Optimizing the Hierarchical Prediction and Coding in HEVC for Surveillance and Conference Videos With Background Modeling

Xianguo Zhang, *Member, IEEE*, Yonghong Tian, *Senior Member, IEEE*, Tiejun Huang, *Senior Member, IEEE*, Siwei Dong, and Wen Gao, *Fellow, IEEE*

***Abstract*— For the real-time and low-delay video surveillance and teleconferencing applications, the newly video coding standard HEVC can achieve much higher coding efficiency over H.264/AVC. However, we still argue that the hierarchical prediction structure in the HEVC low-delay encoder still does not fully utilize the special characteristics of surveillance and conference videos that are usually captured by stationary cameras. In this case, the background picture (G-picture), which is modeled from the original input frames, can be used to further improve the HEVC low-delay coding efficiency meanwhile reducing the complexity. Therefore, we propose an optimization method for the hierarchical prediction and coding in HEVC for these videos with background modeling. First, several experimental and theoretical analyses are conducted on how to utilize the G-picture to optimize the hierarchical prediction structure and hierarchical quantization. Following these results, we propose to encode the G-picture as the long-term reference frame to improve the background prediction, and then present a G-picture-based bit-allocation algorithm to increase the coding efficiency. Meanwhile, according to the proportions of background and foreground pixels in coding units (CUs), an adaptive speed-up algorithm is developed to classify each CU into different categories and then adopt different speed-up strategies to reduce the encoding complexity. To evaluate the performance, extensive experiments are performed on the HEVC test model. Results show our method can averagely save 39.09% bits and reduce the encoding complexity by 43.63% on surveillance videos, whereas those are 5.27% and 43.68% on conference videos.**

***Index Terms*— HEVC, hierarchical prediction, surveillance videos, conference videos, background modeling, CU classification.**

I. Introduction

IN RECENT years, video surveillance and teleconferencing systems are more and more widely used for safety and communication applications. For example, more than 5 million surveillance cameras were deployed in UK in 2012. If these cameras were all High-Definition (HD) ones and the generic video codecs such as H.264/AVC [1] were adopted to compress the videos, hundreds of Terabytes data would be produced per minute or thousands of Petabytes per month. Thus to realize real-time security monitoring as well as long-time archiving, there is a great demand for high-efficiency and low-complexity surveillance video coding methods. This is also true for video teleconferencing applications, since real-time and low-bitrate conference video coding can enable a feasible way to attend a video meeting from anywhere using mobile devices with limited bandwidth.

Intuitively, for surveillance and conference videos that are usually captured by stationary cameras, it is crucial to exploit their special characteristics (e.g., relatively fixed background in a period) for high-efficiency video coding [29]. Thus a reasonable approach is to compress foreground objects and background separately, naturally leading to object-based methods [2], [3]. However, the accurate automatic foreground segmentation is still a very challenging problem. Even in the recent works [4]–[7], it is also difficult to use only a few bits to encode both the object description and the prediction residuals. Therefore, hybrid-block-based video coding methods [8]–[13] are more widely utilized for surveillance and conference videos, most of which employ region-based or background-prediction-based techniques within the H.264/AVC coding framework.

Among the hybrid coding methods [8]–[13], our previous work [12], [29] and Paul et al. [13] achieved the relatively high performance, due to their significant improvements on the prediction efficiency of the exposed background regions (EBRs) with background modeling. An example of EBRs in conference video can be found in Fig. 1 (A similar example in surveillance video can be found in [29]). Clearly, we can exactly find the corresponding references in the background frame (called *G-picture* hereafter) for the circled EBRs, despite they cannot be found in the key and recent reference frames. Nevertheless, there are also some problems in [12], [13], and [29].

Firstly, except for [29], the G-picture was utilized to improve the coding efficiency in an improper way. In [12], it was only used to calculate the difference data from the current frame. After background subtraction, the dependency

Manuscript received May 27, 2014; revised August 19, 2014; accepted August 19, 2014. Date of publication August 26, 2014; date of current version September 11, 2014. This work was supported by the National Natural Science Foundation of China under Contract 61390515, Contract 61035001, and Contract 61121002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joan Serra-Sagrista. *(Corresponding author: Yonghong Tian.)*

The authors are with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: zxgvideo@gmail.com; yhtian@pku.edu.cn; tjhuang@pku.edu.cn; dosdong@pku.edu.cn; wgao@pku.edu.cn).

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

Digital Object Identifier 10.1109/TIP.2014.2352036

1057-7149 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

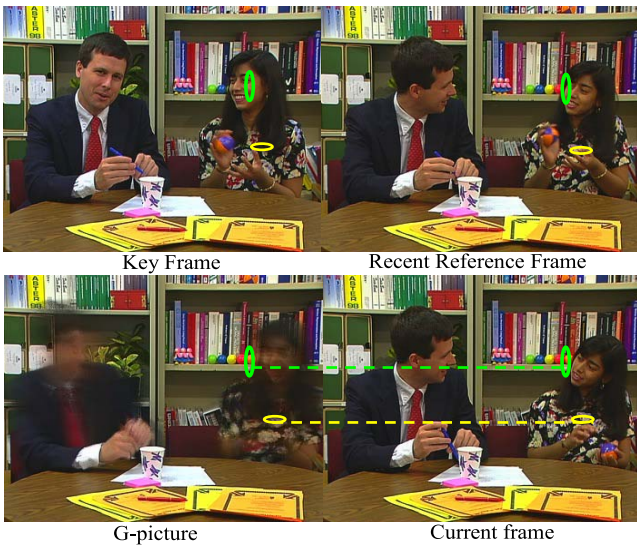

Fig. 1. An example of the "exposed background regions (EBRs)" in the current frame for conference video coding. For the circled exposed regions in the "current frame," we can only find the good reference in the G-picture rather than the "key frames" and the "recent reference frames." The lines only connect the similar regions.

among foreground pixels would be inevitably reduced. While in [13], the G-picture was only used as the second reference for each frame by replacing the original second reference picture. However, this would decrease the coding efficiency of foreground pixels since they usually needed to refer to the original second reference picture, especially for the pixels with large motion. Note that this problem has been successfully solved in our recent work [29] by introducing two novel background-based inter prediction modes, namely the background reference prediction (BRP) and the background difference prediction (BDP). Secondly, the quantization process in [12], [13], and [29] simply utilized a relatively fixed Quantization Parameter (QP) to quantize the G-picture, while ignoring the difference between the coding bitrates of the input frames with different proportions of moving objects. Thirdly, the encoding complexity in [12], [13], and [29] is not reduced or even higher than the used reference platform. In [29], for example, despite the selection between BRP and BDP for each macro-block improves the prediction efficiency, there is also some unavoidable complexity increase in the encoding time and memory. Finally, they all were based on the H.264/AVC framework. Thus if the G-picture was directly utilized in the latest video coding standard, we would not obtain as good results as in H.264/AVC.

More recently, the new-generation hybrid-block-based video coding standard, H.265/HEVC [14], has been developed by the Joint Collaborative Team on Video Coding (JCT-VC). By adopting various high-efficiency coding tools, HEVC can achieve much higher compression efficiency over H.264/AVC. In HEVC, the quad-tree picture partition for coding units (CUs) and a mass of intra-and-inter prediction patterns of prediction units (PUs) (shortly as *CU partitioning* and *PU pattern selection*) can significantly improve the coding efficiency, despite largely increasing the encoding complexity. Therefore, several recent works [16]–[18] were proposed to reduce the complexity of HEVC.

In the HM (HEVC Test Model [15]), the low-delay hierarchical prediction structure (called HPS for short) is possibly the best configuration for the real-time video surveillance and teleconferencing applications. Generally, HPS utilizes the hierarchical reference frame selection and hierarchical quantization for each short group of frames (referred to as HPS GOP) [21]. It always encodes the frame before an HPS GOP with a smaller QP (called more important frame, MIF, hereafter), and for each current frame, utilizes its previous frame and the MIFs in the three previous HPS GOPs as the four reference frames. However, the low-delay encoder with HPS still has not made full use of the special characteristics of these videos to further optimize the coding efficiency and reduce the complexity. That is, no clean background reference is provided to improve the coding efficiency of the huge amount of background pixels, consequently leading to greatly increase the overall coding bit-rate; while for the complexity, the mode decision process for the recursive CU partitioning and multiple PU candidate patterns is too complex to meet the low-complexity coding requirements in the real-time video applications.

To fix these problems, we conduct an analysis to compare the coding efficiency of HPS with the traditional reference frame selection and bit-allocation strategy. Results show that, the hierarchical reference selection and hierarchal quantization are two important components in HPS that contribute to the high-efficiency coding. Following this, we also carry out two sets of experimental and theoretical analyses on how to utilize background modeling to improve the coding efficiency of the two components while reducing the overall coding complexity of HPS. On one hand, experimental results show that, if replacing the fourth reference with the G-picture, HPS would be more likely to select the fourth reference frame. Meanwhile, for surveillance and conference videos, the G-picture should be quantized with a much smaller QP while some frames whose content are similar to the G-picture could be quantized with a larger QP so as to obtain better coding performance. On the other hand, by classifying the regions to be partitioned into different CU categories, we can see that the foreground CUs (FCUs), background CUs (BCUs) and hybrid foreground-background CUs (XCUs) always have different distributions of CU partitions, PU patterns and motion vector differences. This fact enlightens us to design an adaptive speed-up algorithm for different CU categories.

Motivated by these analytical results, this paper proposes a Background modeling based HPS Optimization (BHO) method for surveillance and conference videos. Basically, our BHO method consists of two key components, i.e., the G-picture-based HPS optimization algorithm and the adaptive speed-up algorithm based on CU classification. On one hand, the BHO improves the prediction efficiency of HPS by employing the intra-coded G-picture as the long-term reference for each frame in an HPS GOP and improves the coding efficiency of HPS by designing a G-picture-based bit-allocation optimization algorithm. In the bit-allocation process, the G-picture is adaptively quantized with a smaller QP and then a Background-similar HPS GOP detection algorithm is embedded to determine whether to adjust the QPs of

each HPS GOP. On the other hand, BHO adopts an adaptive speed-up algorithm using G-picture-based CU classification. That is, it classifies each CU into one category in {FCU, BCU, XCU} according to the difference between itself and the corresponding background data in the G-picture, and then adopts the individualized speed-up strategy for each category, including fast CU partitioning, PU pattern selection and motion estimation simplification. Because CU partitioning for each category can be early-terminated and there are much fewer candidate PU patterns and narrower search range, the encoding complexity can be remarkably reduced.

Extensive experiments are performed to evaluate the efficiency and complexity of BHO compared with the test model HM12.0, with the HEVC recommended low-delay configuration. The test sequences include ten surveillance videos (resolutions vary from CIF to HD) from the PKU-SVD-A dataset [30], [31] and eight conference videos that are widely used in the evaluation of HEVC and H.264/AVC. Results show that BHO can averagely save 39.09% bits and reduce the encoding complexity by 43.63% on the surveillance videos, while the results are 5.27% and 43.68% on the conference videos. Here the performance gain is larger for surveillance videos than conference videos, mainly because the latter generally contains a large proportion of approximately-stationary foreground objects (e.g., swaying heads, arms and bodies) and consequently it is difficult to generate a clean G-picture to predict the following frames.

The rest of this paper is organized as follows. The related works are briefly discussed in Section II. Following the experimental or theoretical analysis results in Section III, we present the BHO method in Section IV. Section V reports the experimental results. Finally, the paper is concluded in Section VI.

II. Related Works

This section firstly reviews the related works, and then discusses some techniques in HEVC which can be used for high-efficiency and low-complexity surveillance and conference video coding.

