HADAMARD COST–BASED FAST CU DEPTH DECISION ALGORITHM FOR HEVC INTRA CODING

Yu-Ru Tsai (蔡育儒), Zong-Yi Chen (陳宗毅), and Pao-Chi Chang (張寶基)

Dept. of Communication Engineering,
National Central University, Taiwan
E-mail: {yrcai, zychen}@vaplab.ce.ncu.edu.tw, pcchang@ce.ncu.edu.tw

ABSTRACT

The quadtree coding unit structure is adopted in High Efficiency Video Coding to achieve high coding efficiency. Up to 35 intra prediction modes are available for each prediction unit for accurate predictions. However, the significantly increased encoding complexity due to the advanced encoding structure can not be neglected. A Hadamard cost–based fast intra CU depth decision is proposed in this paper to reduce the computational complexity. The Hadamard cost of each mode is calculated in rough mode decision to preselect a few candidate modes before full rate-distortion optimization. The proposed algorithm utilized the minimum Hadamard cost to develop the criterion of early CU splitting and termination. The threshold was modeled by the sequence characteristic parameters and QPs. Experimental results showed that the proposed algorithm saves at most 51.57% and on average 43.46% encoding time compared with HM 15.0 all-intra coding.

***Keywords** HEVC; All intra; CU; Fast algorithm; Rough Mode Decision; Hadamard transform;*

1. INTRODUCTION

To meet the increasing demands for high quality and high resolution video which has become the dominant form of multimedia, the new generation video coding standard, High Efficiency Video Coding (HEVC), was established by the Joint Collaborative Team on Video Coding (JCT-VC) in 2013. HEVC can save half the bit rate of H.264/AVC [1] under the same subjective video quality.

The coding unit (CU), which is similar to the macroblock (MB) in H.264/AVC, is adopted in HEVC. Being the basic encoding unit, a CU contains prediction units (PUs) and transform units (TUs). The quadtree structure provides variable CU sizes ranging from 64×64, 32×32, 16×16 to 8×8 which are denoted as depth 0, 1, 2, and 3 respectively. Figure 1 shows the architecture of quadtree–based CU, and this kind of structure provides high flexibility to represent complex and homogenous areas.

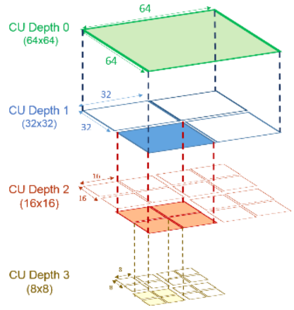

Fig. 1: Quadtree CU structure.

The PU is used for prediction. Intra prediction exploits the spatial correlation within a frame. Compared with 9 intra prediction modes in H.264/AVC, there are 35 candidate modes in HEVC for finer predictions. Figure 2 shows the comparison of intra modes in H.264/AVC and HEVC.

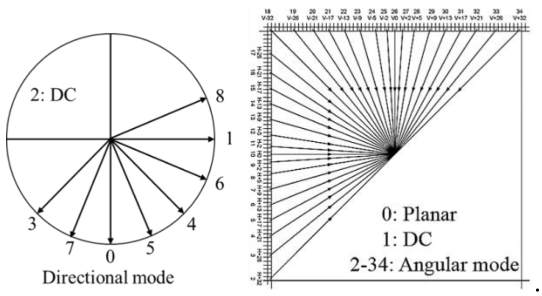

Fig. 2: Comparison of intra prediction modes in H.264/AVC and HEVC.

However, the rate-distortion optimization (RDO) process examines all combinations of CU depths and prediction modes in HEVC encoder to decide the optimal partition. The coding efficiency is thus substantially improved at the cost of heavy computation burden.

