Background-Modeling-Based Adaptive Prediction for Surveillance Video Coding

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

***Abstract*— The exponential growth of surveillance videos presents an unprecedented challenge for high-efficiency surveillance video coding technology. Compared with the existing coding standards that were basically developed for generic videos, surveillance video coding should be designed to make the best use of the special characteristics of surveillance videos (e.g., relative static background). To do so, this paper first conducts two analyses on how to improve the background and foreground prediction efficiencies in surveillance video coding. Following the analysis results, we propose a background-modeling-based adaptive prediction (BMAP) method. In this method, all blocks to be encoded are firstly classified into three categories. Then, according to the category of each block, two novel inter predictions are selectively utilized, namely, the background reference prediction (BRP) that uses the background modeled from the original input frames as the long-term reference and the background difference prediction (BDP) that predicts the current data in the background difference domain. For background blocks, the BRP can effectively improve the prediction efficiency using the higher quality background as the reference; whereas for foreground–background-hybrid blocks, the BDP can provide a better reference after subtracting its background pixels. Experimental results show that the BMAP can achieve at least twice the compression ratio on surveillance videos as AVC (MPEG-4 Advanced Video Coding) high profile, yet with a slightly additional encoding complexity. Moreover, for the foreground coding performance, which is crucial to the subjective quality of moving objects in surveillance videos, BMAP also obtains remarkable gains over several state-of-the-art methods.**

***Index Terms*— Surveillance video, background modeling, background difference, background reference, block classification.**

I. Introduction

ACCORDING to a recent report from IDC [1], by 2020, as much as 5,800 exabytes of surveillance videos will be stored, transmitted and analyzed. Traditionally, the video coding standards such as MPEG-4 and H.264/AVC (MPEG-4 Advanced Video Coding) [2] that were originally designed for generic videos are widely used to compress surveillance videos. If we still follow this technology roadmap, in the next several years, the growth rate of surveillance videos will be much higher than the compression rate that AVC and even the HEVC [3] can achieve. In this sense, the exponential growth of surveillance videos presents an unprecedented challenge for high-efficiency surveillance video coding technology.

To address this challenge, one key point is how to make the best use of the special characteristics of surveillance videos (e.g., relatively fixed background in a period [4]) to design surveillance video coding. This is mainly due to the fact that most of surveillance videos are often captured by stationary cameras that always stand towards the same scene for a long time. Here the background is a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminance conditions and geometry settings [5].

In existing works, there are three categories of methods that were proposed to utilize these characteristics to improve the compression efficiency. The first one is the model-based coding methods [6]–[10], which model the objects-of-interest and then encode the model parameters and the remaining contents. Since it is difficult to utilize one or a set of parametric models to perfectly characterize the diverse objects in complex scenes, the object-oriented methods [11]–[14], [23]–[27] thus follow a slightly different technical solution. Namely, they segment the foreground objects from the background and encode them separately. However, pixel-level accurate foreground segmentation is still an open problem even in the field of computer vision. Moreover, a large number of bits are needed to represent the objects' borders. To address these problems, the hybrid block-based coding techniques [4], [28]–[37] have attracted much more attention for surveillance video coding in recent years. In this framework, it is natural to utilize high-quality background frames to improve the prediction efficiency for surveillance videos. Following this idea, the long-term key-frame based coding (shortly as LKC) [30]–[32], [44] and the background prediction based coding (shortly as BPC) [33], [34] methods are proposed to remarkably improve the performance, respectively by utilizing the high-quality key-frames (BKF) or the background frame generated from the reconstructed frames (BRF) as the long-term reference. For more readability, Table I shows a list of abbreviations used in this paper.

In surveillance video coding, one of the key factors affecting the compression efficiency is the so-called "exposed

Manuscript received February 26, 2013; revised September 20, 2013; accepted November 19, 2013. Date of publication December 11, 2013; date of current version January 9, 2014. This work was supported in part by the Chinese National Natural Science Foundation under Contracts 61035001, 61390515, and 61121002, and in part by the National Basic Research Program of China under Contract 2009CB320900. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Joan Serra-Sagrista.

The authors are with the National Engineering Laboratory for Video Technology, School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: tjhuang@pku.edu.cn; yhtian@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.2013.2294549

TABLE I
TERMINOLOGY AND ABBREVIATIONS

Abbr.	Description
BOF	The background data modeled from the original input frames.
BRF	The background data modeled from the reconstructed frames
BKF	The high-quality key-frame as the background frame
BDP	Background Difference Prediction
BRP	Background Reference Prediction
SRP	Short-term Reference Prediction
LKC	Video coding method utilizing BKF as the long-term reference
BPC	Video coding method utilizing BRF as the long-term reference
BDC	Directly coding the difference between input frames and BOF.
BMAP	Background-modeling based adaptive prediction method

background regions." These regions appear in the current frame being coded but are covered by objects in the reference frames. As shown in Fig. 1, for example, the exposed background regions are covered by the moving objects in the reference frames (i.e., the key-frame, and the recently decoded frame). In this case, the encoder cannot find the matched regions in these reference frames. Thus when using LKC, more bits must be paid to encode these exposed background regions. Despite BPC can boost the prediction efficiency somewhat using BRF from the reconstructed frames as the reference, the quality of both the reconstructed frames and the generated BRF cannot be guaranteed due to the quantization loss. Besides, the background generation process inevitably increases the decoding complexity since it must be embedded into the video decoder. To solve these problems, our previous work [4] (called BDC) and Paul's work [36] utilized the reconstructed background that was modeled from the original input frames (referred to as BOF) as the reference. Nevertheless, in BDC [4], directly subtracting BOF from input frames would inevitably reduce the dependency among foreground pixels in the input frames (this problem was denoted as *foreground pollution* in [46]); whereas in [36], BOF is only utilized to replace the original second reference, which is highly useful when foreground pixels take a certain proportion in the current frame. As a consequence, both the ways to utilize BOF would lead to a notable decrease of the prediction efficiency of foreground pixels.