A. Surveillance and Conference Video Coding

As one of the most direct solutions for surveillance and conference videos, the object-based coding can be traced back to [2], [3]. Musmann et al. [2] proposed the object-oriented-analysis-synthesis coding method, in which each video was coded with motion and shape of objects, color information and prediction residuals. Using MPEG-4 object representation techniques, Francois et al. [3] proposed to encode videos based on the accurate foreground region segmentation. To achieve higher coding efficiency for surveillance and conference videos, Vetro et al. [4] further proposed an approach to code the segmented foreground objects, whereas neglecting the background variations. Nevertheless, such a processing severely degraded the coding results in terms of objective quality metrics (e.g. PSNR). To solve the problem, Babu et al. [5] and Hakeem et al. [6] tried to encode the background residuals in the hybrid block-based coding framework. In a similar way, they also proposed to encode the object representation difference together with the object prediction residuals between adjacent frames. Afterwards, Venkatraman et al. [7] utilized the direct and transform-based compressive sensing information to represent the sparse signal of the residual object error. Overall speaking, the accurate foreground segmentation, low-cost object representation and high-efficiency foreground residual coding are three main challenges that object-based coding needs to deal with.

Instead, hybrid block-based methods encoded each picture block by block in the traditional hybrid coding framework. These methods could be classified into two categories, i.e., region-based coding and background prediction based coding. Among them, region-based methods aimed at achieving better subjective quality of foreground regions with low coding complexity. For example, the method in [8] used much more bits to encode foreground regions. However, the objective rate-distortion (RD) results of these methods are usually not the optimal. Instead, background-prediction based methods (see [9]–[13]) attempted to improve the objective compression efficiency by utilizing one background picture as the reference for the following pictures. The underlying assumption was that in surveillance and conference videos, there might be one background picture that kept unchanged for a long time. Following this idea, Chen et al. [9] utilized some "key frames," which could well represent the video scene in a given period, as the background picture.

However, there are still some "*exposed background regions*" (EBRs) that may appear in the current frame but are covered by objects in the recent reference frames or the key frame. As a result, it is impossible to improve the coding efficiency of these EBRs by using the key frame as the background. To address this problem, several background modeling based methods were proposed in [10]–[13]. Both [10] and [11] made use of the reconstructed pictures to model the background. Although it is very efficient, the reconstructed pictures could not guarantee the quality of the generated background due to the quantization loss, especially in the case of low-bit-rate video coding. In addition, the background modeling process in [10] and [11] would be embedded in the video decoder, leading to the increase of the decoding complexity. Therefore, our previous work [12] and Paul et al. [13] proposed to utilize the background picture that was modeled from the original input frames as the reference for more efficient background prediction. More recently, we proposed a Background-Modeling based Adaptive Prediction (BMAP) method for surveillance video coding [29]. Its basic idea is to adaptively adopt different prediction modes for each macro-block according to the block classification results. Experimental results show that BMAP can achieve twice the compression ratio on surveillance videos as H.264/AVC High Profile, with a slightly additional encoding complexity.

B. Techniques in HEVC That Can be Potentially Used for Surveillance and Conference Video Coding

It is reasonable to employ the HEVC for more efficient surveillance and conference video coding since it introduces

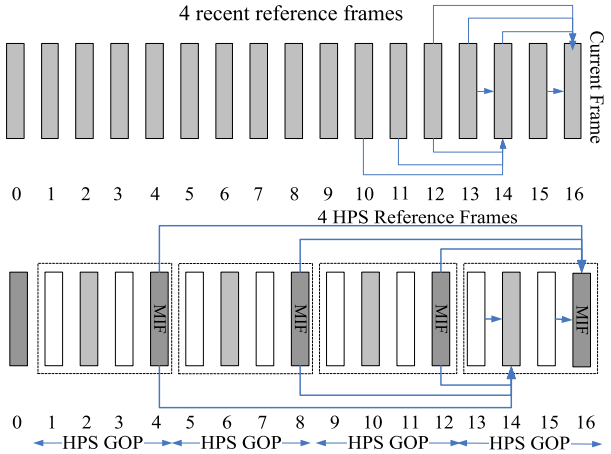

Fig. 2. Prediction structures with 4 recent reference frames and 4 HPS reference frames for low-delay coding. Reference frames of 14-th and 16-th frames are shown. Darker pictures are with smaller QPs.

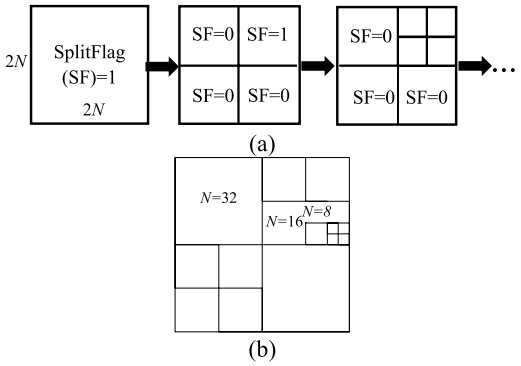

Fig. 3. (a) CU partitioning process; (b) one possible result of CU partition for a 128×128 region.

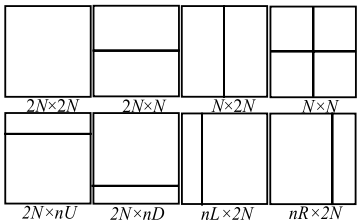

Fig. 4. PU patterns of the inter prediction.

many efficient tools, such as the quad-tree coding structure and hierarchical prediction reference. Among all the recommended configurations in HEVC (e.g., random access, low-delay and only intra), the low-delay one is probably most applicable to compress the real-time surveillance and conference videos, mainly because it adopts the low-delay HPS as the encoder-optimization tool without backward prediction reference.

Different from the low-delay prediction structure in H.264/AVC [19], [20] that utilizes the recent n pictures as the candidate references, the low-delay HPS predicts each current frame using the four non-adjacent forward reference frames. The frames include its previous frame and the last pictures of three previous GOPs (group of pictures). Moreover, the middle and last frames in each low-delay HPS GOP are encoded with smaller QPs. In this paper, the last picture in a GOP is referred to as *more important reference frame* (MIF). Fig. 2(a) shows the traditional recent-reference prediction structure, while Fig. 2(b) shows the low-delay HPS, where each current frame is predicted by its previous frame and three previous MIFs. Note that with the long-distance and finer-quantized MIFs as the most reference frames, the low-delay HPS may significantly save coding bits for surveillance and conference videos. Nevertheless, the long-time static background is still not fully utilized for better prediction efficiency in the low-delay HPS. Thus in this paper, we will explore how to utilize the G-picture to optimize the low-delay HPS.

Beside the coding efficiency issue, the increase of the encoding complexity produced by quad-tree CU partition for CTUs (coding tree units) is another main obstacle to the wide deployment of the low-delay encoders in surveillance cameras and video telephones. This is due to the relatively large number of the corresponding intra-and-inter prediction patterns of PUs for each CU. In quad-tree coding, CU is a squared unit composed of one $2N\times2N$ luma sample block and two $N \times N$ chroma sample blocks. In the general test scenario of HM, the largest CU size is often set to 64×64 (i.e., N=32) while the smallest is 8×8 (i.e., N=4). As for the candidate PU patterns, there are symmetric motion partitions of $2N\times2N$, $2N \times N$, $N\times2N$, $N \times N$ and asymmetric motion partitions (AMP) of $2N\times nU$, $2N\times nD$, $nL\times2N$ and $nR\times2N$. Fig. 3(a) and (b) show the recursive partitioning process and one possible partition result of a 128×128 region, respectively. Fig. 4 represents the available PU patterns for each CU. Note that each CTU usually has different combinations of CUs with different partition depths, while each CU has several kinds of candidate PU prediction patterns. As a result, a mode decision process should be used to determine the best CU partition and the optimal PU patterns. This will result in a remarkable increase of the total encoding time. To reduce the complexity of the mode decision, HM has adopted some tools to optimize the encoder, including the fast encoder decision [16], fast decision for merging RD cost [17], and fast transform skipping [18]. In spite of these efforts, it is still necessary to further reduce the encoding complexity for surveillance and conference videos via encoder optimization. Towards this end, this paper proposes to use the G-picture to implement CU classification and then adaptively speed up the encoding process.

III. Methodology for Optimization

In this section, several theoretical and experimental analyses are carried out to investigate the ways of utilizing background modeling to improve the coding efficiency and reduce the encoding complexity. In Part A, we analyze how to utilize the G-picture to improve the coding efficiency of the low-delay HPS for surveillance and conference videos; while in Part B, we present the distributions of prediction information for different CU categories and analyze how to speed up the encoding process. The experiments are conducted on the HM 12.0 low-delay main profile with a video dataset including four surveillance videos and four conference videos (as shown in Fig. 5). The G-picture generation method and test sequences will be described in details in Section V.

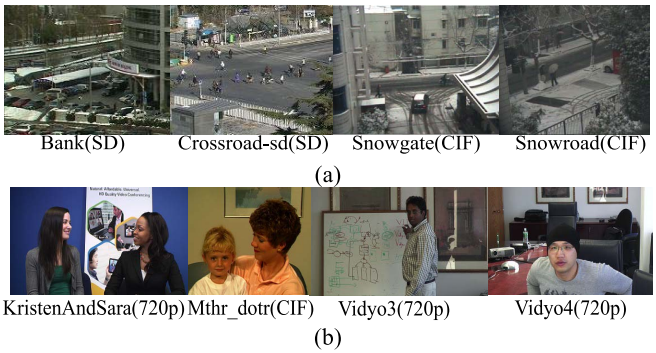

Fig. 5. The dataset for the experimental analyses. (a) Surveillance videos. (b) Conference videos.

TABLE I
BD-RATE COMPARISON RESULTS OF HR VS. 4RF AND HPS VS. HR

Surveillance Videos	HR vs. 4RF	HPS vs. HR	Conference Videos	HR vs. 4RF	HPS vs. HR
Bank-sd	-22.38%	-3.28%	Kristen&Sara-720p	-18.36%	-4.00%
Crossroad-sd	-17.03%	-2.90%	Mthr_dotr-cif	-14.04%	-3.45%
Snowgate-cif	-27.78%	-8.57%	Vidyo3-720p	-12.05%	-2.86%
Snowroad-cif	-27.26%	-6.94%	Vidyo4-720p	-14.38%	-4.49%
Average	**-23.61%**	**-5.42%**	**Average**	**-14.71%**	**-3.70%**

A. How to Optimize the Efficiency of Low-delay HPS

Typically, the low-delay HPS of HM contains two key components: (1) Hierarchical Reference (HR): Predicting each current frame with the previous frame and the MIFs in three previous HPS GOPs as the references; (2) Hierarchical Quantization (HQ): The QP of each MIF equals to that of its neighboring picture minus 2, while the QP of the middle picture in HPS GOP equals to that of the MIF plus 1. Intuitively, the efficiency of HR can be evaluated by the BD rate [27] between the HM only using HR and the HM using the 4 recent reference frames (namely 4RF, a HM encoder with reference indexes like $\{-1, -2, -3, -4\}$). Meanwhile, the efficiency of HQ can be evaluated by the BD rate between the HM using both HR and HQ and the HM using only HR. Thus we conduct a set of experiments to analyze the bit savings of HR and HQ with QP values in {22, 27, 32, 37}, on HM12.0 with the recommended low-delay configurations [26].

As shown in Table I, HR can averagely save 23.61% bits for surveillance videos and 14.71% for conference videos over the traditional 4RF; after using HQ, 5.42% and 3.70% additional bit savings are obtained respectively for these videos. Overall, the results confirm that the low-delay HPS can significantly improve the coding efficiency for both surveillance and conference videos. Therefore, we will conduct more experimental and theoretical analyses in A.1 and A.2 so as to find the possible ways that exploit the G-picture to optimize HR and HQ respectively.

