# An Adaptive Inter CU Depth Decision Algorithm for HEVC

Jie Liu, Huizhu Jia*, Guoqing Xiang, Xiaofeng Huang, BinBin Cai, Chuang Zhu, Don Xie
*National Engineering Laboratory for Video Technology, Peking University, Beijing, China*
{liuzimin, hzjia, gqxiang, xfhuang, bbcai, czhu, xdxie}@jdl.ac.cn

*Abstract*—**The emerging High-Efficiency Video Coding (HEVC) standard has introduced a number of new coding tools, such as a quad-tree based coding unit (CU). The quadtree-structured coding unit achieves significant coding efficiency improvements compared to H264/AVC. However, the complexity of CU depth decision associated with Rate-Distortion (R-D) cost computation dramatically increased. In order to alleviate the computational burden in HEVC inter coding, a fast CU depth decision algorithm is proposed in this paper. Firstly, zero CU detection method for HEVC is proposed as early termination algorithm. Secondly, the CU depth pruning strategies are adaptively determined according to standard deviation of statistic spatiotemporal depth information. Finally, when the neighbors are not available or have a very weak correlation, edge gradient of current coding tree unit (CTU) is considered as main factor for CU depth pruning method. Experimental results demonstrate that, compared with the original HM16.0 implementation, the proposed algorithm achieves about 40.5% encoding time saving with ignorable coding performance degradation.**

***Index Terms*—High-Efficiency Video Coding (HEVC), CU depth decision, zero CU detection, standard deviation, edge gradient.**

## I. INTRODUCTION

High Efficiency Video Coding [1], is currently being developed by the Joint Collaborative Team on Video Coding (JCT-VC) jointly established by ISO/IEC MPEG and ITU-T VCEG. Compared to H.264/AVC, HEVC achieves approximately 50% bit-rate reduction for equal perceptual quality [2].

In HEVC, the quad-tree structure of CU is one of the most efficient tools to improve performance. A coding CTU represents the basic processing unit in HEVC. HEVC supports variable CTU sizes from 64×64 down to 8×8 samples. Each CU has an associated partitioning into prediction units (PUs) and a tree of transform units(TUs). Although HEVC achieves significant improvement on coding performance, it causes a dramatic increase in encoding complexity because the splitting process needs to exhaust all the CUs from 8 × 8 to 64 × 64 size where the best PU size and the best TU partition must be decided for all possible PU and TU sizes.

Recently, many efforts [3-6] have been made to reduce the encoder complexity of HEVC. In [4][5], a fast CU depth decision method proposed by using the depth information of neighboring CUs and the co-located CUs of the previous frame. These methods don't work effectively when the current CU in inhomogeneous regions. In this situation, the depth search range always determines to be all the four depth and it has some limitation on encoding time saving. Shen's [6] method proposed early termination algorithm using CU split information and the SKIP mode information for reducing encoding time. The early termination algorithm in [6] checks the SKIP mode from the current CU and the parent CU. Since SKIP detection criterion requires the RD cost, additional computational cost can be not neglected. In addition, since Shen's method also utilizes motion vectors and RD costs from the neighboring CU blocks, performance degradation in inhomogeneous motion regions is inevitable. Xiong [3] proposed CU selection algorithm based on pyramid motion divergence (PMD). The PMD is evaluated as the variance of the optical flows of the current CU and the sub-CUs. However, the pixel-wise optical flow estimation adds significant extra computation with its heavy floating-point operation.

In this paper, in order to incorporate the features of current CTU and the neighboring information efficiently, we propose an adaptive inter CU depth decision algorithm for HEVC. For achieving significant time saving, firstly, an early termination method is introduced by using zero CU detection. Then, the proposed algorithm effectively analyses standard deviation of statistic spatiotemporal depth information. According to the standard deviation, we make different CU depth pruning strategies. Finally, when the neighbors have a very weak correlation or are not available, edge gradient of current CTU is considered as a main factor for CU depth pruning method.

The remainder of this paper is organized as follows. Section II presents the overall adaptive inter CU depth decision algorithm. The experimental results are shown in Section III. Finally, we draw a conclusion in Section IV.

## II. INTER CU DEPTH DECISION ALGORITHM

In this section, we propose an adaptive inter CU depth decision algorithm which includes an early termination method by using zero CU detection, statistic spatiotemporal depth information and edge gradient of current CTU.

