Unimodal Model-Based Inter Mode Decision for High Efficiency Video Coding

ZENG, Huanqiang; XUANG, Wenjie; CHEN, Jing; CAI, Canhui; NI, Zhangkai; MA, Kai-Kuang

Published in:
IEEE Access

Published: 01/01/2019

Document Version:
Final Published version, also known as Publisher’s PDF, Publisher’s Final version or Version of Record

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2019.2902196

Publication details:
https://doi.org/10.1109/ACCESS.2019.2902196

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.
Unimodal Model-Based Inter Mode Decision for High Efficiency Video Coding

HUANQIANG ZENG1, (Senior Member, IEEE), WENJIE XIAN2, JING CHEN1, (Member, IEEE), CANHUI CAI1, (Senior Member, IEEE), ZHANGKAI NI3, (Member, IEEE), AND KAI-KUANG MA4, (Fellow, IEEE)

1School of Information Science and Engineering, Huaqiao University, Xiamen 361021, China
2Minnan Science and Technology Institute, Fujian Normal University, Quanzhou 362300, China
3Department of Computer Science, City University of Hong Kong, Hong Kong
4School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Jing Chen (jingzi@hqu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61871434, Grant 61802136, and Grant 61602191, in part by the Natural Science Foundation for Outstanding Young Scholars of Fujian Province under Grant 2019J06017, in part by the Natural Science Foundation of Fujian Province under Grant 2016J01308 and Grant 2017J05103, in part by the Fujian-100 Talented People Program, in part by the High-Level Talent Innovation Program of Quanzhou City under Grant 2017G027, in part by the Promotion Program for Young and Middle-Aged Teacher in Science and Technology Research of Huaqiao University under Grant QZN-YY403 and Grant QZN-PY418, and in part by the High-Level Talent Project Foundation of Huaqiao University under Grant 14BS201, Grant 14BS204, and Grant 16BS108.

ABSTRACT In this paper, a fast inter mode decision algorithm, called the unimodal model-based inter mode decision (UMIMD), is proposed for the latest video coding standard, the high-efficiency video coding. Through extensive simulations, it has been observed that a unimodal model (i.e., with only one global minimum value) can be established among the size of different prediction unit (PU) modes and their resulted rate-distortion (RD) costs for each quad-tree partitioned coding tree unit (CTU). To guarantee the unimodality and further search the optimal operating point over this function for each CTU, all the PU modes need to be first classified into 11 mode classes according to their sizes. These classes are then properly ordered and sequentially checked according to the class index, from small to large so that the optimal mode can be early identified by checking when the RD cost starts to arise. In addition, an effective instant SKIP mode termination scheme is developed by simply checking the SKIP mode against a pre-determined threshold to further reduce the computational complexity. The extensive simulation results have shown that the proposed UMIMD algorithm is able to individually achieve a significant reduction on computational complexity at the encoder by 61.9% and 64.2% on average while incurring only 1.7% and 2.1% increment on the total Bjontegaard delta bit rate (BDBR) for the low delay and random access test conditions, compared with the exhaustive mode decision in the HEVC. Moreover, the experimental results have further demonstrated that the proposed UMIMD algorithm outperforms multiple state-of-the-art methods.

INDEX TERMS HEVC, coding tree unit, prediction unit, inter mode decision, unimodal model.

I. INTRODUCTION

The latest video coding standard, called the high efficiency video coding (HEVC), is a joint effort of two standardization organizations—the video coding experts group (VCEG) and the moving picture experts group (MPEG) [1]. Compared with its predecessor—the H.264/AVC [2], the HEVC is able to deliver about 50% bit rate reduction while maintaining a similar subjective video quality [3]. However, this is accomplished at a very high cost on computational complexity of HEVC encoder. Unlike the H.264/AVC, the HEVC adopts a much more flexible quad-tree partitioned coding tree unit (CTU) structure, which consists of a set of variable-sized coding units (CUs), the prediction units (PUs), and the transform units (TUs). For each CTU, an exhaustive evaluation of all available modes under various sizes are performed for identifying the best configuration that will yield the least rate distortion (RD) cost. Such so-called inter mode decision
performed for each CTU needs to be determined before the actual encoding process of image sequence starts. Inevitably, this inter mode decision process is fairly time consuming. How to reduce the computational complexity while maintaining almost the same coding efficiency has become the central issue of encoding optimization for realizing any practical HEVC encoder.

Since the inception of the H.264/AVC, fast intra and inter mode decision methods have been developed as an effective solution for encoding optimization (e.g., [4]–[8]). However, these mode decision methods cannot be directly applied to the HEVC, since the coding structure of the HEVC is much more flexible and sophisticated than that of the H.264/AVC. It is therefore necessary to develop fast mode decision methods specifically for HEVC and some most recently introduced ones will be highlighted in Section II-C. The main idea of these methods is to only check those more likely CUs/PUs rather than all of them so that the computational complexity can be effectively reduced with acceptable loss in coding performance. Those more likely CUs/PUs for current CTU are generally selected by using various coding information of the spatial/temporal neighboring CTUs, e.g., residual [9], motion vector (MV) [10], [11], CU depth [12], [13], optimal mode [14], RD cost [15], texture information [16], all zero blocks [17], sample-adaptive-offset coefficients [18], coded block flag information [18], etc. and various strategies, such as, Bayesian approach [9], [19], [20], conditional random field [21] neighboring prediction [14], [22], adaptive ordering [23], complexity control [24]–[26], linear programming [10], machine learning [27]–[29], etc. Although many efforts has been devoted to HEVC encoder complexity reduction, there is still ample room for further improvement.

For that, an efficient inter mode decision algorithm, called the unimodal model-based inter mode decision (UMIMD), is developed for the HEVC in this paper. In our approach, a special relationship among the PU modes and their resulted RD costs has been observed and modeled as a unimodal function. That is, there is only one global minimum value existing in this function. This greatly facilitates the search process on the identification of this minimum point. To guarantee the unimodality and further identify this point, firstly, all the PU modes need to be classified into several mode classes according to the PU’s size. All these classes need to be properly ordered such that the above-mentioned unimodal property can be yielded for each CTU. To search for the optimal operating point, the RD costs of each pair of two consecutive mode classes are sequentially compared to determine whether the mode class with higher class-index starts to yield a larger RD cost. If so, the global minimum is considered just found (namely, the one with lower class index). In addition, to possibly reduce the computational complexity further, an instant SKIP mode termination scheme is proposed, which is based on an evaluation of the RD cost of the SKIP mode against a pre-determined threshold. Extensive experimental results have shown that the proposed UMIMD algorithm is able to significantly reduce the computational complexity while maintaining almost the same video coding quality and the total bit rate, compared with the exhaustive mode decision used in the HEVC.