1) Experimental Analysis: How to Optimize HR: Firstly, experimental analysis is conducted on the distribution of the reference frames used in the process of surveillance and conference video coding. Results in Fig. 6(a) show that the fourth reference frame only takes a very small percentage. However, if we replace the fourth reference frame by the G-picture as the long-term reference, where the G-picture is

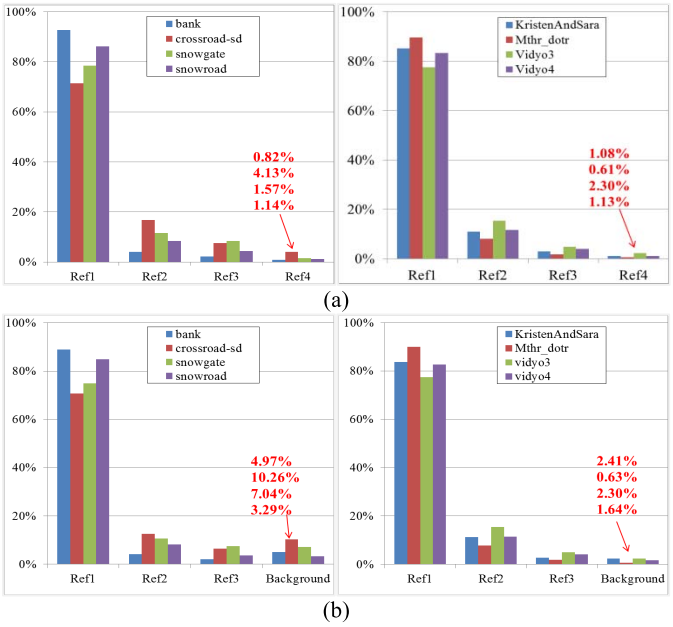

Fig. 6. The distribution of reference frames of the low-delay HPS. (a) Without the long-term reference. (b) With the long-term reference.

generated with the low-complexity running average algorithm (ref. to Sec IV-A), we can obtain different findings from Fig. 6(b): (1) The G-picture is selected 2∼6 times more than the original fourth reference on the surveillance videos and 1∼2 times on the conference videos; (2) The second and third reference frames still take very large proportions, and thus cannot be neglected. Therefore, we can conclude that the G-picture plays a more important role than the original fourth reference frame in the low-delay HPS. Naturally, we can optimize HR by background-based prediction. That is, the G-picture is utilized to replace the fourth reference frame and served as the long-term reference in the four reference frames.

Note that the probability of selecting the G-picture as the fourth reference frame is much larger on surveillance videos than that on conference videos. This is because a conference video usually has a much smaller proportion of the EBRs. Our experimental results in Section V have also verified that, the performance gain on surveillance videos is much larger than that on conference videos.

2) Theoretical Analysis: How to Optimize HQ: As discussed above, the G-picture should be used as the long-term reference in the low-delay HPS. Naturally, a new hierarchical quantization method based on the G-picture should be specially designed to improve the coding efficiency. Theoretically, the *Lagrange* RDO theory can be used to evaluate the RD cost J by $J = D+\lambda R$, where D measures the picture quality of the reconstructed video with respect to the original video, R measures the bits to encode the reconstructed video, and λ is the *Lagrange* multiplier related to QP. Given n input frames, let $\Psi(I_j, q)$ be the picture-level RD cost of coding the j-th picture I_j with QP equal to q, then J can be calculated from a CU-level RD cost function Γ for any I_j's k-th CU $I_{j,k}$ by [23]

$$J = \sum_{j}^{n} \Psi(I_j, q) = \sum_{j}^{n} \sum_{k} \Gamma\left(q, I_{j,k}, P_{j,k,q}, V_{j,k,q}\right) \quad (1)$$

where $P_{j,k,q}$ is the predicted data of $I_{j,k}$ from the picture quantized with q, and $V_{j,k,q}$ is the corresponding motion vectors.

When encoding I_i with a smaller q' and still using q to encode $I_1 \sim I_{i-1}$ and $I_{i+1} \sim I_n$, we suppose that the smaller-QP quantized I_i can provide better reference for the following x pictures ($I_{i+1} \sim I_{i+x}$). Then we suppose that in I_{i+1}, totally m_{i+1} CUs with indexes of $d(i+1, 1) \sim d(i+1, m_{i+1})$ can get better reference from I_i, while the other p_{i+1} CUs with indexes of $s(i+1, 1) \sim s(i+1, p_{i+1})$ cannot. Similarly, any picture I_{i+t} among $I_{i+2} \sim I_{i+x}$ has m_{i+t} better predicted CUs indexed by $d(i+t, 1) \sim d(i+t, m_{i+t})$ and other p_{i+t} CUs indexed by $s(i+t, 1) \sim s(i+t, p_{i+t})$. Then, the new RD cost J' satisfies

$$J' = \mathrm{K}_1 + \mathrm{K}_2 + \mathrm{K}_3 + \mathrm{K}_4 + \mathrm{K}_5,$$

where

$$\begin{aligned}
\mathrm{K}_1 &= \sum_{j=1}^{i-1} \Psi(I_j, q), \quad \mathrm{K}_2 = \Psi(I_i, q'), \\
\mathrm{K}_3 &= \sum_{j=i+1}^{i+x} \sum_{k=1}^{p_j} \Gamma\left(q, I_{j,s(j,k)}, P_{j,s(j,k),q}, V_{j,s(j,k),q}\right), \\
\mathrm{K}_4 &= \sum_{j=i+1}^{i+x} \sum_{k=1}^{m_j} \Gamma\left(q, I_{j,d(j,k)}, P_{j,d(j,k),q'}, V_{j,d(j,k),q'}\right), \\
\mathrm{K}_5 &= \sum_{j=i+x+1}^{n} \Psi(I_j, q). \qquad (2)
\end{aligned}$$

In Eq. 2, the first term K_1 is the total RD cost of pictures before I_i, K_2 is the RD cost by using q' to encode I_i, K_3 is the total RD cost of $I_{i+1} \sim I_{i+x}$'s CUs which cannot get the better reference, K_4 is the total RD cost of all the CUs which can get the better reference from the reconstructed result of I_i, and K_5 is the RD cost for the pictures after I_{i+x}.

By comparing Eq. 1 and Eq. 2, we can find that J also has K_1, K_3 and K_5, while the difference between J and J' includes the RD cost of encoding I_i and the cost for the CUs which can get better reference. Therefore, we can rewrite J by

$$\begin{aligned}
J &= \mathrm{K}_1 + \mathrm{A} + \mathrm{K}_3 + \mathrm{B} + \mathrm{K}_5 \qquad (3) \\
\mathrm{A} &= \Psi(I_i, q), \\
\mathrm{B} &= \sum_{j=i+1}^{i+x} \sum_{k=1}^{m_j} \Gamma\left(q, I_{j,d(j,k)}, P_{j,d(j,k),q}, V_{j,d(j,k),q}\right) \qquad (4)
\end{aligned}$$

Here, A represents the original RD cost of using q to encode I_i, and B denotes the original RD cost of coding the CUs which will have better reference in J'.

By subtracting Eq. 2 from Eq. 3, we have

$$J - J' = (\mathrm{B} - \mathrm{K}_4) - (\mathrm{K}_2 - \mathrm{A}) \qquad (5)$$

Since each $P_{j,d(j,k),q'}$ in K_4 is the data predicted from the smaller-QP quantized I_i, it has less quantization loss than $P_{j,d(j,k),q}$ ($j=i+1 \sim i+x$, $k=1 \sim m_j$) in B. As a result, K_4 has a better coding result for each $I_{j,d(j,k)}$. Thus the following inequality is satisfied

$$\begin{aligned}
&\mathrm{B} - \mathrm{K}_4 \\
&= \sum_{j=i+1}^{i+x} \sum_{k=1}^{m_j} \left(\begin{array}{l} \Gamma\left(q, I_{j,d(j,k)}, P_{j,d(j,k),q}, V_{j,d(j,k),q}\right) \\ -\Gamma\left(q, I_{j,d(j,k)}, P_{j,d(j,k),q'}, V_{j,d(j,k),q'}\right) \end{array} \right) \\
&> 0 \qquad (6)
\end{aligned}$$

Intuitively, $\mathrm{B}-\mathrm{K}_4$ denotes the decrease of the RD cost for the pictures following I_i, whereas $\mathrm{K}_2-\mathrm{A}=\Psi(I_j, q)-\Psi(I_j, q')$ in Eq. 5 represents the RD cost change caused by utilizing a smaller QP q' to quantize the input frame I_i. Consequently, $\mathrm{K}_2-\mathrm{A}$ is not surely larger or smaller than 0, and thus $J - J'=(\mathrm{B}-\mathrm{K}_4)-(\mathrm{K}_2-\mathrm{A})$ is also not guaranteed to be larger than 0. However, considering that some pictures quantized with a smaller QP might produce the better reference for a wide range of CUs in a lot of the following pictures (i.e., x and $m_{i+1} \sim \mathrm{m}_{i+x}$ are very large), we have

$$\begin{aligned}
&If\ J'\ has\ large\ x\ (\text{i.e., a large range } m_{i+1} \sim m_x) \\
&such\ that\ \mathrm{B} - \mathrm{K}_4 > \mathrm{K}_2 - \mathrm{A}, \\
&Then\ J - J' > 0. \qquad (7)
\end{aligned}$$

The derivation procedure of $J - J'$ can be explained by Fig. 7. As is shown, $J - J'$ is just the result of subtracting $(\mathrm{B}-\mathrm{K}_4)$, which is produced by using the smaller-QP quantized I_i as the reference for the following pictures ($I_{i+1} \sim I_{i+x}$), from the possible RD cost increase $(\mathrm{K}_2-\mathrm{A})$ produced by using a smaller QP to encode I_i. If the x and $\mathrm{B}-\mathrm{K}_4$ are large

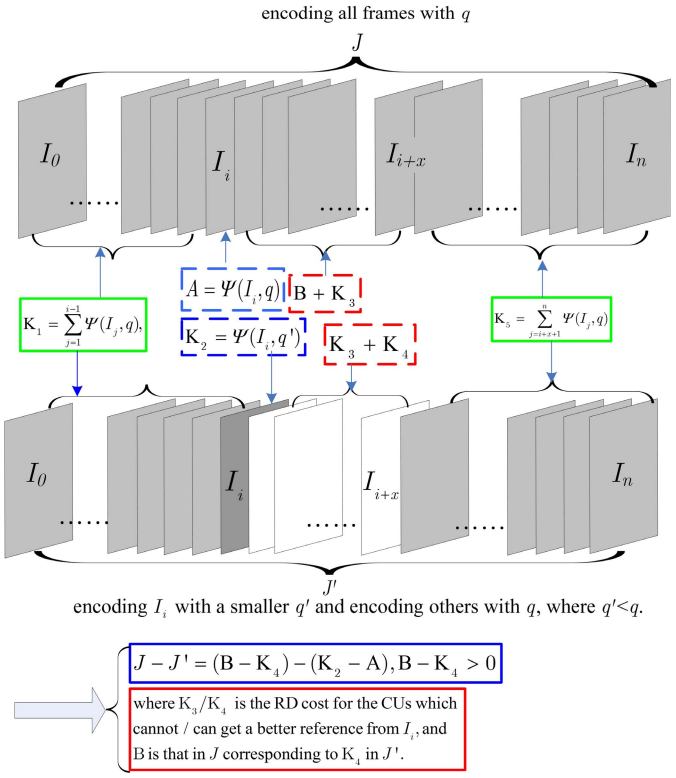

Fig. 7. The RD cost of J and J'. On the "white" pictures, the RD cost in J' is reduced and $\mathrm{B}-\mathrm{K}_2 > 0$. This is because $I_{i+1} \sim I_{i+x}$ have a better prediction reference from the reconstructed result of the smaller-QP quantized I_i. For the blackest picture I_i, a smaller QP is utilized in the quantization, so $\mathrm{K}_2-\mathrm{A}$ is not surely larger or smaller than 0.

enough (that is, the smaller-QP quantized I_i can provide good reference for a large number of the following frames), J will be larger than J' and a RD cost reduction can be obtained.

