Received: 17 February 2019 | Revised: 3 December 2019 | Accepted: 10 July 2020

DOI: 10.4218/etrij.2019-0217

ORIGINAL ARTICLE

ETRI Journal WILEY

# Adaptive block tree structure for video coding

**Aram Baek**[1] | **Daehyeok Gwon**[1] | **Sohee Son**[1] | **Jinho Lee**[2] | **Jung-Won Kang**[2] | **Hui Yong Kim**[2*] | **Haechul Choi**[1]

[1]Department of Multimedia Engineering, Hanbat National University, Daejeon, Rep. of Korea

[2]Broadcasting Media Research Group, Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea

**Correspondence**
Haechul Choi, Department of Multimedia Engineering, Hanbat National University, Daejeon, Rep. of Korea.
Email: choihc@hanbat.ac.kr

**Present address**
[*]Department of Computer Science and Engineering, Kyung Hee University, Yongin, Rep. of Korea.

**Funding information**
This work was supported by Institute for Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. 2016-0-00572, Development and Standardization of 5th Generation Video/Audio Coding Technology for Ultra High Quality Media Service, and No. 2020-0-00452, Development of Adaptive Viewer-centric Point cloud AR/VR (AVPA) Streaming Platform)

**Abstract**

The Joint Video Exploration Team (JVET) has studied future video coding (FVC) technologies with a potential compression capacity that significantly exceeds that of the high-efficiency video coding (HEVC) standard. The joint exploration test model (JEM), a common platform for the exploration of FVC technologies in the JVET, employs quadtree plus binary tree block partitioning, which enhances the flexibility of coding unit partitioning. Despite significant improvement in coding efficiency for chrominance achieved by separating luminance and chrominance tree structures in I slices, this approach has intrinsic drawbacks that result in the redundancy of block partitioning data. In this paper, an adaptive tree structure correlating luminance and chrominance of single and dual trees is presented. Our proposed method resulted in an average reduction of −0.24% in the Y Bjontegaard Delta rate relative to the intra-coding of JEM 6.0 common test conditions.

**KEYWORDS**

block structure, coding efficiency, quadtree plus binary tree, versatile video coding, video coding

## 1 | INTRODUCTION

High-efficiency video coding (HEVC) [1] has been widely used following its standardization in January 2013. HEVC adopts a quadtree (QT)-based block partitioning and multiple partition concepts, including a coding unit (CU), prediction unit (PU), and transform unit (TU) [2]. The optimal CU partitioning may range from 64 × 64 to 8 × 8 pixels. It is a key feature of HEVC, contributing to higher coding efficiency compared with previous video coding standards that mainly used 16 × 16 macroblocks [3].

Nevertheless, video compression techniques are needed to provide substantially higher coding efficiency than the current video coding standards allow because of the demand for an increased bit rate to support higher resolution video. The Joint Video Exploration Team (JVET) is a group of video coding experts from the ITU-T Study Group 16, Visual Coding Experts Group (VCEG), and the ISO/IEC JTC 1/SC 29/WG 11 Moving Picture Experts Group (MPEG) [4]. The JVET activities include study of future video coding (FVC) technology with a potential compression capability that significantly exceeds that of the HEVC standard and evaluation of compression technology.

The QT-based CU block partitioning of HEVC reduces flexibility because of the square-shaped CU and fixed PU types. The JVET employs a QT plus binary tree (QTBT) [5] block partitioning, which enhances the flexibility of CU partitioning of shapes with QTs [6] and binary trees (BTs) [7], and eliminates multiple partition concepts to increase the coding efficiency beyond HEVC. In addition, the QTBT partitioning separates luminance and chrominance block partitioning for a coding tree unit (CTU) in I slices; the luminance and chrominance CTUs in the P and B slices share the same block partitioning. The joint exploration test model (JEM) 1.0 [8] including only QTBT provides a common platform for the exploration of FVC technologies by the JVET, resulting in a coding efficiency increase of 4.70%, 8.94%, and 10.46% for average Y, U, and V Bjontegaard Delta (BD) rates, respectively, with a 179% average encoding time and 104% average decoding time compared with JEM 1.0 with a random access (RA) configuration [9].

To increase the coding efficiency of the JEM, new approaches based on QTBT have been proposed [10–12]. In addition to QTBT, multi-type tree (MTT) partitioning with horizontal and vertical centers is used to capture the objects in the middle of a block. The MTT partitioning yielded a reduction of 3.18% in average Y BD-rate with 201% encoding time relative to JEM 3.1 with an RA configuration [11]. In [12], BT split types of QTBT were extended to support asymmetric binary partitioning of a CU, in which a horizontal or vertical BT split is divided into partitions 1/4 and 3/4 in sizes. The asymmetric binary partitioning resulted in average coding gains of 1.54% in luminance and approximately 7.8% in chrominance with 407% encoding time reported using the RA configuration, when compared with JEM 3.0 [13].

The increased luminance and particularly the coding efficiency of the chrominance is attributed to QTBT supporting separable and sharable luminance and chrominance block partitioning depending on the slice type. In QTBT partitioning, dual trees were set in I slices and single trees were set in P and B slices. The dual trees for regions of luminance and chrominance were established separately and signaled in I slices. However, a single partitioning tree was shared by luminance and chrominance regions. The dual tree in QTBT increases the coding efficiency when the characteristics of luminance and chrominance signals differ. However, if the luminance and chrominance signals are similar, the luminance and chrominance partitioning is likely to exhibit similar partitioning trees, increasing the redundancy of the respective signals. Therefore, the coding efficiency can be increased by reducing the redundant QTBT partitioning information during rate-distortion (RD) optimization. To reduce the redundancy, we present an adaptive tree structure for the intra-coding of JEM, which determines whether to use a dual or single tree for the luminance and chrominance regions based on the comparative RD costs.

The remainder of this paper is organized as follows. Section II introduces the QTBT partitioning, and Section III explains the overall structure of the proposed method. Experimental results are shown in Section IV, and the conclusions are drawn in the last section.