The remaining of this paper is organized as follows. Section II provides a brief overview of the inter mode decision conducted in the HEVC as the background and related fast mode decision methods. Section III describes the proposed fast inter mode decision algorithm for the HEVC in detail. Extensive simulation results are presented and discussed in Section IV. Finally, the conclusion is drawn in Section V.

II. OVERVIEW OF THE INTER MODE DECISION IN THE HEVC AND RELATED WORKS

A. FLEXIBLE CODING STRUCTURE

One of the new features introduced in the HEVC is its highly flexible coding structure, allowing a fixed-sized large block (64 × 64) to be partitioned in various ways for pursuing better coding efficiency [30]. Analogous to the concept of 16 × 16 macroblock (MB) as used in the previous video coding standard—H.264/AVC, the so-called coding tree unit (CTU) is defined in the HEVC as the basic processing block unit with a size of 64 × 64. Each CTU is subject to be further divided into non-overlapping smaller blocks, called as the coding units (CUs) in the HEVC, using the quad-tree partition operation. Note that each CU might be further partitioned to even smaller CUs likewise. Each stage of such quad-tree partition is referred as reaching to the next depth, a special parlance as defined in the HEVC. As demonstrated in Fig. 1, the block size of CU ranges from 64 × 64 (i.e., the CTU) to 8 × 8. In other words, CU_{64×64} corresponds to depth = 0; hence, CU_{32×32} is at depth = 1, CU_{16×16} is at depth = 2, and CU_{8×8} is at depth = 3.

For each 2N × 2N CU in any depth, it has multiple combinations of various-sized partitioned rectangular blocks, and each combination is called a prediction unit (PU) and defined as an unique mode in the HEVC. This set of modes as defined by the HEVC is subject to conduct inter prediction or intra prediction for each CU individually to determine which one is the best mode for that CU (Refer to Fig. 2). In addition to the SKIP mode, these modes include 2N × 2N, 2N × N, N × 2N, N × N, nL × 2N, nR × 2N, 2N × nU, and 2N × nD, to effectively employ the temporal correlation inherited in video. For performing intra prediction, there are only two types of PU modes: 2N × 2N and N × N, where the latter one is only available for the 8 × 8 CU (i.e., N = 4).

Lastly, there are three transform unit (TU) modes have been introduced in the HEVC for conducting DCT-based transform and quantization of prediction residuals. Three TU modes are with different block sizes, including 2N × 2N, N × N, and N/2 × N/2. In summary, Fig. 2 shows the available PU modes and TU modes for a 2N × 2N CU.

B. EXHAUSTIVE MODE DECISION

Although the above-described flexible coding structure is able to yield a much higher coding efficiency (compared to
FIGURE 1. An example of the coding tree unit (CTU), which is the basic processing unit as practiced in the HEVC. Each CTU consists of a set of coding units (CUs) that are recursively generated using quad-tree partition process. Note that the block size of the CU is variable.

FIGURE 2. PU modes and TU modes for a $2N \times 2N$ CU.

that of the previous video coding standard—H.264/AVC), the key concern is its involved computational complexity on the determination of the optimal mode combination—namely, the best combination of the optimal CU partitions, PU, and TU modes among all available modes. For that, the HEVC reference software (i.e., HM) exploits the so-called exhaustive mode decision strategy to assure that the optimal one will be identified at the end of mode decision process.

In the HM, the Lagrangian rate distortion optimization (RDO) function [31] is exploited as the mode decision criterion. That is, all the CU, PU and TU modes need to be exhaustively evaluated by computing the RD cost resulted from each mode. These computed costs will be compared to identify which one has the least RD cost that corresponds to the optimal mode combination. The Lagrangian RDO function is defined as

$$J = SSE(s, c|QP) + \lambda \cdot R(s, c|QP),$$

where $QP$ is the quantization parameter, $\lambda$ is the Lagrangian multiplier related to $QP$, $R$ represents the number of bits required, $SSE$ stands for the sum of squared error calculated between the original luma block (i.e., $s$) and its reconstructed block (i.e., $c$); that is,

$$SSE = \sum_{(x,y) \in s,c} |s(x,y) - c(x,y)|^2.\) (2)

Although the above-described exhaustive mode decision strategy is able to find the optimal mode combination for ensuring the high coding efficiency, however the involved computational complexity is extremely high as expected. Therefore, it is very essential to develop a fast mode decision method for the HEVC, aiming to significantly reduce the computational complexity without causing the degradation on coding efficiency. In what follows, we shall highlight several state-of-the-art fast inter mode decision methods for HEVC before presenting ours.