Notice that the condition part in Eq. 7 might be satisfied in surveillance and conference video coding. In these videos, there are usually some pictures that have lots of pixels similar to those in a large number of the following pictures (most of them are the long-time unchanged background pixels). Therefore, when such a picture is quantized with a smaller QP, the following pictures and CUs will obtain the better prediction reference data (i.e., x and $m_{i+1}{\sim}m_{i+x}$ can be much larger in such a case). This will make $\mathrm{B}-\mathrm{K}_4$ large enough to exceed $\mathrm{K}_2-\mathrm{A}$. In particular, when the G-picture is encoded into stream as a long-term reference, the values of x and $m_{i+1}{\sim}m_{i+x}$ will be very large so as to make $(\mathrm{B}-\mathrm{K}_4)$ much larger than $(\mathrm{K}_2-\mathrm{A})$, because nearly each of the following pictures has lots of background pixels similar to those in the G-picture. In such a case, the q' for the G-picture can even be much smaller, because $\mathrm{B}-\mathrm{K}_4$ will be much larger than $\mathrm{K}_2-\mathrm{A}$.

From the analysis, we can derive one conclusion: *The pictures, if they would be frequently selected as the reference for the following pictures, should be quantized with smaller QPs than the other frames*. By extending this conclusion to surveillance and conference video coding, we have the following deduction: *The G-picture, which is frequently used as the long- term reference, should be quantized with a much smaller QP*.

As mentioned above, the prerequisite of quantizing the G-picture with a small QP is that there are a large number of the pictures that have lots of pixels similar to those in the G-picture. With loss of generality, such groups of pictures (GOPs) are called as *Background-similar HPS GOPs* (BGOPs) in this study, while the other GOPs are thus referred to as *Normal HPS GOPs* (NGOPs). In our method, NGOPs can still follow the same quantization strategy in the HM low-delay HPS encoder, since the G-picture cannot provide significantly better reference for frames in these NGOPs. Thus the remaining problem is how to quantize BGOPs in a better way so as to save more encoding bits while remaining almost the same or even better picture quality of the reconstructed frames. Note that how to detect BGOPs will be discussed in Section III-B.1.

Typically, for a frame I_i in a BGOP, we can have two quantization strategies: 1) using the same QP (denoted by q) as that used for the corresponding frame in an NGOP by the HM low-delay HPS encoder, and 2) using another QP (denoted by q') where $q'{\neq}q$ so that a better RD performance can be obtained. Similarly, let J and J' denote the RD costs for the two strategies, respectively. Moreover, let Bg denote the corresponding G-picture. Then we can follow the similar derivation procedure as above to compare the difference between J and J': Firstly, because Bg can provide good reference for frames in the BGOP, thus no matter whether $q' > q$ or $q' < q$, we have $\mathrm{B}-\mathrm{K}_4 \approx 0$. Secondly, according to Eq. 5, in order to make $J - J' > 0$, we should make $\mathrm{K}_2-\mathrm{A}<0$, or equivalently $\mathrm{K}_2 <\mathrm{A}$. Note that here $\mathrm{K}_2 < \mathrm{A}$ indicates $q' > q$, because for a BGOP frame that can be well predicted by Bg, decreasing its QP will produce less improvement in quality distortion but more increase in coding bits. Therefore, we can derive the following conclusion for BGOPs: *When using the G-picture as the long-term reference, any frame in a BGOP can be quantized with a larger QP than that used for the corresponding frame in an NGOP*.

The corresponding quantization algorithm for BGOPs will be described in Section IV-B.2.

B. How to Speed Up the Coding With CU Classification

To investigate the potential encoding complexity optimization strategies, several experiments are carried out to analyze how to early terminate CU partitioning, select the best PU patterns and simplify motion estimation (ME). These analyses are based on the statistical distributions of CU sizes, PU patterns and MVDs on BCUs, FCUs and XCUs separately. Basically, each input CU is classified according to how many basic $4{\times}4$ units (shortly BUs) in the current CU belong to foreground units. Let $T(b)$ denote the type of an input BU b, $T(b) \in \{B,F\}$, $b_{i,j}$ denote the pixel value at row i and column j in the BU b, $Bg_{i,j}(b)$ be the corresponding pixel value in the G-picture, and then $T(b)$ is

$$T(b) = \begin{cases} B, & \text{if } \sum_{i=1}^{4}\sum_{j=1}^{4} \left|b_{i,j} - Bg_{i,j}(b)\right| \le \alpha; \\ F, & \text{Otherwise} \end{cases} \tag{8}$$

where α is a predefined threshold (80 is used in our experiment). This equation shows, the current BU is judged as background BU B or foreground BU F according to the sum of the difference between itself and the corresponding background data.

After identifying all the BUs in the current CU (denoted by c), we can then determine the CU's category $Class(c)$ by calculating the proportion of its foreground BUs. Let $b(i)$ be the i-th BU in c with the size of $2N{\times}2N$, then this process can be expressed as

$$Class(c) = \begin{cases} FCU, & \text{if } 4 \times \|\{i|T(b(i)) = F\}\|/N^2 > \delta; \\ XCU, & \text{if } \delta \ge 4 \times \|\{i|T(b(i)) = F\}\|/N^2 > \varepsilon; \\ BCU, & \text{if } 4 \times \|\{i|T(b(i)) = F\}\|/N^2 \le \varepsilon. \end{cases} \tag{9}$$

where δ is practically set to 0.5 and ε is 0.0625, $||X||$ represents the size of a set X. This equation shows that, if the proportion of the foreground BUs in a CU is no more than ε, then this CU will be categorized as BCU; if the proportion is more than δ, then it is an FCU; otherwise, it will be an XCU.

1) Analysis of CU Partitioning Termination: Before coding a $2N{\times}2N$ region, the HEVC encoder often needs to determine whether this region should be encoded as a whole $2N{\times}2N$ CU or recursively encoded in the form of four separate parts. This process is very time-consuming because the encoder usually makes the decision by recursively calculating the RD cost for each kind of partitions. To address this problem, some CU partitioning termination methods (see [16]) are integrated in the HM to early terminate the further partition. However, for surveillance and conference videos, the CU partitioning termination can be sped up to a greater extent by utilizing the long-time static background. In this paper, we firstly

TABLE II
THE PROPORTION OF THE FURTHER-PARTITIONED POTENTIAL BCUS, XCUS, FCUS, AND PURE BACKGROUND CUS

	Size	BCU	XCU	FCU	Pure-background CUs
Surveillance Videos	64×64(N=32)	12.08%	63.82%	93.02%	1.61%
	32×32(N=16)	7.60%	47.81%	35.27%	0.90%
	16×16(N=8)	1.38%	20.95%	25.50%	1.13%
Conference Videos	64×64(N=32)	4.17%	29.10%	65.91%	0.61%
	32×32(N=16)	4.56%	23.72%	33.84%	1.08%
	16×16(N=8)	1.61%	6.35%	8.66%	1.32%

TABLE III
THE DISTRIBUTION OF PU PATTERNS OF BCUS, XCUS AND FCUS RESPECTIVELY ON SURVEILLANCE AND CONFERENCE VIDEOS

	CUs	N=	2N×2N	2N×N&N×2N&N×N	AMP
Average results on Surveillance Videos	BCU	32	98.81%	0.87%	0.32%
		16	98.82%	0.53%	0.64%
		8	90.18%	5.77%	4.05%
		4	88.27%	7.69%	4.04%
	XCU	ALL	94.45%	3.28%	1.69%
	FCU	ALL	90.30%	5.72%	3.36%
Average results on Conference Videos	BCU	32	99.10%	0.74%	0.16%
		16	96.85%	1.78%	1.37%
		8	92.03%	4.27%	3.70%
		4	82.50%	10.68%	6.82%
	XCU	ALL	89.04%	7.63%	2.29%
	FCU	ALL	80.25%	12.90%	4.53%

regard each input region as a potential CU. Then using Eq. 9, each potential CU will be divided into BCU, XCU or FCU. Table II shows the proportions of BCUs/XCUs/FCUs that are further partitioned at different sizes. We can see that on surveillance videos, only 12.08/7.60/1.38% of the potential BCUs with N=32/16/8 will be further partitioned, while the proportions are 4.17/4.56/1.61% on conference videos. Although these proportions are much smaller than those of FCUs and XCUs, the BCUs with $N > 8$ still take a relatively large proportion. That is, only the partition of the potential BCUs with N=8 should be early terminated.

To speed up the partitioning process of BCUs, we further denote the potential BCUs without any foreground BU as the potential *pure background CUs*. The proportions of the split/non-split pure background CUs are also illustrated in Table II. We can observe that over 98% of the potential pure background CUs will not be partitioned any more for both surveillance and conference videos. Therefore, the CU partitioning termination strategy can be summarized as follows: *If the current region is a 16×16 potential BCU or can be regarded as a potential pure background CU, it should not be partitioned any more*.

2) Analysis of PU Pattern Selection: Similarly, when coding an input CU, the HEVC encoder usually compares the prediction distortion or RD cost for each available PU pattern. To reduce the complexity, here we conduct an experiment to analyze the distribution of each PU pattern.

After classifying the input CUs into BCUs, XCUs or FCUs, it is obvious that the proportion of selecting the PU pattern varies among three categories. Table III shows the distribution of the PU patterns for BCUs, XCUs and FCUs. We can see that, for BCUs with $N>8$, PU patterns of $2N\times N$, $N\times 2N$, $N\times N$ and AMP account for a very little proportion, much smaller than 5% on both surveillance and conference videos. But for BCUs with $N\leq 8$, XCUs and FCUs, the proportion of $2N\times 2N$ does not exceed 95%. This means that in the PU prediction process, we cannot disable all the $2N\times N$, $N\times 2N$, $N\times N$ and AMP for BCUs with $N\leq 8$, XCUs and FCUs. Moreover, we find that AMP takes a very little proportion for XCUs. Therefore, the PU pattern selection strategy is: *Only $2N\times 2N$ can be used for BCUs with $N>8$, all the candidate PU patterns should be tried for FCUs and BCUs with $N\leq 8$, and only AMP patterns are disabled for XCUs.*

TABLE IV
THE DISTRIBUTION OF THE MVDS

CU category \ mvd range	<=1pixel	1pixel~4pixel	>4pixel
Surveillance Videos BCU	99.854%	0.132%	0.014%
Surveillance Videos XCU	98.920%	1.051%	0.029%
Surveillance Videos FCU	98.749%	1.095%	0.156%
Conference Videos BCU	99.652%	0.304%	0.044%
Conference Videos XCU	98.633%	1.245%	0.122%
Conference Videos FCU	97.212%	2.492%	0.296%

3) Analysis of ME Simplification: Search range is an important factor in ME and should be no smaller than the final motion vector difference (MVD), which is the difference between the predicted motion vector (PMV) and the best matched motion vector. Table IV shows the distribution of the MVDs of BCUs, XCUs and FCUs for both surveillance and conference videos. From the table, we can see that for BCUs, more than 99.6% MVDs are less than 1 pixel. This means 1 integer search range is sufficient for BCUs. For XCUs and FCUs, the number of MVDs with more than 1 pixel is about 3∼10 times larger than that of BCUs. Thus their search range should not be narrowed. Therefore, the ME simplification strategy should be: *The motion search range should be set to 1 pixel for BCUs and kept unchanged for XCUs and FCUs.*

In summary, the ways of utilizing the background to reduce the encoding complexity can be expressed by: *Each input CU should be firstly classified according to their difference to the G-picture, and then we should employ different CU partitioning termination, PU pattern selection and ME simplification methods for different CU categories.*

IV. THE PROPOSED METHOD

Motivated by the above analyses, we propose the BHO method for surveillance and conference video coding. On one hand, to improve the coding efficiency, a background-based prediction structure is developed to optimize the low-delay HPS by utilizing the intra-coded G-picture as the long-term reference among the total four hierarchical reference frames, and a background-based hierarchical quantization is adopted to use a much smaller QP to encode the G-picture and adjust the QPs for each BGOP. On the other hand, in order to reduce the coding complexity, BHO firstly classifies the input CUs into BCUs, FCUs and XCUs, and then adopts

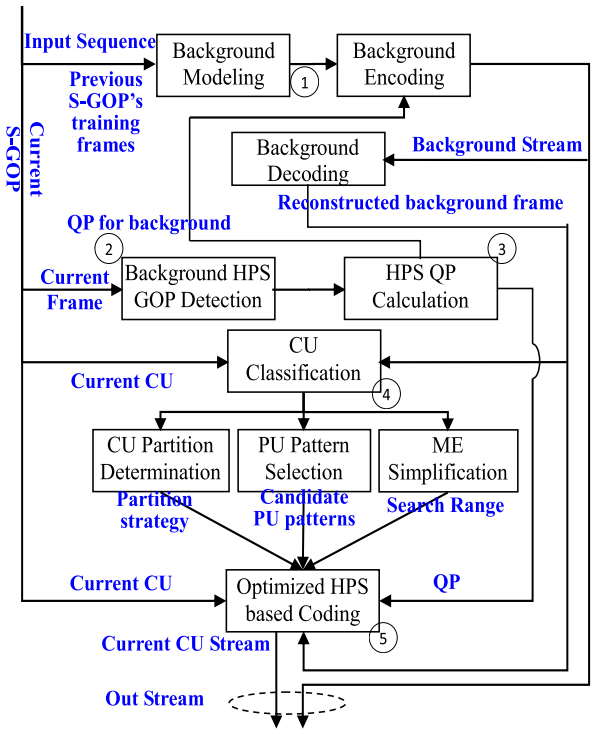

Fig. 8. The framework of the HEVC encoder with BHO. In this figure, the numbering indicates the order of each step in our method.

