

Quadtree Based Nonsquare Block Structure for Inter Frame Coding in High Efficiency Video Coding

Yuan Yuan, Il-Koo Kim, Xiaozhen Zheng, Lingzhi Liu, *Senior Member, IEEE*, Xiaoran Cao, Sunil Lee, Min-Su Cheon, Tammy Lee, Yun He, *Senior Member, IEEE*, and Jeong-Hoon Park

Abstract—A concept of a quadtree based nonsquare block coding structure is presented in this paper for the emerging High Efficiency Video Coding standard, which includes a quadtree based asymmetric motion partitioning scheme and a nonsquare quadtree transform (NSQT) algorithm. Nonsquare motion partitioning in inter frame coding provides the possibility of getting more accurate prediction results by splitting one coding block into two nonsquare prediction blocks. Contrary to the traditional symmetric motion partitions (SMP), asymmetric motion partitions (AMP) are proposed to improve the coding efficiency, especially for the coding blocks with irregular object boundaries. NSQT is designed for nonsquare prediction blocks (SMP and AMP), which combines square and nonsquare transform blocks in a unified transform structure. It exploits the directional characteristic of an image block to improve the transform efficiency. The combination of nonsquare partitions and NSQT provides high coding flexibility and low implementation cost for both encoder and decoder design. Simulation results show that about 0.9%–2.8% bit-rate saving can be achieved in terms of different configurations, and subjective quality can also be improved.

Index Terms—Asymmetric motion partitions, High Efficiency Video Coding (HEVC), nonsquare quadtree transform (NSQT), quadtree structure.

I. INTRODUCTION

THE GOAL of video coding techniques is to achieve a high compression ratio while providing good image quality. In the past few decades, block partitioning techniques have been proposed to solve the nonstationary problem in

Manuscript received April 15, 2012; revised July 20, 2012; accepted August 21, 2012. Date of publication October 5, 2012; date of current version January 8, 2013. This work was supported in part under Grants 973-2009CB320903 and 2010ZX03004-003. This paper was recommended by Associate Editor J. Ridge.

Y. Yuan, X. Cao, and Y. He are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: y-yuan08@mails.tsinghua.edu.cn; caoxr05@mails.tsinghua.edu.cn; hey@tsinghua.edu.cn).

I.-K. Kim, S. Lee, M.-S. Cheon, T. Lee, and J.-H. Park are with the Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon 443-742, Korea (e-mail: ilkoo.kim@samsung.com; sunil.lee@samsung.com; minsu.cheon@samsung.com; tammy.lee@samsung.com; jeonghoon@samsung.com).

X. Zheng is with the Research Department, Hisilicon Semiconductor and Component Business Department, Huawei Technologies, Shenzhen 518129, China (e-mail: zhengxiaozhen@huawei.com).

L. Liu was with the ASIC Department, Huawei Technologies, Santa Clara, CA 95050 USA. He is now with Intel Corporation, Santa Clara, CA 95052 USA (e-mail: lingzhi.liu@ieee.org).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2012.2223037

video coding [1]. Before prediction and residual coding, a frame is segmented into blocks with variable sizes according to its contents.

The concept of block partitioning has been used for prediction block and transform block in many video coding standards [2], [3]. In the AVC/H.264 standard [2], [4], a two-level tree-structured partitioning scheme is developed for motion estimation. A macroblock of size 16×16 is motion-compensated by 16×16 , 16×8 , 8×16 , or 8×8 prediction blocks, while each 8×8 prediction block can be further split into 8×4 , 4×8 or 4×4 subblocks. An adaptive block transform technique, which selects a transform size between 4×4 and 8×8 , is designed for improving the transform efficiency [5]. The undergoing High Efficiency Video Coding (HEVC) standard [6] extends the two-level tree-structured partitioning scheme to a unified quadtree structure. The quadtree structure is used to represent variable coding block sizes and transform block sizes [7]–[9].

The work described in this paper is based on the framework of HEVC. HEVC is developed by a Joint Collaborative Team on Video Coding (JCT-VC) that is led by ITU-T SG16 WP3 Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG 11 (MPEG). This activity began by issuing a joint call for proposals (CfP) in January 2010 [10], and attracted the interest of many organizations from academia and industry. Afterward, a HEVC test model (HM) was established as the reference software used to evaluate the tools proposed to HEVC. HEVC aims to provide a video compression technology that has significantly higher compression capability than the AVC/H.264 standard, especially for a high resolution video [10]. Its applications include video streaming, surveillance, videoconferencing, video storage, and so on [10].

HEVC is still based on the conventional block-based hybrid coding scheme, including intra and inter prediction, residual transform coding, entropy coding of transform coefficients, in-loop filtering, and some high-level functionalities [3]. Three types of units make up the coding structure of HEVC: coding unit (CU), prediction unit (PU), and transform unit (TU). A CU is similar to the concept of macroblock in AVC/H.264 that contains coding block information and associated coding data. A PU contains prediction blocks and the side information for the prediction process, e.g., reference index, motion vector deviation. A TU contains transform blocks and information for describing the transform tree [3], [11].

A quadtree is an efficient structure for representing the various coding block sizes and subdivision process of trans-

form blocks in HEVC [7], [8]. The recursive decomposition of quadtree not only provides a flexible block partitioning mechanism, but also gives an inherent and highly efficient signaling solution.

In view of the different motion properties of pixels, a coding block is allowed to be further divided into smaller prediction blocks in motion estimation. As a complement to the previous square-shaped or nonsquare symmetrically partitioned prediction blocks, asymmetric motion partitions (AMP) are proposed in this paper. AMP includes four partition modes: $2N \times nU$, $2N \times nD$, $nR \times 2N$ and $nL \times 2N$, which separate a coding block into two asymmetric prediction blocks along with horizontal or vertical directions.

In addition, a nonsquare quadtree transform (NSQT) structure is proposed to provide a more appropriate transform structure for nonsquare prediction blocks, which include symmetric and asymmetric motion partitions. It integrates two types of nonsquare transforms ($0.5N \times 2N$ and $2N \times 0.5N$, in which N is an integer value) into the transform quadtree structure. NSQT preserves the flexibility of the quadtree structure and improves the transform efficiency by using the properties of image content. AMP and NSQT were adopted into HM4 and the 4th working draft (WD4) at the sixth JCT-VC meeting [12], [13] due to its improvement of coding efficiency with reasonable implementation complexity. In light of the maturities of the tools, AMP is included in the HEVC main profile [3] and NSQT is under consideration by the HEVC future profile(s).

The rest of this paper is organized as follows. Section II introduces the quadtree structure in video coding, which is a basic framework of this paper. The proposed quadtree-based nonsquare motion partitioning and nonsquare quadtree transform are discussed in Sections III and IV, respectively. Some related works are also explained in these two sections. Section V describes the implementation details of the proposed structure. The statistical analysis and simulation results are given in Section VI. Conclusions are given in Section VII.

II. QUADTREE STRUCTURE IN VIDEO CODING

In video coding, block partitioning aims to select the best coding block size according to the image content. Dealing with regions with complex texture, smaller blocks can get more accurate prediction results but require extra signal bits. Thus, the optimization of the tradeoff between coding efficiency and complexity is the major problem. In previous video coding standards, a tree-based coding structure has been explored to solve this problem. Taking AVC/H.264 as an example in Fig. 1, a two-level tree-structured block partitioning strategy is used at the basic processing unit, named macroblock. A macroblock with 16×16 pixels can be motion estimated by 16×16 , 16×8 , 8×16 or 8×8 prediction blocks. If 8×8 prediction blocks get the best R-D performance among the four, the 8×8 block, so-called submacroblock, can be further partitioned into 8×8 , 8×4 , 4×8 or 4×4 subblocks [4].

A more flexible block partitioning algorithm, which is adopted into the undergoing standard HEVC, employs quadtree as the basic coding structure. A picture is first separated into a number of coding blocks with the maximum

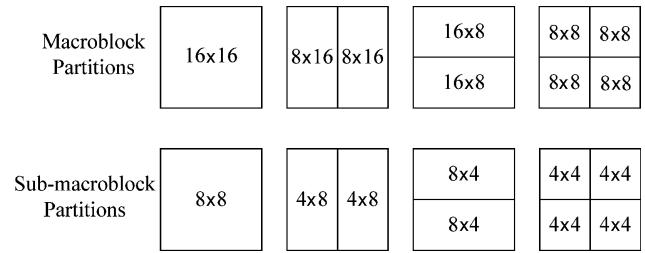


Fig. 1. Prediction blocks in AVC/H.264. Each 8×8 block can be further partitioned into 4×8 , 8×4 , and 4×4 subblocks.

