TELKOMNIKA, Vol. 11, No. 10, October 2013, pp. 5703 ~ 5710
ISSN: 2302-4046

Fast Intra Prediction Mode Decision Algorithm for HEVC

Mengmeng Zhang*[1], Jianfeng Qu[1], Huihui Bai[2]
[1]College of Information Engineering, North China University of Technology, Beijing, China
[2]Institute of Information Science, Beijing Jiaotong University, Beijing, China
*Corresponding author, e-mail: muchmeng@126.com

Abstract

This paper proposes a novel fast intra-prediction algorithm that exploits the Sobel operator to replace the Hadamard transform used in the Rough Mode Decision (RMD) process of intra prediction in High Efficiency Video Coding (HEVC). First, the Sobel operator is used to calculate the vector direction of each pixel. A judgment is then made on which prediction mode the vector belongs to, and a histogram is applied in the scheme to generate the statistics of prediction modes for each prediction unit. Finally, the prediction mode candidates are placed in the candidate list for the rate-distortion optimization process. Experimental results show that our proposed algorithm for RMD significantly reduces the complexity of the encoder with an acceptable degradation of quality and BD-rate compared with HM7.0.

***Keywords**: High Efficiency Video Coding, Intra, Prediction Modes, Sobel*

1. Introduction

High Efficiency Video Coding (HEVC) [1] provides significantly better video coding efficiency than the last generation video coding, that is, Advance Video Coding (AVC). Under the condition of maintaining the same video quality, the goal of HEVC is to reduce bit-rate demand by 50% compared with AVC at the expense of increased computational complexity, which is in an accepted range.

Although HEVC still belongs to the block-based hybrid video coding framework, it provides a highly flexible hierarchy of unit representation [2], which includes three block concepts, namely, coding unit (CU), prediction unit (PU), and transform unit (TU). CU is a unit similar to the macroblock. CU can be split and is always square. The dimension of CU ranges from 8x8 up to the largest coding unit (LCU). The definition of CU allows itself to split into four equal-sized PU is a basic unit used for carrying information related to the prediction process. PU can only be used in CU. PU has two sizes that are supported in intra prediction, namely, 2N×2N and N×N. In addition to CU and PU, TU is a unit related to transformation and quantization and its size cannot exceed that of CU. Based on the recursive structure, the encoder must exhaust all combinations of CU, PU, and TU to determine an optimal solution. However, this process is time-consuming.

To improve the efficiency and accuracy of video coding, HEVC provides up to 34 prediction modes for intra prediction [3], which far exceeds the nine prediction modes of AVC. HEVC uses rate-distortion optimization (RDO) technique to decide the coding mode for a CU. Figure shows the RDO process. As can be seen from Figure, to choose the best coding mode for an CU, HEVC encoder calculates the rate-distortion cost (RD cost) of every possible mode (34 or 17 modes), after that HEVC chooses the mode which has the minimum value. This process is repeatedly carried out for all the possible modes for CU. However, the encoder cannot afford the total computation if all the 34 prediction modes pass through the RDO process. Therefore, release the computational burden of RDO in HEVC is far more demanding than any existing video coding algorithm. The Hadamard transform is used in HM7.0, and three or eight candidates are selected through a Rough Mode Decision (RMD) process to reduce the computation of the encoder. All prediction modes in the RMD process are tested using the minimum absolute sum of Hadamard Transformed coefficients of residual signal [4]. RDO is then applied to every candidate mode selected by the RMD process. However, the computation

Received March 29, 2013; Revised June 30, 2013; Accepted July 14, 2013

of the encoder process is still very huge despite the application of the Hadamard transform in the RMD process. Thus, reducing the number of candidates for RDO is necessary to minimize the computational burden of the encoder. For example, for this reason, a fast algorithm is proposed in this study to replace the Hadamard transform in HM 7.0. HM 7.0 made many optimizations on the foundation of other HEVC Test Models. Some of these optimizations on intra prediction will be discussed in section 2.

The remainder of this paper is organized as follows: Section 2 briefly describes intra prediction in HEVC. Section 3 introduces the algorithm and the principle of the fast intra prediction process. Section 4 presents the experimental results. Finally, the last section presents the conclusion.

2. Intra Prediction in HEVC

Figure 1 shows that the intra coding tool of HEVC provides up to 34 prediction modes, as well as the planar mode for the luma component of different PU sizes. In HM7.0, PU sizes of 4×4, 8×8, 16×16, 32×32, and 64×64, correspond to 18, 35, 35, 35, and 35 prediction modes. However, in previous visions, there are just 4 prediction modes for 64×64. This optimization is very important for the video which have large blocks, at this case LCU will be chosen as the best CU [5]. So this optimization can offer us better prediction modes, and lower the BD-rate.

