Effective Intra Mode Decision and Motion Estimation Based on Edge Information

Yule Yuan[1], Jie Kong[2], Yong Zhao[*3], Xinan Wang[*4]

[*1, 3, 4]The Key Laboratory of Integrated Microsystem, School of Electronic and Computer Engineering, Peking University, China

[2]Samsung Guangzhou R&D Mobile Centre, China

[1]yuanyl@pkusz.edu.cn; [2]114253997@qq.com; [*3]yongzhao@pkusz.edu.cn; [*4]wangxa@pkusz.edu.cn

***Abstract*- In this paper, we use edge information to fast intra mode decision and improve the performance of binary block motion estimation. For intra mode decision, we present a very simple, yet efficient, algorithm in AVS video coding based on edges detection and Support Vector Machine (SVM). The technique uses the SOBEL operator to check the edge information from the total image before its intra prediction. Then we use SVM classifier which is trained by the edge points to decide the mode of block for AVS intra prediction. Because only one prediction mode is chosen for rate-distortion optimization calculation using the proposed algorithm, simulation results also show that the proposed scheme achieves up to 16% computational saving with no video quality degradation, compared with results of the existing method. For binary block motion estimation, we studied that 2BT utilizes mean and standard deviation of a transforming block to get the two bit-planes. Due to the binary blocks coming from local adaptive thresholds, the binary blocks do not take advantage of global information. In the proposed 2BTGLE method, we use one global threshold and one local threshold to get the two bit-planes. The experiments demonstrated that our algorithm gets better performance than the 2BT method.**

***Keywords*- Video Coding; Intra Prediction; Motion Estimation; SOBEL; SVM; 2BT**

I. INTRODUCTION

In the latest few years, the emerging advanced video coding standards have become the focus of consumer electronic industries and research institutes, such as HEVC, H.264, VC-1, and AVS. The AVS video coding standard was developed by the China Audio Video Coding Standard (AVS) Working Group, which was founded in June 2002. So far, AVS standards are becoming more and more internationalized. The AVS video coding standard has been accepted as an option by ITU-TFGIPTV for IPTV applications [1]. As is shown in the top-level block diagram of an AVS encoder in Fig. 1, the AVS video encoder is based on the traditional hybrid framework which incorporates many state-of-the-art techniques to achieve outstanding coding performance.

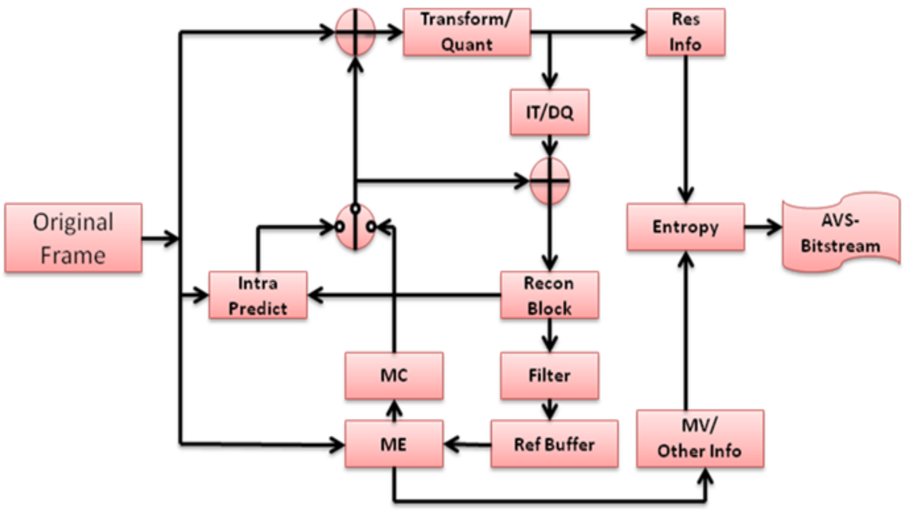

Fig. 1 The block diagram of AVS video coding system

As MPEG standard, the improvements are mainly obtained from various employments of improved intra- and inter-predictions, variable block motion estimation, and multiple reference frames. According to these features, enhanced inter and intra-prediction techniques are key elements. To achieve the highest coding efficiency, AVS also employs the rate-distortion optimization (RDO) technique to select the best coding mode to achieve the minimum rate distortion cost [2]. However, the computational complexity of the AVS video encoder is dramatically increased by this optimization technique. The encoder has to encode the target block by searching all possible modes exhaustively for the best mode in the intra-frame coding process. This comes with a cost of higher complexity encoder and higher computational time, which makes it difficult for real time applications.

Spatial prediction is used in intra coding in AVS part 2 to exploit spatial correlation of picture. Intra prediction algorithm predicts the pixels in a MB using the pixels in the available neighbouring blocks. The intra prediction is based on 8x8 block. That is because AVS part 2 is mainly targeted to high resolution, such as Standard/High Definition coding where the small block prediction is not efficient as the bigger block prediction for coding performance improvement. There are five 8x8 luma prediction modes designed in a directional manner, and four chrominance intra prediction modes, less than the number of H.264/AVC [20]. The reconstructed pixels of neighbouring blocks before de-blocking filtered is used as reference pixels for the current block. Four luminance prediction directions are illustrated in Fig. 2. The pixels 0 to 16 belong to the neighbouring blocks and are assumed to have already been encoded and reconstructed and therefore available in the encoder and decoder to generate a prediction for the current MB.