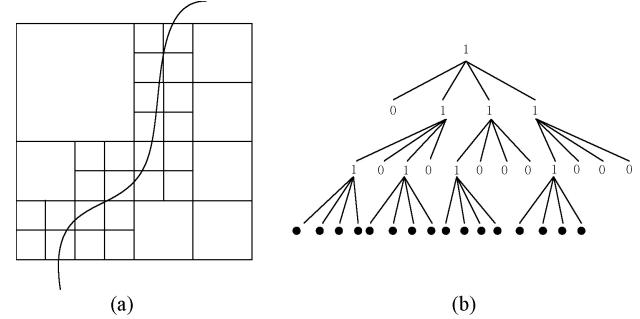


Fig. 2. Example of quadtree structure. (a) Quadtree decomposition for an arbitrary boundary. (b) Corresponding signaling structure.

coding block size, and then the segmentation of each maximum coding block is signaled by the quadtree semantics that indicates how the blocks are subdivided for prediction and residual coding. Regarding decomposition process of a $2^N \times 2^N$ coding block with a D depth quadtree, the blocks at level n (the root node corresponds to level 0) will have a size of $2^{N-n} \times 2^{N-n}$, where $0 \leq n < D$. Fig. 2(a) shows an example of the quadtree segmentation for a coding block separated by a random curve which represents an object boundary. Smaller blocks are distributed around the boundary, while larger blocks are located at the background or the flat areas. The corresponding signaling structure is described in Fig. 2(b). The decomposition of a coding block is determined by using a Lagrangian cost function described in [14], which considered both the coding bits and the distortion between the original block and the reconstructed block.

In HEVC, a quadtree is also used to decompose the transform block. The root node of the transform quadtree corresponds to the leaf node of coding block quadtree and the transform quadtree can be seen as an extension of the leaf node of the coding block quadtree in block partitioning [15]. The recursive splitting method for transform blocks is called residual quadtree transform (RQT) [8], [16]. Quadtree-structured decomposition for coding block and transform block realizes a good tradeoff between coding efficiency and implementation cost.

In traditional coding structure, square and nonsquare symmetric motion partitions are used in inter prediction, and only square blocks are allowed in the transformation process. To further improve prediction accuracy and transform efficiency, the asymmetric motion partitions and nonsquare transform structure will be discussed in the next two sections, respectively.

III. QUADTREE BASED NONSQUARE BLOCK PARTITIONING

A. Nonsquare Motion Partitioning Strategies in Literature

Motion-compensated prediction, which is a core part of modern video codec design, is performed based on a certain unit basis. In previous video coding standards, such unit is defined as partition or motion partition. The shapes of motion partitions are fixed, e.g. 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , and 4×4 for AVC/H.264 [2], [4]. Among fixed-shape partitions, a shape that has optimal coding efficiency is chosen and signaled to the decoder. This type of partitioning strategy can be classified as a predetermined partition shape approach with explicit signaling. This approach is adopted into most of video coding standards, e.g., MPEG-1/2, AVC/H.264 and even in HEVC [3].

Another type of motion partitioning strategy is to use arbitrary partition shapes with explicit shape signaling [17]–[19]. In this type a simplified motion partition map that can be represented by a line, an angle of the line, a center position, etc. is encoded and signaled to the decoder. One typical scheme of this type called geometry motion partition was proposed to HEVC [19], in which the partition is defined by an angle and the distance of the partition line from the origin. It reported a 3% average bit-rate saving was achieved with 90% encoding time increase. The significantly encoding time increase is due to the traverse of a number of possible partitions [20]. Moreover, the motion partition map is also need to be stored in decoder side that increases the memory storage.

In HEVC, the fixed-shape partition approach is adopted as a basic form of motion partitioning strategy based on the consideration of complexity issue. Different from macroblock and submacroblock partition in AVC/H.264, the prediction block size in HEVC is related to the coding block size. Since the coding block size ranges from 64×64 to 8×8 , the prediction block size can range from 64×64 to 8×8 according to the corresponding coding block size. This kind of generalized partition scheme leads to a more consistent and unified design of prediction and video coding structures. It should be noted that the prediction block is defined only for a leaf node of a coding block quadtree and its size cannot be greater than the coding block size.

In initial HEVC design, there are four different possible partition modes for inter prediction: two square partition modes (PART_2N×2N and PART_N×N) and two symmetric motion partition (SMP) modes (PART_2N×N and PART_N×2N). It should be noted that PART_N×N is allowed only for the minimum coding block size and has been disabled for inter frame coding in the HEVC main profile [3]. Fig. 3(a) shows possible partition mode according to the different prediction mode for a coding block of size $2N \times 2N$ in which N is an integer value. All possible prediction modes are traversed, and the one having minimum R-D cost defined in [14] will be used.

B. Proposed Asymmetric Motion Partitions

The square and symmetric partition modes are well known since they exist in AVC/H.264. By contrast, AMP are new inter prediction partitions in HEVC design. The AMP illustrated

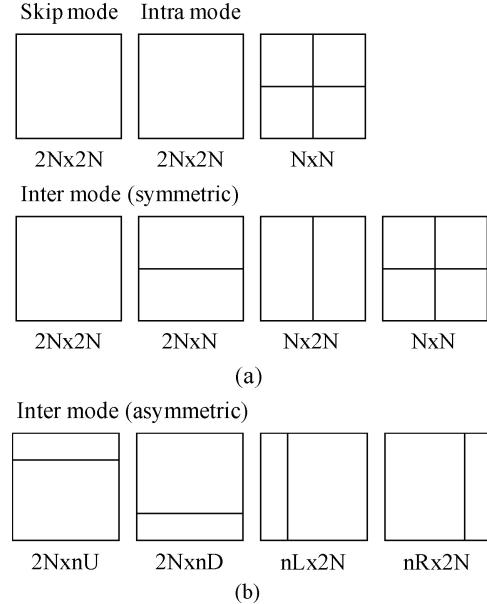


Fig. 3. Partition modes specified in HEVC standard.

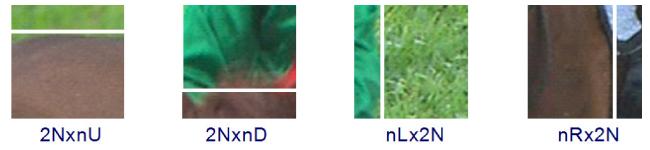


Fig. 4. Examples of AMP modes from *RaceHorses* sequence with 832×480 resolution.

in Fig. 3(b) includes four partitions modes: PART_2N×nU, PART_2N×nD, PART_nL×2N, and PART_nR×2N. “n” in AMP modes represents smaller size partition and U, D, L, and R represent up, down, left, and right, respectively. As an example, PART_2N×nD means the horizontal splitting mode that has a smaller size partition in the lower part of the $2N \times 2N$ block.

The AMP modes are designed to improve the coding efficiency for the irregular object boundaries, which cannot be represented by square partitions. Without AMP modes, those boundaries can only be represented with smaller square partitions which will spend more bits. That is, by adding four additional partition modes, efficient representation of irregular patterns using flexible motion partitions can be realized. Although more motion partition modes are better for coding efficiency, considering the tradeoff between coding efficiency and complexity, especially, encoder side complexity, four AMP modes are carefully selected. Fig. 4 depicts the example cases in which AMP modes are used, which illustrates that AMP modes are effectively used for textures that cannot be represented by square or symmetric motion partitions. The two asymmetric partitions having different textures can be well predicted.

Table I shows the possible shapes of the nonsquare prediction block if $2N$ is equal to 64, 32, 16, and 8 that is in accord with the design of HEVC. It is should be noted that AMP modes for minimum coding block size are not allowed in the HEVC design.

TABLE I
SHAPES OF NONSQUARE PREDICTION BLOCKS

Coding Block Size	Horizontal Partition Modes ($2N \times N$, $2N \times nD$, and $2N \times nU$)	Vertical Partition Modes ($N \times 2N$, $nR \times 2N$, and $nL \times 2N$)
64×64	64×32 , 64×48 , 64×16	32×64 , 48×64 , 16×64
32×32	32×16 , 32×24 , 32×8	16×32 , 24×32 , 8×32
16×16	16×8 , 16×12 , 16×4	8×16 , 12×16 , 4×16
8×8	8×4	4×8

Given the prediction mode and the partition mode for the current coding block, all information related to prediction is signaled on a prediction block basis, for instance, merge flag, merge index, inter prediction flag, motion vector prediction index, reference index, and motion vector difference for inter [3]. Motion vector prediction and motion compensation are also performed on a prediction block basis.

IV. NONSQUARE QUADTREE TRANSFORM

A. Nonsquare Transform in Literature

The properties of nonsquare transforms have been studied in the past few decades [21]–[25]. Reference [21] is one of the earliest designs of nonsquare transform for image coding, in which nine rectangular block sizes are used for prediction and transform, corresponding to the combinations of 8, 16, and 32 pixels in width and height. In [22], 16×8 , 8×16 , 8×4 , and 4×8 block sizes are proposed for H.26L, and the transform blocks have the same size as the corresponding prediction blocks. Afterwards, a simplified adaptive block transform (SABT) [23] was proposed as an improvement of [22]. In SABT, the prediction blocks whose width and height are above or equal to eight are transformed by the 8×8 transform. In contrast, the other prediction blocks that have smaller sizes in width or height, the transform size are equal to the size of the prediction block. VC-1 also employed 4×8 and 8×4 nonsquare transforms [24]. A spatially varying transform [25] strategy is also proposed to utilize the characteristics of prediction error by varying the transform block size.

