Edge-Based Intra Mode Selection for Depth-Map Coding in 3D-HEVC

Chun-Su Park

Abstract—The 3D video extension of High Efficiency Video Coding (3D-HEVC) is the state-of-the-art video coding standard for the compression of the multiview video plus depth (MVD) format. In the 3D-HEVC design, new depth-modeling modes (DMMs) are utilized together with the existing intra prediction modes for depth intra coding. The DMMs can provide more accurate prediction signals and thereby achieve better compression efficiency. However, testing the DMMs in the intra mode decision process causes a drastic increase in the computational complexity. In this paper, we propose a fast mode decision algorithm for depth intra coding. The proposed algorithm first performs a simple edge classification in the Hadamard transform domain. Then, based on the edge classification results, the proposed algorithm selectively omits unnecessary DMMs in the mode decision process. Experimental results demonstrate that the proposed algorithm speeds up the mode decision process by up to 37.65% with negligible loss of coding efficiency.

Index Terms—3D-HEVC, mode decision, depth maps, depth-modeling modes, Hadamard transform, intra coding.

I. INTRODUCTION

The popularity of 3D video services has lead to the efficient representation of 3D video data being actively investigated over the past few years [1]-[3]. The multiview video plus depth (MVD) format is one of the most promising methods for 3D video representation. In the MVD format, a small number of videos captured from different viewpoints and associated depth maps are coded and multiplexed into a 3D video bitstream. After the video and depth data have been decoded, additional intermediate views can be synthesized using a depth-image-based rendering (DIBR) technique [4], [5].

The 3D extension of High Efficiency Video Coding (3D-HEVC), which is being developed by the Joint Collaborative Team on 3D Video Coding Extension Development (JCT-3V), is the state-of-the-art video coding standard for the compression of MVD data [6]-[9]. The 3D-HEVC standard introduces new prediction techniques to improve the efficiency of MVD data compression. Especially, in the 3D-HEVC design, the dependent view prediction and the coding of depth maps have been actively studied [9].

In contrast to natural video, depth maps are characterized by sharp edges and large regions with nearly constant values [10]. While the existing intra prediction of HEVC is well-suited for nearly constant regions, it may result in significant coding artifacts at sharp edges. For a better representation of edges in depth maps, two depth-modeling modes (DMMs) have been added to the 3D-HEVC standard [6]. In the DMMs, a depth block is approximated by a model that divides the block into two non-rectangular regions and each region is represented by a constant partition value (CPV). Note that two partition types are used, namely Wedgelets for segmentations using a straight line and Contours for arbitrary segmentations [11]. Table I specifies the value for each intra prediction mode and its associated name in 3D-HEVC, where the mode numbers 35 and 36 are assigned to the DMMs.

It was presented in [8] that testing the DMMs in the mode decision process introduces a huge computational load. For use in practical applications such as high-resolution 3D television, the complexity of the DMMs should be reduced significantly without sacrificing coding efficiency. Only a small number of algorithms have been proposed for reducing the complexity of the DMMs [12]-[14]. In [12], Zhang et al. proposed an algorithm that accelerates the Wedgelet search process of the DMMs. The algorithm presented in [12] classifies all Wedgelet patterns into several sets and examines only the patterns in the most probable set in the mode decision. The authors of [13] proposed a fast algorithm for simplifying the mode decision process. Using the most probable mode (MPM) as the indicator, the algorithm selectively omits the rate-distortion (RD) cost calculation of the DMMs. If the MPM is INTRA_PLANAR, the algorithm does not test the DMMs in the mode decision. Recently, to further enhance the performance of the algorithm in [13], an improved mode decision algorithm was introduced in [14]. The algorithm proposed in [14] determines whether to test the DMMs by using the pixel variance of the target block as well as the MPM. The state-of-the-art algorithm in [14] has been implemented in the reference software of 3D-HEVC [15].

In this paper, we propose a fast mode decision algorithm based on edge classification in the Hadamard transform domain. This paper focuses on the mode decision process of depth intra coding. We first investigate the relationship between the Hadamard coefficients of adjacent depth levels. Using this relationship, we present a simple method for obtaining the edge classification results of all possible PUs without calculating their transformed blocks. Finally, the proposed algorithm selectively omits unnecessary DMMs in the mode

Chun-Su Park is with Department of Digital Contents, Sejong University, Seoul, Korea. E-mail: cspark@sejong.ac.kr
decision based on the edge classification results.

