A FAST MODE DECISION ALGORITHM FOR H.264/AVC INTRA PREDICTION

Ling-Jiao Pan and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong Buk-gu, Gwangju, 500-712, Korea
{hedypan, hoyo}@gist.ac.kr

ABSTRACT

In this paper, we present a fast mode decision algorithm for H.264/AVC intra prediction. The goal of this work is to reduce computational complexity of the H.264 encoder without significant rate-distortion degradation. For luminance and chrominance mode decision, we design two separate optimization methods. In order to select the candidate modes for Intra4x4 and Intra16x16 prediction efficiently, we use spatial correlation and simple directional information. Additionally, we apply an early block size selection method to further reduce the searching time. Experimental results show that the proposed algorithm can save the entire encoding time by 82% on average while introducing negligible loss in PSNR values and small increment of bit rates.

***Index Terms*—**H.264, video coding, intra prediction

1. INTRODUCTION

The latest H.264 video coding standard can greatly outperform other existing coding standards in both PSNR and visual quality. This efficiency is achieved by using several powerful coding approaches [1]. One important approach is the intra mode prediction which exploits the directional spatial correlation to reduce the spatial redundancy [2]. The problem arises when choosing the best mode from all the intra modes. In order to take the full advantage of all the intra modes, H.264 provides a rate distortion optimization (RDO) technique to select the best mode [3]. In this technique, by searching all the mode combinations for each MB exhaustively, we can achieve the best coding quality while minimizing the bit rate. However, the RDO technique increases complexity and computation load drastically. This makes H.264 unsuitable for real-time applications. Thus a fast mode decision method is required to reduce the encoding time.

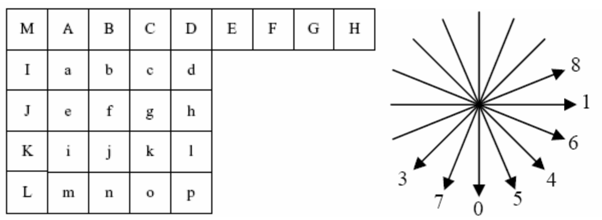
Until now, several efforts have been made to reduce the complexity of H.264. Pan et al. [4] proposed a fast intra mode decision algorithm based on pre-procession of the edge direction information. This scheme reduces intra prediction modes using the dominating edge direction. However, it needs additional operations in calculating the edge direction information. Another approach is based on the idea of reducing possible candidate directions [5]. Several possible candidates are chosen and their RD costs are computed and compared. In this method, RD costs of 6 to 7 out of 9 modes need to be computed in Intra4x4 prediction. This approach does not select the best candidate mode efficiently and some unnecessary modes are considered all the time.

In this paper, we propose a simple yet effective mode decision algorithm for H.264 intra prediction. Fast algorithms are separately designed for luma and chroma mode decisions. Moreover, the early block type selection approach is jointly used to reduce the complexity. It is observed that pixels along the direction of the local edge are normally of the similar value. This provides a clue for obtaining direction information from the pixel values along the direction. Based on the direction information, we can pre-predict the best candidate mode for the current block. Furthermore, considering the spatial correlation, we can check the reliability of the best candidate mode and select few final candidate modes efficiently. This approach not only avoids the time consuming calculations for getting the direction histogram but also selects a small numbers of good candidate modes efficiently. Precisely speaking, we only check one mode in the best case, and no more than 4 candidate modes in the worst case for the Intra4x4 prediction. Experimental results show that the fast intra mode decision algorithm increases the speed of coding significantly.

2. INTRA MODE DECISION FOR H.264

In this section, we briefly review the H.264/AVC intra modes as well as the main RDO procedure, utilized to choose the best intra mode. H.264 uses three different types of intra prediction for the luminance component Y. They are Intra4x4 (I4_MB), Intra8x8 (I8_MB, only used for High profile) and Intra16x16 (I16_MB). There are total of nine optional prediction modes for each I4_MB and I8_MB, four modes for I16_MB, and four modes for the chroma components. In this paper, we only consider I4_MB and I16_MB.

