IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 18, 2019, accepted October 27, 2019, date of publication October 30, 2019, date of current version November 11, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2950388*

# Fast Affine Motion Estimation for Versatile Video Coding (VVC) Encoding

**SANG-HYO PARK, (Member, IEEE), AND JE-WON KANG, (Member, IEEE)**
Department of Electronics and Electrical Engineering, Ewha Womans University, Seoul 03760, South Korea

Corresponding author: Je-Won Kang (sagittak@gmail.com)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korean Government (MSIT) under Grant 2018-0-00765, Development of Compression and Transmission Technologies for Ultra High Quality Immersive Videos Supporting 6DoF.

**ABSTRACT** In this paper, we propose a fast encoding method to facilitate an affine motion estimation (AME) process in versatile video coding (VVC) encoders. The recently-launched VVC project for next-generation video coding standardization far outperforms the High Efficiency Video Coding (HEVC) standard in terms of coding efficiency. The first version of the VVC test model (VTM) displays superior coding efficiency yet requires higher encoding complexity due to advanced inter-prediction techniques of the multi-type tree (MTT) structure. In particular, an AME technique in VVC is designed to reduce temporal redundancies (other than translational motion) in dynamic motions, thus achieving more accurate motion prediction. The VTM encoder, however, requires considerable computational complexity because of the AME process in the recursive MTT partitioning. In this paper, we introduce useful features that reflect the statistical characteristics of MTT and AME and propose a method that employs these features to skip redundant AME processes. Experimental results show that—when compared to VTM 3.0—the proposed method reduces the AME time of VTM to 63% on average, while the coding loss is within 0.1% in the random-access configuration.

**INDEX TERMS** Video compression, encoding complexity, motion estimation, HEVC, VVC, affine motion, reference frame search.

## I. INTRODUCTION

The amount of video data has increased rapidly, especially with the growing use of Internet-based streaming services and devices that receive video broadcasts. The bandwidth and storage capacity of video applications is limited, requiring efficient video compression techniques. This need for video compression will further increase due to the higher resolutions of volumetric content such as 360-degree and high dynamic range (HDR) videos. Considering this diverse and growing demand for more powerful compression, a new video coding standardization project called *versatile video coding* (VVC) was launched recently by the Joint Video Exploration Team (JVET) of two expert groups: ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG). The JVET published the initial draft of VVC in 2018 [1] and released the VVC test model (VTM). VTM has a similar structure to the High Efficiency Video Coding (HEVC) test model (HM), but it uses advanced tools that provide better compression performance.

A key concept among these tools is the multiple-type tree (MTT) segmentation structure [2]. While the HEVC standard can only support a quad-tree (QT) structure to split a block into multiple coding units (CUs), the MTT structure in VVC can have a binary tree (BT) or ternary tree (TT) as an additional sub-tree structure in a QT. Thus, MTT can support more diverse CU block shapes than QT, contributing to more effective coding performance.

The flexibility of MTT, however, leads to high computational complexity in encoding. In the JVET meeting, it is reported that equipping the QT plus BT (QTBT) structure increases by 1.8 times (for the random-access case) the encoding complexity of the joint exploration test model (JEM) that preceded VTM [3]. Several researchers targeted the QTBT structure to reduce the encoding complexity of JEM [4], [5]. It is no surprise that MTT comprising QT, BT, and TT further increases the complexity of video encoders. For example, the software implementation of VTM (version 1.0), which used the MTT structure, was

The associate editor coordinating the review of this manuscript and approving it for publication was Donatella Darsena.

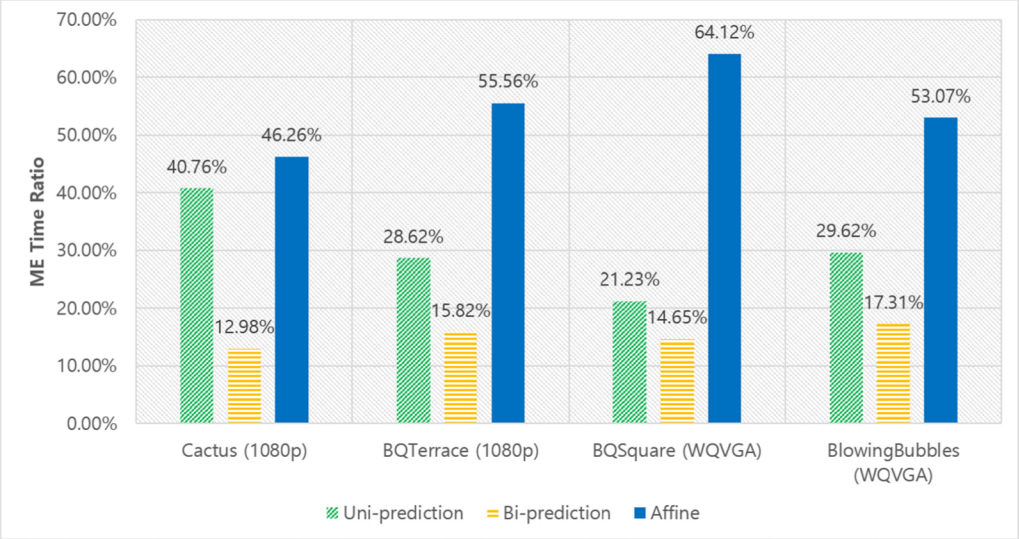



**FIGURE 1.** Time complexity of unidirectional prediction, bi-prediction, and affine prediction of ME in VTM3.0. Each percentage represents the portion of the total time in ME. Quantization parameter is set to 37.

approximately half the speed of the software implementation of HM (version 16.18), even though the VTM implementation turned on SIMD instructions [6]. This high encoding complexity with MTT must be overcome to improve real-time applications and multimedia services, particularly for high battery-drain devices.

Among the tools in each block of MTT, motion estimation (ME) shows the highest encoding complexity in VVC. The computational complexity of ME increases even more than in HEVC due to more advanced inter-prediction schemes, recursively conducted in fine partitioning blocks of MTT. Affine motion estimation (AME) [7], [8], which characterizes non-translational motions such as rotating and zooming, turns out to be efficient in rate-distortion (RD) performance at the expense of high encoding complexity. Reducing the complexity of the VTM encoders thus requires speeding up the AME process and the associated affine motion compensation (AMC).