Although the previous work have taken account of some properties and benefits of nonsquare transform, there is still space to further improve the transformation efficiency of nonsquare prediction blocks. Take the 8×16 partition mode for example, the residual blocks composed of two 8×16 prediction blocks are transformed by four 8×8 blocks, or by two 8×16 blocks in previous designs. It is worthwhile considering both features of 8×8 transform and 8×16 (or the transform with nonsquare shape) in a unique design with low implementation cost. Considering that one 16×16 transform block will perform more efficiently than subdivided transform blocks if the pixels can be well predicted and the residuals inside the prediction blocks are near zero value, it is desirable to incorporate larger square and nonsquare transform in the design.

In the next subsection, the energy compaction capabilities of square and nonsquare transforms are analyzed for Laplacian sources. Based on the analysis, a unified design for square and nonsquare transforms will be provided in Section IV-C.

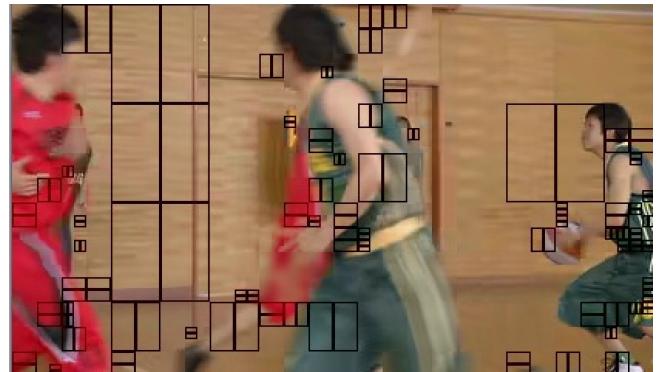


Fig. 5. Nonsquare motion partitions marked with dark bold lines in the 399th frame of *BasketballPass* sequence.

B. Transform Efficiency of Laplacian Sources

Motion estimation and compensation cannot always produce accurate results as there are many different motions in real world. It is better to partition a coding block into smaller prediction blocks along the object boundary in such complex motions. The nonsquare partition modes SMP and AMP use prediction boundaries to approach the object boundary. Fig. 5 demonstrates the distribution of nonsquare prediction blocks in a frame of *BasketballPass* sequence. The nonsquare prediction blocks are marked by dark bold line and the unmarked areas are coded by $2N \times 2N$, skip or intra modes. It can be observed that nonsquare prediction blocks are mostly located at the areas with object boundaries especially with horizontal or vertical structural details, such as the doorframe and the contour of the arm in Fig. 5. In other words, the shape of prediction block or the partition mode in HEVC can reveal the local texture. A study of adaptive rectangular partitions also provides evidence that nonsquare partitions are mostly distributed in detailed boundary regions [26]. Furthermore, statistical study results show that horizontal and vertical frequencies take dominated power of the directional energy in an image [27]. Therefore, a coding block often contains vertical texture if the block is motion compensated by vertical partition modes ($2N \times N$, $2N \times nU$ or $2N \times nD$) while horizontal texture would exist if horizontal partition modes ($N \times 2N$, $nL \times 2N$, or $nR \times 2N$) are used.

The features of motion compensation residuals have been studied in [28]. It stated that the object boundaries have 1-D structures in motion compensation residual and the correlation of residual pixels in one direction is significantly larger than that of the other direction. Therefore, the texture properties of image content also exist in residual signal which can be used to optimize the transform structure. The properties of residual signal for nonsquare partition modes used in the design of NSQT which will be further discussed in the subsequent section.

To compare the transform efficiency, the residual distribution of nonsquare prediction block is statistically modeled. In previous work, the pixels inside a residual block are often modeled by a zero mean Laplacian distribution [29], [30].

The probability density function of a Laplacian distribution is

$$f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right) \quad (1)$$

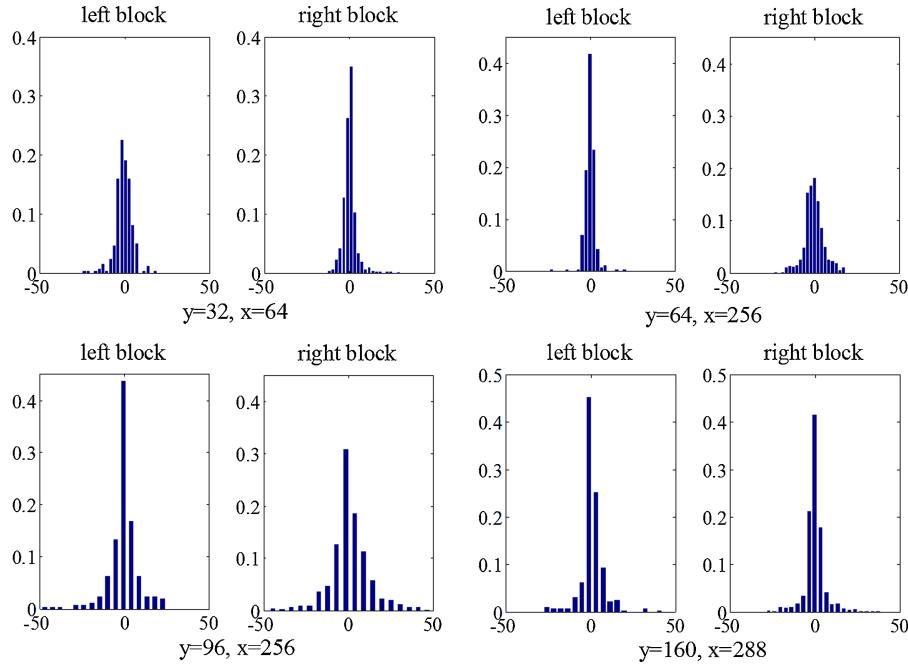


Fig. 6. Residual distributions of nonsquare prediction blocks. The values of the parameters y and x represent the coordinate of the top left corner of each coding block.

where μ is the mean of the distribution and b is a scale parameter. $\mu = 0$ represents zero mean distribution.

The statistics results from the authors show that the residuals inside nonsquare prediction blocks also follow Laplacian distributions with zero mean. Fig. 6 illustrates the residual distributions of four 32×32 coding blocks with $nL \times 2N$ partition mode, which means the left block has a size of 8×32 and the right block is with 24×32 block size. These four coding blocks are randomly chosen from the eighth frame of *BasketballPass* sequence and coded by HM7.0 in Low Delay B Main configuration. The locations of the four coding blocks are listed in Fig. 6. To further verify the Laplacian property of the residual in the two prediction blocks, Fig. 7 shows scatter plots of b_0 and b_1 from all 32×32 blocks with $nL \times 2N$ partition mode in the *BasketballPass* sequence. The diagonal dash line means $b_0 = b_1$. The statistics show that b_0 and b_1 have different values in most cases.

Based on the above analysis, it can be concluded that the two nonsquare residual blocks follow two Laplacian distributions with zero mean and scale parameters b_0 and b_1 . In the test described below, a 32×32 residual block predicted by $nL \times 2N$ partition mode is used as the example. The performance of nonsquare transform and square transform are compared based on the Laplacian model mentioned above.

Typically, a 32×32 residual block can be transformed by a 32×32 block or 16×32 , 8×32 or 16×16 subblocks. Therefore, we perform one 32×32 transform, two 16×32 transforms, four 8×32 transforms and four 16×16 transforms to the residual block respectively. The transformed blocks are represented by $C_{N \times M}$, where $C_{N \times M}$ is a 32×32 matrix with $c_k (k = 1 \dots 1024)$ as its elements. N and M are the width and height of its transform block.

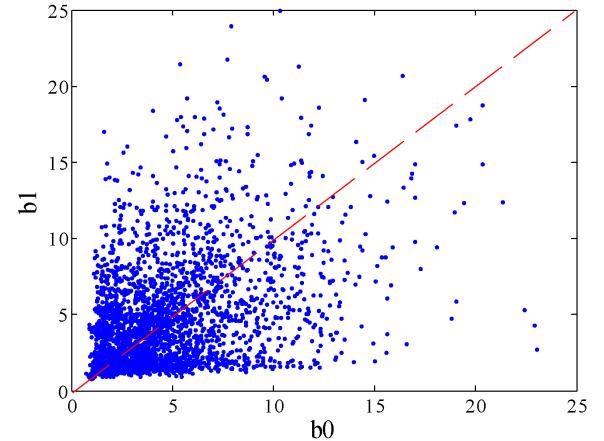


Fig. 7. Scatter plots of different b_0 and b_1 of Laplacian distribution for nonsquare residual blocks. b_0 and b_1 are the scale parameters for left and right blocks, respectively.

