Low Complexity Quadtree based All Zero Block Detection Algorithm for HEVC

Jia Su*, Yiqing Huang*, Lei Sun*, Shinichi Sakaida† and Takeshi Ikenaga*
* Waseda University, Fukuoka, Japan
E-mail: selene@suou.waseda.jp Tel/Fax: +81-93-6925319
† Science & Technology Research Laboratories, NHK, Tokyo

***Abstract*—High Efficiency Video Coding (HEVC) is a proposed video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding). It aims to substantially improve coding efficiency compared to H.264/AVC High Profile, in which transform unit (TU) has been added as one of the key advanced features, at the expense of traversal computation of transform and quantization quadtree. To release the computation burden brought by TU, all-zero block (AZB) detection algorithms can be introduced, which early terminate transform and quantization with almost no PSNR loss. However, the previous works can't support the tree structure based (depth-2 or more) new standard. And by setting the thresholds with sum of absolute difference (SAD) only, they can not work directly in case of intra prediction or other criterion based inter prediction. This paper proposes a quadtree based variable sized all-zero block detection scheme for high efficiency video coding which mainly consists of all-zero block detection for prediction unit (PU) and quadtree based up-bottom zero node detection. Firstly, 32×32 with depth-3 integer transform and quantization based all-zero block detection has been derived by gaussian distribution modeling, of which the approach can also be extended to sum of absolute transformed differences (SATD) and sum of squared differences (SSD). Further more, the chroma based condition has also been proposed by study of the color component correlation in this paper. Moreover, the up-bottom quadtree based adaptive to variable size and depth all-zero node detection has been designed. By implementing in the HEVC reference test model (HM), evaluation results illustrate the proposed algorithm can save total encoding time 4%-25% averagely with at most -0.24dB BDPSNR and +3.55% BDBR loss.**

I. Introduction

With an ever growing diversification of usage models and ever increasing demands for higher quality, ISO-IEC/MPEG and ITU-T/VCEG have recently formed the Joint Collaborative Team on Video Coding (JCT-VC). The JCT-VC group is aiming to develop the next generation video coding standard, called High Efficiency Video Coding (HEVC). There are two distinct goals for the HEVC standard, a low complexity operation point with significant coding efficiency improvement suitable for mobile and real-time communication to enable emerging high resolution video services, and a high performance operating point with more substantial improvement in coding efficiency with additional complexity. Comparing with H.264/AVC, HEVC is demonstrated to achieve a bit-rate reductions of around 30% based on objective measures and around 60% based on subjective testing with 1080p sequences. HEVC is still based on hybrid coding structure. The main change is about the "coder control" module which is utilized for directing the work of Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU). These three types of units compose a powerful descriptor to strengthen the flexibility of one macroblock (MB) in previous standard with an expense regressive complexity. Fig. 1 displays the coding structure of HEVC. It is still based on hybrid coding structure. And the boarder line locates this paper's main target module. Early video coding standards such as MPEG-1 [1] and MPEG-2 [2] employ fixed-size block motion compensation method with a MB size of 16×16. In the state-of-the-art standard H.264/AVC [3], this MB size is retained but with the addition of a depth-2 quadtree, which significantly improves coding efficiency. However, it has been observed that the maximum 16×16 block size is too small when applied to high resolution video and causes inefficiency [4] [5]. And depth-2 quadtree still can not meet with the requirement of the flexibility requirement. And quadtree based transform, adapting to the quadtree based coding structure, has also been added in to HEVC with multiple times of computation burden. In order to solve this problem, this paper introduces the widely used All-Zero Block (AZB) detection algorithm. Many works have been done in H.263 8×8 transformation based AZB detection and H.264 4×4 integer transformation based AZB detection in this field. Alice Yu [6] first proposed a method of early predicting those blocks, however, the algorithm incurs image degradation due to the improper threshold, which will give