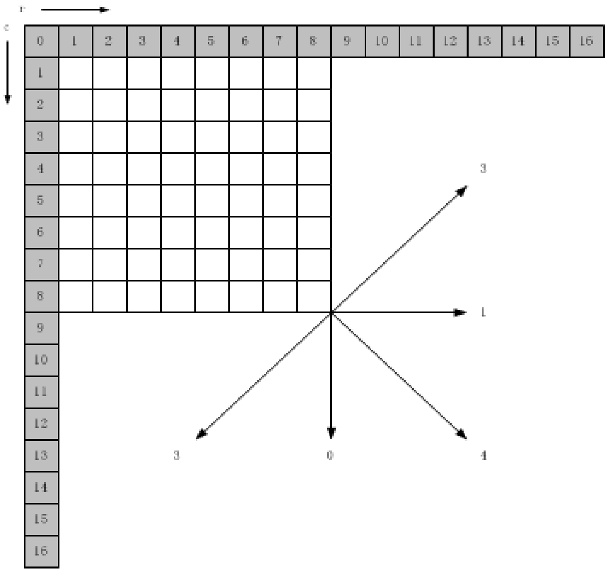

Fig. 2 Intra-prediction directions in AVS Part 2

Each 8x8 luma prediction mode generates 64 predicted pixel values using some or all of the neighbour pixels r[0..16](or c[0..16]) as shown in Fig. 3. The arrows indicate the direction of prediction in each mode. The predicted pixels are calculated by a weighted average of the neighbour pixels r[0..16] (or c[0.16]) for each mode except Vertical and Horizontal. When the neighbour samples r[8..16] (or c[8..16]) are not available, they are replaced with r[8] (or c[8]). Fig. 3 shows five intra-prediction modes in AVS Part 2. Mode 0 (vertical prediction) and Mode 1 (horizontal prediction) in AVS Part 2 are the same as that in H.264, for example, the sample at position [i, j] is predicted with r[i + 1] for Mode 0 and c[j + 1] for Mode 1. For the other modes, the filtered neighbour samples in the predictions direction will be used as prediction samples.

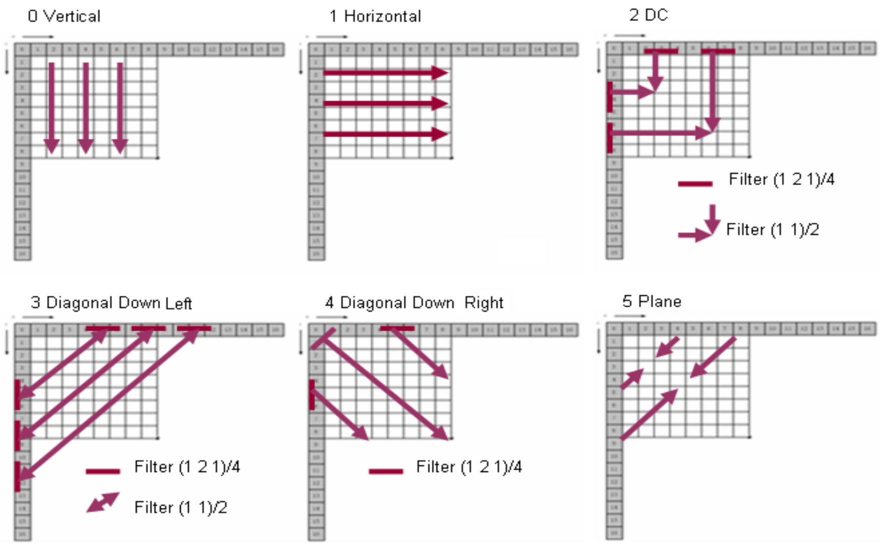

Fig. 3 The intra-prediction modes in AVS Part 2

For the chrominance component of a MB, a predicted 8x8 chroma block is formed for each 8x8 chroma component by performing intra prediction for the MB. AVS Part 2 also has four chroma prediction modes (vertical, horizontal, DC, and Plane mode). More details about AVS intra-prediction can be seen in [21].

Meanwhile, we all know that video coding can effectively exploit the redundancy in spatial and temporal domain. Motion Estimation (ME) and compensation play a key role in reducing temporal redundancy. For a video sequence, how small it can be compressed is mainly due to inter-prediction. Video coding is primarily based on block concept. It divides a frame into non-overlapping rectangular blocks. Block based Motion Estimation is widely used for its simplicity and efficacy. The idea is to find a best match block in reference frame (from previous or later frame) to predict current block. And the displacement between the two blocks is called Motion Vector (MV). Thus, after motion estimation, the encoder just needs to transform the coefficient difference (Residual) and the Motion Vector Difference (MVD) between the two blocks. But Motion Estimation (ME) is very computation-intensive, performing up to 89.2% computation power is consumed by ME part [11].