### A. Zero CU Detection Algorithm

In HEVC, RD costs are calculated to decide whether a CU should be split or not. The cost function is shown as (1), where $j$ is the depth level of $CU_j$ and $D_j(i)$ represents the $i^{th}$

CU in $j^{th}$ depth level. The bit-rate $R^c$ and $R^h$ are the estimated coding bits of the quantized coefficient and header syntax.

$$J_{UnSplit\_CU_j} = D_j + \lambda(R_j^c + R_j^h) \quad J_{Split\_CU_j} = \sum_{i=0}^{3} D_{j+1}(i) + \lambda \sum_{i=0}^{3} (R_{j+1}^c(i) + R_{j+1}^h(i)) \quad (1)$$

Zero CU is defined as all DCT coefficients are equal to zero after quantization (Q). If DCT coefficients of the $CU_j$ are equal to zero, all the coefficients of the $CU_{j+1}$ also will be equal to zero. As shown in (2), the distortion and coefficient bits of $CU_j$ are equal to the sum of $CU_{j+1}$. Since the split flag bits will be added to the header bits, the RD costs of splitting is greater than that of un-splitting. The current CU is most likely to be not splitting when it is detected as zero CU.

$$D_j = \sum_{i=0}^{3} D_{j+1}(i), \; R_j^c = \sum_{i=0}^{3} R_{j+1}^c(i) = 0, \; R_j^h < \sum_{i=0}^{3} R_{j+1}^h(i) \quad (2)$$

Therefore, zero CU detection is an efficient scheme as early termination method. There are masses of algorithms for zero block (ZB) detection in H.264/AVC. In [7], Hadamard coefficients are utilized for ZB detections. In HEVC, the largest CU size is 64 and additional computation for larger CU size Hadamard transforms is significant. Moreover, the DCT and quantization operation of H264/AVC differs from that of HEVC. In this paper, a novel zero CU detection algorithm for HEVC is proposed using SAD of current CU.

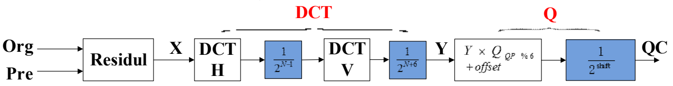


Fig. 1 The transform and quantization operation of HEVC.

Fig. 1 shows the transform and quantization operation in HEVC. DCT transform and uniformly scaling operation for a $W \times W$ residual block are shown in (3). Where $W$ denotes the width of DCT matrix, $N = log_2 W$ and $scale = 2N+5$. The superscript $T$ denotes the transpose of the matrix. $X$ represents the $W \times W$ residual matrix and $C$ represent the DCT matrices.

$$\mathbf{Y}(u, v) = \left[ \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} \mathbf{C}_W(x, u) \mathbf{X}(x, y) \mathbf{C}_W^T(y, v) \right] >> \text{scale} \quad (3)$$

Equation (4) shows the quantization operation of HEVC. Where $QP_{QP\%6}$ is constant scale of quantization according to $QP$ modulo 6 and $Q\_SHIFT=14$, $D\_RANGE=15$. $BitDepth$ indicates the number of the bits used to represent each pixel. For P or B slice, $offset = 85 \times 2^{shift-9}$ and $shift = Q\_SHIFT + D\_RANGE + QP/6 - BitDepth - N$.

$$\mathbf{QC}(u, v) = (\mathbf{Y}(u, v) \times Q_{QP\%6} + \text{offset}) >> \text{shift} \quad (4)$$

In [7], the current CU will be considered as a zero block when the every quantized coefficient is zero. The condition of zero block detection is shown in (5). Therefore, with (3) and (5), the derivation of the threshold value for SAD is shown in (6). Where **Max[]** represents the maximum value of the matrix.

$$|\mathbf{QC}(u, v)| = |(\mathbf{Y}(u, v) \times Q_{QP\%6} + \text{offset}) >> \text{shift}| < 1 \quad (5)$$

$$\left[ \left| \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} \mathbf{C}_W(x, u) \mathbf{X}(x, y) \mathbf{C}_W^T(y, v) \right| \times (Q_{QP\%6} >> \text{scale}) + \text{offset} \right] >> \text{shift} < 1 \quad \Rightarrow \quad (6)$$

$$\left[ \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} |\mathbf{X}(x, y)| \times \mathbf{Max}\left[ \mathbf{C}_W(u, v) \mathbf{C}_W^T(u, v) \right] \times (Q_{QP\%6} >> \text{scale}) + \text{offset} \right] >> \text{shift} < 1$$