To reduce the complexity of intra coding, a fast intra mode decision has been proposed and adopted in the reference software of HEVC (HM) [2][3]. N candidate modes are preselected among all 35 prediction modes by the rough mode decision (RMD) process, and N is set to {8, 8, 3, 3, 3} for each PU sizes (from 4×4 to 64×64). In RMD, the rate-distortion cost is evaluated by the following function,

$$J_{pred}^{H} = HSAD + \lambda_{pred} \times R_{pred} \quad (1)$$

, where $HSAD$ represents the sum of absolute difference of Hadamard transformed residual coefficients of a PU and R_{pred} represents the required bits for the prediction mode. Then, full RDO process is performed on the N RMD candidate modes and the most probable modes (MPM) from the neighboring coded blocks.

Numbers of fast algorithms [4]-[7] have been proposed to further reduce the computational complexity of the HEVC intra coding. [4] utilized the variance of input image to decide whether the current CU would be early terminated. By Sobel edge detector, [5] calculated the edge density of encoding area to classify the set of depth levels and reduce the computation burden for encoder. Since the CU depth level is highly content-dependent, the possible depth range of current CU was predicted using spatially nearby CUs in [6]. Hence, RDO process was only applied at those predicted depths to reduce the encoding time. The fast CU size decision method for intra coding in [7] was based on the proposed global and local edge complexity. Combining the edge complexity of the CU and its four sub-CUs, the CU was then determined to be split, non-split, or undetermined.

Most previous works utilized the information from the pixel domain, e.g. variance, gradient and edge, to develop the criteria of CU early determination. The calculation for features cost extra computations. Some works made decision by the depths and prediction modes of nearby coded CUs. The current prediction result which is depended on previous prediction may cause error propagation. By contrast, the coding information of current CU during encoding would be a good choice for developing fast CU depth decision algorithm. The minimum Hadamard cost in RMD was utilized in this study with no computation overhead.

The remainder of this paper is organized as follows. In section 2, the proposed Hadamard cost–based fast CU depth decision is introduced. Section 3 shows the experimental results and section 4 concludes this paper.

2. PROPOSED FAST CU DEPTH DECISION

In this section, the minimum Hadamard cost used for predicting whether the current CU should be split or not is introduced first. Then the threshold for decision is modeled by image properties and QP.

2.1. Observation and analysis

Being the pre-process of the full RDO, RMD reduces the candidate of intra prediction modes by calculating the Hadamard cost of each mode. Then, the first few modes with smaller Hadamard costs are tested with full RDO. The minimum Hadamard cost was utilized in this study as follows,

$$J_{min}^{H} = min\{ J_{pred,K}^{H} \} \quad (2)$$

, where K represented the number of intra prediction mode and ranged from 0 to 34.

First, the statistical properties of the J_{min}^{H} in intra prediction was analyzed. Figure 3 shows the distribution of the J_{min}^{H} in depth 2 and Table 1 shows the statistical results of the mean values of J_{min}^{H} for non-splitting and splitting CUs.

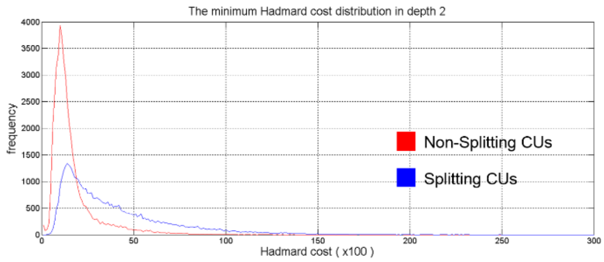

Fig. 3: The distribution of J_{min}^{H} in depth 2.

Table 1: Mean of J_{min}^{H} for non-splitting and splitting CUs.

QP 22	*Depth 0*	*Depth 1*	*Depth 2*
Non-splitting	28266	6959	1548
Splitting	61656	15101	3822
QP 27	*Depth 0*	*Depth 1*	*Depth 2*
Non-splitting	24417	6981	1647
Splitting	62620	15957	4276
QP 32	*Depth 0*	*Depth 1*	*Depth 2*
Non-splitting	23136	7146	1775
Splitting	64098	16574	4704
QP 37	*Depth 0*	*Depth 1*	*Depth 2*
Non-splitting	24076	7491	1971
Splitting	65882	17667	5379

From Fig. 3 and Table 1, CUs with large cost are likely to split. The J_{min}^{H} values between non-splitting and splitting CUs are distinct. Moreover, J_{min}^{H} varies with QPs and depths. Therefore, J_{min}^{H} can be used for CU partition strategy.