The AME has a significant portion of computational complexity of the overall ME processing time, and therefore it is important to reduce the complexity. In Fig. 1, we show the computational complexity of unidirectional prediction, bi-prediction, and affine prediction of the ME process when encoding several video sequences with VTM 3.0. Each percentage shows the ratio of the total ME time. We observe that the AME has the significant computational complexity around 54.75% on average. Therefore, we focus on developing the fast AME technique rather than the conventional unidirectional prediction and bi-prediction techniques. In fact, many researchers have attempted to alleviate the complexity of the conventional unidirectional prediction and bi-prediction in previous video coding standards [9]–[13], [24], [25] based on QT-based partitioning structures. However, there are only few works to alleviate the complexity of AME in VVC. For VTM, there is much room to further reduce the ME complexity, particularly in AME, in an MTT structure.

In this paper, we propose a fast encoding method to efficiently reduce the encoding complexity of AME in VTM when MTT is used. The proposed method consists of two procedures. The first procedure eliminates redundant AME and AMC processes, using an early termination scheme by exploiting parent CUs. Specifically, motion information from the parent CU that has been previously encoded in MTT is exploited. The second procedure reduces the number of reference frames used for AME. To the best of our knowledge, these approaches are the first attempts to reduce the AME complexity in the VVC literature. To demonstrate the efficiency of the proposed method, the associated ME time in VTM is measured under a random-access (RA) configuration. Experimental results show that the AME time of the VTM 3.0 is reduced to 64% on average with negligible coding loss.

This paper is organized into five sections. Section II reviews fast ME methods and provides an overview of AME in VTM (from now on, we regard VTM as the VTM 3.0). Section III analyzes the characteristics of the MTT and the affine model and presents the proposed fast encoding method. Section IV reports the experimental results in comparison with VTM. Finally, Section V presents the conclusions.

## II. RELATED WORK

### A. OVERVIEW OF MOTION ESTIMATION IN VVC

Conventional ME (CME) uses a block-matching algorithm with a rectangular block shape to compute a translation motion vector (MV). The HEVC standard adopts a QT structure so that a CU size for ME can vary within a size of 64 x 64 pixels. The CU can be further split with up to eight partitions in the scope of a prediction unit (PU) in a CU [14].

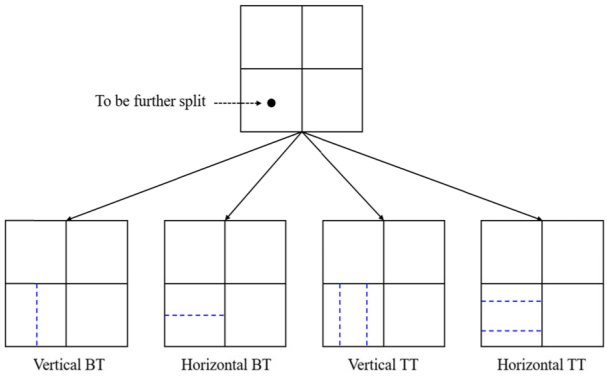



**FIGURE 2.** An example of possible BT/TT splitting (dotted blue line) within a CTU after QT split.

To achieve even greater compression efficiency, the recently-introduced MTT structure [2] enables more flexible partitioning structures for prediction. MTT is a tree structure in which a block can be split into a QT, BT, or TT from the root: a *coding tree unit* (CTU). A CTU can be first split into three tree types, but only the BT and TT types have an interchangeable structure: this implies that the BT can have a sub-BT or sub-TT, as can TT. On the contrary, the QT structure can only be a starting point for a CTU. Accordingly, from the leaf node of a QT, BTs and TTs may be tested. One example of the MTT in a CTU is shown in Fig. 2. A CTU is first split into a QT with four sub-CUs, and subsequently, the third sub-CU is split by either a BT or TT at a horizontal/vertical direction. Furthermore, each BT-partitioned or TT-partitioned sub-CU can be split until the pre-defined maximum depth of each tree structure is reached. These variations could produce the best motion shape to be encoded for improving the compression performance of the VTM.

Compared to block shape flexibility, CME for the prediction of translational motion has not been extensively studied in the JVET community. Rather, AME has been attractive to video coding experts because it enlarges the variety of motions that can be estimated. For CME, the existing video coding standards such as AVC/H.264 and HEVC use a MV that covers translational motion. However, AME enables the prediction of not only translational motion but also linearly transformed motion such as scaling and rotation. If a camera zooms or rotates to capture a video, AME can predict the motion more accurately than translation-based CME. Li *et al.* [7] reported that the coding gain from AME is approximately 10% under the RA case on top of HM for affine test sequences. In the recent JEM implementation, AME provides a meaningful coding gain as well.[8].

To generate the MV for AME, VTM can choose one of two affine models depending on the control point parameters. One is the four-parameter affine model, and the other is the six-parameter model [2]. As shown in Fig. 3, two or three vectors can generate an affine-transformed block. We can denote an affine MV to be predicted as $\boldsymbol{mv}$ in the two-dimensional Cartesian coordinate system. Thus, $\boldsymbol{mv}$ can be represented

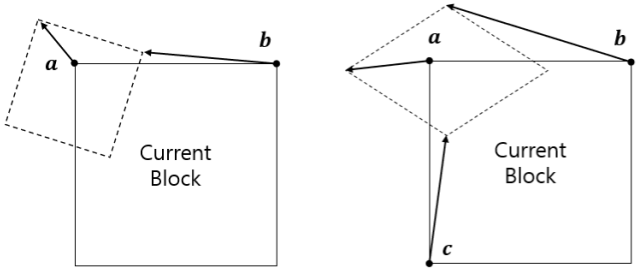



**FIGURE 3.** Examples of the affine model: (a) 4-parameter model and (b) 6-parameter model.

as $(mv^h, mv^v)$ at a sample location $(x, y)$ in a block to be predicted, where $mv^h$ represents a point on the $x$-axis, and $mv^v$ represents a point on the $y$-axis. If we have two vectors $(a^h, a^v)$ at the top-left corner of a block and $(b^h, b^v)$ at the top-right corner of a block, the $\boldsymbol{mv}$ for point $(x, y)$ can be solved by (1):

$$\begin{cases} mv^h = \dfrac{b^h - a^h}{w}x + \dfrac{b^v - a^v}{w}y + a^h \\ mv^v = \dfrac{b^v - a^v}{w}x + \dfrac{b^h - a^h}{w}y + a^v, \end{cases} \tag{1}$$

where $w$ represents the width of a block. With an additional vector $(c^h, c^v)$ at the bottom-left corner as shown in Fig. 3 (b), the $\boldsymbol{mv}$ for point $(x, y)$ can be solved by (2):

$$\begin{cases} mv^h = \dfrac{b^h - a^h}{w}x + \dfrac{c^h - a^h}{h}y + a^h \\ mv^v = \dfrac{b^v - a^v}{w}x + \dfrac{c^v - a^v}{h}y + a^v, \end{cases} \tag{2}$$

where $h$ represents the height of a block.

