# A Fast Mode Determination Algorithm Using Spatiotemporal and Depth Information for High Efficiency Video Coding (HEVC)

Kyung-Soon Jang, Jong-Hyeok Lee
Department Of Computer Engineering
SunMoon University
A-san, Rep. of Korea,
{ks.jang, jh.Lee}@mpcl.sunmoon.ac.kr

Chan-seob Park, Byung-Gyu Kim
Department Of Computer Engineering
SunMoon University
A-san, Rep. of Korea,
{cs.Park, bg.kim}@mpcl.sunmoon.ac.kr

*Abstract*— **High Efficiency Video Coding (HEVC) is a successor to the H.264/AVC standard and is the newest video coding standard using a quad-tree structure with three block types of a coding unit (CU), a prediction unit (PU), and a transform unit (TU). HEVC uses all possible depth levels to determine the lowest RD-cost block. Thus, HEVC is more computationally complex than the previous H.264/AVC standard. To overcome this problem, an early skip and merge mode detection algorithm is proposed using spatiotemporal and depth information. Experimental results show that the proposed algorithm can achieve an approximate 40% time saving with a random-access profile while maintaining comparable rate-distortion performance, compared with HM 12.0 reference software.**

***Keywords-High Efficiency Video Coding (HEVC); Early Skip Mode Detection; Inter Prediction; Depth; Coding Unit (CU).***

## I. INTRODUCTION

High Efficient Video Coding (HEVC) is the latest international video coding standard issued by the Joint Collaborative Team on Video Coding (JCT-VC) [1], which is a partnership between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), two prominent international organizations that specify video coding standards [2].

Increasing demands for high quality full high definition (Full HD), ultra high definition (UHD), and higher resolution video necessitate bitrate savings for broadcasting and video streaming. HEVC aims to achieve a 50% bitrate reduction, compared with the previous H.264/AVC standard, while maintaining quality.

HEVC is based on a coding unit (CU), a prediction unit (PU), and a transform unit (TU). The CU is a basic coding unit analogous to the concept of macroblocks in H.264/AVC. However, a coding tree unit (CTU) is the largest CU size that can be partitioned into 4 sub-CUs of sizes from 64x64 to 8x8. Figure 1 shows an example of the CU partitioning structure. This flexible quad-tree structure leads to improved rate-distortion performance and also HEVC features for advanced content adaptability. The PU is the basic unit of inter/intra-prediction containing large blocks composed of smaller blocks of different symmetric shapes, such as square, and rectangular, and asymmetric. The TU is the basic unit of transformation defined independently from the PU, but whose size is limited to the CU, to which the TU belongs. Separation of the block structure into three different concepts allows optimization according to role, which results in improved coding efficiency [3], [4]. However, these advanced tools cause an extremely high computational complexity. Therefore, a decrease in the computational complexity is desired.

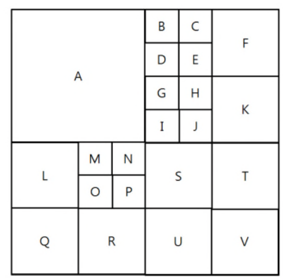


Figure 1. An example of the CU partitioning structure.

Previous work has focused on reduction of the computational complexity. Pai-Tse *et al.* [6] proposed a fast zero block detection algorithm based on SAD values using inter-prediction results. Features of the proposed algorithm were applied to different HEVC transform sizes. A 48% time saving for quantization parameter (QP) = 32 was achieved. Zhaoqing Pan *et al.* [7] proposed a fast CTU depth decision algorithm using the best quad-tree depth of spatial and temporal neighboring CTUs, relative to the current CTU, for an early quad-tree depth 0 decision. Correlations between the PU mode and the best CTU depth selection were also used for a depth 3 skipped decision. A 38% time reduction for all QPs was achieved under common testing conditions. Hassan Kibeya *et al.* [8] proposed a fast CU decision algorithm for the block structure encoding process. Based on early detection of zero quantized blocks, the number of CU partitions to be searched was reduced. Therefore, a significant reduction of encoder complexity was achieved and the proposed algorithm had almost no loss of bitrate or peak signal to noise ratio (PSNR), compared with HM 10.0 reference software.

