and the displaced blocks in the forward or backward reference frames. Motion estimation searches the motion vector just based on this measure. Hence temporal prediction direction is preselected at the fractional pixel motion estimation stage using the WSAD criterion. The selected temporal prediction mode may be forward, backward or symmetric, and we label it as f/b/s; while the MB partition mode is used to describe the motion consistency of 1 MB. If four blocks in a MB have consistent motion, the optimal MB partition mode will be 16×16. If four blocks in a MB have highly irregular motion, the optimal MB partition mode will be 8×8. Certainly, the quantized DCT coefficients should also be considered. Thus, the MB partition mode selection is chosen by the RDO based mode decision algorithm.

With the above two simplified measures, candidate modes of *P* and *B* frames are largely reduced. The worst case occurs in the case of B frame with the most inter candidate modes. The temporal prediction of each 8×8 mode (B_8×8) in B frames may be forward, backward, bidirectional (symmetric), or skip/direct. Similarly, the temporal prediction forward, backward, and symmetric (f/b/s) of B_8×8 mode are selected using the WSAD criterion. For each 8×8 block in B_8×8 MB mode, there are two candidate modes to be selected, i.e. direct mode (B_8×8_direct) and normal f/b/s mode (B_8×8_f/b/s). Thus, there are still seven candidate modes to be checked and selected, which are respectively skip/direct, 16×16, 16×8, 8×16, 8×8_f/b/s, 8×8_direct, and the preselected intra mode based on WSAD criterion.

System clock frequency is crucial for hardware venture and the circuit power. In the following, we will quantitatively analyze the relationship between throughput and system clock frequency. Hardware pipeline technology is employed to achieve the desired high throughput and improve the circuit utilization, and five-stage block level mode decision pipeline structure is adopted as shown in Fig. 4 in this work.

The system clock frequency and the mode decision structure parallelism are directly determined by the candidate modes to be checked. Each MB has six blocks, and seven candidate modes result in 42 blocks to be processed. Suppose T is the cycle consumption at each stage

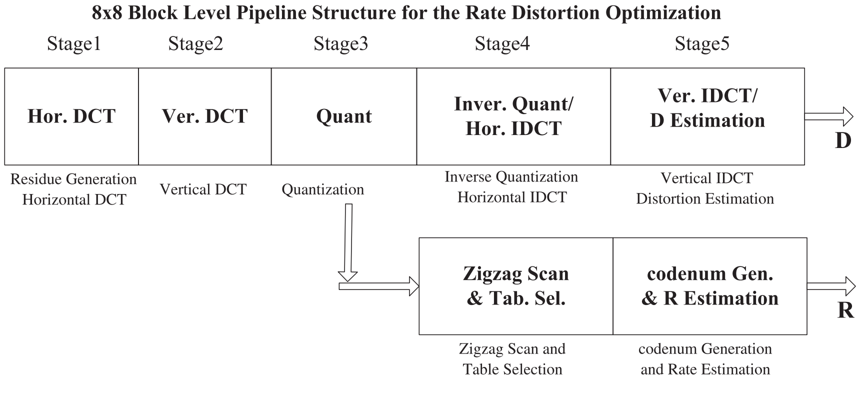

8x8 Block Level Pipeline Structure for the Rate Distortion Optimization
Stage1
Stage2
Stage3
Stage4
Stage5
Hor. DCT
Ver. DCT
Quant
Inver. Quant/ Hor. IDCT
Ver. IDCT/ D Estimation
D
Residue Generation Horizontal DCT
Vertical DCT
Quantization
Inverse Quantization Horizontal IDCT
Vertical IDCT Distortion Estimation
Zigzag Scan & Tab. Sel.
codenum Gen. & R Estimation
R
Zigzag Scan and Table Selection
codenum Generation and Rate Estimation

Fig. 4. Five-stage block level mode decision pipeline structure.

in the block level pipeline. With the pipeline setup cycle consumption considered, there are 50T cycles approximately for 1 MB mode decision in B frames, which is just the worst case. Parallel structure is adopted in the mode decision architecture, and T is 25 cycles approximately. As a result, the MB level pipeline cycle consumption is 1250 cycles. If 1080P 30 fps format is targeted, the system clock will approach to 306 Mhz (8160 MBs × 30 fps × 1250=306 Mhz). This clock frequency is still relatively high resulting in high power consumption and technology venture. Hence further simplification is still desired.

In fact, different intra and inter modes have different probabilities of being selected. Figs. 5 and 6 show the probability statistics of inter and intra modes of ten typical 720P test sequences such as Night, Sailormen, city, Spincalendar, etc. According to Fig. 5, the skip/direct inter mode occurs with the highest probability. Also, there are three or four inter modes occur with relatively high probabilities, although the modes vary in different test sequences.

On the other hand, there is also intrinsic relationship between WSAD distribution and optimal mode selected by RDO based mode decision. We find that the mode with the smallest WAD value usually is the optimal mode in the sense of RDcost criterion. Certainly, these two modes mismatch sometimes too. If we can preselect partial modes based on the WSAD criterion, what about the matching probability between these preselected modes and the optimal mode selected by RDcost criterion? We had made investigation on this mode matching statistics. Fig. 7 gives this mode matching probability between two and three candidate modes and the optimal mode, which are selected by WSAD criterion and the RDcost criterion, respectively. According to Fig. 7, the matching probability varies from 0.6 to 0.8 in the case of two candidate modes; while the probability varies from 0.8 to 0.99 in the case of three candidate modes. These statistics results are obtained using ten standard 720P test sequences.

With this conclusion, we can preselect three inter modes with higher probabilities based on the WSAD criterion. Then, the selected three inter modes, the selected intra mode, and the skip/direct mode are checked using the RDO based mode decision. As a result, there are only 30 blocks to be processed by the mode decision pipeline, and 35T cycles are enough for one MB mode decision with pipeline setup considered, which will be analyzed in Section 4. Thus, 108 and 220 MHz system clock frequency are enough for 720P and 1080P 30fps real-time video coding, respectively.

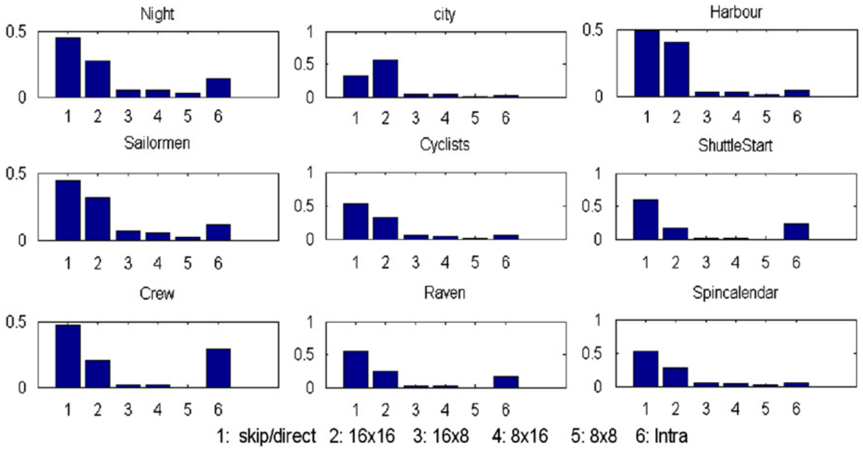


Fig. 5. Inter coding mode statistics results in P/B frame.