In (6), $\sum_{x=0}^{W-1} \sum_{y=0}^{W-1} |\mathbf{X}(x, y)|$ is the definition of $SAD_{W \times W}$ for the residual data. Then, we can get (8).

$$SAD_{W \times W} \times \mathbf{Max}\left[ \mathbf{C}_W(u, v) \mathbf{C}_W^T(u, v) \right] \times (Q_{QP\%6} >> \text{scale}) + \text{offset}) >> \text{shift} < 1 \quad (7)$$

$$SAD_{W \times W} < \frac{2^{N+5} \times (2^{R(QP)} - 85 \times 2^{R(QP)-9})}{\mathbf{Max}\left[ \mathbf{C}_W(u, v) \mathbf{C}_W^T(u, v) \right] \times Q_{QP\%6}} \quad (8)$$

Therefore, the $SAD_{W \times W}$ threshold value for HEVC, denoted by $Thd\_SAD_{W \times W}(QP)$, is shown in (9). Where $R(QP) = Q\_SHIFT + D\_RANGE + QP/6 - BitDepth$. It is noted that the proposed SAD threshold value is only correlated with the width of DCT matrix and QP.

$$Thd\_SAD_{W \times W}(QP) = \frac{2^{N+5} \times (2^{R(QP)} - 85 \times 2^{R(QP)-9})}{\mathbf{Max}\left[ \mathbf{C}_W(u, v) \mathbf{C}_W^T(u, v) \right] \times Q_{QP\%6}} \quad (9)$$

The early termination algorithm is proposed based on the zero CU detection. Before the current CTU encoding, integer motion estimation (IME) is conducted for the current CTU and it is centred at predicted motion vector and the search range is set to [-2, 2]×[-2, 2]. The search results such as one $SAD_{64x64}$ and four $SAD_{i,32x32}$ are achieved. Then, if $SAD_{64x64}$ is smaller than the $Thd\_SAD_{64x64}(QP)$, the depth range of current CTU is set from 0 to 1. If not, statistic spatiotemporal depth information is used to decide the depth range.

### B. Statistic Spatiotemporal Depth Information

In HEVC, the splitting depth of the CTU will be achieved after the current CTU encoded. The depth values $N_{Depth} = 0$, 1, 2 and 3 correspond to CU size of 64×64, 32×32, 16×16 and 8×8. The depth decision process using spatiotemporal information are presented below.

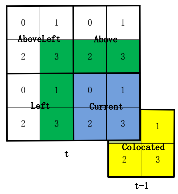


Fig. 2 Spatiotemporal adjacent CTU of current 64x64 CTU.

1) *Statistic Neighbor Depth Information*. Fig. 2 illustrates spatiotemporal CTU of current 64x64 CTU. Every neighbouring 64×64 CTU is divided into four 32×32 regions. The correlation between current 64x64 CTU and neighbouring five 32×32 regions (green region) is much stronger than the other seven 32×32 regions (white region). The correlation degree [4] of neighbors and the current CTU are shown in Table I. Collocated (yellow region) represents the co-located CTU in previous frame. ΔQP is the difference QP between current CTU and co-located CTU. According to the correlation degree, white region depth information is not adopted in this paper.

TABLE I
CORRELATION DEGREE OF SPATIOTEMPORAL ADJACENT CTUS AND CURRENT CTU

| | **White Region** | **Green Region** | **Collocated ΔQP<=1** | **Collocated ΔQP>1** |
|---|---|---|---|---|
| Correlation | 0.66 | 0.81 | 0.78 | 0.72 |

The statistics of spatiotemporal value is shown as (10) and (11), where $i$ represents the index of four 32×32 region as shown in Fig. 2 and $j$ is depth level from 0 to 3. Taking left neighboring CTU as an example, $N$ is the number of CU in $i^{th}$ 32×32 region and $LDepth_i(k)$ is depth value of $k^{th}$ CU in the $i^{th}$