## 2 | RELATED STUDIES

The block partitioning of HEVC is based on a QT representation and multiple partition concepts, including CU, PU, and TU [14]. The block partitioning structure divides each CTU into four square CUs for optimal intra- or inter-prediction at the CU level. The CU can be divided into smaller CUs based on a QT until the minimum CU size is reached. Each CU is divided again into one, two, or four PUs according to the PU partitioning type. Each PU acquires the residual signal depending on the prediction type based on the optimal prediction and PU division type. The residual signal is divided into TUs according to residual QT transform (RQT) partitioning [15]. The various concepts of HEVC block partitioning exhibit improved performance compared with previous video coding standards [16].

However, the HEVC block partitioning has disadvantages based on the following analysis. The block partitioning loses flexibility because of the square shape of the CU and a few fixed PU types [17]. In addition, RQT for TU leads to slight improvement in performance compared with the significant increase in computational complexity [18] and the complex and redundant concepts of CU, PU, and TU [19]. Finally, the same block partitioning is used for the luminance and chrominance signals; however, the characteristics of luminance and chrominance signals are often diverse. Therefore, the block partitioning for the next-generation video coding standards should generate flexibility for the CU to match the local characteristics of video information. Thus, the QTBT partitioning is proposed and adopted in JEM.

The QTBT partitioning utilizes QT and BT partitioning, as shown in Figure 1. The QT partitioning proposes the division of a single block into four sub-blocks of the same

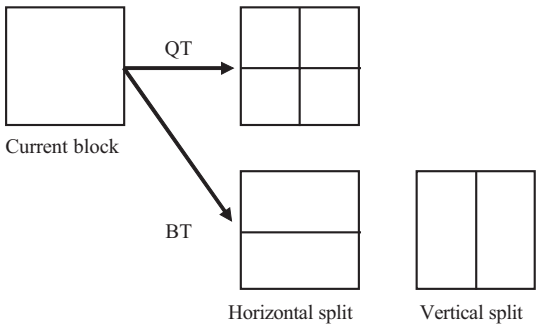


**FIGURE 1** Block partitioning of QT and BT

**TABLE 1** Examples of QTBT parameter sets

| QTBT parameter | I slices | | P and B slices |
|---|---|---|---|
| | Luma | Chroma | |
| CTU size | 128 × 128 | 128 × 128 | 128 × 128 |
| MinQTSize | 16 × 16 | 4 × 4 | 16 × 16 |
| MaxBTSize | 32 × 32 | – | 128 × 128 |
| MaxBTDepth | 4 | 0 | 4 |
| MinBTSize | 4 | – | 4 |

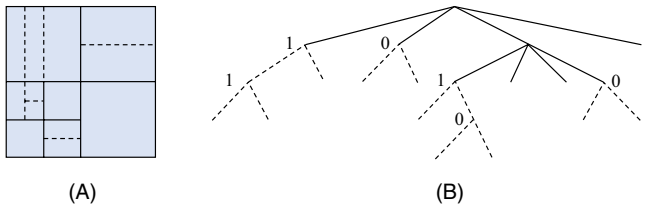


**FIGURE 2** (A) An example of QTBT block partitioning and (B) its corresponding tree structure

size, and the BT partitioning divides a single block into two sub-blocks using a symmetric horizontal division or a symmetric vertical division. In JEM, a CTU is partitioned into four blocks based on QT partitioning. The partitioned blocks are further divided to a minimum size for QT partitioning. If the block size based on QT is not larger than the maximum size for BT partitioning, it is possible to divide the block using BT partitioning. Blocks can be divided by BT partitioning until a minimum size or maximum depth is reached. Table 1 lists parameters that can be set relative to QTBT in JEM. Figure 2A shows an example of QTBT partitioning and Figure 2B shows the corresponding tree structure representing the block partitioning of Figure 2A. In the tree structure, solid lines indicate QT partitioning and dotted lines indicate BT partitioning. As shown in Figure 2, a CTU is first divided by QT partitioning, and each divided node can be recursively partitioned using QT or BT partitioning. Leaf nodes are used for prediction and transformation.

In order to validate the performance of QTBT, results under various experimental conditions in JEM 1.0 were introduced. The results obtained after turning off other tools and applying only QTBT showed a decrease in the average BD-rate by 4.93%, 8.35%, and 8.9% for Y, U, and V in the all intra (AI) configuration, respectively, and the computational complexity increased by 1102% and 123% for the encoder and decoder, respectively. After QTBT was adopted in JEM 3.0, experimental results were obtained for the six test sets shown in Table 2 and are shown in Table 3 [20]. The evaluation was based on the common test conditions of JEM 3.0. In Table 3, the test 1 configuration showed coding efficiency of −0.77%, 3.04%, and 3.27% for Y, U, and V, respectively, with an increase in the encoding complexity of 33%. Test 4 resulted in improved coding efficiency of −0.42%, −1.07%, and −0.95% for Y, U, and V, respectively, with an increase in encoding and decoding complexities of 5% and 8%, respectively. In the remaining tests, the coding complexity was reduced with a coding loss. In tests 2 and 3, even the decoding complexities were increased. In order to improve the high computational complexity of QTBT partitioning, several rapid methods were also proposed in refs [21–27].

## 3 | PROPOSED METHOD

### 3.1 | Analysis and improvement of QTBT

As shown in Table 4, QTBT partitioning provides various configurations and the parameters can be changed via the sequence parameter set (SPS). Depending on the conditions, most settings ascertain whether or not to increase or decrease the number of comparisons during the RDO process of the current block. For example, the maximum BT depth

**TABLE 2** Various test sets to analyze the performance depending on the different QTBT parameter setups

| Parameter | Anchor | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 |
|---|---|---|---|---|---|---|---|
| CTUSize | 128 | 128 | 128 | 256 | 256 | 128 | 256 |
| MinQTLumaISlice | 8 | 16 | 8 | 8 | 8 | 8 | 8 |
| MinQTChromaISlice | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| MinQTNonISlice | 8 | 16 | 8 | 8 | 8 | 8 | 8 |
| MaxBTDepthISliceL | 3 | 4 | 2 | 2 | 3 | 2 | 2 |
| MaxBTDepthISliceC | 3 | 0 | 2 | 2 | 3 | 0 | 0 |
| MaxBTDepth | 3 | 4 | 2 | 2 | 3 | 2 | 2 |
| MaxBTSizeISliceL | 32 | 32 | 32 | 32 | 32 | 32 | 32 |
| MaxBTSizeISliceC | 16 | 16 | 16 | 16 | 16 | 16 | 16 |
| MaxBTSizeBPSlice | 128 | 128 | 128 | 128 | 128 | 128 | 128 |