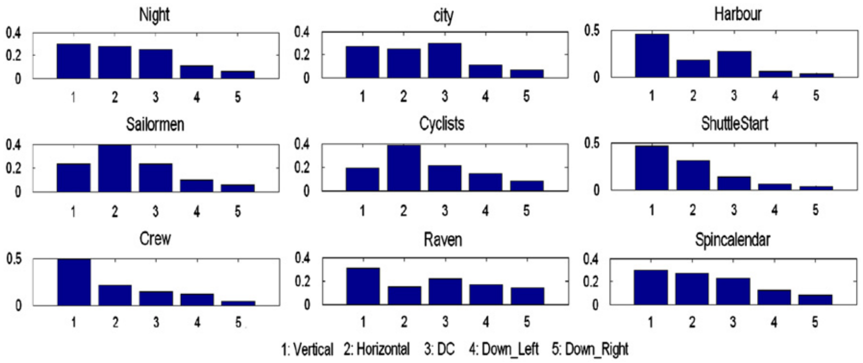


Fig. 6. Intra coding mode statistics results in I frame.

Relatively, the number of candidate intra modes is small, and the throughput burden in I frames is not as high as those of P and B frames. Hence intra mode early preselection is not used in I frames. The simplified mode decision algorithm achieves fast decision speed by mode preselection and breaking the dependency between temporal prediction direction and MB partition mode. The resulting performance is almost negligible, and the results will be given in Section 5.

4. VLSI architecture for RDO based mode decision

4.1. Data dependency removal in VLSI architecture

An important problem in mode decision VLSI architecture design is the block level data dependency due to intra prediction in I and P/B frames. This data dependency breaks the normal pipeline rhythm for intra prediction and mode decision. Data dependency should be broken to improve the pipeline efficiency. Chrominance intra prediction in AVS has not data dependency, and the critical dependency is the block level luminance intra prediction. An intelligent mode decision scheduling mechanism is proposed in Figs. 8 and 9 to eliminate the data dependency in P/B and I frames, respectively.

Inter mode decision scheduling in P and B frames in Fig. 8 is used as the illustration example. First, intra modes of B_{00}, U, and V blocks are successively fed to the pipeline for RDcost estimation. Then four luminance blocks and U, V blocks of the skip/direct modes are followed. At the time of 6T, the intra mode of B_{00} has finished the pipelining and the reconstructed pixels are ready. During the period from 7 to 8T, the intra modes of B_{01} is preselected based on WSAD criterion and then intra mode RDcost calculation for B_{01} can initiate at the time of 10T. Using the same mechanism, intra mode RDcost calculation of B_{10} and B_{11} are inserted between luminance blocks and initialized at the time of 17 and 24T. With this intelligent pipeline scheduling strategy, the data dependency problem of intra prediction is solved in P and B frames with 100% hardware utilization efficiency.

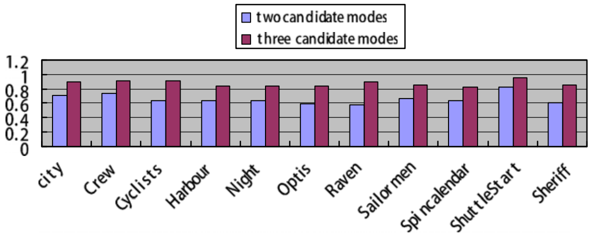


Fig. 7. Mode matching probability between two and three candidate modes and the optimal mode using the criterion of RDO.

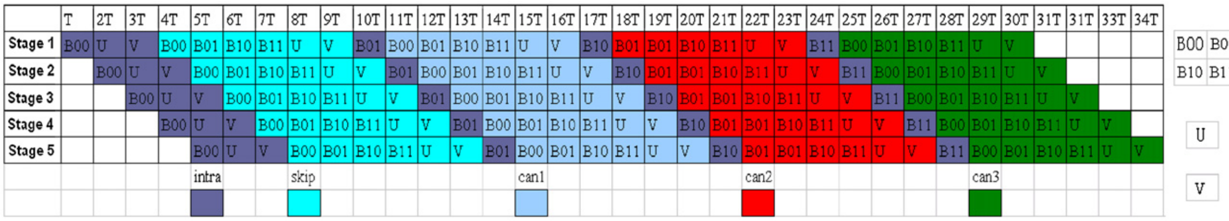
Similarly, the intra mode decision scheduling mechanism in Fig. 9 is implemented with inevitable utilization discount, in which the period from 15 to 18T is idle to wait for pixel reconstruction. The RDO based intra mode decision for I frame can achieve 85.7% hardware utilization efficiency.

4.2. VLSI architecture

The proposed VLSI architecture for RDO based mode decision is given in Fig. 10. Seven prediction versions of the current MB are buffered in the Ping-pang buffers for data share between fractional motion estimation (FME), intra prediction (IP), and mode decision (MD). Seven prediction MB buffers (Pred. Pels. Buf.) from no. 1 to no. 7 store the predicted versions of the 16 × 16_f/s/b, 16 × 8_f/s/b, 8 × 16_f/s/b, 8 × 8_f/s/b, 8 × 8_direct, intra_preselected and direct/skip modes in P and B frames, In each mode, there are six blocks B_{00}, B_{01}, B_{10}, B_{11}, U, and V. Also, the current MB is also buffered for fluent MD pipeline.

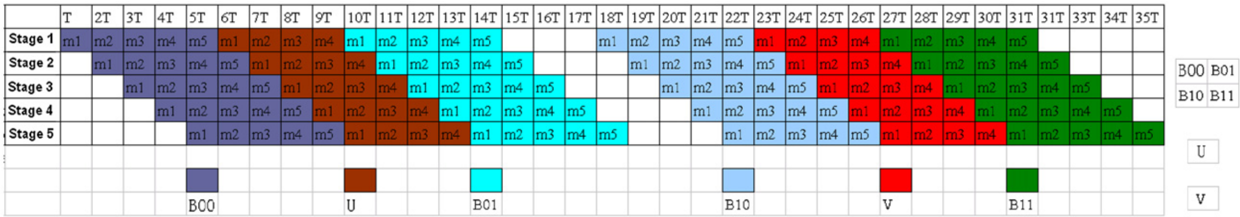
To achieve the throughout of T=25 cycles at MB level mode decision pipeline mentioned in Section 4.1, eight pixels of one line in the original block and its predicted block are fetched from the buffers in each cycle and fed into the residue generation (residue Gen.) module to calculate the residue in parallel. The integer DCT in AVS is based on 8 × 8 block. Horizontal DCT (Hor. DCT) and vertical DCT (Ver. DCT) should be processed in turn. Thus, Hor. DCT and Ver. DCT are arranged into adjacent block level pipeline stages to achieve high throughput with the transpose DCT buffer. Eight residue pixels in one line are fed into the Hor. DCT module in parallel.