32×32 region in left neighbour CTU. *$LeftDepth_{i,j}$* represents the number of CU in $j^{th}$ depth level in the $i^{th}$ 32×32 region. $\alpha_s$ is the weighting factor based on the relative distance between the current CTU and its neighbors. The weighting factor $\alpha_t$ is adjusted according to the ΔQP. Finally, statistic depth value of five spatial 32×32 regions (green region) and four temporal 32×32 regions (yellow region) are achieved.

$$\mathrm{LeftDepth}_{i,j}=\sum_{k=0}^{N-1}[\alpha_s\cdot(\mathrm{LDepth}_i(k)==j)] \tag{10}$$

$$\mathrm{CollDepth}_{i,j}=\sum_{k=0}^{N-1}[\alpha_t\cdot(\mathrm{CDepth}_i(k)==j)] \tag{11}$$

2) *Tow Step DR Decision*. Step1: Statistic depth value of all 32×32 regions should be used to decide search depth range from 0 to 1. In (12), *$CurrDepth64_j$* represents the number of CU in $j^{th}$ depth level and $j$ is the depth value from 0 to 3.

$$\begin{aligned}\mathrm{CurrDepth64}_j &= \mathrm{LeftDepth}_{1,j}+\mathrm{LeftDepth}_{3,j}+\mathrm{AboveDepth}_{2,j}\\ &+\mathrm{AboveDepth}_{3,j}+\mathrm{AboveLeftDepth}_{3,j}+\sum_{i=0}^{3}\mathrm{CollDepth}_{i,j}\end{aligned} \tag{12}$$

Then, the expectation ($\mu_{64x64}$) and variance ( $\sigma^2_{64\times64}$ ) are calculated as shown in (13). Variance reflects difference between neighbors. The variance value of zero indicates that all statistical neighbors are identical and a small variance indicates that the group of CTUs tend to be very close. Variance is used to see how current CTU relates to neighbors.

$$\mu_{64\times64}=\frac{\sum_{j=0}^{3}(j\cdot\mathrm{CurrDepth64}_j)}{\sum_{j=0}^{3}\mathrm{CurrDepth64}_j},\ \sigma^2_{64\times64}=\frac{\sum_{j=0}^{3}[j\cdot(\mathrm{CurrDepth64}_j-\mu_{64\times64})^2]}{\sum_{j=0}^{3}\mathrm{CurrDepth64}_j} \tag{13}$$

Since the standard deviation has the same dimension as depth value, the current CTU is likely to search in a small depth range if the standard deviation of statistical depth value is smaller than 1. The depth range decision strategies are shown in (14). Step 1 mainly decide the depth range of current CTU from 0 to 1. If $\mu_{64x64}$ is larger than 1, it indicates that current CTU size has most probability to be smaller than 64×64. Thus, the neighboring statistic depth information of current four 32×32 CUs are discussed respectively in Step2.

$$\mathrm{PredRange}_{64\times64}\in\begin{cases}\{0\} & if\ \ (0=\mu_{64\times64})\cap(\sigma_{64\times64}=0)\cap \mathit{Zero\ CU};\\ \{0,1\} & if\ \ (0\le\mu_{64\times64}\le1)\cap(\sigma_{64\times64}<1);\\ \text{Step 2} & else\end{cases} \tag{14}$$

Step2: In Fig. 3, the bold black border represents current 64×64 CTU and four current 32×32 CUs (red border) use the depth value of different adjacent CTUs as its statistic information. The green blocks represent spatial 32×32 CU and the yellow blocks represent temporal collocated 32×32 CU.

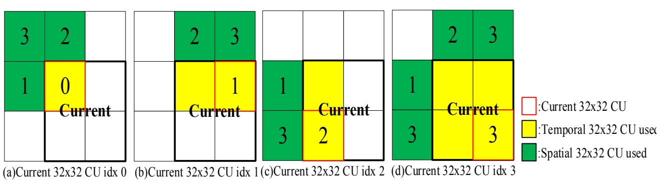


Fig. 3 Neighboring statistic depth information of current four 32x32 CUs.