C. LITERATURE REVIEW

Multiple fast mode decision methods for the HEVC can be found in the literature. Shen et al. [9] developed a fast CU decision method by exploiting the Baysian decision rule, where the conditional probability is computed based on the evaluation of the sum of absolute transformed difference (SATD), the magnitude of MV, and the RD cost. Zhao et al. [10] presented a flexible CU decision method by exploiting the mode mapping scheme, together with the use of an optimal stopping criterion, to reduce the computational load. Mallikarachchi et al. [11] presented a content-adaptive feature-based CU decision method. In this method, two CU split likelihood models are trained to model the CU split and non-split decisions, and a motion vector reuse mechanism is developed to further expedite the motion estimation process.
Shen et al. [12] presented a fast mode decision algorithm that consists of two schemes—adaptive depth range determination and early termination of unnecessary motion estimation (ME) process that might be encountered on small-sized CUs. Tai et al. [13] avoided CU partition procedure and truncated PU prediction by investigating the depth correlation between the current CU and the collocated CU, and analyzing the RD cost in different cases. Shen et al. [14] proposed a hybrid mode decision method, in which all the candidate modes are firstly classified into three mode classes. Among them, only one will be chosen to conduct further evaluations based on the mode complexity of the current CU, which is measured according to the coding information of its neighboring and temporally co-located CUs. Two early termination schemes are further incorporated to early select the SKIP/PU mode as the optimal mode of the current CU by using the mode and RD cost correlations between the current CU and its spatially and/or temporally adjacent CUs. Vanne et al. [15] addressed the complexity overhead of the symmetric motion partition and asymmetric motion partition in the HEVC by analyzing the rate-distortion complexity of the HEVC inter prediction as a function of different block partition structures. Goswami et al. [16] proposed a low complexity coding method for HEVC through texture analysis. Pan et al. [17] proposed an early MERGE mode decision method by utilizing ME information and all zero blocks. Ahn et al. [18] proposed a fast CU encoding scheme by making full use of the spatial and temporal encoding parameters produced by the HEVC encoder. In this method, the spatial encoding parameters, sample-adaptive-offset coefficients, and the temporal encoding parameters, including MV, TU sizes, and the coded block flag information, are used to individually estimate the texture characteristic and temporal complexity of each CU, which are then utilized to conduct early SKIP mode detection and fast CU split for significantly reducing the computational complexity. Lee et al. [19] proposed a method involving three stages of early terminations, in which three thresholds are designed by using Bayes’ rule with a complexity factor. Goswami and Kim [20] developed a fast scheme for HEVC, in which the Bayesian classifier is designed by treating SKIP detection and CU termination as a two-class decision problem and trained by using Markov chain Monte Carlo model. Zhang et al. [21] developed a two-stage fast inter CU decision method by exploring CU classification based on Bayesian rule, and early CU skipping/pruning scheme based on conditional random fields. Duan et al. [22] presented an adaptive quad-tree depth range prediction mechanism based on the depth correlation within the neighboring region. Jung and Park [23] explored an adaptive ordering of various modes and prediction of the RD cost and bit cost to reduce the computational complexity of HEVC. Jimenez-Moreno et al. [24] developed an effective complexity control algorithm in a hierarchical manner. This method estimated all the parameters based on fly so that the encoder behavior can be adapted to the video content, the encoding configuration, and the target complexity over time. Zhang et al. [25] presented a novel CTU-level complexity control method, in which the complexity of each CTU is first estimated by a statistical model and is then used to optimize its depth range. Also, a complexity control scheme, including complexity estimation, complexity allocation, and PU adaption is designed for HEVC intra coding [26]. Xiong et al. [27] formed the pyramid motion divergence (PMD) by computing the optical flows on the down-sampled frames to obtain the motion information, followed by employing a k-nearest neighborhood like algorithm to determine whether the current CU needs to be split or not. Zhang et al. [28] presented a machine learning-based fast CU decision method. Firstly, the CU decision process is modeled as a three-level hierarchical decision problem. Then, several techniques, including an improved CU decision structure, a three-output classifier, and a sophisticated RD-complexity model, are developed to achieve a good trade-off between the complexity reduction and the RD maintenance. Zhu et al. [29] formulated the CU decision process as a cascaded multi-level classification task and solved it with the fuzzy support vector machine.

III. PROPOSED UNIMODAL MODEL-BASED INTER MODE DECISION (UMIMD) ALGORITHM FOR HEVC

A. UNIMODAL MODEL

Note that the above-mentioned prediction modes as exploited in the HEVC were designed for better adapting to different video contents for yielding high coding efficiency. To be more specific, the SKIP mode is more proper for the smooth region, the larger block-sized modes are more beneficial to those regions with more homogeneous content and with slow motion, the smaller block-sized modes are suitable for those regions involving fast motion, and the intra modes are most appropriate for the inhomogeneous regions with complex texture. Intuitively, it would be logical to skip the checking process for the smooth CTU with motionless or fairly slow motion. In other words, according to the evaluation of the smoothness and motion activity of the CTU, the checking process of those unlikely modes can be skipped to reduce the computational complexity of inter mode decision in HEVC. Further studies have shown that the smoothness and motion activity of the CTU can be accurately reflected by its RD cost value. That is, if a CTU is not smooth, not only the difference value (i.e., SSE) between the original pixels and the reconstructed pixels will be larger but also more bits will be required for coding the residual and mode information. In this case, a larger RD cost value as computed in (1) will be resulted. Hence, this relationship can be utilized on the design of fast mode decision algorithm as proposed in this paper.

Considering a PU is the basic processing unit for inter/intra prediction, the relationship between each PU mode and its yielded RD cost value is important to be investigated as follow. First, note that the total bits denoted as $R$ in (1) includes the bits required to code the prediction residual (denoted as $R_p$) and that for coding the MVs and mode information (denoted as $R_m$); thus, the RD cost function in (1) can be
expressed as

\[ J = SSE + \lambda \cdot (R_r + R_m). \]  

(3)

Note that when the size of chosen PU mode is small, the prediction tends to be more accurate. This leads to less normalized RD cost to PU size and a smaller residual, which requires less \( R_r \) bits to represent the residual information. However, all these are achieved at the expense of more prediction vectors need to be encoded. For example, in a given CU, only one MV needs to be transmitted for inter \( 2N \times 2N \) mode while four MVs are required to be represented and transmitted for inter \( N \times N \) mode. Hence, this will inevitably cost more bits on \( R_m \) due to the increment of mode information and thus increase the RD cost value. In other words, when the size of PU mode becomes smaller and smaller, the normalized bits to PU size consumed on \( R_r \) will be monotonically decreasing, while the normalized bits to PU size spent on \( R_m \) will be monotonically increasing.