In VTM, a block for affine motion can also be predicted in two ways for CME: unidirectional prediction and bi-prediction. Both four-parameter and six-parameter affine models can employ the two predictions as shown in Fig. 4. Either unidirectional prediction or bi-prediction for an AME process requires the associated reference frames, thereby increasing the encoding complexity of VTM. When only counting the number of required reference frames per the ME process, the AME process requires twice that of the CME process. Since AME has high complexity, it is better to use a threshold in deciding whether AME should be conducted or not; thus, the VTM uses a threshold-based decision scheme for fast AME encoding. Let us denote $J_{CME}$ and $J_{aff-4}$ as the best RD cost of the conventional ME and the best RD cost of four-parameter ME, respectively. The costs, $J_{CME}$ and $J_{aff-4}$, are compared with a threshold to decide whether six-parameter AME can be skipped or not. Although VTM uses this threshold-based skipping method, AME still has a high complexity, as shown in Fig. 1.

To obtain the best MV for each CU, the VTM conducts the ME process for the integer-pixel first, and subsequently, the sub-pixel ME from the best integer-pixel MV, which is in the same order as HM. However, because VTM has no PU partitions, VTM cannot use the best result of a square-shaped PU for other partitions. Thus, the ME process in the VTM

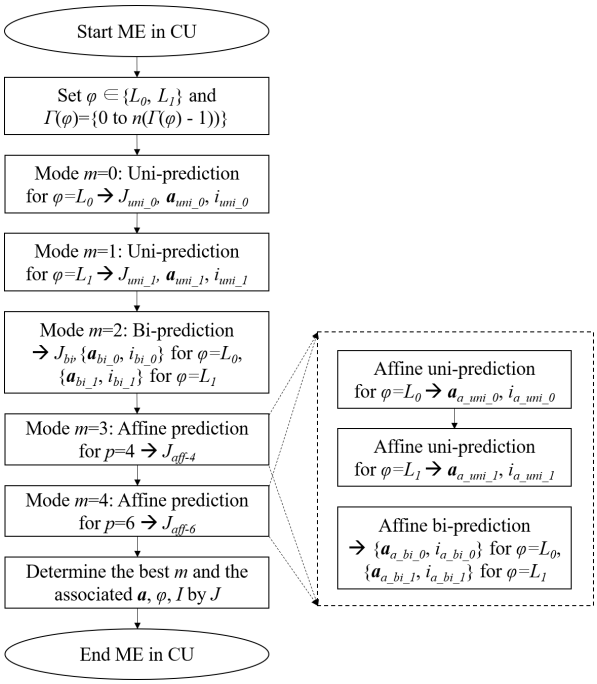



**FIGURE 4.** Overview of ME for a CU in VTM.

differs slightly from the HM [15]. It is noteworthy that the detailed process can be configured differently depending on the parameters set for ME.

In the RA configuration, an encoder searches multiple available reference frames to obtain the best MV and the best reference frame to minimize the RD cost of a block. As shown in (3), the method of Lagrange multipliers is used to compute the RD cost $J$ and compare the costs of the prediction results.

$$J = D + \lambda \cdot R, \tag{3}$$

where $D$ represents the block's distortion (image quality) cost, $\lambda$ represents the Lagrange multiplier, and $R$ represents the block bits (or estimated bits). Because two reference frame lists denoted by $L_0$ and $L_1$ are used for motion prediction, the ME process for unidirectional prediction should be tested with both lists, thereby generating all available frames in both lists. Let the set of reference frames be $\Gamma$, and let $\Gamma(\varphi)$ be the set of all available frames in the reference list $\varphi$. Accordingly, obtaining the minimum RD cost for a block in a reference frame $i$ can be formalized in (4) for unidirectional prediction and (5) for bi-prediction.

$$\underset{i\in\Gamma(\varphi)|\varphi=L0,L1}{\operatorname{argmin}} \{J(\varphi(i))\}, \tag{4}$$

and

$$\underset{i\in\Gamma(\varphi)|\varphi=L0,L1}{\operatorname{argmin}} \{\frac{J(\varphi(i)|\varphi=L_0)}{2} + \frac{J(\varphi(i)|\varphi=L_1)}{2}\}, \tag{5}$$

where $\Phi$ is a set of all reference frame lists.

The overall process of ME in VTM is depicted in Fig. 4. A CU in MTT node starts CME first, followed by AME. In both processes, the same reference frame sets $\Gamma(\varphi)$ are used in general within the number of $\Gamma(\varphi)$, denoting $n(\Gamma(\varphi))$. When all ME processes are conducted and each RD cost $J$ is obtained, the best mode $m$ with the best MV $\boldsymbol{a}$ and the best reference frame index $i$ given $\varphi$, can be chosen and stored as the best motion for the inter-prediction in the CU. As shown in Fig. 4, affine prediction has two variations depending on the parameter number, $p$. When $p = 4$, the 4-parameter model is used; when $p = 6$, the 6-parameter model is used. Based on the encoding configuration, either CME or AME or both can be accelerated in VTM using search range reduction, early termination, or skipping a part of ME based on RD cost.

### B. FAST AFFINE MOTION ESTIMATION

A primary complexity problem in CU encoding involves the ME and motion compensation (MC) processes. In particular, when the number of reference frames used for the ME increases, the associated complexity is also increased with a greater result accuracy, leading to more coding gains, and vice versa. Recognizing that videos in the real world have a variety of motions, this increased range could create computational and memory complexity. This should be overcome for fast encoders.

The VTM encoder adopts several strategies of HM in ME such as diamond search, raster search, and refinement processes [15]. Thus, the related work on HM could also be useful for determining whether the VTM can be easily accelerated. Research on fast ME can be classified under three strategies: simpler search pattern, adjusting search range, and early termination. The two former strategies have been studied for many years. Two diamond search patterns [16] were presented and are currently the core part of ME in both HM and VTM. Recently, to reduce the encoding complexity of HM, a directional search pattern method [13], a rotating hexagonal pattern [12] with some skipping methods, and an adaptive search range [17] showed reasonable results in a much smaller search range (i.e., 64) than with VTM.

Another approach to reducing the complexity of ME is reducing the number of reference frames to be searched. Pan *et al.* [26] present a reference frame selection method to reduce the encoding complexity of ME in HM. The number of initial reference frames is studied for fast ME encoding, since in general adjacent reference frames with high similarity tend to be selected [27]. As discussed in [28], [29], coding efficiency and computational complexity substantially vary depending on which reference frames are employed. Thus, reference frames should be carefully chosen in the context of other prediction tools.

