

A FAST H.264 INTRA PREDICTION ALGORITHM USING MACROBLOCK PROPERTIES

Chun-Ling YANG^{1,2}, Lai-Man PO¹, Wing-Hong LAM¹

1. Department of Electronic Engineering, City University of Hong Kong,
83 Tat Chee Avenue, Kowloon Tong, Hong Kong, China

2. College of Electronic and information engineering, South China University of Technology,
Guangzhou, Guangdong, 510640, China

Emails: eeclyang@scut.edu.cn, celmpo@cityu.edu.hk

ABSTRACT

In the newest video coding standard of H.264, intra frame prediction is conducted in the spatial domain. There are two types of intra prediction for luma -- Intra_16×16 does prediction for the whole 16×16 luma macroblock and Intra_4×4 does prediction for 4×4 block. Four and nine modes are supported by Intra_16×16 and Intra_4×4, respectively. The Full Search algorithm is used in the JVT reference software to choose the best modes, however it is very computationally expensive. In this paper, a new fast intra prediction algorithm using macroblock properties (FIPAMP) is proposed. Experimental results show that the proposed fast algorithm can achieve 10% to 40% computation reduction while maintaining similar PSNR and bit rate performance of H.264 codes.

1. INTRODUCTION

H.264 is the newest international video coding standard [1]. New and advanced techniques are introduced in this new standard, such as intra prediction for I-frame encoding, multi-frames inter prediction, small block-size transform coding, context-adaptive arithmetic entropy coding, deblocking filtering, etc. These advanced techniques make this new standard provides approximately 50% bit rate saving for equivalent perceptual quality relative to the performance of prior standards. In contrast to H.261/263 and MPEG-1/2/4 standards, where intra prediction has been conducted in the transform domain, intra prediction in H.264 is always conducted in the spatial domain, by referring to neighboring samples of previously coded blocks which are to the left and/or above the block. There are two intra prediction types in H.264. They are called Intra_16×16 and Intra_4×4 types. In addition, the Intra_16×16 supports four intra prediction modes while the Intra_4×4 supports nine modes.

In the Intra_4×4 prediction type [1-2], each 4×4 block is predicted from spatially neighboring samples as illustrated on the left-hand side of Fig. 1. The 16 samples of the 4×4 block which are labeled as *a-p* are predicted using prior encoded samples in adjacent blocks labeled as *A-Q*. In addition to "DC" prediction mode (mode 2) where one value is used to predict the entire 4×4 block, eight directional prediction modes are specified as illustrated on the right-hand side of Fig. 1. For Intra_16×16 prediction, the whole macroblock is predicted with four modes.

The four Intra_16×16 modes are listed in Table.1. Prediction mode 0 (vertical prediction), mode 1 (horizontal prediction), and mode 2 (DC prediction) are similar to the modes in Intra_4×4 prediction except that instead of 4 neighbors, 16 neighbors are used to predict a 16×16 macro-block. The mode 3 is a plane prediction and the details of this mode could be found in [1, 3].

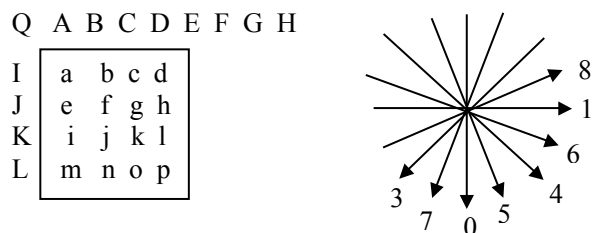


Fig. 1. Left: Intra_4×4 prediction is conducted for samples *a-p* of a block using samples *A-Q*. Right: 8 of the 9 prediction modes for Intra_4×4 prediction.