If the elements c_k are ranked in a decreasing order of their absolute values, i.e., $abs(c_1) \geq abs(c_2) \geq \dots \geq abs(c_{1024})$, then we define a function as follows:

$$J_m = \sum_{k=m+1}^{1024} |c_k| \Big/ \sum_{k=1}^{1024} |c_k|. \quad (2)$$

The J_m in (2) is the transformation error for a restriction m [31]–[33], which represents the normalized transformation error of the first largest m elements. In other words, a smaller J_m implies better energy compaction of the transform. Different to the covariance matrix used at the experiment in [31], the data from ten thousand blocks with the same Laplacian distribution are used as the input signal in this paper. Considering that (2) represents the absolute error of restricting c_k to m degrees of

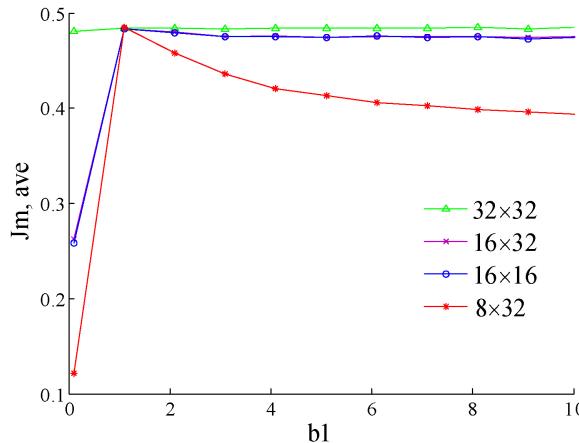


Fig. 8. Transformation errors of different b_1 for 32×32 , 16×32 , 16×16 , and 8×32 block sizes when $b_0 = 1$ and $m = 256$.

freedom (i.e., c_k is allowed to have only m out of $N = 1024$ nonzero elements), the value of J_m can be used to measure the transformation efficiency for those blocks. In the test, the average transformation error is represented as $J_{m,ave}$.

Fig. 8 depicts the transformation error $J_{m,ave}$ of different b_1 when b_0 is fixed to 1. When b_1 is equal to b_0 , all residuals obey the same Laplacian distribution $f(x|0,1)$. Therefore, all types of transforms get similar performance in this case. If the difference between b_1 and b_0 is larger, 8×32 transform shows better compaction compared to other transforms. In other words, if the residual in two prediction blocks have more different characteristics, 8×32 transform will perform better. Two 16×32 and four 16×16 transform blocks get similar performance in this test model, because the residuals along vertical direction are homogeneous for these two transforms. A similar analysis can be performed on an image block containing horizontal scenarios.

C. Nonsquare Quadtree Transform

From the analysis in the previous subsection, 8×32 (or 32×8) nonsquare transforms are suitable for the residual block that contains directional object boundaries or textures. The size of these nonsquare transforms can be extended according to the size of the prediction blocks. On the other hand, one larger transform block will be more efficient for homogenous residual signal. Therefore, both the square and the nonsquare transforms are advisable to be considered for the transformation process in the region with directional structure. A novel transform quadtree structure that exploits both square and nonsquare transform properties is provided in this paper. The structure is named as NSQT, which will be described in the following paragraphs.

To avoid the introduction of additional transform cores, a transform block is split into four equal nonsquare blocks in NSQT structure. That is a $2N \times 2N$ block will be divided into four $2N \times 0.5N$ or four $0.5N \times 2N$ subblocks in which the block size is represented by width \times height format. Compared to $1.5N \times 2N$ or $2N \times 1.5N$ transforms, $2N \times 0.5N$ and $0.5N \times 2N$ transforms in NSQT can reuse the transform cores in square transform design, while new transform cores such as 12-point and 24-point transform need to be designed

TABLE II
TRANSFORM BLOCK SIZES IN NSQT

Coding Block Size	Horizontal Partition Modes ($2N \times N$, $2N \times nD$ and $2N \times nU$)	Vertical Partition Modes ($N \times 2N$, $nR \times 2N$ and $nL \times 2N$)
64×64	$32 \times 32 \rightarrow 32 \times 8$	$32 \times 32 \rightarrow 8 \times 32$
32×32	$32 \times 32 \rightarrow 32 \times 8 \rightarrow 16 \times 4$	$32 \times 32 \rightarrow 8 \times 32 \rightarrow 4 \times 16$
16×16	$16 \times 16 \rightarrow 16 \times 4 \rightarrow 4 \times 4$	$16 \times 16 \rightarrow 4 \times 16 \rightarrow 4 \times 4$
8×8	$8 \times 8 \rightarrow 4 \times 4$	$8 \times 8 \rightarrow 4 \times 4$

for $1.5N \times 2N$ and $2N \times 1.5N$ transforms. Different from $N \times 2N$ and $2N \times N$ transforms used in the previous standards, NSQT still preserves the quadtree decomposition whose superiorities will be further analyzed in the last part of this section.

Because the nonsquare transform with horizontal or vertical shape is beneficial for coding a region with horizontal or vertical characteristics, those two kinds of nonsquare transforms are used in the NSQT structure. Assuming that the coding block to be transformed is with $2N \times 2N$ block size. At the root of NSQT, $2N \times 2N$ transform is performed based on the consideration that one square transform can get better transformation efficiency and save indicating bits when the residual distribution is homogenous. At subdivision levels, nonsquare transforms are used. Considering the relationship between residual signal correlation and partition modes, the shape of nonsquare transform is bound up with the partition modes. For prediction blocks with size of $2N \times N$, $2N \times nD$ or $2N \times nU$, the $2N \times 2N$ transform block will be split into four $2N \times 0.5N$ transform blocks. In the other case, $0.5N \times 2N$ shaped transforms are performed on $N \times 2N$, $nR \times 2N$ and $nL \times 2N$ partition blocks. A transform block can be separated unless the subdivided transform is unavailable.

Fig. 9 illustrates a splitting process of a three-level NSQT for $2N \times N$, $2N \times nD$ and $2N \times nU$ partition modes. A $2N \times 2N$ block at the root of NSQT (corresponding to level 0) will be split into four $2N \times 0.5N$ blocks located at level 1, and the block at level 1 can be further split into four $N \times 0.25N$ blocks located at level 2.

In the HEVC common condition, transform size can be variable from 32×32 to 4×4 , and the maximum depth of the transform quadtree is 3. Since the maximum transform block size and the minimum transform block size is 32×32 and 4×4 , respectively, NSQT contains four nonsquare block sizes, including 32×8 , 8×32 , 16×4 and 4×16 . The transform block sizes at each subdivision level of NSQT for all coding block sizes are listed in Table II. Since 2-point transform is not used in HEVC, the 4×16 and 16×4 are proposed to be separated into four 4×4 blocks instead of 2×8 blocks in NSQT. The complexity of 2×8 transform was discussed in [34] and [35], and it was pointed out that the hardware cost of 2×8 block is higher than 4×4 block for the same throughput at the encoder side. Moreover, the gate count of 2×8 transform is almost twice that of 4×4 transform [35].

NSQT efficiently utilizes the properties of the image block. A larger transform at the first level will be efficient for smooth residuals or for areas with coarse quantization, whereas a smaller nonsquare transform is more adaptable for the block with directional scenarios. The inherent selection method

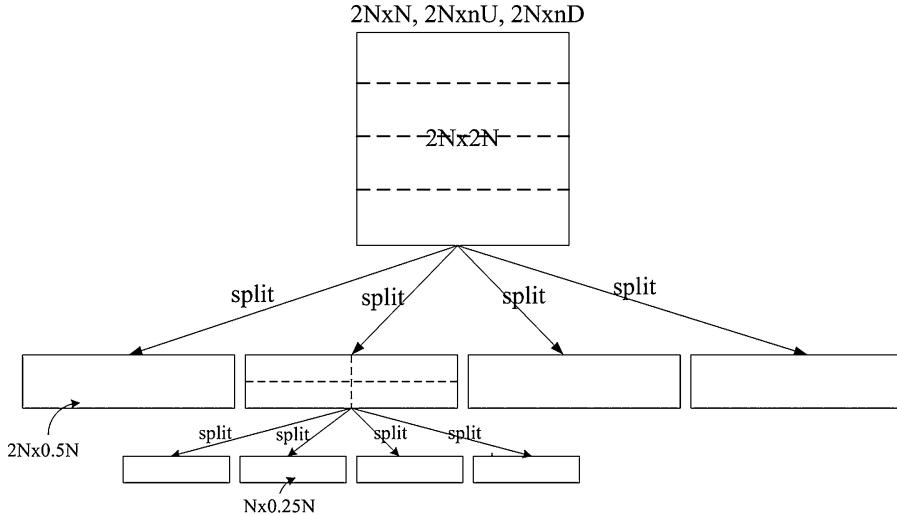


Fig. 9. Three-level NSQT structure for horizontal partitioned nonsquare prediction blocks. The dashed lines indicate the partitioning strategy from the Dth depth to the D+1th in a transform quadtree.

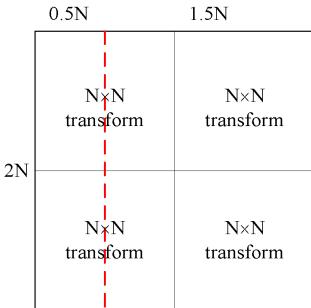


Fig. 10. $nL \times 2N$ partition mode transformed by four $N \times N$ blocks.

in nonsquare quadtree can choose the best shape and size from square and nonsquare transforms by RD criterion as specified in [14]. Besides, NSQT resolves a problem of square transform quadtree, whose split mechanism might cause that a transform block crosses the boundary of the prediction block. An example is described in Fig. 10. In this case, the $nL \times 2N$ partitions mode is used and the residuals are transformed by four $N \times N$ blocks. The $N \times N$ residual blocks on the left side will contain different motion-compensated signals, and the residuals may change around the prediction boundary and has the negative impact on the transformation efficiency. Using $0.5N \times 2N$ transform in NSQT can solve this problem.