A simpler, effective method is an early termination strategy that terminates redundant ME processes (either for unidirectional prediction or bi-prediction) per block. Skipping such ME process can significantly reduce the associated encoding time, but may result in quality degradation, thereby requiring high accuracy on the decision process for the early termination strategy. Recently, it has been discovered that the recursive block partitioning process in HEVC encoding provides a strong correlation of motion information so that

the ME process can be terminated with high accuracy. For example, previously encoded PU information in QT structure is exploited in [9] and extended with some modifications of motion search in [10], showing almost no coding loss. Also, the bi-prediction ME process can be terminated early with given PU information in QT structure [18], [19]. However, those early termination methods cannot be directly applied to the MTT structure as PU partitioning is no longer available in VTM. Thus, a new statistical analysis of the MTT is required for low-complexity VTM encoders.

VTM 3.0 used in our experiments includes the state-of-the-art algorithm on lightweight affine motion estimation (AME) in VVC. Since the first VVC test model (VTM 1.0) was released in 2018, there have been several works [30], [31] applied to the test model for the VVC standardization. Those efforts were made to search a better trade-off between coding efficiency and computational complexity. To be specific, in [30], Zhou proposed to reuse affine motion parameters of neighboring blocks instead of deriving the parameters again. In [31], Zhang *et al*. proposed to avoid the parameter derivation process of a small chroma block. The two methods are to reduce the total memory bandwidth and the encoding complexity of VTM 2.0, and they were integrated to VTM 3.0. However, there is much room to further reduce the AME complexity of VTM by using the MTT structure as discovered in the next section.

## III. PROPOSED METHOD

In this section, we propose a fast AME algorithm to tackle the computational complexity of ME in VTM. For this, we develop two features reflecting motion characteristics of the current CU from previously coded information and use them in a two-step fast AME algorithm. In the first stage, we conduct the early termination of the AME process at the level of a parent CU in MTT. The best prediction mode of a parent CU is examined, and then it is determined whether or not to skip the entire AME process based on the prediction mode. In the second stage, the prediction direction of the best reference frame in CME is examined to reduce the number of reference frames in the current CU.

In the following subsections, two proposed features—(a) the best inter-prediction mode of the parent CU and (b) the prediction direction in the CME of the current CU—are analyzed statistically to determine if they can be effectively used for the decision. Then, the two-step fast coding schemes using these features are presented in detail.

### A. STATISTICAL ANALYSIS OF FEATURE SELECTION

$p(A)$ computes a prior belief that the affine prediction mode is the best case among the inter-prediction coding tools. The event $A$ represents the case that the RD cost of the affine prediction, $J_{aff}$, is smaller than the RD cost results of CME, $J_{CME}$, obtained by solving (4) and (5). According to Bayes' theorem, after observing evidence, the probability distribution will provide more decisive information. In other words, the posterior probability given the proposed features can help us efficiently develop a fast AME method.

Following the notion of Bayes' theorem, we compute the posterior probability when the features are observed. For the first feature, we define $p(S_{par})$ as the probability that the best prediction mode of the parent CU is Skip mode. The best prediction mode is revealed after comparing the RD costs of other available prediction modes. The available modes include prediction tools (inter- and intra-prediction), not sub-tree structures (i.e., QT, BT or TT). If the RD cost of the Skip mode, $J_{skip}$, is smaller than the RD cost results of other prediction tools, then we regard the Skip mode as the best prediction mode of the CU. The Skip mode needs only merge index without residual coding, so a block can be inferred to have only slight motion. If a block prefers the Skip mode to the other inter-prediction modes, then the block could be regarded as a static area, which might not require motion-intensive tools such as affine prediction. For the second feature, we define $p(U_{CME})$ as the probability that the best prediction mode in CME is unidirectional prediction. The term ''best prediction'' is when the minimum RD cost of unidirectional prediction acquired by solving (4) is smaller than the minimum RD cost of bi-prediction acquired by solving (5). Then, $p(U_{CME})$ can be counted during the CME process for each available block. Intuitively, unidirectional prediction is chosen to predict a translational motion or a long-distance motion between scenes, but it is weak in predicting a non-linear motion such as zoom-in, zoom-out, or rotation. In this context, if a block is coded with unidirectional motion prediction, it can be inferred to have a simple motion, in which affine prediction is not likely chosen. By collecting the observation $p(S_{par})$ with the likelihood $p(S_{par}|A)$, we can compute the posterior probability $p(A|S_{par})$ as defined in (6):

$$p(A|S_{par}) = \frac{p(S_{par}|A)p(A)}{p(S_{par})}, \tag{6}$$

where $p(S_{par}|A)$ is the probability of the case that the best prediction mode of the parent CU is Skip mode given that $J_{aff} < J_{CME}$ for a CU, and $p(A|S_{par})$ is the probability of the case that $J_{aff} < J_{CME}$ given that the best prediction mode of the parent CU is Skip mode. Similarly, the posterior probability $p(A|U_{CME})$ can be computed with observation $p(U_{CME})$ and likelihood $p(U_{CME}|A)$ as defined in (7):

$$p(A|U_{CME}) = \frac{p(U_{CME}|A)p(A)}{p(U_{CME})}, \tag{7}$$

where $p(S_{par}|A)$ is the probability of the case that the best prediction mode in CME is unidirectional prediction given that $J_{aff} < J_{CME}$ for a CU, and $p(A|U_{CME})$ is the probability of the case that $J_{aff} < J_{CME}$ for a CU given that the best prediction mode in CME is unidirectional prediction.

Table 1 shows the probabilities, obtained from four UHD video sequences [22], encoded by VTM 3.0 under an RA configuration. 200 frames per sequence were encoded with two quantization parameters (QPs) of 25 and 35 for simplicity.