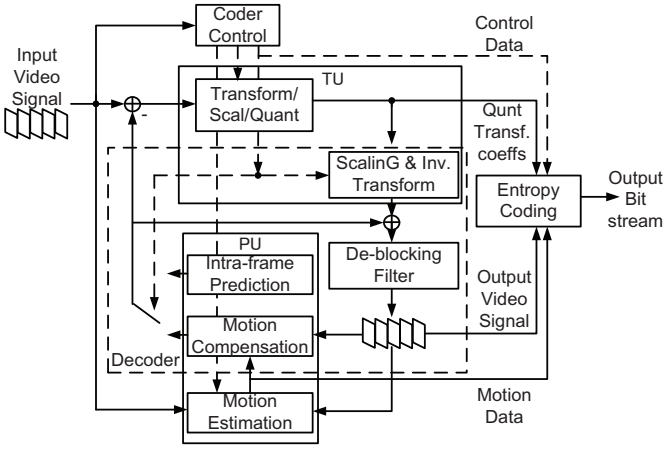


Fig. 1: Coding structure of HEVC

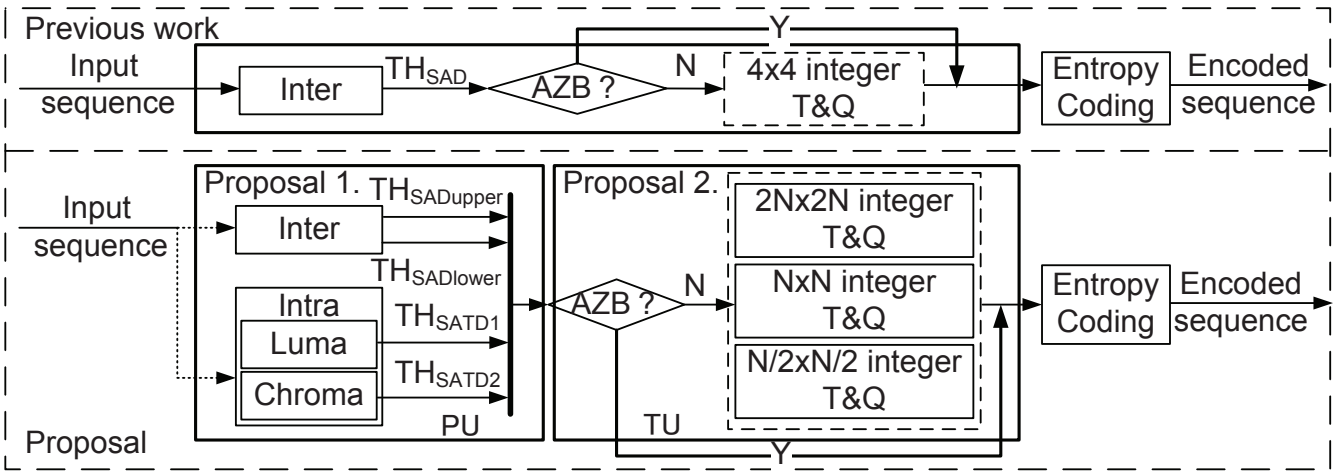


Fig. 2: Proposed algorithm compare with previous work

false detection (means the block detected to be a zero block is not a zero block). Zhou Xuan [7] proposed a new method for early detection of all-zero discrete cosine transform (DCT) coefficients, which does not have false detection and can still uphold image fidelity. However, the method of Zhou does not analyze the judgment for all-zero DCT coefficient blocks in detail, and only give the sufficient condition which will lead to miss detection (means the block detected not to be a zero block is a zero block). Sousa [8] theoretically derived a precise condition and improved Xuan's algorithm. Shi Jun [9] proposed a judgment with low miss detection based on the analysis of the sufficient and necessary condition for the early detection of all-zero coefficient block. Wang et al. [10] applied Sousas model to the DCT-like 4×4 integer transform in H.264/AVC. Kim et al. [11] proposed a novel all-zero blocks detecting algorithm for the DCT-like 4×4 integer transform in H.264/AVC. Recently, Wang et al. [12] derived more effective sufficient conditions to early determination of all-zero 4×4 blocks in H.264/AVC. All the work above based on previous video coding standard is inefficiency to directly extend to HEVC mainly for two reasons. First, for the quadtree based transform and quantization in HEVC, the AZB detection algorithm should adapt to both the size and depth. Note that in HEVC while giving a CU the transform unit will be recursively partitioning into quadtrees with a user defined depth, so the previous research on AZB can be recognized as one dimensional methods without transform size or depth changed. Second, there is no intra prediction faced algorithm and the chroma hasn't been taken in to consideration specifically.

Therefore, this paper proposes 2-D quadtree based AZB detection algorithm based on new CU, PU and TU in coming standard HEVC. It mainly consists of two proposals as Fig. 2 presents, quadtree analysis based 2-D zero node detection and all-zero block detection for prediction unit (inter and inter prediction). Foremost, by gaussian distribution modeling, 32×32 with depth-3 integer transform and quantization based all-zero

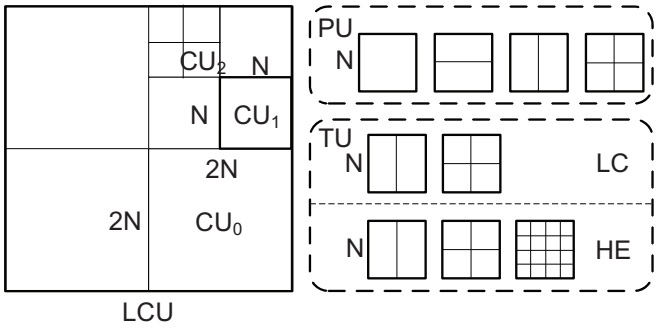


Fig. 3: Relation between TU PU CU

block detection has been derived, of which the approach can also be extended to sum of absolute transformed differences (SATD) and sum of squared differences (SSD). And by study of the color component correlation the chroma based condition has also been proposed. Subsequently, quadtree descriptors have been designed for 2-D method. And by comparing and setting the predetermine conditions between the neighbor nodes, the quadtree based 2-D to variable size and depth all-zero node detection has been designed.

The rest of paper is organized as follows. The next section deals with PU related large transform and quantization based all-zero block detection. And Section 3 highlights the proposed quadtree analysis based 2-D zero node detection. Section 4 summarizes our proposed quadtree based algorithm by combining with all the proposals in this paper. Section 5 presents the experimental results and the last section concludes the paper.

II. Prediction Unit Related All-Zero Block Detection

A. Relationship between CU PU and TU

A highly flexible hierarchical structure based on the generic quadtree scheme is defined by three independent block concepts: coding unit (CU), prediction unit (PU), and transform

unit (TU) in Fig. 3 CU is the basic unit of region splitting. CU is analogous to the concept of macroblock, but it does not restrict the maximum size and it allows recursive splitting into four equal size CUs to improve the contents-adaptivity. PU is the basic unit of inter/intra prediction and it may contain multiple arbitrary shape partitions in a single PU to effectively code irregular image patterns. TU is the basic unit of transform. It can be defined independently from the PU, however, its size is limited to the CU which the TU belongs to. This separation of the block structure into three different concepts allows each to be optimized according to its role, which results in the improved coding efficiency. In TU structure representation, residual quadtree structure is adopted. The same maximum quadtree depth is applied for both luma and chroma components of each CU. TU have different splitting for low complexity (LC) and high efficiency (HE) configurations. For LC configuration, there are only two possible splitting from a CU to TUs depending on the value of syntax element $transform\_split\_flag$. That is to say, the residual quadtree structure is restricted to 2 levels in LC configuration. However, for HE configuration, the splitting from a CU to TUs is done recursively depending on the value of $transform\_split\_flag$, but restricted to maximum 3 levels of the residual quadtree and minimum supported transform size. The syntax element $transform\_split\_flag$ may not be present in the bitstream, which should be inferred according to the coding context, such as intra coding or inter coding modes, low complexity or high efficiency cases. And the test and analysis in this paper mainly focus on the low complexity configuration. Once the division of coding unit hierarchical tree is done, the leaf nodes CUs can be further split into prediction units. Different PU partitions correspond to different PU types, which consist of skip, intra and inter. Meanwhile, skip and inter can take the same all zero block detection condition, as in the mode decision process they chose same criterion for checking the RDO cost. In inter prediction users can calculate the rate distortion by SAD, SATD or sum of squared difference (SSD). We collectively called these as distortion values as previous section mentioned. Currently, as HEVC adopted 32×32 with depth-3 integer transform and quantization as the upper limit for configuration, although 64×64 transform can be indirectly calculated by utilizing 32×32. Therefore, this section derives SAD and SATD based new sufficient conditions for large (32×32 with depth-3) AZB early termination from inter to intra and obtains color component correlation based chrominance condition in the third part.

B. SAD based New Sufficient Condition for Large All-zero Block Determination

For smooth data, a large transform has several advantages such as better energy compaction and reduced quantization error. In HD sequences, most image patterns in a coding unit represent a small part of objects or backgrounds which can be described as homogeneous texture patterns with little variation. Therefore, the coding efficiency of high resolution

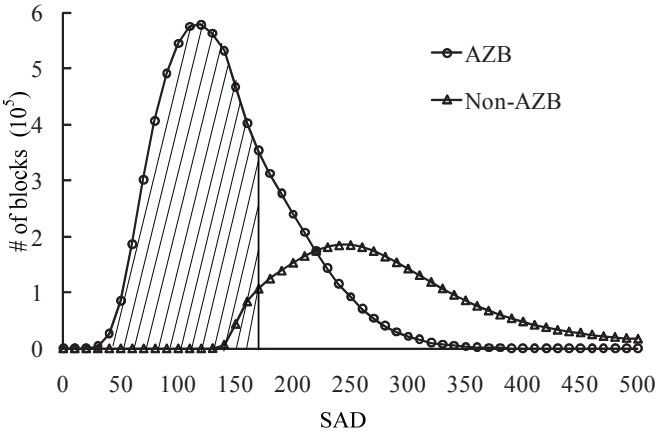


Fig. 4: AZB and non-AZB distribution

video can be improved by the use of large transforms as well as large block sizes. In Fig. 4, the X axis represents the sum of absolute difference (SAD) range (±5) of each 8×8 block, while, the Y axis is the number of blocks for 200 frames, furthermore, the vertical line is the threshold line. Form this figure, it obviously showed that if the SAD value is smaller than the threshold it is all-zero block; while, if the SAD value is bigger than the threshold there are false detection (the Non-AZB was detected as AZB in this case) occurs. The shadow area is the total number detected AZB. Through the AZB and non-AZB distribution test, while the quantization parameter (QP) value increased from 20 to 40, total number of detected AZB increased.

