A Fast Intra Prediction Based on Haar Transform in H.264/AVC

Shwu-Huey Yen[1] Meng-Ju Lin[3] Shin-Jia Hwang[2] Tai-Kuang Li[4]

Department of Computer Science and Information Engineering
Tamkang University, Taipei, Taiwan, R.O.C.
{105390[1], sjhwang[2] }@mail.tku.edu.tw lmru@so-net.net.tw[3] 695410141@s95.tku.edu.tw[4]

***Abstract*—H.264/AVC is one of the state of the art in video compressing standards. It has advantages of low bit-rate and high image quality. However, the complexity of H.264/AVC encoder limits applications in the real time video communication. In this study, a fast intra prediction with Haar transform algorithm (FIPHTA) is proposed to simplify the complexity of luma intra prediction. For each luma macroblock (MB), a Haar transform is first executed. By observing LL coefficients, this MB will be classified as I16-MB, I4-MB, or both. An I16-MB indicates that this MB is smooth and only Intra 16×16 (I16) prediction modes are to be considered. An I4-MB indicates that this MB is textured and only Intra 4×4 (I4) prediction modes are to be considered. And the rest of MBs (both) have to consider both prediction modes as in the standard H.264/AVC. Next, to choose an I16 prediction mode for a MB, the vertical and horizontal energies of the LL coefficients are calculated to reduce candidate modes. Similarly, the pixel-based vertical and horizontal energies are calculated for a 4x4 block to reduce I4 candidate modes. The simulation showed that the proposed algorithm can maintain the similar mode selection results comparing to the full search algorithm of H.264 but with a remarkable computation deduction.**

Keywords- Intra predictio; H.264/AVC; Haar transform

I. Introduction

The H.264/MPEG-4 Part 10 Advanced Video Coding (H.264/AVC) is the latest coding standard developed by Joint Video Team of ITU-T VCEG and ISO/IEC MPEG [1] [2] [3]. To achieve good coding performance, H.264/AVC adopts quite a few state-of-the-art techniques, and the enhanced Intra mode prediction technique is one of the important factors that contribute to the success of H.264/AVC. It adopts the rate-distortion optimized (RDO) method for mode decision by comparing all possible combination of mode selections and the corresponding coding efficiencies for a MB. The intra mode prediction supports 3 kinds of block sizes, 4×4, 8×8, and 16×16, where the 8×8 size is for chroma blocks and the others are for luma blocks. There are 9 modes for 4×4 luma block, 4 modes for 16×16 luma block, and 4 modes for 8×8 chroma block. In the H.264/AVC encoder, a full search (FS) is used to find the best modes combination with minimum RD cost, and chroma component and luma component are combined together for evaluation. Thus, for a MB, a total of $C_8*(I_4*9+I_{16})$ RDO computations must be completed in order to find the best modes selection. C_8 is the number of prediction modes for chroma 8×8, I_4 and I_{16} are for luma 4×4 and 16×16. That is a total of 4*(16*9 + 4) = 592 RDO computations. As a result, the full search in H.264 can have the optimal RDO performance, but it also leads to drastic increasing of the coding complexity. Therefore, a fast intra prediction algorithm that reduces the computational complexity and maintains the video quality is very practical and important in the real time applications.

From the definition of intra modes, a strong relation between mode selection and directionality can be observed. Thus, edge information is the basis of several existing research towards fast intra prediction algorithms [4]-[8]. Pan et al. [4] proposed an algorithm by performing Sobel operator to detect the edge direction in the current block and then utilize the edge direction histograms to decide the best prediction mode. However, applying Sobel operator on every pixel and constructing histograms still are time consuming. Wang et al. [5] proposed a dominant-edge-strength (DES) method to reduce computation complexity. DES detects the dominant edge direction in a 4×4 luma block such that only modes accordance with the detected edge direction are proceeded to further RDO calculation. The DES operation first divides a 4×4 block into 4 sub-blocks each with dimension of 2×2. The average pixel magnitude of each sub-block is then calculated to form a 2×2 synthesized block Take this 2×2 block further to calculate with 5 filters (corresponding to four major directions-horizontal, vertical, 45° diagonal, 135° diagonal- and one non-direction) to find the dominant edge direction. Later, an improvement was proposed in [6]. Lin et al. [6] proposed a direction-detection (DD) method to detect the dominant edge direction inside a 4×4 luma block. The idea is similar to that in [5], but they used a pixel-based difference method to find the dominant edge direction among 4 major directions. The DD method is more calculation efficient than the DES and they claimed that it also improved the prediction accuracy in 45° diagonal and 135° diagonal- directions comparing to that in [5].