2.2. Early splitting and early termination

According to different J_{min}^{H} values, two CU partition strategies were applied: early splitting (ES) and early termination (ET). A threshold (TH_{ES}) was set at each depth to early split current CU with large costs. On the contrary, a threshold (TH_{ET}) was set to early terminate current CU with small J_{min}^{H}. To avoid serious error decision, a range between two thresholds was reserved without fast decision. Figure 4 shows the relationship of the J_{min}^{H} value and two CU partition strategies.

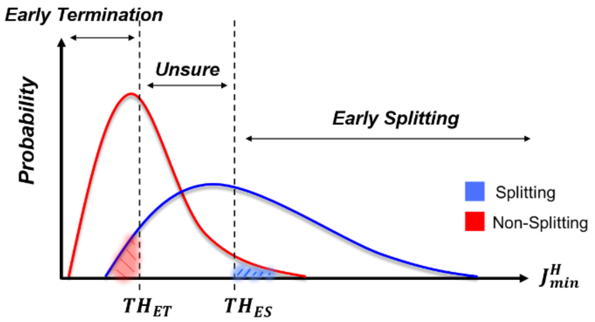

Fig. 4: Early splitting and early termination.

The CU depth splitting is decided on the basis of Table 2 to skip RDO process in some conditions to accelerate the HEVC encoding.

Table 2: Proposed fast CU depth decision.

Condition	*Operation*
$J_{min}^{H} \leq TH_{ET}$	Do RDO at current depth & no splitting. (ET)
$TH_{ET} < J_{min}^{H} < TH_{ES}$	Do RDO & split to next depth. (Unsure)
$J_{min}^{H} \geq TH_{ES}$	Split to next depth without RDO at current depth. (ES)

2.3. Threshold selection strategy

The threshold selection is crucial for CU depth decision, and selecting the threshold by making trade-offs between coding performance and time saving is intuitive and reasonable. In this study, the threshold for J_{min}^{H} at training stage was selected by adjusting the error rate caused by ET/ES to achieve good trade-off. Then, the threshold was modeled using training data.

When ET is performed, error decision occurs when originally splitting CUs (S_{HM}) are not split ($NS_{Proposed}$). Oppositely, error decision occurs when originally non-splitting CUs (NS_{HM}) are split ($S_{Proposed}$) in ES. Error rates caused by ET and ES are denoted as E_{ET} and E_{ES} and defined in (3) and (4) respectively. Figure 5 illustrates the error decision caused by ET and ES. The error rates vary with the thresholds. The next step is to select proper error rate at each depth.

$$E_{ET}(\%) = \frac{NS_{Proposed}}{S_{HM}} \tag{3}$$

$$E_{ES}(\%) = \frac{S_{Proposed}}{NS_{HM}} \tag{4}$$

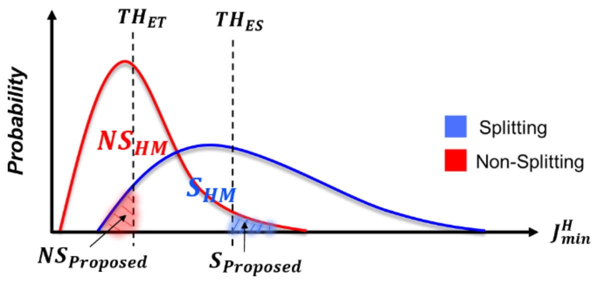

Fig. 5: Error decision caused by ET/ES.