In this study, an early skip and merge mode detection algorithm is proposed using neighboring block and depth

TABLE I. CONDITIONAL PROBABILITY BETWEEN CURRENT CTU AND NEIGHBORING CTUS.

| | | Spatial CTUs (%) | | | | Temporal CTUs (%) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *P(O\|A)* | *P(O\|B)* | *P(O\|C)* | *P(O\|D)* | *P(O\|E)* | *P(O\|F)* | *P(O\|G)* | *P(O\|H)* | *P(O\|I)* | *P(O\|J)* | *P(O\|K)* | *P(O\|L)* | *P(O\|M)* |
| **Class B** | ***SKIP*** | 70.4 | 69.1 | 70.8 | 69.0 | 53.7 | 52.4 | 51.8 | 52.8 | 51.2 | 51.9 | 52.4 | 52.6 | 51.5 |
| | ***CBF*** | 86.6 | 85.8 | 86.8 | 84.8 | 69.1 | 68.2 | 68.1 | 68.8 | 68.0 | 68.2 | 67.8 | 68.4 | 67.6 |
| | ***MERGE*** | 89.1 | 83.2 | 83.8 | 82.7 | 75.8 | 75.0 | 73.5 | 74.5 | 72.7 | 72.3 | 72.8 | 73.8 | 66.7 |
| **Class C** | ***SKIP*** | 64.3 | 62.7 | 64.6 | 56.0 | 48.2 | 46.7 | 48.7 | 45.7 | 43.0 | 42.9 | 41.9 | 43.9 | 44.5 |
| | ***CBF*** | 85.1 | 85.6 | 86.7 | 84.0 | 69.7 | 60.8 | 69.5 | 67.1 | 61.0 | 56.7 | 55.9 | 58.2 | 64.1 |
| | ***MERGE*** | 77.2 | 75.8 | 73.8 | 74.44 | 70.0 | 67.7 | 69.2 | 65.8 | 66.0 | 67.4 | 67.9 | 67.3 | 67.2 |

information. A statistical analysis of the proposed method is presented in Section II. Experimental results are shown in Section III. A conclusion is presented in Section IV

## II. PROPOSED ALGORITHM

HEVC includes too many stages of quad tree based structure to determine the best mode. The CU range starts from 64x64 to 8x8, so HEVC has four depth levels. In each depth level, an exhaustive RD-cost analysis is performed to determine the best mode among many modes, including SKIP, INTER_2Nx2N, INTER_Nx2N, INTER_2NxN, INTER_NxN, INTER_2NxnU, INTER_2NxnD, INTER_nLx2N, INTER_nRx2N, INTRA_2Nx2N, INTRA_NxN, and PCM. Equation (1) represents a way of determining the best mode. $\lambda_{MODE}$ indicates a Lagrangian multiplier in (1), and SSE is a cost function, as defined in (2).

$$J_{MODE} = SSE + \lambda_{MODE} \quad B_{MODE} \tag{1}$$

$$SSE = \sum_{i,j} (BlockA(i,j) \quad BlockB(i,j))^2 \tag{2}$$

However, if a close correlation between the current CU and other CUs can be defined, there is no need to calculate all modes. In this study, the skip flag, merge flag, and coded block flag (CBF) of neighboring CTUs are used to reduce the computational complexity.

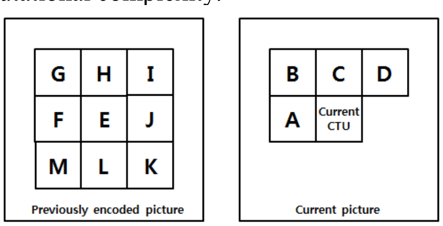


Figure 2. 13 neighboring CTUs around the current CTU.

### A. *Statistical Analysis*

Statistical analysis was performed for identification of correlations between the current CTU and neighboring CTUs for skip and merge modes. There are thirteen neighboring CTUs around a current CTU (Figure 2). Four spatially neighboring CTUs (A ~ D) are represented as left, above-left, above, and above-right. In a previously encoded picture, there are nine temporally adjacent CTUs (E ~ M), which indicate collocated-current, collocated-left, above-left, collocated-above, collocated-above-right, collocated-right, collocated-below-right, collocated-below and collocated-below-left, respectively. Conditional probabilities include all adjacent CTUs (A ~ M) and skip and merge flag = true and

CBF = 0 are also checked. At the same time, probabilities that the current CTU, expressed alphabetically as **O**∈(O) were skip and merge flag true and CBF 0 were also checked. HM 12.0, the reference software for the HEVC standard was used. The first 10 frames of the Kimono (1920x1080), ParkScene (1920x1080), Cactus (1920x1080), BQTerrace (1920x1080), BasketballDrive (1920x1080), RaceHorses (832x480), BQMall (832x480), PartyScene (832x480) and BasketballDrill (832x480) sequences were used with QP=22, 27, 32, and 37 with a random access profile.