Most of research works focus on reducing candidate modes for luma I4 because there are 9 complex prediction modes. To reduce the number of candidates in I16, the 16×16 MB is usually down sampled to be a 4×4 block then apply the same algorithm as for I4. When considering the final mode selection in luma only, all possible combinations of modes of I4 and I16

978-1-4244-5228-6/09/$26.00 c2009 IEEE

still are considered together. In [8], they divided a MB into four 8×8 blocks, and calculate AC/DC ratio in order to decide whether such 8×8 block is suitable for I4 or should be larger (I8 or I16 in H.264/AVC FRExt). By this way, the number of RDO computations can be reduced further. However, the decision is unclear when there are some 8×8 blocks classified as I4 and some are classified as I8 or I16 within one MB.

In this paper, we present a fast intra prediction algorithm which utilizes the LL coefficients of the Haar wavelet transform of a MB. The concept is that if a MB is determined to be an intra 16×16 (I16) prediction mode by the H.264 encoder, then it must be a smooth MB. Haar transform is adopted because only addition and subtraction operations are involved, and the LL coefficients contain exactly the smoothness information of the MB. The standard deviation of the LL coefficients is used to determine whether a given MB is smooth enough that only I16 luma predictions are candidate modes, or it is textured that only intra 4×4 (I4) luma predictions are appropriate, or it has to check for both predictions as in the standard H.264. In addition, an LL-based fast I16 intra mode decision algorithm and a pixel-based fast I4 intra mode decision algorithm are also proposed to reduce the computation complexity in I16 and I4 mode selection. By this way, we can reduce more than half of the candidate modes in the intra prediction with similar results as the full search in H.264/AVC.

The rest of the paper is organized as follows. Section II gives an overview of intra prediction in H.264/AVC. Section III presents the proposed algorithm. Experimental results are presented in Section IV and conclusion will be given in Section V.

II. OVERVIEW OF INTRA PREDICTION IN H.264/AVC

In the H.264/AVC codec, intra prediction supports 3 kinds of block size, 4×4, 8×8 and 16×16, where the size 8×8 is for chrominance and the others are for luminance. The luma intra prediction has two prediction strategies. The intra 4×4 (I4), which predicts each 4×4 luma block individually, is well suited for coding parts of a MB with significant details. The intra 16×16 (I16), which predicts the entire 16×16 luma block, is more suitable to process smooth image area where an prediction is perform for the MB.

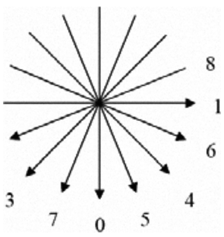

Figure 1. Directions of 9 Intra4x4 prediction modes in H.264

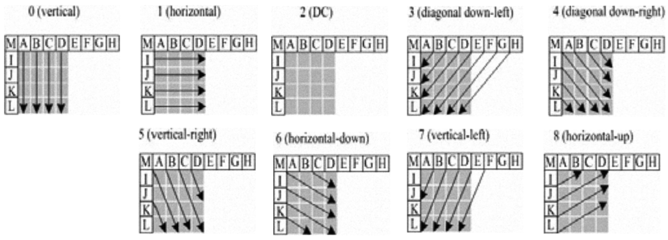

Figure 2. Nine modes of Intra4x4 prediction

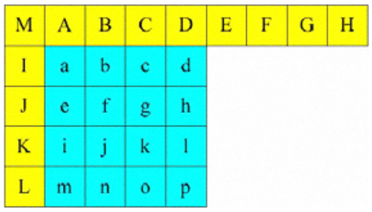

Figure 3. Intra4x4 block and its neighboring pixels

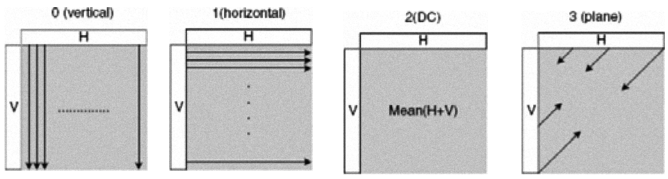

Figure 4. Four modes of Intra16x16

A. *Intra 4×4 prediction*

Fig.s 1 and 2 show the nine prediction modes of the Intra 4×4 prediction's direction. Mode 2 is called the DC prediction which all pixels from a to p are predicted by (A+B+C+D+I+J+K+L) / 8 as in Fig. 3.