The rest of this paper is organized as follows. In Section II, preliminaries and the basic idea of the proposed algorithm are introduced. Section III presents the fast edge classification algorithm in the Hadamard transform domain. The details of the proposed mode decision algorithm are also illustrated in Section III. Comparative experimental results of the proposed and conventional algorithms are presented in Section IV. Finally, our conclusions are drawn in Section V.

II. PRELIMINARIES AND BASIC IDEA

As did the HEVC standard, the 3D-HEVC standard adopted a highly flexible block partitioning structure by introducing several block concepts with separated roles [16]-[18]. The coding tree unit (CTU) is the basic processing unit and conceptually corresponds to the macroblock units that were used in several previous video standards [19], [20]. Inside the CTU, a quadtree structure is built to allow more flexibility in the partitioning of the CTU; each leaf node of the coding tree is called a coding unit (CU). The CU can be split into multiple prediction units (PUs) sharing the same prediction-related information. For intra prediction, the CU size is the same as the PU size, except for the smallest CU, which can be split into four equally-sized PUs [21]. For transform and quantization, the CU can be recursively partitioned into transform units (TUs) and the partitioning is signaled by a residual quadtree.

Let L be the number of possible depth levels of PUs. When the size of the CTU is $64 \times 64$, L is equal to 5 and the size of the PU can be $64 \times 64$, 32×32, 16×16, 8×8, and 4×4. Since a PU is partitioned into four sub-PUs at each level, the number of subblocks at depth level $l$, $0 \leq l < L$, is equal to $4^l$. Let $x_l$ be a $2^N\times2^N$ PU at depth $l$ and $X_l$ be the Hadamard-transformed block of $x_l$. Then, the Hadamard transform of $x_l$ is expressed as

$$X_l = H_{2^N}x_lH_{2^N}$$

(1)

where $H_{2^N}$ is the symmetric Hadamard matrix of order $2^N$ [22]-[25]. Fig. 1 shows the basis vectors of the $8\times8$ Hadamard transform as an example. As shown in Fig. 1, the basis vectors in the first column represent the straight horizontal edges and those in the first row represent the straight vertical edges. When $x_l$ contains only the straight horizontal (vertical) edges, all coefficients have zero values except those in the first column (row) [26].

These observations suggest that edge classification can be simply performed in the Hadamard transform domain. Let $X_l(v), v = 0, 1, \ldots, 2^N - 1$, be the $v$th column of $X_l$ and

$$X_l = [X_l(0) \ X_l(1) \ \cdots \ X_l(2^N - 1)]$$

(2)

Then, the classifier $F^H$, which determines whether the input contains only straight horizontal edges, can be defined as

$$F^H(X_l) = \begin{cases} 1, & \text{if } X_l(v) = \theta_{2^N\times1}, \forall v = 1, 2, \ldots, 2^N - 1, \\ 0, & \text{otherwise}, \end{cases}$$

(3)

where $\theta_{2^N\times1}$ is the $2^N\times1$ column vector with all elements equal to 0. Similarly, the vertical classifier $F^V$ can be defined. Using the above notations, we construct a set $\Psi$ of the PUs containing only the straight horizontal or vertical edges as follows

$$\Psi = \{x_l|F^H(X_l) = 1 \text{ or } F^V(X_l) = 1\}.$$  

(4)

In order to investigate the efficiency of the DMMs for the PUs in $\Psi$, we measured several probabilities using the intra-only structure. The results are summarized in Table II where $m$ is the best mode number with the minimum RD cost. In our simulations, the quantization parameters (QPs) of the natural video and depth map were set to 25 and 34, respectively [27]. Table II shows that the probability that a PU belongs to $\Psi$ is quite high. For example, 84.02% of the PUs belong to $\Psi$ on average. Therefore, we can expect that a fast algorithm designed for the PUs in $\Psi$ could achieve a considerable performance improvement.