Quantization (Quant) in AVS needs two multipliers with wide bit width. If eight-way parallel structure is employed. The circuit consumption of multiplexers will be high. Thus, we adopt four-way parallel structure for the Quant module. Different data throughput rate between Ver. DCT and Quant are buffered and balanced by the Quant buffer. The quantized coefficients are buffered to the inverse quantization (Inver. Quant) module and zigzag buffers simultaneously with necessary data store and format transform for the following zigzag scan and Inver Quant modules. Inverse quantization is very simple, thus it is combined with Hor. IDCT at the same stage. Hor. IDCT and Ver. IDCT are similar with those of DCT. The Ver. IDCT



	T	2T	3T	4T	5T	6T	7T	8T	9T	10T	11T	12T	13T	14T	15T	16T	17T	18T	19T	20T	21T	22T	23T	24T	25T	26T	27T	28T	29T	30T	31T	31T	33T	34T
Stage 1	B00	U	V	B00	B01	B10	B11	U	V	B01	B00	B01	B10	B11	U	V	B10	B01	B01	B10	B11	U	V	B11	B00	B01	B10	B11	U	V				
Stage 2		B00	U	V	B00	B01	B10	B11	U	V	B01	B00	B01	B10	B11	U	V	B10	B01	B01	B10	B11	U	V	B11	B00	B01	B10	B11	U	V			
Stage 3			B00	U	V	B00	B01	B10	B11	U	V	B01	B00	B01	B10	B11	U	V	B10	B01	B01	B10	B11	U	V	B11	B00	B01	B10	B11	U	V		
Stage 4				B00	U	V	B00	B01	B10	B11	U	V	B01	B00	B01	B10	B11	U	V	B10	B01	B01	B10	B11	U	V	B11	B00	B01	B10	B11	U	V	
Stage 5					B00	U	V	B00	B01	B10	B11	U	V	B01	B00	B01	B10	B11	U	V	B10	B01	B01	B10	B11	U	V	B11	B00	B01	B10	B11	U	V
					intra			skip							can1							can2							can3					
					🟪			🟦							🟦							🟥							🟩					

Fig. 8. Pipeline scheduling strategy for intra and inter mode decision in P and B frames.



	T	2T	3T	4T	5T	6T	7T	8T	9T	10T	11T	12T	13T	14T	15T	16T	17T	18T	19T	20T	21T	22T	23T	24T	25T	26T	27T	28T	29T	30T	31T	31T	33T	34T	35T
Stage 1	m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				
Stage 2		m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5			
Stage 3			m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5		
Stage 4				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5	
Stage 5					m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5				m1	m2	m3	m4	m5	m1	m2	m3	m4	m1	m2	m3	m4	m5
					B00					U				B01								B10					V				B11				

Fig. 9. Pipeline scheduling strategy for intra mode decision in I frames.

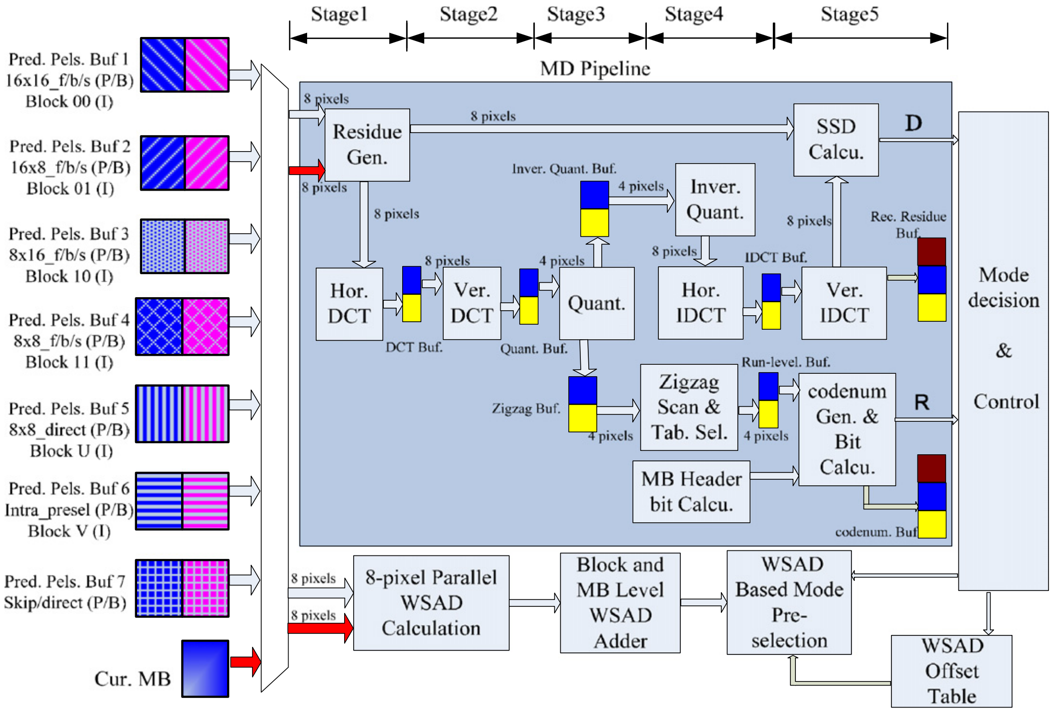


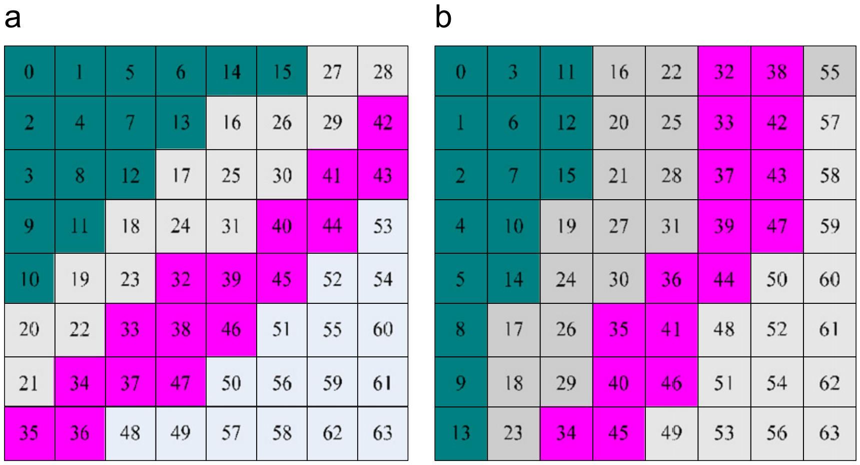
Fig. 10. Proposed VLSI architecture of RDO based mode decision.

module produces the reconstructed residue, which is fed to SSD calculation (Calcu.) module for SSD calculation and MB reconstruction by the reconstructed residue buffer.