For I4_MB, the prediction unit is a block of 4x4 pixels, as shown in Fig. 1. The samples above and to the left (labeled as A-M in Fig. 1) have previously been coded and reconstructed; therefore, they are available both at the encoder and decoder to form a prediction reference. The pixels in the prediction unit are calculated based on the samples A-M using one of the nine prediction modes. Fig. 1 shows the eight specific prediction directions for each mode. Mode 2 (DC mode) is a non-directional mode. All pixels are predicted by the mean of the samples A-M. For modes 3-8, each sample is predicted by a weighted sum of the prediction samples A-M.



(a) (b)

Fig. 1 (a) 4x4 block and neighboring pixels (b) eight prediction directions for intra 4x4 prediction

For I16_MB, only four prediction modes are applied to the whole macroblock, including vertical prediction, horizontal prediction, DC prediction and plane prediction. Here, the plane prediction uses a linear function between the neighboring samples to the left and to the top to predict the current samples. Plane prediction works well in areas of smoothly varying luminance. The other prediction modes are the same as I4_MB; the only difference is that they are applied to the whole macroblock instead of the 4x4 unit. The four chroma prediction modes are very similar to that of the I16_MB prediction except that the order of the mode is different.

In order to choose the best mode, H.264 uses the RDO method, where a mode with the smallest rate-distortion cost is chosen as the best mode. The RDO method is based on the Lagrangian function that considers both rate and distortion. The RD cost for the best intra mode is decided by Eq. (1):

$$J(s,c,MODE\ /QP) = SSD(s,c,MODE\ /QP) + \lambda_{MODE} \bullet R(s,c,MODE\ /QP) \quad (1)$$

where *SSD* represents the sum of squared differences between the original block s and the reconstructed block c; λ_{MODE} is the Lagrange multiplier, calculated as a function of the quantization parameter. The prediction residual is transformed, quantized and then entropy encoded to calculate the rate *R*.

According to the exhaustive procedure, all the possible mode combinations for the luma and chroma blocks in a macroblock need to be checked. The one with the minimum *J* value is then chosen as the best mode. This procedure can find the best result, but its computational load is very high.

3. FAST INTRA MODE DECISION ALGORITHM

In the proposed algorithm, we design fast algorithms for chroma and luma intra prediction separately. Furthermore, an early termination of the block type decision approach is applied to reduce the computational complexity. Fig. 2 shows the flowchart of the proposed procedure for intra mode decision.

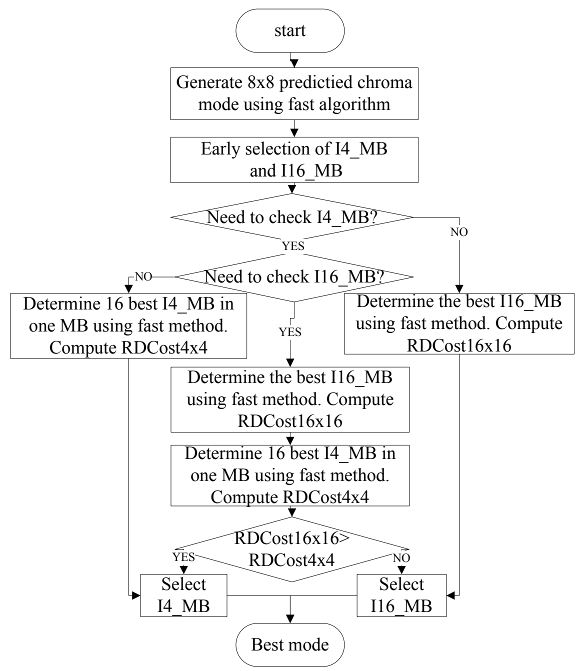


Fig. 2 Flowchart for the fast intra mode decision algorithm

The procedure can be summarized as:

Step 1: Select an intra predicted chroma mode by the fast method described in section 3.1.

Step 2: Decide the best block size according to the method detailed in section 3.2. If no necessary to check Intra16x16, go to Step 4.