Table II also presents the probability that one of the DMMs will be selected as the best prediction under several conditions. In general, the probability that one of the DMMs will be the best prediction is 2.35% on average. If $x_l \in \Psi$, the probability is extremely low and only 0.43% of the PUs in $\Psi$ are coded in the DMMs. Therefore, if $x_l \in \Psi$, it is likely that the DMMs can be safely omitted with negligible loss of coding efficiency. On the contrary, if $x_l \notin \Psi$, the probability that one of the DMMs will be the best prediction sharply increases to 11.86%. For the Shark sequence, the probability is 13.93%. This results in that, if $x_l \notin \Psi$, the encoder should examine the DMMs in the mode decision in order not to sacrifice coding efficiency significantly. Now, let us explain the proposed fast mode decision algorithm based on the edge classification.
III. PROPOSED FAST MODE DECISION ALGORITHM

A. Fast Edge Classification in the Transform Domain

As shown in (3), the classification result of the PU is obtained by exploiting its Hadamard-transformed block. This results in the Hadamard-transformed blocks of the PUs at all possible depth levels needing to be calculated, thereby placing a heavy computational load on the encoder. In this subsection, we first analyze the relationship between the Hadamard coefficients of adjacent depth levels. Then, we present a fast method for obtaining the classification results of the PUs without calculating their transformed blocks.

Let $X_l(u, v)$, $u, v = 0, 1, \ldots, 2N - 1$, be the $(u, v)$th transform coefficient of $X_l$. Further, let $S_{u,v}$, $u, v = 0, 1, \ldots, N - 1$, be a $2 \times 2$ matrix including four coefficients of $X_l$:

$$S_{u,v} = \begin{bmatrix} X_l(2u, 2v) & X_l(2u, 2v + 1) \\ X_l(2u + 1, 2v) & X_l(2u + 1, 2v + 1) \end{bmatrix}.$$  \hspace{1cm} (5)

Using the notation, $X_l$ can be represented by

$$X_l = \begin{bmatrix} S_{0,0} & S_{0,1} & \cdots & S_{0,N-2} & S_{0,N-1} \\ S_{1,0} & S_{1,1} & \cdots & S_{1,N-2} & S_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ S_{N-2,0} & S_{N-2,1} & \cdots & S_{N-2,N-2} & S_{N-2,N-1} \\ S_{N-1,0} & S_{N-1,1} & \cdots & S_{N-1,N-2} & S_{N-1,N-1} \end{bmatrix}.$$  \hspace{1cm} (6)

As mentioned, the $2N \times 2N$ PU at depth $l$ can be split into four $N \times N$ sub-PUs at depth $l + 1$. Let $x_{l+1}^0$, $x_{l+1}^1$, $x_{l+1}^2$, and $x_{l+1}^3$ be the four sub-PUs split from the parent PU $x_l$ in raster order (see Fig. 2). Then, analogously to (1), the Hadamard transform of the sub-PUs can be expressed as

$$X_{l+1}^k = H_{NX}x_{l+1}^k H_N$$  \hspace{1cm} (7)

where $k = 0, 1, 2, 3$, $X_{l+1}^k$ is the transformed block of $x_{l+1}^k$, and $H_N$ is the Hadamard matrix of order $N$.

![Fig. 2. Graphical explanation of PU splitting.](image)

The main idea of the proposed fast classification relies on the fact that the parent PU and its sub-PUs share a special relationship in the Hadamard transform domain. We can derive the following relationship between the coefficients of $X_l$ and $X_{l+1}^k$:

$$S_{u,v} = H_2 \begin{bmatrix} X_{l+1}^0(\tilde{u}, \tilde{v}) \\ X_{l+1}^1(\tilde{u}, \tilde{v}) \\ X_{l+1}^2(\tilde{u}, \tilde{v}) \\ X_{l+1}^3(\tilde{u}, \tilde{v}) \end{bmatrix} H_2$$  \hspace{1cm} (8)

where $X_{l+1}^k(\tilde{u}, \tilde{v})$, $k = 0, 1, 2, 3$, is the $(\tilde{u}, \tilde{v})$th coefficient of $X_{l+1}^k$ and $H_2$ is the order-2 Hadamard matrix defined as

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$  \hspace{1cm} (9)

The proof of (8) is given in the Appendix. The above result indicates that the Hadamard coefficients of the parent PU can be directly computed by using the coefficients of its sub-PUs. Based on these findings, we propose a fast algorithm that obtains the classification result of the parent PU without computing its transformed block.