At least 64 cycles are necessary for zigzag scan to detect the run-level pairs if no parallel architecture is employed. To achieve the desired throughput, we adopt four-way parallel architecture in the zigzag scan and run-level pair detection module. The 64 consecutive coefficients in a block are separated into four groups shown with different colors according to one of two zigzag scan orders shown in Fig. 11. Four coefficients from four groups are inputted in the four-way parallel zigzag scan module in each cycle, and each way scans 16 consecutive coefficients. After 16 cycles, the run-level pairs checked by four-way parallel structure are further concatenated to revise the run level discontinuity caused by coefficients separation. The final run level pair results are stored in a four-way parallel double-buffered run-level buffer.

There are seven intra and seven inter VLC tables, respectively, for luminance. Each run-level pair will use different tables. We obtain the corresponding table numbers of all run-level pairs during zigzag scan, which is done by table selection (Tab. Sel.). Then, the codenum generation (codenum Gen.) module will calculate the codenum values of all run-level pairs and predict the bit consumption by bit calculation (Bit Calcu.) module with the MB head bits considered. The coding bits of the run-level pairs, motion vector, and the coding parameters (such as mb_type, cbp etc.) are summed up to obtain the coding bits (R) of a certain mode. Finally, R and D (SSD) are fed in the RDcost comparison (Comp.) and control module to calculate RDcost for mode decision, state machine translation and data flow control.

The other parallel data processing loop is the mode preselection. The intra mode in P and B frames and the inter modes are preselected based on WSAD criterion. During the MD stage, mode preselection is simultaneously done by employing the eightpixel parallel WSAD calculation, block and MB level WSAD adder, and the WSAD based mode preselection modules. WSAD offset table interface is used for algorithm optimization for WSAD based mode preselection in the future. The method in [20] is used for WSAD offset initialization and refreshment.



a

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

b

0	3	11	16	22	32	38	55
1	6	12	20	25	33	42	57
2	7	15	21	28	37	43	58
4	10	19	27	31	39	47	59
5	14	24	30	36	44	50	60
8	17	26	35	41	48	52	61
9	18	29	40	46	51	54	62
13	23	34	45	49	53	56	63

Fig. 11. Two scanning modes in AVS.

The codenum fields of one MB of the optimal coding mode are buffered in the codenum buffer for data share between MD and the following bitstream generation (BG). BG generates the final bitstream simply according to codenum fields instead of quantized coefficients using *k*th Exp-Golomb coding. Redundant run-level, table selection, VLC table are eliminated by codenum data share between MD and BG instead of quantized coefficients.

4.3. Table selection

AVS adopts 2D context adaptive variable length coding (2D-CAVLC), which is the implemented by combination of Exp-Golomb coding and 2D-VLC. In 2D-CAVLC, two-dimensional DCT-quantized coefficients of an 8×8 block are translated into a sequence of run-level pairs by zigzag scan, in which *level* indicates the magnitude of a nonzero coefficient and *run* indicates the number of successive zeros before the nonzero coefficient. Statistical research shows that there are different probabilities in the case of different run-level combinations. It means that there is strong statistical correlation existing in different run-level pair combinations. To fully utilize this statistical correlation, AVS adopts multiple 2D-VLC tables for different run-level pairs, and the tables are selected context adaptively.

There are nineteen 2D-VLC tables used in AVS, they are VLC0_Intra∼VLC6_Intra, VLC0_Inter∼VLC6_Inter and VLC0_Chroma∼VLC4_Chroma respectively. Each run-level pair is mapped and denoted with a codenum by looking up one of these nineteen 2D-VLC tables. Each table contains a specific relationship between run-level pairs and their codenum values respectively. Finally, the codenum is coded using *k*th Exp-Golomb coding.

Two adjacent run-level pairs in the same block may use different VLC tables, and the table selection rule is described with the following pseudo C model shown in Fig. 12. Here, level is the magnitude of the nonzero coefficient of the next run-level pair, and current_tablenum and next_tablenum are the table indices of the current and the next run-level pairs, respectively.

```
static const int incVlc_intra[7] = { 0,1,2,4,7,10,3000};
static const int incVlc_inter[7] = { 0,1,2,3,6,9,3000};

if(abs(level) > incVlc_intra[current_tablenum])
{
    if(abs(level) <= 2)         next_tablenum = abs(level);
    else if(abs(level) <= 4)    next_tablenum = 3;
    else if(abs(level) <= 7)    next_tablenum = 4;
    else if(abs(level) <= 10)   next_tablenum = 5;
    else                        next_tablenum = 6;
}
if(abs(level) > incVlc_inter[current_tablenum])
{
      if(abs(level) <= 3)         next_tablenum = abs(level);
      else if(abs(level) <= 6)    next_tablenum = 4;
      else if(abs(level) <= 9)    next_tablenum = 5;
      else                        next_tablenum = 6;
}
```

Fig. 12. VLC table switch rule in AVS video coding standard.

As shown in Fig. 4, table selection is simultaneously done by combination with zigzag scan and run-level coding. This measure is adopted to synchronize the rate and distortion estimation in five-stage block level pipeline. According to former analysis, table selection for VLC tables is context-adaptively. If zigzag scan for 64 coefficients in a block is sequentially processed, the table selection can be easily done without any difficulty. However, the pipeline period *T* is approximately 64, which is larger than the desired threshold 25. Thus, four-way parallel architecture is used for zigzag scan to achieve the desired throughput. The discontinuities of run level pairs as shown in Fig. 11 produced by the four-way parallel structure directly complicate the context-adaptive table selection. Thus, dedicated optimization is desired for VLC table selection to achieve the throughput of 25 cycles for one block processing.