**TABLE 1. Mode classes and their involved PU modes.**

<table>
<thead>
<tr>
<th>Class ( n )</th>
<th>Mode</th>
<th>0</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>SKIP_64 ( \times ) 64</td>
<td>64 ( \times ) 64</td>
<td>64 ( \times ) 32, 32 ( \times ) 64</td>
<td></td>
</tr>
<tr>
<td>Class ( n )</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>SKIP_32 ( \times ) 32</td>
<td>32 ( \times ) 32</td>
<td>32 ( \times ) 16, 16 ( \times ) 32</td>
<td></td>
</tr>
<tr>
<td>Class ( n )</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>SKIP_16 ( \times ) 16</td>
<td>16 ( \times ) 16</td>
<td>16 ( \times ) 8, 8 ( \times ) 16</td>
<td></td>
</tr>
<tr>
<td>Class ( n )</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mode</td>
<td>SKIP_8 ( \times ) 8</td>
<td>P8 ( \times ) 8, Intra</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Inspired by the above-mentioned trend, how to classify and group all the available HEVC’s modes in an effective way is a crucial step for the mode decision process. For that, all the HEVC’s PU modes are categorized into 11 *mode classes* in this paper and summarized in Table 1. This classification and grouping is according to the sizes of various PU modes. Note that although the SKIP mode and the inter \( 2N \times 2N \) (where \( N = 4, 8, 16, \) and \( 32 \) have the same size (e.g., SKIP_64 \( \times \) 64 and 64 \( \times \) 64), but they are more suitable for different kinds of regions. For example, the SKIP mode is beneficial to homogeneous regions and thus classified as a special mode class. Besides, the intra modes are grouped together P8 \( \times \) 8 (i.e., 8 \( \times \) 8, 8 \( \times \) 4, 4 \( \times \) 8, and 4 \( \times \) 4), since they are all more suitable on coding those regions with highly textures. Moreover, considering the fact that the asymmetric PU modes has lower possibility to be selected as the optimal mode, therefore, only the symmetric PU modes are considered in our work for conducting the mode decision. For ease of development and presentation, the class index \( n \) used in Table 1 is assigned to the PU modes according to their sizes, from large to small. We further denote the resulted bits of \( R_r \) and \( R_m \) incurred at class \( n \) as \( R_r(n) \) and \( R_m(n) \), respectively. When the size of PU mode is small (i.e., large \( n \)), less \( R_r(n) \), but more \( R_m(n) \), will be resulted. As a result, when the mode class changes from \( n \) to \( n+1 \), it becomes difficult to evaluate the net contribution or gain based on (3). To analyze this, let \( J(n) \) be the minimum RD cost resulted from the mode class \( n \), \( \Delta D_r(n) \) be the reduced RD cost resulted from the improvement of prediction accuracy, and \( \Delta D_m(n) \) be the increased RD cost for encoding more mode information. Consequently, the relationship between the RD cost of the mode classes \( n \) and \( n+1 \) can be described as

\[ J(n+1) = J(n) + \Delta D_r(n) + \Delta D_m(n). \]  

(4)

Clearly, if \( |\Delta D_r(n)| > |\Delta D_m(n)| \), the usage of a smaller block-sized PU mode (i.e., using mode class \( n+1 \) instead of \( n \)) will reduce the RD cost and thus be beneficial to improve the coding efficiency. On the contrary, if \( |\Delta D_r(n)| < |\Delta D_m(n)| \), the usage of smaller block-sized PU mode will increase the RD cost, which is obviously undesirable let alone it has incurred extra computational complexity. Further investigation has found that when \( |\Delta D_r(n)| < |\Delta D_m(n)| \), the residual has become quite small, which is hard to further reduced by using a smaller block-sized PU mode. In this situation, if the size of PU mode is further reduced (i.e., with a larger mode class number \( n \)), the change of \( |\Delta D_r(n)| \) value tends to become ignorable and with very limited coding gain to be yielded. On the other hand, the mode information will be substantially increased by the geometric progression, leading a much higher increment of \( |\Delta D_m(n)| \). Therefore, \( |\Delta D_r(n+1)| < |\Delta D_m(n+1)| \) will be usually satisfied when \( |\Delta D_r(n)| < |\Delta D_m(n)| \). Hence, a special relationship among the PU modes and their resulted RD costs can be observed and modeled as a *unimodal* model (i.e., with only one global minimum value incurred on the RD cost function).

**TABLE 2. Sequences for training and modeling.**

<table>
<thead>
<tr>
<th>Sequences</th>
<th>Resolutions</th>
<th>Sequences</th>
<th>Resolutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasketballPass</td>
<td>416×240</td>
<td>PartyScene</td>
<td>832×480</td>
</tr>
<tr>
<td>BlowingBubbles</td>
<td>416×240</td>
<td>Kimono</td>
<td>1920×1080</td>
</tr>
<tr>
<td>BasketballDrill</td>
<td>832×480</td>
<td>BQTerrace</td>
<td>1920×1080</td>
</tr>
</tbody>
</table>

To verify this unimodal model, extensive experiments have been conducted by applying the exhaustive mode decision in the HEVC to the video sequences as listed in Table 2. Fig. 3 illustrates the relationship between the RD cost of different block-sized PU modes \( J(n) \) and mode class \( n \) under various optimal PU modes. Note that \( J(n) \) for each mode class \( n \) is computed and normalized by its PU size when various PU modes are exploited as the optimal mode for the current CU. Take Fig. 3(d) as an example. When the SKIP_32 \( \times \) 32 (i.e., the mode class 3) is the optimal mode of CU32\( \times \)32, one can see that the global minimum point over the \( J(n) \) function is incurred at \( n = 3 \). Same interpretation can be applied to other sub-graph in Fig. 3 when their global minimum (i.e., the optimal mode) are incurred at various class \( n \) values. It is important to note that no matter which optimal PU mode is identified, the \( J(n) \) always has such unimodal property (i.e., with only one global minimum) in terms of the size of PU mode (i.e., the mode class \( n \)). By utilizing this unique...
property, the principle of the proposed UMIMD algorithm is described as follows.

The mode classes as listed in Table 1 can be sequentially checked according to the class index $n$, either from small to
large or from large to small, so that the optimal mode can be easily identified by checking when the RD cost starts to arise. When this happens, it means that the optimal mode (i.e., the global minimum point) has just incurred in the previous mode class checking. In this work, considering that the larger PU mode is more likely to be the optimal one while incurs lower computational complexity than the smaller PU mode, the mode classes as listed in Table 1 will be sequentially checked according to the class index n, from small to large. In other words, the RD costs of two consecutive mode classes will be sequentially compared to determine whether the mode class with higher class-index starts to yield a larger RD cost; i.e., checking whether \( J(n+1) > J(n) \). If so, the global minimum is considered found at class index \( n \), and the checking process of the remaining modes can be skipped, since the optimal mode has been identified. Consequently, the computational load can be significantly saved.