To reduce the complexity of intra-prediction, there are many faster methods proposed for H.264. Pan el al. [3] developed a fast intra-mode decision scheme which uses edge direction histogram to measure the edge angle of 4x4 blocks and 16x16 macro block to reduce the number of probable modes for complexity reduction. Wang et al. [4] proposed a simple edge detection algorithm, which is proposed in MPEG-7 as feature descriptors, to achieve a better result in comparison to the previous algorithm [3]. The technique proposed in [5] checks the equality of the neighbouring pixel. If the neighbouring pixels used for calculating the predicted pixels by an intra 4x4 prediction mode are equal, the predicted pixels by this mode are equal to one of these neighbouring pixels. Therefore, the prediction equations simplify to a constant value and prediction calculations for this mode become unnecessary. Furthermore, if the neighbouring pixels used for calculating the predicted pixel by an intra 16x16 or an intra 8x8 prediction mode are equal, the prediction equations used by this mode simplify significantly. In this way, the amount of computations performed by H.264 intra-prediction algorithm is reduced significantly without any PSNR loss. Meanwhile, Tsai et al. [6] proposed two fast, efficient but reliable direction detection algorithms by computing sub block and pixel direction differences. Both proposed methods effectively estimate the edge direction inside the block to narrow down the predictive mode to reduce the RDO computation. In this way, the proposed method can reduce the encoding time by about 60% with negligible loss of coding performance. Cheolhong [7] applied a statistical learning method such as Support Vector Machines for Regression (SVR) to improve the performance of current H.264 intra prediction via batch learning. This scheme improves significantly the average Peak Signal-to-Noise Ratio of intra prediction than the performance of H.264. A more recent study of intra-prediction on fast intra-coding is also published [8].

Different fast intra prediction schemes have been proposed for different video codecs, but only a few works on AVS especially for AVS-Part 2 can be found in the literature. Despite using edge information for intra mode selection is a common technique. In our previous work [9], we presented an alternative method of achieving significant time saving while keeping the PSNR loss at minimal. The technique proposed in this paper uses the SOBEL operator to check the edge information from the total image before its intra-prediction. Therefore, these edges that from edge map can describe the spatial correlation well and use them to decide the 8x8 block prediction mode. Then, we employ SVM classifier which is trained by the edge points to decide the mode of the block for AVS intra-prediction.

Recently, SVM [10] has been widely used and has better performance in some applications compared to statistical classifiers and neural networks classifiers. This method uses SVM instead of neural networks in [9]. Due to the BP neural network has three layers and the middle layer exists some floating point operation. It is very time-consuming. While, the SVM classifier only operates support vectors and the input vectors are all binary numbers which remove the floating point operations and this speed up the computing.

To enhance the performance of ME, many state-of-the-art motion estimation algorithms have been put forward. One way is to use some special search patterns like three-step search (TSS) [12], 2-D logarithmic search (LOGS) [13], new three three-step search (NTSS) [14], four-step search (4SS) [15], and diamond search (DS) [16] and so on, which can markedly reduce candidate points in a search window. These algorithms are based on the assumption that the error plane is monotonic. But for realistic videos, there are more than one local minimum in the error plane. So these algorithms are easily fallen into a local minimum not an absolute minimum. There also exist some algorithms using some different matching criteria to reduce the computation load, such as 1BT [17], 2BT [18] and E2BT [19]. These algorithms use bit planes to descript the original pixel plane, so we can just use logic operations instead of arithmetic operations. But the transformation for a coding block is carried out in a local small area which contains current block. This is complicated. And these algorithms are not accurate enough. They do not consider MVS's effect in the bitrate.

The proposed binary block motion estimation algorithm transforms a pixel plane into two-bit plane through a global threshold and a local threshold based on the image's edge feature. Like FS, the proposed algorithm exhaustively searches every candidate point in a search window and picks up possible absolute minimums. For the matching criterion, it only needs some logical operations. It dramatically reduces the computation load. And for the possible absolute minimums, the proposed algorithm compares these points' bitrates in both MV and SAD consideration. Finally, the previous picked-up point is used as the start point for the other fast ME algorithms like DS. As we can see from the late experiment results, the proposed algorithm improves DS obviously. Especially for the video with large motion objects, the PSNR's improvement achieves about 0.13dB.

The rest of the paper is organized as follows. In Section II, we introduced the proposed algorithms. For intra-prediction, we use SOBEL operator to detect edges. After that, we train the SVM classifier using all the possible values of the edges. Using the detected edge information as the input set, we can decide the intra mode directly. For Motion Estimation, we introduce 2BT and descript our proposed 2BTGLE algorithm. In section III, a performance evaluation of the proposed novel algorithm is presented. Finally, the conclusion of this work is addressed in Section IV.

II. EFFECTIVE INTRA MODE DECISION AND MOTION ESTIMATION

A. Edge Information Detection

In this paper, we use the SOBEL operator [22] to detect its edge information. Technically, the SOBEL operator uses two kernels which are convolved with the image to calculate approximations of the gradient of the image – one for horizontal changes, and one for vertical. The computations are as follows:

$$xsobel(x,y)=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}\otimes I(x,y) \tag{1}$$