Step 3: Determine the best intra mode for I16_MB among few candidate modes which are selected by the method described in section 3.3. Then code chroma components with the given mode and calculate the rate distortion RDCost16x16 for both the luma and chroma components. If no necessary to check Intra4x4, go to Step 6.

Step 4: Determine the best intra mode for I4_MB among few candidate modes which are selected by the method described in section 3.3. Repeat this process for the sixteen 4x4 blocks in one MB. Then, code the chroma components with the given modes and calculate RDCost4x4 for both luma and chroma components. If not necessary to check Intra16x16, go to Step 6.

Step 5: Compare RDCost16x16 with RDCost4x4 and select the best block type with the minimum RDCost.

Step 6: Save the best mode for the current MB and repeat all the process for the next MB.

3.1. Intra mode decision for 8x8 chroma blocks

Since the choice of prediction mode for the chroma component is independent of the luma component, we can optimize chroma and luma components separately. Since ultimately, the transformed coefficients are coded, we can achieve a better estimation for the mode cost by using the Hadamard transform instead of the DCT transform. The performance of SATD (sum of absolute Hadamard transform differences) is close to the Lagrangian function while the computational load is much lower [6]. In this work, we determine the best chroma mode by choosing the mode results in the minimum SATD value. The following mode decision processes are then performed with the best chroma mode.

3.2. Early block type selection

It is observed that the block size mainly depends on the smoothness of a region. Large block size tends to be used in homogeneous regions and the small block size works well for complex textures. The idea behind our approach is that the smooth filter does not affect the homogeneous regions but will blur the detail information in the complex regions. In our approach we apply a 1x5 and a 5x1 mean value filter to the top and left boundary of each macroblock separately. The flitted pixel values can be obtained by Eq. (2) and Eq. (3) respectively.

$$p_{ij} = \frac{1}{5}\sum_{k=j-2}^{j+2} p_{ik} \tag{2}$$

$$p_{ij} = \frac{1}{5}\sum_{k=i-2}^{i+2} p_{kj} \tag{3}$$

Then we calculate SAD between the original pixel values and the flitted pixel values (SADOF). Two thresholds Th1 (bottom threshold) and Th2 (up threshold) are applied. If SADOF<Th1, the 16x16 intra prediction is further explored. If SADOF>Th2, the 4x4 intra prediction is adopted for the following mode decision. If SADOF locates between the two thresholds, both block sizes need to be checked. Since for higher QP values, large block sizes are preferred, the thresholds should vary with QP to reflect the quantization effect. Linear equations of QP (Eq. (4)) are found to give good performance. a_1, b_1, a_2, b_2 are decided by exhaustive experiments.

$$\begin{aligned} Th1 &= a_1 * QP + b_1 \\ Th2 &= a_2 * QP + b_2 \end{aligned} \tag{4}$$

3.3. Intra mode decision for 4x4 and 16x16 luma blocks

For the Intra4x4 prediction, H.264 supports 9 prediction modes. According to the prediction process of the reference software, all the predicted pixels along the same prediction direction should have the same value. For example, in Mode 0 (vertical mode), the predicted pixel values in position a, e, i, m should all equal to the pixel value in position A (see Fig.1). If Mode 0 is selected as the best mode, predicted pixel values along the vertical direction equal to one another. This also indicates in the original block, the pixel values in these positions are very similar to one another. Therefore, we can roughly predict the best candidate mode by checking the NSAD (normalized sum of absolute differences) for some selected pixel positions in the original block. Firstly, we select some pixels along the prediction directions and calculate the NSAD for each mode using the equations in Table 1. Note that in these equations, “a” to “p” indicate the pixel values in the original block.