Statistical results showed that P(O|A ~ D), indicating a high percentage of spatial CTUs. The probabilities of each CTU having the same skip flag true were 69.1 ~ 70.8% in class B and 55.99 ~ 64.6% in class C, and also, 84.75 ~ 85.8% in class B and 84.0 ~ 86.7% in class C (Table I). Also, the probabilities of CBF 0 that CTU should be chosen as skip mode were 84.8 ~ 86.8% in class B and 84.0 ~ 86.7% in class C. In addition, the probabilities of each CTU having the same merge flag true were 82.7 ~ 89.1% in class B and 73.8 ~ 77.2% in class C. In temporal CTUs with P(O|E ~ M), percentages of each mode were 51.5% ~ 53.7% in class B and 41.9 ~ 48.7% in class B for skip flag. Also, 67.6 ~ 69.1% in class B and 56.7 ~ 69.7% in class C were recoded for CBF. Lastly, 66.7 ~ 75.8% in class B and 65.8 ~ 70.0 % in class C were recoded for merge mode.

### B. *Early skip and merge mode decision*

If only the information mentioned above is used, loss is excessive. However, combining adjacent CTUs can reduce the loss. CTUs were divided into groups of built-in CTUs (BIC) and user defined CTUs (UDC) based on conditional probabilities, and statistical analysis was performed. Two groups were used because BICs have higher probability than UDCs, based on statistical results. Thus,

$$D = \sum_{BIC \in i} CU_i + \left( \sum_{UDC \in j} CU_j \quad \omega \right) \geq \alpha \tag{3}$$

where BIC are the left, above-left, above, and above-right CTUs in the current picture and collocated-current CTUs in the previously encoded picture. Also, UDC contains collocated-left, collocated-above-left, collocated-above, collocated-above-right, collocated-right, collocated-below-right, collocated-below, and collocated-below-left CTUs in the previously encoded picture. When $CU_i$ and $CU_j$ have a depth = 0, and skip flag = true or merge flag = true or CBF = 0, they are set to 1. ω is a weighting factor for UDC set to

TABLE II. STATISTICAL RESULTS ACCORDING TO $\alpha$

| | α | 4 | 4.75 | 5 | 5.5 | 5.75 | 6.5 |
|---|---|---|---|---|---|---|---|
| Class B | *BDBR (%)* | 3.22 | 2.87 | 1.18 | 1.92 | 1.02 | 1.12 |
| | *ΔBitrate(%)* | 0.08 | -0.10 | -0.38 | -0.13 | -0.40 | -0.36 |
| | *ΔPSNR(%)* | -0.23 | -0.16 | -0.12 | -0.15 | -0.12 | -0.12 |
| | *TS (%)* | 43.28 | 39.31 | 44.78 | 39.44 | 42.39 | 39.48 |
| Class C | *BDBR (dB)* | 3.88 | 3.17 | 1.33 | 2.82 | 1.44 | 1.27 |
| | *ΔBitrate(%)* | 1.32 | 0.80 | 0.10 | 0.75 | 0.098 | 0.07 |
| | *ΔPSNR(%)* | -0.35 | -0.29 | -0.17 | -0.29 | -0.17 | -0.16 |
| | *TS (%)* | 31.20 | 30.45 | 28.37 | 30.45 | 29.82 | 28.10 |

0.75 when $CU_j$ has the lowest RD-cost. Otherwise, ω is set to 0. α is a threshold value to specify the boundary of skip or merge modes. Lastly, Equation (3) is used for three types of calculations of skip flag ($D_{SKIP}$), CBF ($D_{CBF}$), and merge flag ($D_{MERGE}$). To obtain the value of $D_{SKIP}$, skip flag is used. For $D_{CBF}$, CBF is used, and for $D_{MERGE}$, merge flag is used.

Experiments were performed to determine an optimal threshold value of α with the lowest and the second lowest RD-cost values used to obtain the optimal α. Results are shown in Table II. When α = 4, four BICs are used. Bjøntegaard difference bitrate (BDBR) values were 3.22%, and 3.88%, and TS values were 43.28% and 31.2% for classes B and C. When α = 4.75, four BICs were used and one UDC was used, BDBR values were 2.87% and 3.17% and TS values were 39.31% and 30.45% in class B and class C, respectively. When α = 5, all BICs were used. BDBR values were 1.18% and 1.44%, and TS values were 44.78% and 28.82% in classes B and C, respectively. When α 5.5, four BICs were used and two UDCs were used. BDBR values were 1.18% and 1.33% and TS values were 44.78% and 28.37%, respectively. Also, when α = 5.75, all BICs and one UDC were used. BDBR values were 1.02% and 1.44% and TS values were 42.39% and 28.32% in classes B and C, respectively. When α = 6.5, all BICs and two UDCs were used. BDBR values were 1.12% and 1.27%, and TS values were 39.48% and 28.1% in classes B and C, respectively. Good efficiency was observed when α = 5.75 in both BDBR values and TS values based on the above simulation. Therefore, 5.75 was used as a threshold value.