The prediction error signals of an N×N inter predicted block with motion-compensated prediction, can be computed by

$$f(x,y) = s(x,y) - p(x,y), \quad \text{for} \quad x,y = 0,1,...,N-1. \quad (1)$$

Here, s and p represent the original block and the motion-compensated prediction block, respectively. In encoding, DCT coefficients of each prediction error block are quantized to represent them in a reduced range of values. For a uniform scalar quantization in HEVC which is the same with H.263 [1] and MPEG-4 Part 2 [2], the quantized coefficient $F_Q(u,v)$ can be obtained by

$$|F_Q(u,v)| = \text{sign}(F(u,v)) \left\lfloor \frac{|F(u,v)| - (Qp/2)}{2Qp} \right\rfloor \quad (2)$$

where the symbol $\lfloor \rfloor$ denotes rounding to the nearest integer.

When applying 2-D DCT to a given block f of N×N samples, each frequency component F(u, v) can be calculated by

$$F(u,v) = C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \times \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (3)$$

with

$$C(n) = \begin{cases} \sqrt{\frac{1}{N}}, & \text{for} \quad n = 0 \\ \sqrt{\frac{2}{N}}, & \text{for} \quad n > 0. \end{cases} \tag{4}$$

Moreover, if

$$|F_Q(u,v)| < 1 \tag{5}$$

it is obvious that all transform coefficients are simultaneously quantized into zero. It can be observed for an N×N block that when

$$\begin{aligned} |F(u,v)| &\le C(u)C(v) \sum_{x=0}^{7} \sum_{x=0}^{7} |f(x,y)| \\ &\times \max |\cos(\tfrac{(2x+1)\mu\pi}{2N}) \cos(\tfrac{(2y+1)v\pi}{2N})| \\ &= C(u)C(v) \times SAD \\ &\times \max |\cos(\tfrac{(2x+1)\mu\pi}{2N}) \cos(\tfrac{(2y+1)v\pi}{2N})| \\ &< \tfrac{5}{2}Qp \end{aligned} \tag{6}$$

Therefore, all DCT coefficients of an N×N block will be quantized to zeros if

$$SAD < 10Qp/\cos^2(\frac{\pi}{2N}). \tag{7}$$

The inequality Eq. 6 is achieved generally by analyzing the dynamic range of each frequency component based on the SAD of an entire block. In fact, within a predicted error block, the energy of predicted error signals in different parts is usually inhomogeneous and thus, it is possible to derive a more precise sufficient condition to early determining all-zero block-based on the SADs of partial blocks. However, due to the variable size transform and quantization in the HEVC and also considering the increasing complexity, this paper focus on an entire transform unit for different sizes. The condition in Eq. 6 for DCT coefficient F(u, v) quantized to zero can be further rewritten as follows:

$$TH_{SAD_{32\times32}} < 40Qp/\cos^2(\frac{\pi}{64}). \tag{8}$$

$$TH_{SAD_{16\times16}} < 20Qp/\cos^2(\frac{\pi}{32}). \tag{9}$$

$$TH_{SAD_{8\times8}} < 10Qp/\cos^2(\frac{\pi}{16}). \tag{10}$$

$$TH_{SAD_{4\times4}} < 5Qp/\cos^2(\frac{\pi}{8}). \tag{11}$$

C. SATD based Sufficient Condition for Large All-zero Block Determination

In the sub-sampling motion estimation, SATD will be obtained for RDO process. In this section the sufficient SATD based all-zero block will be deviled based on the Gaussian distribution model. For the sake of simplicity, the previous work [13] [14] used Laplacian distribution to model them. However, the generalized Gaussian distribution is the most accurate representation of transformed coefficients. And here the 8×8 will be derived as an example. Other size of derivations can follow a same method. In the reference software, the SATD of an 8×8 block is defined as half of the sum of absolute values of a two-dimensional (2-D) 8×8 Hadamard-transformed residue f as shown in Eq. 12, the Hadamard matrices H are made up entirely of 1 and -1.

$$G = HfH^T \tag{12}$$

$$SATD = \frac{1}{2} \cdot \sum_{v=0}^{7} \sum_{u=0}^{7} |G(u,v)| \tag{13}$$

The correlation for the pixel values after linear prediction is separable in both horizontal and vertical directions. Therefore, we can approximate the input values at the input of Hadamard-transformed by a Gaussian distribution with zero mean and a separable covariance

$$r(m,n) = \sigma_f^2 \rho^{|m|} \rho^{|n|} \tag{14}$$

where m and n are the horizontal and vertical distances, respectively, between two pixels, and $|\rho| < 1$ is the correlation coefficient. Typically, ρ ranges from 0.4 to 0.75, and $\rho = 0.6$ is commonly set for simulations. we can assume that the residues after Hadamard transform is Gaussian distributed, the variance of the (u, v)th Hadamard coefficients can be written as

$$\sigma_H^2(u,v) = \sigma_f^2 (HRH^T)_{u,u} (HRH^T)_{v,v} \tag{15}$$

where

$$R = \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 & \rho^4 & \rho^5 & \rho^6 & \rho^7 \\ \rho & 1 & \rho & \rho^2 & \rho^3 & \rho^4 & \rho^5 & \rho^6 \\ \rho^2 & \rho & 1 & \rho & \rho^2 & \rho^3 & \rho^4 & \rho^5 \\ \rho^3 & \rho^2 & \rho & 1 & \rho & \rho^2 & \rho^3 & \rho^4 \\ \rho^4 & \rho^3 & \rho^2 & \rho & 1 & \rho & \rho^2 & \rho^3 \\ \rho^5 & \rho^4 & \rho^3 & \rho^2 & \rho & 1 & \rho & \rho^2 \\ \rho^6 & \rho^5 & \rho^4 & \rho^3 & \rho^2 & \rho & 1 & \rho \\ \rho^7 & \rho^6 & \rho^5 & \rho^4 & \rho^3 & \rho^2 & \rho & 1 \end{bmatrix} \tag{16}$$

And $[\cdot]_{u,u}$ is the (u, u)th component of the matrix. The ${\sigma_f}^2$ shown below is rounding the nearest integer for simplicity.