**TABLE 3** Experimental results of various test sets in Table 2

| Num. of test sets | Coding performance (%) (Y, U, V BD-rate (avg.)) | Enc. time (%) | Dec. time (%) |
|---|---|---|---|
| Test 1 | −0.77/3.04/3.27 | 133 | 100 |
| Test 2 | 1.46/1.93/1.97 | 66 | 107 |
| Test 3 | 1.28/0.89/1.13 | 66 | 106 |
| Test 4 | −0.43/−1.07/−0.95 | 105 | 108 |
| Test 5 | 1.37/5.90/6.08 | 63 | 100 |
| Test 6 | 1.23/4.82/5.21 | 63 | 100 |

**TABLE 4** Example parameters related QTBT partitioning

| Parameter | Description | SPS value |
|---|---|---|
| CTUSize | Size of the CTUs | 128 |
| MinQTLumaISlice | Minimum luma QT Size on I slices | 8 |
| MinQTChromaISlice | Minimum chroma QT Size on I slices | 4 |
| MinQTNonISlice | Minimum QT Size on P and B slices | 8 |
| MAX_BT_DEPTH | Maximum BT depth | 4 |
| MAX_BT_SIZE | Maximum BT size | 32 |
| MIN_BT_SIZE | Minimum BT size | 4 |
| MAX_BT_DEPTH_C | Maximum chroma BT depth | 0 |
| MAX_BT_SIZE_C | Maximum chroma BT size | 64 |
| MIN_BT_SIZE_C | Minimum chroma BT size | 4 |

(MaxBTDepth) was set to 3 under the common test condition (CTC) [28] and the BT partitioning was changed to 3 from a QT. When MaxBTDepth was changed to 4, the BT partitioning was split four times from a QT and facilitated comparison to a smaller area. Thus, increasing the number of comparisons reduces the distortion of the current block; however, the computational complexity in the RDO process also increases. To measure the coding performance based on the parameters related to QTBT partitioning, several experiments were conducted, as shown in Tables 5 and 6. In the experiments, the common test sequences of the JVET Call for Evidence (CfE) [29] including Ultra High Definition (UHD) and High Definition (HD) resolutions were used, and the encoder control followed the JEM CTC.

Table 5 shows the experimental results of JEM 6.0 [30] AI configuration according to the MaxBTDepth. The coding efficiency of luminance varied under constant BT depth of luminance and varying BT depth of chrominance. The degree of improvement in coding efficiency decreased as the luminance BT depth was higher and almost saturated as the BT depth of luminance was close to BT depth 4 of chrominance. According to BT depth, the BD-rate of Y decreased from

**TABLE 5** Experimental results of JEM 6.0 depending on luminance and chrominance BT depth (JEM 6.0 Common Test Condition, AI configuration, Max. luma BT depth = 3, Max. chroma BT depth = 3)

| Max. BT depth | | BD-rate (%) | | | Time (%) | |
|---|---|---|---|---|---|---|
| Luma | Chroma | Y | U | V | Enc. | Dec. |
| 3 | 4 | 0.02 | −1.33 | −1.48 | 105 | 102 |
| | 3 | 0 | 0 | 0 | 100 | 100 |
| | 2 | 0.02 | 1.84 | 1.80 | 100 | 101 |
| | 1 | 0.05 | 5.3 | 5.36 | 99 | 100 |
| | 0 | 0.06 | 12.43 | 12.72 | 98 | 101 |
| 2 | 2 | 0.54 | 1.98 | 1.96 | 74 | 100 |
| | 1 | 0.56 | 5.49 | 5.63 | 67 | 101 |
| | 0 | 0.57 | 12.65 | 12.92 | 65 | 101 |
| 1 | 1 | 1.83 | 6.05 | 6.18 | 33 | 100 |
| | 0 | 1.83 | 13.22 | 13.51 | 31 | 99 |

1.83% to 0.02% compared with the anchor at a BT depth of 3. Encoding times changed dramatically from 31% to 105%, whereas decoding times were approximately similar to that of the JEM 6.0 CTC, which suggests that the parameters related to QTBT significantly affect coding efficiency and encoding time but not the decoding complexity [31].

Table 6 shows several trade-offs between coding efficiency and complexity according to minimum QT size (MinQTSize), maximum BT depth (MaxBTDepth), and maximum BT size (MaxBTSize). In the experiments, the three parameters were changed as follows from the CTC setting: MinQTSize was set from 8 to 4, MaxBTDepth was set from 3 to 4, and MaxBTSize ranged from 32 to 64. The change in MinQTSize suggests that all experiments in Table 6 exhibit a marginal impact on coding efficiency and complexity. The MaxBTSize change showed a coding efficiency of −0.53% BD-rate for Y and the MaxBTDepth change showed the best coding efficiency of −1.25% and −1.34% BD-rate for U and V, respectively. The encoding complexity of the MaxBTDepth change was about 166%, which was 26% higher than the MaxBTSize change of 140%. In addition, the experimental results related to changes in MaxBTDepth and the size showed significant improvements in coding efficiency of −1.03%, −1.84%, and −2.10% BD-rate for Y, U, and V, respectively, but the encoding complexity increased by about 282%. A significant difference in trade-off was observed in coding performance and complexity according to the QTBT settings [32].