### B. INSTANT SKIP MODE TERMINATION SCHEME

The above-described fast mode decision scheme using the unimodal model (denoted as UM) needs to compute the RD cost of the next mode class \( n + 1 \) in order to determine whether the RD cost \( J(n+1) \) is greater than \( J(n) \). Obviously, this consumes extra computational complexity. To further reduce unnecessary computation of UM algorithm, an instant-termination scheme is desired to instantly determine whether the mode class under consideration is the optimal one. For that, extensive simulation experiments have been conducted to study the distribution of the optimal mode class presented in the HEVC by exploiting the exhaustive mode decision based on the video sequences as listed in Table 1 under the common test conditions [32], and the results are shown in Table 3.

This investigation clearly indicates that for the variable block-sized CUs, the mode classes 0, 3, 6, 9 (i.e., the SKIP modes with variable block sizes) have higher possibility to be the optimal one, especially for those sequences with motionless or slow-motion content. This is because large homogeneous, motionless or slow motion scenes are often encountered in the real-world video sequences while the SKIP mode is more suitable for coding such kind of scenes. On the other hand, the checking process of the SKIP mode has lower computational complexity, since the SKIP mode does not require to conduct ME operation; therefore, neither the MV/reference indices nor the texture residuals will be incurred for encoding and transmission (In fact, the MV is derived from the MVs of the spatial/temporal neighboring CUs). Therefore, if the SKIP mode can be determined instantly (i.e., the RD cost of the next mode class needs not to be computed), the computational complexity will be substantially reduced again. Since the motion activity of the CU is highly related to its RD cost value, an instant SKIP mode termination scheme is developed by evaluating whether the RD cost computed at the SKIP mode is below an empirically pre-determined threshold. More specifically, for the variable block-sized CUs represented by different depths within the current CTU (See Fig. 1), if the RD cost of the SKIP mode computed at the current depth is small enough (that is, below a threshold \( T \) corresponding to the current depth), then the current CU is quite likely smooth and/or with fairly slow motion. In this case, the SKIP mode is thus selected as the optimal mode, and the mode decision process proceeds to the next CU.

Now, the key question is how to find a robust value of the threshold for each depth. For the CTU as shown in Fig. 1, four thresholds, \( T_0 \), \( T_1 \), \( T_2 \), and \( T_3 \), are proposed in our mode decision algorithm for variable block-sized CUs that correspond to the depth levels 0, 1, 2, and 3, respectively. For example, to determine \( T_0 \) for CU \( 64 \times 64 \) (i.e., depth = 0), all the \( 64 \times 64 \) CUs from a set of commonly-used test sequences listed in Table 2 are employed to determine the most reliable value of the threshold \( T_0 \)—with a goal of achieving 95% degree of confidence. In other words, if the RD cost value of the SKIP mode computed at the \( 64 \times 64 \) CU (i.e., for the mode class 0, SKIP_64) is smaller than \( T_0 \), the SKIP_64 mode is assigned to the current \( 64 \times 64 \) as its optimal mode, and the probability that this SKIP_64 mode is indeed the optimal mode according to the exhaustive mode decision is not less than 0.95. The thresholds \( T_1 \), \( T_2 \), and \( T_3 \) are determined in a similar manner as performed for the \( T_0 \). Further consider that

<table>
<thead>
<tr>
<th>Sequences</th>
<th>BasketballPass</th>
<th>BlowingBubbles</th>
<th>BasketballDrill</th>
<th>PartyScene</th>
<th>Kimono</th>
<th>BQTTree</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class n</td>
<td>Mode</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>SKIP_64×64</td>
<td>18.40</td>
<td>8.55</td>
<td>26.05</td>
<td>19.51</td>
<td>26.96</td>
<td>47.84</td>
</tr>
<tr>
<td>1</td>
<td>64×64</td>
<td>4.96</td>
<td>3.73</td>
<td>14.64</td>
<td>3.33</td>
<td>14.21</td>
<td>8.21</td>
</tr>
<tr>
<td>2</td>
<td>64×32/32×64</td>
<td>2.85</td>
<td>4.06</td>
<td>4.12</td>
<td>3.05</td>
<td>5.29</td>
<td>4.89</td>
</tr>
<tr>
<td>3</td>
<td>SKIP_32×32</td>
<td>26.47</td>
<td>27.51</td>
<td>18.80</td>
<td>23.24</td>
<td>17.94</td>
<td>16.83</td>
</tr>
<tr>
<td>4</td>
<td>32×32</td>
<td>5.32</td>
<td>8.65</td>
<td>10.39</td>
<td>9.76</td>
<td>16.58</td>
<td>5.59</td>
</tr>
<tr>
<td>5</td>
<td>32×16/16×32</td>
<td>1.59</td>
<td>3.85</td>
<td>2.06</td>
<td>4.74</td>
<td>2.75</td>
<td>1.62</td>
</tr>
<tr>
<td>6</td>
<td>SKIP_16×16</td>
<td>18.83</td>
<td>18.07</td>
<td>8.58</td>
<td>12.54</td>
<td>6.36</td>
<td>6.41</td>
</tr>
<tr>
<td>7</td>
<td>16×16</td>
<td>9.28</td>
<td>12.00</td>
<td>8.77</td>
<td>11.53</td>
<td>6.93</td>
<td>3.59</td>
</tr>
<tr>
<td>8</td>
<td>16×8/8×16</td>
<td>1.57</td>
<td>2.07</td>
<td>1.11</td>
<td>2.63</td>
<td>0.42</td>
<td>0.65</td>
</tr>
<tr>
<td>9</td>
<td>SKIP_8×8</td>
<td>5.69</td>
<td>4.46</td>
<td>3.00</td>
<td>6.09</td>
<td>1.62</td>
<td>2.68</td>
</tr>
<tr>
<td>10</td>
<td>PS×8/Intra</td>
<td>5.04</td>
<td>7.05</td>
<td>2.48</td>
<td>3.58</td>
<td>0.94</td>
<td>1.71</td>
</tr>
</tbody>
</table>

| Table 3. Distribution of the optimal mode class in the HEVC (%) |
different QP values could yield various degrees of quantization of the video data, these thresholds are dependent on the QP, and their values for various QP are shown in Table 4. It has been observed that, approximately, there is an exponential relationship existing between the threshold values and various QP values. To mathematically model this relationship, a Matlab curve fitting tool, cftool, is applied to approximate an exponential function with the format as: \( T_i = a T_0^b \times Q^p \), where parameters \( a \) and \( b \) are to be determined based on the data points in Table 4. The resulted formulas are