Table 1. The four Intra_16×16 prediction modes

Mode 0	Mode 1	Mode 2	Mode 3
Vertical prediction	Horizontal prediction	DC prediction	Plane prediction

With use of these advanced intra prediction techniques in H.264, the coding efficiency for I-frame compression is significantly improved. The drawback, however, is the extremely high computing requirement to find the best prediction mode(s), which may limit the practical implementation of this newest video standard. In this paper, we propose a new fast H.264 intra prediction type selection algorithm based on macroblock properties to speed up the mode selection process. The rest of this paper is organized as follows. In section II the H.264 intra prediction mode selection process is first summarized. The properties of macroblocks and the new algorithm are given in section III. Section IV presents the simulation results to demonstrate the efficiency of the new algorithm and some concluding remarks are given in the last section.

2. H.264 INTRA MODE(S) SELECTION

In the newest H.264 JVT reference software JM7.3 [4], the Full Search (FS) algorithm is used to examine all the possible intra

prediction modes to find the best mode(s). The major steps for the FS intra prediction mode(s) selection process can be summarized as:

Step 1: Find the Best Intra_16×16 Prediction Mode

Generate four prediction blocks according to the four Intra_16×16 modes and then calculate their residual macroblocks. For each residual macroblock, do:

- 1.1: Perform Hadamard transform for each 4×4 block.
- 1.2: Extract all the DC from these sixteen 4×4 blocks and divided them by 4, to form another 4×4 block. Perform Hadamard transform for this 4×4 block.
- 1.3: Sum up the absolute value of all the Hadamard transform coefficients (except the DC in step 1.1), use the summation as the cost.

Find the mode with the smallest cost as the best Intra_16×16 prediction mode.

Step 2: Find the Best Intra_4×4 Prediction Modes

Divide the macroblock into sixteen 4×4 non-overlapped blocks. For each 4×4 block find the best mode by:

- 2.1: Generate nine 4×4 blocks based on the nine Intra_4×4 prediction modes.
- 2.2: For each mode, compute the Cost_{4×4} defined as:

$$\text{Cost}_{4 \times 4} = \text{MSE}_{4 \times 4} + R\lambda(Q_p)$$

where MSE_{4×4} is the encoded Mean Square Error of this 4×4 block, R is the bit number for coding this block, and λ(Q_p) is an approximately exponential function of the quantization factor Q_p.

- 2.3: The best mode is the one with the smallest Cost_{4×4}.

Step 3: Find the Best Intra Prediction Mode(s)

Compute the overall Cost = MSE + Rλ(Q_p) for this macroblock using the best Intra_16×16 prediction mode and the best sixteen Intra_4×4 prediction modes. The mode (modes) with the smallest Cost will be chosen.

Based on this full search mode selection process, the computational load of the I-frame encoding of H.264 is extremely high. In order to tackle this problem, a new fast algorithm using macroblock properties to speed up this selection process is presented in the next section.

3. Prediction Modes Analysis

The Intra_16×16 prediction type does prediction for the whole 16×16 macroblock, it should be more suitable for encoding of very smooth areas of a picture. The Intra_4×4, however, is based on predicting each 4×4 block separately and is more suitable for encoding of picture's region with significant detail. Based on this assumption, we can select the intra prediction type according to the features of the macroblock instead of full search algorithm. In the following of this subsection, the four Intra_16×16 prediction modes listed in Table 1 are analyzed and the fast intra prediction type selection method is proposed.

The DC Prediction Mode

With the DC prediction mode (mode 2), the mean of the above pixels and left pixels is used to predict all the pixels of this macroblock. It means that the values of all the pixels are almost same in the Intra_16×16 DC prediction macroblock (see Fig. 2 (c)). The Mean Absolute Deviation (MAD) can be used to distinguish this kind of macroblock. The MAD of a macroblock is defined as:

$$MAD = \frac{1}{16 * 16} \sum_{y=0}^{15} \sum_{x=0}^{15} |p(x, y) - m| \quad (1)$$

where p(x, y) is pixel value at (x, y), m is the mean of all the pixels in this macroblock, and there are 16 pixels in each column or each row in a macroblock.

The DC prediction mode macroblock has smaller MAD. We can choose a threshold T_{DC}, if MAD of a macroblock is smaller than or equal to this threshold, then it is smooth.