### C. *Overall Algorithm*

The proposed algorithm is summarized in a flowchart in Figure 3.

**Step 1.** Start encoding a CTU.

**Step 2.** Check current depth. If depth level 0, go to Step 3. Otherwise, go to Step 8.

**Step 3.** Select one CTU with the lowest RD-cost value from UDC (F ~ M CTUs).

**Step 4.** Get all skip flags, CBFs, and merge flags from adjacent CTUs, which include all BIC (A ~ E CTUs) and one UDC selected through Step 3.

**Step 5.** Calculate Equation (3). If $D_{SKIP}$ is equal to or greater than α, the current CTU is selected as skip mode. Also, go to Step 9. Otherwise, go to Step 6.

**Step 6.** Calculate Equation (3). If $D_{CBF}$ is equal to or greater than α, the current CTU is selected as skip mode. Also, go to Step 9. Otherwise, go to Step 7.

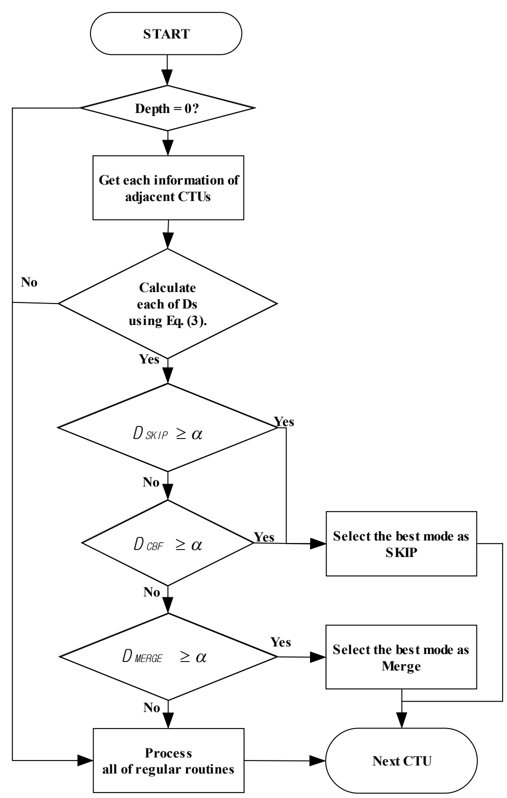


Figure 3. Flow chart of the proposed algorithm.

**Step 7.** Calculate Equation (3). If $D_{MERGE}$ is equal to or greater than α, the current CTU is selected as merge mode. Also, go to Step 9. Otherwise, go to Step 8.

**Step 8.** Process all regular routines.

**Step 9.** Encode the next CTU.

## III. EXPERIMENTAL RESULTS

The proposed algorithm was implemented on HEVC test model HM 12.0 and tested based on test conditions, configurations, and sequences recommended by JCT-VC [5]. These conditions and configurations are summarized in Table III. Performance evaluation was based on BDBR [8] and a computational complexity reduction in time saving (TS) as:

$$TS = \frac{Time(origin) \quad Time(prop)}{Time(origin)} \times 100 \qquad (4)$$

where $Time_{origin}$ and $Time_{prop}$ are the encoding times of reference software HM 12.0 and the proposed algorithm, respectively. For BDBR and TS, positive values indicated an increase and negative values a decrease.

TABLE III. TEST CONDITOINS.

| | |
|---|---|
| CPU | Intel i5-3470 CPU @ 3.20GHz |
| RAM | 4.00GB |
| OS | Microsoft Windows 7 |
| Profile | RA-Main, LD-Main |
| Motion Search | TZ search |
| Search Range | 64 |
| Max CU Size | 64x64 |
| Max CU Depth | 4 |
| QP | 22, 27, 32, 37 |
| FEN, FDM | On |

TABLE IV. SUMMARY OF ENCODING RESULTS IN RA.