**TABLE 1. Probability of the motion data of previously encoded CU.**

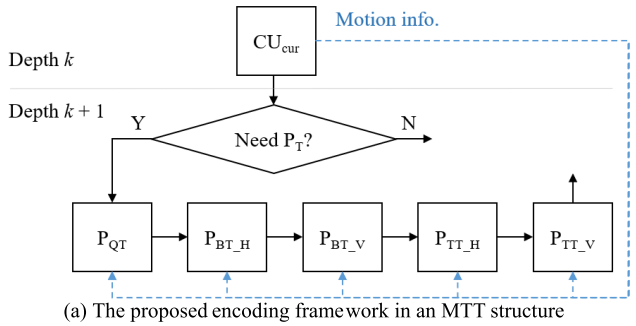
| Sequence | QP | $p(A)$ | $p(S_{par})$ | $p(A \mid S_{par})$ | $p(U_{CME})$ | $p(A \mid U_{CME})$ |
|---|---|---|---|---|---|---|
| Bosphorus | 25 | 27% | 39% | 12% | 43% | 24% |
| | 35 | 18% | 47% | 8% | 67% | 14% |
| Jockey | 25 | 29% | 28% | 10% | 74% | 25% |
| | 35 | 22% | 35% | 8% | 84% | 19% |
| YachtRide | 25 | 36% | 23% | 15% | 50% | 34% |
| | 35 | 26% | 34% | 14% | 64% | 23% |
| HoneyBee | 25 | 7% | 78% | 4% | 89% | 5% |
| | 35 | 4% | 79% | 2% | 96% | 3% |
| Average | | 21% | 45% | 9% | 71% | 18% |

We used sequences and QP values for this statistic different from those in Section IV to separate the training data from the test data. By distinguishing the material, we believe that the statistic presented in this section would not be biased against the test data presented in Section V.

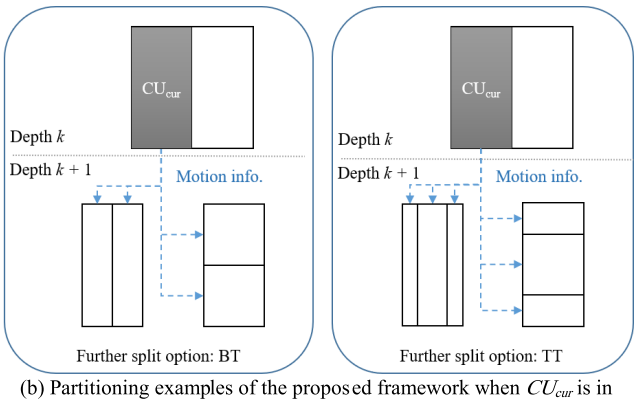
In Table 1, $p(A)$ varies with each sequence since the accuracy of the affine prediction is easily affected by the motion characteristics of a video sequence. However, as compared to prior $p(A)$, $p(A|\ S_{par})$ becomes quite small and steady. The result implies that a VTM encoder can skip much of the redundant AME process before coding the current CU when the parent CU is the Skip mode. For instance, in the *Bosphorus* and *Jockey* sequences, even though prior $p(A)$ is as high as around 29%, the posterior probability becomes dramatically smaller than the prior. Due to the various characteristics of video, motion-intensive sequences such as *YachtRide* are likely not to prefer the Skip mode, with observation $p(S_{par})$ below 40%. In such cases, the second observation $p(U_{CME})$ can be used instead. $p(U_{CME})$ remains high, at 64% in the *YachtRide* sequence and 84% in the *Jockey* sequence in QP 35. The posterior probability $p(A|U_{CME})$ becomes smaller than $p(A)$. When considering two given conditions—the prediction mode of the parent CU and the prediction direction of CME—the redundancy of AME can be determined more accurately. The probability $p(A|S_{par})$ of four sequences is only 9% on average, and $p(A|U_{CME})$ is below 20% on average.

### B. FAST AFFINE MOTION ESTIMATION METHOD

The proposed method consists of two parts: one is to extract features within an MTT structure and the other is to apply the algorithm of fast AME encoding. We describe the former one as shown in Fig. 5 with partitioning examples in an MTT structure. Fig. 5(a) shows the proposed encoding framework equipped with an MTT structure to filter out redundant AME processing. In VVC, there are more CU shapes, so a parent node can have a large number of sub-CUs. Therefore, the proposed method is designed to allow for sub-CUs in recursive block-partitioning to use the previously-encoded information when building QT ($P_{QT}$), BT horizontal ($P_{BT\_H}$), BT vertical ($P_{BT\_V}$), TT horizontal ($P_{TT\_H}$), and TT vertical ($P_{TT\_V}$)




(a) The proposed encoding framework in an MTT structure




(b) Partitioning examples of the proposed framework when $CU_{cur}$ is in vertical BT

**FIGURE 5. The proposed encoding framework in MTT structure for passing motion information and associated examples.**

structures in the child nodes. In Fig. 5(a), the prediction information of $CU_{cur}$ is delivered along a dashed line. In this framework, the first key feature—the best prediction mode of a parent CU—is used to determine whether to conduct the AME in the sub-CUs. For instance, as shown in Fig. 5(b), a CU partitioned in BT can further split into at most four cases ($P_{BT\_H}$,$P_{BT\_V}$,$P_{TT\_H}$, and $P_{TT\_V}$). In this case, ten sub-CUs use the information of $CU_{cur}$ to decide whether AME in each sub-CU is redundant or not. This example shows the efficiency of the proposed framework, applied to the variety of partitioning in an MTT structure. Recall that the second feature can be used for further complexity reduction if the first feature does not meet the condition to skip the AME process.

Fig. 6 shows a block diagram of the proposed method for how the conventional ME in the original VTM software is changed. The steps in the VTM software algorithm are highlighted. As shown, the best prediction mode of a parent CU, $CU_{par}$, is checked in the first stage. If the best mode of $CU_{par}$ is Skip mode, then the entire AME process is skipped at the current CU. Thus, in the case that the best mode of $CU_{par}$ is Skip mode, the best mode $m$ of the ME process for this CU is one of unidirectional prediction or bi-prediction of CME. The best MV $\boldsymbol{a}$ of the ME process for this CU is likely to have translational motion. Here the ME process refers to the entire ME process including CME and AME, excluding other processes such as the regular skip/merge prediction and the affine skip/merge prediction.

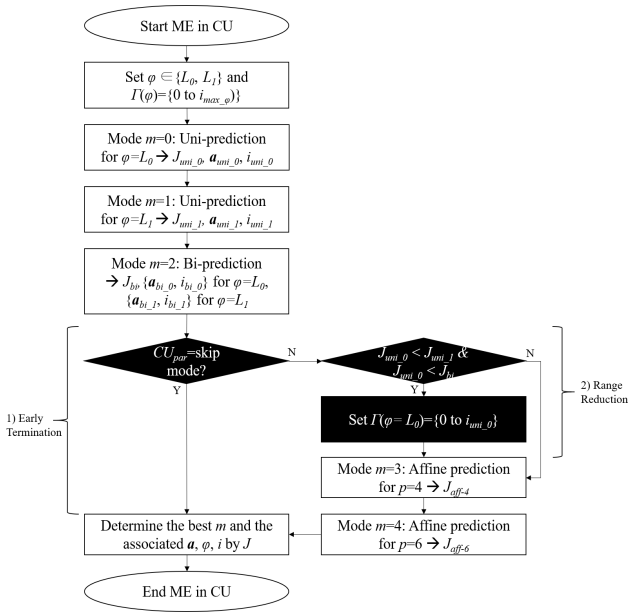



**FIGURE 6.** Flowchart of the proposed fast AME method in VTM software.

In the second stage, the best mode of $CU_{par}$ is not Skip mode; instead, the best prediction direction mode $m$ of CME is checked by comparing the RD cost $J$ of each prediction. If $m$ is 0 (unidirectional prediction and thereby the best $\varphi$ is $L_0$), then the reference frame set for AME, $\Gamma(\varphi = L_0)$, is reduced in size. In this case, the maximum reference frame index for $\Gamma(\varphi = L_0)$ is decreased to the best reference frame index of $L_0$ for CME, $i_{uni\_0}$, if $i_{uni\_0}$ is smaller than the maximum reference frame index. The other reference list, $L_1$, is not changed for the proposed method, as the minimization of $L_1$ could lose coding performance noticeably.