$$thrshold=\frac{1}{w*h}\sum_{i=0,j=0}^{w-1,h-1}(xsobel(i,j)) \tag{2}$$

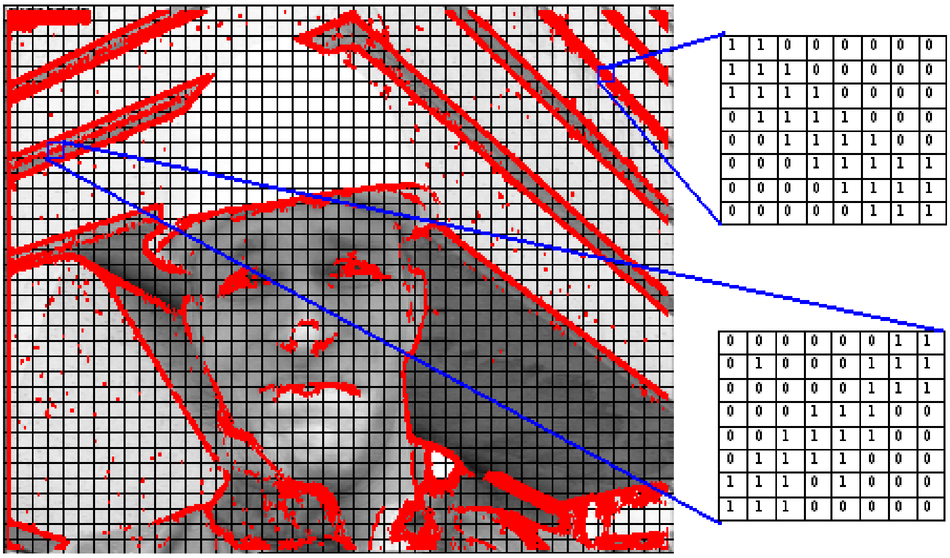

Fig. 4 An image comes from foreman sequence and its edges detected by SOBEL operator. The right two edge maps of the 8x8 blocks descript the down-right and down-left modes of the blocks

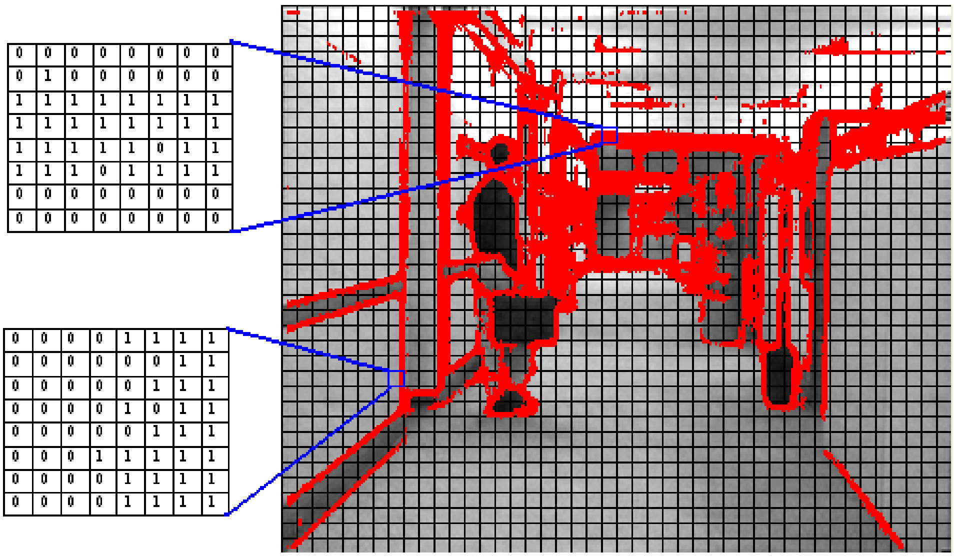

Fig. 5 An image edges from hall sequence and its edges detected by SOBEL operator. The left two edge maps of the 8x8 blocks descript the horizontal and vertical modes of the bock

Where, ⊗ denotes the 2-dimensional convolution operation. In order to speed up the algorithm, we only use the Equation (1) to get the edge information. Because the edge information got forms the image frame before it is encoded, the overall complexity of the method is reduced. The threshold value for binary is the mean of the SOBEL values, which is calculated from Equation (2). Figure 4 illustrates the edge point detection resulting from the SOBEL operator processing for one of the picture of the foreman test sequence. The left top box in Fig. 4 illustrates one 8x8 block's edge map and its direction is down-left. As well as, the left down box in Fig. 4 demonstrates one 8x8 block's edge map and its direction is right-down. Meanwhile, Fig. 5 illustrates the edge point detection resulting for one of the picture of the hall test sequence, and the two block's edge maps are illustrated. One is horizontal and the other is vertical.

B. Intra Mode Decision Based on Edge Information

Pan in [23] used SOBEL to detect the edge direction information to speed up intra-prediction. But it is very different to our scheme proposed in this paper. Pan uses edge direction histogram to measure the edge angle of 4x4 blocks and 16x16 macro blocks to reduce the number of probable modes for complexity reduction. In our previous method [9], it detects the edges of the image before its coding. From these edges we can get the spatial characteristics of the picture. Then using these edge features as the input layer, the trained neural network can decide the mode of intra prediction directly. Recently, SVM [10] has been widely used and has better performance in some applications compared to statistical classifiers and neural networks classifiers. The SVM trains a classifier by solving an optimization problem to decide which instances of the training data set are support vectors.

In order to train the SVM classifier to determine the intra mode, our algorithm is constructed by the flowing steps. Firstly, the SVM classifier is trained by all possible values of L0…L4 and c0…c63. The value of the L0 to L4 is decided by the current 8x8 sub-block location relations, as shown in Fig. 6. The value of c0 to c63 are the all possible 0, 1 combination of 8x8 edge map block. Then we manually determined the training set, such as horizontal set, vertical set, downright set and so on.