B. *Intra 16×16 prediction*

The Intra 16×16 prediction is similar as Intra 4x4 prediction described above, the entire MB may be predicted. This is well suited for smooth area where a uniform prediction is performed for the whole MB. Four prediction modes are supported such as Fig 4.

C. *Chroma 8×8 prediction*

The Chroma 8×8 prediction in H.264/AVC is similar as Intra 16×16 prediction described above except that the mode order is different: DC (Mode 0), horizontal (Mode 1), vertical (Mode 2) and plane (Mode 3).

Since the chroma and luma components are independent, if the RDO computations for chroma and luma components are executed separately, then the total number of RDO operations will be reduced to as low as 152 (= 4 + (9·16+4)). In H.264/AVC codec JM Version 12.3 Reference Software [9], it provides an enhancement in computations for intra chroma components- FastCrIntraDecision option. Comparing to the full search of chroma mode selection in H.264, this option can save approximately 70% of computational time and maintains almost the same PSNR quality and bit rate. Thus, we will focus the study on simplifying the prediction mode selections on luma I4 and luma I16 only.

III. Proposed Fast Intra Prediction Haar Transform Algorithm (FIPHTA)

The proposed algorithm is composed of three sub-algorithms. For a given MB, after the Haar transform is applied, the prediction block size is first determined by the standard deviation of LL coefficients (LL-SD Alg.). If I16 is to be performed, an I16 LL-based fast prediction algorithm (LL-based I16 Alg.) will be executed. If I4 is to be performed, an I4 pixel-based fast prediction algorithm (Pixel-based I4 Alg.) will be executed. An overall flowchart of the proposed algorithm is given in Fig. 5.

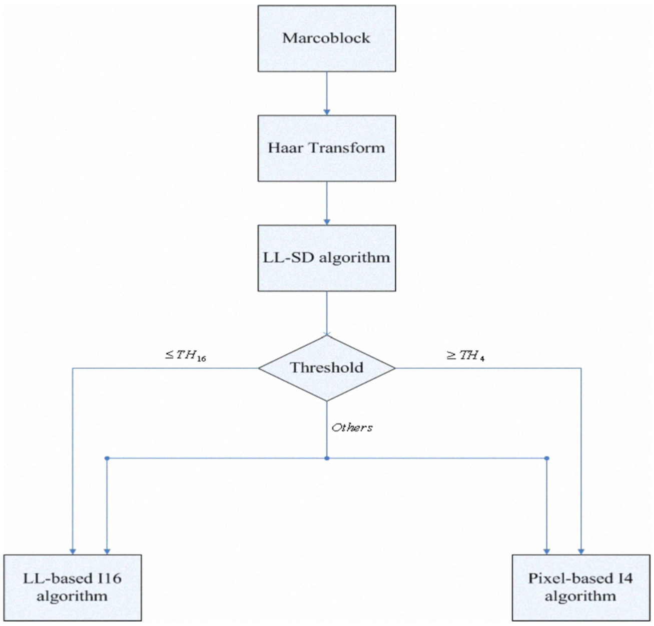

Figure 5. The overall flow diagram of the proposed algorithm for one MB

A. LL-Coefficients Standard Deviation (LL-SD) Algorithm

The concept of LL-SD is that intra coding block size is highly dependent on the smoothness of the image block. A smooth image block tends to use large the intra coding block size, and a complex image block tends to use small one. As shown in (1), let LL-SD be the standard deviation of the 8×8 LL coefficients after Haar transform of a given MB. If the texture is more complex, the LL-SD of the image block is lager. Otherwise, the LL-SD will be smaller. Therefore, we can use the LL-SD to predict the intra prediction block size. The formula for LL-SD is given in (1).

$$\mathrm{LL-SD} = \text{Square Root}\,[\sum_{u=0,j=0}^{7}\left(X_{i,j}-\overline{X}\right)^2/63)], \qquad (1)$$

where $X_{i,j}$ is the wavelet coefficient in LL_{8x8}, $\overline{X}$ is the average of these coefficients.