In (15), *$CurrDepth32_{i,j}$* represents the number of CU in $j^{th}$ depth level of $i^{th}$ 32×32 CU in current CTU. Then, the corresponding the expectation and variance is calculated using the similar formula (13).

$$\begin{aligned}\mathrm{CurrDepth32}_{0,j} &= \mathrm{LeftDepth}_{1,j}+\mathrm{AboveDepth}_{2,j}+\mathrm{AboveLeftDepth}_{3,j}+\mathrm{CollDepth}_{0,j}\\ \mathrm{CurrDepth32}_{1,j} &= \mathrm{AboveDepth}_{2,j}+\mathrm{AboveDepth}_{3,j}+\mathrm{CollDepth}_{0,j}+\mathrm{CollDepth}_{1,j}\\ \mathrm{CurrDepth32}_{2,j} &= \mathrm{LeftDepth}_{1,j}+\mathrm{LeftDepth}_{3,j}+\mathrm{CollDepth}_{0,j}+\mathrm{CollDepth}_{2,j}\\ \mathrm{CurrDepth32}_{3,j} &= \sum_{i=0}^{3}\mathrm{CollDepth}_{i,j}+\mathrm{LeftDepth}_{1,j}+\mathrm{LeftDepth}_{3,j}+\mathrm{AboveDepth}_{2,j}\\ &+\mathrm{AboveDepth}_{3,j}\end{aligned} \tag{15}$$

The depth range of current 32×32 CTU drifts upwards or downwards along the average depth value. If $\sigma_{i,32x32}$ is smaller than 1, it means that two depth range will cover the most possible depth range. In addition, when the $\sigma_{i,32x32}$ is greater than 1 and less than 1.5, three depth search range will be adopted as shown in (16). Larger standard deviation indicates that the current CTU has a very weak correlation with its neighbors. Thus, edge gradient is used to decide depth range and the detail will be discussed in Section II-C.

$$\mathrm{PredRange}_{i,32\times32}\in\begin{cases}\{0,1\} & if\ \ (0\le\mu_{i,32\times32}\le1)\cap(\sigma_{i,32\times32}<1);\\ \{1,2\} & if\ \ (1<\mu_{i,32\times32}\le2)\cap(\sigma_{i,32\times32}<1);\\ \{2,3\} & if\ \ (2<\mu_{i,32\times32}\le3)\cap(\sigma_{i,32\times32}<1);\\ \{0,1,2\} & if\ \ (\mu_{i,32\times32}\le1.5)\cap(1\le\sigma_{i,32\times32}<1.5);\\ \{1,2,3\} & if\ \ (\mu_{i,32\times32}>1.5)\cap(1\le\sigma_{i,32\times32}<1.5);\\ \mathit{Edge\ Gradient\ Detection} & if\ \ \sigma_{i,32\times32}\ge1.5\end{cases} \tag{16}$$

## C. Edge Gradient Detection

In general, the final CU depth has a correlation with the texture complexity. Edge gradient can reflect the texture complexity of the image. Sobel edge detector uses a pair of 3×3 convolution masks. For a pixel $p_{i,j}$, in a luma picture, we define the corresponding edge vector, $\vec{D}_{i,j}=\{dx_{i,j},dy_{i,j}\}$ as

$$\begin{aligned}dx_{i,j} &= p_{i-1,j+1}+2\times p_{i,j+1}+p_{i+1,j+1}-p_{i-1,j-1}-2\times p_{i,j-1}-p_{i+1,j-1}\\ dy_{i,j} &= p_{i+1,j-1}+2\times p_{i+1,j}+p_{i+1,j+1}-p_{i-1,j-1}-2\times p_{i-1,j}-p_{i-1,j+1}\end{aligned} \tag{17}$$

where x and y represent the degree of difference in vertical and horizontal directions respectively. Therefore, the amplitude of the edge vector can be computed by

$$\mathrm{Amp}(\vec{D}_{i,j})=|dx_{i,j}|+|dy_{i,j}| \tag{18}$$

Depth search range is divided into two categories by threshold *Thd_Edge*. *Thd_Edge* is the average amplitude of the edge which the final CU depth is from 0 to 2 in previous frame. *Thd_Edge* is updated after the current frame is encoded.

$$\mathrm{PredRange}_{64\times64}\in\begin{cases}\{0,1,2\} & if\ \ \mathrm{Edge}_{W\times W}=\sum_{i,j\in W\times W}\mathrm{Amp}(\vec{D}_{i,j})<\mathrm{Thd\_Edge}\\ \{1,2,3\} & if\ \ \mathrm{Edge}_{W\times W}=\sum_{i,j\in W\times W}\mathrm{Amp}(\vec{D}_{i,j})\ge\mathrm{Thd\_Edge}\end{cases} \tag{19}$$