In this study, we first conduct two analyses on how BOF can be used to improve the background and foreground prediction efficiency in surveillance video coding. On one hand, we perform an experimental analysis to validate whether BOF is a better long-term reference than BRF and BKF. On the other hand, we theoretically derive some cases where the foreground-background-hybrid blocks can be encoded more efficiently after subtracting their corresponding data in BOF.

Inspired by the above analysis results, we propose a Background-Modeling based Adaptive Prediction (BMAP) method for surveillance video coding. Its basic idea is to adaptively adopt different prediction methods for the current data according to the block classification results. To do so, BMAP firstly generates a BOF for every group of input frames, using the no-delay and one-frame buffered running average method [45]. This BOF will be encoded using a small quantization parameter (QP) and used as the long-term reference in the following encoding process. With the BOF, each block in an input frame is then classified into three categories: *background block* (GB), *foreground-background-hybrid block* (HB) and *foreground block* (FB). After that, three prediction modes are designed for the three categories of blocks, including: (1) the traditional *short-term reference prediction* (SRP) that utilizes the recently decoded data as the reference, (2) the proposed *background reference prediction* (BRP) that uses the BOF as the long-term reference, and (3) the proposed *background difference prediction* (BDP, also called the prediction in the background difference domain) in which the inter prediction is performed after subtracting the background data from both the current block and its reference. Specifically, the GBs are encoded using either SRP or BRP, and the FBs are only with SRP; while for the HBs, BRP, SRP or BDP can be used, mainly dependent on a fast decision algorithm. In practice, because BRP is implemented by replacing the last reference in SRP by BOF, the fast decision is only designed between BRP and BDP. In this way, BRP improves the background prediction efficiency for GBs by using the high-quality BOF as the reference; meanwhile BDP can improve the foreground prediction efficiency by finding the more accurate reference for HBs in the background difference domain. Moreover, such a selection among SRP, BRP and BDP will effectively avoid both the foreground pollution problem [46] and potential blocking artifacts. For each block category, only a subset of inter prediction partitions (e.g., 4×4, 4×8, 8×4, 8×8, 8×16, 16×8 and 16×16 in AVC and $2N \times 2N$, $2N \times N$, $N \times 2N$ and $N \times N$ in HEVC) are utilized in these prediction modes. As such, BMAP only suffers a slight increase in the encoding and decoding complexity.

Extensive experiments are carried out to evaluate the performance of BMAP on three datasets, each containing several long surveillance video sequences with diversified

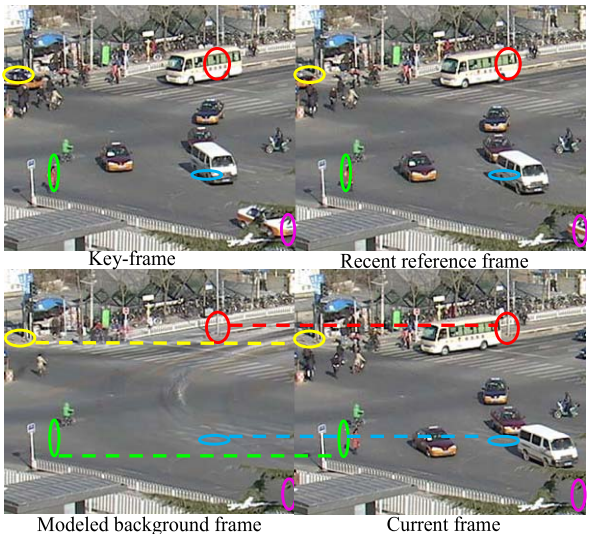

Fig. 1. The exposed background regions in the current frame, marked by color ellipses, can only match the corresponding regions in the modeled background frame, as the same locations in the key-frame or the recent reference frame are covered by the moving objects.

foreground/background distributions and foreground motion characteristics. These datasets include eight common-used CIF and SD surveillance sequences in [4], [41] four SD videos from TRECVID Surveillance Event Detection Task, and two HD surveillance videos from Hisense Co. Ltd. Experimental results on SD/CIF videos indicate that BMAP averagely saves about half of the total bit-rate for both IBBP and IPPP. As for the foreground coding performance, BMAP achieves 0.87/1.21 dB foreground coding gains over AVC High Profile for IBBP and 1.13/1.50 dB for IPPP on SD/CIF videos. Even on the other two datasets with busy traffic or crowded people, BMAP also significantly reduces the total bit-rate and obtains foreground coding gains. In addition, BMAP outperforms BPC, LKC and BDC remarkably.

The rest of this paper is organized as follows. Section II briefly reviews the related works. Section III presents the analysis. Section IV describes the proposed BMAP method. Experimental results are given in Section V, and Section VI concludes this paper.

II. Related Work

Among the existing surveillance video coding solutions, the model-based coding can be traced back to the early method [6] for video conferencing. Following this pioneer work, many studies such as [7]–[10] were devoted to utilizing the synthesized models of faces or heads in video coding. This technology roadmap was not changed until Musmann et al. [11] proposed the object-oriented-analysis-synthesis coding, in which a video is coded with the motion, shape and color information of the objects, as well as prediction residuals. After that, the works in [12]–[14] further developed the object-oriented coding methods for the surveillance videos with few foreground objects. Following the object representation techniques in MPEG-4 [2], as well as the more accurate object detection and segmentation methods via background modeling [15]–[22], the work in [23] proposed an efficient video coding scheme based on region segmentation. To obtain larger storage efficiency for surveillance videos, Vetro et al. [24] focused on the coding of segmented foreground objects, whereas neglecting the background variations. However, this severely degraded the coding results in terms of objective quality metrics (e.g. PSNR). To solve the problem, Babu et al. [25] and Hakeem et al. [26] encoded background residuals in the hybrid block-based coding framework. They also proposed to encode the prediction difference between the object representations in adjacent frames, together with the residual data generated from object prediction. Towards a more efficient residual coding, Venkatraman et al. [27] utilized the direct and transform based compressive sensing information to represent the sparse signal-residual object error. Overall speaking, the main challenges for model-based and object-oriented methods are accurate foreground segmentation, low-cost object representation and high-efficiency foreground residual coding.