In addition, the QTBT partitioning supports dual and single trees depending on the slice type. A dual tree is used to signal the respective RDO results of luminance and chrominance block partitioning in I slices. In P and B slices, a single tree is used to signal the RDO results based on both luminance and chrominance block partitioning. In order to verify

**TABLE 6** Experimental results of JEM 6.0 depending on QTBT parameters (JEM 6.0 Common Test Condition, All Intra configuration)

| QTBT parameters (Anchor: MinQTSize = 8, MaxBTDepth = 3, MaxBTSize = 32) | BD-rate (%) | | | Enc. time (%) |
|---|---|---|---|---|
| | Y | U | V | |
| MinQTSize = 4 | 0.02 | 0.03 | 0.05 | 103.24 |
| MaxBTDepth = 4 | −0.28 | −1.25 | −1.34 | 165.95 |
| MaxBTSize = 64 | −0.53 | −0.41 | −0.47 | 140.67 |
| MinQTSize = 4, MaxBTDepth = 4 | −0.29 | −1.23 | −1.39 | 186.64 |
| MaxBTDepth = 4, MaxBTSize = 64 | −1.03 | −1.84 | −2.10 | 282.02 |
| MinQTSize = 4, MaxBTSize = 64 | −0.52 | −0.47 | −0.40 | 140.14 |
| MinQTSize = 4, MaxBTDepth = 4, MaxBTSize = 64 | −1.03 | −1.83 | −2.00 | 283.09 |

the characteristics depending on the tree types, the number of bin strings for partitioning luminance and chrominance regions was measured and compared depending on the AI and RA configurations of JEM 6.0 CTC, as shown in Table 7. The experimental environment is similar to the experiment described above.

Table 7 shows that the number of bin strings for partitioning the chrominance region in the AI configuration was significantly higher than in the RA configuration. In most cases, the number of bin strings for partitioning the luminance region in the AI configuration showed a much higher usage of bin strings than the RA configuration because the dual tree resulted in a more precise division that reflected the spatial characteristics of each region based on the RDO of the luminance and chrominance regions, respectively. In other words, when the partition similarity of luminance and chrominance was low, the coding performance can be improved by increasing the accuracy of the division for each region in the dual tree. However, when the partitioning of luminance and chrominance was similar, the dual tree exhibited redundant partitioning signals.

Based on the performance analysis, we propose an adaptive selection method between the two trees to improve the coding efficiency of QTBT partitioning in I slices. The proposed method can be used to select the best tree with a lower RD cost. In the proposed method, the number of partitioning signals was measured to verify the reduction in redundant signals. In addition, a different setting for partitioning according to the tree types is further proposed using the two trees together. The change in MaxBTDepth settings, which

**TABLE 7** Rate of increase in the number of bins for QT and BT on JEM 6.0 AI condition in comparison with JEM 6.0 RA (AI, JEM 6.0 (Anchor: JEM 6.0 RA))

| Sequence (JVET CfE) | QT split bins (%) | | | BT split bins (%) | | |
|---|---|---|---|---|---|---|
| | Total (a + b) | Luma (a) | Chroma (b) | Total (a + b) | Luma (a) | Chroma (b) |
| Resolution: UHD | | | | | | |
| Crosswalk1 | 215 | 157 | 770 | 146 | 102 | 649 |
| FoodMarket3 | 229 | 165 | 769 | 157 | 94 | 671 |
| Tango1 | 171 | 139 | 669 | 168 | 141 | 632 |
| CatRobot1 | 239 | 183 | 609 | 245 | 195 | 586 |
| DayLightRoad | 215 | 185 | 660 | 234 | 212 | 616 |
| BuildingHall1 | 282 | 206 | 611 | 319 | 247 | 590 |
| ParkRunning | 139 | 71 | 653 | 127 | 62 | 649 |
| CampfireParty | 138 | 103 | 688 | 138 | 113 | 644 |
| Resolution: HD | | | | | | |
| BQTerrace | 193 | 154 | 579 | 205 | 141 | 1133 |
| RitualDance | 154 | 112 | 758 | 142 | 110 | 718 |
| Timelapse | 226 | 192 | 645 | 255 | 230 | 600 |
| BasketballDrive | 163 | 114 | 680 | 163 | 126 | 639 |
| Cactus | 223 | 170 | 644 | 249 | 208 | 637 |
| Average | 199 | 150 | 672 | 196 | 152 | 674 |

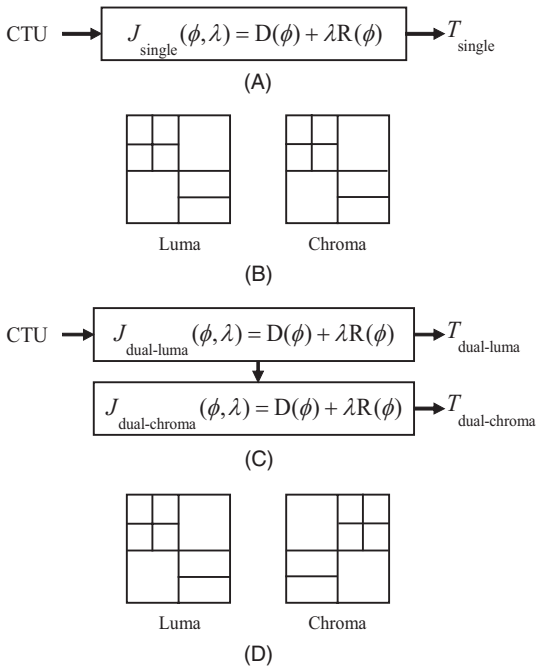


**FIGURE 3** Block tree generation structures and partitioning result examples depending on the slice type. (A) single tree generation on P and B Slices, (B) single tree partitioning results, (C) dual tree generation structure on I slices, and (D) dual tree partitioning results

showed the highest coding performance in the experimental results of Table 6, was applied to the single tree of the proposed method and the experimental results were analyzed.

## 3.2 | Adaptive block tree structure (ABTS)

The peak signal-to-noise ratio (PSNR), which utilizes the mean squared error of degraded images compared with original images, is a typical method for objective evaluation of the image quality in video compression. The PSNR is a numerical representation of the ratio of distortion to the maximum value of the signal. When the same codec is used, it is possible to compare the error loss of each image with the PSNR of the compressed images. In order to evaluate the compression performance of a video codec, RD is used to consider both the PSNR and the bit rate of the final compressed stream [33,34]. RD is a method used to evaluate the compression performance and is also used as a reference to increase the compression performance in the encoder process. During block compression, the encoder selects an optimal compression method according to the relationship between the block characteristics and the surrounding image, which is designated as RDO. During the encoding, RDO is calculated by

$$J(\phi, \lambda) = D(\phi) + \lambda R(\phi), \tag{1}$$

where $J$ denotes the RD cost function for the comparison of each candidate model, $D$ is the block distortion caused by quantization, and $R$ stands for the number of bits required. $\lambda$ is the Lagrangian multiplier that is used to control the target bit rate and $\phi$ is the prediction mode.