- \( T_0 = 512.8026 e^{0.1884 \times QP} \),
- \( T_1 = 39.3489 e^{0.2206 \times QP} \),
- \( T_2 = 49.6989 e^{0.1820 \times QP} \),
- \( T_3 = 6.1716 e^{0.1937 \times QP} \).

### C. Algorithm Description of the Proposed UMIMD Algorithm

In summary, the proposed UMIMD algorithm for the current CTU can be described as follows:

1. For the CU_{64x64}, calculate the RD cost of the SKIP_{64} × 64 mode (i.e., the Class 0), \( J(0) \).
2. If \( J(0) < T_0 \), then the SKIP_{64} × 64 mode will be chosen as the optimal mode, and go to Step (7).
3. Calculate the RD cost of the inter 64 × 64 (i.e., the Class 1), \( J(1) \). If \( J(1) > T_0 \), choose SKIP_{64} × 64 (i.e., Class 0) as the optimal mode, and go to Step (7).
4. Calculate the RD cost of the inter 64 × 32 and the inter 32 × 64 (both from the Class 2), respectively. Select the one with the smaller RD cost as the mode 2 and set the corresponding RD cost as \( J(2) \). If \( J(2) > T_1 \), choose the inter 64 × 64 (i.e., the Class 1) as the optimal mode, and go to Step (7).
5. Divide the CU_{64x64} into four CU_{32x32} units. Calculate the RD cost of the SKIP_{32} × 32 (i.e., the Class 3) for each CU_{32x32} and denote them as \( J_i(3) \) (where \( i = 0, 1, 2, \) and 3), respectively. The RD cost of the CU_{64x64} computed at the Class 3 is calculated by summing them together; i.e., \( J(3) = \sum_{i=0}^{3} J_i(3) \). If \( J(3) > J(2) \), choose the mode 2 as the optimal mode, and go to Step (7).
6. If all the CU_{32x32} are checked, go to Step (7); otherwise, go to Step (6.1).
6.1 For the \( i \)-th CU_{32x32}, if \( J_i(3) < T_1 \), then the SKIP_{32} × 32 mode will be chosen as the optimal mode, and go to Step (6).
6.2 Calculate the RD cost of the inter 32 × 32 (i.e., the Class 4), \( J_4(4) \). If \( J_4(4) > J_i(3) \), choose the SKIP_{32} × 32 as the optimal mode, and go to Step (6).
6.3 Calculate the RD cost of the inter 32 × 16 and the inter 16 × 32 (both from the Class 5), respectively. Set the one with the smaller RD cost as the mode 5 and the corresponding RD cost as \( J_5(5) \). If \( J_5(5) > J_i(3) \), choose the inter 32 × 32 as the optimal mode, and go to Step (6).
6.4 Divide the current (i.e., \( i \)-th) CU_{32x32} into four CU_{16x16} units. Calculate the RD cost of the SKIP_{16} × 16 (i.e., the Class 6) for each CU_{16x16} and denote them as \( J_{ij}(6) \) (for \( j = 0, 1, 2, \) and 3), respectively. The RD cost of the current CU_{32x32} computed at the Class 6 is calculated by summing the computed values together; i.e., \( J(6) = \sum_{j=0}^{3} J_{ij}(6) \). If \( J(6) > J_5(5) \), choose the mode 5 as the optimal mode, and go to Step (6).
6.5 If all the CU_{16x16} are checked, go to Step (6); otherwise, go to Step (6.5.1).
6.5.1 For the (\( i, j \))-th CU_{16x16}, if \( J_{ij}(6) < T_2 \), then the SKIP_{16} × 16 mode will be chosen as the optimal mode, and go to Step (6.5).
6.5.2 Calculate the RD cost of the inter 16 × 16 (i.e., the Class 7), \( J_7(7) \). If \( J_7(7) > J_{ij}(6) \), choose the SKIP_{16} × 16 as the optimal mode, and go to Step (6.5).
6.5.3 Calculate the RD cost of the inter 16 × 8 and the inter 8 × 16 (both are from the Class 8), respectively. Set the one with the smaller RD cost as the mode 8 and the corresponding RD cost as \( J_8(8) \). If \( J_8(8) > J_{ij}(6) \), choose the inter 16 × 16 as the optimal mode, and go to Step (6.5).
6.5.4 Divide the current (i.e., (\( i, j, k \))-th) CU_{16x16} into four CU_{8x8} units. Calculate the RD cost of the SKIP_{8} × 8 (i.e., Class 9) for each CU_{8x8} and denote them as \( J_{ijk}(9) \) (for \( k = 0, 1, 2, \) and 3), respectively. The RD cost of the current CU_{16x16} computed at the Class 9 is calculated by summing them together; i.e., \( J(9) = \sum_{k=0}^{3} J_{ijk}(9) \). If \( J(9) > J_{ij}(8) \), choose the mode 8 as the optimal mode, and go to Step (6.5).
6.5.5 If all the CU_{8x8} are checked, go to Step (6.5); otherwise, go to Step (6.5.5.1).
6.5.5.1 For the (\( i, j, k \))-th CU_{8x8}, if \( J_{ijk}(9) < T_3 \), then the SKIP_{8} × 8 mode will be chosen as the optimal mode, and go to Step (6.5).
6.5.5.2 Calculate the RD cost of the P8 × 8 and the intra-prediction modes (i.e., the Class 10), respectively. Set the one with the smaller RD cost as the mode 10 and...
TABLE 5. Performance comparison between proposed UMIMD algorithm and recent works under the LD condition.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BDBR</td>
<td>TS</td>
<td>BDBR</td>
<td>TS</td>
<td>BDBR</td>
<td>TS</td>
</tr>
<tr>
<td>Class A</td>
<td>Traffic</td>
<td>2.0</td>
<td>41.3</td>
<td>1.0</td>
<td>54.7</td>
<td>2.0</td>
<td>41.7</td>
</tr>
<tr>
<td>2560×1600</td>
<td>PeopleOnStreet</td>
<td>2.2</td>
<td>19.9</td>
<td>0.4</td>
<td>32.5</td>
<td>1.5</td>
<td>28.0</td>
</tr>
<tr>
<td>Class B</td>
<td>Kinemon</td>
<td>0.3</td>
<td>27.8</td>
<td>0.8</td>
<td>38.0</td>
<td>0.8</td>
<td>47.9</td>
</tr>
<tr>
<td>1920×1080</td>
<td>ParkScene</td>
<td>0.9</td>
<td>28.9</td>
<td>0.8</td>
<td>40.5</td>
<td>1.1</td>
<td>48.3</td>
</tr>
<tr>
<td>Cactus</td>
<td>3.1</td>
<td>36.4</td>
<td>0.5</td>
<td>43.5</td>
<td>2.2</td>
<td>48.0</td>
<td>1.8</td>
</tr>
<tr>
<td>BasketballDrive</td>
<td>1.0</td>
<td>25.7</td>
<td>0.8</td>
<td>42.3</td>
<td>1.2</td>
<td>42.6</td>
<td>2.2</td>
</tr>
<tr>
<td>BQTerrace</td>
<td>1.8</td>
<td>32.2</td>
<td>1.3</td>
<td>42.7</td>
<td>0.3</td>
<td>47.5</td>
<td>2.1</td>
</tr>
<tr>
<td>Class C</td>
<td>BasketballDrill</td>
<td>1.3</td>
<td>38.5</td>
<td>1.9</td>
<td>44.0</td>
<td>2.0</td>
<td>37.0</td>
</tr>
<tr>
<td>832×480</td>
<td>BQMail</td>
<td>2.2</td>
<td>25.3</td>
<td>2.0</td>
<td>43.5</td>
<td>1.2</td>
<td>40.9</td>
</tr>
<tr>
<td>PartyScene</td>
<td>3.3</td>
<td>25.1</td>
<td>1.0</td>
<td>40.0</td>
<td>0.4</td>
<td>33.0</td>
<td>1.3</td>
</tr>
<tr>
<td>RaceHorses</td>
<td>1.1</td>
<td>14.0</td>
<td>1.0</td>
<td>34.1</td>
<td>1.1</td>
<td>26.1</td>
<td>2.0</td>
</tr>
<tr>
<td>Class D</td>
<td>BasketballPass</td>
<td>1.6</td>
<td>14.8</td>
<td>1.5</td>
<td>35.4</td>
<td>1.2</td>
<td>27.7</td>
</tr>
<tr>
<td>416×240</td>
<td>BQSquare</td>
<td>0.4</td>
<td>44.3</td>
<td>0.4</td>
<td>36.0</td>
<td>0.2</td>
<td>38.8</td>
</tr>
<tr>
<td>Blowingbubbles</td>
<td>2.4</td>
<td>22.0</td>
<td>2.2</td>
<td>39.7</td>
<td>0.3</td>
<td>31.5</td>
<td>0.9</td>
</tr>
<tr>
<td>RaceHorses</td>
<td>1.0</td>
<td>12.9</td>
<td>0.7</td>
<td>30.8</td>
<td>0.7</td>
<td>21.2</td>
<td>1.2</td>
</tr>
<tr>
<td>Class E</td>
<td>FourPeople</td>
<td>2.4</td>
<td>47.8</td>
<td>1.1</td>
<td>64.0</td>
<td>1.6</td>
<td>65.6</td>
</tr>
<tr>
<td>1280×720</td>
<td>Johnny</td>
<td>2.0</td>
<td>56.0</td>
<td>1.3</td>
<td>66.9</td>
<td>-0.3</td>
<td>73.9</td>
</tr>
<tr>
<td>KristenAndSara</td>
<td>2.8</td>
<td>48.5</td>
<td>1.4</td>
<td>60.9</td>
<td>0.5</td>
<td>69.6</td>
<td>2.3</td>
</tr>
<tr>
<td>Average</td>
<td>1.8</td>
<td>31.2</td>
<td>1.1</td>
<td>43.9</td>
<td>1.0</td>
<td>42.7</td>
<td>1.7</td>
</tr>
</tbody>
</table>