Different from model-based and object-oriented methods, hybrid block-based surveillance video coding methods follow the normal hybrid coding framework by encoding frames block by block. These methods can be classified into three categories: region-based hybrid coding, long-term key-frame based coding and background prediction based coding. Among them, the region-based coding mainly focuses on achieving better subjective quality of foreground regions with low encoding complexity, while keeping the total bitrate nearly unchanged. For example, the work in [28] reduced the complexity by coding background blocks with fewer modes and foreground regions with most bits. The long-term key-frame based coding (LKC) is engaged to improve the compression efficiency by using the high-quality encoded key-frame as the long-term reference for the frames that follow. The long-term reference mechanism, firstly proposed by T. Wiegand et al. [30] and accepted by AVC, is an effective tool for sequences with few scene shots. After that, many works (see [31] and [32]) were developed to select a better reference among the short-term and long-term reference frames. For surveillance videos, there are always several "key-frames" that can well represent the video scene in a period. Therefore, Ding et al. [44] utilized the high-quality encoded key-frame as the long-term reference to improve the coding performance. Such a method was also used in our previous work [4] as the anchor JM-OPT for the performance evaluation. It periodically encoded a key-frame using high-quality intra-coding (with a small QP), which was then used as the long-term reference for the frames that follow. Note that, the work [4] utilized the number of changed blocks contrast to the background frame to determine when to encode a high-quality key-frame; whereas the method in [44] utilized the scene change between adjacent frames to update key-frames.

To further improve the efficiency of coding the exposed background regions, the background prediction based coding (BPC) was proposed in [33] and [34]. One common feature of [33] and [34] is to exploit the reconstructed frames to generate the background. Despite the generation process is very efficient, the quality of the generated background cannot be effectively guaranteed, especially in low-bit-rate video coding. In addition, this process should be embedded into the video decoder, consequently leading to a notable increase of the decoding complexity. Nevertheless, the two works also enlighten us to improve the compression efficiency of the "exposed background regions" by using a better modeled background as the reference in a low-complexity way. Thus in [4] and [36], the background frame is modeled from the original input frames and then encoded into the final stream. This so-called BOF can be utilized for better prediction.

However, there are still several problems in [4] and [36]. Firstly, the mean-shift algorithm was used in [4] and Gaussian mixture model was employed in [36] to construct BOF. Often, these methods required a large amount of memory and many float operations, making them not applicable in hardware implementation. Secondly, there should be a better way to utilize the modeled BOF. That is, BOF was used in [4] to calculate the difference frames for high-complexity 9-bit coding. This would inevitably reduce the dependency among foreground pixels in input frames, leading to the so-called

TABLE II
MAIN NOTATIONS IN THIS PAPER

Notation	Meaning
BG, BC	The BOF and the background data of the current block
I, C,ΔC	The input frame, the current block and the difference data between the current block C and its background BC
$W_{x,y}$	The searched reference block at position (x, y)
Ref	The short-term reference of the current frame
ΔRef	The difference data between Ref and BG
R, ΔR	The block in Ref (or ΔRef) that best matches C (or ΔC)
J_S,J_D,J_B	Minimal rate-distortion costs of coding C with Ref as the reference in SRP, coding ΔC with ΔRef as the reference in BDP, and coding C with BG as the long-term reference in BRP

foreground pollution problem [46]. Meanwhile, BOF was used in [36] to replace the second reference. This means to disable the original second reference, which is highly useful when foreground pixels take a certain proportion in the current frame. Therefore, the two ways to utilize BOF would lead to a notable decrease of the performance in foreground coding. To address these problems, this study will investigate whether and how BOF can be used to improve the background and foreground prediction efficiency so as to increase the coding performance.

III. ANALYSIS

As discussed above, the key to improve the performance of background and foreground coding is to make use of the visual characteristics of surveillance videos. In general, background prediction efficiency relies on the quality of the generated background used as the reference, while foreground coding performance depends on how to utilize the background to reduce the distortion of foreground prediction. Although the idea is very straightforward, there is no detailed analysis so far on what is the optimal background and how to utilize such a background to obtain better prediction efficiency. This section firstly experimentally validates that BOF is the optimal long-term reference for high-efficiency background coding. Then several conditions are derived to guarantee that coding the blocks in the background difference domain can improve foreground prediction efficiency. To begin with our discussion, Table II lists some main notations used in this paper.

A. Why BOF is Optimal for Efficient Background Prediction

To avoid taking the multiple long-term references and bit-allocation problems into account, we only employ one long-term reference in the multi-reference prediction structure. Moreover, if needed, the long-term reference is quantized with QP $=$ 4 for the least quantization loss. Note that in BPC, BRF is a clean modeled background but trained from the quantization-lossy reconstructed frames; while in LKC, BKF is encoded with high quality but may be not a clean background. On the contrary, BOF should have the advantages of both BRF and BKF because it is a clean background modeled from the original input frames without quantization loss. In this subsection, we will conduct some experiments to validate this conjecture.

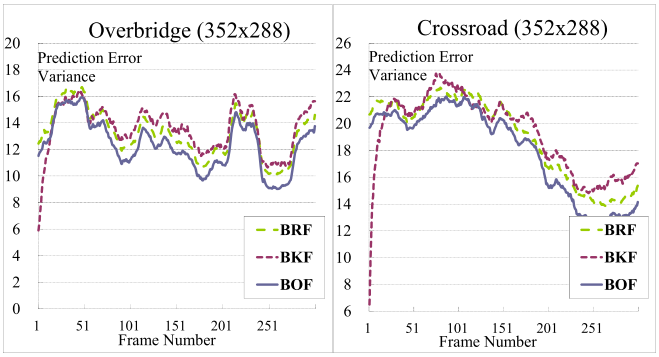

Fig. 2. The PEV curves for each frame using three long-term reference frames, namely BRF, BKF and BOF.

