Fast Intra-Prediction Mode Selection for 4x4 Blocks in H.264

*Bojun Meng, Oscar C. Au**

Dept. of Electrical and Electronic Engineering, Hong Kong University of Science and Technology
Clearwater Bay Road, Hong Kong, China. Tel: +852 2358-7053 Email: keen@ust.hk, eeau@ust.hk**

Abstract

In the upcoming H.264, intra-prediction for each 4x4 block is used to compress I-frame. However, the full search algorithm to choose one of the 9 prediction modes is computationally expensive. In this paper, we propose a fast intra-prediction mode selection (FIPMS) method based on partial computation of the cost function, early termination and selective computation of highly probable modes. The proposed FIPMS can reduce the complexity considerably while maintaining similar PSNR and bit rate.

1. Introduction

Video Coding made a major impact in home entertainment in recent years. The ISO MPEG-1 standard made it possible to store a VCR-quality movie about 72 minutes in duration on one VCD. However, the reconstructed video quality of MPEG-1 tended to be low with lots of blocking and ringing artifacts. Using improved features and more bits, the MPEG-2 standard can offer considerably higher visual quality at higher bit rates, allowing an ITU-R 601 movie to be stored on a DVD. The MPEG-4 standard encompasses more advanced features and can achieve even higher compression efficiency with other added functions such as object-based manipulation and error-resilience, etc. The ITU-T H.261 standard was designed for video conferencing, though its quality tended to be low. H.263 was the improved standard with better visual quality and some error resilience capability. H.26L is recently being developed by ITU-T as the long-term video coding solution. A comparison between H.26L and MPEG-4 revealed that H.26L can achieve significantly better compression efficiency [1]. Thus the ISO MPEG and ITU-T VCEG decided to form a Joint Video Team (JVT) to carry on the H.26L development. The end product is called Advanced Video Coding (AVC) and will be designated as H.264 in ITU-T and MPEG-4 Version 10 in ISO.

In the emerging H.264 [2], the basic image block unit is of size 4x4, in contrast with the 8x8 block size in MPEG-1/2/4 and H.261/3. Some advance features of H.264 include 4x4 integer DCT, intra-prediction in I-frame coding, quarter-pixel motion compensation, multiple reference frames and multiple block size for P-frame coding, etc. These features help H.264 to achieve significantly higher compression efficiency than the existing standards.

The 4x4 intra-prediction is shown in Fig.1a. A 4x4 block contains 16 pixels labeled from a to p. The pixels A to Q are from the neighboring blocks and are assumed to be already decoded. A prediction mode is a way to generate 16 predictive pixel values (named a' to p') using some or all of the neighboring pixels A to Q. There are 9 prediction modes designed in a directional manner. Mode 0 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 shown in Fig. 2. Mode 1 is the vertical prediction mode in which pixels a, e, i and m are predicted by A. Mode 2 is the horizontal prediction mode in which pixels a, b, c, and d are predicted by I. The other modes are similar except that the directions are different.

To encode the prediction mode for each 4x4 block efficiently, the correlation between spatially adjacent blocks is exploited in H.264. In Fig.1b, C is the current 4x4 block to be encoded. It is observed that, depending on the prediction modes of the top block A and left block B, the probability of the 9 modes being the optimal intra-prediction mode for C is different. Thus a probability list is generated by JVT for each combination of the modes of A and B. The list is arranged in decreasing likelihood. Rather than sending the selected mode number, the position of the selected mode in the probability list is sent.

In the reference software called JM4.0d from JVT, a full search (FS) is used to examine all the 9 modes to find the one with the smallest cost. The main steps are:

1. Generate a 4x4 predicted block according to a mode;
2. Calculate sum of absolute difference (SAD_{16}) between the original 4x4 block and the predicted block;
3. Compute $Cost_{16} = SAD_{16} + 2R\lambda(Q_P)$ (1)
   where $\lambda(Q_p)$ is an approximately exponential function of the quantization factor Qp, R is the position of the present mode in the probability list;
4. Repeat 1 to 3 for all the 9 modes, and choose the one that has the minimum cost.

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.