In JEM with QTBT partitioning, the block tree is determined by the slice type. The structure and partitioning results of the block tree generated are illustrated in Figure 3. Here, $J_{\text{dual-luma}}$, $J_{\text{dual-chroma}}$, and $J_{\text{single}}$ denote the RDO costs of a luminance region in a dual tree, a chrominance region in a dual tree, and a single tree, respectively. Figure 3A presents a structure that compares $J_{\text{single}}$ by performing an RDO and then generates $T_{\text{single}}$, which is the optimal single tree. As shown in Figure 3B, a single tree shares a block tree and exhibits the same partitioning of the luminance and chrominance regions. Figure 3C presents a structure that sequentially performs RDO using $J_{\text{dual-luma}}$ and $J_{\text{dual-chroma}}$ for the luminance and chrominance regions in I slices and generates the optimal dual tree $T_{\text{dual-luma}}$ and $T_{\text{dual-chroma}}$. As shown in Figure 3D, the dual tree carries a different partitioning of the luminance and chrominance.

As described in the previous section, the coding performance and complexity of each tree differ depending on the similarity between the luminance and chrominance regions and the characteristics of the two trees. Therefore, we propose an adaptive block tree structure (ABTS) to select the best tree with a lower RD cost instead of a block tree based on the slice type of QTBT partitioning. Figure 4 shows the ABTS for selection of a block tree based on RD cost competition in I slices. In order to split a CTU, the RDO of single and dual trees is sequentially performed. After completing all the RDOs for both trees for a CTU, the tree with a lower

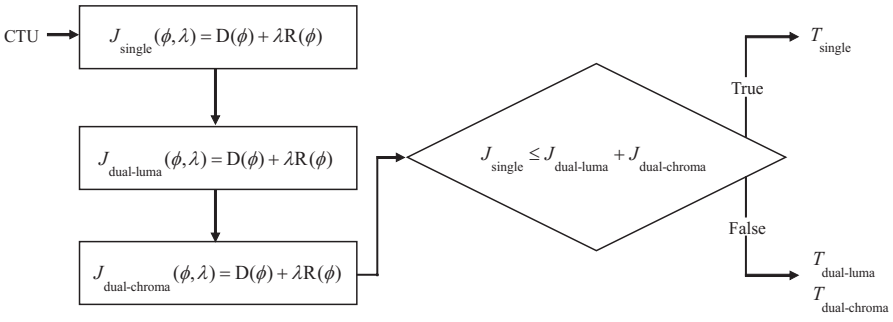


**FIGURE 4** Best tree decision structure of the ABTS based on the RD cost comparison on I slices

RD cost $J_{single}$ or $J_{dual\text{-}luma} + J_{dual\text{-}chroma}$ is determined to be the best tree. The proposed method generates a flag at the CTU level to inform the decoder of the best tree information selected. In the case of the added signal, a frame is delayed at the slice level, and when generated at the CU level, the signals increase significantly. Therefore, it is appropriate to generate the signal at the CTU level.

In contrast to the method of using a block tree according to the slice type in the QTBT partitioning, the proposed method uses two block trees in I slices. The structure of the proposed method can set different values of a parameter related to QTBT for the single and the dual trees. Based on this structure, an adaptive block tree structure with a different setting (ABTSds) can be used to improve the coding efficiency of ABTS. The proposed ABTSds sets MaxBTSize, which shows the most efficient trade-off by altering the value of parameters in the previous section, as shown in Table 6, to 32 and 64 in the dual and the single trees, respectively. Compared to the dual tree that performs independent RDO for luminance and chrominance regions, the single tree that generates a block tree based on luminance and chrominance regions exhibits a relatively low coding complexity. Therefore, the CTC value of 32 was set for the dual tree, and a value of 64 with increasing complexity was applied to the single tree to minimize the overall increase in coding complexity of the proposed method.

## 4 | EXPERIMENTAL RESULTS

In this section, we evaluate the impact of ABTS, ABTSds, and Fast ABTS and its behavior using the common test sequence JVET Call for Evidence (CfE), which consists of eight UHD test sequences and five HD test sequences. We also show how different signals were related to the QTBT partitioning in JEM 6.0 and the ABTS. All of these methods were implemented on the top of JEM 6.0, and the JEM 6.0 CTC was referenced as an anchor to compare the proposed methods. Encoder controls follow the JEM CTC with the AI Main configuration. All presented bit-rate savings and PSNR improvement were measured in terms of the BD-rate. The values of the BD-rate were calculated from a total of four rate points as defined in the JEM 6.0 CTC. To compare the performance with respect to computational complexity, the encoding times were measured according to the encoding time ratio (ETR) as follows:

$$\mathrm{ETR}(\%) = (\mathrm{ET}_{test}/\mathrm{ET}_{ref}) \times 100, \quad (2)$$

where $\mathrm{ET}_{test}$ and $\mathrm{ET}_{ref}$ refer to the encoding times of JEM 6.0 and a test method, respectively.

Additionally, the signals generated in order to inform the selected tree information to the decoder in all the proposed methods were signaled via context-adaptive binary arithmetic coding [35]. Multiple direct modes [36], a technology related to chrominance intra-coding, was excluded from the anchor and all the proposed methods. The evaluations were conducted in a computing environment exhibiting similar computing power.