## IV. EXPERIMENTS AND RESULTS

The compression performance was measured by the Bjøntegaard-Delta bitrate (BDBR) measurement method [20], using bitstream results encoded by four QP values: 22, 27, 32, and 37. The test materials were chosen from the common test condition (CTC) for standard dynamic range (SDR) video [21]. Tested sequences herein are categorized in five classes as in [21]. Class B, C, and D represent 1920 × 1080, 832 × 480, and 416 × 240 video resolutions, respectively. Class A1 and Class A2 with 3840 × 2160 video resolutions are also tested, and due to the limited resources, Class A2 sequences are encoded with 100 frames only. We compared the proposed method with the VTM 3.0 implementation that contains the state-of-the-art of AME. Conforming to CTC [21], test videos were encoded by the VTM 3.0 as an anchor in the experiments under the RA configuration. The proposed method is also implemented using the same reference software.

The time complexity of AME was measured by the running time. The AME time ratio, *ATR*, is reported in comparison with the anchor, by measuring the entire time of AME and the associated AMC functions. Since AME time (AT) may vary with QP values, *ATR* per each sequence is calculated by the geometric mean of four AT results as shown in (8):

$$ATR = \left( \prod_{i=1}^{4} \frac{AT_{\psi}(QP_i)}{AT_{o}(QP_i)} \right)^{\frac{1}{4}}, \tag{8}$$

where $QP_i$ is the QP value of the $i^{th}$ bitstream, $AT_o$ is the AME time of the anchor, and $AT_\psi$ is the AME time of the method $\psi$ to be compared. In addition, the total encoding time ratio, *ETR*, is reported in comparison with the anchor. Similar to calculating *ATR*, the geometric mean is employed for the average of four encoding time results corresponding to QP values. The experiments are conducted in computing platforms of 64-bit Windows OS, 32 GB RAM, and Intel i7-8700 series for a CPU.

Table 2 shows the performance comparisons of the proposed method and the anchor in terms of both coding efficiency and encoding complexity. The proposed method reduces the AME time of the VTM to 63%, while the coding loss is 0.1%, on average, in comparison with the anchor. At the maximum performance, the *ATR* of the *BQSquare* sequence reaches 39%, with a 0.47% loss. At the minimum performance, the *ATR* of the *RitualDance* sequence reaches 79%, which is still a noticeable improvement. As shown in Table 2, the difference in coding performance for all tested sequences between the anchor and the proposed method is minimal. The *BQSquare* sequence is the worst case in terms of BDBR of the Y (luma) component, yet the loss is still within 0.5%. Moreover, when considering other chrominance components (i.e., U and V), the average BDBR loss of Y, U, and V is less than the loss of BDBR Y only. In particular, in several sequences, BDBR-U and BDBR-V performed better than the anchor. For example, the *BQMall* sequence showed a 0.20% decrease in both BDBR-U and BDBR-V, and the *BasketballPass* sequence showed a coding gain in both U and V components. It is noteworthy that, in general, the proposed method sustained coding efficiency robustly, especially in higher resolutions (i.e., Class A1, A2 and B). It is observed that the ETR is around 95% with the slight coding loss. The best ETR is observed in the BQSquare as in the ATR. It is noted that the AME is implemented with Single Instruction Multiple Data (SIMD) as an optimized parallel processing technique. This aspects imply that the complexity of AME could increase furthermore if an encoder does not support such the architecture-dependent optimization technique.

The actual running time of the AME process is measured per QP value as shown in Fig. 7. In QP 22, the sum of running time for the AME process of all sequences is 36 hours in the anchor. However, the proposed method reduces the AME time by 8 hours, approximately. This trend can be similarly observed in other QP values. Moreover, Fig. 7 shows that the AME time is reduced to about 55% in QP 37. When considering the complexity reduction is more challenging in a higher QP value, this aspect can be efficiently used for an encoder of low-end devices.

**TABLE 2.** Results of the performance of the proposed method compared with the anchor.

| Sequence name | BDBR-Y | BDBR-U | BDBR-V | ETR | ATR |
|---|---|---|---|---|---|
| Tango2 | 0.04% | -0.19% | 0.08% | 93% | 63% |
| FoodMarket4 | 0.04% | 0.08% | 0.05% | 95% | 72% |
| CatRobot1 | 0.05% | 0.55% | 0.13% | 93% | 55% |
| DaylightRoad2 | 0.14% | 0.04% | 0.18% | 94% | 59% |
| ParkRunning3 | 0.05% | 0.00% | 0.03% | 96% | 65% |
| MarketPlace | 0.10% | 0.15% | 0.11% | 95% | 61% |
| RitualDance | 0.04% | 0.01% | 0.13% | 97% | 79% |
| Cactus | 0.11% | -0.03% | 0.04% | 96% | 65% |
| BasketballDrive | 0.08% | -0.01% | 0.10% | 95% | 66% |
| BQTerrace | 0.04% | 0.00% | -0.20% | 92% | 40% |
| BasketballDrill | 0.06% | -0.04% | 0.02% | 97% | 73% |
| BQMall | 0.05% | -0.20% | -0.20% | 95% | 61% |
| PartyScene | 0.26% | 0.06% | 0.04% | 96% | 60% |
| RaceHorses | 0.06% | 0.09% | 0.12% | 98% | 75% |
| BasketballPass | 0.08% | 0.08% | 0.16% | 98% | 73% |
| BQSquare | 0.47% | -0.13% | 0.01% | 90% | 39% |
| BlowingBubbles | 0.12% | -0.17% | 0.14% | 94% | 57% |
| RaceHorses | 0.08% | -0.12% | -0.03% | 95% | 71% |
| Class A1 | 0.04% | -0.05% | 0.06% | 94% | 68% |
| Class A2 | 0.08% | 0.20% | 0.11% | 94% | 60% |
| Class B | 0.07% | 0.02% | 0.04% | 95% | 62% |
| Class C | 0.11% | -0.02% | -0.01% | 96% | 67% |
| Class D | 0.19% | -0.08% | 0.07% | 94% | 60% |
| Overall | 0.10% | 0.01% | 0.05% | 95% | 63% |