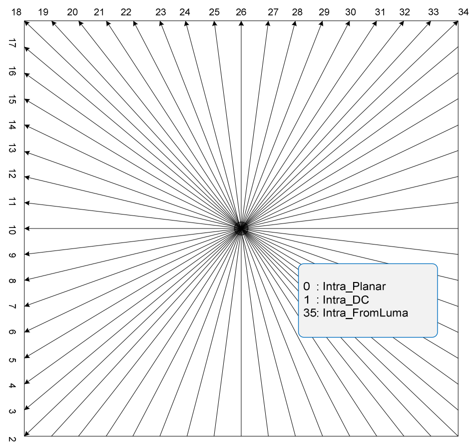

Figure 1. 35 prediction modes

Based on the 4×4 prediction unit illustrated in Figure 2, we can make a rough judgment from its luma texture. The prediction vector is the same as the arrow. The RMD process should find a way to select candidates from the 34 prediction modes and try its best to make the optimal mode is one of these candidates.

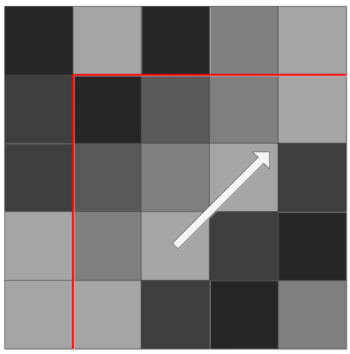

Figure 2. Illustration of a 4x4 PU

The following intra prediction process must be performed to select the best intra prediction mode for the luma component of each PU. The first step is an RMD process, which will make a rough decision on the choice of the best prediction mode. HM7.0 then applies the Hadamard transform. Afterward, the prediction cost Jpred,SATD is calculated for all possible prediction modes, and several prediction modes with lower costs are chosen as candidate modes. The different prediction unit sizes [6] of 4×4, 8×8, 16×16, 32×32, and 64×64 have 8, 8, 3, 3, and 3 candidate modes, as listed in Table 1. After that, a process of MPM is adopted. In previous editions, the MPM process has two candidates, however there are three candidates in HM7.0. What's more, the MPM process has changed tremendously. In HM7.0, the MPM process does not just simply push the intra directions of above and left PU into the candidate list. The MPM process of HM7.0 make the most use of the above and left prediction modes through a weighted coefficient when above and left prediction modes are equal. Finally, the selected modes are placed in the candidate list, which is used for the RDO process. The flowchart of RDO is shown in Figure 3.

Table 1 Number of prediction modes after the process of RMD

PU size	Number of prediction modes
4x4	8
8x8	8
16x16	3
32x32	3
64x64	3

Since the HM7.0 has three MPM [7] which is more than other editions, the computational burden is more than other editions. Release the burden of the encoder is the key point of this proposed algorithm.

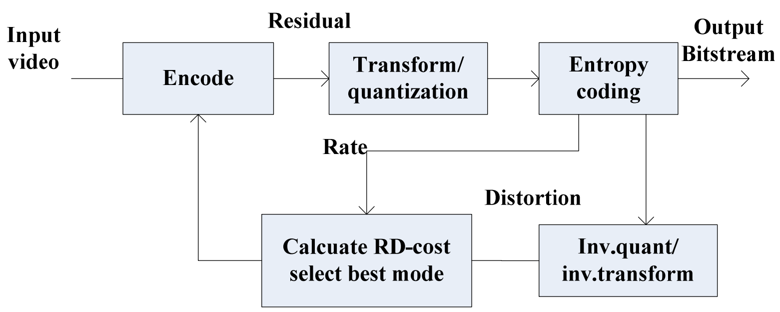

Figure 3. The process of RDO

3. Fast Intra Prediction Algorithm

Our proposed algorithm introduces a new method for the RMD process, which intends to reduce the number of candidates and the computation load. The flowchart of the algorithm is shown in Figure 4.

The core of this algorithm utilize the texture of a CU to make out which prediction mode is most probably be the best mode. We found that the pixels (both luma and chroma) along the direction of local edge nearly have the similar values. Therefore by predict the pixels using those neighboring pixels in the same direction of the edge, a prediction mode can be got. After that an edge map which represents the vector of local edge is created, and a local edge direction histogram is then established for each CU.