We adopt four-way table selection array (TSA) indexed from 0 to 3, and 4 TSA modules are associated with four-way zigzag scan modules shown in Fig. 13. In the first TSA,

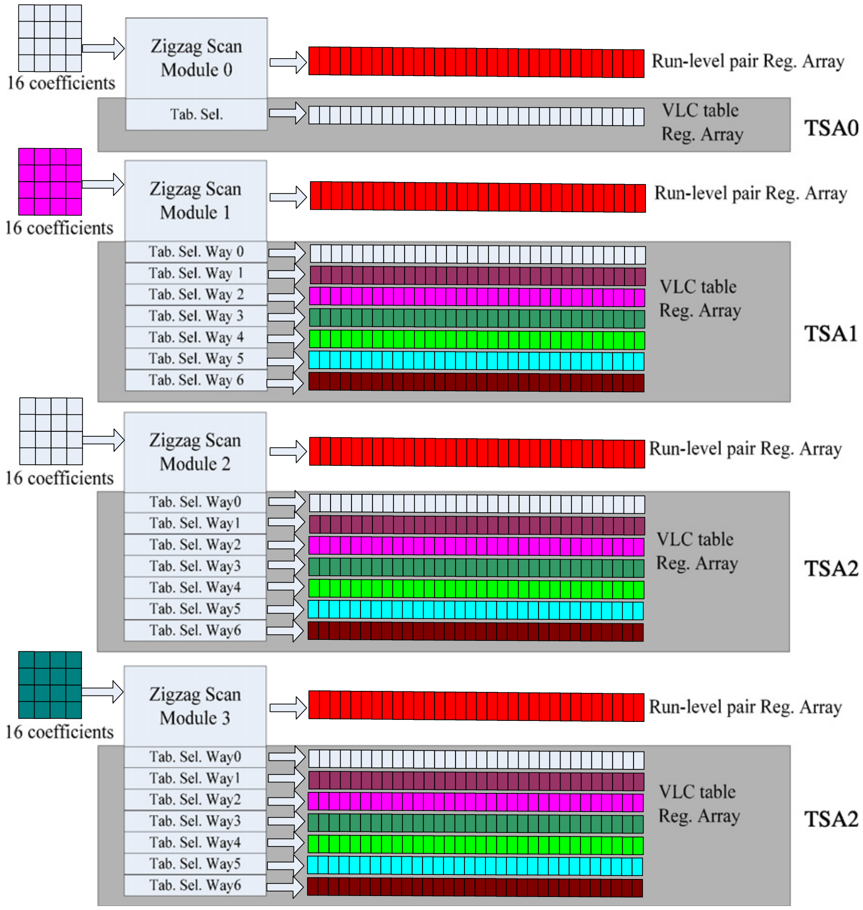


Fig. 13. Four-way parallel structure for zigzag scan and table selection.

the first 16 coefficients are sequentially checked and the table indices of all run-level pairs are determined according to the rule in Fig. 12. In the second TSA, the table number of the first nonzero coefficient is context-dependent to the last nonzero coefficient in the first TSA. The table number of the first nonzero coefficient in the third TSA is dependent on the last nonzero coefficient in the second TSA. Similarly, this dependency also exists in the fourth TSA and the third TSA. In order to break this dependency for simultaneous table selection during the zigzag scan stage, we adopt seven way daisy chains for table selection in the TSA1, TSA2, and TSA3 as shown in Fig. 13.

There are seven VLC tables possibly used in intra or inter luminance blocks, and only five tables in chrominance blocks. Thus, the table of the first nonzero coefficient in TSA1, TSA2, or TSA3 may be one of seven tables. Seven way daisy chains have checked all possible table switches due to the uncertainty of the table number of the first coefficient in TSA1, TSA2, and TSA3. Table switch control is very simple and the circuit consumption due to seven parallel structure of TSA is very small. With this parallel TSA structure, data dependency in table selection due to four-way parallel zigzag scan is eliminated. After 16 cycles, the four-way VLC table chains are connected together to finish table decision at the zigzag scan stage. One example is shown in Fig. 14 to illustrate the table connection. There are eight nonzero coefficients in the first TSA, and the last table index of the last nonzero coefficient is 2, which is decided during the zigzag scan of 16 sequential cycles. Simultaneously, all table switch cases are checked as shown in Fig. 14 by the seven way daisy chains in the TSA1, TSA2, and TSA3. There are 6, 6, and 4 nonzero coefficients in TSA1, TSA2, and TSA3 respectively. After 16 cycle zigzag scan, one table switch path is selected and connected for final table selection. The final table chains selected is illustrated with the boldfaced black lines.

In addition, VLC table reuse technology in [22] is adopted in this work for VLC table on-chip ROM usage.

4.4. *Seamless integration into video encoder architecture*

Another important problem is how to integrate the proposed mode decision VLSI architecture into the whole video encoder architecture. In fact, we had made systematic algorithm and architecture investigation on AVS video encoder, including multiresolution integer motion estimation [23], symmetric fractional pixel motion estimation [25], and efficient macroblock pipeline

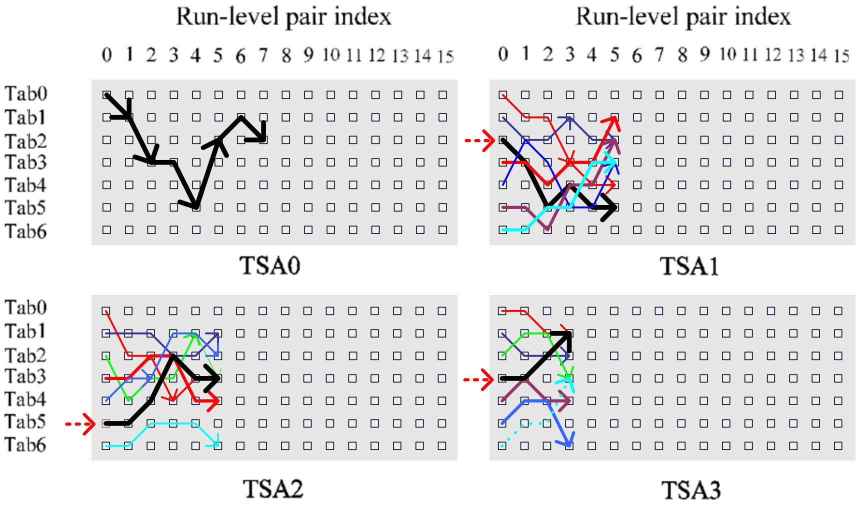


Fig. 14. Table selection example with data dependency elimination.

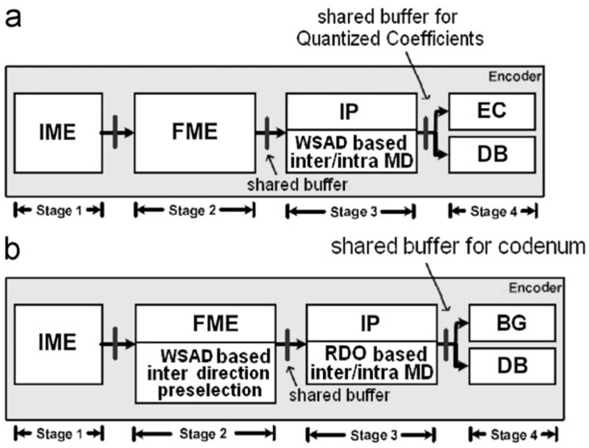


Fig. 15. Traditional four stage MB pipeline and the improved MB pipeline structure with the proposed RDO based mode decision.

structure [26]. The whole MB level pipeline structure was analyzed in [26], in which RDO based mode decision was integrated into four stage MB pipeline structure with on-chip buffer optimization.