Table 1: NSAD for each intra prediction direction

Mode	Direction	NSAD
0	vertical	(|a-m|+|b-n|+|c-o|+|d-p|)/4
1	horizontal	(|a-d|+|e-h|+|i-l|+|m-p|)/4
3	diagonal down-left	(|b-e|+|d-m|+|l-o|)/3
4	diagonal down-right	(|a-p|+|i-n|+|c-h|)/3
5	vertical- right	(|a-j|+|b-k|+|c-l|)/3
6	horizontal- down	(|a-g|+|e-k|+|i-o|)/3
7	vertical-left	(|b-i|+|c-j|+|d-k|)/3
8	horizontal-up	(|e-c|+|i-g|+|m-k|)/3

Since Mode2 (DC mode) has no direction and is predicted by the mean of sample A-M, we apply Eq. (5) to deal with DC mode.

$$DDC = \sum_{i=0}^{3}\sum_{j=0}^{3} \left| m - p(x+i, y+j) \right| \tag{5}$$

In Eq. (5), m is the mean pixel value of sample A-M. If DDC (difference of DC mode) is less than a threshold, DC mode is selected as the best candidate mode for the current block and the mode with the smallest NSAD among the other eight modes is chosen as the second best mode. Otherwise, the modes with the smallest NSAD and the second smallest NSAD are selected as the best candidate mode and the second best candidate mode separately. Denote the best candidate mode as mode C and the second best candidate mode as mode S.

Using NSAD and DDC we can roughly predict the best mode. Considering the spatial correlation information can help to evaluate the reliability of the best candidate further. Observations show that the best mode of the current block is highly correlated to its neighboring blocks. The most probable mode can be obtained from left and above blocks. Fig. 3 shows the neighboring modes of current block.

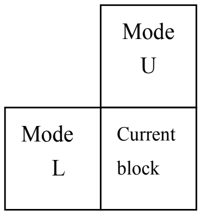


Fig. 3 **N**eighboring blocks of the current block

Through experiments on various video sequences with different textures, we find the average probability of mode U=L=current mode is 40.62% and current mode=L or current mode=U is 80.86%. When U=L, the conditional probability of current mode=U=L is up to 87.5%. That means when U=L, current mode has very high probability to choose the same mode as U and L. Therefore, mode U and L can be used to check the reliability of pre-predicted best candidate mode C. According to the reliability we decide the number of candidate modes. If it is reliable we consider only the pre-predicted best mode C, otherwise we need to consider both pre-predicted candidate mode (C, S) and neighboring modes.

Decisions for final candidate modes are summarized in Table 2.

Table 2: Candidate mode decision table

Condition	Reliability of mode C	Candidate modes
U=L=C	reliable	C
C=U&&C! =L	unreliable	C, S, L
C=L&&C! =U	unreliable	C, S, U
L=U&&C! =L	unreliable	C, S, L
C=!U&&C!=L&& L!=U	totally unreliable	C, S, L, U
U not available&& C=L	unreliable	C, S
U not available&& C! =L	totally unreliable	C, S, L
L not available&& C=U	unreliable	C, S
L not available&& C!=U	totally unreliable	C, S,U

The same idea is applied to the Intra16x16 luma block except the different block size and the plane mode prediction. The plane prediction estimates a bilinear function from the neighboring pixels to the 16x16 block. It is not mathematically correct to associate the plane prediction to any directional edge. Based on the plane prediction method used in the reference software [7], we use Eq. (6) to calculate the NSAD for plane prediction. In the equation, Org (A_{diff}) and Org (B_{diff}) indicate the SAD of original pixel values whose positions are pointed out by arrow A and arrow B (Fig. 4) separately. Est(A_{diff}) and Est(B_{diff}) indicate the SAD of estimated pixel values whose positions are pointed out by arrow A and arrow B (Fig. 4) separately.

$$\begin{aligned} NSAD &= (A+B)/2 \\ A &= (Org\,(A_{diff}) - Est\,(A_{diff}))/7 \\ B &= (Org\,(B_{diff}) - Est\,(B_{diff}))/7 \end{aligned} \tag{6}$$

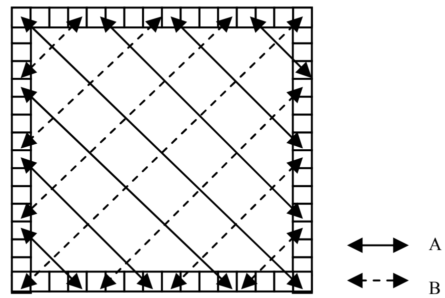


Fig. 4 NSAD for plane mode

4. SIMULATION RESULTS

The proposed algorithm is implemented on JM 11.0. We have tested six CIF (352x288) video sequences (Foreman, Bus, Coastguard, Mobile, City and Crew). For each sequence 100 frames are encoded with I-frame only. The frame rate is 30 fps. CABAC is adopted as the entropy coding method. The Hadamard transform is enabled. Experiments were conducted for four quantization parameters: QP=28, 32, 36, and 40. The simulations were implemented using Intel Pentium Ⅳ 2.8G PC.