$$\left|\sigma_H^2(u,v)\right| \approx \sigma_f^2 \times \begin{bmatrix} 606 & 61 & 133 & 90 & 327 & 78 & 195 & 86 \\ 61 & 6 & 13 & 9 & 33 & 8 & 20 & 9 \\ 133 & 13 & 29 & 20 & 72 & 17 & 43 & 19 \\ 90 & 9 & 20 & 13 & 49 & 12 & 29 & 13 \\ 327 & 33 & 72 & 49 & 176 & 42 & 105 & 46 \\ 78 & 8 & 17 & 12 & 42 & 10 & 25 & 11 \\ 195 & 19 & 43 & 29 & 105 & 25 & 63 & 28 \\ 86 & 9 & 19 & 13 & 46 & 11 & 28 & 12 \end{bmatrix} \tag{17}$$

Moreover, the expected mean absolute value of a zero-mean Gaussian distributed random variable x is

$$\begin{aligned} E(|x|) &= \textstyle\int_{-\infty}^{\infty} |x| \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-|x|^2}{2\sigma^2}} dx \\ &= \textstyle\int_{0}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} dx^2 = \sqrt{\frac{2}{\pi}}\sigma \end{aligned} \tag{18}$$

We can approximate the expected mean absolute value by SAD related to current N×N TU, so we have

$$\sigma_f = \sqrt{\frac{\pi}{2}} \cdot \frac{SAD}{N \times N} \tag{19}$$

And the SATD of an 8×8 block can be approximated by Eq. 28

$$\begin{aligned} SATD &= \tfrac{1}{2} \cdot \sum_{v=0}^{7} \sum_{u=0}^{7} |G(u,v)| \cong \tfrac{1}{2} \cdot \sum_{v=0}^{7} \sum_{u=0}^{7} E(|G(u,v)|) \\ &= \tfrac{1}{2} \sum_{v=0}^{7} \sum_{u=0}^{7} \sqrt{\tfrac{2}{\pi}} \sigma_G(u,v) = \tfrac{\sigma_f}{\sqrt{2\pi}} \sum_{v=0}^{7} \sum_{u=0}^{7} g(u,v) \\ &\cong 435.83 \tfrac{\sigma_f}{\sqrt{2\pi}} \end{aligned} \tag{20}$$

The relation between SATD and SAD can be seen from Eq. 19 and Eq. 20. As for the relation between different sizes, it can be deduced by the same probability theory. Finally, the different sized threshold for SATD have been obtained based on the relation between SATD and SAD together with Eq. 21−24.

$$TH_{SATD_{32\times32}} < 373Qp/\cos^2(\frac{\pi}{64}). \tag{21}$$

$$TH_{SATD_{16\times16}} < 107Qp/\cos^2(\frac{\pi}{32}). \tag{22}$$

$$TH_{SATD_{8\times8}} < 30.405Qp/\cos^2(\frac{\pi}{16}). \tag{23}$$

$$TH_{SATD_{4\times4}} < 8.75Qp/\cos^2(\frac{\pi}{8}). \tag{24}$$

D. SSD based Sufficient Condition for Large All-zero Block Determination

SSD will be obtained for RDO process in mode decision. In this section the sufficient SSD based all-zero block will be deviled based on the proposed model.

$$SSD(S,C) = \sum\sum (s_{ij} - c_{ij})^2 = ||S - C||_F^2 \tag{25}$$

$$\begin{aligned} S &= D + P \\ C &= \hat{D} + P \end{aligned} \tag{26}$$

The correlation for the pixel values after linear prediction is separable in both horizontal and vertical directions. Therefore, we can approximate the input values at the input of Hadamard-transformed by a Gaussian distribution with zero mean and a separable covariance

$$\begin{aligned} &SSD(S,C) = ||S - C||_F^2 = ||D + P - \hat{D} - P||_F^2 \\ &= ||D - \hat{D}||_F^2 = SSD(D,\hat{D}) \end{aligned} \tag{27}$$

where s_{ij} and c_{ij} represent the original and reconstructed blocks, respectively, between two pixels, and $|\rho| < 1$ is the correlation coefficient. Typically, ρ ranges from 0.4 to 0.75, and $\rho = 0.6$ is commonly set for simulations. we can assume that the residues after Hadamard transform is Gaussian distributed, the variance of the (u,v)th Hadamard coefficients can be written as

$$\sigma_H^2(u,v) = \sigma_f^2 (HRH^T)_{u,u} (HRH^T)_{v,v} \tag{28}$$

where And $[\cdot]_{u,u}$ is the (u,u)th component of the matrix. The ${\sigma_f}^2$ shown below is rounding the nearest integer for simplicity. Moreover, the expected mean absolute value of a zero-mean Gaussian distributed random variable x is And the SSD of an 8×8 block can be approximated by Eq. 28

$$\begin{aligned} SSD &= \tfrac{1}{2} \cdot \sum_{v=0}^{7} \sum_{u=0}^{7} |G(u,v)| \cong \tfrac{1}{2} \cdot \sum_{v=0}^{7} \sum_{u=0}^{7} E(|G(u,v)|) \\ &= \tfrac{1}{2} \sum_{v=0}^{7} \sum_{u=0}^{7} \sqrt{\tfrac{2}{\pi}} \sigma_G(u,v) = \tfrac{\sigma_f}{\sqrt{2\pi}} \sum_{v=0}^{7} \sum_{u=0}^{7} g(u,v) \\ &\cong 435.83 \tfrac{\sigma_f}{\sqrt{2\pi}} \end{aligned} \tag{29}$$

The relation between SSD and SAD can be seen from Eq. 19 and Eq. 29. As for the relation between different sizes, it can be deduced by the same probability theory. Finally, the different sized threshold for SSD have been obtained based on the relation between SSD and SAD together with Eq. 30−33.

$$TH_{SSD_{32\times32}} < 133.7Qp/\cos^2(\frac{\pi}{64}). \tag{30}$$

$$TH_{SSD_{16\times16}} < 74.2Qp/\cos^2(\frac{\pi}{32}). \tag{31}$$

$$TH_{SSD_{8\times8}} < 42.2Qp/\cos^2(\frac{\pi}{16}). \tag{32}$$

$$TH_{SSD_{4\times4}} < 10.3Qp/\cos^2(\frac{\pi}{8}). \tag{33}$$

III. Quadtree analysis based 2-D Zero Node Detection

In order to fully utilize the feature of quadtree and to link it to previous non-tree structure method, the method of analysis a quadtree has been designed novelly by defining a set of descriptors in this section first, which can also be utilized for extending other kinds of algorithm to HEVC quadtree structure. Fig. 8 displays the correlation between TU with CU. In HEVC, the depth of CU together with the maximum CU size can be user defines. For instance, CU_0 is the biggest coding unit. Then it can be split in to smaller sized CU_1 by the CU depth configuration and rate distortion optimization (RDO) cost. The related TU has same way of division according to the user defined TU depth with the limitation of no bigger than related CU. In Fig. 5, the prediction blocks are traversed in alphabetical order. The transform blocks are traversed once a leaf associated with a prediction block is reached. Using depth-1 traversal has the benefit that both the left neighboring block(s) and the top neighboring block(s) are always encoded/transmitted before the current block. Thus, the data already transmitted for these blocks can be used to facilitate rate-constrained encoding of the current block such as, e.g., for the purpose of motion vector prediction, merging of prediction blocks, or context modeling in entropy coding. Each prediction block can be further subdivided for the purpose of transform coding with the subdivision being determined by the corresponding RQT.

