Fast 4×4 Intra-prediction Mode Selection for H.264

Zhang Yong-dong Dai Feng Lin Shou-xun
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, 100080, P.R.China
E-mail:{ zhyd, fdai, sxlin}@ict.ac.cn

Abstract

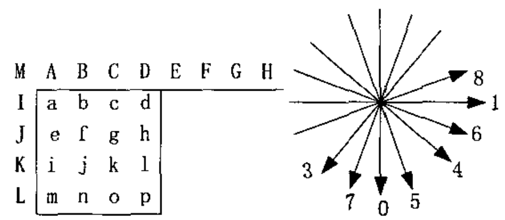
In this paper, we propose a fast intra-prediction mode selection method for JVT video coding standard H.264. Based on local edge information obtained by calculating edge feature parameters, the method can reduce the computational complexity considerably while maintaining similar PSNR and bit rate. Experimental results are presented to show the effectiveness of the proposed method.

1. Introduction

In the upcoming video coding standard H.264 [1], in order to improve the efficiency of video coding, intra-prediction is used to remove the correlation among the neighboring 4x4 blocks. The block types for intra-prediction(luminance) can be 4×4 and 16×16. The 4×4 block intra-prediction is shown in Figure 1. A 4×4 block contains 16 pixels labeled from *a* to *p*. A prediction mode is a way to generate 16 predictive pixel values using some or all of the neighboring pixels *A* to *M*. There are 9 different intra-prediction modes designed in a directional manner. Mode 2 is called DC prediction in which all pixels (*a* to *p*) are predicted by *(A+B+C+D+I+J+K+L)/8*. The other 8 modes are show in Figure 1(b). Mode 0 is vertical prediction mode, and Mode 1 is horizontal prediction mode.

In the Reference software [2] from JVT (Joint Video Team), a full search (FS) is used to examine all the 9 4x4 intra prediction modes to find the best one with the smallest cost. The main steps are:

1. Generate a 4×4 predicted block according to a mode *i*;
2. Calculate sum of absolute difference (SAD_i) between the original 4x4 block and the predicted block;
3. Compute $COST_i = SAD_i + 4\lambda(Q_P)R$, where $\lambda(Q_P)$ is an approximately exponential function of the quantization factor Q_P, R equals 0 for the most probable mode, and 1 for other modes.
4. Repeat 1 to 3 for all the 9 modes, and choose the one that has the minimum cost.



(a) (b)

Figure 1. 4×4 intra-prediction mode (a) A 4×4 block and its neighboring pixels (b) 4×4 intra-prediction mode direction

Although full search can achieve optimal prediction mode selection, it is computationally expensive. In fact, it is a bottleneck in I-frame coding. It is thus highly desirable to develop fast intra-prediction mode selection. A fast 4x4 intra prediction mode selection, which employs a partial computation of the cost function and early termination, is proposed in [3]. Pan *et al.* [4] propose a fast intra-prediction mode selection method based on edge direction histogram. This method reduced the number of possible modes to achieve fast intra-prediction. But the computation of edge direction introduces a lot of additional complexity, which is not taken into account in the article.

In this paper, we propose a new fast 4x4 intra-prediction mode selection method, which is able to reduce the computational complexity considerably while maintaining similar PSNR and bit rate. This method is based on some observations in our experiments. Firstly, we observe that the best prediction mode of a block is most likely in the direction of local edge within that block. Secondly, we observe that DC prediction mode has higher possibility to be the best prediction mode than other 8 modes.

In the next section, we explain how to obtain the local edge direction within a 4×4 block. After that, in Section 3, we present the fast intra-prediction mode selection method. Experimental results showing the effectiveness of the method are reported in Section 4.

2. Extraction of Local Edge Information

First, we divide a 4×4 block into four 2×2 blocks as shown in Figure 2. Let A, B, C and D respectively denotes the sum of intensity of all pixels in the corresponding 2×2 blocks (Equation (1)). $P(x, y)(0 \le x \le 3, 0 \le y \le 3)$ denotes the intensity of the pixel at position (x, y) of the current 4×4 block.

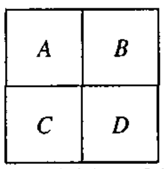


Figure 2. Division of 4×4 block

$$A=\sum_{y=0}^{1}\sum_{x=0}^{1}P(x,y) \quad B=\sum_{y=0}^{1}\sum_{x=2}^{3}P(x,y)$$
$$C=\sum_{y=2}^{3}\sum_{x=0}^{1}P(x,y) \quad D=\sum_{y=2}^{3}\sum_{x=2}^{3}P(x,y) \tag{1}$$

Next, in order to obtain the local edge direction within a 4×4 block, we introduce two edge feature parameters: vertical edge parameter F_v and horizontal edge parameter F_h,

$$F_v=\left\lfloor \frac{(A+C)-(B+D)}{S} \right\rfloor \tag{2}$$

$$F_h=\left\lfloor \frac{(A+B)-(C+D)}{S} \right\rfloor \tag{3}$$

Where S is a scaling factor, $\lfloor\ \rfloor$ represents the floor function. The physical meanings of the two parameters are shown pictorially in Figure 3. F_v and F_h represent intensity differences between the left and right parts and between the upper and lower parts of the block respectively. According to the two parameters, we can obtain the edge direction information within the current 4×4 block [5], which is showed in Table 1.