For performance comparison, we compared the luma PSNR and chroma PSNR, the bit rate and the total time required for encoding. Table 3 shows the simulation results. In the table, positive number means increasing, and the negative number means decreasing. All are relative to results by the reference software.

$$\Delta \text{PSNR\_Y} = \text{PSNR\_Y(proposed)} - \text{PSNR\_Y(reference)}\ [\text{dB}]$$
$$\Delta \text{PSNR\_UV} = \text{PSNR\_UV(proposed)} - \text{PSNR\_UV(reference)}\ [\text{dB}]$$
$$\Delta \text{Bits} = \frac{Bits(proposed) - Bits(reference)}{Bit(reference)} \times 100[\%]$$
$$\Delta Time = \frac{Time(proposed) - Time(reference)}{Time(reference)} \times 100[\%] \tag{7}$$

The simulation results show that the encoding time in the proposed algorithm is obviously less than JM11.0. The proposed fast intra prediction algorithm can achieve 82% time saving on average with negligible loss in PSNR and increment in bit rates. Fig. 5 to Fig. 8 show the RD curve of several test sequences. The RD performance of the proposed algorithm is almost the same as JM. Since in H.264 standard luma and chroma parts are overally optimized whereas in our algorithm chroma part is independently optimized, we can find PSNR-UV have positive deltas in some occasions. Additionally, comparing with other algorithms ([4], [5]), the proposed method also consistently outperforms them with about 20%~30% encoding time saving while keeping the similar PSNR and bit rates. Table 4 outlines the comparison results with some other algorithms. The results in Table 4 are the average results of QP=28, 32, 36, 40.

Table 3: Performance comparison with JM11.0

QP	Sequence	△PSNR_Y (dB)	△PSNR_UV (dB)	△Bits (%)	△Time (%)
28	Foreman	-0.04	0	2.99	-82.11
	Bus	-0.13	-0.025	1.68	-83.46
	Coastguard	-0.11	0.095	0.46	-82.96
	Mobile	-0.19	-0.04	2.05	-84.31
	City	-0.09	-0.015	2.45	-81.92
	Crew	-0.05	-0.035	3.16	-81.56
32	Foreman	-0.05	0.09	3.39	-81.73
	Bus	-0.09	0.055	2.13	-82.75
	Coastguard	-0.06	0.215	0.97	-82.05
	Mobile	-0.17	-0.01	2.41	-83.61
	City	-0.06	0.125	3.20	-81.09
	Crew	-0.04	0.045	2.91	-81.36
36	Foreman	-0.02	0.22	4.03	-81.63
	Bus	-0.06	0.16	3.17	-82.07
	Coastguard	-0.03	0.24	2.16	-81.27
	Mobile	-0.13	0.005	3.05	-82.96
	City	-0.02	0.375	3.90	-80.39
	Crew	-0.01	0.13	3.30	-81.49
40	Foreman	-0.02	0.405	4.17	-81.69
	Bus	-0.06	0.235	4.74	-81.56
	Coastguard	-0.02	0.645	4.63	-80.55
	Mobile	-0.08	0.07	4.03	-82.37
	City	-0.01	0.555	4.90	-80.24
	Crew	0	0.28	4.30	-81.87

Table 4: Performance comparison with different algorithms

Sequence	Method	△PSNR_Y (dB)	△Bits (%)	△Time (%)
Foreman	[4]	-0.285	+4.437	-65.378
	[5]	-0.008	+3.483	-58.27
	Ours	-0.0325	+3.645	-81.79
Coastguard	[4]	-0.106	+2.361	-55.026
	[5]	-0.018	+1.343	-56.07
	Ours	-0.055	+2.055	-81.71
Mobile	[4]	-0.255	+3.168	-59.086
	[5]	-0.049	+1.107	-49.7
	Ours	-0.1425	+2.885	-83.31
Bus	[4]	-0.218	+3.849	-58.118
	[5]	/	/	/
	Ours	-0.085	+2.93	-82.46