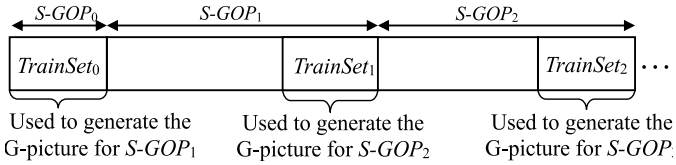

Fig. 9. The sequence structure for background generation.

different CU partitioning termination, PU pattern selection and ME simplification strategies for each category of CUs.

Fig. 8 describes the framework of the HEVC encoder with BHO. We can see that the encoder works as follows:

1) The input sequence is encoded S-GOP (*super large group of frames*, as shown in Fig. 9) by S-GOP. In the no-delay coding, each G-picture is generated from the training pictures in the previous S-GOP, and then utilized by the current S-GOP. Such a background picture will be encoded into the final stream to guarantee the decoding match. Meanwhile, the reconstructed background picture is decoded from the background stream.
2) For each input frame, BHO will check whether the current frame belongs to an NGOP or a BGOP. Here we detect the BGOP by checking whether the current GOP has a small number of foreground pixels compared with the G-picture.
3) According to the different types of HPS GOPs, BHO calculates the QP for each frame by either using the original quantization method in HM or a novel algorithm designed to adjust the QPs for all BGOP frames. When the current frame is the G-picture, a much smaller QP should be used.
4) For each input CU, BHO calculates the difference with its reconstructed background data and then classifies it into FCU, BCU or XCU. According to the CU category, the corresponding CU partitioning strategy, PU pattern candidates and motion search range are calculated for the coding process.
5) Finally, BHO performs the optimized HPS-based coding by replacing the fourth reference frame with the reconstructed G-picture, quantizing each picture with the calculated QP, terminating each CU partition with the CU partitioning strategy, predicting each CU with the selected PU pattern candidates, and searching the matched blocks within the estimated search range.

No doubt, BHO is an encoding optimization tool, since both the long-term reference and the non-display mechanism are supported by HEVC and the G-picture is actually a special I-picture. When encoding a surveillance or conference video, the optimized HEVC encoder with BHO can online train a G-picture without any delay, and then encodes such a frame into stream in forms of a non-display I-frame by intra-coding. Then the encoder marks the reconstructed G-picture as the only long-term reference for the following frames. When decoding, the G-picture is decoded without being displayed and used as the long-term reference.

A. Background Modeling and Updating

Since existing background modeling methods such as GMM [24] and mean-shift [25] often require a number of buffering frames for modeling and fraction-point calculation and thus are difficult to implement in video codecs, the running average algorithm is used in our BHO method. Its key idea is to estimate the average pixel values as the background pixels in a running way. Let I denote the current training frame, and a matrix A with unsigned 8-bit integers to represent the previous average result for all the pixels, then the algorithm calculates the current result A' by

$$A' = (A \times (n-1) + I + (n >> 1))/n, \tag{10}$$

where n is the number of the training frames. Therefore, this algorithm only requires one buffered frame to store A or A'. Each time given a training frame, only one multiply, shift, floor, divide and three add operations are performed.

Fig. 9 describes the sequence structure for background generation and updating. In this structure, the G-picture is generated S-GOP by S-GOP. That is, an initial GOP is utilized as $TrainSet_0$ to generate the G-picture for $S\text{-}GOP_1$, whereas the last GOP in $S\text{-}GOP_1$ is utilized as $TrainSet_1$ to generate the G-picture for $S\text{-}GOP_2$, and that in $S\text{-}GOP_2$is utilized as $TrainSet_2$ to generate the G-picture for $S\text{-}GOP_3$, … Here the first picture in the sequence is treated as the G-picture for coding the pictures in $TrainSet_0$ (which is also regarded as $S\text{-}GOP_0$). In this way, each S-GOP can utilize the corresponding G-picture to encode its pictures without delay. In our experiments, the number of pictures in each *TrainSet* and the length of an S-GOP are set as follows: 120 and 900 for surveillance videos, and 30 and 570 for conference videos since a clean background is hardly generated in a conference video.

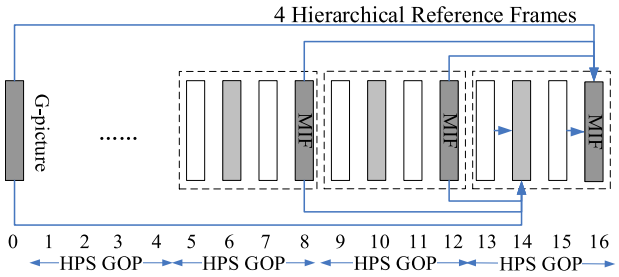

Fig. 10. The reference frame selection in the BHO method.

Note that, the coding bits of the G-pictures have been counted into the final bitrates in our experiments.

B. Background-Based Prediction Structure

In our BHO method, the G-picture should replace the fourth reference frame in HPS for surveillance and conference video coding. Therefore, BHO predicts each picture using its previous frame, two MIFs in the two previous HPS GOPs, and the long-term G-picture as the four reference frames. Fig. 10 depicts the prediction structure of the low-delay HPS in BHO, where indexes of the four reference frames for the 14-th picture are the 13, 12, 8 and 0; those for the 16-th picture are 15, 12, 8 and 0. In the following discussion, we will describe the hierarchical quantization method in BHO, which contains the BGOP detection and the HPS QP calculation.

1) Detecting the BGOPs: Following the conclusion in Section III-A.2, BHO should detect the BGOP and adjust the QPs for its frames. Intuitively, the first picture in a GOP can mostly represent the scene content. So if it has a large proportion of similar data to the G-picture, the GOP can be regarded as a BGOP. In practice, the BGOP detection only using the first picture has the benefit to accomplish the no-delay encoding. Following these ideas, the BGOP detection can be formulated as follows: Supposing that the length of an HPS GOP is an even L, $S(A,B)$=1 (or 0) represents that A and B have a large proportion of similar (or no similar) data, Bg denotes a G-picture, then the GOP type $G(I_n)$ of any n-thin put frame I_n, in which n is re-written by $k\times L+j\,(k\geq 0,\ j=0\sim L-1)$ such that $n=k\times L$ corresponds to the first picture in the current GOP, calculated by

$$G(I_n)=\begin{cases} BGOP, & S(I_{k\times L},Bg)=1;\\ NGOP, & S(I_{k\times L},Bg)=0.\end{cases} \tag{11}$$

As for $S(A,B)$, we just adopt an integer ME with 1-pixel range to search each picture A's BU in the picture B. If the similar BUs between A and B are in the majority, we can infer that A and B have a large proportion of similar data. This statement is

$$S(A,B)=\begin{cases}1, & \text{if } 16\times\|M(A,B)\|/w\times h>0.8;\\ 0, & \text{Otherwise.}\end{cases}$$

$$M(A,B)=\left\{(p,q)\,\middle|\,\sum_{s,t=1}^{4}\left|A_{4p+s,4q+t}-B_{4p+s,4q+t}\right|\leq 80,\ p<\frac{h}{4},q<\frac{w}{4}\right\}, \tag{12}$$

Algorithm 1 BGOP Detection

Input: Current $h\times w$ picture I_n where $n=k\times L+j$

Output: I_n's GOP type $G(I_n)$ in $\{NGOP,BGOP\}$

```
If j ≠ 0 then G(I_n) = G(I_{n-j}); return.
A = I_n ; B = Bg ; M(A,B) = ∅ ;
for p=1 to h/4 do
    for q=1 to w/4 do
        if Σ_{s,t=1}^{4} |A_{4p+s,4q+t} − B_{4p+s,4q+t}| ≤ 80
        then M(A,B) = M(A,B) ∪ {(p,q)} ;
    end
end
if 16×‖M(A,B)‖/w×h > 0.8 then S(A,B) = 1
                         else S(A,B) = 0 ;
if S(A,B) == 1 then G(I_n)=BGOP
               else G(I_n) = NGOP ;
return.
```

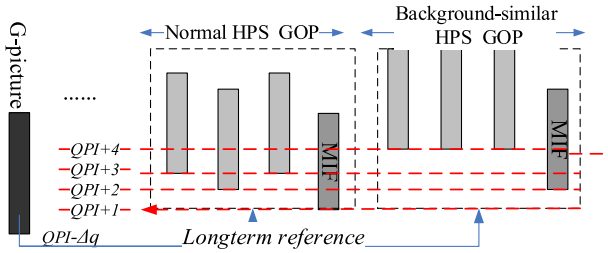

Fig. 11. The optimized hierarchical quantization.

where $A_{x,y}$ and $B_{x,y}$ are pixels at position (x, y) of A and B, h and w are the height and width of each input frame, and $||X||$ denotes the number of elements in set X. Therefore, we can derive the following Algorithm 1 to calculate the HPS GOP type $G(I_n)$.

2) HPS QP Calculation: Supposing the first intra picture in the low-delay HPS based video coding (which is also the first picture in the sequence) is quantized with QPI, the QP for each NGOP can be calculated as

$$QP(I_{k\times L+j})=\begin{cases}QPI+1, & \text{if } j=L-1;\\ QPI+2, & \text{if } j=L/2;\\ QPI+3, & \text{if } j\neq L/2 \text{ or } L-1.\end{cases} \tag{13}$$

Note that this quantization follows the HM low-delay HPS encoder, as shown in the left part of Fig. 11.

For BGOPs, however, BHO adopts a different quantization strategy so as to save more bits while remaining almost the same or even better coding quality. Following the analysis results in Section III-A.2, BHO is to adaptively calculate the QPs for frames in each BGOP according to the following rules:

1) The frames in a BGOP should be quantized with larger QPs than those used for frames in an NGOP, as suggested by the analysis conclusion in Section III-A.2.
2) In a BGOP, all the frames except the MIF are quantized in the same way. This is because they have the approximately same importance in the prediction process for the following frames.

3) It is well known that larger QP differences between the neighboring frames may produce a worse subjective quality of the reconstructed video. Thus the maximal QP difference in a BGOP should keep the same as that in an NGOP (i.e., it is 2 in the HM encoder). As shown in Fig. 11, such a strategy can produce the least impact on the subjective quality.