Let us consider the case where four $N \times N$ sub-PUs contain only straight horizontal edges. Then, from the definition, the Hadamard coefficients of the sub-PUs satisfy the following constraint:

$$X_{l+1}^k = \begin{bmatrix} X_{l+1}^k(0, 0) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X_{l+1}^k(N-1, 0) \end{bmatrix}$$  \hspace{1cm} (10)

As all elements of $S_{u,v}$ should have zero values. Therefore, if (10) is satisfied, then we have

$$x_{l+1}^k = \begin{bmatrix} S_{0,0} & 0_{2 \times 2} & \cdots & 0_{2 \times 2} & 0_{2 \times 2} \\ S_{1,0} & 0_{2 \times 2} & \cdots & 0_{2 \times 2} & 0_{2 \times 2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ S_{N-2,0} & 0_{2 \times 2} & \cdots & 0_{2 \times 2} & 0_{2 \times 2} \\ S_{N-1,0} & 0_{2 \times 2} & \cdots & 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix}$$  \hspace{1cm} (11)

where $0_{2 \times 2}$ is a $2 \times 2$ matrix with all elements equal to 0. The above results indicate that only the coefficients belonging to $S_{0,0}$, $\tilde{u} = 0, 1, \ldots, N - 1$, need to be considered in order to obtain the horizontal classification result of $X_l$. Then, from (11), (3) can be simplified as

$$F^H(X_l) = \begin{cases} 1, & \text{if } X_l(1) = 0_{2N \times 1}, \\ 0, & \text{otherwise}. \end{cases}$$  \hspace{1cm} (12)

In a similar manner, the vertical classification process can be simplified.
For each CTU, the proposed algorithm constructs two pyramids of classification results, one for the horizontal classifier and one for the vertical classifier, as follows.

(a) For each PU at the highest depth level \( L - 1 \), the Hadamard-transformed block is computed. Then, the results of the horizontal and vertical classifiers, \( F^H \) and \( F^V \), are obtained using (3). The classification results of the PUs at the highest depth level are stored in memory. The following step is performed for each classifier separately.

(b) The classification results at depth \( l \) are computed using those at depth \( l + 1 \). If all the classification results of four sub-PU's at depth \( l + 1 \) are equal to 1, the classification of the parent PU at depth \( l \) is performed using (12) instead of (3). Otherwise, if any of the sub-PU's classification results are equal to 0, the parent PU's result is instantly set to 0 without any further computation. This procedure is repeated by decreasing \( l \) from \( L - 2 \) to 0.

Through the above procedure, the proposed algorithm minimizes the computational overhead required for obtaining the classification results at all possible depth levels. The resultant pyramids with the classification results are used in the fast mode decision process.

**B. Intra Mode Decision for Depth Maps**

In the current implementation, the intra mode decision of the depth PU follows two steps, the rough mode decision and the full RD cost calculation [28]. In the rough mode decision, the encoder first calculates the sum of the absolute Hadamard-transformed differences (SATD) of the 35 intra prediction modes of HEVC. From among these, a few modes are selected and inserted into the full-RD search list according to the SATD results and the mode information of the neighboring PUs. In addition, the two DMMs are also added to the full-RD search list. Finally, the true RD costs of all modes included in the list are calculated by performing the actual encoding and decoding tasks [29].

If the DMMs can be skipped at an early stage in the mode decision, the encoding complexity can be reduced significantly. It was observed in [14] that the pixel variances of the PUs selecting DMMs as the best mode are generally higher than those of the PUs selecting other intra modes as the best mode. Therefore, if the variance of the current PU is below a certain threshold \( T_h \), the proposed algorithm omits the DMMs in the mode decision process. Similar to the experiments presented in [14], we determine the threshold \( T_h \) as

\[
T_h = \{(\max(Q \gg 3 - 1, 3))^2 - 8\} \ll 2
\]

where \( Q \) is the QP of the current PU; the operators "\( \gg \)" and "\( \ll \)" indicate the bitwise left-shift and right-shift operations, respectively.

At the beginning of the mode decision process for the current CTU, the proposed algorithm obtains the classification results of all possible PUs as illustrated in Subsection III-A. Then, for each PU, the proposed mode decision algorithm proceeds as follows:

(a) In the same way as the conventional algorithm [15], a few of the 35 intra prediction modes are inserted into the full-RD search list.

(b) The proposed algorithm calculates the pixel variance of the current PU \( x_l \). If the variance is larger than the threshold \( T_h \), the DMMs are added to the full-RD search list and the algorithm goes to step (d). Otherwise, go to step (c).

(c) The proposed algorithm determines whether the current PU \( x_l \) belongs to \( \Psi \) using the precalculated classification results, \( F^H(x_l) \) and \( F^V(x_l) \). If \( x_l \notin \Psi \), the algorithm adds the DMMs to the full-RD search list. Otherwise, if \( x_l \in \Psi \), the DMMs are not used in the mode decision process.

(d) Calculate the RD costs of the modes included in the full-RD search list. The final RD cost of the current PU \( x_l \) is set to the minimum value among all examined ones.