As in [35], [37], and [38], the reference with less prediction distortion and smaller power spectral density $\Phi(\Lambda)$ can help achieve better rate-distortion performance for an input sequence Λ; moreover, less prediction error variance (PEV) always leads to less power spectrum of the residual noise $\Phi_{\rm nn}(\Lambda)$, and consequently determines the $\Phi(\Lambda)$. Thus it can be concluded that the PEVs using BRF, BKF and BOF to predict Λ is consistent with their distortion or $\Phi(\Lambda)$ to a large extent. Fig. 2 illustrates the comparison results of PEVs when coding two surveillance videos, Crossroad and Overbridge (352×288) using the three long-term frames as reference, respectively. We can see that, after several initial frames, PEV when using BOF becomes less than those of BKF and BRF. Moreover, the PEV gap between BOF and BKF/BRF becomes larger and larger as the number of frames increases. Since no scene change happens, the total PEV when using BOF is definitely smaller than BKF and BRF. This is because BOF contains much more higher-quality background pixels and less noise or foreground pixels. Note that the background can be updated once every hundreds of frames, so the bit cost of BOF is negligible for a long surveillance video. In summary, it is reasonable to utilize BOF as the optimal long-term reference.

B. How to Improve Foreground Prediction Using BOF

B.1 Why More Efficient Prediction is Desired for Foreground

In the traditional hybrid coding such as AVC and HEVC, block-matching based prediction is effective to reduce prediction residual [3]. However, one single motion vector for each prediction partition cannot well represent the motion characteristics of all its inner pixels. This is especially true for surveillance videos, in which the background and foreground pixels in a prediction partition usually have different motion characteristics (e.g., the static background vs. the moving foreground objects). Thus if a single motion vector were used for both background and foreground pixels in a prediction partition, a large quantity of prediction residuals would be inevitably produced. This fact can be further illustrated by Fig. 3. We can see that only the foreground pixels of block A, rather than its background pixels, can well match the searched block A' in the reference frame. In this case, larger prediction residuals will be produced for the background pixels. On the contrary, the background pixels for block B, rather than its

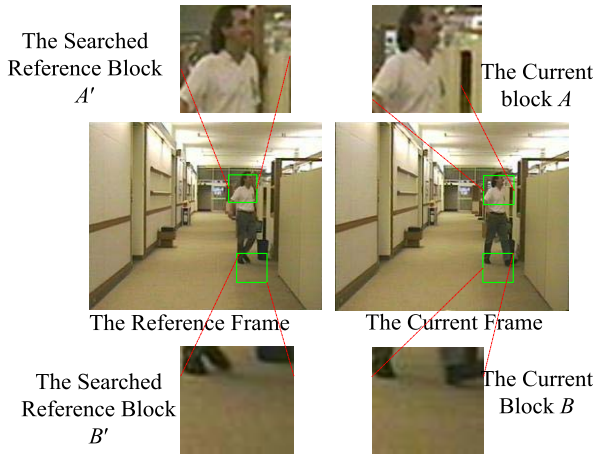

Fig. 3. An example of the imperfect matching result for block-matching based prediction in surveillance video coding.

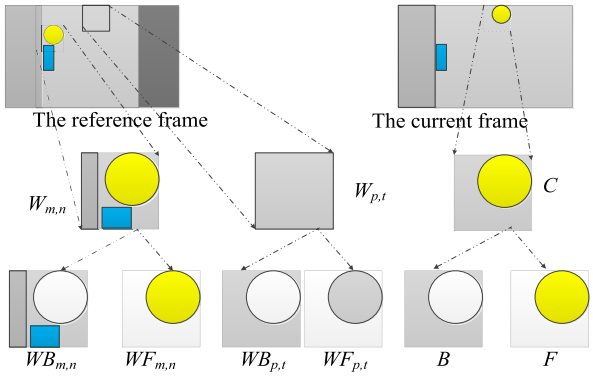

Fig. 4. The calculation of F/B and $WB_{x,y}/WF_{x,y}$. White regions are pixels with 0 values. F and B denote the foreground and background parts of C, whereas $WB_{x,y}/WF_{x,y}$ denotes the foreground and background parts of $W_{x,y}$.

foreground pixels, can well match the block B' in the reference frame. Similarly, this will produce larger prediction residuals for the foreground pixels in B. Therefore, both cases will result in larger prediction distortion and rate cost.

More formally, let C be the current block, $W_{x,y}$ be the searched block at position (x, y), and $Ref(x, y)$ be the pixel at (x, y) in the reference frame *Ref*. Then the searched block R can be formulated from SAD (sum of absolute difference) by

$$R = \arg\min_{W} \{SAD(C - W_{x,y}) | W_{x,y}(i, j) = Ref(x+i, y+j)\}. \tag{1}$$

Suppose F and B are matrices for the foreground and background pixels in C (as shown in Fig. 4). Similarly, $WF_{x,y}$ and $WB_{x,y}$ are those for $W_{x,y}$. Then the equation can be re-written as

$$\begin{aligned} SAD(C-R) &= \min\left\{SAD(F-WF_{x,y})+SAD(B-WB_{x,y})\right\}, \\ &s.t.\ F+B=C, \\ &WF_{x,y}+WB_{x,y}=W_{x,y}, \end{aligned} \tag{2}$$

where for each element (i, j) in matrices F, $WF_{x,y}$, C and $W_{x,y}$

$$(F(i, j), WF_{x,y}(i, j)) = \begin{cases} (C(i, j), W(i, j)), & if\ C(i, j)\ is\ foreground; \\ 0, & otherwise. \end{cases}$$

From Eq. 2, we can get

$$\begin{aligned} SAD(C-R) &= \min\left\{SAD(F-WF_{x,y})+SAD(B-WB_{x,y})\right\} \\ &\geq \min\left\{SAD(F-WF_{x,y})\right\}+\min\left\{SAD(B-WB_{x,y})\right\}. \end{aligned} \tag{3}$$