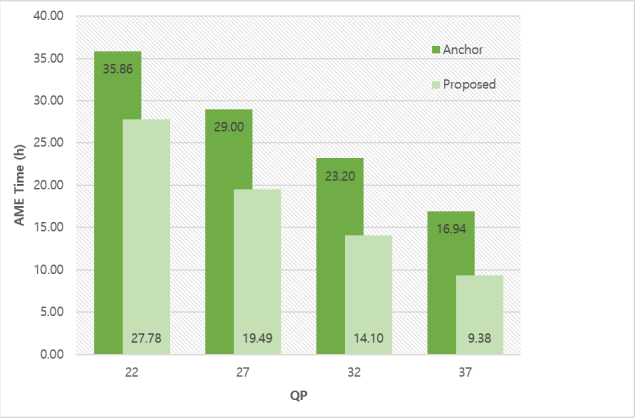



**FIGURE 7.** Sum of actual running time (hour) per QP value of the proposed method and the anchor for the proposed method.

As the proposed method contain two stages to accelerate the AME process, an additional experiment was conducted to evaluate the implementation of each stage on top of the anchor. For simplicity, sequences were encoded with the smaller number of frame counts: 100 frames for Class A1 and A2 sequences, but 3 seconds for other sequences. As shown in Fig. 6, Stage I is the early termination scheme that terminates the entire AME process, whereas Stage II is the range reduction scheme that reduces the number of reference frames for AME.

**TABLE 3.** Performance of two separate implementations of the proposed method compared to the anchor.

| Sequence name | Stage I | | Stage II | |
|---|---|---|---|---|
| | BDBR-Y | ATR | BDBR-Y | ATR |
| Tango2 | 0.01% | 69% | 0.01% | 95% |
| FoodMarket4 | 0.02% | 81% | 0.02% | 94% |
| Campfire | -0.02% | 85% | 0.00% | 95% |
| DaylightRoad2 | 0.13% | 62% | 0.05% | 95% |
| ParkRunning3 | 0.02% | 68% | 0.02% | 96% |
| MarketPlace | 0.16% | 57% | 0.05% | 96% |
| RitualDance | 0.01% | 83% | -0.06% | 95% |
| Cactus | 0.15% | 66% | 0.06% | 96% |
| BasketballDrive | 0.04% | 74% | 0.02% | 95% |
| BQTerrace | 0.14% | 44% | 0.12% | 97% |
| BasketballDrill | -0.05% | 76% | 0.00% | 95% |
| BQMall | 0.07% | 70% | 0.01% | 97% |
| PartyScene | 0.10% | 66% | 0.05% | 98% |
| RaceHorsesC | 0.07% | 81% | -0.13% | 97% |
| BasketballPass | 0.05% | 69% | 0.10% | 98% |
| BQSquare | 0.43% | 42% | -0.08% | 98% |
| BlowingBubbles | 0.05% | 63% | 0.16% | 98% |
| RaceHorses | 0.09% | 79% | 0.01% | 97% |
| Class A1 | 0.00% | 78% | 0.01% | 95% |
| Class A2 | 0.08% | 65% | 0.03% | 96% |
| Class B | 0.10% | 65% | 0.04% | 96% |
| Class C | 0.05% | 73% | -0.02% | 97% |
| Class D | 0.16% | 63% | 0.05% | 98% |
| Overall | 0.08% | 69% | 0.02% | 96% |

Table 3 shows the result of the additional experiment in comparison with the same anchor. Stage I reduced the ATR to 69%, as this process could skip the entire AME process if applicable. The best performance of Stage I was achieved with the *BQSquare* sequence as in Table 2 (results of the full integration of the proposed method). Stage II saved 4% of ATR but sustained most of the original compression performance, showing only a 0.02% decrease of BDBR-Y. It is noteworthy that Stage II sometimes gained compression performance: 0.13%, 0.08%, and 0.06% gains for *RaceHorsesC*, *BQSquare* and *RitualDance* sequences. In conclusion, both stages in the proposed method contributed to reducing encoding complexity while sustaining coding efficiency.

## V. CONCLUSION

VTM achieves far better compression performance than its predecessor, the HEVC standard, according to the literature. Among the newly adopted technologies for VTM, affine prediction contributes substantially to compression performance by capturing a greater variety of motions found in nature. However, the complexity of AME is a bottleneck for low-complexity encoder applications. In this paper,

the encoding complexity of AME was investigated, and the associated key observation was discovered using statistics. A fast AME method was proposed for a VVC encoder. The proposed method showed that the AME time of VTM was reduced to 64% on average, with negligible coding loss. We believe that these contributions could help promote future research on the complexity of VVC encoders. In future work, we will develop a machine learning-based fast algorithm to automatically learn the features as practiced in [23].

## REFERENCES

[1] B. Bross, *Versatile Video Coding (Draft 1), Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-J1001, Apr. 2018.

[2] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 3 (VTM 3), Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L1002, Oct. 2018.

[3] J. An, H. Huang, K. Zhang, Y. -W. Huang, and S. Lei, *Quadtree plus binary tree structure integration with JEM tools, Joint Video Experts Team (JVET)*, Document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-B0023, Feb. 2016.

[4] Z. Wang, S. Wang, J. Zhang, S. Wang, and S. Ma, "Probabilistic decision based block partitioning for future video coding," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1475–1486, Mar. 2018.

[5] S.-H. Park, T. Dong, and E. S. Jang, "Low complexity reference frame selection in QTBT structure for JVET future video coding," in *Proc. IWAIT*, Chiang Mai, Thailand, Jan. 2018, pp. 1–4.

[6] F. Bossen, X. Li, K. Suehring, *AHG Report: Test Model Software Development (AHG3), Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-K0003, Jul. 2018.

[7] L. Li, H. Li, D. Liu, Z. Li, H. Yang, S. Lin, H. Chen, and F. Wu, "An efficient four-parameter affine motion model for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1934–1948, Aug. 2018.

[8] K. Zhang, Y.-W. Chen, L. Zhang, W.-J. Chien, and M. Karczewicz, "An improved framework of affine motion compensation in video coding," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1456–1469, Mar. 2019.