## D. Overall Algorithm

Fig. 4 shows the overall flowchart of the proposed algorithm and $i$ represent $i^{th}$ 32×32 CU in current CTU. Firstly, zero CU detection is conducted as early termination scheme. If this condition is not satisfied, the proposed algorithm will use statistic spatiotemporal depth information. Finally, when the current CTU has a very weak correlation with its neighbors or neighbors are not available, the edge gradient be calculated and depth search range is divided into two categories as shown in (19). In addition, every category need to further conduct zero 32×32 $CU_i$ detection. If all the 32×32 $CU_i$ are finished, current CTU encoding will be executed.

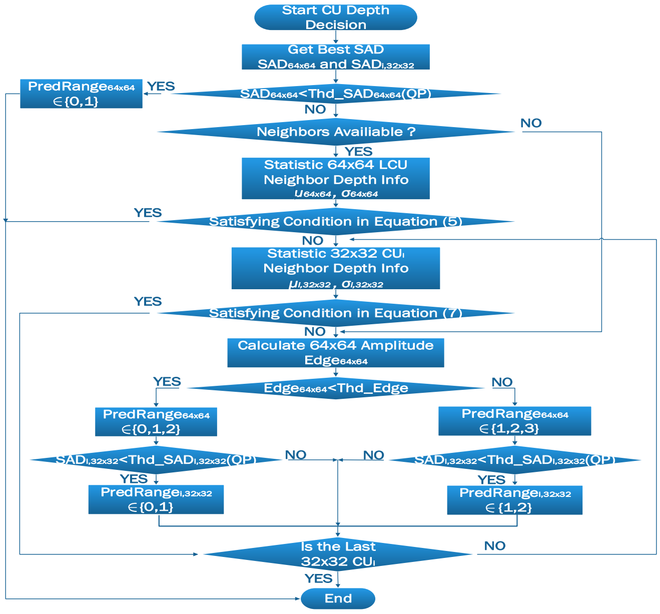

Fig. 4 The overall flowchart of the fast CU depth decision algorithm.

## III. Experimental Results

Our proposed algorithm is implemented into HM16.0 and tested it under the common test conditions of HEVC standardization [8]. The performance of the proposed algorithm is measured in terms of encoding time saving and BD-rate [9] increments for low-delay P configuration with the 64×64 CTU and the max CU depth of 4 for QP = {22, 27, 32, 37}. The encoding time saving is defined as

$$\Delta T(\%) = \frac{Time_{HM16.0} - Time_{proposed}}{Time_{HM16.0}} \times 100\% \tag{20}$$

For comparison, we also implemented the fast CU depth decision algorithm proposed in papers [4][6] on the test. As shown in Table III, our proposed algorithm can achieve 40.5% reduction of the coding time with average 1.6% BD-rate loss. The proposed overall algorithm outperforms Shen's fast inter-mode decision method [6] with 5.3% more encoding time saving with 0.1% more BD-rate decrease. Shen's algorithm mainly depends on neighbour depth information. When the current CU has a week correlation with its neighbors, performance degradation is inevitable in these method. Zhang's method [4] obtained the encoding time saving with average 20.9% time saving with average 0.8 % BD-rate loss. Although the BD-rate losses of Zhang's method is smaller, it doesn't bring effective encoding time saving. This is because Zhang's method performs very conservatively where most CU depth search range is set to 3 when the similarity degree of current CTU is medium similarity and low similarity. Zhang's method didn't use any early termination methods.

## IV. Conclusion

In this paper, zero CU detection and standard deviation of spatiotemporal depth information are proposed to reduce the depth range of current CTU. Edge gradient of current CTU is considered main factor when the current CTU have a weak correlation with its neighbors. The proposed algorithm can effectively reduce the computational complexity by 40.5% on average with 1.6% BDBR increase.