Besides NSQT, some other tools are also proposed to improve the coding performance at the area with local nonstationary properties during HEVC standardization. Transform skipping [36] introduces three new transform modes: 1-D horizontal, 1-D vertical and no transform, among which 1-D transform that can deal with the sparser signals or the signals with different properties in horizontal or vertical direction. Compared to NSQT design, transform skipping requires additional logic for switching the transform and the scaling of dequantization which will complicate the implementation of the encoder and decoder. Reference [37] proposes to use $2N \times N$, $N \times 2N$ and $2N \times 2N$ transforms at the root level of RQT. For instance, $2N \times N$ and $2N \times 2N$ transforms are preformed at the

root level for $2N \times N$ partition mode, and the one with less R-D cost will be selected. The transform block sizes at other levels are the same as RQT. Since square and nonsquare transforms are both used at the root level, this design more or less complicates the RQT split strategy. Moreover, $2N \times N$ and $N \times 2N$ transforms are not suitable for AMP due to the above-mentioned crossing boundary problem.

V. IMPLEMENTATION DETAILS

A. Nonsquare Hadamard Transform for Motion Estimation

In the process of motion estimation, Hadamard transform is used to estimate the transformation error for the motion estimation as its implementation cost is much lower than that of DCT-like transform. Define $R_{N \times M}$ as a $N \times M$ residual matrix derived by motion estimation, and H_N and H_M are N -point and M -point Hadamard transform matrix. Then, the transformed block $H_{N \times M}$ can be obtained as follows:

$$H_{N \times M} = H_M \times R_{N \times M} \times H_N^T. \quad (3)$$

Then, the transform error of $R_{N \times M}$ can be calculated as follows:

$$D_{N \times M} = \sum_{j=0}^{M-1} \sum_{i=0}^{N-1} |H_{N \times M}(i, j)|. \quad (4)$$

For square transform blocks, 8×8 or 4×4 Hadamard transform is used to estimate the transformation error in HEVC. Similar to nonsquare transforms in NSQT, the nonsquare concept can also be considered in Hadamard transform design. Considering that the motion estimation process is performed before the transformation process, the actual width and height of the transform block cannot be obtained at this stage. To solve this chicken and egg dilemma, a preanalysis was performed on NSQT cases. From the statistical results, the second subdivision level of NSQT is used with most probability, which means that nonsquare transform are chosen in most

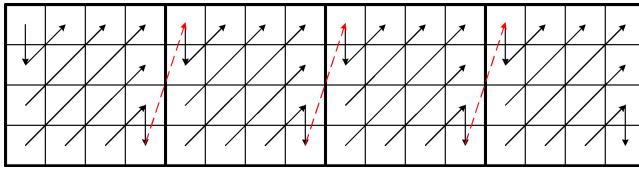


Fig. 11. Coefficients scanning order for a 16×4 block in entropy coding.

cases. Consequently, nonsquare Hadamard transform is used for nonsquare prediction blocks.

Suppose a block, by the width of N and height of M , is performed by motion estimation. When N is greater than M , the transformation error is calculated as follows [38]:

$$D_{HAD} = \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} |\mathbf{H}_{16 \times 4}(i, j)| \quad (5)$$

where $n = N/16$, $m = M/4$.

When $N < M$, the transformation error is calculated as follows [38]:

$$D_{HAD} = \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} |\mathbf{H}_{4 \times 16}(i, j)| \quad (6)$$

where $n = N/4$, $m = M/16$.

B. Transform Cores and Quantization

To avoid introducing additional transform cores, the $2N \times 0.5N$ and $0.5N \times 2N$ transform can be realized by reusing $0.5N$ -point and $2N$ -point transform cores. In general, an $N \times M$ block is transformed by the following steps as described in [22]:

$$\mathbf{C}_{N \times M} = \mathbf{T}_M \times \mathbf{B}_{N \times M} \times \mathbf{T}_N^T \quad (7)$$

where $\mathbf{B}_{N \times M}$ denotes a block with N rows and M columns. \mathbf{T}_N and \mathbf{T}_M are N -point and M -point transform matrix, respectively. $\mathbf{C}_{N \times M}$ denotes the transformed $N \times M$ block.

The quantization process for square blocks in HEVC is implemented as follows [39]:

$$Level = (coeff \times Q + offset) \gg (29 + QP/6 - \log_2^N - B) \quad (8)$$

where Q is a function of QP value, N is the transform block size, and B denotes the intermediate bit-depth. N is calculated as $\sqrt{N_{width}} \times \sqrt{N_{height}}$, which is the only parameter related to transform block size. Since 2-D nonsquare transforms in NSQT use $2N$ -point 1-D transform in one direction and $0.5N$ -point 1-D transform in the other direction, the amplifier factor of $2N \times 0.5N$ or $0.5N \times 2N$ transform is the same as that of $N \times N$ transform. Therefore, the quantization process designed for square transform can be reused for nonsquare transform.

C. Coefficients Scanning for Entropy Coding

In HEVC, the scanning order of transform coefficients follows a 4×4 block based diagonal scanning order for facilitating hardware implementation [40], [41]. The transform block is first divided into 4×4 subblocks, and diagonal scan is performed inside 4×4 subblocks and is also used among

TABLE III
PERCENTAGES OF AREA OF SQUARE AND NONSQUARE PREDICTION BLOCKS (LOW DELAY B MAIN, NSQT ON)

Sequence	Square	Nonsquare (SMP+AMP)	SMP	AMP
<i>Kimono</i>	29%	20%	10%	10%
<i>ParkScene</i>	16%	21%	11%	11%
<i>Cactus</i>	20%	18%	9%	9%
<i>BasketballDrive</i>	23%	24%	13%	11%
<i>BQTerrace</i>	23%	17%	10%	7%
<i>BasketballDrill</i>	35%	18%	10%	8%
<i>BQMall</i>	17%	26%	14%	12%
<i>PartyScene</i>	25%	36%	19%	17%
<i>RaceHorses</i>	28%	29%	15%	14%
<i>BasketballPass</i>	21%	24%	13%	11%
<i>BQSquare</i>	24%	32%	17%	15%
<i>BlowingBubbles</i>	21%	32%	16%	16%
<i>RaceHorses</i>	26%	33%	17%	15%
<i>FourPeople</i>	7%	10%	6%	5%
<i>Johnny</i>	6%	9%	6%	4%
<i>KristenAndSara</i>	8%	11%	6%	5%

4×4 subblocks. The same scanning strategy is extended to nonsquare transform blocks [42]. Fig. 11 illustrates the scanning order of a 16×4 block. The 16×4 data array is first scanned and transposed into 4×4 block array. Within each 4×4 block, diagonal scan is used to convert the 2-D signal to 1-D array. Then, the 1-D signal of the 16×4 block is processed with the same entropy coding method as square block, whose context derivation mechanism, and coefficient coding are not changed [42].

VI. STATISTICAL ANALYSIS AND SIMULATION RESULTS

The statistics and simulation results are generated by HM7.0 reference software. The anchor data provided in this paper is generated by HM7.0 with disabling AMP and NSQT.

A. Statistical Analysis

As discussed in Section III, HEVC framework contains three prediction modes: skip, intra, and inter. In inter prediction mode, the partition modes can be classified into square motion partitions ($2N \times 2N$ and $N \times N$) and nonsquare motion partitions ($2N \times N$, $N \times 2N$ and AMP). To inspect the efficiency of the square and the nonsquare motion partitions, the percentages of the area of these two categories are stated in Table III. Since the nonsquare category can be further split into SMP that include $2N \times N$ and $N \times 2N$ sizes, and AMP subcategories, the data of SMP and AMP are also listed in Table III. The data listed in nonsquare (SMP+AMP) column is the summation of data in AMP and SMP columns. It should be noted that the data of skip and intra modes are not listed in Table III. All sequences in common test conditions in HEVC [43] are tested when NSQT is switched on and the data in Table III is the average value of four QPs: 22, 27, 32, and 37. The low delay B main configuration is used in all the tests in this subsection. It can be observed that the percentages of nonsquare prediction blocks are almost the same as or