Since texture is the core of edge detection, the proposed algorithm will use the Sobel operator as the core algorithm for its excellent performance when used in edge detection [8].

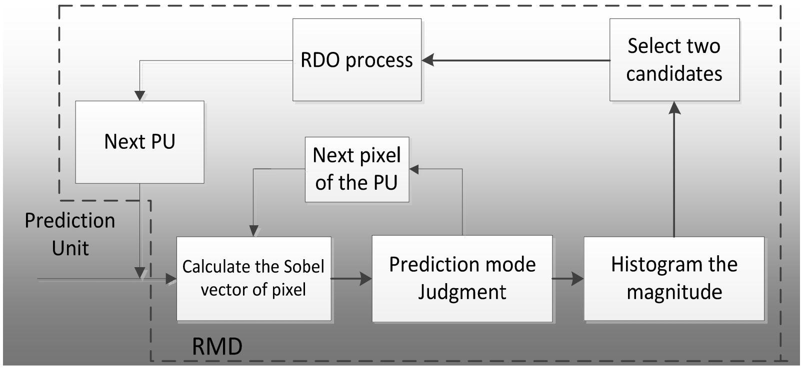

Figure 4. Algorithm flowchart

The formula of Sobel operator is given by (1), and matrix A is a 3×3 pixel unit.

$$G_x=\begin{bmatrix}-1 & 0 & +1\\ -2 & 0 & +2\\ -1 & 0 & +1\end{bmatrix}*A\ ,G_y=\begin{bmatrix}+1 & +2 & +1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix}*A,\ A=\begin{bmatrix}p_{i-1,j-1} & p_{i-1,j} & p_{i-1,j+1}\\ p_{i,j-1} & p_{i,j} & p_{i,j+1}\\ p_{i+1,j-1} & p_{i+1,j} & p_{i+1,j+1}\end{bmatrix} \quad (1)$$

Two convolution kernels exist for the Sobel operator, and each kernel is related to the degree of difference between the horizontal and vertical directions [9]. The corresponding direction vector $\{dx_{i,j},dy_{i,j}\}$ for a luma pixel in a picture is defined as:

$$dx_{i,j}=p_{i-1,j+1}+2\times p_{i,j+1}+p_{i+1,j+1}-p_{i-1,j-1}-p_{i,j-1}-p_{i+1,j-1} \quad (2)$$

$$dy_{i,j}=p_{i+1,j-1}+2\times p_{i+1,j}+p_{i+1,j+1}-p_{i-1,j-1}-p_{i-1,j}-p_{i-1,j+1} \quad (3)$$

The degree of difference on the horizontal and vertical directions are correspondingly represented by $dx_{i,j}$ and $dy_{i,j}$. The magnitude of the vector is accurately presented by (4). In addition, (5) is adopted when the computation load is considered.

$$Amp(\vec{D}_{i,j})=\sqrt{(dx_{i,j})^2+(dy_{i,j})^2} \quad (4)$$

$$Amp(\vec{D}_{i,j})'=\left|dx_{i,j}\right|+\left|dy_{i,j}\right| \quad (5)$$

The angle of vector is presented by (6). In fact, (7) can be used to present the angle of the vector because it is a simpler threshold technique for building up the edge direction histogram.

$$Ang(\vec{D})=\frac{180^o}{\pi}\times\arctan(\frac{dy_{i,j}}{dx_{i,j}}) \quad (6)$$

$$Ang(\vec{D})'=\frac{dy_{i,j}}{dx_{i,j}} \quad (7)$$

After obtaining the vector in a coordinate system, Figure 5 shows how to make the decision on which prediction mode the vector belongs to. The region boundary of the two adjacent prediction mode vectors is the bisectrix of the two vectors. So if a vector is in the area between one prediction mode and a bisectrix, it will be judged to belongs to the prediction mode. Taking Figure 5 as an example, the vector we got is in the area between the prediction mode 3 and the Boundary line of the judgment, So the vector belongs to mode 3, and (5) is the corresponding magnitude.