This paper is motivated by some observations in our experiments. Firstly, we observe that the residue values of intra-prediction are usually large compared with inter-prediction using motion estimation (ME), as shown in Fig. 3. For example, the man's head is clearly visible in Fig. 3a, but almost missing in Fig. 3b. Secondly, we observe that the optimal mode (found by full search) and other "good" (second or third best) modes are most likely in similar directions. Thirdly, we observe that the direction features of the 4x4 blocks can be preserved roughly after down-sampling. Based on these observations, we propose a fast intra-prediction mode selection algorithm. The proposed algorithm is explained in Section 2, followed by simulation results in Section 3.

2. Fast Intra-Prediction Mode Selection (FIPMS)

In general, there are two ways to reduce the complexity of intra-prediction mode selection: simplify the cost function, or examine fewer modes. Typically, a 4x4 block would have rather smooth texture due to its small block size. It is thus possible to represent, at least roughly, a 4x4 block by its subsampled pixels. So instead of checking all 16 pixels in a 4x4 block at one time, we organize the 16 pixels into 4 groups: (1) pixels a,c,i,k; (2) pixels f,h,n,p; (3) pixels e,g,m,o; (4) pixels b,d,j,l. Each group is a "down-sampled" version of the original block. If the prediction mode can be decided by checking only some of the groups, we achieve computation reduction. The cost for one group is :

$$Cost_4 = SAD_4 + R\lambda(Q_p)/2 \qquad (2)$$

where SAD_4 is the sum of absolute difference for the 4 pixels in the group, $\lambda(Q_p)$ and R are the same as Eqn. 1. The sum of the $Cost_4$ for the 4 groups gives Eqn. 1.

Another important thing mentioned in the previous section is that, when the prediction modes of the top block and left block are known, we have a probability list of the mode to be chosen for present block. This offers a very good opportunity to decrease the overall complexity. We can test the most probable mode first. If the cost is smaller than a threshold T, it can be chosen as the intra-prediction mode for the current block. If not, the other modes need to be examined. Note that the 9 modes are based on directions. If one mode gives minimum cost after a partial search (using only some groups of pixels), the optimal prediction mode may most likely be either this mode or one of its two neighbors in Fig. 2 (for example, the two neighbors of mode 3 are modes 5 and 8). So we can reduce the number of candidate modes down to 3. Here is the proposed fast intra-prediction mode selection (FIPMS).

1. If the block is at the top or left of the frame, check 2 modes (DC and horizontal modes for the top block, DC and vertical modes for the left block) using all 16 pixels. Choose the mode with minimum cost, and stop. Otherwise, go to step 2;
2. If the threshold T is larger than some number, take the most probable mode as the prediction mode and stop. Otherwise, compute the $Cost_4$ of groups 1 and 2 pixels (8 pixels) for the most probable mode. If the cumulative cost is smaller than threshold T, output the most probable mode and stop. Otherwise, go to step 3;
3. For the remaining 8 modes, compute $Cost_4$ of the group 1 pixels. Choose the mode with minimum $Cost_4$ among the 8. If it is modes 1 or 2, go to step 6. Otherwise, go to step 4;
4. Compute $Cost_4$ of the group 2 pixels for the chosen mode in step 3, and its two neighboring modes (with similar direction in Fig. 2). Choose the one whose cumulative cost is the minimum, and go to step 5;
5. Compare the cost of the most probable mode and the chosen mode in step 4. Select the one with smaller cost. If it is mode 1 or 2, go to step 6. Otherwise, output it and stop;
6. (for mode 1 and 2) If the selected mode is 1, compute $Cost_{16}$ for modes 0, 1, 2, 5, and 6. Output the mode with minimum $Cost_{16}$. If the selected mode is 2, compute $Cost_{16}$ for modes 0, 1, 2, 7, and 8, and then output the one with minimum $Cost_{16}$.

The step 6 of FIPMS is complicated. Its computational complexity is about half of the full search. Many points need to be examined in detail because it is observed that if the selected mode is 1 or 2, the optimal or good mode may still be quite chaotic.