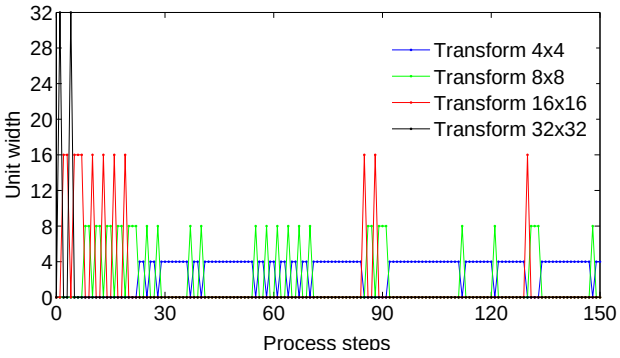


Fig. 5: Process step of Tree structured transform

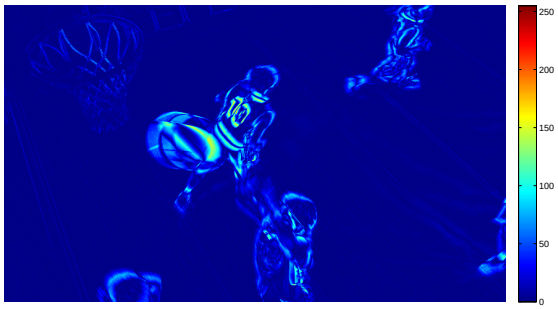
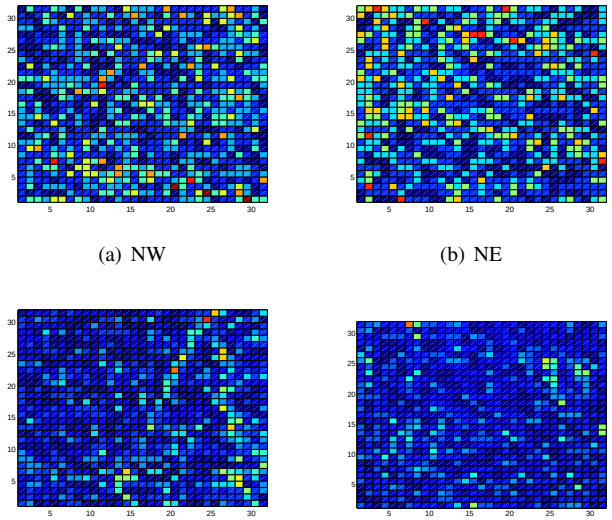


Fig. 6: Distortion of BasketballDrill_832x480

Transform block sizes in the range of 4×4 to 64×64 for the luma component and correspondingly scaled block sizes for both chroma components are supported.

For the current transform and quantization node, which are going to be analysis. The "upper node" together with the "current node" are highly correlated, as well as the neighbor node. In a quadtree, one node has 3 neighbor nodes, however, in image processing, the position and the processing have different weights or orders. Therefore, 4 descriptor have been designed to describe one concrete node. In one split block, the "NW" north west, "NE" north east, "SW" south west, "SE" south east have been marked by the position related to images. While, reflected in a quadtree, they are arranged from left to right. Through these descriptors, for analyzing one node in a quadtree structure, it can be separated into vertical and horizontal analysis, which can fully take advantage of the feature of quadtree structure. That means, from vertical consideration, by analyzing the correlation between upper and current node, the condition can be adaptive to variable size in a quadtree. Meanwhile, by analyzing the correlation between spatial distributions, the condition can be adaptive to spacial feature between neighbor nodes horizontally. After analysis in this two dimension (2-D), we can obtain conditions considering both of the size and depth. With these 5 descriptors, the position of one particular node can be figured out directly, no matter the how many the depth will be. It is particularly suitable for HEVC with recursive CU splits related TU splits. From here, the analysis method will be applied into quadtree based AZB detection. The 2-D conditions have been discussed and proposed here above all. While the vertical conditions can be derived from the traditional AZB method will be described in next section. Fig. 7 displays the block distortion by proposed neighbor node analysis. These are a set of neighbor nodes' distortion values in BasketballDrill_832x480 test sequence between frame 10th and frame 11th, with the block size as 32×32 split from one random selected 64×64 upper node. From this figure, it provides a common image that in most cases between neighbor nodes there will be one energy concentrated like the SE node in this quadtree. The reason of dissecting the spacial distortion is obvious that the all-zero block distribution are



(a) NW (b) NE

(c) SW (d) SE

Fig. 7: Block distortion by proposed neighbor node analysis

mainly affected by two factors, sum of absolute difference (SAD) and quantization parameters (QP), which can examine both distortion and quantization features for AZB, more details can be refereed in next section. The distortion value in Fig. 6 take SAD as an example, the difference or prediction error will concentrated in a small area with big value, and most of part with rare difference or error energy. Residue data as well as the transformed data will follow a distribution like this. It proved the high correlation between neighbor nodes in a quadtree. That is the zero node can be detected not only by fixed conditions as previous works, but also the neighbor node information. Moreover, zero node not the same as previous AZB is that it may become a mother upper node of another quadtree, once the zero node has been predetermined there will be much more AZB calculation released, which will accelerate the encoding process exponentially. After split from the upper node, we are ready to get current nodes and its 3 neighbor nodes. If the correlation between each neighbor nodes can be take advantage to detect AZB also, the calculation of some of

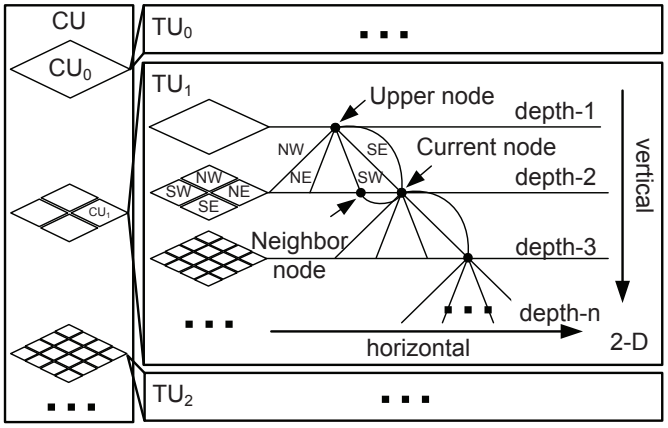


Fig. 8: Proposed descriptors of analysis one quadtree

the neighbor node can be skipped.
Then the 2-D condition is defined below,

$$\begin{aligned}&\text{If } \ max\{D_{NW,}D_{NE,}D_{SW,}D_{SE}\}\\&> D_{upper} - max\{D_{NW,}D_{NE,}D_{SW,}D_{SE}\},\\&\text{Non}-max\{D_{NW,}D_{NE,}D_{SW,}D_{SE}\}\text{are zero nodes.}\end{aligned}$$

Here D means the distortion value, the subscript related to different positions of current quadtree. This predetermine judgement will not adding extra computation as the calculation of TU follows a top-down order. Before the upper node is calculated, the distortion values have been already obtained during the intra or inter process. Horizontally, the D_{upper} can be substituted with $sum\{D_{NW,}D_{NE,}D_{SW,}D_{SE}\}$ while the TU depth is set to 1 or switching off the considering the of the vertical effects.
Therefore, the 1-D horizontal condition together with 2-D condition can be summarized as,

$$\begin{cases} 2max\{D_{neighbors}\} > sum\{D_{neighbors}\}, & depth = 1 \\ 2max\{D_{neighbors}\} > D_{upper}, & otherwise \end{cases} \tag{34}$$

If the neighbor nodes $D_{NW,}D_{NE,}D_{SW,}D_{SE}$ salifying the Eq. 34, these nodes are defined as zero nodes and the related operations are omitted.