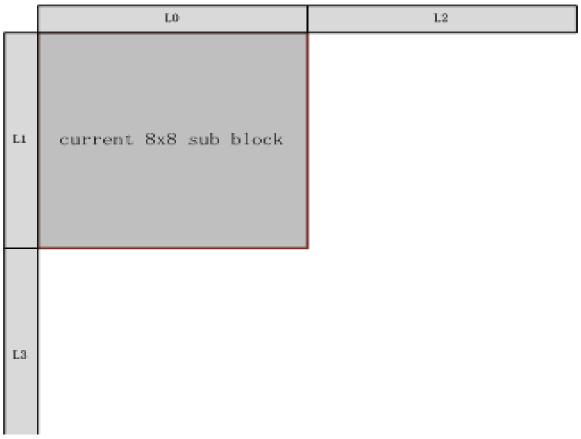

Fig. 6 The symbols L0, L1, L2 and L3 represent the current sub-block's reference blocks respectively
If the reference block exists, the value of the symbol is 1, otherwise the value is 0

After that, input the edge map of each block which comes from above section and its reference block information to the SVM classifier to decide the mode of the 8x8 block. The each value of the edge map of the 8x8 block is marked as c0, c1, …, c63, as shown in Fig. 7.

C0	C1	C2	C3	C4	C5	C6	C7
C8	C9	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	C63

Fig. 7 The flags of edge map for an 8x8 block

Because the input feature vector for the classifier is only composed by '0' and '1', this saves a lot of multiplication operations. Therefore, our proposed method is very fast to decide the mode of a block. We all know that, if the input vectors include '0' and '1' only, the support vector multiplication formula can be achieved through logical operation. More importantly, only one prediction mode is chosen for RDO calculation using our proposed algorithms. The trained SVM classifier can decide the right mode directly.

C. Improved 2BT Algorithm Based on Global and Local Edge Information

How to define a block best matches the current block, most algorithm is based on the minimal SAD. For two blocks size of M x N, the distortion between the two blocks expressed in the form of

$$SAD(i,j)=\sum_{m=1}^{M}\sum_{n=1}^{N}|S_k(m,n)-S_{k-1}(m+i,n+j)| \tag{3}$$

Where $SAD(i,j)$ stands for the distortion of the two blocks with the displacement of (m, n). In FS, we check all the candidate points in a search window. The displacement getting the minimal D is the final MV, which we need.

2BT algorithm chooses a different matching criterion to reduce the computation load. 2BT algorithm firstly transforms the multi-bit pixel value into 2 bit-planes through the mean and deviation value of the transform block. So we can just use Exclude-or(XOR) and Boolean-or operations to compute the distortion of two blocks. One bit-plane is acquired through the following way

$$B_1(i,j)=\begin{cases}1, if P(i,j)\geqslant\mu\\0, otherwise\end{cases} \tag{4}$$

Where, μ is the mean value, $P(i,j)$ is the luminance or chrome value of the pixel. The other bit-plane is obtained as below

$$B_2(i,j)=\begin{cases}1, if P(i,j)\geqslant\mu+\sigma\ or\ P(i,j)\leqslant\mu+\sigma\\0, otherwise\end{cases} \tag{5}$$

Where, σ is the standard deviation. For reducing blocking effect, the authors choose a large window (17 x 17) surrounding in the block. In [18], a block averaging kernel is used to get the mean given below

$$K(i,j)=\begin{cases}1, if i,j\in(0,4,8,12,16)\\0, otherwise\end{cases} \tag{6}$$

Moreover, there is a linear approximation function to get the approximate standard deviation as

$$\sigma\approx 15+0.125*\sigma^2 \tag{7}$$

Where, σ^2 is the variance of the transform block. After we transform the two blocks into 2 bit-planes respectively, we could use the following simple matching criteria to compute their distortion.

$$D(x,y)=\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[B_1^c(i,j)\oplus B_1^r(x+i,y+i)]||[B_2^c(i,j)\oplus B_2^r(x+i,y+i)] \tag{8}$$

From the above, we know that the two thresholds, mean and standard deviation, are adaptive. For a previous matched block, its near block may exist some correlation with it in spatial. The global information can describe it. But 2BT does not take advantage of global information. In our proposed 2BTGLE method, we use one global threshold and one local threshold to get the two bit-planes.

Like FS, the proposed 2BTGLE algorithm exhaustively searches every candidate point in a search window and picks up possible absolute minimums. For the matching criterion, it only needs some logical operations. It dramatically reduces the computation load. And for the possible absolute minimums, the proposed algorithm compares these points' bitrates in both MV and SAD consideration. Finally, the previous picked-up point is used as the start point for the other fast ME algorithms. In this paper, we use DS. The proposed algorithm transforms a pixel plane into 2 bit-planes through one global threshold and one local threshold base on the image's edge feature.

The following formulas describe how to transform multi-bits image into 2-bit-planes.

$$\mathrm{B}_1(i,j)=\begin{cases}1, if sobel(i,j)\geqslant GlobalThr\\0, otherwise\end{cases} \tag{9}$$

$$B_2(i,j)=\begin{cases}1, if P(i,j)\geqslant LocalThr\\0, otherwise\end{cases} \tag{10}$$

For global binary method, we use the method mentioned above in our intra mode decision algorithm that is from Equation (2). For local binary method, the local threshold gets from OTSU method [24].

III. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed fast intra mode decision scheme, we use the AVS reference software version rm52k (Jizhun profile) as our simulation platform on the frames of some well-known sequences, and they are Forman, Bus, and Hall. Each sequence is 50 frames for each test and CIF format is applied. These frames are all encoded by I-frames only on different QPs and RDO option is on. Entropy coding method is VLC. The experimental results in terms of PSNR (Peak Signal to Noise Ratio), BR (bit rate) and TS (Time Saving) using FS, Pan's method [3] and the proposed methods are listed in Table 1. The FS is the full search method, which is used in the reference mode rm52k program. The rm52k code complied in vs2008 C++ environment. All the results are performed on a Celeron/1.50 GHZ personal computer.

It can be seen from Table 1 that the proposed intra mode decision method based on SVM improves the encoder's efficiency. More importantly, the scheme has less video quality degradation and less bit rate gain than the performance of FIPBN. The time saving is faster than the method in [3] by about 16% and the method in [9] by about 5% respectively. It is because that the BP neural network has three layers. The middle layer exists some floating point operation. While, the SVM classifier only operates support vectors and the input vectors are some binary numbers that remove the floating point operations and the Boolean operations speed up the computing.

To evaluate the performance of the 2BTGL algorithm, several typical CIF video sequences have been tested in method of FS, DS, 2BT and 2BTGL, respectively. Each sequence is encoded of 100 frames. The first frame is encoded as I frame and the others all P-frames, such as I, P, P, P…, with the RDO off. The number of reference frames is fixed to be 1. And the quantization parameter is fixed to be 30. A block size of 16x16 with a search range of 16 pixels, a block size of 8x8 with a search range of 8 pixels have been tested, respectively.

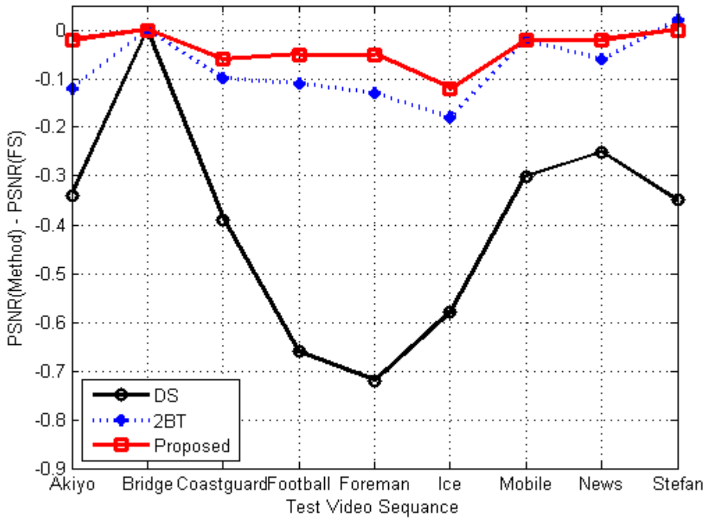

Fig. 8 PSNR experiment results of algorithms when the motion block size is 16x16 (search range = ±16)

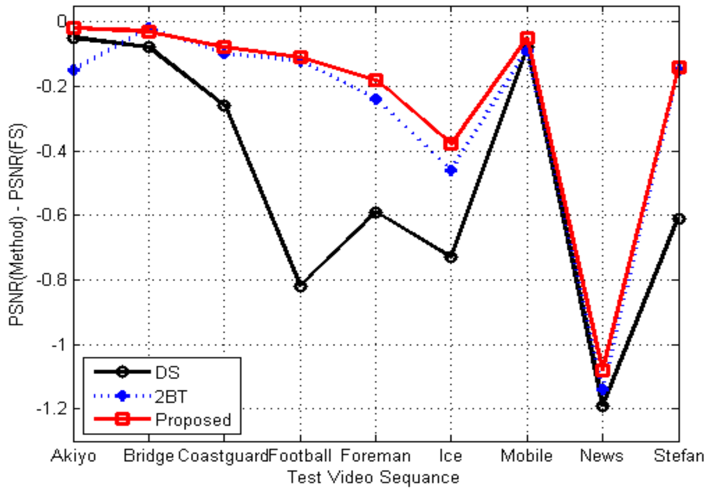

Fig. 9 PSNR experiment results of algorithms when the motion block size is 8x8 (search range = ±8)

Fig. 8 shows the PSNR comparison results on different video sequences when the motion block size is 16×16 and the search range is ±16. Fig. 9 shows the PSNR comparison results when the motion block size is 8×8 and the search range is ±8. We can see that the performance of the 2BTGLE algorithm is always better than the other algorithms. It is obvious to conclude that the proposed 2BTGLE method gets better performance than 2BT.