In terms of complexity, the full search (FS) algorithm checks roughly 16x9=144 pixels to decide the prediction mode for a 4x4 block. In the best case, FIPMS algorithm ends in step 1 without checking any pixel. In the second best case, only 8 pixels are checked. The worst case can happen when the most probable mode is not 1 or 2, but the selected mode in step 5 is 1 or 2, so that a five-mode full search needs to be performed in step 6. Considering some groups of pixels have been checked in the previous steps, we just examine the rest to get the $Cost_{16}$. In this case, the total number of pixels we need to check is:

$$8 + 4 * 8 + 4 * 3 + 8*2 + 12 * 3 = 104$$

As we can see, the worst case degrades the computational efficiency greatly.

3. Simulation Results and Discussions

The proposed FIPMS and the full search (FS) are simulated on four CIF sequences, Stefan, Coastguard, Container, and Akiyo. For each sequence, 100 frames are encoded with I-frame coding only, and with fixed Qp factors. Various Qp factors and thresholds for FIPMS are tested. The threshold T is set such that the there is not much drop in PSNR. The results are shown in Tables 1-4. The complexity is the total number of additions and shifts. When Qp is as small as 5, FIPMS can typically reduce computational complexity by about 3.7 times with essentially the same PSNR as FS, and a slight increase in bit rate. When *Qp*=16, FIPMS has a average computation reduction factor of 4.3. When *Qp*=31, the average computation reduction factor is 5.5. When *Qp*=48, the average reduction factor is 158, which is very large.

The reason for the large computation reduction factor at *Qp*=48 is that, when the *Qp* is large enough, the threshold *T* can be set to infinity while maintaining similar PSNR, such that no *SAD* computation is performed at all. For Stefan, the bits are even dropped by 9.4% with similar PSNR.

The inter-predicted image using motion estimation, the intra-predicted image using FS and FIPMS for Coastguard and Akiyo are shown in Fig. 3 and 4. It can be seen that the predicted images by FS and by FIPMS have very similar visual quality, though both are not as good as the inter-predicted image using motion estimation.

4. Conclusion

In this paper, we propose a fast intra prediction mode selection (FIPMS) algorithm. Computation reduction is achieved by partial computation of the cost function, early termination and selective computation of highly probable modes. Simulation results suggest that FIPMS can achieve considerable computation reduction while achieving similar PSNR and bit rate. FIPMS is especially good when *Qp* is large.

5. Acknowledgement

This work was supported by the Research Grants Council of Hong Kong SAR.

Reference

1. P. Topiwala, G. Sullivan, et. al., "Performance Evaluation of H.26L, TML 8 vs. H.263++ and MPEG-4", *Document VCEG-N18*, ITU-T Video Coding Experts Group(VCEG) Meeting, Santa Barbara, CA, USA, 24-27 September 2001.
2. Joint Video Team (JVT), "Working Draft of JVT Standard, Number 2, Revision 2", *Document JVT-B118r2*, Generated on March 15, 2002.
3. A.M Tourapis, O.C. Au, M.L. Liou, "New Results on Zonal Based Motion Estimation Algorithms-Advanced Predictive Diamond Zonal Search", *Proc. Of IEEE Int. Sym. on Circuits and Systems*, pp. 241-244, vol.2, May 2001.

Qp	Method	PSNR(dB)	Complexity	Bits
5	FS	55.91	347.6 M	65693 K
	FIPMS, T=24	55.90	128.7 M	66136 K
16	FS	46.44	347.6 M	34573 K
	FIPMS, T=40	46.42	120.0 M	34905 K
31	FS	33.92	347.6 M	11481 K
	FIPMS, T=60	33.90	109.8 M	11646 K
48	FS	21.74	347.6 M	1680 K
	FIPMS, T=∞	21.73	2.2 M	1522 K

Table 1 Results of 'Stefan', CIF, 100 frames

Qp	Method	PSNR(dB)	Complexity	Bits
5	FS	55.81	347.6 M	60158 K
	FIPMS, T=80	55.81	80.3 M	60512 K
16	FS	45.86	347.6 M	31285 K
	FIPMS, T=100	45.85	63.6 M	31474 K
31	FS	33.27	347.6 M	7963 K
	FIPMS, T=110	33.26	60.6 M	8060 K
48	FS	23.92	347.6 M	676 K
	FIPMS, T=∞	23.97	2.2 M	679 K