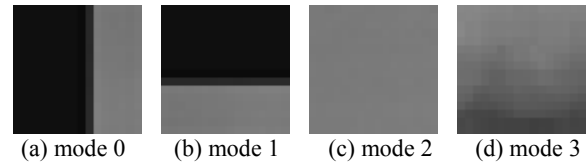


Fig. 2: Macroblocks for the four Intra_16×16 prediction modes.

The Vertical and Horizontal Prediction Modes

With vertical prediction mode (mode 0), the above pixel value is used to predict all the pixels in a column in the macroblock. It means that the vertical prediction mode macroblock has smooth columns --- all the pixels values in a column are almost equal (see Fig.2 (a)). The MAD_v can be used to distinguish this kind of smooth column macroblock, and MAD_v is defined as

$$MAD_v = \frac{1}{16} \sum_{x=0}^{15} MAD_x \quad (2)$$

where MAD_x is the mean absolute deviation of column x in the macroblock, which is defined as:

$$MAD_x = \frac{1}{16} \sum_{y=0}^{15} |p(x, y) - m_x| \quad (3)$$

where, m_x is the mean of the column x. The vertical prediction mode macroblock has small MAD_v.

The Horizontal prediction mode is similar to the vertical prediction mode, except that the left pixel value is used to predict all the pixels of each row in the macroblock. So the horizontal prediction mode macroblock has smooth rows -- the pixels in a row have the similar values (see Fig.2 (b)). The MAD_h can be used. Computing MAD_h is similar with computing MAD_v, except that MAD_v is used to compute MAD_h and MAD_v is the mean absolute deviation of row y. Another threshold can be chosen, if MAD_v or MAD_h of a macroblock is smaller than or equal to this threshold, then it is smooth.

The Plane Prediction Mode

The plane prediction mode (Mode 3) is a little complex, the prediction equation for each pixel in a macroblock is given [1]:

$$pred(x, y) = clip(a + b * (x - 7) + c * (y - 7) + 16) \gg 5 \quad (4)$$

where (x, y) is the position of the pixel in this macroblock, $clip(\bullet)$ is the operator to make the value not smaller than zero and not bigger than 255; a , b , and c are constants depending on the above and left pixel values of the macroblock.

From equation (4), the plane prediction mode macroblock has the following features: The $Block_v$ and $Block_H$ have smooth columns and smooth rows respectively, $Block_v$ and $Block_H$ are obtained from the macroblock by the following equations.

$$Block_v(x, y) = p(x, y + 1) - p(x, y) \quad (5)$$

$$Block_H(x, y) = p(x + 1, y) - p(x, y) \quad (6)$$

We call MAD_v of the $Block_v$ as MAD_{vv} , MAD_H of the $Block_H$ as MAD_{HH} . So MAD_{vv} and MAD_{HH} are all smaller in the plane prediction macroblock. The macroblock in Fig.2 (d) is suitable for mode 3. The third threshold can be used, if MAD_{vv} and MAD_{HH} are all smaller than or equal to this threshold, it is smooth.

The MADs (MAD , MAD_v , MAD_H , MAD_{vv} and MAD_{HH}) of detailed area are generally larger than that of smooth area. Thus, we can use the predefined thresholds to perform this classification. If the MADs are smaller than or equal to the corresponding thresholds, the macroblock is smooth, thus the four Intra_16×16 prediction modes are used to find the best one as this macroblock prediction mode. Otherwise, Intra_4×4 prediction should be used.

The FIPAMP

The steps of the proposed fast intra prediction algorithm (FIPAMP) can be summarized as:

Step 1. Find the best intra prediction type---Intra_16×16 or Intra_4×4

- 1.1: Compute the MAD of the macroblock by eq. (1), if $MAD \leq T_{DC}$, the macroblock is smooth and goto [step 2](#).
- 1.2: Compute the MAD_v of the macroblock, if $MAD_v \leq T_{VH}$, the macroblock is smooth goto [step 2](#).
- 1.3: Compute the MAD_H of the macroblock, if $MAD_H \leq T_{VH}$, the macroblock is smooth goto [step 2](#).
- 1.4: Compute the MAD_{VV} and MAD_{HH} of the macroblock, if $MAD_{VV} \leq T_p$ and $MAD_{HH} \leq T_p$, this macroblock is smooth and goto [step 2](#). Otherwise goto Step 3

Step 2. Find the best prediction mode of the macroblock from the four Intra_16×16 modes.