| Class (resolution) | Sequence | Frame count | BDBR (%) | TS (%) |
|---|---|---|---|---|
| A – 4K (2560x1600) | Traffic | 150 | 1.6 | 45.2 |
| | PeopleOnStreet | 150 | 1.4 | 17.7 |
| B – 1080p (1920x1080) | ParkScene | 240 | 1.4 | 47.0 |
| | Cactus | 500 | 1.4 | 42.1 |
| | BQTerrace | 600 | 1.3 | 46.5 |
| | BasketballDrive | 500 | 1.0 | 36.2 |
| C - WVGA (832x480) | BQMall | 600 | 3.1 | 42.4 |
| | PartyScene | 500 | 1.1 | 30.5 |
| | BasketballDrill | 500 | 0.9 | 34.7 |
| D - WQVGA (416x240) | BQSquare | 600 | 0.5 | 36.0 |
| | BlowingBubbles | 500 | 1.2 | 29.1 |
| | BasketballPass | 500 | 0.5 | 21.2 |

TABLE V. SUMMARY OF ENCODING RESULTS IN LD.

| Class (resolution) | Sequence | Frame count | BDBR (%) | TS (%) |
|---|---|---|---|---|
| B – 1080p (1920x1080) | ParkScene | 240 | 1.0 | 29.7 |
| | Cactus | 500 | 0.5 | 29.2 |
| | BQTerrace | 600 | 0.8 | 36.1 |
| | BasketballDrive | 500 | 1.0 | 26.2 |
| C - WVGA (832x480) | BQMall | 600 | 2.5 | 34.7 |
| | PartyScene | 500 | 0.2 | 18.8 |
| | BasketballDrill | 500 | 0.9 | 26.3 |
| D - WQVGA (416x240) | BQSquare | 600 | -0.1 | 18.4 |
| | BlowingBubbles | 500 | 0.2 | 16.3 |
| | BasketballPass | 500 | 0.3 | 16.1 |

Tables IV and V show performance results for random-access and low-delay, respectively. On average, the BDBR value was 1.3% and TS value was 31.45% in the random-access profile for A class sequences. In B class sequences, a 1.275% BDBR value was observed with a speed-up of 42.95% in random-access. In low-delay profiles, 0.825% for BDBR and 30.3% for TS were achieved.

Also, in C class sequences, BDBR values were 1.7% and 1.2% while obtaining 35.87% and 26.6% for TS with random-access and low-delay profiles respectively. In D class sequences, BDBR values of 0.73% and 0.13% with TS values of 28.77% and 16.93% with random access and low delay profiles respectively, were achieved. The proposed algorithm reduced the time required with minimal quality degradation, compared with the original encoder.

## IV. CONCLUSION

An early skip and merge mode decision algorithm has been proposed based on spatially and temporally adjacent CTUs. If skip or merge mode is identified, further processes are omitted. Experimental results show that the proposed algorithm achieved an average time reduction of 34.76% in random access profile and 24.61% in low delay profile, while maintaining a comparable RD performance. The proposed method can be useful for supporting a real-time HEVC encoder implementation.

## REFERENCES

[1] B. Bross, W. -J. Han, G. J. Sullivan, J. -R. Ohm, and T. Wiegand, “High efficiency video coding (HEVC) text specification draft 8,” ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J1003, July 2012.

[2] G. J. Sullivan, J. –R. Ohm, W.-J. Han and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard”, IEEE Trans. Circuits Syst. Video Technol., Vol. 22, no. 12, Dec. 2012, pp. 1649-1706.

[3] I. -K. Kim, J. Min, T. Lee, W. -J. Han and J. -H. Park, “Block partitioning Structures in HEVC”, IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, Dec. 2012, pp. 1697-1706.

[4] “Common test conditions and software reference configurations”, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-I1100, May 2012.

[5] P. –T. Chiang, T. S. Chang, “Fast zero block detection and early CU termination for HEVC video coding”, IEEE International Symposium on Circuits and Systems (ISCAS), May 2013, pp. 1640-1643.

[6] Zhaoping Pan, Sam Kwong, Yun Zhang, Jianjun Lei, “Fast coding tree unit depth decision for high efficiency video coding”, IEEE International Conference on Image Processing (ICIP), Oct, 2014, pp. 3214-3218.

[7] Kibeya, H., Belghith, F., Ben Ayed, M. A, Masmoudi, N., “A fast CU partitionning algorithm based on early detection of zero block quantified transform coefficients for HEVC standard”, IEEE International Conference on Image Processing, Applications and Systems (IPAS), Nov 2014, pp.1-5.

[8] G. Bjøntegaard, “Calculation of Average PSNR Differences between RD Curves”, document VCEG-M33, Austin, TX, USA, Apr. 2001.

 ISBN: 978-1-61208-452-7 