Fig. 3 presents the flowchart of the proposed mode decision algorithm for encoding the current PU \( x_l \).

It is obvious that the proposed algorithm can be easily integrated into conventional algorithms designed for the fast implementation of the DMMs [8], [12]. Basically, these algorithms accelerate the search process by simplifying the partitioning patterns of the DMMs. Therefore, the performance of the existing algorithms can be improved significantly by adopting the proposed algorithm which adaptively omits the DMMs in the mode decision.

The novelty of the proposed algorithm is as follows:

- The proposed algorithm achieves better time saving performance than the state-of-the-art algorithm of the reference software. Comparative experimental results are presented in the next section.
### IV. EXPERIMENTAL RESULTS

In the experiments, we used the reference software HTM 9.1 of the 3D-HEVC standard [15]. We mainly followed the common test condition (CTC) [27]. Most of the encoder configurations including QP settings were inherited from the CTC. All test sequences were coded using the intra-only configurations including QP settings were inherited from the common test condition (CTC) [27]. Most of the encoder algorithms. For example, the recursive calculation can be implemented using a single function. This is a desirable property for practical hardware and software applications.

#### TABLE III
SIMULATION RESULTS OF GU’S AND THE PROPOSED ALGORITHMS IN TERMS OF BD-RATE AND MODE DECISION TIME

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Gu’s [14] BD-rate</th>
<th>Proposed BD-rate</th>
<th>( \Delta T )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BD(video)</td>
<td>BD(synth)</td>
<td>BD(video)</td>
</tr>
<tr>
<td>Balloons</td>
<td>-0.01%</td>
<td>0.04%</td>
<td>16.18%</td>
</tr>
<tr>
<td>Kendo</td>
<td>-0.01%</td>
<td>0.01%</td>
<td>17.68%</td>
</tr>
<tr>
<td>Newspaper CC</td>
<td>-0.02%</td>
<td>-0.01%</td>
<td>11.26%</td>
</tr>
<tr>
<td>GT Fly</td>
<td>-0.01%</td>
<td>0.00%</td>
<td>16.65%</td>
</tr>
<tr>
<td>Poznan Hall2</td>
<td>-0.02%</td>
<td>0.09%</td>
<td>20.06%</td>
</tr>
<tr>
<td>Poznan Street</td>
<td>-0.02%</td>
<td>0.04%</td>
<td>17.63%</td>
</tr>
<tr>
<td>Undo Dancer</td>
<td>-0.02%</td>
<td>0.05%</td>
<td>19.03%</td>
</tr>
<tr>
<td>Shark</td>
<td>-0.01%</td>
<td>0.02%</td>
<td>18.45%</td>
</tr>
<tr>
<td>Average (1024×768)</td>
<td>-0.01%</td>
<td>0.01%</td>
<td>15.04%</td>
</tr>
<tr>
<td>Average (1920×1088)</td>
<td>-0.01%</td>
<td>0.04%</td>
<td>18.36%</td>
</tr>
<tr>
<td>Average (all)</td>
<td>-0.01%</td>
<td>0.03%</td>
<td>17.12%</td>
</tr>
</tbody>
</table>

- The proposed algorithm accelerates the depth mode decision process using the fast straight edge classification technique. To the best of our knowledge, there does not exist any work that applies straight edge classification to depth map coding.
- The hierarchical relationship in (8) makes it possible to calculate the Hadamard coefficients of the parent PU using those of its sub-PUs. This fast approach can be applied to various video applications which exploit a flexible block partitioning structure.
- Due to the recursive nature of the relationship in (8), the implementation of the proposed algorithm could be much simpler than other edge classification algorithms. For example, the recursive calculation can be implemented using a single function. This is a desirable property for practical hardware and software applications.

#### A. Processing Time Comparison

Table III shows the BD-rate performance of Gu’s and the proposed algorithms for the luminance component [30]. In Table III, BD(video) and BD(synth) denote, respectively, the BD-rates calculated using the video PSNR and the synthesized view PSNR versus the total bitrate. Since the algorithms are applied only to the coding of depth maps, their video PSNRs and bitrates are exactly the same as those of the reference software. In other words, there is no change in the coding of the natural videos. In addition, the results shown in Table III indicate that both the algorithms have a minor effect on the BD-rate performance of the synthesized views. For example, BD(synth)’s of Gu’s and the proposed algorithms are 0.03% and 0.13% on average, respectively.