Using the method described in [step 1](#) in section II, find the best from the four Intra_16×16 modes. It is chosen as the best intra prediction mode of the macroblock. Then end the process.

Step 3. Find the best prediction modes of the macroblock from nine intra_4×4 modes.

Using the method described in [step 2](#) in section II, find the best one from the nine Intra_4×4 modes for each 4×4 block, and they are chosen as the best modes of this macroblock. Then end the process.

Unlike the FS, where two intra prediction type, Intra_16×16 and Intra_4×4, are all used to find the best prediction mode(s) for each macroblock, only one type, Intra_16×16 or Intra_4×4, is used to find the best prediction mode(s) in our FIPAMP, which can save much computation, especially for the sequences with more smooth area, because the computation load is larger in Intra_4×4 than that in Intra_16×16. And the computation of macroblock classification added in the FIPAMP is very smaller comparing to Intra_16×16 or Intra_4×4. So the FIPAMP can save much computation than FS, which will be demonstrated by the simulation results in the next section.

4. SIMULATION RESULTS

Experimental simulations are performed on the first frames of some well-known QCIF (176×144) video sequences, they are MissA, Bream and Claire with more smooth areas, and Coastguard, Carphone and Foreman with more detailed areas. The simulation results in terms of PSNRs, bit-rate and encoding time using FS, and the proposed FIPAMP are listed in Table 2. The FS is the full search method, which is used in the JVT reference software JM7.3 program [4]. The FIPAMP is the proposed fast algorithm, which implemented by modifying JM7.3 program. All the results are performed on a PIII/866MHZ personal computer. The three thresholds, T_{DC} , T_{VH} , T_p , used to distinguish the intra prediction type of a macroblock -- Intra_16×16 or Intra_4×4, are obtained by intensive simulation experience. We found that, generally, $T_{DC} = 3$, $T_{VH} = T_p = 1$ is better.

In Table2, PSNRY, PSNRU and PSNRV are the Peak Signal to Noise Ratio of color component Y, U and V respectively. Bits/Pic is the number of bit for encoding the first frame of each sequence. Bit Increase is the percent increase in bit using the proposed algorithm as compared with using FS. Times is the encoding time and Time saving is the percent of time saving using FIPAMP as compared with FS. Some decoded images of Component Y are shown in Fig.3.

The results in Table 2, show that the proposed FIPAMP can achieve quite good computation reduction while maintaining almost the same PSNR and bit rate, especially for the low detailed sequences, the most computation time saving is 41.21%.

5. CONCLUSION

In this paper, we have analyzed the feature of Intra prediction modes used in H.264. Based on these observations, a new Fast Intra Prediction Algorithms based on macroblock properties (FIPAMP) is proposed for fast H.264 intra frame coding. The FIPAMP uses a smoothness classification based on the predefined MAD thresholds for Intra_16×16 and Intra_4×4 selection. A lot of computation could be saved especially for smooth sequence encoding while maintaining almost the same PSNRs and bit rate. Experimental results show that the proposed FIPAMP algorithm can achieve 10% to 40% computation reduction while maintaining similar PSNRs and bit rate.

6. ACKNOWLEDGMENTS

The work described in this paper was substantially supported by a grant from the Research Grants Council of Hong Kong, City University of Hong Kong, Hong Kong SAR, China. [Project No. 9040602 (CityU 1121/01E)]

7. REFERENCES

- [1] Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG, "Draft ITU-T Recommendation and Final Draft international Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)", JVT-G050, March 2003.
- [2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H.264/AVC Video coding Standard", IEEE Trans. On CAS for video Technology, No.7, Vol. 13, p560-576, July 2003.
- [3] H.264/MPEG-4 Part 10 tutorials: H.264 Intra prediction. <http://www.vcodex.fsnet.co.uk/h264.html>
- [4] H.264 JVT reference software, <http://bs.hhi.de/~suehring/tml/download>.