### 4.1 | ABTS

Table 8 shows the overall performance of the ABTS under the same QTBT configuration for the dual and the single trees. The test results showed an average Y reduction of 0.24% in BD-rate with 225% ETR and almost similar decoding time relative to the anchor. In terms of complexity, that of the encoder was greatly increased by the sequential RDO of the two trees; however, the complexity of the decoder was almost similar because it selectively used only a single block tree. The increased encoder complexity can be reduced to a similar level to the anchor because the RDOs for each tree can be processed in parallel. Compared with the anchor, which only selects the dual tree on I slices, the proposed method selected 26.5% of the total number of CTUs as single trees. In particular, the proposed method yielded higher coding gains for UHD sequences with an average Y BD-rate reduction of 0.30% compared with an average Y BD-rate reduction of 0.16% for HD sequences. Table 8 shows a 31.6% single tree selection ratio, which was higher in the UHD sequences compared with the 18.3% single tree selection ratio of the HD sequences. Moreover, the single tree selection ratios according to the QP were 13.95%, 23.23%, 32.27%, and 38.6% at 22, 27, 32, and 37, respectively. These results indicate that sequences with higher resolution and QP values contain substantially higher cases of similar block segmentation results for luminance and chrominance regions, which resulted in higher coding gain when the single tree was selected according to the proposed method. As shown in Table 9, the single tree selection ratios in UHD sequences with higher QPs were greatly increased compared with HD sequences. Therefore, in the case of bigger sequences, images with a larger background or objects were divided more frequently into the entire

**TABLE 8** Overall ABTS performance applying the same QTBT configuration for each tree (ABTS with the same QTBT configuration, AI configuration (Anchor: JEM 6.0 CTC, AI configuration))

| | BD-rate (%) | | | ETR (%) | Single tree selection ratio (%) |
|---|---|---|---|---|---|
| | Y | U | V | Enc. | |
| Overall | −0.24 | 0.27 | 0.23 | 225 | 26.5 |
| UHD | −0.30 | 0.31 | 0.10 | 225 | 31.6 |
| HD | −0.16 | 0.20 | 0.44 | 225 | 18.3 |

**TABLE 9** Single tree selection ratios depending on the resolution of the sequence and the QP configurations

| QP | Single tree selection ratio (%) |
|---|---|
| Resolution: UHD | |
| 22 | 15.7 |
| 27 | 27.8 |
| 32 | 36.9 |
| 37 | 45.8 |
| Resolution: HD | |
| 22 | 11.0 |
| 27 | 15.8 |
| 32 | 19.6 |
| 37 | 26.9 |

region of a CTU with size similar to all sequence sizes. In the case of higher QPs, since the high-frequency component was substantially lost in the encoding process, the partitioning similarity with the luminance and chrominance regions increased because of the low precision of block partitioning.

The coding gain of the proposed method was attributed to a decline in bit rate rather than PSNR improvement in most experiments. In order to confirm the decreased bit rate, the number of bins related to the QTBT partitioning was measured by the anchor and the ABTS, as shown in Table 10. The proposed method generated an average of about 9% fewer bins for the QTBT partitioning compared with the anchor. The experiment shows that the overhead of the partitioning signal was actually reduced. In addition, the additional signal at the CTU level generated in the proposed method of transmission of tree selection information to the decoder may cause overhead. However, the reduction in BD-rate occurred because the proposed method resulted in further reduction in the partitioning signals. Overall, several experiments confirm that a partitioning structure using a block tree according to the slice type can be improved using multiple block trees according to the regional characteristics of luminance and chrominance.

## 4.2 | ABTS with different QTBT configurations

In the experiment involving the ABTSds, three parameters related to the QTBT partitioning were set differently for the single and the dual trees. Table 11 shows several trade-offs in coding efficiency and complexity depending on the ABTSds settings according to MinQTSize, MaxBTDepth, and MaxBTSize. The JEM 6.0 CTC configuration was used for the dual tree, and the three parameters were changed as follows from the CTC settings for the single tree: MinQTSize was set from 8 to 4, MaxBTDepth was set from 3 to 4, and MaxBTSize was set from 32 to 64.

Similar to the experiment results shown in Table 6, the MinQTSize suggests that individual and combined changes with other parameters have no impact on the coding

| Sequence (JVET CfE) | QT split syntax (%) | | | BT split syntax (%) | | |
|---|---|---|---|---|---|---|
| | Total | Luma | Chroma | Total | Luma | Chroma |
| Resolution: UHD | | | | | | |
| Crosswalk1 | 70 | 86 | 40 | 91 | 115 | 47 |
| FoodMarket3 | 63 | 77 | 39 | 82 | 114 | 45 |
| Tango1 | 88 | 93 | 74 | 95 | 99 | 79 |
| CatRobot1 | 84 | 95 | 62 | 89 | 100 | 64 |
| DayLightRoad | 92 | 95 | 77 | 97 | 99 | 86 |
| BuildingHall1 | 75 | 95 | 44 | 80 | 101 | 47 |
| ParkRunning | 91 | 98 | 85 | 93 | 102 | 86 |
| CampfireParty | 90 | 99 | 69 | 97 | 103 | 75 |
| Resolution: HD | | | | | | |
| BQTerrace | 96 | 97 | 91 | 99 | 127 | 47 |
| RitualDance | 83 | 94 | 60 | 91 | 100 | 66 |
| Timelapse | 95 | 97 | 86 | 98 | 99 | 90 |
| BasketballDrive | 69 | 81 | 46 | 76 | 86 | 50 |
| Cactus | 94 | 98 | 85 | 97 | 100 | 87 |
| Average | 199 | 84 | 93 | 66 | 91 | 103 |

**TABLE 10** Rate of increase in the number of bins for QT and BT on the ABTS in comparison with the JEM 6.0 AI configuration (AI, JEM 6.0 (Anchor: JEM 6.0 RA))