In this equation, $\min\{SAD(C-R)\}$ denotes the minimal distortion for block-matching based prediction, while $\min\{SAD(F-WF_{x,y})\}$ and $\min\{SAD(B-WB_{x,y})\}$ represent the minimal foreground and background prediction distortion that we can get. In Fig. 4, $\min\{SAD(F-WF_{x,y})\} = SAD(F-WF_{m,n})$, $\min\{SAD(B-WB_{x,y})\} = SAD(B-WB_{p,t})$. As a result, $\min\{SAD(C-R)\} \geq \min\{SAD(F-WF_{x,y})\} + \min\{SAD(B-WB_{x,y})\}$. Therefore, $SAD(C-R)$ is probably not the minimal distortion that we can obtain. That is, a larger distortion may be produced for these HBs (i.e., foreground part F and background part B co-exists in the current block C), consequently leading to a worse foreground prediction efficiency.

B.2 Why and When the Prediction in the Background Difference Domain Can Reduce Foreground Prediction Distortion

As discussed above, the BRP with BOF as the long-term reference cannot provide a good prediction for foreground pixels in HBs. Thus SRP and BDP are two possible prediction methods. But for SRP, the different motion characteristics of background pixels B and foreground pixels F in the current HB C will cause a large prediction distortion. In this subsection, we will prove that BDP can obtain less distortion for HBs at several conditions, by comparing the prediction distortions in SRP and BDP.

Similar to C that is composed of F and B, BC (i.e., the data in BOF co-located with C) is divided into BCF (the data in BC co-located with F) and BCB (the data in BC co-located with B):

$$BC = BCF + BCB,$$

where

$$BCF(i, j) = \begin{cases} 0, & C(i, j)\ is\ background; \\ BC(i, j), & Otherwise. \end{cases} \tag{4}$$

According to the pixel distribution of C, $BG_{x,y}$ (i.e., background block of $W_{x,y}$) can also be divided into $BGF_{x,y}$ and $BGB_{x,y}$ by

$$BG_{x,y} = BGF_{x,y} + BGB_{x,y},$$

where

$$BGF_{x,y}(i, j) = \begin{cases} 0, & if\ C(i, j)\ is\ background; \\ BG_{x,y}(i, j), & Otherwise. \end{cases} \tag{5}$$

Then Theorem 1 summarizes the conditions when BDP works.

Theorem 1. Let $SAD(\Delta C-\Delta R)$ denote the prediction distortion in BDP, and $SAD(C-R)$ is that in SRP. For each block with foreground pixels, if the best matched block R is the $W_{m,n}$ at position (m, n) in the reference frame, then each

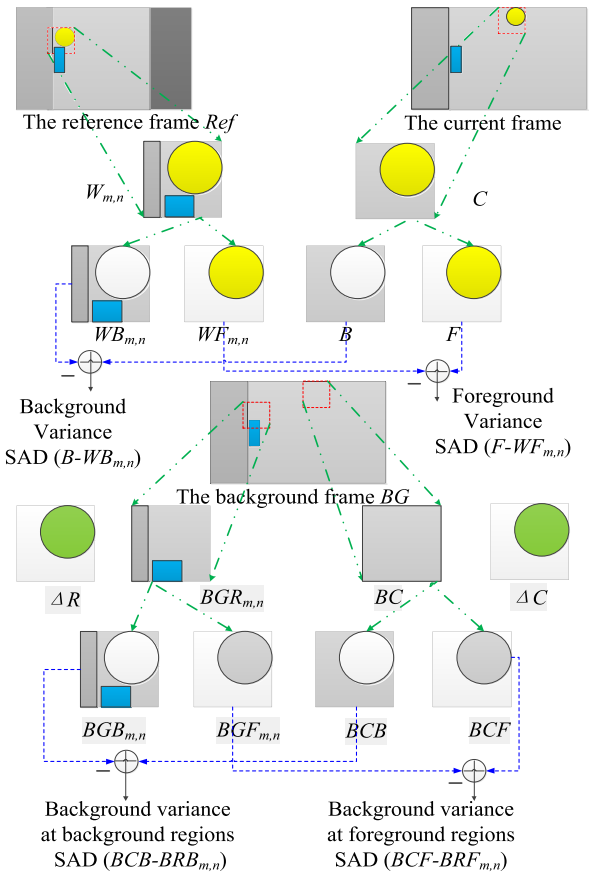

Fig. 5. An example satisfying condition (1) in Theorem 1. In this case, ΔC and ΔR are better matched with C and R. That is, the prediction in background difference domain produces less distortion. This figure also shows the calculation procedures for F–$WF_{m,n}$, B–$WB_{m,n}$, BCF–$BGF_{m,n}$ and BCB–$BGB_{m,n}$.

of the following inequalities makes $SAD(\Delta C-\Delta R) < SAD(C-R)$:

$$\begin{aligned}
&1) F = WF_{m,n} \text{ and } SAD(BCF-BGF_{m,n}) < SAD(B-WB_{m,n}).\\
&2) B = WB_{m,n} \text{ and } SAD(F-BCF) < SAD(F-WF_{m,n}).\\
&3) SAD(F-WF_{m,n}-(BCF-BGF_{m,n})) < SAD(F-WF_{m,n}).
\end{aligned} \tag{6}$$

The proof of Theorem 1 is given in the Appendix. Fig. 5 presents an example for the condition (1), which makes $SAD(\Delta C-\Delta R) < SAD(C-R)$. Here F–$WF_{m,n}$ is the foreground difference between the current block and the searched reference block, B–$WB_{m,n}$ is the background difference between the current block and the searched reference block. BCF–$BGF_{m,n}$ and BCB–$BGB_{m,n}$ are shown in a similar way.