[9] Z. Pan, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast motion estimation based on content property for low-complexity H.265/HEVC encoder," *IEEE Trans. Broadcast.*, vol. 62, no. 3, pp. 675–684, Sep. 2016.

[10] S. Park and E. S. Jang, "Fast motion estimation based on content property for low-complexity H.265/HEVC Encoder," *IEEE Trans. Broadcast.*, vol. 63, no. 4, pp. 740–742, Dec. 2017.

[11] S. Y. Jou, S. J. Chang, and T. S. Chang, "Fast motion estimation algorithm and design for real time QFHD high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 9, pp. 1533–1544, Sep. 2015.

[12] P. Nalluri, L. N. Alves, and A. Navarro, "Complexity reduction methods for fast motion estimation in HEVC," *Signal Process.-Image Commun.*, vol. 39, pp. 280–292, Nov. 2015.

[13] S.-H. Yang, J.-Z. Jiang, and H.-J. Yang, "Fast motion estimation for HEVC with directional search," *Electron. Lett.*, vol. 50, no. 9, pp. 673–675, Apr. 2014.

[14] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[15] C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan, *High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 9, Joint Collaborative Team on Video Coding (JCT-VC)*, document ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, JCTVC-AB1002, Jul. 2017.

[16] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, Feb. 2000.

[17] W.-D. Chien, K.-Y. Liao, and J.-F. Yang, "Enhanced AMVP mechanism based adaptive motion search range decision algorithm for fast HEVC coding," in *Proc. ICIP*, Paris, France, Oct. 2014, pp. 3696–3699.

[18] S.-H. Park, S.-H. Lee, E. S. Jang, D. Jun, and J.-W. Kang, "Efficient biprediction decision scheme for fast high efficiency video coding encoding," *Proc. SPIE*, vol. 25, no. 6, Nov. 2016, Art. no. 063007.

[19] C. E. Rhee and H. J. Lee, "Early decision of prediction direction with hierarchical correlation for HEVC compression," *IEICE Trans. Inf. Syst.*, vol. 96, no. 4, pp. 972–975, Apr. 2013.

[20] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, Document ITU-T SG16/Q6 VCEG, VCEG-M33, Apr. 2001.

[21] F. Bossen, J. Boyce, K. Suehring, X. Li, and V. Seregin, *JVET common test conditions and software reference configurations for SDR video Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L1010, Oct. 2018.

[22] Ultra Video Group. *Laboratory of Pervasive Computing at Tampere University of Technology*. Accessed: Nov. 9, 2018. [Online]. Available: http://ultravideo.cs.tut.fi/

[23] S. Ryu and J. Kang, "Machine learning-based fast angular prediction mode decision technique in video coding," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5525–5538, Nov. 2018.

[24] M. Paul, W. Lin, C. T. Lau, and B.-S. Lee, "Direct intermode selection for H.264 video coding using phase correlation," *IEEE Trans. Image Process.*, vol. 20, no. 2, pp. 461–473, Feb. 2011.

[25] P. K. Podder, M. Paul, and M. Murshed, "Efficient video coding using visual sensitive information for HEVC coding standard," *IEEE Access*, vol. 6, pp. 75695–75708, 2018.

[26] Z. Pan, P. Jin, J. Lei, Y. Zhang, X. Sun, and S. Kwong, "Fast reference frame selection based on content similarity for low complexity HEVC encoder," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 516–524, Oct. 2016.

[27] S. Wang, S. Ma, S. Wang, D. Zhao, and W. Gao, "Fast multi reference frame motion estimation for high efficiency video coding," in *Proc. ICIP*, Melbourne, VIC, Australia, Sep. 2013, pp. 2005–2009.

[28] D. M. M. Rahaman and M. Paul, "Virtual view synthesis for free viewpoint video and multiview video compression using Gaussian mixture modelling," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1190–1201, Mar. 2018.

[29] M. Paul, "Efficient multiview video coding using 3-D coding and saliency-based bit allocation," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 235–246, Jun. 2018.

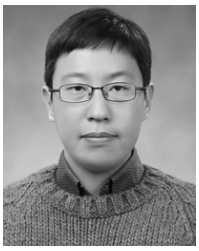
[30] M. Zhou, *CE4: Test Results of CE4.1.11 on Line Buffer Reduction for Affine Mode, Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0045, Oct. 2018.

[31] K. Zhang, L. Zhang, H. Liu, Y. Wang, P. Zhao, D. Hong, *CE4: Affine Prediction with 4ÃU4 Sub-Blocks for Chroma Components (Test 4.1.16), Joint Video Experts Team (JVET)*, document ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, JVET-L0265, Oct. 2018.

**SANG-HYO PARK** received the B.S. and Ph.D. degrees in computer engineering and computer science from Hanyang University, Seoul, South Korea, in 2011 and 2017, respectively.

From 2017 to 2018, he held a postdoctoral position with the Intelligent Image Processing Center, Korea Electronics Technology Institute, and a Research Fellow with the Barun ICT Research Center, Yonsei University, in 2018. Since 2019, he holds a postdoctoral position with the Department of Electronic and Electrical Engineering, Ewha Womans University, Seoul. He is the author/coauthor of several scientific/technical articles in International conferences/journals, of several pending or approved patents, and of more than 50 MPEG/JCT-VC/JVET contribution documents. His research interests include HEVC, VVC, encoding complexity, omnidirectional video, and deep learning.

Dr. Park has been actively participating in the standardization work of the Moving Picture Experts Group (MPEG) and the Joint Collaborative Team on Video Coding (JCT-VC), since 2011, and the Joint Video Exploration Team (JVET), since 2015. He has served as a Co-Editor of Internet Video Coding (IVC, ISO/IEC 14496-33) for standardization for six years.



**JE-WON KANG** received the B.S. and M.S. degrees in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2006 and 2008, respectively, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2012.

He was a Senior Engineer with the Multimedia RnD and Standard team in Qualcomm Technologies, Inc., San Diego, CA, from 2012 to 2014. He was a Visiting Researcher with Tampere University, the Nokia Research Center, Tampere, Finland, in 2011, and the Mitsubishi Electric Research Laboratory, Boston, USA, in 2010. He is currently an Associate Professor with Ewha Womans University, Seoul, South Korea, and the Head of the Information Coding and Processing Laboratory, Department of Electronics and Electrical Engineering, Ewha Womans University. He has been an active contributor to the recent international video coding standards in JCT-VC, including High Efficiency Video Coding (HEVC) standard and the extensions to multiview videos, 3D videos, and screen content videos. His current research topics include image and video processing and compression, computer vision, and machine learning.