TABLE III
Results of the Proposed Algorithm Compared with the State-of-the-Art Fast Algorithm

| Class | Sequence | Proposed | | Shen[6] | | Zhang[4] | |
|---|---|---|---|---|---|---|---|
| | | ΔT (%) | BDBR (%) | ΔT (%) | BDBR (%) | ΔT (%) | BDBR (%) |
| Class A | Traffic | 47.4 | 2.8 | 42.5 | 2.1 | 24.7 | 1.1 |
| | PeopleOnstreet | 42.5 | 1.3 | 35.7 | 1.1 | 12.6 | 0.9 |
| Class B | BasketballDrive | 43.9 | 2.2 | 33.5 | 1.8 | 25.2 | 1.2 |
| | BQTerrace | 45.7 | 1.7 | 35.1 | 1.9 | 24.6 | 0.6 |
| | Cactus | 45.2 | 1.6 | 41.8 | 2.1 | 24.3 | 0.7 |
| | Kimono | 47.7 | 0.8 | 22.8 | 1.3 | 24.1 | 0.6 |
| | ParkScene | 45.8 | 2.4 | 35.4 | 1.3 | 22.5 | 0.6 |
| Class C | BasketballDrill | 35.9 | 2.3 | 32.8 | 2.5 | 18.8 | 0.6 |
| | BQMall | 34.8 | 1.9 | 41.6 | 5.8 | 16.6 | 0.5 |
| | PartyScene | 37.8 | 1.2 | 28.7 | 0.7 | 18.6 | 0.7 |
| | RaceHorses | 35.3 | 1.1 | 27.5 | 0.8 | 19.8 | 0.7 |
| Class D | BasketballPass | 31.2 | 1.3 | 29.2 | 1.2 | 16.5 | 0.6 |
| | BlowingBubbles | 27.8 | 0.6 | 23.1 | 0.8 | 15.6 | 0.6 |
| | BQSquare | 33.9 | 0.8 | 20.3 | 0.5 | 17.1 | 0.6 |
| | RaceHorses | 29.6 | 1.1 | 23.5 | 0.3 | 20.2 | 0.8 |
| Class E | FourPeople | 46.8 | 2 | 51.2 | 1.3 | 23.5 | 0.9 |
| | Johnny | 48.5 | 2.2 | 56.1 | 2 | 25.7 | 0.9 |
| | KristenAndSara | 48.8 | 2.3 | 52.5 | 2.2 | 27.2 | 1.1 |
| Average | | 40.5 | 1.6 | 35.2 | 1.7 | 20.9 | 0.8 |

## Acknowledgment

This work is partially supported by grants from the Chinese National Natural Science Foundation under contract No.61171139, National Key Technology Research and Development Program of the Ministry of Science and Technology of China under contract No.2014BAK10B00, Major National Scientific Instrument and Equipment Development Project of China under contract No. 2013YQ-030967, also with Beida (Binhai) Information Research.

## References

[1] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, *WorkingDraft 6 of High Efficiency Video Coding, document JCTVC-H1003, JCTVC of ISO/IEC and ITU-T*, San José, CA, USA, Feb. 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, pp. 1649-1668, Dec.2012.

[3] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[4] Y. Zhang, H. Wang, and Z. Li, "Fast coding unit depth decision algorithmfor interframe coding in HEVC," in *Proc. DCC*, Mar. 2013, pp. 53–62.

[5] C. Zhou, F. Zhou, and Y. Chen, "Spatio-temporal correlation-based fast coding unit depth decision for high efficiency video coding," J. *Electron. Image.* vol. 22, no. 4, p. 043001, Oct. 2013.

[6] L. Shen, Z. Liu, X. Zhang, W. Zhao, and Z. Zhang, "An effective CU size decision method for HEVC encoders," *IEEE Trans. Multimedia*,vol. 15, no. 2, pp. 465–470, Feb. 2013..

[7] Z. Liu, L. Li, Y. Song, S. Li, S. Goto, and T. Ikenaga, "Motion featureand Hadamard coefficient-based fast multiple reference frame motion estimation for H.264," *IEEE Trans. Circuits Syst. Video Technol.*, vol.18, no. 5, pp. 620–632, May 2008.

[8] F. Bossen, "Common HM test conditions and software reference configurations,*" document Joint Collaborative Team on Video Coding of ISO/IEC and ITU-T, JCTVC-K1100,* Shanghai, China, Oct. 2012.

[9] G. Bjontegaard, "Calculation of average PSNR difference between RD curves," *13th VCEG-M33 Meeting*, Austin, TX, Apr. 2-4, 2001