Intuitively, Theorem 1 indicates that, BDP will produce less prediction residual for HBs at following three conditions: (1) the foreground of the current block is matched with the searched block R, and SAD of the background difference at foreground regions of C and R (i.e., BCF–$BGF_{m,n}$) is less than SAD of the background difference between C and R (i.e., B–$WB_{m,n}$); (2) the background of the current block is matched with the searched block R, and SAD of the difference between the foreground of C and its background is less than SAD of the foreground difference between C and R; and (3) the background difference at foreground regions of C and R is not negatively correlated to their foreground difference. Note that foreground and background pixels never co-exist in FBs or GBs. Therefore, BDP should be only used for HBs.

IV. The Proposed Method

Based on the analyzed results, we propose a Background Modeling based Adaptive Background Prediction (BMAP) method. To begin with our discussion, we define the following notations: Θ, Π and Ω are sets of all possible modes for SRP, BRP and BDP, respectively; J_S, J_B and J_D are the corresponding minimal rate-distortion costs (RDCosts); Rec is the reconstructed result of the prediction residuals Res for C or ΔC in the background difference domain; Rec_f is the finally reconstructed result of the current block. Taking AVC high profile as an example, the universal set of inter prediction partitions E is $\{16\times16, 16\times8, 8\times16, 8\times8, 8\times4, 4\times8, 4\times4\}$. Obviously, the initial sizes of Θ, Π and Ω equal to that of E. Let Θ_S, Π_S and Ω_S denote the candidate partitions for SRP, BRP and BDP of the block category S where $S\epsilon\{\text{FB, GB, HB}\}$. For example, Θ_{FB} is the set of available prediction partitions for SRP.

Generally, let $K(M, A, B)$ represent the following prediction procedure: employing the set of modes M to predict the matrix A when using matrix B as the reference. Then, supposing BG is the reconstructed result of BOF, the best prediction result K^* for each current block C in BMAP is calculated by

$$K^* = \begin{cases}
\mathrm{K}(\Theta_{FB}, C, Ref), & if\ S = FB;\\
\mathrm{K}(\Theta_{GB}, C, Ref), & if\ S = GB \text{ and } J_S \le J_B;\\
\mathrm{K}(\Pi_{GB}, C, BG), & if\ S = GB \text{ and } J_S > J_B;\\
\mathrm{K}(\Theta_{HB}, C, Ref), & if\ S = HB\\
& \quad \text{and } J_S \le \min\{J_B, J_D\};\\
\mathrm{K}(\Pi_{HB}, C, BG), & if\ S = HB \text{ and } J_B < J_S\\
& \quad \text{and } J_B \le J_D\};\\
\mathrm{K}(\Omega_{HB}, \Delta C, \Delta Ref), & if\ S = HB\\
& \quad \text{and } J_D < \min\{J_B, J_S\}.
\end{cases} \tag{7}$$

This equation shows that the selection should be made among predictions for each block according to the minimal RDCost J_S, J_B, J_D of their available partition sets in SRP, BRP and BDP. For example, the candidate partition set Ω_{HB} for BDP is selected only when the current block is HB and J_D is less than J_B and J_S.

It should be noted that, several problems remain open in Eq. 7, including how to generate BOF, how to classify each block, how to utilize BOF for BRP, how to use it to calculate and reconstruct ΔC and ΔRef for BDP, and how to calculate the candidate partitions for the mode decision process, etc. The implementation details about these problems will be discussed in the rest of this section.

A. The Codec Framework

Fig. 6 illustrates the overall framework of BMAP. The encoding process is described as follows:

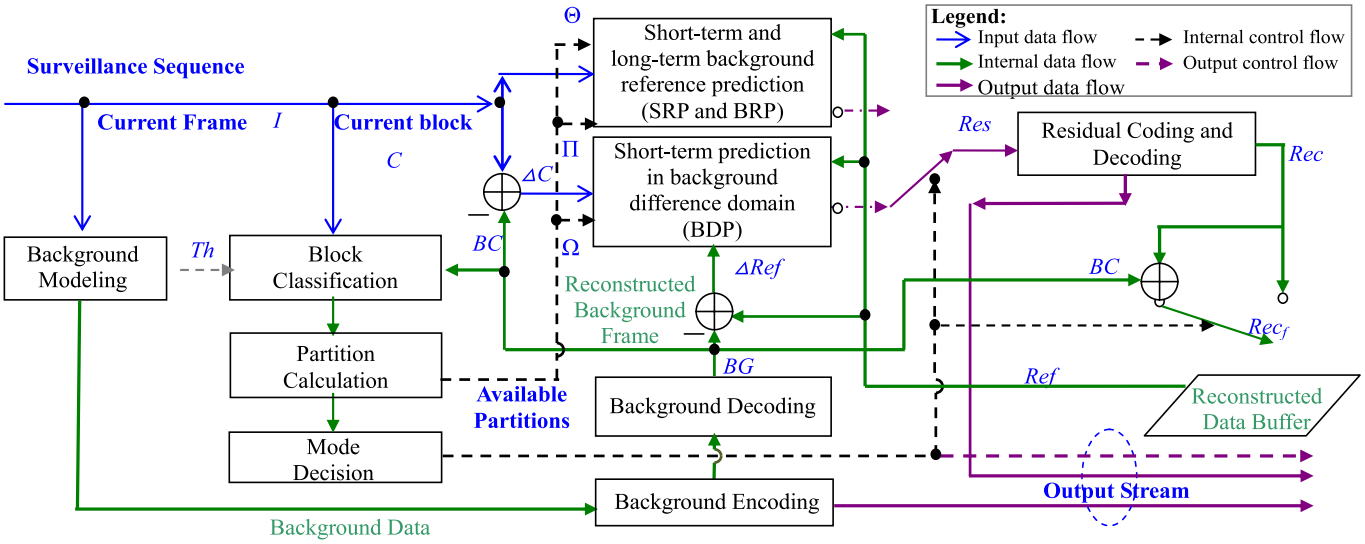

Fig. 6. The framework of BMAP.

Step 1. **Initialize or Update the Background**. At the very beginning, the BOF (denoted by *BG*) is initialized as the first reconstructed frame. Afterwards, for each original input frame (denoted by *I* in Fig. 6), the *Background Modeling* module is used to update the background. Such a background is encoded by the *Background Encoding* module. Then the *Background Decoding* module is used to generate the decoded BOF.