Traditional four stage MB pipeline with WSAD based mode decision and the improved MB pipeline structure with the proposed mode decision is compared in Fig. 15. In traditional MB pipeline structure, entropy coding (EC) is at the fourth stage, and the modules such as zigzag scan, run-level pair detection, table selection, Exp-Golomb coding, and bitsteam generation are all the submodules of EC. In the improved pipeline structure, the modules including zigzag scan, run-level pair detection, and table selection are all arranged at the third stage (WSAD based intra/inter mode decision). Only Exp-Golomb coding and bitstream generation are arranged at the fourth stage (BG).

As a result, efficient circuit reuse is achieved on the modules such as zigzag scan, run-level pair detection, table selection between IP/MD and BG/DF stages, the MB codenum SRAM is employed to store the codenum fields of all coefficients in the blocks of the selected optimal mode. Thus, bitstream can be easily generated at the following BG stage according to the codenum using Exp-Golomb coding.

5. Simulation results and conclusions

The modified mode decision algorithms based on AVS reference code RM52J are tested for performance evaluation. Seven 720P standard video sequences with different intrinsic image characteristics are used for simulation. They are “city”, “Sailormen”, “Night”, “Optis”, “Harbour”, “Crew”, “Raven”, and “Cyclists” respectively. The frame rate is 30fps. IPBBPBB format with GOP length 15 is used. Search range 256×192 is used for variable block size 1/4 pixel motion estimation, in which one forward and one backward reference frames are used for B frame. All inter and intra coding modes are supported and RDO based mode decision is adopted.

We had evaluated the rate distortion performance differences among the genuine RDO based MD (RDO), WSAD based MD (no-rdo), the typical mode decision Ref. [21], and the proposed RDO based MD algorithm. In the proposed AVS encoder C model, hardware friendly multiresolution motion estimation algorithm [23] with simplified motion vector prediction [24] and symmetric fractional pixel motion estimation [25] is adopted for VLSI implementation simplification. These algorithm simplifications are both used in the proposed MD and other reference MD algorithms for fair comparison.

The rate distortion results are shown in Figs. 16 and 17. The simulation results indicate that the proposed RDO based MD algorithm can achieve superior coding performance compared with no-rdo one. The Ref algorithm gives similar performance to the proposed modified mode decision algorithm. Also, the average PSNR degradation of

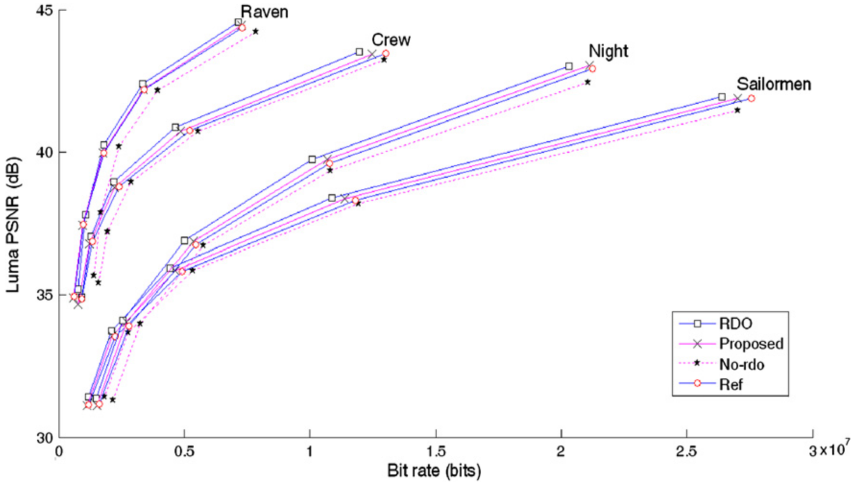


Fig. 16. Rate distortion curves of four 720P test sequences.

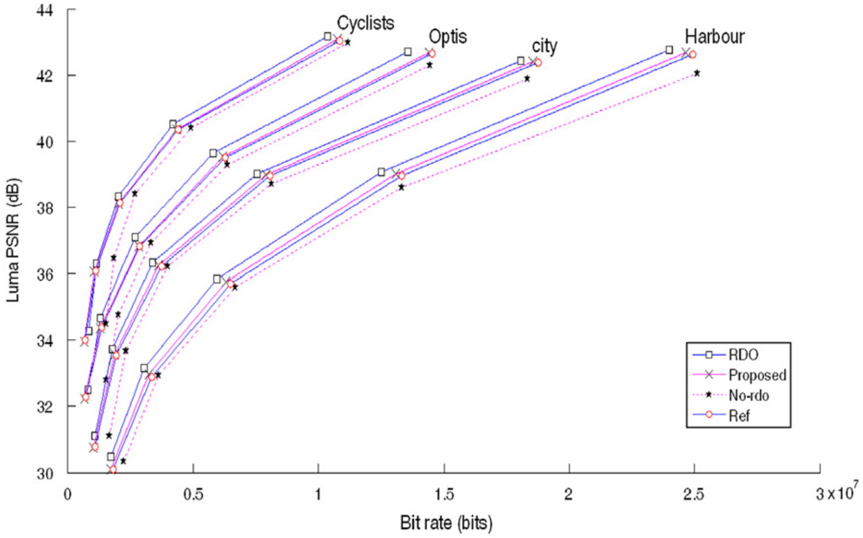


Fig. 17. Rate distortion curves of four 720P test sequences.

the proposed MD algorithm compared with the RDO based MD algorithm is relatively small.

The "Harbour" sequence is critical to mode decision algorithm due to complex motion patterns and occlusion. Thus, we give its decoded images of four algorithms for subjective image quality comparison in Fig. 18. Q_p=40 is used and rate control is turned off to test image quality of the medium bit rate case. We can find image quality loss of the proposed mode decision algorithm relative to the genuine RDO algorithm (RDO) is almost negligible.

Although it is claimed that RDO based MD is used, the actual additional implementation complexity is not very high due to algorithm simplifications. On the one hand, simplified WSAD based intra mode decision algorithm is used in P and B frames. On the other hand, inter mode preselection technology is used also in P and B frames. These two measures greatly alleviate the throughput burden and computation consumption.

For visualization evaluation, we show the mode decision results including MB partition and prediction direction using MATLAB. QVGA format test sequence "football" is used for example for enough illustration definition. The MB partition modes and the 5th frame (B frame) are shown in Fig. 19. The forward and backward prediction mode results are given in Fig. 20. Here, MB labeled "Intra" means it is a intra coded MB, and "Forward" and "Backward" means forward and backward predicted blocks, block labeled "Sym" means it is bidirectional (symmetric) predicted block, "No" means this temporal direction prediction do not exist. "Skip" means that it is skip mode MB. According to the results in Figs. 19 and 20, we can find that the mode decision is reasonable and consistent with the image characteristics.