IV. Overall scheme

On the basis of the description in the previous sections, we now present our quadtree analysis based 2-D AZB detection algorithm, as shown in Fig. 9. To begin with, all coding tree blocks are coded. The compressing CU_0 is initialized and processed in following steps.

- Step 1. Initialize CU_0 or the CU in upper (depth in 3 depth case).
- Step 2. Inter prediction with SAD AZB detection Eq. 8−11 of current CU.
- Step 3. Intra prediction with SATD AZB detection of CU for both Luma Eq. 21−24 and Chroma Eq. 31−33.
- Step 4. Check the $transform\_split\_flag$ [15]. If yes, jump to Step 6. If no, jump to Step 7.

TABLE I: Important settings for the encoding process

Config Parameters	Values
MaxCUWidth & MaxCUHeight:	64
IntraPeriod :	1
FrameSkip:	0
FrameToBeEncoded:	100
QP:	22,28,33,38
SearchRange:	64
NumberReferenceFrames:	2
Entropy coder :	CABAC

- Step 5. Check the proposed 2-D zero node conditions Eq. 34.
- Step 6. Initialize CU_1 size. Then jump to Step 2.
- Step 7. Check the RDO of Step 3 and Step 4.
- Step 8. Out put the best compressed CU.

V. Experimental Result

In all-zero block detection field, the detection ratio, time saving of DCT and quantization part and objective video performance are generally utilized for proving the efficiency of the algorithm. This section evaluates the proposed scheme the same as other referenced works. However, the previous definition of detection ratio is ambiguous for tree structure, the main effort is focused on the total encode time saving together with objective video performance. The proposed quadtree based 2-D all-zero block detection is implemented in HM1.0 software [16] and tested under the recommend test condition with Intra, high-efficiency setting according to [17]. Three resolution types of HEVC test sequences, class B, D and E [17], are used which include commonly scene cases: fast motion, slow motion, close shot, long shot, detailed, less detailed and etc.. Table I displays the important settings for encoding process. Since the difference between two RD curves is trivial, BDBR and BDPSNR [18] are employed, which are respectively the average difference in the bit rate and the PSNR difference between our algorithm and the original HM respectively, to measure video quality. The result is shown in Table II. 'P' means the overall proposal, the superscript of 'P' denotes the depth of TU and the subscript of 'P' denotes the maximum TU size. The '+' in BDBR represents the bit-rate gain while '-' in BDPSNR means PSNR degradation. It is shown that, the overall quality loss is very small. The largest bit rate loss (+3.55%) and PSNR degradation (-0.024dB) are in the fast motion BasketballDrive 1080p sequence, proposed AZB method are also proved effectiveness for slow motion sequences especially. sequence, and are still very small. Furthermore, the total encoding time reduction is listed in Table III. Here we calculate the total encoding time reduction (TS) based on Eq. 35, where $TIME_{HEVC}$ is the total encoding time consumed by the HM1.0 and $TIME_{proposal}$ is the total encoding time for our algorithm. Our proposed algorithm achieved a time reduction of 3.81% to 25.47% for the different sequences. To be mentioned, the proposed algorithm plays a more important role in total encode time saving, not only DCT and quantization, than previous method in H.264. By adding the tree structure of