**TABLE 11** Overall ABTS performances applying the different QTBT configuration for each tree (ABTS with the different QTBT configuration, AI configuration (Anchor : JEM 6.0 CTC, AI configuration))

| QTBT parameters for single tree (Anchor : MinQTSize = 8, MaxBTDepth = 3, MaxBTSize = 32) | BD-rate (%) | | | ETR (%) | Single tree selection ratio (%) |
|---|---|---|---|---|---|
| | Y | U | V | | |
| The ABTS with the same QTBT configuration | −0.24 | 0.27 | 0.23 | 225 | 26.5 |
| MinQTSize = 4 | −0.23 | 0.29 | 0.34 | 226 | 26.4 |
| MaxBTDepth = 4 | −0.32 | 0.62 | 0.45 | 345 | 31.3 |
| MaxBTSize = 64 | −0.69 | 0.68 | 0.71 | 270 | 40.8 |
| MinQTSize = 4, MaxBTDepth = 4 | −0.32 | 0.59 | 0.53 | 305 | 31.3 |
| MaxBTDepth = 4, MaxBTSize = 64 | −0.93 | 0.84 | 0.81 | 403 | 49.5 |
| MinQTSize = 4, MaxBTSize = 64 | −0.68 | 0.60 | 0.59 | 268 | 40.5 |
| MinQTSize = 4, MaxBTDepth = 4, MaxBTSize = 64 | −0.92 | 0.79 | 0.76 | 405 | 49.6 |

performance compared to ABTS. The experimental results with altered MaxBTSize showed an average Y BD-rate reduction of 0.69% with 270% ETR, which suggested a significantly enhanced coding performance compared with the change in MaxBTDepth with an average Y BD-rate reduction of 0.32% and 345% ETR. Experiments containing more than two parameters showed ineffective trade-offs with 305%–405% in the ETR. In particular, when the MaxBTDepth and MaxBTSize were changed together, an average Y BD-rate reduction was −0.93% with 403% ETR, which was much less efficient than the altered MaxBTsize. Therefore, we confirmed the maximum efficiency of coding performance when the parameter changes in MaxBTSize were used in the various ABTSds experiments.

In addition, in the results of all experiments, the coding efficiency of the chrominance region was reduced with respect to the ABTS. Therefore, the RDO of a single tree was further compared with that of a dual tree according to parameter changes in the single tree. Thus, the coding performance of the single tree was improved and the selection ratio of the single tree increased at the optimal tree decision stage of the ABTS. As a result, the performance of the chrominance region was reduced in the experiments by decreasing the selection ratio of the dual tree, resulting in enhanced coding efficiency of the chrominance region.

## 5 | CONCLUSION

An ABTS was presented in this paper to improve the QTBT partitioning structure. The proposed tree decision using RD cost comparison of the dual and single trees resulted in bit-saving, suggesting improved partitioning of the structure using a block tree based on slice type. This approach minimizes redundant signals generated in the luminance and chrominance regions in I slices. In addition, different block partitioning configurations for each tree improve the coding efficiency of the proposed method. The experimental results showed an average reduction of 0.24% in Y BD-rate with 225% encoding complexity without affecting the decoding complexity when compared with the intra-coding performance of JEM 6.0 CTC. In addition, the encoding complexity can be reduced to a level similar to that of the anchor through parallel processing.

## ORCID

*Jinho Lee* https://orcid.org/0000-0002-7558-5731
*Haechul Choi* https://orcid.org/0000-0002-7594-0828

## REFERENCES

1. High efficiency video coding (HEVC), Rec, ITU-T H.265 and ISO/IEC 23008–2, Jan. 2013.
2. I. Kim et al., *Block partitioning structure in the HEVC standard*, IEEE Trans. Circuits Syst. Video Technol. **22** (2012), 1697–1706.
3. T. Wiegand et al., *Overview of the H.264/AVC video coding standard*, IEEE Trans. Circuits Syst. Video Technol. **13** (2003), 560–576.
4. ITU-T/ISO/IEC JVET, *Algorithm description of joint exploration test model 1*, JVET-A0001, Oct. 2015.
5. ITU-T SG16, *Block partitioning structure for next generation video coding*, COM16-C966, Sept. 2015.
6. G. J. Sullivan and R. L. Baker, *Efficient quadtree coding of images and video*, IEEE Trans. Image Process. **3** (1994), 327–331.
7. P. Salembier and L. Garrido, *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval*, IEEE Trans. Image Process. **9** (2000), 561–576.
8. ITU-T/ISO/IEC JVET, *Report of AHG on JEM software development*, JVET-B0006, Feb. 2016.
9. ITU-T/ISO/IEC JVET, *Quadtree plus binary tree structure integration with JEM tools*, JVET-B0023, Feb. 2016.
10. ITU-T/ISO/IEC JVET, *Grouped signaling for transform in QTBT*, JVET-C0054, May. 2016.
11. ITU-T/ISO/IEC JVET, *Multi-type-tree*, JVET-D0117, Jan. 2016.
12. ITU-T/ISO/IEC JVET, *Asymmetric coding units in QTBT*, JVET-D0064, Jan. 2016.