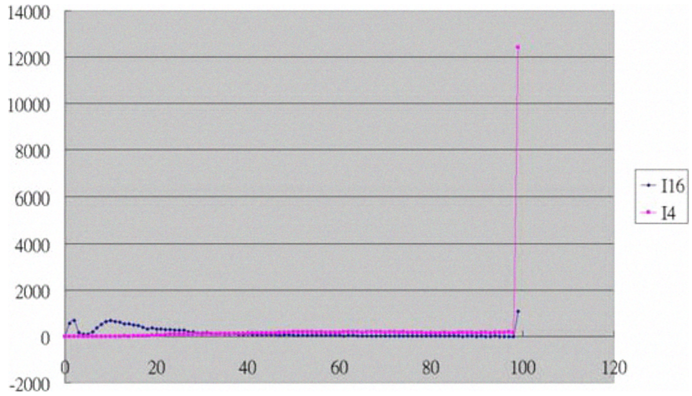

Figure 6. The histogram of the LL-SD for intra prediction block size

Fig. 6 shows the histogram of the LL-SD values for different intra prediction block size used in 100 I-frames of video sequence "Foreman". The red curve indicates the number of MBs using 4×4 prediction block size, and the blue for number of MBs using 16×16 prediction block size. As expected, most of I16 blocks have small LL-SD and I4 blocks have large LL-SD values. So, according to the LL-SD, We define two thresholds TH_4 and TH_{16}. If LL-SD is small, i.e. LL-SD ≤ TH_{16}, I16 prediction modes are the only candidates to be considered. If LL-SD is large, ie. LL-SD ≥ TH_4, I4 prediction modes are the only candidates to be considered. For those MBs with LL-SD betweens thresholds, i.e. TH_{16} < LL-SD < TH_4, all of I4 and I16 prediction modes should be explored. In our approach, the thresholds vary with Quantization Parameter (QP), because higher QP tends to smooth the image block. According to experiments, TH_4 and TH_{16} are defined in (2), (3).

$$TH_4 = \begin{cases} 0.712\times QP+59.75 & if\ QP\le 40, \\ \infty & if\ QP>40. \end{cases} \qquad (2)$$

$$TH_{16} = \begin{cases} 0 & if\ QP<20, \\ 0.683\cdot Temp+4.814 & if\ 20\le QP\le 28, \\ Temp & if\ QP>28. \end{cases} \qquad (3)$$

where $Temp = 0.000215\cdot QP^{3.351} - 4.65$.

B. *Intra_16x16 LL-Base Fast Mode Decision* (*LL-based I16*) *Algorithm*

To choose a prediction mode for I16, we use the LL coefficients of the Haar transform to calculate the vertical energy and horizontal energy to determine the dominant edge direction. The steps are described below.

Step1: Check the location of the given I16 block. If it is located at the upper margin of the frame, then only mode 1 (horizontal) and mode 2 (DC mode) are explored in RDO computations. If it is located at the left margin, then only mode 0 (vertical) and mode 2 are explored in RDO computations. For other locations, the algorithm proceeds to the following steps 2~4.

Step2: Calculate the vertical energy (VE_L) and horizontal energy (HE_L) of LL coefficients by (4).

$$\begin{aligned} VE_L &= \sum_{i=0}^{7}\left(\sum_{j=0}^{7} |X_{i,j} - X_{i,j+1}|\right), \\ HE_L &= \sum_{j=0}^{7}\left(\sum_{i=0}^{7} |X_{i,j} - X_{i+1,j}|\right). \end{aligned} \tag{4}$$

Step3: Define R_{16} to be the ratio of $\log(VE_L)$ and $\log(HE_L)$ as in (5). If R_{16} is smaller than or equal to TH_{16V}, then only mode 0 and mode 2 are explored in RDO computations. If R_{16} is larger than or equal to TH_{16H}, then only mode 1 and mode 2 are explored in RDO computations.

$$R_{16} = \log(VE_L)/\log(HE_L)\,. \tag{5}$$

Step4: If R_{16} is between Th_{16V} and TH_{16H}, i.e., $Th_{16V} < R_{16} < TH_{16H}$, then 4 prediction modes of I16 are explored in RDO computations.