TABLE 1 THE PROPOSED ALGORITHM SIMULATION RESULTS COMPARED WITH THE STATE-OF-THE ART METHOD

SEQUENCE	***QP***	***METHOD***	***PSNRY***	***PSNRU***	***PSNRV***	***BITRATE***	***TOL. TIME***	***TS [%]***
foreman	22	FS{AVS}	43.14	45.80	48.12	5372.51	66.71	
		PAN [3]	43.01	45.60	48.01	5398.04	48.21	30.7%
		FIPBN [9]	43.08	45.80	48.12	5412.38	41.87	37.24%
		Proposed	43.07	45.80	48,12	5410.21	40.01	40.02%
	30	FS{AVS}	38.63	42.41	45.05	2815.36	55.31	
		PAN [3]	38.60	42.39	45.01	2793.43	41.96	24.13%
		FIPBN [9]	38.59	42.40	45.05	2912.27	39.01	29.41%
		Proposed	38.60	42.38	45.03	2918.31	38.76	29.92%
	38	FS{AVS}	34.90	39.72	41.68	1480.27	50.95	
		PAN [3]	34.88	39.72	41.65	1501.37	44.21	13.21%
		FIPBN [9]	34.75	39.73	41.63	1537.01	40.73	20.06%
		Proposed	34.81	39.65	40.31	1519.12	38.54	24.36%
bus	22	FS{AVS}	42.71	45.48	46.72	8197.48	69.01	
		PAN	42.66	45.47	46.70	8201.63	50.82	26.36%
		FIPBN [9]	42.67	45.48	47.72	8215.51	44.19	35.97%
		Proposed	42.65	45.44	46.70	8210.33	40.18	41.78%
	30	FS{AVS}	37.47	41.86	43.17	5000.18	61.04	
		PAN [3]	37.41	41.85	43.13	5187.13	52.13	14.60%
		FIPBN [9]	37.39	41.86	43.18	5237.21	41.07	32.72%
		Proposed	37.40	41.85	43.14	5209.45	40.29	33.99%
	38	FS{AVS}	32.76	38.87	40.11	2902.70	55.24	
		PAN [3]	32.73	38.85	40.08	2938.84	45.17	18.23%
		FIPBN [9]	32.71	38.87	40.10	3951.69	39.10	29.21%
		Proposed	32.72	38.85	40.09	3200.32	37.33	32.42%
hall	22	FS(AVS)	43.58	45.05	46.01	4595.87	59.09	
		PAN [3]	43.51	45.03	45.96	4627.31	45.21	23.49%
		FIPBN [9]	43.57	45.05	46.00	4608.94	40.34	31.73%
		Proposed	43.52	45.05	45.99	4599.77	38.31	35.17%
	30	FS(AVS)	39.66	41.90	43.24	2382.06	52.96	
		PAN [3]	39.61	41.88	43.21	2493.83	42.98	18.84%
		FIPBN [9]	39.62	41.90	43.24	3481.20	37.75	28.72%
		Proposed	39.61	41.89	43.22	2980.21	35.63	32.72%
	38	FS{AVS}	36.04	39.21	40.72	1406.72	50.16	
		PAN [3]	36.01	39.17	40.70	1473.88	40.13	19.99%
		FIPBN [9]	35.99	39.21	40.71	1497.51	38.27	23.70%
		Proposed	36.00	39.01	40.69	1490.89	31.85	36.50%

IV. CONCLUSIONS

By using the edge information, we enhance the efficiency of intra prediction and improvement the performance of binary block motion estimation. For intra mode decision, we use the SOBEL operator to check the edge information from the total image before its intra prediction. Then we use SVM classifier which is trained by the edge points to decide the mode of block for AVS intra prediction. This achieves up to 16% computational saving with little video quality degradation, compared with results of the existing method.

For binary block motion estimation, we found that the binary blocks which get from 2BT [18] do not take advantage of global edge information. In the 2BTGLE proposed method, we use one global edge threshold and one local edge threshold to get the two bit-planes. The experiments demonstrated that our algorithm gets better compression performance and achieves PSNR gains about 0.13dB. Although the test platform is AVS-P2, the proposed methods can easily be easily modified for HEVC or H.264.

ACKNOWLEDGMENT

This study was supported in part of the technology project of Shenzhen, China. And we would like to thank the anonymous reviewers for their useful comments and valuable suggestions.

REFERENCES

[1] Lu Yu, Feng Yi, Jie Dong, Cixun Zhang, "Overview of AVS-Video: tools, performance and complexity," *SPIE Visual Communications and Image Processing* 5960, pp. 679-690, 2005.

[2] Wen Gao, Siwei Ma, Li Zhang, Li Su and Debin Zhao, "AVS Video Coding Standard", *Intel. Multimedia Communication: Tech. and Application* SCI280, pp. 125-166, 2010.

[3] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. Wu, and S. Wu, "Fast mode decision algorithm for intra-prediction in H.264/AVC video coding," *IEEE Trans. Circuits Syst. Video Technol*., vol. 15, iss. 7, pp. 813-822, Jul. 2005.

[4] J. C. Wang, J. F. Wang, J. F. Yang, and J. T. Chen, "A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction," *IEEE Trans. Circuit Syst. Video Technol*., vol. 17, iss. 10, pp. 1414-1422, Oct. 2007.

[5] M. Parlak, Y. Adibelli and I. Hamzaoglu, "A novel Computational Complexity and Power Reduction Technique for H.264 Intra Prediction," *IEEE Trans. On Consumer Electronics*, vol. 54, iss. 4, pp. 2006-2014, Nov. 2008.

[6] An-Chao Tsai, Jhing-Fa Wang, Jar-Ferr Yang, Wei Guang Lin, "Effective Subblock-Based and Pixel Based Fast Direction Detections for H.264 Intra Prediction," *IEEE Trans. on circuits and systems for video technology*, vol. 18, iss. 7, pp. 975-982, Jul. 2008.