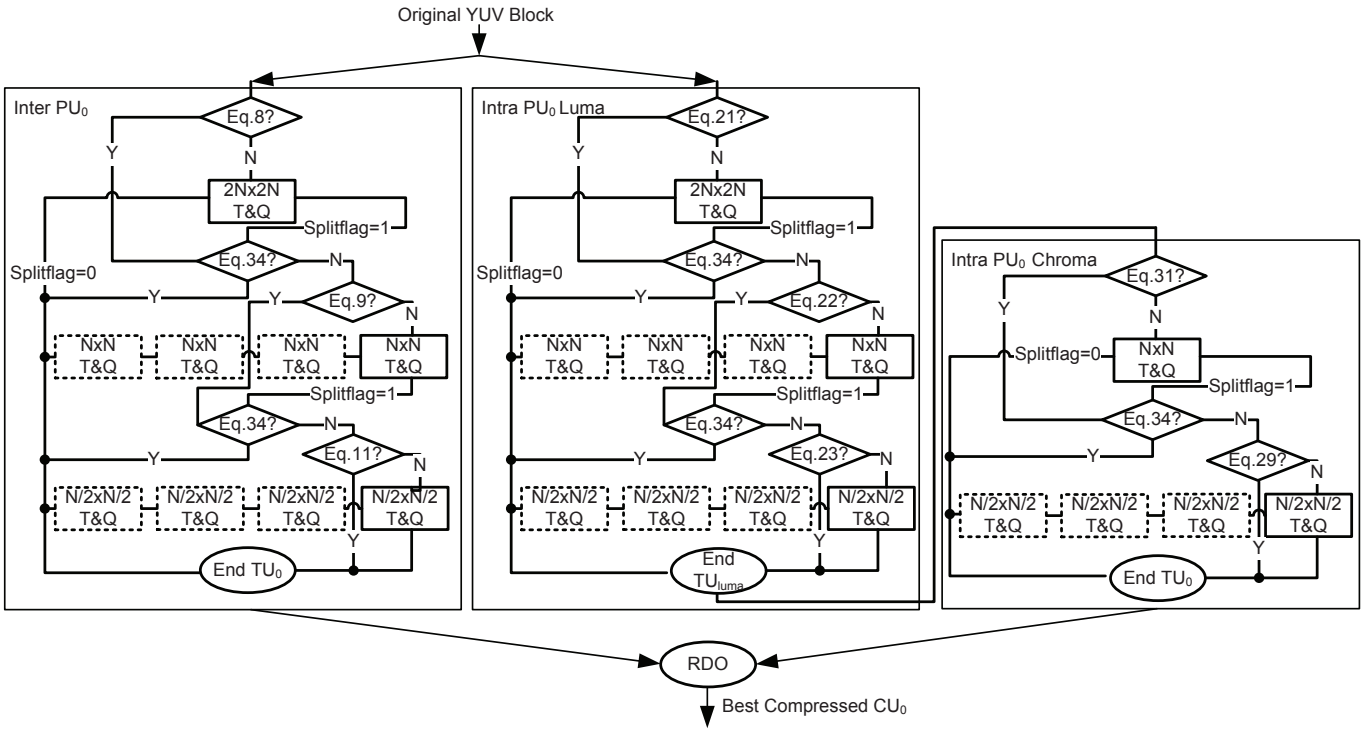


Fig. 9: Flow chart of proposed overall scheme

TABLE III: Total encode time saving

Sequences	TS			
	$P^2_{16\times16}$	$P^3_{16\times16}$	$P^2_{32\times32}$	$P^3_{32\times32}$
BasketballDrill	15.30%	5.45%	24.21%	13.87%
PartyScene	16.77%	6.08%	26.82%	15.15%
BQMall	16.30%	5.92%	25.37%	14.38%
832x480 Average	16%	5.82%	25.47%	14.47%
vidyo1	12.59%	3.67%	20.53%	11.86%
vidyo2	13.25%	3.93%	21.89%	11.83%
vidyo3	13.03%	3.85%	21.36%	12.06%
1280x720 Average	12.95%	3.81%	21.26%	11.92%
BQTerrace	14.86%	5.23%	24.53%	12.23%
BasketballDrive	13.51%	4.64%	21.39%	12.54%
Cactus	15.03%	12.02%	23.58%	13.23%
Kimono	13.26%	3.74%	21.75%	12.11%
ParkScene	15.33%	5.23%	24.85%	13.02%
1920x1080 Average	14.40%	6.17%	23.22%	12.63%

traversal calculating and large transform and quantization, it is much more effective to accelerate whole encoder by applying proposed tree structure based AZB method.

$$TS = \frac{TIME_{HEVC} - TIME_{proposal}}{TIME_{HEVC}} \times 100\% \quad (35)$$

Where TS is the total encoding time saving as a percentage between the proposed method and the HEVC.

VI. Conclusions

In this paper, a fast quadtree based all-zero block detection algorithm is proposed. It mainly consists of two contributions. First, for tree structured all-zero block detection, this paper designs a way of analysis the quadtree structure which can be extend to other kind of algorithm related to tree structure. And the variable sized adjustment scheme can reduce the number of redundant all-zero block calculation using the strategy of zero node detection, which predetermine useless calculate for the set of subblocks. Second, for large sized transform and quantization based all-zero block sufficient SAD and SSD condition have been derived for both inter and intra prediction. Moreover, to coordinate with multiple block sized quadtree structure, small sized sufficient conditions are also been defined as well as the chroma. And the proposal is a hardware-friendly algorithm that can be prospected to be embedded into a real-time hardware encoder in the future.

Acknowledgment

This research is supported by KAKENHI (23300018) and KDDI Foundation. The research is also supported by the Ambient SoC Global COE Program of Waseda University under the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] Int. Standards Org. /Int. Electrotech. Comm. (ISO/IEC) JTC 1, ISO/IEC 11 172-2 (MPEG-1), “Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s-Part 2: Video”, Mar, 1993.

[2] Int. Telecommun. Union-Telecommun. (ITU-T), Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, Rec. H.262 and ISO/IEC 13 818-2 (MPEG-2 Video), “Generic coding of moving pictures and associated audio information-Part 2: Video”, Nov, 1994.

[3] Int. Telecommun. Union-Telecommun. (ITU-T), Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC) JTC 1, Rec. H.264 and ISO/IEC 14 496-10 (MPEG-4) AVC, “Advanced video coding for generic audiovisual services”, 2003.

TABLE II: Performance analysis

Sequences		BD PSNR				BD BR			
		$P^2_{16\times16}$	$P^3_{16\times16}$	$P^2_{32\times32}$	$P^3_{32\times32}$	$P^2_{16\times16}$	$P^3_{16\times16}$	$P^2_{32\times32}$	$P^3_{32\times32}$
832x480	BasketballDrill	-0.0802	-0.0522	-0.1033	-0.0937	+3.0802	+2.5522	+5.1375	+3.4689
	PartyScene	-0.0296	-0.0152	-0.0451	-0.0372	+1.9325	+1.2522	+0.6096	+0.3147
	BQMall	-0.0764	-0.019	-0.0536	-0.0396	+3.0702	+2.0011	+4.9096	+1.6096
	Average	-0.0621	-0.0288	-0.0673	-0.0568	+2.6943	+1.935	+3.5522	+1.7978
1280x720	vidyo1	-0.0135	-0.0035	-0.0268	-0.0172	+0.7837	+0.6522	+0.8330	+0.7172
	vidyo3	-0.0296	-0.0152	-0.0311	-0.0244	+0.0766	+0.6589	+0.6096	+0.3147
	vidyo4	-0.1161	-0.0199	-0.1524	-0.0987	+1.0802	+0.9522	+3.9096	+2.6096
	Average	-0.0531	-0.0129	-0.0701	-0.0468	+0.6468	+0.7544	+1.7841	+1.214
1920x1080	BQTerrace	-0.0135	-0.0733	-0.0216	-0.0145	+2.0802	+1.5522	+0.3172	+2.0271
	BasketballDrive	-0.1376	-0.1030	-0.2435	-0.202	+2.3694	+1.5721	+1.2535	+0.3681
	Cactus	-0.0183	-0.0057	-0.0322	-0.0194	+0.0433	+0.3577	+0.5865	+0.1413
	Kimono	-0.0374	-0.0401	-0.0711	-0.0588	+2.7522	+2.0851	+3.1617	+3.004
	ParkScene	-0.0159	-0.0118	-0.437	-0.0266	+0.1452	+0.0855	+0.3632	+0.2708
	Average	-0.0445	-0.0469	-0.1611	-0.0643	+1.4781	+1.1305	+1.1364	+1.1623

[4] K. H. Lee, E. Alshina, J. H. Park, W. J. Han, and J. H. Min, "Technical considerations for ad hoc group on new challenges in video coding standardization", in Proc. 85th MPEG Meeting, Jul. 2008, no. M15580.

[5] E. Alshina, K. H. Lee, W. J. Han, and J. H. Park, "Technical considerations for ad hoc group on new challenges in video coding standardization", in Proc. 86th MPEG Meeting, Oct. 2008, no. M15899.

[6] A. Yu, R. Lee, and M. Flynn, "Early detection of all-zero coefficients in H.263", in Proc. Picture Coding Symp. (PCS), 1997, pp. 159C164.

[7] X. Zhou, Z. Yu, and S. Yu, "Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantization", Electron. Lett., vol. 34, no. 19, pp. 1839C1840, Sep. 1998.

[8] L. A. Sousa, "General method for eliminating redundant computations in video coding", Electron. Lett., vol. 36, no. 4, pp. 306-307, Feb. 2000.

[9] S. Jun and S. Yu, "Effcient method for early detection of all-zero DCT coeffcients", Electron. Lett., vol. 27, no. 3, pp. 160C161, Feb. 2001.