Table III also presents the reduction in the total encoding time of depth maps achieved by using Gu’s and the proposed algorithms. The time savings are calculated as

\[
\Delta T(\%) = \frac{T_{reference} - T_{algorithm}}{T_{reference}} \times 100
\]

where \(T_{reference}\) denotes the measured encoding time of the reference software and \(T_{algorithm}\) denotes the measured time of each algorithm. We can see that both the algorithms tend to accelerate the encoding process without sacrificing the coding efficiency. For all sequences, Gu’s algorithm achieves a 17.12% reduction in encoding time without increasing the BD-rate. For the XGA (1024×768) sequences, the average processing time of the proposed method is 22.65% less than that of the reference software. Further, the amount of complexity reduction is 21.91% for the HD (1920×1088) sequences. For all sequences, the proposed algorithm reduces the processing time by 22.19% with a 0.13% BD-rate increase in the synthesized views. Both the algorithms achieve the best performance for the Poznan Hall2 sequence: Gu’s and the proposed algorithms reduce the complexity by up to 20.06% and 37.65%, respectively.

We can observe that, as compared to Gu’s algorithm, the proposed algorithm achieves a better performance for most of the sequences except the Poznan Street and Shark sequences. This is because the number of PUs containing only straight horizontal or vertical edges is relatively small for the the sequences. However, as demonstrated by the results in Table III, the proposed algorithm tends to achieve a better performance than Gu’s algorithm.

#### B. Statistical Analysis

In this subsection, we present a statistical analysis of Gu’s and the proposed algorithms to provide more insight into the results. The statistical analysis is listed in Table IV. In Table IV, \(R_A\) denotes the ratio of PUs of which the DMMs are examined in the mode decision process; \(R_C\) denotes the ratio of the DMM PUs which are correctly classified with...
reference to the exhaustive algorithm; \( R_{H} \) is the ratio of the DMM PUs among the examined PUs in the mode decision.

In Table IV, \( R_{A} \)’s of Gu’s and the proposed algorithms are 33.7% and 20.7%, respectively. This means that \( R_{A} \) of the proposed algorithm is 38.6% less than that of Gu’s algorithm. Although the proposed algorithm omits the DMMs of much more PUs, \( R_{C} \) of the proposed algorithm is similar to that of Gu’s algorithm. In our simulations, \( R_{C} \’s \) of Gu’s and the proposed algorithms are 90.2% and 86.0%, respectively.

However, for some sequences, \( R_{C} \) of the proposed algorithm is relatively low as compared to that of Gu’s algorithm. For the Poznan Hall2 sequence, \( R_{C} \) of the proposed algorithm is 76.8% and that of Gu’s algorithm is 89.4%. Basically, this is because the proposed algorithm examines much less PUs than Gu’s algorithm. For the Poznan Hall2 sequence, the proposed algorithm examines the DMMs of only 4.0% PUs. On the contrary, Gu’s algorithm examines 34.8% PUs. The accuracy of both the algorithms can be objectively evaluated using \( R_{H} \). For all sequences, \( R_{H} \’s \) of Gu’s and the proposed algorithms are 2.9% and 4.5%, respectively.