[7] Cheolhong An, and Truong Q. Nguyen, "Statistical learning based intra prediction in H.264," *International Conference on Image Processing*, pp. 2800-2803, 2008.

[8] Yusuf Adibelli, Mustafa Parlak, and Ilker Hamzaoglu, "Pixel Similarity Based Computation and Power Reduction Technique for H.264 Intra Prediction," *IEEE Transactions on Consumer Electronics*, vol. 56, iss. 2, pp. 1079-1087, May 2010.

[9] Yule Yuan, Xiaohang Sun, "Edge Information Based Effective Intra Mode Decision Algorithm," *IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, pp. 628-633, Aug. 2012.

[10] V.N. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, 1998.

[11] Y. W. Huang, ctal, "Analysis and complexity reduction of multiple reference frames motion estimation in h.264/avc," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, iss. 4, pp. 507-522, April 2006.

[12] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," *in Proc. NTC* 81, New Orleans, LA, pp. C9.6.1-9.6.5, 1981.

[13] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Transactions on Communications*, vol. 29, iss. 12, pp. 1799-1808, Dec. 1981.

[14] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. On Circuits and System for Video Technology*, vol. 4, iss. 4, pp. 438-442, Aug. 1994.

[15] L. M. Po and W. C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 313–317, Jun. 1996.

[16] Shan Zhu and Kai-Kuang Ma, "A new diamond search algorithm for fast block matching motion estimation," *IEEE Transactions on Image Processing*, vol. 9, iss. 2, pp. 287-290, Feb. 2000.

[17] B. Natarajan, V. Bhaskaran, and K. Konstantinides, "Low-complexity block-based motion estimation via one-bit transforms," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, iss. 4, pp. 702-706, Aug. 1997.

[18] Erturk A. and Erturk S., "Two bit transform for binary block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, iss. 7, pp. 938-946, July 2005.

[19] Changryoul Choi and Jechang Jeong, "Enhanced two-bit transform based motion estimation via extension of matching criterion", *IEEE Trans. on Consumer Electronics*, vol. 56, iss. 3, pp. 1883-1889, Aug. 2010.

[20] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, iss.7, pp. 560-576, July 2003.

[21] VS Video Expert Group, Information Technology - Advanced Audio Video Coding Standard Part 2: Video, in Audio Video Coding Standard Group of China, Doe, AVS-N1063, Dec. 2003.

[22] L. S. Davis, "A survey of edge detection techniques," *Computer Graphics Image Processing*, vol. 4, pp. 248-270, 1975.

[23] F. Pan, X. Lin, S. Rahardja, K.P. Lim, and Z.G. Li, "A Directional Field Based Fast Intra Mode Decision Algorithm For H.264 Video Coding," *Proc. of IEEE ICME*, vol. 2, pp. 1147-1150, 2004.

[24] Otsu N, "A threshold selection method from gray-level histogram," *IEEE Trans, SMC*, vol. 9, iss. 1, pp. 62-66, 1979.

Yule Yuan was born in Hubei, China, in 1978. He graduated in computer science from Three Gorges University, Yichang, China, in 2000. He received the M.E. degree in software engineering from Peking University, Beijing, China, in 2008. He is currently a candidate for a Ph.D. degree in electrical and computer engineering at Peking University. From 2001 to 2004, he was a senior software engineer at e-Future Information Technology Inc. From 2008 to 2009, he was an engineer at C2 Microsystems, Beijing, China. From 2009 to 2011, he works as a research engineer at Shenzhen Graduate School of Peking University, Shenzhen, China. His research interests and areas of publications include video codec, image processing and machine learning.

Jie Kong was born was born in 1987. He received the B.E. degree in electronic science and technology from Wuhan University, in 2008, Wuhan, China. He received the M.E. degree in microelectronics from Peking University, in 2012, Beijing China. He is currently an engineer at Samsung Guangzhou R&D mobile center. His research interests and areas of publications include signal processing.

Yong Zhao was born was born in 1963. He received the B.E. degree in mathematics from Guizhou University, Guiyang, China, in 1985, and the M.E. degree from Northwestern Polytechnic University, Xi'an, China, in 1988, and the PHD degree in Engineering from Research Institute of Automation, Southeast University, Nanjing, China, in 1991. From 1991 to 1994, he worked as a research assistant in Dept. of

Biomedical Engineering, Zhejiang University. From 1994 to 1997, he worked as a vice dean/associate professor in Dept. of Electrical Engineering, Hanzhou University (Now the Zhejiang University). From 1997 to 2000, he worked as a post-doctoral fellow in Center for Signal Processing and Communications, Electrical and Computer Engineering, Concordia University, Montreal, Canada. From 2000 to 2004, he worked as a senior software engineer in Honeywell Corporation, Ottawa, Canada. Since 2004, he has been with Shenzhen Graduate School of Peking University, Shenzhen, China, where he is currently an associate professor. His research interests and areas of publications include signal processing, video codec and machine learning.

Xinan Wang was born in Henan, China, in 1963. He received the B.S. degree in Computer Science from Wuhan University, Wuhan, China, in 1983, and the M.S. and Ph.D. degrees in microelectronics from Shanxi Microelectronics Institute, Xi'an, China, in 1989 and 1992, respectively. He is currently a Professor in the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He is now working in the school of electronic and computer engineering, Peking University Shenzhen Campus. His research interests focus on the area of application-specified integrated-circuit design and IC design methodology.