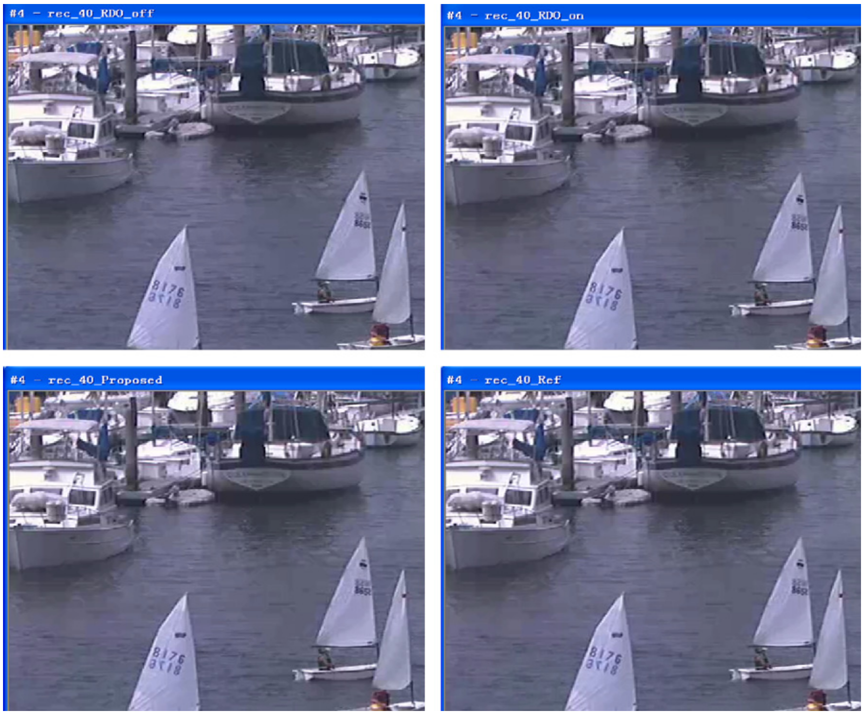


Fig. 18. Decoded image comparison example for four mode decision algorithms.

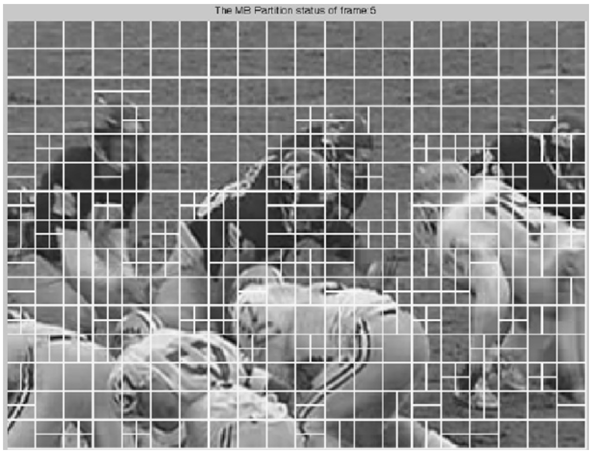


Fig. 19. MB partition results of the "football" sequence.

The proposed VLSI architecture was implemented by Verilog-HDL language and synthesized by Design complier with SMIC 0.18 μm 1P6M standard cell library.

The proposed architecture was verified on Virtex5 FPGA ASIC development system. Efficient hardware architecture results in reasonable area, 17% slice consumption of V5LX330 with RDO based MD for all inter and intra modes with data dependency avoidance. The equivalent ASIC circuit gate number is nearly 220 k. 220 Mhz is enough for real-time 1080P 30fps video coding.

Hardware modules for DCT, inverse DCT, quantization, inverse quantization, zigzag scan, and entropy coding are all indispensable even though in no-rdo mode decision architecture. These modules are desired both in the RDO based MD algorithm and bitstream coding for the final selected mode. Efficient hardware reuse mechanism is achieved by employing the on-chip codenum buffer and the reconstructed residue buffer. The additional hardware cost in the proposed architecture is derived from parallel architectures adopted in the quantization, inverse quantization, and zigzag scan modules. Thus, the hardware cost consumption increment is relatively small. The detailed parameters are given in Table 1. According to the results in Table 1, the additional hardware cost of the proposed architecture compared with no-rdo one is approximately 90 K gates, which is moderate. Currently, no-RDO based architecture is available, thus no reference is used for hardware cost comparison.

The proposed algorithm modifications are well-suited for VLSI implementation, and balance between performance and complexity is achieved. These algorithm modifications are also well-suited for high definition main profile H.264 video encoder. Also, the proposed VLSI architecture for mode decision is also well-suited for H.264 video encoder with small architecture modification on the DCT, IDCT, Zigzag scan, and VLC modules.

Because inter and intra mode preselection is done based on WSAD criterion. As shown in Fig. 7, although the mode matching probability in the case of three candidate modes is high usually, the mismatch probability is no smaller than 15% in some special test sequences such as "Harbour", "Sailormen", and "Spincalendar". The performance loss of the proposed mode decision algorithm is just derived from the mode mismatch in mode preselection. WSAD offset table interface is reserved for future

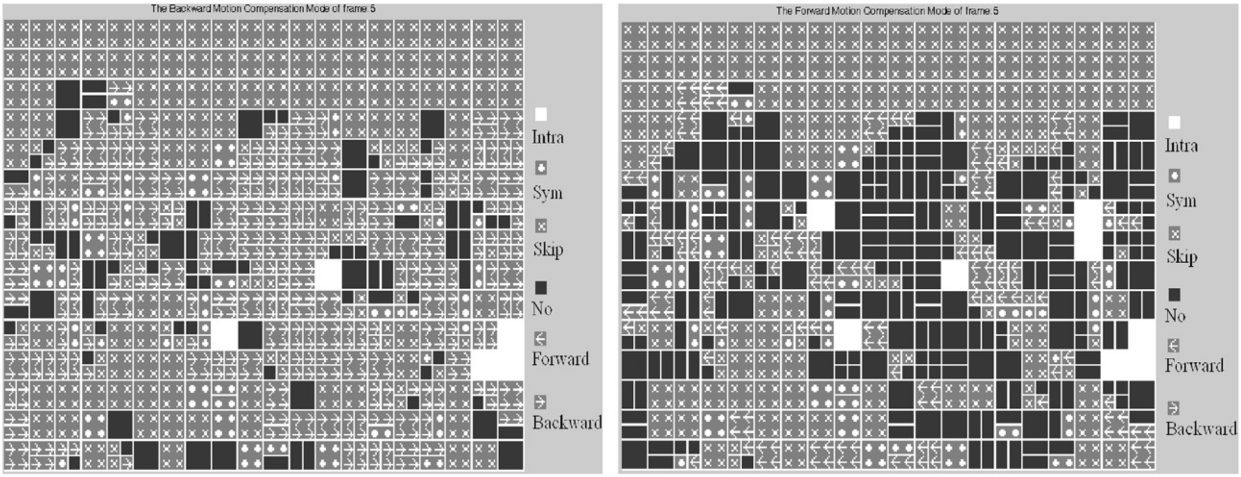

The Backward Motion Compensation Mode of frame 5
Intra
Sym
Skip
No
Forward
Backward
The Forward Motion Compensation Mode of frame 5
Intra
Sym
Skip
No
Forward
Backward

Fig. 20. Temporal prediction results of the "football" sequence.