Figure 6 is an example of the percentage of splitting and non-splitting CUs at each depth for two classes of HEVC test sequences encoded with HM15.0 all-intra configuration at QP 32. From Fig. 6, the percentage of splitting CU is decreasing with depth. Most CUs at depth 0 is split. ES is preferable at depth 0. Hence higher E_{ES} is tolerable to bring about more splitting CUs. To avoid considerably different decision of CU depth between HM and the proposed method, e.g. optimal CU depth 3 is terminated at depth 0, E_{ET} must be kept low. However, more CUs at depth 2 are not split. ET is preferable at depth 2. Higher E_{ET} is tolerable but E_{ES} must be kept low. Therefore, the error rates caused by fast decision were limited as Table 3. Six HEVC test sequences were chosen as training materials. Proper thresholds for each training sequence were selected on the basis of Table 3 from extensive experiments at fixed QP 32. Then, thresholds can be modeled by the training data.

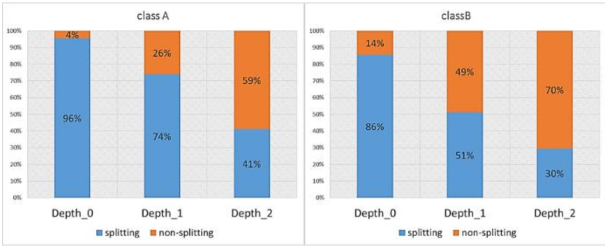

Fig. 6: Percentage of splitting and non-splitting CUs.

Table 3: Tolerable error rate at each depth for threshold selection.

	Depth 0	*Depth 1*	*Depth 2*
E_{ET} (%)	1~3	3~5	10~15
E_{ES} (%)	10~15	3~5	1~3

2.4. Threshold modeling

The thresholds are considerable different among training sequences. Videos with different contents result in different J_{min}^{H} values. The video characteristics must be considered in threshold calculation. Moreover, from Table 1, J_{min}^{H} values vary with QPs and CU depths. Hence, the proposed threshold model aims to be adaptive to both video contents and QPs at each depth.

Different videos exist different image properties. Two parameters, image complexity (C) and image gradient (G), are used to represent the image properties in this study and are calculated using (5) and (6) respectively.

$$C = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \left| Y_{i,j} - Y_{mean} \right| \quad (5)$$

$$G = \frac{1}{(W-1) \times (H-1)} (G_{Hor} + G_{Ver}) \quad (6)$$

$$G_{Hor} = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \left| Y_{i,j} - Y_{i+1,j} \right| \quad (7)$$

$$G_{Ver} = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} \left| Y_{i,j} - Y_{i,j+1} \right| \quad (8)$$

, where W and H are the width and height of the video respectively, $Y_{i,j}$ is the pixel value at (i, j) and Y_{mean} is the mean of all pixels in current frame.

Using the thresholds selected in section 2.3, the threshold, $TH(C, G, QP)$, is modeled by linear function as (9).

$$TH(C, G, 32) = \alpha + \beta \times C + \gamma \times G. \quad (9)$$

By curve fitting, six sets of parameters (α, β, and γ) for TH_{ES} and TH_{ET} at each depth can be calculated. However, these parameters are only applicable to QP 32. The coding performance may degrade when the QP is not equal to 32. To improve the coding performance, investigations for the relationship between average J_{min}^{H} and QPs are conducted and Fig. 7 exhibits the results. The relationship can be approximated by a linear function as (10).

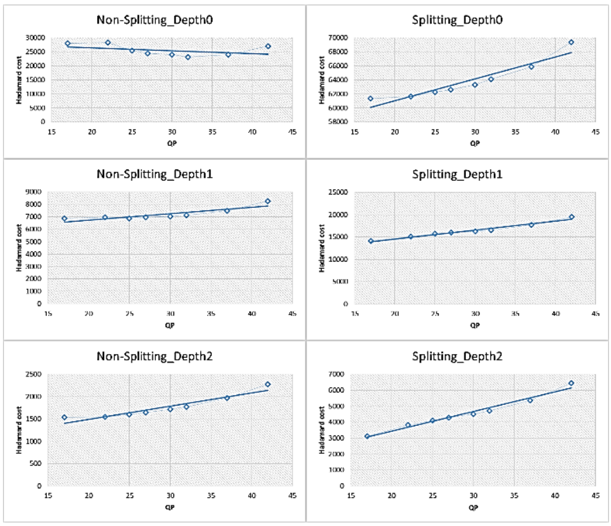

Fig. 7: The relationship between average J_{min}^{H} and QP.