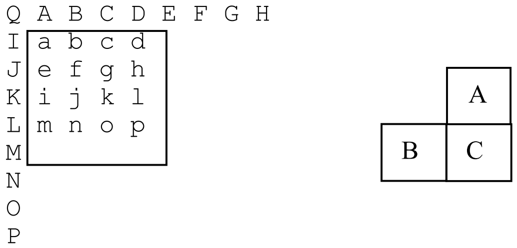
Table 2 Results of 'Coastguard', CIF, 100 frames

Qp	Method	PSNR(dB)	Complexity	Bits
5	FS	56.12	347.6 M	49961 K
	FIPMS, T=24	56.11	95.1 M	50465 K
16	FS	47.22	347.6 M	22703 K
	FIPMS, T=32	47.21	79.7 M	22979 K
31	FS	36.63	347.6 M	5616 K
	FIPMS, T=60	36.60	45.4 M	5663 K
48	FS	26.89	347.6 M	1138 K
	FIPMS, T=∞	26.88	2.2 M	1140 K

Table 3 Results of 'Container', CIF, 100 frames

Qp	Method	PSNR(dB)	Complexity	Bits
5	FS	56.14	347.6 M	39912 K
	FIPMS, T=16	56.11	82.8 M	40281 K
16	FS	47.37	347.6 M	14452 K
	FIPMS, T=20	47.33	81.0 M	14660 K
31	FS	38.29	347.6 M	3475 K
	FIPMS, T=40	38.24	68.1 M	3515 K
48	FS	28.06	347.6 M	701 K
	FIPMS, T=∞	27.94	2.2 M	708 K

Table 4 Results of 'Akiyo', CIF, 100 frames



(a) *(b)*

Fig 1. (a) A 4x4 block and its neighboring pixels
(b) Mode prediction among adjacent 4x4 blocks

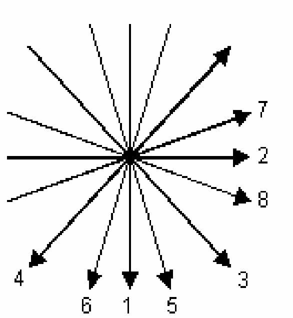
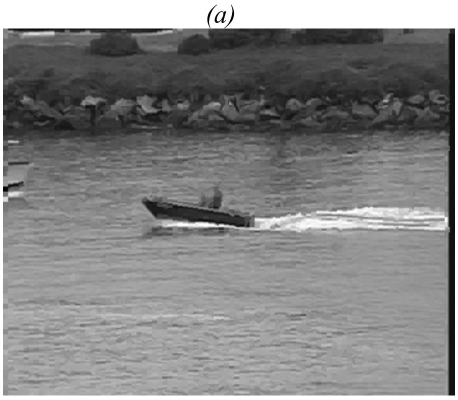
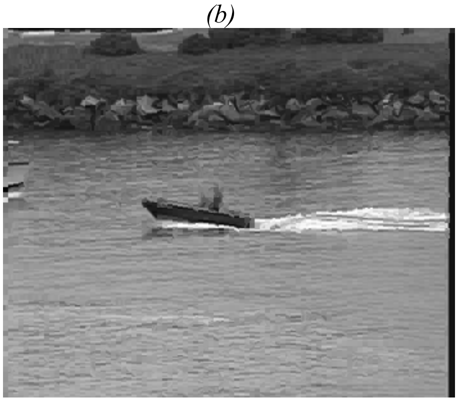


Fig 2. Eight of the 9 intra-prediction modes in H.264

(a)



(b)



(c)

Fig 3. (a) Inter-predicted image using ME, PSNR=27.3dB (b) Intra-predicted image using FS, PSNR=26.9dB, (c) Intra-predicted image using FIPMS, T=80, PSNR=26.5dB

(a)

(b)

(c)

Fig 4. (a)Inter-predicted image using ME, PSNR=45.5dB (b) Intra-predicted image using FS, PSNR=31.6dB, (c) Intra-predicted image using FIPMS, T=20, PSNR=31.4dB