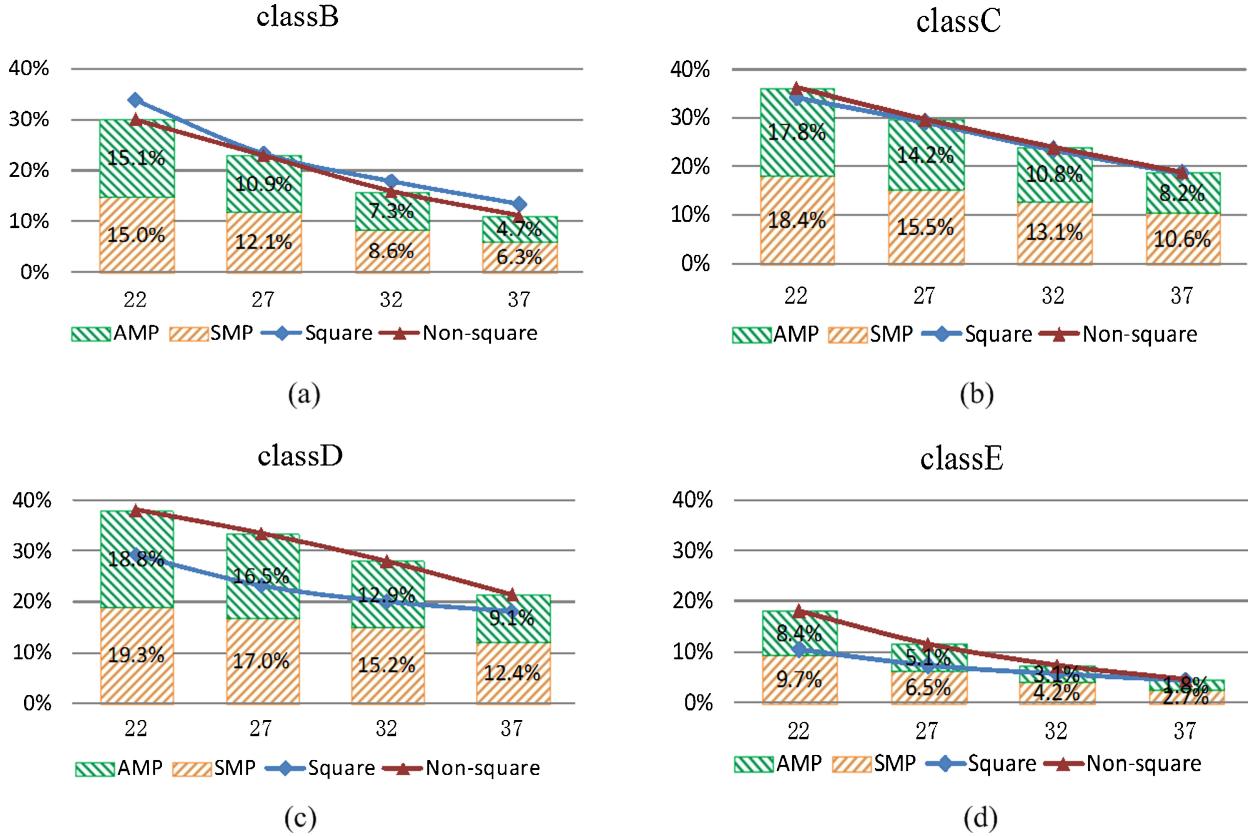


Fig. 12. Percentages of area of square and nonsquare prediction blocks. The proportion of SMP and AMP are also provided separately. (a)–(d) Average values in each class defined in [43]. SMP: symmetric motion partitions, including $2N \times N$ and $N \times 2N$ partition modes.

larger than those of the square prediction blocks in most cases. Within nonsquare motion partitions, the percentage of AMP is very close to that of SMP, which reflects that AMP is a good complement to symmetric motion partitions. Fig. 12(a)–(d) shows the average data for each class [43] of different QP values. The difference between the nonsquare and the square blocks decreases along with the increase of QP value, and this is because the majority of coding blocks use skip mode when coarse quantization is performed.

To explore the interaction between nonsquare motion partitions and NSQT, the variation of nonsquare motion partitions percentage conditioned on NSQT is shown in Fig. 13(a)–(d). It is observed that the percentages of nonsquare motion partitions are all increased when NSQT is switched on. Those results show the benefit for the combination of nonsquare motion partitions and NSQT. Since NSQT can improve the transformation compaction efficiency of nonsquare motion partitions, the prediction accuracy of nonsquare motion partitions is improved by using NSQT. It is also noticed that the interaction effect between nonsquare partitions and NSQT decreases at high QP range. This phenomenon is due to the larger usage of skip mode at lower bit-rate applications.

From the statistical analysis results, it can be concluded that the quadtree based nonsquare motion partitions and NSQT are benefit for each other, and this combination plays an important role in motion estimation and compensation process, particularly in the case of high bit-rate coding, or high quality video coding.

B. R-D Performance

The R-D performance of the proposed AMP, NSQT, and the combination of AMP and NSQT are shown in Table IV. The test condition is identical to the common test condition in HEVC, as specified in [43]. All test sequences are classified into classA to classE in terms of different resolutions and applications. The coding efficiency is measured by BD rate [44], and the encoding and decoding time increases are also provided as one of the complexity measurement metric. A fast mode decision strategy for AMP is used in this test, whose detailed description can be found in [45]. The experimental results show that AMP contributes 0.78% bit-rate saving in random access (average of main and HE10) and 1.16% in low delay B, respectively. NSQT achieves 0.25% (random access) and 0.71% (low delay B) performance improvement. The combination of AMP and NSQT reveals an average of 1.18% bit-rate saving in random access and 2.03% in low delay B.

About a 17% encoding time increase is produced with nearly no increase of decoding time. The implementation cost mainly comes from the R-D optimization process of newly introduced four asymmetric motion partition modes at the encoder side. The decoding time will not increase because the decoding process of the asymmetric prediction block is the same as that of symmetric partitions. Since NSQT is compatible with the quadtree structure and retains the one-pass coding strategy, there is hardly any increase of encoding and decoding time. The implementation cost for NSQT is mainly related to the calculation of the address for nonsquare

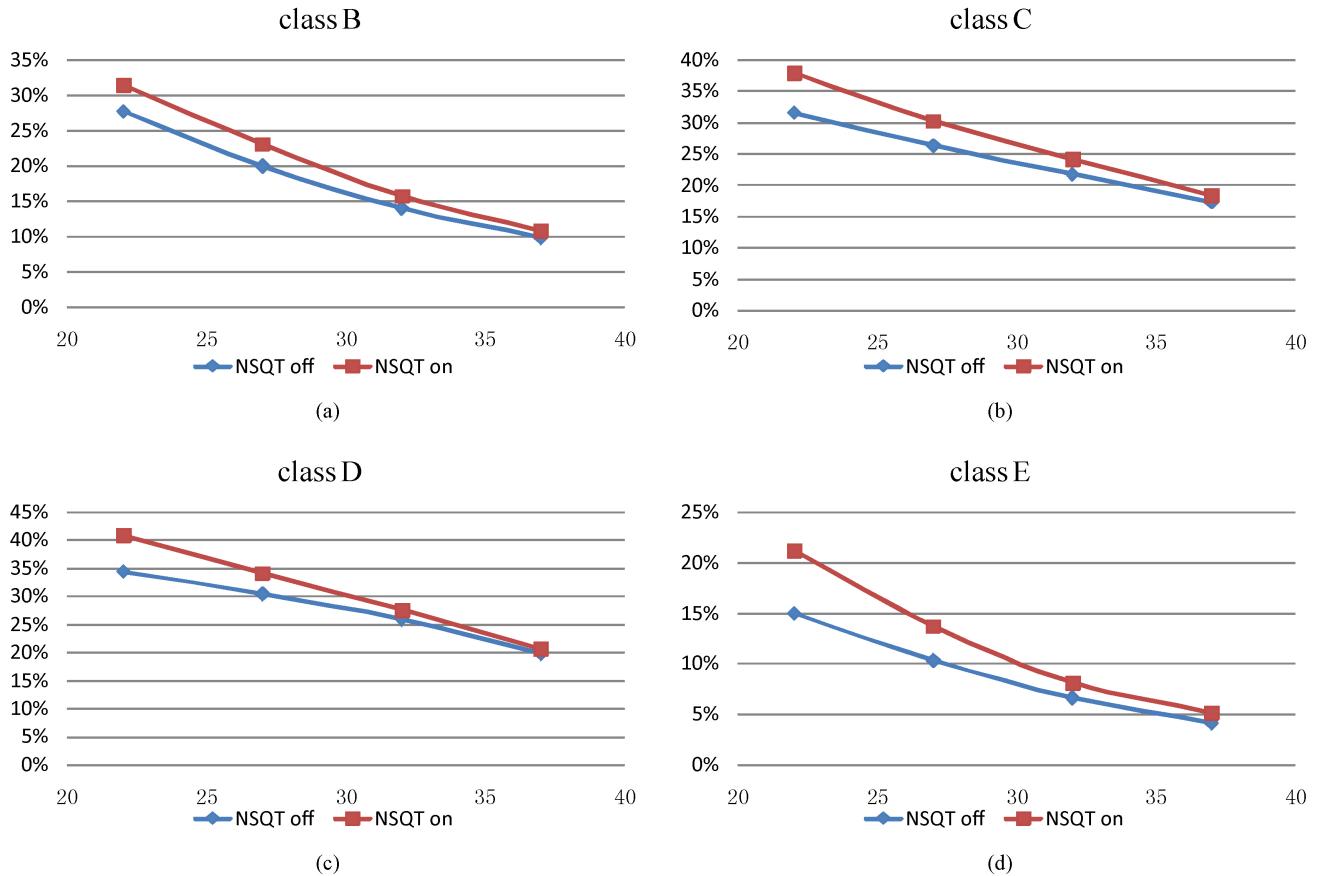


Fig. 13. Comparison of percentages of nonsquare prediction blocks with and without NSQT. (a)–(d) Average values in each class defined in [43].