As a result, QPs for frames in each BGOP can be set as:

$$QP(I_{k\times L+j}) = \begin{cases} QPI+2, & \text{if } j = L-1; \\ QPI+4, & \text{if } j \neq L-1. \end{cases} \quad (14)$$

The remaining problem is how to quantize the G-picture. As mentioned in Section III-A.2, *the G-picture should be quantized with a much smaller QP than the other frames*. Here we let ΔQP denote the QP difference between QPI and the QP of the G-picture. Obviously, with a larger ΔQP, there will be quite a lot bits produced by encoding the G-picture. However, it is certain that the additional bit cost of the G-picture with a smaller QP should be no more than the bit saving on all the P/B pictures in the S-GOP which utilizes that G-picture as the reference. That is, if the bit cost of the G-picture is much larger than the bit saving on all the P/B pictures in the S-GOP, we should employ a relatively small ΔQP; and vice versa. To make the values of ΔQP adaptive to different video contents, BHO thus calculates ΔQP according to the bit cost of the G-picture (denoted by C_1) and the average bit cost of all the P/B pictures (denoted by C_{bp}) in the previous S-GOP.[1] This strategy is expressed as follows

$$\Delta QP = \begin{cases} 5, & \text{if } C_1/C_{bp} > LS/3; \\ 10, & \text{if } LS/20 < C_1/C_{bp} < LS/3; \\ 20, & \text{if } C_1/C_{bp} < LS/20. \end{cases} \quad (15)$$

Here we set ΔQP be 5, 10 and 20 for different C_1/C_{bp} values, by experimentally selecting the best one from several typical groups ({2,4,8}, {3,6,12}, {4,8,16}, {5,10,20} and {6,12,22} in our experiment). For simplicity, in each group, the ΔQP value for the lower C_1/C_{bp} case (e.g., $C_1/C_{bp} \geq LS/3$) increasingly doubles that for the larger case (e.g., $LS/20 \leq C_1/C_{bp} < LS/3$). And the maximal ΔQP value is set to 22 because 22 is the smallest QPI in the HM common test condition. In this way, although we employ $QPI - \Delta QP$ to quantize the G-picture, its coding bits still take up a very small proportion in the total coding bits.

[1]Note that in the setting of low-delay coding, when encoding the G-picture, the bit cost of all the P/B pictures in the current S-GOP that utilizes that G-picture as the reference is still not available. So in our BHO, it is approximated by the average bit cost of all the P/B pictures in the previous S-GOP.

Current region satisfies	CU Partition
Being a BCU with $N<16$ (i.e., $2N<32$) or having no foreground BUs	SF=0
Others	SF=0 / SF=1

Fig. 12. CU partitioning termination.

TABLE V
PU PATTERN CANDIDATES FOR DIFFERENT CU CATEGORIES

category	N	PU pattern candidates
BCU	>8	$2N\times2N$
	≤8	$2N\times2N$,$2N\times N$,$N\times2N$,$N\times N$, AMP
XCU	ALL	$2N\times2N$,$2N\times N$,$N\times2N$,$N\times N$
FCU	ALL	$2N\times2N$,$2N\times N$,$N\times2N$,$N\times N$, AMP

In summary, the hierarchical quantization in the BHO can be formulated by combing Eq. 13, Eq. 14 and Eq. 15 as shown in (16), shown at the bottom of the page.

C. Complexity-Reduction Strategies Using CU Classification

In Section III-B, we have derived from the experimental analyses on the strategies for CU partitioning termination, PU pattern selection and ME simplification. Thus given the category of each CU, these strategies can be stated as follows:

1) For CU partition, if the current CU is a 16×16 BCU, the recursive CU partitioning for BCUs should be terminated; otherwise, we only terminate the potential BCUs without foreground BUs. This strategy is visualized in Fig. 12.
2) For PU pattern selection, we only utilize $2N\times2N$ for BCUs with $N>8$, try all the possible prediction patterns for FCUs and BCUs with $N\leq8$, and disable AMP patterns in XCUs. Table V lists the candidate PU patterns for each CU category.
3) To further reduce the complexity, the motion search range will be set to 1 pixel for BCUs. In contrast, the range is not changed for XCUs and FCUs.

V. EXPERIMENTS

A. Experimental Setup

Two datasets with totally sixteen CIF∼HD videos are used:

1) Surveillance Dataset: Besides the four CIF&SD surveillance videos shown in Fig. 5, we also employ another four CIF&SD videos and two 1600×1200 HD videos from Hisense Co. Ltd to evaluate the BHO. Fig. 13(a) shows all these videos. We can see that, these ten surveillance videos cover different monitoring scenes, including bright and

$$QP(I_{k\times L+j}) = \begin{cases} QPI-\Delta QP, & \text{if } I_{k\times L+j} = Bg; \\ QPI, & \text{if } k\times L+j = 0; \\ QPI+1, & \text{if } G(I_{k\times L+j}) = NGOP \text{ and } j = L-1; \\ QPI+2, & \text{if } G(I_{k\times L+j}) = NGOP \text{ and } j = L/2; \\ QPI+2, & \text{if } G(I_{k\times L+j}) = BGOP \text{ and } j = L-1; \\ QPI+3, & \text{if } G(I_{k\times L+j}) = NGOP \text{ and } (j \neq L/2 \text{ or } L-1); \\ QPI+4, & \text{if } G(I_{k\times L+j}) = BGOP \text{ and } j \neq L-1. \end{cases} \quad (16)$$

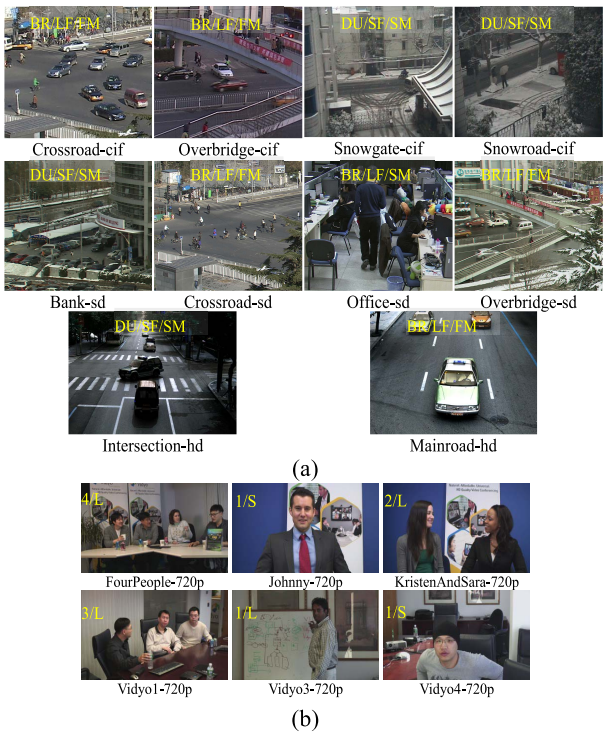

Fig. 13. All tested surveillance videos and conference videos. (a) Surveillance videos. (b) Conference videos.

TABLE VI

HM'S LOW-DELAY HIGH-EFFICIENCY 10-BIT DEPTH CONFIGURATIONS

Config.	Value	Config.	Value	Config.	Value
Frame Structure	*low delay IBBB*	FastSearch	*Enable*	Search Range	*64*
		GOPSize	*4*	IBDI depth	*10*
SAO	*Enable*	IntraPeriod	*-1*	Rate Control	*Disable*
RDOQ	*Enable*	AMP	*Enable*	Hadamard ME	*1*

dusky lightness (BR/DU), large and small foreground (LF/SF), fast and slow motion (FM/SM). Note that they have been either utilized in the standardization process of AVS2 [28] or obtained from the PKU-SVD-A dataset [30], [31].

2) Conference Dataset: As shown in Fig. 13(b), totally 6 conference videos are used. Note that four of them have been shown in Fig. 5. These videos were originally used to evaluate the performance of HEVC and H.264/AVC. They contain 1∼4 persons, having large or small (L/S) proportions of swaying regions.

To evaluate the coding performance of the BHO method, the HEVC test model HM12.0 with low-delay configuration is used as the basic experimental platform (shortly as HM). Here our objective is to evaluate how much the efficiency improvement and complexity reduction that our BHO method can achieve over HM12.0. Table VI shows the details of the HEVC with low-delay configuration in [26]. In the experiments, BD-rate and BD-PSNR [27] are utilized as the metrics for the coding performance.

B. The Overall Bit Saving and Complexity Reduction

In the first set of experiments, we will evaluate the overall bit saving and time saving of the BHO method on surveillance and conference videos, respectively.

TABLE VII

THE BD-RATE AND TIME SAVING (%) ON SURVEILLANCE VIDEOS

Surveillance Videos	BHO vs. HM			
	BD Rate (Y,U,V)			Time saving
Crossroad-cif	-18.39%	-46.41%	-43.20%	32.28%
Overbridge-cif	-30.60%	-79.59%	-51.80%	26.03%
Snowgate-cif	-55.88%	-77.13%	-74.02%	44.22%
Snowroad-cif	-53.18%	-66.21%	-66.40%	60.06%
Bank-sd	-48.88%	-72.46%	-73.78%	60.79%
Crossroad-sd	-29.24%	-71.06%	-67.37%	37.73%
Office-sd	-16.17%	-54.70%	-50.88%	27.28%
Overbridge-sd	-46.91%	-71.84%	-70.48%	56.05%
Intersection-hd	-21.45%	-33.74%	-31.28%	26.28%
Mainroad-hd	-70.15%	-83.13%	-75.49%	65.59%
Average	**-39.09%**	**-65.63%**	**-60.47%**	**43.63%**

TABLE VIII

THE BD-RATE AND TIME SAVING (%) ON CONFERENCE VIDEOS

Conference Videos	BHO vs. HM			
	BD Rate (Y,U,V)			Time saving
FourPeople-720p	-8.02%	-15.86%	-14.41%	37.31%
Johnny-720p	1.82%	-15.91%	-14.53%	48.33%
Kristen&Sara-720p	-9.06%	-19.28%	-18.70%	41.18%
Vidyo1-720p	-5.99%	-11.15%	-13.02%	38.14%
Vidyo3-720p	-10.10%	-16.53%	-33.67%	56.90%
Vidyo4-720p	-0.26%	-13.37%	-15.18%	40.19%
Average	**-5.27%**	**-15.35%**	**-18.25%**	**43.68%**

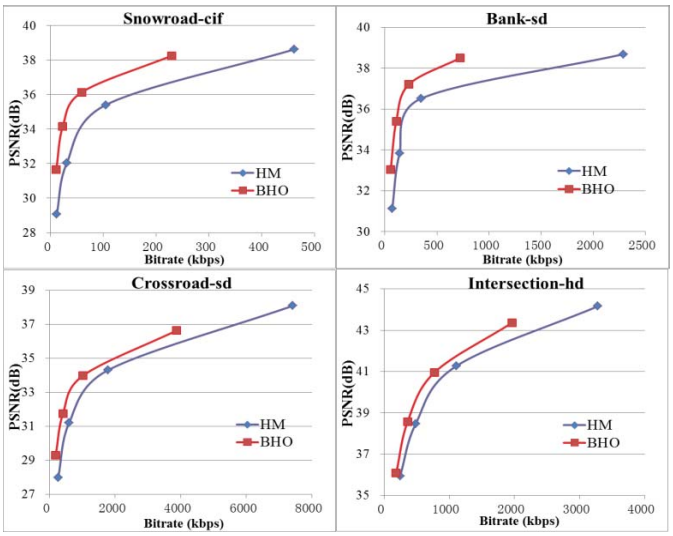

Fig. 14. RD curves of the four surveillance videos, Snowroad-cif, Bank-sd, Crossroad-sd and Intersection-hd.

1) Overall Results Analysis: Table VII illustrates the overall results of BD-rate, BD-PSNR and time saving by comparing BHO with HM on surveillance videos, while Table VIII presents the results on conference videos. Compared with HM, BHO averagely saves 39.09% of the total coding bits and 43.63% of the encoding time on surveillance videos. On conference videos, BHO averagely achieves 5.27% bit saving, 43.68% time saving over HM. Fig. 14 and 15 illustrate some RD curve examples of BHO and HM.