C. *Intra_4x4 Pixel-Based Fast Mode Decision (Pixel-based I4) Algorithm*

Since a 4x4 block is small in size, pixel values are used instead of LL coefficients to calculate the vertical and horizontal energies. The algorithm is similar to that of I16, but the selected modes include the major direction mode as well as the neighboring modes. The algorithm is described below.

Step1: Check the location of the given I4 block. If it is located at the upper margin of the frame, then only modes 1, 2, 8 are explored in RDO computations. If it is located at the left margin, then only modes 0, 2, 3, 7 are explored in RDO computations. For other locations, the algorithm proceeds to the following steps 2~4.

Step2: Calculate the vertical energy (VE_p) and horizontal energy (HE_p) of pixel values by (6).

$$\begin{aligned} VE_p &= \sum_{i=0}^{3}\left(\sum_{j=0}^{3} |X_{i,j} - X_{i,j+1}|\right), \\ HE_p &= \sum_{j=0}^{3}\left(\sum_{i=0}^{3} |X_{i,j} - X_{i+1,j}|\right). \end{aligned} \tag{6}$$

Step3: Define R_4 to be the ratio of $\log(VE_p)$ and $\log(HE_p)$ as in (7). If R_4 is smaller than or equal to TH_{4V}, then only modes 0, 5, 7 and mode 2 are explored in RDO computations. If R_4 is larger than or equal to TH_{4H}, then only modes 1, 6, 8 and mode 2 are explored in RDO computations.

$$R_4 = \log(VE_p)/\log(HE_p)\,. \tag{7}$$

Step4: If R_4 is between Th_{4V} and TH_{4H}, i.e., $Th_{4V} < R_4 < TH_{4H}$, then 9 prediction modes of I4 are explored in RDO computations.

IV. Simulation Results

The proposed algorithm (FIPHTA) is implemented in JM12.3 encoder. Six well-known CIF (352×288) video sequences, *Akiyo*, *Coastguard*, *Foreman*, *Mobile*, and *Mother & Daughter* are tested. Several quantization parameters are tested, but only QPs 20, 28, 36, and 42 are listed because of the limited space. A default decision algorithm in the JM serves as the reference algorithm for comparison indicated as Full Search. The direction-detection (DD) algorithm [6] is also compared. The figures shown on the simulation results (Table I & Table II) are the average of 100 I-frames. The thresholds TH_{16v} and TH_{16v} in LL-Based I16 algorithm are 0.8 and 1.2, and TH_{4v} and TH_{4v} in Pixel-Based I4 algorithm are 0.7 and 1.2. Since MBs have been examined by LL-SD algorithm before coming to the execution of LL-Based I16 algorithm and/or Pixel-Based I4 algorithm, these thresholds are independent to the content of the video as well as QP value.

Tables I and II summarize the simulation results under different QPs. Table I is the average number of RDO computations needed in one MB. Table II summaries the precision rates of the DD[6] and FIPHTA comparing to the results of the Full Search. There are 592 RDO computations in one MB in the Full Search as mentioned before. In the DD algorithm, there are 2 selected modes for I16 and chroma 8×8, 6 selected modes for I4. Thus, it results a total of 196 (= 2·(2+6·16)) RDO computations for one MB. The Full Search and the DD algorithm have fixed number of the RDO computations for all video sequences and QPs. As the proposed FIPHTA, we adopt the FastCrIntraDecision option and separate the RDO operation in chroma and luma components. Hence, additional 2 RDO computations from chroma components have been added in the tables. To the number of RDO computations, the proposed FIPHTA is much less to that of the Full Search as well as the DD algorithm. This becomes more obvious when QP turns larger due to the LL-SD algorithm. As for the accuracy in mode selection comparing to the results of Full Search, FIPHTA outperforms the DD algorithm for most of cases. Those cases in which FIPHTA does not perform as good as DD occur on large QP values because our decision is more favorable to I16 prediction modes. Contrary to higher accuracy with higher QPs in DD, our proposed algorithm remains an accuracy of 70% or above for almost all cases.