Table 2. The simulation results comparing FIPAMP with FS for the first frame of sequences

Sequences (176*144)	method	PSNRY	PSNRU	PSNRV	Bits/Pic	Bit Increase(%)	Times (ms)	Time saving (%)
MissA	FS(H.264)	40.95	39.64	40.00	10680		2164	
	FIPAMP	40.84	39.64	40.02	10656	-0.22	1389	35.81
Bream	FS(H.264)	39.00	40.73	41.04	23088		2938	
	FIPAMP	38.91	40.73	41.04	23144	0.24	2152	26.75
Claire	FS(H.264)	40.96	40.64	42.57	12312		2424	
	FIPAMP	41.08	40.64	42.57	12472	1.3	1425	41.21
Coastguard	FS(H.264)	35.57	44.64	46.16	25984		3486	
	FIPAMP	35.57	44.64	46.16	26064	0.31	3125	10.36
Carphone	FS(H.264)	37.82	40.81	41.94	21936		3110	
	FIPAMP	37.87	40.74	41.89	22072	0.82	2633	15.34
Foreman	FS(H.264)	37.44	41.17	43.01	20976		3088	
	FIPAMP	37.46	41.28	42.97	21120	0.69	2651	14.15

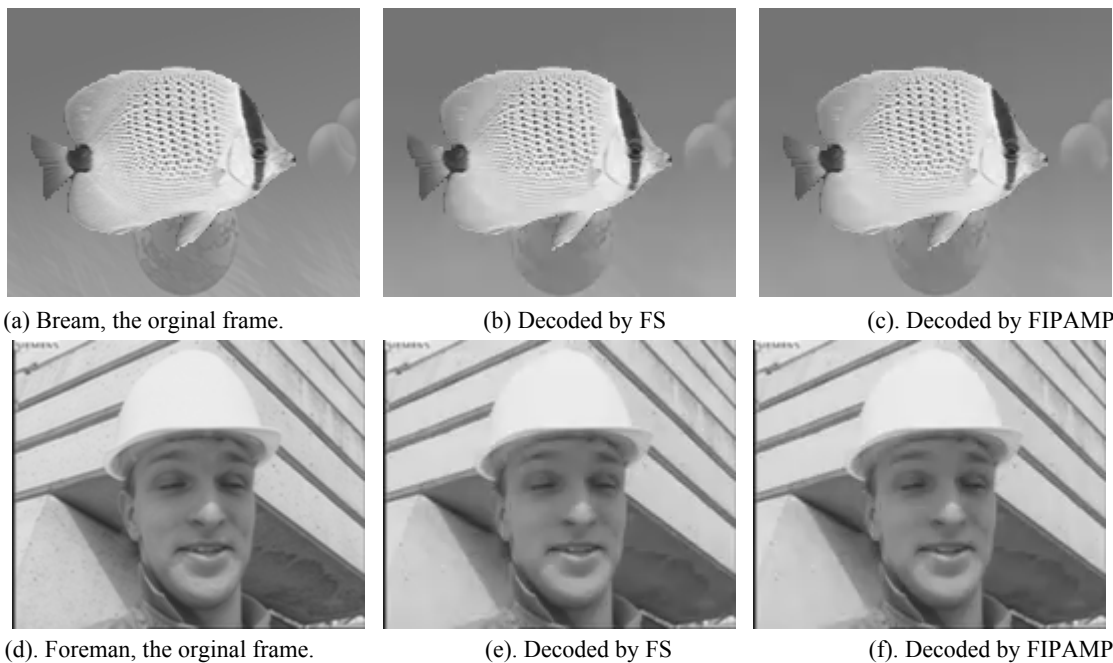


Fig.3. The decoded images of Component Y