From the results and RD curves, we can observe that BHO tends to obtain more bit savings and larger

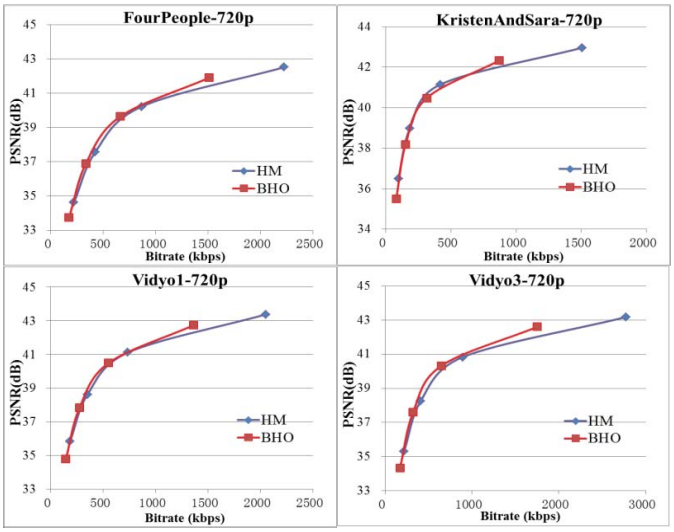

Fig. 15. RD curves of the four conference videos, FourPeople, Kristen&Sara, Vidyo1 and Vidyo3.

time savings on some videos with large proportions of background (e.g., Snowgate-cif, Bank-sd, Mainroad-hd, and FourPeople-720p). This is mainly because the large proportions of background pixels can be better predicted from the G-picture that is quantized by a smaller QP. While on the videos with large proportions of foreground (e.g., Intersection-hd, Office-sd, Johnny-720p), the background proportion is the smallest and the modeled background is not very clean, so BHO obtains the least bit saving over HM.

We can also see that there are much smaller performance gains on conference videos than on surveillance videos. On conference videos, the background is usually covered by the tightly-swaying heads, arms and bodies all the time. Thus it is difficult to generate a *clean* G-picture. In this case, such a *noisy* background has lower prediction efficiency for the EBRs. However, considering the low-delay HEVC encoder has already saved a remarkable number of bits in compressing conference videos, the additional 5.27% bit saving on the luma component and more than 10% percentage BD-Rate gains on the chroma components are still meaningful. It should be noted that, although there is a little performance loss (i.e., 1.82%) on the luma component of Johnny-720p, the gains on its chroma components are large enough to achieve a total gain. Moreover, the time saving on conference videos is as large as that on surveillance videos, making BHO applicable for real-time conferencing systems.

2) Experimental Analysis Based on CU Classification: To further validate the fact that a larger background proportion in videos would lead to more performance gain of BHO, we conduct one supplementary experiment for CU classification. Table IX presents the CU category distributions for all sequences, while Fig. 16 visualizes two examples of CU category distributions for Mainroad-hd and FourPeople-720p.

We can see that on the surveillance sequences with a smaller BCU proportion (e.g., $\leq$30%), BHO can obtain less than 30% bit saving and smaller than 35% time saving (e.g., Office-sd, Crossroad-cif, Overbridge-cif and Intersection-hd) over HM. This is because more-efficient

TABLE IX
DISTRIBUTIONS OF CU CATEGORIES, BIT SAVINGS AND TIME SAVING (%)

Surveillance & conference Videos	BCU	XCU	FCU	Luma BD Rate	Chroma BD Rate	Time saving
Crossroad-cif	28.45	29.44	42.11	-18.39	-44.81	32.28
Overbridge-cif	17.68	42.89	39.42	-30.60	-65.70	26.03
Snowroad-cif	35.48	36.81	27.70	-55.88	-75.58	44.22
Snowgate-cif	65.96	11.95	22.09	-53.18	-66.31	60.06
Bank-sd	75.06	9.30	15.65	-48.88	-73.12	60.79
Crossroad-sd	41.01	32.15	26.84	-29.24	-69.22	37.73
Office-sd	22.24	30.16	47.59	-16.17	-52.79	27.28
Overbridge-sd	68.43	15.20	16.38	-46.91	-71.16	56.05
Intersection-hd	30.05	27.30	42.66	-21.45	-32.51	26.28
Mainroad-hd	83.52	5.11	11.37	-70.15	-79.31	65.59
Surveillance Average	**46.79**	**24.03**	**29.18**	**-39.09**	**-63.05**	**43.63**
FourPeople-720p	47.65	20.03	32.32	-8.02	-15.14	37.31
Johnny-720p	59.29	7.49	33.22	1.82	-15.22	48.33
Kristen&Sara-720p	48.17	15.85	35.98	-9.06	-18.99	41.18
Vidyo1-720p	46.37	15.75	37.88	-5.99	-12.09	38.14
Vidyo3-720p	44.40	27.52	28.08	-10.10	-25.10	56.90
Vidyo4-720p	49.30	14.40	36.30	-0.26	-14.28	40.19
Conference Average	**41.87**	**18.36**	**39.77**	**-5.27**	**-16.80**	**43.68**

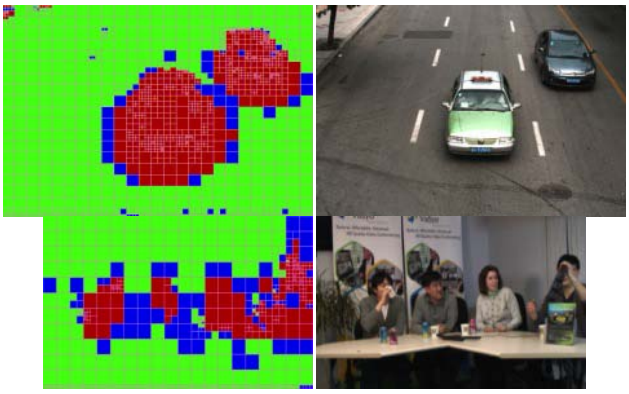

Fig. 16. CU category distribution examples in Mainroad and FourPeople, where blue, red and green grids are XCUs, FCUs and BCUs.

background prediction plays a critical role in improving the coding performance on surveillance videos, only if clean G-pictures can be generated. Whereas for conference videos where no clean G-pictures can be built, such a rule may not be followed. In this case, for videos with similar BCU proportions, the coding performance gains will be larger if there is more motion involved. For example, on the sequences with smaller swaying regions (e.g., Johnny-720p and Vidyo4-720p), the bitrate savings are obviously relative small.

C. Experiments on Different Components of BHO

The second set of experiments is performed to evaluate the different roles of the components in BHO, i.e., the G-picture-based HPS optimization algorithm and the adaptive speed-up algorithm based on CU classification. Towards this end, we evaluate the performance of BHO over HM with different numbers of reference frames in C.1; while in C.2, we check whether the adaptive speed-up algorithm can guarantee little encoding efficiency loss.

TABLE X
THE BD-RATE COMPARISON(%) FOR LUMA AND CHROMA AVERAGE BETWEEN BHO AND HM FOR DIFFERENT REFERENCE FRAME NUMBERS

Reference frame		2		3		4	
Components		luma	chroma	luma	chroma	luma	chroma
Surveillance	Crossroad-cif	-22.93	-47.40	-21.30	-47.40	-18.39	-44.81
	Overbridge-cif	-31.47	-69.40	-27.90	-66.20	-30.60	-65.70
	Snowgate-cif	-62.43	-77.70	-58.09	-76.30	-55.88	-75.58
	Snowroad-cif	-59.26	-68.75	-55.62	-68.25	-53.18	-66.31
	Bank-sd	-54.04	-75.30	-51.07	-74.00	-48.88	-73.12
	Crossroad-sd	-36.79	-72.35	-33.65	-71.10	-29.24	-69.22
	Office-sd	-21.08	-57.65	-19.34	-55.55	-16.17	-52.79
	Overbridge-sd	-53.45	-74.45	-50.18	-72.75	-46.91	-71.16
	Intersection-hd	-24.69	-36.10	-23.06	-33.40	-21.45	-32.51
	Mainroad-hd	-76.56	-83.00	-73.47	-81.50	-70.15	-79.31
Conference	FourPeople-720p	-11.22	-16.80	-9.54	-16.13	-8.02	-15.14
	Johnny-720p	2.18	-11.60	1.27	-15.70	1.82	-15.22
	Kristen&Sara-720p	-9.79	-18.05	-9.84	-19.61	-9.06	-18.99
	Vidyo1-720p	-6.40	-10.85	-6.87	-12.65	-5.99	-12.09
	Vidyo3-720p	-11.39	-24.80	-11.48	-26.00	-10.10	-25.10
	Vidyo4-720p	1.25	-11.65	-1.45	-14.52	-0.26	-14.28

1) Gains Over HM12.0 With 2, 3 or 4 Reference Frames: In practice, different video applications often have different demands of the memory size and memory-access bandwidth. Because the reference frame number has an important impact on these factors, it is necessary to validate the practicability of BHO with 2, 3 or 4 reference frames. Table X presents the comparison results between BHO and HM12.0 by utilizing different numbers of reference frames. We can see that the less reference frames are utilized, the more BHO gains over HM12.0. Especially in the low-memory-bandwidth case where 2 reference frames are used, BHO can generally save more bits than HM12.0. This is mainly because that when there are less available reference frames, more EBRs will exist and the G-picture will be referenced more.

2) Efficiency Reduction Caused by the Speed-Up Algorithm: Intuitively, the adaptive speed-up algorithm based on CU classification in BHO might reduce the coding efficiency more or less because the number of candidate PU patterns would be reduced, the CU partition might be early terminated and the motion search range would be narrowed. Despite of this, the proposed speed-up strategies make a good tradeoff between the coding efficiency and the complexity by reducing the number of PU patterns only for BCUs with N>8 and XCUs, terminating CU partition for BCUs with N=16 and the pure background CUs, and narrowing the search range only for BCUs. Table XI shows the comparison between BHOs *with* and *without* the speed-up strategies. We can see that, there are only 0.07dB and 0.05dB PSNR decreases on average respectively for surveillance and conference videos. This means that BHO can obtain a remarkable speed-up ratio (averagely 43.63% on surveillance videos and 43.68% on conference video) only at an ignorable loss of the coding efficiency.