TABLE I. THE AVERAGE NUMBER OF RDO COMPUTATION FOR ONE MB UNDER DIFFERENT QP AND VIDEOS

	Method \ *QP*	*20*	*28*	*36*	*42*
Full Search		592			
DD [6]		196			
FIPHTA	*Akiyo*	93.69	83.10	77.57	52.12
	Coastguard	102.07	101.53	96.64	72.67
	Foreman	114.51	122.53	118.14	94.67
	Mobile	134.79	136.46	134.32	151.20
	Mother	119.44	119.44	185.09	47.09
	Average	112.90	112.61	122.35	83.55

TABLE II. PRECISION RATE (COMPARING TO THE RESULTS OF THE FULL SEARCH OF THE H.264)

Sequence	Method \ *QP*	*20*	*28*	*36*	*42*	**Ave.**
Akiyo	DD [6]	0.48	0.61	0.68	**0.81**	0.65
	FIPHTA	**0.83**	**0.76**	**0.72**	0.71	**0.76**
Coastguard	DD [6]	0.62	0.69	**0.73**	**0.85**	0.72
	FIPHTA	**0.82**	**0.72**	0.67	0.74	**0.74**
Foreman	DD [6]	0.59	0.68	0.65	**0.78**	**0.68**
	FIPHTA	**0.68**	**0.72**	**0.71**	0.62	0.68
Mobile	DD [6]	0.42	0.48	0.48	0.69	0.52
	FIPHTA	**0.71**	**0.73**	**0.66**	**0.71**	**0.70**
Mother	DD [6]	0.71	0.72	0.74	**0.77**	0.74
	FIPHTA	**0.82**	**0.78**	**0.75**	0.71	**0.77**

V. CONCLUSION

This paper presented a fast intra prediction with Haar transform algorithm for H.264/AVC encoder. The LL-coefficients of a Haar transform of a MB are used for determining whether it is smooth or textured (LL-SD alg.). Next, an LL-based I16 algorithm and/or a Pixel-based I4 algorithm is applied to further reduce the number of candidate modes. The algorithm has been tested on various video sequences under different QPs, The complexity on RDO computation is greatly reduced with satisfying accuracy performance. In determining the dominant direction for candidate modes, our algorithm is emphasized in horizontal and vertical directions. How to improve the prediction on diagonal directions is our future work.

REFERENCES

[1] I. E.G. Richardson, "H.264 and MPEG-4 video compression: Video Coding for Next-generation", Wiley, December 2003.

[2] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)", in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003.

[3] C. G. Rafael, E. W. Richard, "Digital Image Processing," Prentice Hall, 2002.

[4] F. Pan, X. Lin, S. Rahardja, K.P. Lim, Z. G. Li, D. Wu, S. Wu, "Fast Mode Decision Algorithm for Intra Prediction in H.264/AVC Video Coding," IEEE Trans. on CAS for Video Technology, Vol. 15, Issue 7, July 2005, pp. 813 – 822.

[5] J.-F. Wang, J.-C. Wang, J.-T. Chen, A.-C. Tsai, A. Paul, "A Novel Fast Algorithm for Intra Mode Decision in H.264/AVC Encoders," IEEE Int. Symp. on Circuits and Systems 2006 (ISCAS 2006), May 2006, pp. 3498-3501.

[6] W.-G. Lin, A.-C. Tsai, J.-F. Wang, J.-F. Yang, "A Simple Direction Detection Algorithm for Fast H.264 Intra Prediction," TENCON 2007 - 2007 IEEE Region 10 Conf., 2007, pp.1 – 4.

[7] C.-C. Cheng, and T.-S. Chang, "Fast Three Step Intra Prediction Algorithm for 4x4 Blocks in H.264," IEEE Int'l Symp. on Circuits and Systems 2005 (ISCAS 2005), 2005, pp. 1509-1512.

[8] Y.-K. Lin, T.-S. Chang, "Fast Block Type Decision Algorithm for Intra Prediction in H.264 FRext," IEEE Int'l Conf. on Image Processing 2005 (ICIP 2005), 2005, pp. 585-588.

[9] H.264/AVC Reference Software JM12.3, http://iphome.hhi.de/suehring/tml/download/, Aug-2007.