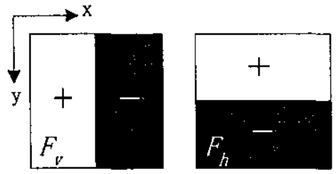
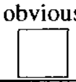
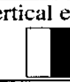
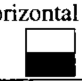


Figure 3. Physical meanings of F_v and F_h

3. Fast 4×4 Intra-Prediction Mode Selection

The best way to achieve fast mode selection is to reduce the number of candidates for the best prediction modes (CBPMs) of the current 4×4 block. The number of CBPMs is 9 in FS. Since the best prediction mode of a block is most likely in the direction of local edge within that block, the fast method only chooses the prediction mode along the edge direction as CBPM. Since DC prediction has higher possibility to be the best prediction mode than other 8 modes, DC prediction is also chosen as CBPM.

Table 1. Edge direction information according to F_v and F_h

Case	The relationship between F_v and F_h	Edge direction
1	$\lvert F_v\rvert = \lvert F_h\rvert = 0$	No obvious edge 
2	$\lvert F_h\rvert = 0$ and $\lvert F_v\rvert > 0$	Vertical edge 
3	$\lvert F_v\rvert = 0$ and $\lvert F_h\rvert > 0$	Horizontal edge 
4	$\lvert F_v\rvert = \lvert F_h\rvert > 0$ and $F_v / F_h > 0$	Diagonal down/left edge 
5	$\lvert F_v\rvert = \lvert F_h\rvert > 0$ and $F_v / F_h < 0$	Diagonal down/right edge 
6	$\lvert F_v\rvert > \lvert F_h\rvert > 0$	Vertical-dominant edge
7	$\lvert F_h\rvert > \lvert F_v\rvert > 0$	Horizontal-dominant edge

According to F_v and F_h, we can determine different CBPMs for the current block corresponding to the 7 cases in Table 1, which is summarized in Table 2. For case 6, because there is vertical-dominant edge, prediction mode 1, 6 and 8, which belong to horizontal-dominant direction, can not be chosen as CBPMs. For case 7, similarly, prediction mode 0, 5 and 7 can not be chosen as CBPMs.

Table 2. CBPMs according to F_v and F_h

Case	CBPMs
1	2
2	0, 2
3	1, 2
4	3, 2
5	4, 2
6	0, 5, 7,3, 4, 2
7	1, 6, 8 3, 4, 2

Here is the proposed fast intra-prediction mode selection method (FMS). The predicted pixels according to a mode should be generated only when they are to be checked.

1. Calculate F_v and F_h
   The scaling factor S is determined according to the quantization factor Q_P, thus F_v and F_h could be obtained by additions and shifts
   $$S = \begin{cases} 8 & Q_P < 20 \\ 16 & 20 \le Q_P < 30 \\ 32 & 30 \le Q_P < 40 \\ 64 & Q_P \ge 40 \end{cases} \quad (4)$$
2. Determine CBPMs for the current block according to F_v and F_h;
3. Generate a 4×4 predicted block according to a mode from CBPMs;
4. Compute the cost of the mode;
5. Repeat 3 to 4 for all the modes in CBPMs, and choose the one with the minimum cost.

4. Experimental Results

The proposed method FMS was implemented into JM7.2 provided by JVT. FMS and FS are simulated on four QCIF sequences and two CIF sequences. For each sequence, 50 frames are encoded with I-frame coding only, and with fixed quantization factors Q_P. Various Q_P factors are tested. The complexity is defined as the average search number of prediction modes (per frame). Although the computation load of obtaining F_v and F_h in FMS is less than half of that of one search, we regard the computation load as half of one search in our experiments.

4.1. Experiments on QCIF Test Sequences

Tables 3-6 show the experiment results on four QCIF test sequences, "News", "Silent", "Container" and "Coastguard".

Table 3 Result of News, QCIF

Q_P	Method	Complexity	PSNR(dB)	Bits
5	FS	13815	56.03	7347.5K
	FMS	6879	56.03	7428.3K
	Saved	-50.2%	0.00	+1.10%
16	FS	13815	46.94	3328.0K
	FMS	6879	46.93	3401.8K
	Saved	-50.2%	-0.01	+2.22%
32	FS	13815	34.36	973.6K
	FMS	5113.2	34.32	1024.7K
	Saved	-63.0%	-0.04	+5.25%
45	FS	13815	25.26	258.5K
	FMS	4576	25.23	267.4K
	Saved	-66.9%	-0.03	+3.42%