TABLE IV
CODING PERFORMANCE OF AMP AND NSQT

	Random Access Main			Random Access HE10			Low Delay B Main			Low Delay B HE10		
	AMP	NSQT	AMP+NSQT	AMP	NSQT	AMP+NSQT	AMP	NSQT	AMP+NSQT	AMP	NSQT	AMP+NSQT
ClassA	-0.58%	-0.20%	-0.84%	-0.61%	-0.17%	-0.96%						
ClassB	-0.66%	-0.33%	-1.14%	-0.71%	-0.30%	-1.09%	-0.9%	-0.79%	-1.88%	-0.92%	-0.69%	-1.75%
ClassC	-0.95%	-0.29%	-1.43%	-1.02%	-0.26%	-1.43%	-1.02%	-0.67%	-1.93%	-1.02%	-0.52%	-1.79%
ClassD	-0.89%	-0.23%	-1.28%	-0.95%	-0.24%	-1.31%	-1.14%	-0.59%	-1.92%	-1.26%	-0.54%	-1.98%
ClassE							-1.42%	-1.09%	-2.73%	-1.53%	-0.95%	-2.78%
Average	-0.76%	-0.26%	-1.17%	-0.81%	-0.25%	-1.19%	-1.09%	-0.77%	-2.06%	-1.14%	-0.66%	-2.01%
Encoding Time	114%	102%	117%	114%	102%	116%	114%	101%	117%	114%	102%	117%
Decoding Time	101%	100%	101%	101%	100%	101%	100%	99%	101%	101%	99%	101%

transform blocks that adds horizontal and vertical scans in addition to the Z-order scan [46]. With respect to hardware implementation, the designs for square block and nonsquare block are almost the same.

C. Subjective Quality

Fig. 14 shows the subjective quality comparison for the 193th frame of *BasketballPass_416 × 240* sequence at QP 32. Three parts of the sequence are zoomed-in on to make the subjective improvement more visible. As can be seen, the ringing artifacts produced by large transform block can be weakened by comparing Fig. 14(a) with (b). Furthermore, the image encoded with AMP and NSQT can preserve more detailed lines, as illustrated in Fig. 14(d)–(f).

The coding bits for the cases with AMP and NSQT are less than the cases without these two tools, while both PSNR and subjective quality have been improved. It can be concluded that AMP and NSQT not only promote R-D performance, but also improve subjective quality.

VII. CONCLUSION

HEVC uses quadtree structure as the basic framework for the separation of the coding and transform block. Based on previous state-of-the-art works, a nonsquare quadtree framework was provided and discussed in this paper, which incorporates the nonsquare concept into the quadtree structure. The proposed nonsquare quadtree includes a nonsquare quadtree-based AMP and a NSQT.



Fig. 14. Subjective quality comparison. Three parts of the 193th frame of the *BasketballPass* sequence at QP = 32. (a), (c), and (e) Encoded without AMP and NSQT (1656 bits, Y 29.13 dB, U 36.39 dB, V 34.94 dB). (b), (d), and (f) Encoded with AMP and NSQT (1544 bits, Y 29.1759 dB, U 36.5951 dB, V 34.9917 dB).

The nonsquare asymmetric quadtree motion partitions can provide a more variable prediction block size with minimum signaling overhead and low encoder complexity. Unlike traditional symmetric motion partitions, AMP is a new proposed partitioning concept that can better match the irregular object's boundary and the region with directional characteristics.

NSQT provides a more efficient transform structure that exploits the superiorities of square transforms and nonsquare transforms in one unified structure. It preserves the flexibility of the quadtree structure and improves the capability of energy compaction of the transform. NSQT implicitly derives the size and shape from the size of prediction block, and thus does not bring extra encoding runtime. NSQT also has the interoperability combining with nonsquare motion partitions. Statistical results showed that the ratio of nonsquare motion partitions increased, which enhances the efficiency of the nonsquare motion partition structure.

The asymmetric quadtree nonsquare motion partition and nonsquare quadtree transform, described in this paper, construct a comprehensive nonsquare quadtree structure that considers the characteristics of nonsquare prediction block and

nonsquare transform block. Simulation results revealed a good tradeoff between bit-rate reduction and encoding complexity. Subjective quality improvement was also observed, especially for the region with horizontal or vertical texture.

REFERENCES

- [1] Cliff Reader, *History of MPEG Video Compression*, document JVT-E066, Geneva, Switzerland Oct. 2002.
- [2] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), version 8, ITU-T and ISO/IEC, Jul. 2007.
- [3] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, *High-Efficiency Video Coding (HEVC) Text Specification Draft 8*, document JCTVC-J1003, 10th Meeting, Stockholm, Sweden, Jul. 2012.
- [4] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
- [5] S. Gordon, D. Marpe, and T. Wiegand, *Simplified Use of 8 × 8 Transforms—Updated Proposal and Results*, document VCEG-K028, 11th Meeting, Munich, Germany, Mar. 2004.
- [6] T. Wiegand, J.-R. Ohm, G. J. Sullivan, W.-J. Han, R. Joshi, T. K. Tan, and K. Ugur, "Special section on the joint call for proposals on high-efficiency video coding (HEVC) standardization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1661–1666, Dec. 2010.

- [7] W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, A. Alshin, T. Lee *et al.*, *Samsung's Response to the Call for Proposals on Video Compression Technology*, document JCTVC-A124, Dresden, Germany, Apr. 2010.
- [8] B. Gross, A. Fuldseth, X. Wang, and W.-J. Han, *BoG Report: Residual Quadtree Structure*, document JCTVC-C319, Guangzhou, China, Oct. 2011.
- [9] T. Wiegand, *Description of Video Coding Technology Proposal by Fraunhofer HHI*, document JCTVC-A116, Dresden, Germany, Apr. 2010.
- [10] *Joint Call for Proposals on Video Compression Technology*, ITU-T SG16 Q6 document VCEG-AM91 and ISO/IEC JTC1/SC29/WG11 document N11113, ITU-T SG16 Q6 and ISO/IEC JTC1/SC29/WG11, Kyoto, Japan, Jan. 2010.
- [11] W.-J. Han, J. Min, I.-K. Kim, E. Alshina, A. Alshin, T. Lee *et al.*, "Improved video compression efficiency through flexible unit representation and corresponding extension of coding tools," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1709–1720, Dec. 2010.
- [12] Y. Yuan, X. Zheng, X. Peng, J. Xu, Y. Wang, X. Cao *et al.*, "CE2: non-square quadtree transform for symmetric motion partitions," document JCTVC-F410, 6th Meeting, Turin, Italy, 14–22 Jul. 2011.
- [13] Y. Yuan, X. Zheng, X. Peng, J. Xu, I.-K. Kim, L. Liu *et al.*, "CE2: nonsquare Quadtree Transform for Symmetric and Asymmetric Motion Partition," document JCTVC-F412, 6th Meeting, Torino, Italy, Jul. 2011.
- [14] G. J. Sullivan and T. Weigand, "Rate-distortion optimization for video compression," *IEEE Process. Mag.*, vol. 15, no. 6, pp. 74–90, Nov. 1998.
- [15] Y. Wang, X. Mao, and Yun He, "A dual quad-tree based variable block-size coding method," *J. Visual Commun. Image Representation*, vol. 21, no. 8, pp. 889–899, Nov. 2010.
- [16] M. Winken, S. Boße, B. Gross, P. Helle, T. Hinz, H. Kirchhoffer *et al.*, *Description of Video Coding Technology Proposal by Fraunhofer HHI*, document JCTVC-A116, Dresden, Germany, Apr. 2010.
- [17] S. Kondo and H. Sasai, "A motion compensation technique using sliced blocks in hybrid video coding," in *Proc. IEEE ICIP*, Sep. 2005, pp. 305–308.
- [18] X. Zheng and H. Yu, "CE2: Huawei and Hisilicon Report on Flexible Motion Partitioning," document JCTVC-D297, 4th Meeting, Daegu, Korea, Jan. 2011.
- [19] E. Francois, P. Bordes, L. Guo, and M. Karczewicz, "CE2: Simplified Geometry Block Partitioning," document JCTVC-D230, 4rd Meeting, Daegu, Korea, Jan. 2011.
- [20] L. Guo, P. Yin, and E. Francois, "TE 3: Simplified Geometry Block Partitioning," document JCTVC-B085, 2nd Meeting, Geneva, Switzerland, Jul. 2010.
- [21] I. H. Dinstein, K. Rose, and A. Heiman, "Variable block-size transform image coder," *IEEE Trans. Commun.*, vol. 38, no. 11, pp. 2073–2078, Nov. 1990.
- [22] M. Wien and A. Dahlhoff, *Adaptive Block Transforms*, document VCEG-M62, 13th Meeting, Austin, TX, Apr. 2001.
- [23] M. Wien and J.-R. Ohm, "Simplified Adaptive Block Transforms," document VCEG-O30, 15th Meeting, Pattaya, Thailand, Dec. 2001.
- [24] SMPTE, *Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process*, SMPTE 421M-2006.
- [25] C. Zhang, K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, "Video coding using spatially varying transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 127–140, Feb. 2011.
- [26] S. Dubuisson and F. Davoine, "Motion compensation using adaptive rectangular partitions," in *Proc. Int. Conf. Image Process.*, 1999, pp. 56–60.
- [27] Q. Yao, Z. Wang, H. Bian, and M. Xu, *Fundamentals in Image Coding*. Zhejiang, China: Zhejiang Univ. Press, 1992.
- [28] F. Kamisl and J.-S. Lim, "Transforms for the motion compensation residual," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2009, pp. 789–792.
- [29] C. Xiong and Y. He, "Forward rate control strategy based on the rate distortion theory for video coding," in *Proc. SPIE VCIP*, vol. 3024. Feb. 1997, pp. 1441–1448.
- [30] I.-M. Pao and M.-T. Sun, "Modeling DCT coefficients for fast video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 608–616, Jun. 1999.
- [31] A. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 4, pp. 356–365, Oct. 1979.
- [32] W.-K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *Proc. IEEE*, vol. 136, no. 4, pp. 276–282, Aug. 1989.
- [33] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
- [34] W. Gao, "CE6 BoG Report on SDIP Throughput," document JCTVC-F755, 6th Meeting, Turin, Italy, Jul. 2011.
- [35] G. Sullivan and J.-R. Ohm, "Report Document From Chairs of JCT-VC," document JCTVC-F800, 6th Meeting, Turin, Italy, Jul. 2011.
- [36] M. Mrak, A. Gabriellini, N. Sprijan, and D. Flynn, "Transform Skip Mode," document JCTVC-F077, 6th Meeting, Turin, Italy, Jul. 2011.
- [37] L. Guo, J. Sole, R. Joshi, P. Chen, X. Wang, and M. Karczewicz, "Non-square Transform for $2N \times N$ and $N \times 2N$ Motion Partitions," document JCTVC-F563, 6th Meeting, Turin, Italy, Jul. 2011.
- [38] X. Zheng, L. Liu, Y. Yuan, and Y. He, "Nonsquare Hadamard Transform for Motion Estimation and Merge Estimation," document JCTVC-G521, 7th Meeting, Geneva, Switzerland, Nov. 2011.
- [39] A. Fuldseth, G. Bjøntegaard, M. Budagavi, and V. Sze, "CE10: Core Transform Design for HEVC," document JCTVC-G495, 7th Meeting, Geneva, Switzerland, Nov. 2011.
- [40] J. Sole, R. Joshi, and M. Karczewicz, "Non-CE11: Diagonal Sub-Block Scan for HE Residual Coding," document JVTVC-G323, 7th Meeting, Geneva, Switzerland, Nov. 2011.
- [41] N. Nguyen, T. Ji, and D. He, "Multi-Level Significance Maps for Small Transform Units," document JCTVC-H0526, 8th Meeting, San Jose, CA, Feb. 2012.
- [42] J. Sole, "JCTVC-Break-Out Report: Harmonization of NSQT With Residual Coding," document JCTVC-G1038, 7th Meeting, Geneva, Switzerland, Nov. 2011.
- [43] F. Bossen, "Common Test Conditions and Software Reference Configurations," document JCTVC-I1100, 9th Meeting, Geneva, Switzerland, 27 Apr.–May 2012.
- [44] G. Bjøntegaard, "Calculation of Average PSNR Differences Between RD-Curves," document VCEG-M33, Q.6/SG16 (VCEG), 13th Meeting, Austin, TX, Apr. 2001.
- [45] I.-K. Kim, W.-J. Han, J. Park, and X. Zheng, "CE2: Test Results of asymmetric motion partition (AMP)," document JCTVC-F379, 6th Meeting, Turin, Italy, Jul. 2011.
- [46] G. M. Morton, "A computer oriented geodetic data base; and a new technique in file sequencing," Tech. Rep., IBM Ltd., Ottawa, ON, Canada, 1966.