$$J_{min,2}^{H} = J_{min,1}^{H} + \frac{\Delta J^{H}}{\Delta QP} \times (QP_2 - QP_1) \quad (10)$$

, where $J_{min,1}^{H}$ and $J_{min,2}^{H}$ are the average J_{min}^{H} at QP_1 and QP_2 respectively. And $\frac{\Delta J^{H}}{\Delta QP}$ denotes the change in average J_{min}^{H} over QP.

Combining image property and QP, the proposed adaptive thresholds are calculated as (11) and (12).

$$TH_{ES,\,Depth}(C,\, G,\, QP) = TH_{ES,Depth}(C, G, 32) + \frac{\Delta J_{ES,\,Depth}^{H}}{\Delta QP} \times (QP - 32) \quad (11)$$

$$TH_{ET,\,Depth}(C,\, G,\, QP) = TH_{ET,Depth}(C, G, 32) + \frac{\Delta J_{ET,\,Depth}^{H}}{\Delta QP} \times (QP - 32) \quad (12)$$

Image property parameters are calculated at each frame level in this study. On the other hand, the threshold is updated on the basis of current video content. The proposed threshold model can be adaptive to image properties and QPs.

Figure 8 is the flowchart of the proposed algorithm. The encoding process starts with LCU (depth 0) and selects the minimum Hadamard cost from RMD process to determine the CU splitting or not. Threshold is updated in each frame.

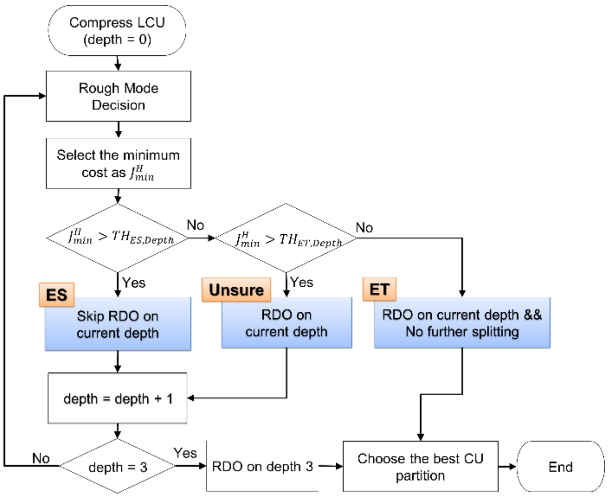

Fig. 8: The flowchart of the proposed Hadamard cost–based fast intra CU depth decision for HEVC.

3. EXPERIMENTAL RESULTS

The proposed fast CU decision was implemented using HEVC reference software (HM 15.0) with all-intra configuration. The test platform was a PC with an Intel(R) Core 2 X9650 @ 3.00-GHz CPU, 8-G RAM, and Windows 7 professional operating system.

Table 4 and 5 show the experimental results of the first 10 frames of training and testing sequences by the proposed fast CU depth decision evaluated with QPs 22, 27, 32, and 37. Coding efficiency was measured in terms of BD-rate (BDBR) (%) [8] and ΔT (%) represented the encoding time saving in percentage compared with HM 15.0.

$$\Delta T\ (\%) = \frac{1}{4}\sum_{i=1}^{QP_i} \frac{Enc.Time_{HM\ 15.0}^{QP} - Enc.Time_{prop.}^{QP}}{Enc.Time_{HM\ 15.0}^{QP}} \times 100 \quad (13)$$

Table 4: Simulation results for training sequences.

Training	*Nishikori [4]*		*Proposed*	
Sequence	BDBR	ΔT	BDBR	ΔT
Traffic	1.42	38.25	0.74	38.74
Kimono	0.96	54.62	0.44	33.81
Cactus	1.80	39.13	0.32	37.32
BasketballDrill	4.81	44.31	0.58	38.19
BasketballPass	2.45	41.44	0.89	47.29
FourPeople	1.66	37.12	0.96	45.52
Average	**2.18**	**42.48**	**0.66**	**40.15**

Table 5: Simulation results for testing sequences.