Table 4 Result of Silent, QCIF

Q_P	Method	Complexity	PSNR(dB)	Bits
5	FS	13815	55.91	7347.5K
	FMS	8080	55.91	7428.3K
	Saved	-41.5%	0.00	+1.10%
16	FS	13815	46.19	3841.7K
	FMS	8079	46.18	3905.5K
	Saved	-41.5%	-0.01	+1.66%
32	FS	13815	33.49	843.6K
	FMS	5690	33.44	887.3K
	Saved	-58.8%	-0.05	+5.19%
45	FS	13815	25.81	159.7K
	FMS	4665	25.79	160.4K
	Saved	-66.2%	-0.02	+0.45%

Table 5 Result of Container, QCIF

Q_P	Method	Complexity	PSNR(dB)	Bits
5	FS	13815	55.94	6676.5K
	FMS	6547	55.94	6755.2K
	Saved	-52.6%	0.00	+1.18%
16	FS	13815	46.34	3333.1K
	FMS	6547	46.32	3380.4K
	Saved	-52.6%	-0.02	+1.42%
32	FS	13815	34.21	835.2K
	FMS	4980	34.19	876.7K
	Saved	-64.0%	-0.02	+4.98%
45	FS	13815	25.70	206.5K
	FMS	4481	25.65	212.5K
	Saved	-67.6%	-0.05	+2.92%

Table 6 Result of Coastguard, QCIF

Q_P	Method	Complexity	PSNR(dB)	Bits
5	FS	13815	55.76	7363.0K
	FMS	8114	55.77	7457.3K
	Saved	-41.3%	+0.01	+1.28%
16	FS	13815	45.70	4011.1K
	FMS	8114.0	45.69	4085.0K
	Saved	-41.3%	-0.01	+1.84%
32	FS	13815	32.37	869.7K
	FMS	5233	32.35	900.6K
	Saved	-62.1%	-0.02	+3.56%
45	FS	13815	25.59	130.6K
	FMS	4321	25.58	134.1K
	Saved	-68.7%	-0.01	+2.75%

4.2. Experiments on CIF Test Sequences

Tables 7-8 show the experiment results on two CIF test sequences, "Flower" and "Funfair".

Table 7 Result of Flower, CIF

Q_P	Method	Complexity	PSNR (dB)	Bits
5	FS	56139	56.11	30568K
	FMS	27841	56.11	30831K
	Saved	-50.4%	0.00	+0.86%
16	FS	56139	47.17	17611K
	FMS	27841	47.16	17803K
	Saved	-50.4%	-0.01	+1.09%
32	FS	56139	32.67	6251K
	FMS	23696	32.65	6391K
	Saved	-57.8%	-0.02	+2.23%
45	FS	56139	22.75	1439K
	FMS	20763	22.70	1447K
	Saved	-63.0%	-0.05	+0.51%

Table 8 Result of Funfair, CIF

Q_P	Method	Complexity	PSNR (dB)	Bits
5	FS	56139	55.83	33312K
	FMS	33499	55.83	33692K
	Saved	-40.3%	0.00	+1.14%
16	FS	56139	45.93	18183K
	FMS	33499	45.92	18493K
	Saved	-40.3%	-0.01	+1.71%
32	FS	56139	32.67	4768K
	FMS	24323	32.62	5018K
	Saved	-56.7%	-0.05	+5.25%
45	FS	56139	24.20	1039K
	FMS	19804	24.15	1078K
	Saved	-64.7%	-0.05	+3.67%

4.3. Summary of Experimental Result

From the results above, we can see that FMS has outstanding reduction in computational complexity of intra-prediction mode selection (saving computational load between 40% and 70%) with less than 5.5% extra bits used and not more than 0.05dB PSNR sacrificed. The computational complexity of FMS decreases with the increase of quantization factor Q_P.

5. Conclusion

In this paper, we propose a fast 4×4 intra-prediction mode selection method for JVT video coding standard H.264. Based on local edge information, the method can reduce computational complexity by reducing the number of CBPMs. Simulation results suggest that the method can achieve considerable computation reduction while maintaining similar PSNR and bit rate. Other techniques such as partial computation of the cost function [3] can be combined to further reduce the computational complexity.

6. Acknowledgement

This work is supported by National Nature Science Foundation of China under grant number 60302028.

Reference

[1] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advanced Video Coding", Final Committee Draft, Document JVT-F100, December 2002

[2] JVT Reference Software JM72, http://bs.hhi.de/~suehring/tml/download/jm72.zip

[3] Meng, B.J., Au, O.C., Wong, C.-W., and Lam, H.-K, "Efficient intra-prediction mode selection for 4x4 blocks in H.264", 2003 IEEE Int. Conf. Multimedia & Expo (ICME2003), Baltimore, Maryland, USA, July 2003

[4] Feng PAN, Xiao LIN, Rahardja SUSANTO, Keng Pang LIM, Zheng Guo LI, Ge Nan FENG, Da Jun WU, and Si WU, "Fast Mode Decision for Intra Prediction",JVT-G013, 7th Meeting: Pattaya II, Thailand, 7-14 March, 2003

[5] Shen, B., and Sethi, I. K., "Direct feature extraction from compressed images", SPIE: Vol.2670 Storage & Retrieval for Image and Video Databases IV, 1996