### TABLE IV

<table>
<thead>
<tr>
<th>Sequences</th>
<th>( R_{A} )</th>
<th>( R_{C} )</th>
<th>( R_{H} )</th>
<th>( R_{A} )</th>
<th>( R_{C} )</th>
<th>( R_{H} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balloons</td>
<td>33.9%</td>
<td>96.3%</td>
<td>2.9%</td>
<td>17.9%</td>
<td>93.6%</td>
<td>5.3%</td>
</tr>
<tr>
<td>Kendo</td>
<td>28.0%</td>
<td>95.9%</td>
<td>1.9%</td>
<td>13.7%</td>
<td>94.0%</td>
<td>3.8%</td>
</tr>
<tr>
<td>Newspaper CC</td>
<td>44.4%</td>
<td>95.4%</td>
<td>3.9%</td>
<td>28.0%</td>
<td>92.6%</td>
<td>6.0%</td>
</tr>
<tr>
<td>GT Fly</td>
<td>32.7%</td>
<td>63.4%</td>
<td>2.5%</td>
<td>17.6%</td>
<td>54.3%</td>
<td>4.1%</td>
</tr>
<tr>
<td>Poznan Hall2</td>
<td>34.8%</td>
<td>89.4%</td>
<td>0.4%</td>
<td>4.0%</td>
<td>76.8%</td>
<td>2.9%</td>
</tr>
<tr>
<td>Poznan Street</td>
<td>34.1%</td>
<td>90.8%</td>
<td>2.3%</td>
<td>44.8%</td>
<td>96.4%</td>
<td>1.8%</td>
</tr>
<tr>
<td>Undo Dancer</td>
<td>35.5%</td>
<td>92.1%</td>
<td>2.1%</td>
<td>12.0%</td>
<td>84.3%</td>
<td>5.7%</td>
</tr>
<tr>
<td>Shark</td>
<td>35.9%</td>
<td>98.5%</td>
<td>7.2%</td>
<td>28.0%</td>
<td>96.2%</td>
<td>6.6%</td>
</tr>
<tr>
<td>Average (1024×768)</td>
<td>35.3%</td>
<td>95.9%</td>
<td>2.9%</td>
<td>19.9%</td>
<td>93.4%</td>
<td>5.1%</td>
</tr>
<tr>
<td>Average (1920×1088)</td>
<td>32.6%</td>
<td>86.8%</td>
<td>2.9%</td>
<td>21.3%</td>
<td>81.6%</td>
<td>4.2%</td>
</tr>
<tr>
<td>Average (all)</td>
<td>33.7%</td>
<td>90.2%</td>
<td>2.9%</td>
<td>20.7%</td>
<td>86.0%</td>
<td>4.5%</td>
</tr>
</tbody>
</table>

### C. Detailed Performance Analysis

The proposed algorithm needs to calculate the Hadamard-transformed blocks of all possible PUs. As mentioned, the proposed algorithm recursively computes the transformed blocks using the fast algorithm in (8). To show the performance improvement of the fast algorithm, we measured the time saving of the proposed algorithm when the transformed blocks are computed with and without using the algorithm in (8). The measured results are briefly summarized in Table V. In our simulations, the time savings with and without using of the fast algorithm are 20.63% and 22.19%, respectively. This means that, the fast algorithm can reduce the mode decision time by 1.56% on average as compared to the direct implementation [31]. It is worthwhile to note that the fast approach accelerates the mode decision process without affecting the coding results at all.

### V. CONCLUSION

A new intra mode selection algorithm based on the edge classification was presented in this paper. We first analyzed the recursive relationship between the Hadamard coefficients of adjacent depth levels. Then, based on the analysis, we present a simple method for obtaining the edge classification results of all possible PUs without calculating their transformed blocks. Finally, we proposed a fast algorithm that selectively omits unnecessary DMMs in the mode decision using the edge classification results. The experimental results clearly demonstrated that the proposed algorithm achieves a substantially higher acceleration than the fastest conventional algorithm.

### APPENDIX

**DERIVATION OF (9)**

The relationship between the Hadamard coefficients of \( x_{l}^{k} \) and \( x_{l+3}^{k} \), for \( k = 0, 1, 2, 3 \), can be derived based on the linearity of the Hadamard transform. From the definition, the 2N×2N PU at depth \( l + 1 \) can be expressed using four N×N sub-PUs at depth \( l + 1 \) as follows:

\[
\begin{align*}
X_{l} &= \begin{bmatrix}
x_{l+1}^{0} \\
x_{l+1}^{1} \\
x_{l+1}^{2} \\
x_{l+1}^{3}
\end{bmatrix}.
\end{align*}
\]  

Let \( \mathbf{H}_{2N}(u) \) and \( \mathbf{H}_{2N}(v) \) be the \( u \)th row and \( v \)th column vectors of \( \mathbf{H}_{2N} \), respectively. Then, for the Hadamard transform in dyadic order, the following relations hold [24]:

\[
\begin{align*}
\mathbf{H}_{2N}(2\hat{u} + \alpha) &= \begin{bmatrix}
\mathbf{H}_{N}(\hat{u}) \\
(−1)^{\alpha}\mathbf{H}_{N}(\hat{u})
\end{bmatrix} \quad \text{(A.2)}
\end{align*}
\]

\[
\begin{align*}
\mathbf{H}_{2N}(2\hat{v} + \beta) &= \begin{bmatrix}
\mathbf{H}_{N}(\hat{v}) \\
(−1)^{\beta}\mathbf{H}_{N}(\hat{v})
\end{bmatrix} \quad \text{(A.3)}
\end{align*}
\]

where \( \hat{u}, \hat{v} = 0, 1, \ldots, N - 1 \) and \( \alpha, \beta = 0, 1 \). For simplicity of explanation, let us consider the case where both \( \alpha \) and \( \beta \) are equal to 0. The other cases can be easily treated in a similar way.