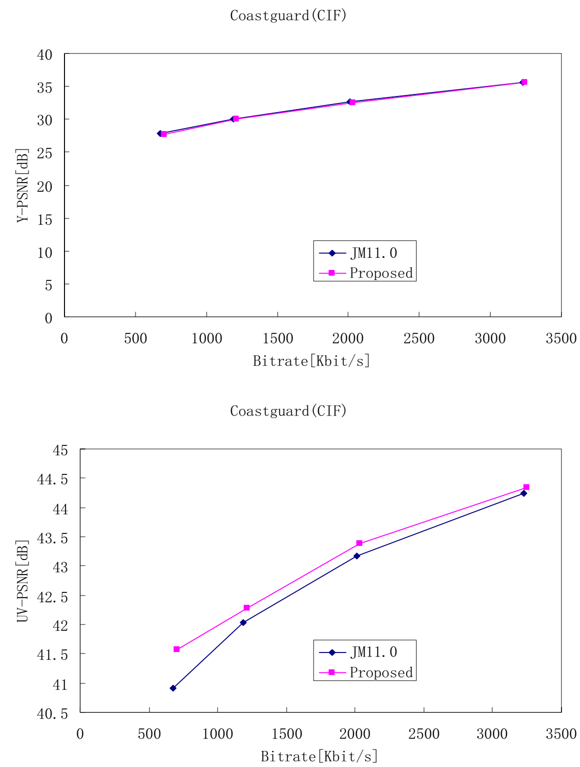


Fig. 5 RD performance of Coastguard (CIF)

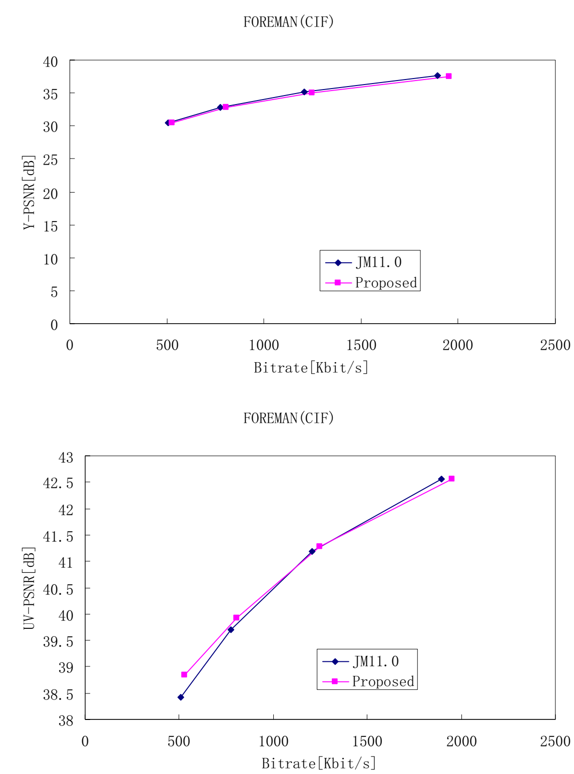


Fig. 6 RD performance of Foreman (CIF)