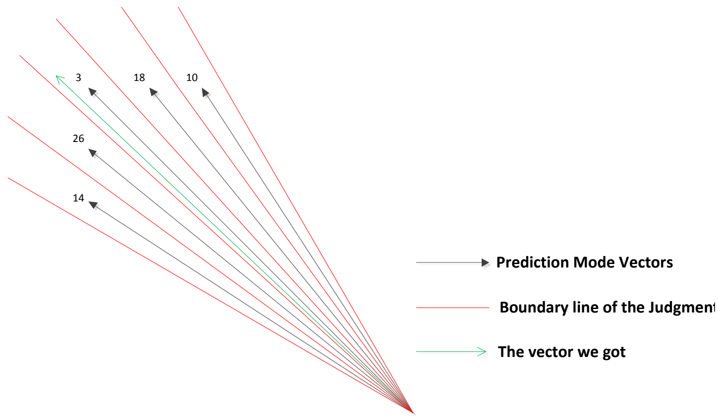

Figure 5. Illustration of the boundary line for prediction mode judgment

The first step is calculate the vector of every pixel in a LCU which will be operated, and put the data of these vector into two array(one array stores magnitudes, another stores angles).

After that the process goes into a recursive structure, in this structure, PU with the size of 64×64, 32×32, 16×16, 8×8, 4×4 will get the vector of every pixel in each PU from the arrays which we got in the first step, and make the judgment which prediction mode the vector belongs to.

The next step is creating a histogram. Figure 6 shows that a histogram was used to count the sum of the magnitude of each prediction mode in a PU. The X-axis contains the prediction modes, whereas the Y-axis contains the sum of the magnitude of each prediction mode in a PU. The highest and the second highest prediction modes in the histogram are then chosen (Taking both BD-rate and coding time into consideration, two candidates are a better choice than one or three candidates) as the candidates for RDO. Though it is not mathematically corresponding to plane prediction to any directional edge, we can for sure associate the prediction to its respective edges. Therefore, it is fairly reasonable for us to try plane prediction if it is not obviously a DC prediction. In Figure 5, as an example, prediction modes 6 and 9 are chosen as the candidate modes. Finally, the two modes are placed in the candidate list.

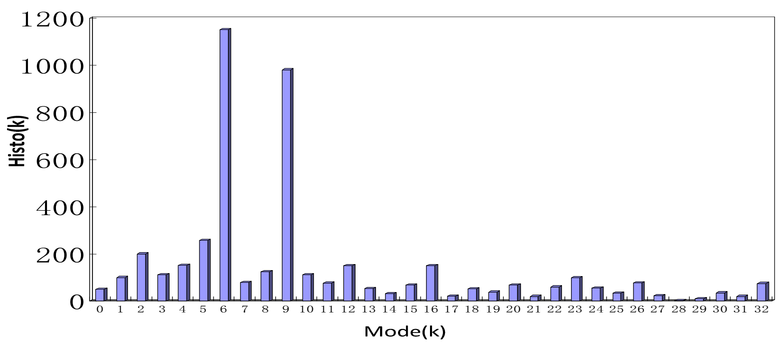

Figure 6. Edge direction histogram

4. Experimental Results

In this experiment, 300 frames of each sequence are coded to test the performance of proposed algorithm, and every frame is intra coded. A computer which has 2.8GHz core is used to do this experiment.

The experimental results [10] of this study are presented in Tables 2, which show that the proposed algorithm saves 16.5% on average coding time and the BD-rate increases by approximately 1.122% in high efficiency (HE) test conditions, respectively. The results in Classes A and B are better than those in other test sequences. Therefore, the proposed algorithm performs better under a high graphics condition.

To calculate the time saving of the fast intra-prediction algorithm, the following calculation is defined to evaluate the time differences. We take T_{HEVC} present the coding time used by HM7.0 encoder, correspondingly T_{pro} be the time taken by the faster intraprediction algorithm, and is defined as:

$$\Delta \mathrm{Time} = \frac{T_{HEVC} - T_{Pro}}{T_{HEVC}} \times 100\% \tag{8}$$

Table 2. Result in HE test condition

	All-Intra HE			
	Y BD-rate	U BD-rate	V BD-rate	∆Time
Class A	0.73%	0.0	0.3%	-15.7%
Class B	0.4%	0.0%	-0.15%	-16.9%
Class C	1.245%	0.45%	0.775%	-16.7%
Class D	2.025%	1.125%	0.575%	-16.8%
Class E	1.21%	1.1%	0.8%	-16.1%
All	1.122%	0.535%	0.46%	-16.44%

The Rate-Distortion (RD) curves of ParkScene and RaceHorses are shown in Figures 7, 8. The curve of the proposed algorithm is very close to the curve of HM7.0. In addition, the efficiency of ParkScene is better than that of RaceHorses.