Table 1
Performance of the proposed MD architecture.

Item	Parameters
Cycle/MB	900 cycles
Capacity	108 MHz (720P 30 fps) 220 MHz (1080P 30fps)
SRAM	9 K bytes
Modes	All intra and inter modes
Gate count	17% slice of FPGA V5LX330 220 K ASIC gates (SMIC 0.18 μm 1P6M)
Hardware cost increase versus no-rdo methods	90 K ASIC gates (SMIC 0.18 μm 1P6M)

algorithm optimization. Future algorithm optimization will focus on WSAD offset for more accurate mode preselection. The WSAD offsets will be content-adaptively estimated and assigned to all candidate modes, this method is referred in Ref. [20].

Acknowledgements

This work was supported by NSFC 60802025, China Postdoc Science Foundation Project 200902015, and Funding Project of Science and Technology Department of Zhejiang Province 2010C31G2080035.

References

[1] Information technology – Advanced coding of audio and video – Part 2: Video. AVS Standard Draft 2005.

[2] Hai bing Yin, Xi Zhong Lou, Zhe Lei Xia, An efficient VLSI architecture for rate distortion optimization in AVS video encoder, in: Proceedings of the IEEE International Symposium on Circuits and Systems, pp. 2805–2808, May 2008.

[3] Zhenyu Liu, Yang Song, Satoshi Goto, HDTV 1080P H.264/AVC encoder chip design and performance analysis, IEEE J. Solid-state Circuits 44 (2) (2009) etc.

[4] T.C. Chen, S.Y. Chien, Y.W. Huang, C.H. Tsai, C.Y. Chen, T.W. Chen, L.G. Chen, Analysis and architecture design of an HDTV720p 30 frames/s H.264/AVC encoder, IEEE Trans. Circuits Syst. Video Technol. 16 (5) (2006).

[5] Y. W. Huang, T. C. Chen, et al., A 1.3 TOPS H.264/AVC single-chip encoder for HDTV applications, in: Proceedings of the IEEE International Solid-State Circuits Conference, San Francisco, CA, USA, Feb. 2005.

[6] Datasheet of Image and Video Compression IP, [Online]. 〈http://www.cast-inc.com〉.

[7] Datasheet of VXE Video Encoder IP Core Family, [Online]. 〈http://www.imgtec.com〉.

[8] Datasheet of H.264 Video Encoders and Decoders, [Online]. 〈http://www.4i2i.com〉.

[9] Datasheet of Multimedia IP, [Online]. 〈http://www.globalunichip.com〉.

[10] Datasheet of Multimedia IP, [Online]. 〈http://www.faraday-tech.com〉.

[11] The Fujitsu Website. [Online]. 〈http://www.fujitsu.com〉.

[12] Datasheet of High-Definition Encoding and Decoding, [Online]. 〈http://www.maxim-ic.com〉.

[13] D. Wu, F. Pan, K.P. Lim, S. Wu, Z.G. Li, X. Lin, S. Rahardja, C.C. Ko, Fast intermode decision in H.264/AVC video coding, IEEE Trans. Circuits Syst. Video Technol. 15 (6) (2005) 953–958.

[14] Y.K. Tu, J.F. Yang, M.T. Sun, Efficient rate-distortion estimation for H.264/AVC Coders, IEEE Trans. Circuits System Video Technol. 16 (5) (2006).

[15] Changsung Kim, C.-C.Jay Kuo, Feature-based intra-/intercoding mode selection for H.264/AVC, IEEE Trans. Circuits System Video Technol. 17 (4) (2007).

[16] M.G. Sarwer, L.-M. Po, Fast bit rate estimation for mode decision of H.264/AVC, IEEE Trans. Circuits Syst. Video Technol. 17 (10) (2007) 1402–1407.

[17] Andy Chia Woo Yu, Graham R. Martin, Heechan Park, Fast intermode selection in the H.264/AVC standard using a hierarchical decision process, IEEE Trans. Circuit Syst. Video Technol. 18 (2) (2008).

[18] Lai-Man Po, Kai Guo, Transform-domain fast sum of the squared difference computation for H.264/AVC rate-distortion optimization, IEEE Trans. Circuits Syst. Video Technol. 17 (6) (2007).

[19] Kuniyasu, H. Kishida, T. Tian Song Shimamoto, T. Tokushima University, Tokushima, Fast Transform and Quantization Architecture with All-ZeroDetection and Bit Estimation for H.264/AVC, in: Proceedings of the IWSDA'07, pp. 334–338,September 2007.

[20] H.F. Ates, Fast inter-mode decision and selective quarter-pel refinement in H.264 video coding, in: Proceedings of the IEEE ICASSP, Las Vegas, USA, March–April 2008.

[21] Byung-Gyu Kim, Novel inter-mode decision algorithm based on macroblock (MB) tracking for the P-slice in H.264/AVC video coding, IEEE Trans. Circuits Syst. Video Technol. 18 (2) (2008).

[22] Ke Zhang, Xiaoyang Wu, Lu Yu, An area-efficient VLSI implementation of CA-2D-VLC decoder for AVS, ISCAS (2007) 3151–3154.

[23] Hai Bing Yin, Xiu Min Wang, Zhe Lei Xia Hong Gang Qi, Cost-effective multiresolution motion estimation algorithm for rate

distortion optimizedhigh definition video encoder, IEEE VLSI-SOC 2009.

[24] Wei Yang, Hai bing Yin, Wen Gao, Hong gang Qi, Xiao dong Xie, Multi-stage motion vector prediction schedule strategy for AVS HD encoder, accepted by ICCE 2010, Las Vegas, USA, January 9–13,2010.

[25] Hai bing Yin, Hong gang Qi, Xiao dong Xie, Wen Gao, Hardware oriented algorithm analysis and modification for high definition AVS video encoder VLSI implementation, in: Proceedings of the ICCE 2010, Las Vegas, USA, January 9–13,2010.

[26] Hai bing Yin, Hong gang Qi, Huizhu Jia, Don Xie, Wen Gao, Efficient macroblock pipeline structure in high definition AVS video encoder VLSI architecture, in: Proceedings of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010), Paris, France, 30 May—2 June, 2010.