Step 2. **Classify the Current Block**. For each block to be encoded (referred to as the current block), the *Block Classification* module is utilized to classify it into FB, GB or HB, according to the percentage of its background pixels. To this end, a threshold $Th_{x,y}$ is used to judge the difference between C and BC at position (x, y). Note that $Th_{x,y}$ is automatically calculated from the *Background Modeling* module by the algorithm shown later in Fig. 9.

Step 3. **Calculate Candidate Modes and Perform Prediction**. The *Partition Calculation* module determines all candidate partitions for each category of blocks, and predicts each block selectively using the following modes:

a) *SRP and BRP* in the original domain: Here we predict C using all candidate partitions in $\Theta_{FB}/\Theta_{GB}/\Theta_{HB}$ with the recently decoded data *Ref* from the *Reconstructed Data Buffer* as the short-term reference. Meanwhile, C is also predicted using partitions in Π_{HB}/Π_{GB} with *BG* as the long-term reference. Note that the total number of the reference frames is the same as that in the traditional video encoder.

b) *BDP* in the background difference domain: In this process, ΔC and Δ *Ref* are firstly generated for each HB by subtracting BC and *BG* respectively from C and *Ref*. Then C is predicted using partitions in Ω_{HB} with *Ref* as the reference.

Step 4. **Determine the Best Mode, and Then Encode and Reconstruct the Current Block**. The *Mode Decision* module selects the best partition from Θ_{HB}, Π_{HB} and Ω_{HB} to encode the block and the *Residual Coding and Decoding* module is used to compensate the residuals *Res*. The index of the selected partition should be written into the stream as the control data to guarantee the decoding match. If Θ_S or Π_S is selected where $S\epsilon\{FB, GB, HB\}$, the reconstructed *Rec* is directly written into the *Reconstructed Data Buffer*. Otherwise, we must add *Rec* by *BC* and write the result Rec_f into the buffer.

B. Implementation for BRP and BDP

Eq. 7 shows that, $K(\Omega, \Pi, BG)$ and $K(\Omega, \Delta C, \Delta Ref)$ can represent the encoding process when selecting BRP and BDP to predict the current block. In this part, we firstly describe the practical implementation of BRP and BDP, and then analyze the total time and the complexity of hardware implementation.

For BRP, the BOF (denoted by *BG*) is utilized to replace the original last reference frame in SRP. Therefore, the total number of the reference frames keeps unchanged. That is, BRP never takes additional motion estimation (ME), mode decision and residual decoding processes. As a result, there is no additional encoding and decoding time. For BDP, to reduce the complexity of ME and residual coding, ΔC and ΔRef should not be the matrices of 9-bit integers, but of 8-bit clipped values calculated using a function *Clip* with any matrix V as its input:

$$\Delta C = Clip(C - BC + 128),\ \Delta Ref = Clip(Ref - BG + 128),$$
$$Clip(I_{i,j}) = \begin{cases} V_{i,j}, & 0 \le V_{i,j} \le 255; \\ 0, & V_{i,j} < 0; \\ 255, & V_{i,j} > 255. \end{cases} \quad (8)$$

Note that the *Clip* function only affects the pixels having a large gap with its background values. Usually, these pixels are

the pure foreground pixels. Because neither BRP nor BDP can obtain the performance gain on the pure foreground pixels, the *Clip* function has little impact on the total bit rate.

Let $V_{i,j}$ denote the value of the element at position (i, j) in any input block V, then the final reconstructed result Rec_f for each block can be calculated by

$$Rec_f = \begin{cases} Rec, & if\ J_S \le \min(J_D, J_B); \\ Rec, & if\ J_B < J_S \\ & \quad and\ J_B \le J_D; \\ Clip\,(Rec - 128 + BC), & if\ J_D < \min(J_S, J_B). \end{cases} \tag{9}$$

This equation also shows that, to obtain Rec_f, if BDP is optimal (i.e., $J_D < \min(J_S, J_B)$), the reconstructed result of the prediction residual should be compensated with BC by $Clip(Rec - 128 + BC)$.

To save the total computational time, an early prediction comparison before residual coding (referred to the subsection E) will be performed in the mode decision process so as to make a selection among SRP, BRP and BDP. Among them, BRP suffers no increase in complexity, since it is implemented by replacing the last reference in SRP; while BDP requires some additional time to perform ME on ΔC. However, the ME always takes little time in the encoders like AVC and HEVC, so the increase in the total encoding time is insignificant.

Moreover, the hardware implementation complexity can also be saved. Also because BRP is just implemented by replacing the last reference frame with BOF in SRP, no additional logic units should be added for SRP in hardware implementation. Thereby the main additional logic units are used for the implementation of BDP. But even for BDP, no additional RAMs are necessarily required to buffer ΔRef and ΔC. This is due to the following implementation: SRP and BDP have the same input data; whereas some additional logic units are used in BDP to subtract BC from C, subtract BR from Ref for each pixel in the ME and add BC to the Rec after residual decoding.

C. Background Modeling

Actually, different kinds of background generation and updating algorithms can be used for background modeling in BMAP, such as the GMM [39] and mean-shift [40]. However, these methods often require a number of buffering frames for modeling and fraction-point calculation. This presents a challenge for the hardware implementation of surveillance video codecs. To avoid the problem, the running average method [45], which estimates the average pixel values as the background pixels in a running way, is used for background generation in BMAP. Let I denote the current training frame and a matrix A with unsigned 8-bit integers be the previous average result for all the pixels, then the algorithm calculates the current result by $A = (A \times (n-1) + I + (n \gg 1))/n$, where n is the number of the training frames. The method only requires one buffered frame to store A. Each time given a training frame, only one multiply, three add, one shift, one divide and one floor operations are used. To guarantee the decoding match, any modeled background used in BMAP must be encoded into the stream as a non-display frame. Such non-display frame mechanism has been supported by the NAL unit syntax structure in the AVC/HEVC high level syntax. It should be noted that, bits for coding the background frames have been counted into the final bitrate in our experiments.