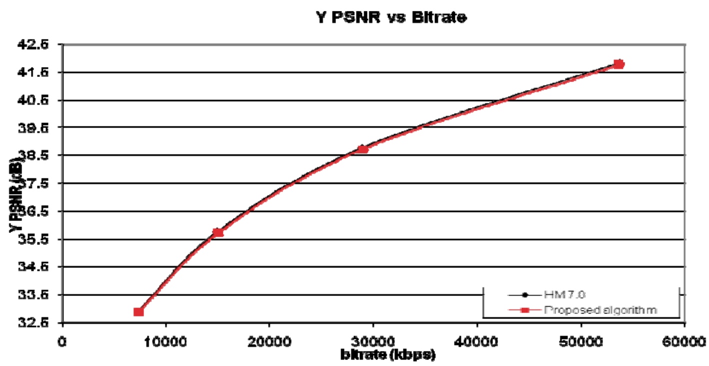

Figure 7. RD Curves of ParkScene (Class B 1920×1080HE)

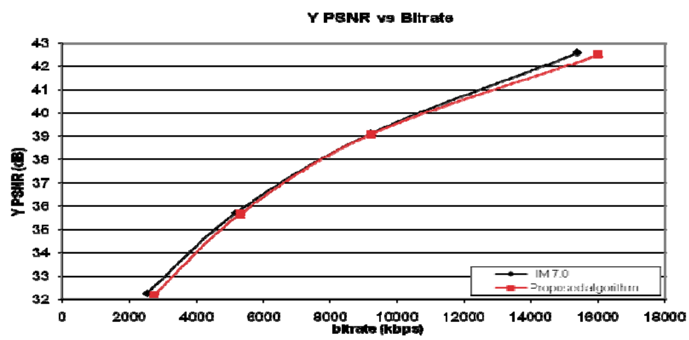

Figure 8. RD Curves of RaceHorses_832×480_30 (Class C 832×480 HE)

Figure 9 shows that the reconstructed image quality difference between HM7.0 and the proposed algorithm is almost negligible under the HE test condition.

(a) Reconstructed by HM7.0(HE)

(c) Reconstructed by proposed algorithm(HE)

Figure 9. The difference between HM4.0 and proposed algorithm (RaceHorses)

5. Conclusion

This paper proposed a fast intra prediction mode decision algorithm for HEVC, which was achieved using the Sobel operator and by reducing the number of prediction mode candidates for RDO. The experimental results in Section 4 show that the proposed algorithm can reduce the complexity of the encoder and that the video quality loss is almost negligible compared with HM7.0. In addition, the proposed algorithm achieved up to 19.8% reduction in coding time with negligible degradation of quality and BD-rate.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. 61103113, No. 61272051), Jiangsu Provincial Natural Science Foundation (BK2011455) and

Beijing Municipal Education Commission Science and Technology Development Program (KM201310009004).

References

[1] Kemal Ugur, Kenneth Andersson. High Performance, Low Complexity Video Coding and the Emerging HEVC Standard. *IEEE Transactions on Circuits and Systems for Video Technology*. 2010; 20(12): 1688-1697.
[2] Woo-Jin Han. Improved Video Compression Efficiency Through Flexible Unit Representation and Corresponding Extension of Coding Tools. *IEEE Transactions on Circuits and Systems for Video Technolog*. 2010; 20(12): 1709-1720.
[3] Liang Zhao. Fast Mode Decision Algorithm for Intra Prediction in HEVC. *Visual Communications and Image Processing (VCIP).* 2011; 1-4.
[4] Frank Bossen, Virginie Drugeon. Video Coding Using a Simplified Block Structure and Advanced Coding Techniques. *IEEE Transactions on Circuits and Systems for Video Technology*. 2010; 2(12): 1667-1675.
[5] Van Wallendael G. Improved intra mode signaling for HEVC. *Multimedia and Expo (ICME)*. 2011; 1-6.
[6] Benjamin Bross. High efficiency video coding (HEVC) text specification draft 7. *JCT-VC 5th Meeting*. 2011; 56-71.
[7] Il-Koo Kim. HM7: High Efficiency Video Coding (HEVC) Test Model 7 Encoder Description. *JCT-VC 9th Meeting*. 2012; 31-46.
[8] Anil K Jain, Aditya Vailaya. Image retrieval using color and shape. *Pattern Recognit*. 29(8): 1233-1244.
[9] Feng Pan. Fast Mode Decision Algorithm for Intraprediction in H.264/AVC Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology*. 2005; 15(7): 813-822.
[10] Frank Bossen. Common test conditions and software reference configurations. *JCT-VC 6th Meeting*. 2011; 14-22.