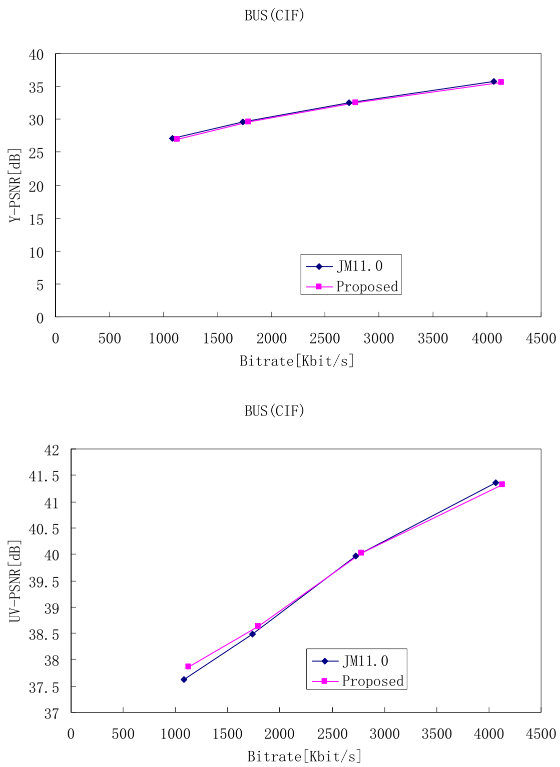


Fig. 7 RD performance of Bus (CIF)

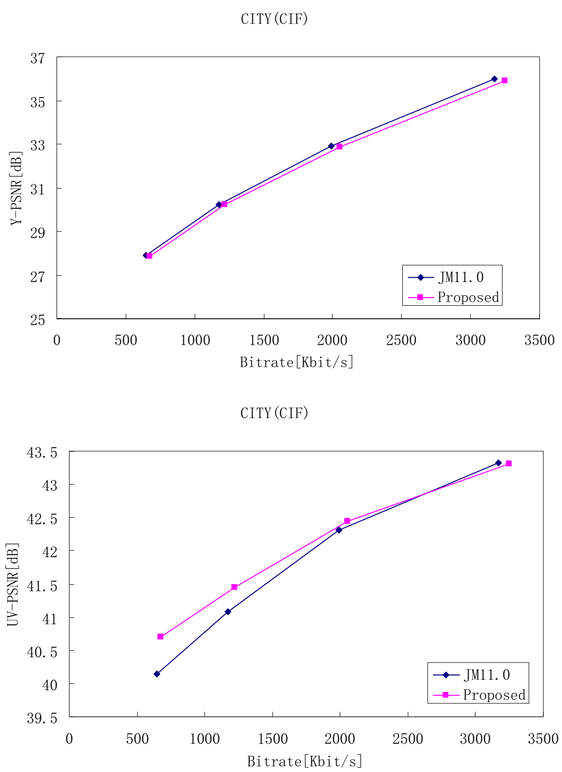


Fig. 8 RD performance of City (CIF)

5. CONCLUSION

In this paper, we proposed a fast mode decision algorithm for H.264 intra prediction. We have used the spatial correlation information and simple direction information to reduce the candidate modes for Intra4x4 and Intra16x16 prediction. We also considered an early block size selection method which based on smooth filters to further reduce the computational complexity. Experimental results show that our method can achieve 82% encoding time reduction with only 0.06dB loss in PSNR value and 3% increment in bit rates.

ACKNOWLEDGMENTS

This work was supported in part by the Institute for Information Technology Advancement (IITA), in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21(BK21) project.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits and Systems for Video Technology*, vol.13, no.7, pp. 560-576, July 2003.

[2] T. Halbach, “Performance comparision: H.26L intra coding vs. JPEG2000.” ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT 4th Meeting, Klagenfurt, Austria, July 2002.

[3] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol.15, pp. 74-90, Nov. 2003.

[4] F. Pan, X. Lin, S. Rahardja, K.P. Lim, Z. G. Li, D. Wu, and S. Wu, “Fast mode decision algorithm for intraprediction in H.264/AVC video coding,” *IEEE Trans. Circuits and Systems for Video Technology*, vol.15, no.7, pp.813-822, July 2005.

[5] S. H. Ri, J. Ostermann, "Intra-Prediction Mode Decision for H.264 in Two Steps," *Picture Coding Symposium 2006*, April 2006.

[6] K. P. Lim, G. J. Sullivan, and T. Wiegand, “Text description of joint model reference encoding methods and decoding concealment methods,” JVT-N046, JVT of ISO/IEC MPEG and ITU-T VCEG, Jan. 2005.

[7] JVT H.264/AVC Reference Software version JM11.0, http://iphome.hhi.de/suehring/tml/download/.