13. ITU-T/ISO/IEC JVET, *Report of AHG3 on JEM software development*, JVET-D0003, Oct. 2016.
14. G. J. Sullivan et al., *Overview of the High Efficiency Video Coding (HEVC) standard*, IEEE Trans. Circuits Syst. Video Technol. **22** (2012), 1648–1667.
15. M. Winken et al., *Transform coding in the HEVC test model*, in Proc. IEEE Int. Conf. Image Process (Brussels, Belgium), Sept. 2011, pp. 3693–3696.
16. J.-R. Ohm et al., *Comparison of the coding efficiency of video coding standards including High Efficiency Video Coding (HEVC)*, IEEE Trans. Circuits Syst. Video Technol. **22** (2012), 1668–1683.
17. I.-K. Kim et al., *Coding efficiency improvement of HEVC using asymmetric motion partitioning*, in Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (Seoul, Rep. of Korea) June 2012, pp. 1–4.
18. Y.-H. Tan et al., *On residual quad-tree coding in HEVC*, in Proc. IEEE Int. Workshop Multimedia Signal Process. (Hangzhou, China), Oct. 2011, pp. 1–4.
19. S. Cho and M. Kim, *Fast CU splitting and pruning for suboptimal CU partitioning in HEVC intra coding*, IEEE Trans. Circuits Syst. Video Technol. **23** (2013), 1555–1564.
20. ITU-T/ISO/IEC JVET, *AHG5: Experiment on JEM default setting*, JVET-D0052, Oct. 2016.
21. ITU-T/ISO/IEC JVET, *AHG5: Fast encoding setting for JEM*, JVET-D0053, Oct. 2016.
22. ITU-T/ISO/IEC JVET, *AHG5: Fast QTBT encoding configuration*, JVET-D0095, Oct. 2016.
23. ITU-T/ISO/IEC JVET, *AHG5: Improved fast encoding setting*, JVET-E0023, Jan. 2017.
24. ITU-T/ISO/IEC JVET, *AHG5: Improved fast algorithm in JEM-4.0*, JVET-E0078, Jan. 2017.
25. T. Lin et al., *Fast binary tee partition decision in H.266/FVC intra coding*, in Proc. IEEE Int. Conf. Consumer Electron. (Taichung, Taiwan), May 2018, https://doi.org/10.1109/ICCE-China.2018.8448619.
26. C. Li, C. Li, and J. Liu, *Fast intra candidate selection and CU split in intra prediction for Future Video Coding*, in Proc. IEEE Int. Conf. Safety Produce Inform. (Chongqing, China), Dec. 2018, pp. 723–727.
27. A. Wieckowski et al., *Fast partitioning decision strategies for the upcoming Versatile Video Coding (VVC) standard*, in Proc. IEEE Int. Conf. Image Process. (Taipei, Taiwan), Sept. 2019, pp. 4130–4134.
28. ITU-T/ISO/IEC JVET, *JVET common test conditions and software reference configurations*, JVET-G1010, Aug. 2017.
29. ITU-T/ISO/IEC JVET, *Joint call for evidence on video compression with capability beyond HEVC*, JVET-F1002, Apr. 2017.
30. ITU-T/ISO/IEC JVET, *Report of AHG3 on JEM software development*, JVET-G0003, Aug. 2017.
31. A. Baek et al., *QTBT performance analysis according to binary tree depth in joint exploration model*, in Proc. Int. Conf. Electron., Electr. Eng., Comput. Sci. (Croatia), July 2017, pp. 74–75.
32. A. Baek et al., *QTBT performance analysis according to maximum binary tree size*, in Proc. Int. Conf. Electron., Electr. Eng., Comput. Science (EEECS), Cambodia, Vol 5, Feb. 2018.
33. G. Bjøntegaard, *Calculation of average PSNR differences between RD curves*, document *VCEG-M33*, ITU-T Q6/16, Apr. 2001.
34. G. Bjøntegaard, *Improvements of the BD-PSNR model*, *document VCEG-AI11*, ITU-T Q6/16, July 2008.
35. D. Marpe, H. Schwarz, and T. Wiegand, *Context-adaptive binary arithmetic coding in the H.264/AVC video compression standard*, IEEE Trans. Circuits Syst. Video Technol. **13** (2003), 620–636.
36. ITU-T/ISO/IEC JVET, *Multiple Direct Modes for chroma intra coding*, JVET-D0111, Oct. 2016.

## AUTHOR BIOGRAPHIES

**Aram Baek** received his BS, MS, and PhD degrees in multimedia engineering in 2012, 2014, and 2019, respectively, from Hanbat National University, Daejeon, Rep. of Korea. From 2019 to 2020, he was a Senior Researcher for Pixtree, Seoul, Rep. of Korea. He is currently a postdoctoral fellow at Hanbat National University, Daejeon, Rep. of Korea. His research interests are image processing, video coding, and computer vision.

**Daehyeok Gwon** received his BS, MS, and PhD in multimedia engineering from Hanbat National University, Daejeon, Rep. of Korea, in 2012, 2014, and 2020 respectively. He is currently a Postdoctoral Fellow at Hanbat National University, Daejeon, Rep. of Korea. His current research areas include video coding, image processing, image compression, signal processing, and computer vision.

**Sohee Son** received her BS and MS degrees in multimedia engineering in 2015 and 2017, respectively, from Hanbat National University, Daejeon, Rep. of Korea. Since 2017, she has been a PhD candidate in multimedia engineering, Hanbat National University. Her research interests are video coding, parallel processing, and computer vision.

**Jinho Lee** received his BS degree in electrical electronics engineering from Korea University, Sejong, Rep. of Korea, in 2007, and his MS degree in telecommunication and digital broadcasting engineering from University of Science and Technology, Daejeon, Rep. of Korea in 2009. Since 2009, he has been a Senior Researcher at the Electronics and Telecommunications Research Institute, Daejeon, Rep. of South Korea. His research interests are video coding and immersive video.

**Jung-Won Kang** received her BS and MS degrees in electrical engineering in 1993 and 1995, respectively, from Korea Aerospace University, Seoul, Rep. of Korea. She received her PhD in electrical and computer engineering in 2003 from the Georgia Institute of Technology, Atlanta, GA, US. Since 2003, she has been a Principal Member of the research staff in the Broadcasting Media Research Laboratory, ETRI, Rep. of Korea. Her research interests are in the areas of video signal processing and video compression.

**Hui Yong Kim** received his BS, MS, and PhD degrees from KAIST in 1994, 1998, and 2004, respectively. From 2003 to 2005, he worked for AddPac Technology Co. Ltd., Seoul, Rep. of Korea as the Leader of the Multimedia Research Team. From 2005 to 2019, he joined ETRI, Daejeon, Rep. of Korea and served as the Managing Director of Realistic Audio and Video Research Group. Since 2020, he has been with the Department of Computer Science and Engineering in Kyung Hee University, Yongin, Rep. of Korea, as an Associate Professor. He has been an active technology contributor, editor, and ad-hoc group chair in developing several international standards including MPEG Multimedia Application Format, ITU-T/ISO/IEC JCT-VC High Efficiency Video Coding and JVET Versatile Video Coding. His current research interests include image and video signal processing and compression for realistic media applications such as UHD, 3D, VR, HDR, and digital holograms.

**Haechul Choi** received his BS in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1997, and his MS and PhD in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1999 and 2004, respectively. He is a Professor in information and communication engineering at Hanbat National University, Daejeon, Korea. From 2004 to 2010, he was a Senior Member of the Research Staff in the Broadcasting Media Research Group of the Electronics and Telecommunications Research Institute (ETRI). His current research interests include image processing, video coding, and computer vision.