If $J_{ijk}(10) > J_{ijk}(9)$, choose the SKIP_8×8 as the optimal mode; otherwise, choose the mode 10 as the optimal mode. Go to Step (6.5.5).

7) Proceed to the mode decision of the next CTU and repeat the above-described steps.

IV. EXPERIMENTAL RESULTS

To evaluate the performance, the proposed fast mode decision, UMIMD, algorithm has been incorporated into the HEVC reference software (i.e., HM16.7) [33] and tested on multiple test sequences, covering a wide range of motion activities and resolutions. All the experiments are conducted on a computer with 3.6 GHz Intel i7 Core 4 processors, 8 GB memory, and Windows 7 operating system. The test conditions are set as follows:

1) Each test sequence is encoded using low delay (LD) and random access (RA) configuration defined in the common test conditions [32].
2) The maximum and minimum CU sizes are 64×64 and 8×8, respectively.
3) Four QPs, 22, 27, 32 and 37, respectively, are considered in our experiments.
4) The RDO is enabled, and the values of other parameters are set at their default, respectively.

In this work, the performance evaluation of the proposed UMIMD algorithm and five state-of-the-art methods (i.e., Shen et al. [12], Shen et al. [14], Ahn et al. [18], Zhang et al. [28], and Zhang et al. [21]) are compared under the LD and RA conditions [32], respectively. In our experiments, the objective measurement of the performance gain of each method is referenced to that of the exhaustive mode decision in the HEVC, measured in terms of the following two performance indexes: 1) the Bjontegaard delta bit rate (BDBR) [34] measures the averaged bit rate changes; 2) TS measures the averaged complexity reduction over four QPs, which can be computed as

$$TS = \frac{1}{4} \sum_{i=1}^{4} \frac{T_{HEVC}(QP_i) - T_{fast}(QP_i)}{T_{HEVC}(QP_i)} \times 100\%$$

where $T_{HEVC}(QP_i)$ and $T_{fast}(QP_i)$ are the encoding time resulted from the exhaustive mode decision in the HEVC and the corresponding fast mode decision methods under four $QP_i$ values as above-mentioned, respectively.