[10] Y. Wang, Y. Zhou, and H. Yang, "Early detection method of all-zero integer transform coeffcients", IEEE Trans. Consum. Electron., vol. 50, no. 3, pp. 923C928, Aug. 2004.

[11] G. Y. Kim, Y. H. Moon, and J. H. Kim, "An improved early detection algorithm for all-zero blocks in H.264 video encoding", IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 8, pp. 1053C1057, Aug. 2005.

[12] H. Wang, S. Kwong, and C.-W. Kok, "Effcient prediction algorithm of integer DCT coeffcients for H.264/AVC optimization", IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 4, pp. 547C552, Apr. 2006.

[13] J. R. Price and M. Rabbani, "Biased reconstruction for jpeg decoding", IEEE Signal Process. Lett., vol. 6, pp. 297-299, Dec. 1999.

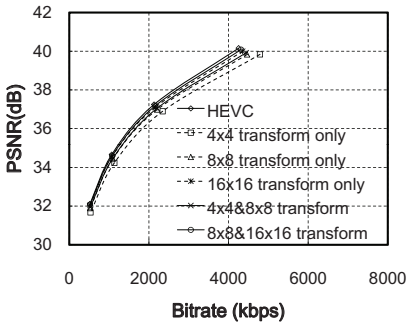
[14] R. C. Reininger and J. D. Gibson, "Distributions of two-dimensional DCT coefficients for images", IEEE Trans. Commun., vol. 31, pp.835-839, Jun. 1983.

[15] T. W. Han, J. R. Ohm, and G. J. Sullivan: "Working Draft 1 of High-Efficiency Video Coding", JCTVC-C403, Guangzhou, China, October 2010.

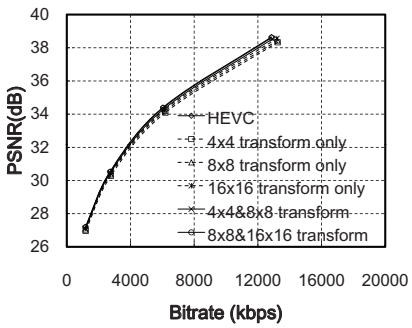
[16] Reference software of HEVC Test Model (HM1.0), Jan, 2011. https://hevc.hhi.fraunhofer.de /svn/svn_TMuCSoftware/tags/HM-1.0.

[17] F. Bossen, "Common test conditions and software reference configurations", JCTVC-D600, Daegu, Korea, Jan. 2011.

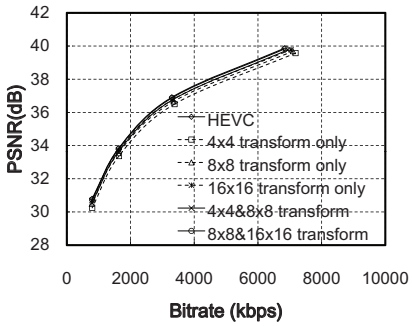
[18] G. Bjontegaard, "Calculation of average psnr differences between rd-curves", VCEG-M33, April 2001.



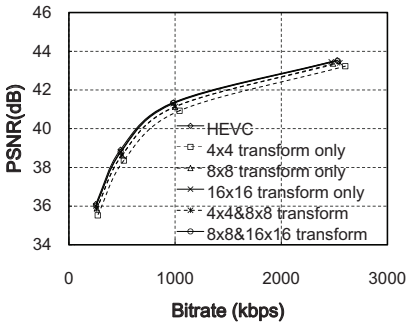
(a) BasketballDrill_832x480



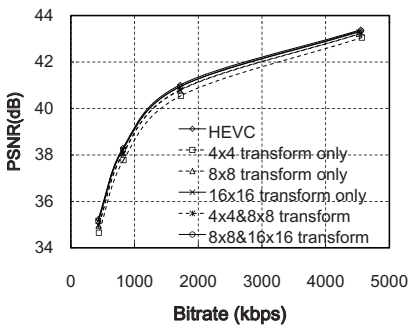
(b) PartyScene_832x480



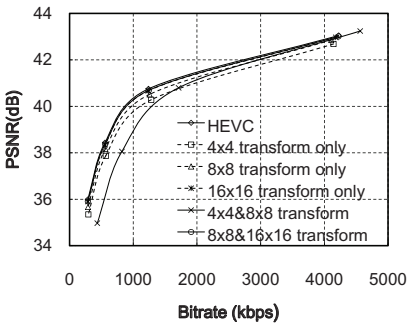
(c) BQMall_832x480



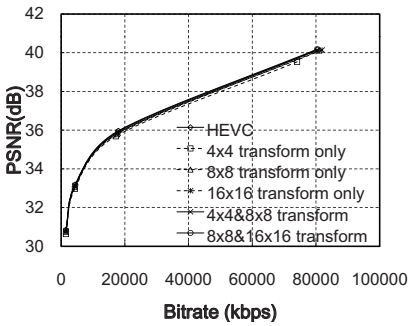
(d) vidyo1_1280x720



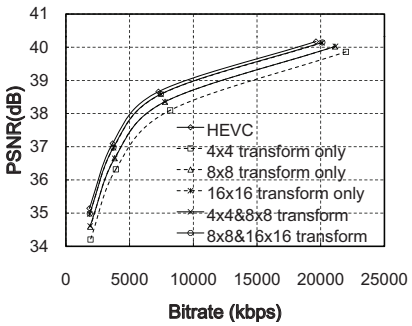
(e) vidyo3_1280x720



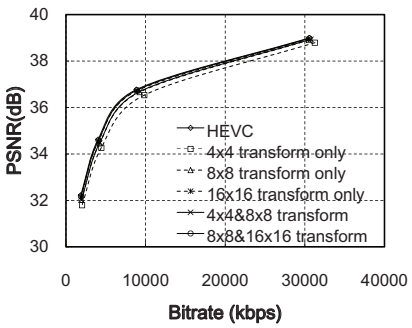
(f) vidyo4_1280x720



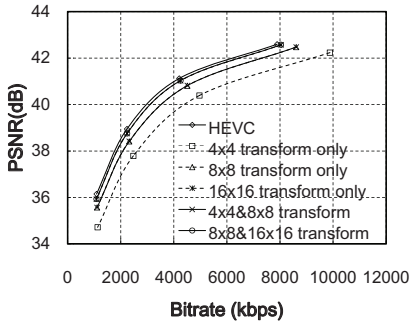
(g) BQTerrace_1920x1080



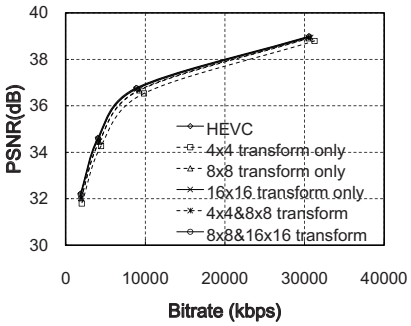
(h) BasketballDrive_1920x1080



(i) Cactus_1920x1080



(j) Kimono_1920x1080



(k) ParkScene_1920x1080

Fig. 10: RD curve of HEVC test sequences