Testing	*Nishikori [4]*		*Proposed*	
Sequence	BDBR	ΔT	BDBR	ΔT
PeopleOnStreet	0.71	31.55	0.70	38.16
ParkScene	1.49	43.00	0.75	40.15
BasketballDrive	2.50	60.42	0.20	47.46
BQTerrace	0.92	33.98	1.08	47.94
PartyScene	0.17	10.67	0.73	41.57
BQMall	0.27	23.00	1.29	47.20
RaceHorsesC	0.55	12.43	1.14	45.56
BQSquare	0.18	14.62	1.41	42.94
BlowingBubbles	0.44	18.81	0.24	37.05
RaceHorses	0.52	10.07	0.67	33.65
KristenAndSara	1.73	55.19	0.47	51.57
Johnny	3.37	62.29	0.67	48.27
Average	**1.07**	**31.34**	**0.78**	**43.46**

The experimental results show that the proposed fast CU depth decision can provide 43.46% time saving on average with only 0.78% BD-rate loss. The proposed algorithm provides better BDBR in the sequence with strong motion exists in some parts of the frame, e.g. BasketballDrive and BlowingBubbles. In such sequences, the J_{min}^{H} values between large size CUs and small size CUs are distinct. Hence the decision of the proposed algorithm is accurate and both BDBR and time saving performance are great.

For sequences with complex texture uniformly distributed over the frame, e.g. BQTerrace, BQMall and BQSquare, the J_{min}^{H} value of each CU is close. More error decisions are made by the proposed algorithm. Hence the BDBR in such sequence is higher than others but the time saving is substantial compared with [4].

4. CONCLUSION

A fast intra CU depth decision is proposed on the basis of Hadamard cost. In the proposed algorithm, adaptive threshold for each depth in the situations of ET and ES is derived. By limiting the error rate caused by ET and ES, proper thresholds are selected for model training. The proposed threshold is modeled by image properties (complexity and gradient) and QP. Therefore, the proposed CU depth decision can be adaptive to different video contents and QPs. The experimental results show that the proposed fast intra CU depth decision can achieve significant time saving: 43.46% on average with 51.57% at most, and only negligible BDBR loss compared with HM 15.0.

5. REFERENCES

[1] *Advanced Video Coding*, ISO/IEC 14496-10, ITU-T Rec. H.264, Version 13, Mar. 2011.

[2] J. Lainema, F. Bossen, W. Han, J. Min, and K. Ugur, “Intra coding of the HEVC standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792-1801, Dec. 2010.

[3] L. Zhao, L. Zhang, M. Siwei, and D. Zhao, “Fast mode decision algorithm for intra prediction in HEVC,” *2011 IEEE Visual Communications and Image Processing (VCIP)*, Nov. 2011, pp. 1-4.

[4] T. Nishikori, T. Nakamura, T. Yoshitome, and K. Mishiba, “A fast CU decision using image variance in HEVC intra coding,” *IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, Kuching, Malaysia, Sep. 2013, pp. 52-56.

[5] Y. C. Lin and J. C. Lai, “Edge density early termination algorithm for HEVC coding tree block,” *2014 International Symposium on Computer, Consumer and Control (IS3C)*, Taichung, Taiwan, Jun. 2014, pp. 39-42.

[6] L. Shen, Z. Zhang, and P. An, “Fast CU size decision and mode decision algorithm for HEVC intra coding,” *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 207-213, Feb. 2013.

[7] B. Min and R. C .C. Cheung “A fast CU size decision algorithm for HEVC intra encoder,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 5, pp. 892-896, May 2015.

[8] G. Bjontegaard, “Calculation of average PSNR difference between RD-curves,” *ITU-T Q.6/SG16 VCEG 13th Meeting*, Document VCEG-M33, 2001.