Using the above notations, the \((2\hat{u}, 2\hat{v})\)th coefficient of \( X_{l} \) can be represented by its sub-PUs as follows:

\[
X_{l}(2\hat{u}, 2\hat{v}) = \mathbf{H}_{2N}(2\hat{u})X_{l}\mathbf{H}_{2N}(2\hat{v})
\]

\[
= \begin{bmatrix}
\mathbf{H}_{N}(\hat{u}) \\
\mathbf{H}_{N}(\hat{u})
\end{bmatrix} \begin{bmatrix}
x_{l+1}^{0} \\
x_{l+1}^{1} \\
x_{l+1}^{2} \\
x_{l+1}^{3}
\end{bmatrix} \begin{bmatrix}
\mathbf{H}_{N}(\hat{v}) \\
\mathbf{H}_{N}(\hat{v})
\end{bmatrix}.
\]  

\[\text{(A.4)}\]
Let \( \mathbf{x}^k_{l+1}(\hat{u}) \) be the \( \hat{u} \)th row of \( \mathbf{x}^k_{l+1} \). Then, (A.4) can be simplified as (A.5) at the top of the page. Similarly, we can derive (A.6), (A.7), and (A.8). Then, the relationship can be summarized as (A.9). From (A.9), we can know that the Hadamard coefficients of the parent PU at depth \( l \) can be directly computed by using the coefficients of its sub-PUs at depth \( l+1 \).

\[
X_l(2\hat{u}, 2\hat{v}) = \mathbf{H}_N(\hat{u}) \mathbf{H}_N(\hat{u})^T \mathbf{H}_N(\hat{v}) \mathbf{H}_N(\hat{v})^T.
\]

Let \( \mathbf{H}_N(\hat{u}) \) be the \( \hat{u} \)th row of \( \mathbf{H}_N \). Then, (A.4) can be simplified as (A.5) at the top of the page. Similarly, we can derive (A.6), (A.7), and (A.8). Then, the relationship can be summarized as (A.9). From (A.9), we can know that the Hadamard coefficients of the parent PU at depth \( l \) can be directly computed by using the coefficients of its sub-PUs at depth \( l+1 \).

\[
X_l(2\hat{u}, 2\hat{v}+1) = X^0_{l+1}(\hat{u}, \hat{v}) - X^1_{l+1}(\hat{u}, \hat{v}) - X^2_{l+1}(\hat{u}, \hat{v}) - X^3_{l+1}(\hat{u}, \hat{v}).
\]

\[
X_l(2\hat{u}+1, 2\hat{v}) = X^0_{l+1}(\hat{u}, \hat{v}) + X^1_{l+1}(\hat{u}, \hat{v}) - X^2_{l+1}(\hat{u}, \hat{v}) - X^3_{l+1}(\hat{u}, \hat{v}).
\]

\[
X_l(2\hat{u}+1, 2\hat{v}+1) = X^0_{l+1}(\hat{u}, \hat{v}) + X^1_{l+1}(\hat{u}, \hat{v}) + X^2_{l+1}(\hat{u}, \hat{v}) + X^3_{l+1}(\hat{u}, \hat{v}).
\]

\[
\mathbf{H}_2 = \begin{bmatrix}
X^0_{l+1}(\hat{u}, \hat{v}) & X^1_{l+1}(\hat{u}, \hat{v}) & X^2_{l+1}(\hat{u}, \hat{v}) & X^3_{l+1}(\hat{u}, \hat{v})
\end{bmatrix}
\]

\[
\mathbf{S}_{\hat{u}, \hat{v}} = \begin{bmatrix}
X^0_l(2\hat{u}, 2\hat{v}) & X^1_l(2\hat{u}, 2\hat{v}+1) & X^2_l(2\hat{u}+1, 2\hat{v}) & X^3_l(2\hat{u}+1, 2\hat{v}+1)
\end{bmatrix}
\]

\[
\text{REFERENCES}
\]


Chun-Su Park received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, in 2003 and 2009, respectively. From 2009 to 2010, He was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. And, he was a senior research engineer at Samsung Electronics from 2010 to 2012. From 2012 to 2014, he was an assistant professor with Dept. of Info. and Telecom. Eng. at Sangmyung University. In 2014, he joined Dept. of Digital Contents at Sejong University where he is currently an assistant professor. His research interests are in the areas of video signal processing, parallel computing, and multimedia communications.