Tables 5 and 6 show the performance comparison under the LD and RA conditions, respectively. From the results as shown in Tables 5 and 6, it can be seen that compared with the outcomes resulted from the exhaustive mode decision in the HEVC, the proposed UMIMD algorithm constantly achieves, on average, 61.9% and 64.2% time saving, with only 1.7% and 2.1% increment in the BDBR bit rate for the LD and RA test conditions, respectively. Moreover, compared with Shen et al. [12], Shen et al. [14], Ahn et al. [18], Zhang et al. [28], and Zhang et al. [21], for the LD test condition, 30.7%, 18.0%, 19.2%, 14.0%, and 16.1% higher computational complexity reduction with similar coding efficiency measured in terms of BDBR bit rate changes are achieved by the proposed UMIMD algorithm, respectively; for the RA test conditions, the proposed UMIMD algorithm also yields 31.6%, 19.1%, 14.6%, 12.3%, and 9.3% more time saving while maintaining the similar coding performance, respectively. All these reveal that the proposed UMIMD algorithm consistently performs well and outperforms...
multiple state-of-the-art methods [12], [14], [18], [21], [28] under both LD and RA test conditions.

V. CONCLUSION

In this paper, a fast and accurate inter mode decision algorithm for the HEVC, called the unimodal model-based inter mode decision (UMIMD), is proposed. Based on the observation that a unimodal model exists between the size of different PU modes and their resulted RD costs for each quad-tree partitioned CTU, all the PU modes are classified into several mode classes according to the PU’s sizes and these classes are then properly ordered and sequentially checked so that the optimal mode can be early identified by checking when the RD cost starts to arise, indicating that the global minimum of the RD cost was just reached in the previous mode checking. Moreover, to further reduce the computational complexity, an instant SKIP mode termination scheme is developed based on the evaluation of the RD cost of the SKIP mode against a pre-determined threshold. Consequently, the encoding complexity reduction can be further achieved by skipping the computation of the RD cost of the next class when the SKIP mode is the optimal one. Experimental results have shown that the proposed UMIMD algorithm not only significantly reduces the computational complexity but also keeps almost the same coding efficiency, compared with the exhaustive mode decision in the HEVC. Furthermore, it has been demonstrated that the proposed UMIMD algorithm consistently outperforms multiple state-of-the-art methods.

REFERENCES


[34] G. Bjontegaard, Calculation of Average PSNR Differences Between RD-Curves, document VCEG-M33, 13th ITU-T SG16/Q6 VCEG Meeting, Austin, TX, USA, Apr. 2001.
ZHANGKAI NI received the M.Eng. degree in communication engineering from the School of Information Science and Engineering, Huaqiao University, Xiamen, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. He was a Research Engineer with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2017 to 2018. His current research interests include perceptual image processing, deep learning, and computer vision.

KAI-KUANG MA (S’80–M’84–SM’95–F’13) received the B.E. degree in electronic engineering from Chung Yuan Christian University, Zhongli, Taiwan, the M.S. degree in electrical engineering from Duke University, Durham, NC, USA, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, NC, USA.

From 1984 to 1992, he was with IBM Corporation, Kingston, NY, USA, and Research Triangle Park, NC, USA, where he focused on various DSP and very large-scale integration advanced product development. From 1992 to 1995, he was a Technical Staff Member with the Institute of Microelectronics, Singapore, where he is involved in digital video coding and the MPEG standards. He is currently a Full Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He served as the Singapore MPEG Chairman and the Head of Delegation, from 1997 to 2001. On the MPEG contributions, two fast motion estimation algorithms (Diamond Search and MVFAST) produced from his research group have been adopted by the MPEG-4 standard, as the reference core technology for fast motion estimation. He has published extensively and holds one U.S. patent on fast motion estimation algorithm. His research interests include the areas of digital image/video processing and computer vision, including digital image/video coding and standards, image/video segmentation, denoising, enhancement, interpolation, and super resolution. His research interests on computer vision include image matching and registration, scene analysis and recognition, and human–computer interaction.

He is a member of Sigma Xi and Eta Kappa Nu. He serves as a Technical Program Committee Member, as a Reviewer, and as the Session Chair for multiple IEEE international conferences. He is an Elected Member of three IEEE technical committees, namely, the Image and Multidimensional Signal Processing Committee, the Multimedia Communications Committee, and the Digital Signal Processing Committee. He was the General Chair of organizing a series of international standard meetings (MPEG and JPEG), JPEG2000, and MPEG-7 workshops held in Singapore, in 2001. He is a General Co-Chair of ISPACS2017, ASIPA2017, ACCV2016 Workshop, and VCIP-2013, a Technical Program Co-Chair of ICIP-2004, ISPACS-2007, IJH-MSP-2009, and PSIVT-2010, and the Area Chair of ACCV-2009 and ACCV-2010. He is the Chairman of the IEEE Signal Processing Singapore Chapter, from 2000 to 2002. He has been serving as an Editorial Board Member for several leading international journals in his research area, such as a Senior Area Editor for the IEEE Transactions on Image Processing, from 2016 to 2019, as an Editor for the IEEE Transactions on Communications, from 1997 to 2012, and as an Associate Editor for the IEEE Transactions on Multimedia, from 2002 to 2009, the International Journal of Image and Graphics, from 2003 to 2015, the Journal of Visual Communication and Image Representation, from 2005 to 2015, the IEEE Transactions on Image Processing, from 2007 to 2010, IEEE Signal Processing Letters, from 2014 to 2016, and the IEEE Transactions on Circuits and Systems for Video Technology, since 2015. He was elected as a Distinguished Lecturer of the IEEE Circuits and Systems Society, from 2008 to 2009.