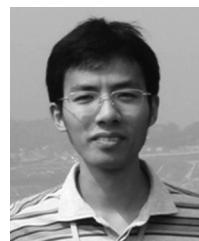
Yuan Yuan received the B.S. degree in electronic engineering from the Beijing Institute of Technology, Beijing, China, in 2008. She is currently pursuing the Ph.D. degree with the Department of Electronic Engineering, Tsinghua University, Beijing.

She participates in the Joint Collaborative Team on Video Coding and has contributed to the next generation of international video coding standard, HEVC. Her current research interests include High Efficiency Video Coding, video communication, and image processing.



Il-Koo Kim received the M.S. and Ph.D. degrees from the School of Electrical Engineering, Seoul University, Seoul, Korea, in 2001 and 2006, respectively.

After joining the Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea, in 2006, he has worked to develop high-efficiency video codecs. His current research interests include video compression and transmission.



Xiaozhen Zheng received the B.E. degree from the Harbin University of Science and Technology, Harbin, China, in 2003, and the Masters by Research degree from the University of York, York, U.K., in 2006.

He is currently a Senior Research Engineer with the Research Department, Hisilicon Semiconductor and Component Business Department, Huawei Technologies, Shenzhen, China. His current research interests include video compression and communication, image processing, and computer vision.



Lingzhi Liu (M'07–SM'09) received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1998, and the Ph.D. degree from Shanghai Jiaotong University, Shanghai, China, in 2005.

He is currently a Graphics Media Architect with the Intel Corporation, Santa Clara, CA. From 2008 to 2012, he was a Senior Research Engineer with Huawei Technologies, Santa Clara, CA. He was a Post-Doctoral Research Associate with the Electrical Engineering Department, University of Washington, Seattle, from 2005 to 2008. He was engaged in research on video coding algorithms, video codec chip architecture, digital application-specific integrated circuits, and field-programmable gate arrays. His general interests include multimedia coding algorithms and implementation, very large scale integration (VLSI) system design, and channel coding theory. His current research interests include High Efficiency Video Coding algorithm development, VLSI design for video technology, and VLSI design for low-power and high-performance communication systems.

Dr. Liu has served on the Technical Program Committees of several international conferences.



Xiaoran Cao received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2009, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering.

Since 2010, he has been an active participant in ITU-T VCEG and ISO/IEC MPEG, contributing to their joint project, Joint Collaborative Team on Video Coding. His current research interests include video coding, visual signal processing, and image processing.



Sunil Lee (S'02–CM'08) received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2002 and 2008, respectively.

From March 2008 to August 2008, he was a Post-Doctoral Researcher with the Division of Electrical Engineering, School of Electrical Engineering and Computer Science, KAIST. He is currently a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. His current research interests include multimedia retrieval, multimedia security, digital watermarking, multirate signal processing, and video coding.



Min-Su Cheon received the B.S. and M.S. degrees in electrical engineering from Kyunghee University, Suwon, Korea, in 2005 and 2007, respectively.

He is currently an Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon. His current research interests include high-efficiency video coding, multiview video coding, 3-D video coding, scalable video coding, and bioimage processing.



Tammy Lee received the B.S. degree in mathematics education from Ewha Women's University, Seoul, Korea, in 2000, and the M.S. degree in computer science from New York University, New York, in 2005.

She is currently a Senior Engineer with the Multimedia Platform Laboratory, Digital Media and Communications Research and Development Center, Samsung Electronics, Suwon, Korea. Her current research interests include image and video compression.



Yun He (M'97–SM'01) received the B.S. degree in signal processing from the Harbin Shipbuilding Institute, Harbin, China, in 1982, the M.S. degree in ultrasonic signal processing from Shanghai Jiaotong University, Shanghai, China, in 1984, and the Ph.D. degree in image processing from Liege University, Liege, Belgium, in 1989.

She is currently a Professor with the Department of Electronic Engineering, Tsinghua University, Beijing, China. Her current research interests include picture coding theory and methodology, picture cod and hardware complexity analysis, video codec very (VLSI) structure, and multiview and 3-D picture coding.

Dr. He was a member of the Technical Committee of the Multi-Media Signal Processing in IEEE SP Society from 2005 to 2008, and Chair of the Technical Committee of Visual Signal Processing and Communications of the IEEE CAS Society from 2008 to 2010. She is a member of the Technical Committee of the Image, Video, and Multidimensional Signal Processing in the IEEE SP Society. She was a General Co-Chair for the 25th Picture Coding Symposium in 2006. She is an Associate Editor of the *Journal of Visual Communication and Image Representation*, and the *Journal of VLSI Signal Processing Systems*. She is a Guest Co-Editor for the special issue on multiview video coding of the *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY* in 2007 and for the special issue on hardware implementation of video codec of the *Journal of VLSI Signal Processing Systems*, in 2007.



Jeong-Hoon Park received the B.S. and M.S. degrees in electrical engineering from Hanyang University, Seoul, Korea, in 1991 and 1994, respectively.

He is currently a Principal Engineer with the Multimedia Platform Laboratory, Digital Media and Communication Research and Development Center, Samsung Electronics, Suwon, Korea. In 1998, he was a Visiting Scholar with the University of California, Los Angeles, where he researched error resilient video compression and transmission. His current research interests include audio and video compression technology, medical image processing, and multimedia systems, including video broadcasting and communications.

Mr. Park has been contributing to the ITU-T Video Coding Experts Group and the ISO/IEC Moving Pictures Experts Group since 1998. From 2002 to 2004, he was involved in developing mobile broadcasting technology as an active member of the WorldDMB forum.