D. Comparison With Three State-of-the-Art Methods

The last set of experiments are used to compare BHO with two state-of-the-art methods [13], [29], also on the 16 surveillance and conference videos. For fair comparison, the two methods are also implemented on HM 12.0. Table XII presents the BD-Rate of BHO vs. the methods in [13] and [29] on Y, U and V components. Overall, BHO can achieve a more robust performance on both surveillance and conference videos than the two methods. This is due to the HPS quantization algorithm and the CU-classification based adaptive speed-up strategies in BHO, which are not utilized in [13] and [29]. Compared with [13], the performance bit-savings of BHO are 5.46%, 15.14%, 13.40% on surveillance videos and 5.69%, 11.52%, 12.27% on conference videos. This is mainly because BHO utilizes the G-picture as the long-term reference and adopts the hierarchical quantization method for each HPS GOP. While compared with [29], the bit-savings of BHO are 5.42%, 15.45%, 13.73% on surveillance videos and 4.19%, 9.37%, 10.37% on conference videos. These performance gains are obtained only due to the hierarchical quantization method for each HPS GOP. It should be noted that when we implemented the method [29] on the HEVC platform, NGOPs and BGOPs are treated equally. Therefore, the performance

TABLE XI
THE BD-PSNR ON Y (dB) AND COMPLEXITY REDUCTION (%) PRODUCED BY THE FAST ALGORITHMS

Sequence	Crossroad-cif	Overbridge-cif	Snowgate-cif	Snowroad-cif
PSNR(dB)	-0.11	-0.05	-0.08	-0.06
Time saving	32.28%	26.03%	44.22%	60.06%
Sequence	Bank-sd	Crossroad-sd	Office-sd	Overbridge-sd
PSNR(dB)	-0.11	-0.07	-0.03	-0.10
Time saving	60.79%	37.73%	27.28%	56.05%
Sequence	Mainroad-hd	Intersection-hd	**Surveillance Video Average**	
PSNR(dB)	-0.01	-0.04	**-0.07**	
Time saving	26.28%	65.59%	**43.63%**	
Sequence	Vidyo1-720p	Vidyo3-720p	Vidyo4-720p	Johnny-720p
PSNR(dB)	-0.04	-0.04	-0.03	-0.08
Time saving	38.14%	56.90%	40.19%	48.33%
Sequence	Kristen&Sara	FourPeople	**Conference Video Average**	
PSNR(dB)	-0.06	-0.03	**-0.05**	
Time saving	41.18%	37.31%	**43.68%**	

TABLE XII
THE BD-RATE (%) COMPARISON BETWEEN BHO AND [13], [29]

BHO vs.	[13]			[29]		
	Y	U	V	Y	U	V
Crossroad-cif	0.25	-6.45	-4.34	-0.70	-7.31	-4.49
Overbridge-cif	-0.18	-10.83	-4.53	-1.01	-14.84	-10.71
Snowgate-cif	-11.35	-27.49	-26.13	-10.28	-26.31	-24.59
Snowroad-cif	-10.39	-22.60	-22.81	-9.52	-21.31	-21.52
Bank-sd	-2.46	-16.07	-13.99	-7.41	-18.34	-16.89
Crossroad-sd	-4.95	-15.11	-14.86	-2.52	-16.12	-14.58
Office-sd	-3.84	-8.12	-6.66	1.60	-7.94	-6.93
Overbridge-sd	-3.69	-17.15	-13.82	-6.93	-18.82	-16.08
Intersection-hd	-3.31	-4.86	-6.03	-5.84	-8.07	-6.99
Mainroad-hd	-14.67	-22.75	-20.83	-11.55	-15.41	-14.48
Average	**-5.46**	**-15.14**	**-13.40**	**-5.42**	**-15.45**	**-13.73**
FourPeople-720p	-4.46	-8.85	-7.84	-5.42	-9.17	-7.79
Johnny-720p	-6.28	-15.82	-14.59	-3.49	-12.36	-10.35
Kristen&Sara-720p	-5.38	-10.93	-11.68	-5.47	-10.86	-11.24
Vidyo1-720p	-5.12	-9.85	-9.85	-4.01	-7.82	-8.01
Vidyo3-720p	-5.46	-10.98	-16.85	-3.88	-8.20	-15.42
Vidyo4-720p	-7.45	-12.70	-12.82	-2.84	-7.82	-9.40
Average	**-5.69**	**-11.52**	**-12.27**	**-4.19**	**-9.37**	**-10.37**

gain of BHO over [29] is equal to the gain of BHO over the way of treating the two kinds of GOPs equally. In addition, the time saving of BHO over [13] and [29] should be even larger than 43.63% and 43.68% on these videos, since BHO adopts the adaptive speed-up strategies for different CU categories, while [13] and [29] increase the coding complexity.

VI. Conclusion

This paper proposes a background modeling based HPS optimization (BHO) method to improve the performance of HEVC low-delay coding on surveillance and conference videos. BHO consists of two key components, i.e., the G-picture-based HPS optimization algorithm and the adaptive speed-up strategies based on CU classification. Extensive experiments are conducted on eighteen CIF∼HD surveillance and conference videos. Results show that compared with HM, BHO can averagely achieve 39.09% and 5.27% bit savings, and 43.63% and 43.68% time savings, respectively on surveillance and conference videos. Our main contributions can be summarized as follows:

1) For surveillance and conference videos, we make a set of experimental and theoretical analyses on the relationships among the G-picture, the efficiency of the low-delay HPS and the coding information distribution. These analyses are highly suggestive to design the high-efficiency and low-complexity encoding optimization algorithms in the following studies.
2) By utilizing background modeling, an optimized prediction structure and hierarchical quantization algorithm is proposed in BHO to improve the low-delay HPS in the HM encoder. This leads to a remarkable increase of the coding efficiency on surveillance and conference videos.
3) By classifying CUs with the G-picture, the CU-category-adaptive CU partitioning termination, PU candidate selection and ME simplification strategies are developed in BHO to remarkably decrease the coding complexity while keeping the efficiency.

In the future work, we will further optimize the efficiency of BHO, especially on designing a better parameter setting method for ΔQP and a better background modeling and reference strategy for conference video coding. With the higher coding efficiency and faster processing performance, it is expected that BHO can better meet various requirements in practical surveillance and teleconferencing applications.

References

[1] *Advanced Video Coding for Generic Audio-Visual Services*, document ITU T Rec. H.264 and ISO/IEC 14496–10 (AVC), 2003.

[2] H. G. Musmann, M. Hötter, and J. Ostermann, "Object-oriented analysis-synthesis coding of moving images," *Signal Process., Image Commun.*, vol. 1, no. 2, pp. 117–138, 1989.

[3] E. François, J.-F. Vial, and B. Chupeau, "Coding algorithm with region-based motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 97–108, Feb. 1997.

[4] A. Vetro, T. Haga, K. Sumi, and H. Sun, "Object-based coding for long-term archive of surveillance video," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2003, pp. 417–420.

[5] R. Venkatesh Babu and A. Makur, "Object-based surveillance video compression using foreground motion compensation," in *Proc. 9th Int. Conf. Control, Autom., Robot. Vis.*, 2006, pp. 1–6.

[6] A. Hakeem, K. Shafique, and M. Shah, "An object-based video coding framework for video sequences obtained from static cameras," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 608–617.

[7] D. Venkatraman and A. Makur, "A compressive sensing approach to object-based surveillance video coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2009, pp. 3513–3516.

[8] X. Jin and S. Goto, "Encoder adaptable difference detection for low power video compression in surveillance system," *Signal Process., Image Commun.*, vol. 26, no. 3, pp. 130–142, 2011.

[9] T.-C. Chen, Y.-W. Huang, C.-Y. Tsai, C.-T. Huang, and L.-G. Chen, "Single reference frame multiple current macroblocks scheme for multi-frame motion estimation in H.264/AVC," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2. May 2005, pp. 1790–1793.

[10] R. Ding, Q. Dai, W. Xu, D. Zhu, and H. Yin, "Background-picture based motion compensation for video compression," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2004.

[11] M. Paul, W. Lin, C. T. Lau, and B.-S. Lee, "Video coding using the most common frame in scene," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 734–737.

[12] X. Zhang, L. Liang, Q. Huang, and W. Gao, "An efficient coding scheme for surveillance videos captured by stationary cameras," in *Proc. Vis. Commun. Image Process.*, Jul. 2010, pp. 1–10.

[13] M. Paul, W. Lin, C.-T. Lau, and B.-S. Lee, "Explore and model better I-frames for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 9, pp. 1242–1254, Sep. 2011.

[14] G. J. Sullivan, J. 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] *High Efficiency Video Coding (HEVC) Test Model 12 (HM12) Encoder Description*, document JCTVC-N1002, Jul. 2013.

[16] K. McCann *et al.*, *Samsung's Response to the Call for Proposals on Video Compression Technology*, document JCTVC-A124.doc, Apr. 2010.

[17] G. Laroche, T. Poirier, and P. Onno, *Encoder Speed-Up for the Motion Vector Predictor Cost Estimation*, document JCTVC-H0178, Feb. 2012.

[18] C. Lan, J. Xu, G. J. Sullivan, and F. Wu, *Intra Transform Skipping*, document JCTVC-I0408_r1.doc, May 2012.

[19] Y. Su and M.-T. Sun, "Fast multiple reference frame motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 3, pp. 447–452, Mar. 2006.

[20] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[21] J. Xu, F. Wu, and H. Li, *Encoding Optimization to Improve Coding Efficiency for Low Delay Cases*, document JCTVC-F701r1, Jul. 2011.

[22] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[23] Q. Tang and P. Nasiopoulos, "Efficient motion re-estimation with rate-distortion optimization for MPEG-2 to H.264/AVC transcoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 262–274, Feb. 2010.

[24] M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. IEEE 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[25] Y. Liu, H. Yao, W. Gao, X. Chen, and D. Zhao, "Nonparametric background generation," *J. Vis. Commun. Image Represent.*, vol. 18, no. 3, pp. 253–263, 2007.

[26] F. Bossen, *HM 8 Common Test Conditions and Software Reference Configurations*, document JCTVC-J1100, Jul. 2012.

[27] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33.doc, Apr. 2001.

[28] AVS2, *Information Technology—Advanced Media Coding, Part 2 Video*, document AVS N1955_r1, Mar. 2013.

[29] X. Zhang, T. Huang, Y. Tian, and W. Gao, "Background-modeling-based adaptive prediction for surveillance video coding," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 769–784, Feb. 2014.

[30] W. Gao, Y. Tian, T. Huang, S. Ma, and X. Zhang, "IEEE 1857 standard empowering smart video surveillance systems," *IEEE Intell. Syst.*, to be published.

[31] *PKU-SVD-A*. [Online]. Available: http://mlg.idm.pku.edu.cn/resources/pku-svd-a.html

Xianguo Zhang (S'12–M'13) received the B.S. degree in computer science and technology from Peking University, Beijing, China, in 2007, and the Ph.D. degree from the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, in 2013. He is currently a Researcher with MediaTek Inc., Beijing. His research interests include video coding, transcoding, and processing.

Yonghong Tian (M'05–SM'10) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing. He has authored or co-authored over 100 technical articles in refereed journals and conferences. His research interests include computer vision, multimedia analysis, and coding. He is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, a Young Associate Editor of the *Frontiers of Computer Science in China*, a member of the IEEE TCMC-TCSEM Joint Executive Committee in Asia. He was a recipient of the National Science and Technology Progress Awards in 2010, the Best Performer Award in the TRECVID content-based copy detection task (2010–2011), the Top Performer Award in the TRECVID retrospective surveillance event detection task (2009–2012), and an award in the WikipediaMM task in ImageCLEF 2008.

Tiejun Huang (M'01–SM'12) is currently a Professor with the School of Electronic Engineering and Computer Science, Peking University, Beijing, China, where he is also the Director of the Institute for Digital Media Technology. He received the Ph.D. degree in pattern recognition and intelligent system from Huazhong (Central China) University of Science and Technology, Wuhan, China, in 1998, and the master's and bachelor's degree in computer science from the Wuhan University of Technology, Wuhan, in 1995 and 1992, respectively. His research area includes video coding, image understanding, digital right management, and digital library. He has authored or co-authored over 100 peer-reviewed papers and three books. He is a member of the Board of Director for Digital Media Project, the Advisory Board of the IEEE Computing Society, and the Board of the Chinese Institute of Electronics.

Siwei Dong received the B.S. degree from Chongqing University, Chongqing, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include surveillance video coding and multimedia learning.

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991. He is currently a Professor of Computer Science with Peking University, Beijing, China. Before joining Peking University, he was a Professor with the Harbin Institute of Technology, Harbin, China, from 1991 to 1995, and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, from 1996 to 2006. He has published extensively, including five books and over 600 technical articles in refereed journals and conference proceedings, in the areas of image processing, video coding and communication, computer vision, multimedia retrieval, multimodal interface, and bioinformatics.

Prof. Gao served or serves on the Editorial Board of several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, the *EURASIP Journal of Image Communications*, and the *Journal of Visual Communication and Image Representation*. He chaired a number of prestigious international conferences on multimedia and video signal processing, such as the IEEE ICME 2007, ACM Multimedia 2009, and IEEE ISCAS 2013, and also served on the Advisory and Technical Committees of numerous professional organizations. He is a member of the Chinese Academy of Engineering, and a fellow of the Association for Computing Machinery.