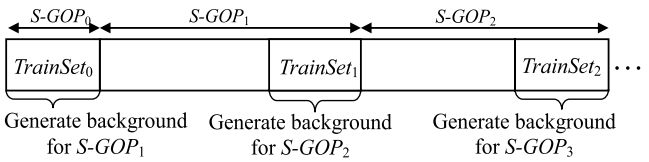

Fig. 7. Sequence structure for background generation.

Fig. 7 describes the sequence structure for background generation and updating. In this structure, the background frame or key-frame is generated S-GOP (i.e., super-group of pictures) by S-GOP. That is, an initial group of frames are utilized as $TrainSet_0$ to generate the background frame for $S\text{-}GOP_1$, whereas the last group of frames in $S\text{-}GOP_1$ are utilized as $TrainSet_1$ to generate the background frame for $S\text{-}GOP_2$, and those in $S\text{-}GOP_2$ are utilized as $TrainSet_2$ to generate that for $S\text{-}GOP_3$, ... Note that the first frame in the sequence is treated as the background frame for coding $TrainSet_0$, and the first $TrainSet$ is regarded as $S\text{-}GOP_0$. In this way, at the initial stage of encoding each S-GOP, the corresponding background frame has been generated during the encoding process of the previous S-GOP. Therefore, BMAP can encode its frames without delay. In our experiments, each $TrainSet$ has 120 frames and the size of one S-GOP is 900.

D. Block Classification

As discussed above, BMAP employs different prediction modes for different categories of blocks. Therefore, a low-complexity classification algorithm should be designed to classify blocks into GB, HB and FB. In practice, an adaptive threshold $Th_{x,y}$ is calculated for each block at position (x, y) to judge its category S. Given $Th_{x,y}$, S is calculated by

$$S = \begin{cases} FB, & \left\| \{(m,n) \mid |\Delta C(m,n) - 256| < Th_{x,y}\} \right\| / \\ & Sizeof(block) < \alpha; \\ GB, & \alpha \le \left\| \{(m,n) \mid |\Delta C(m,n) - 256| < Th_{x,y}\} \right\| / \\ & Sizeof(block) < \beta; \\ HB, & \left\| \{(m,n) \mid |\Delta C(m,n) - 256| < Th_{x,y}\} \right\| / \\ & Sizeof(block) \ge \beta. \end{cases} \tag{10}$$

where (m, n) is the pixel position in C. Eq. 10 means the category of a block is determined by the percentage of its background pixels. In practice, we set $\alpha = 5/64$ and $\beta = 50/64$ for the 16×16 block. Fig. 8 illustrates two classification examples.

Then the remaining problem is how to adaptively calculate the threshold. To identify foreground pixels in a new frame, a feasible idea is to calculate $Th_{x,y}$ using the root-mean-square deviation σ of the difference values between two kinds of pixels: the "potential background pixels" identified by the $Th_{x,y}$ in the previous frame and their corresponding pixels

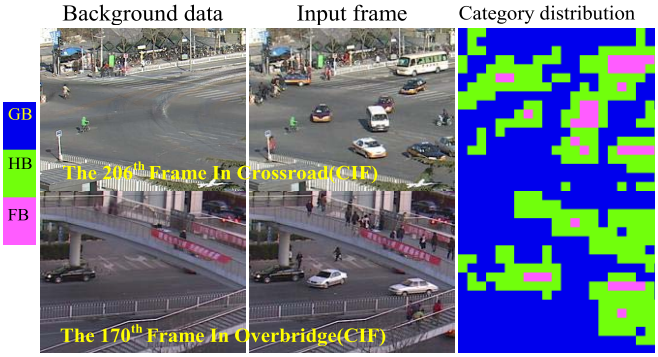

Fig. 8. Examples of the block category distributions for two sequences, crossroad (CIF) and overbridge (CIF).

Input:

$I(m, n)$: the pixel value at position (m, n) of the block at position (x,y) in the current frame.

$Bg(m, n)$: the background pixel that corresponds to $I(m, n)$.

Initialization:

$Th_{x,y}$ is initialized to $Th_{x,y}$ of the block at position (x,y) in the previous frame, or 14 for the first frame

Calculation:

1. For each $0 \le m, n \le 15$, calculate

$$Diff(m,n) = \left| I(m,n) - B(m,n) \right|,$$

2. For each (m, n), detect the potential background pixels by

$$Cmp(m,n) = \begin{cases} 1, & Diff(m,n) \le 2 \times Th_{x,y}; \\ 0, & Diff(m,n) > 2 \times Th_{x,y}. \end{cases}$$

3. Count the number of the potential background pixels by

$$Sum = \sum_{m,n} Cmp(m,n), \quad 0 \le m, n \le 15$$

4. Calculate the root-mean-square deviation as the updated $Th_{x,y}$ for the current block

$$Th_{x,y} = \sqrt{Round\left(\left(\sum_{m,n} Cmp(m,n) \times Diff^2(m,n) \right) \Big/ Sum \right)},$$

where $Round(A)$ denotes the round value of A.

Output: $Th_{x,y}$

Fig. 9. The threshold updating algorithm.

in the background. In practice, we set $Th_{x,y}$ equal to 2σ for each block to guarantee that as few foreground pixels as possible are identified as background pixels. Following this, we propose an adaptive threshold updating algorithm. As shown in Fig. 9, the threshold calculating process for each block includes four steps: (1) Calculate the difference between the current block and its background; (2) Utilize $2 \times Th_{x,y}$ in the previous frame to identify the "potential background pixels" in the current block; (3) Count the number of the potential background pixels in the current block; and (4) Calculate the root-mean-square deviation σ and use it to update the current threshold.

E. Mode Decision Algorithm

Based on the results of block classification, the mode decision algorithm in BMAP includes two steps, i.e., candidate partition calculation and fast mode decision.

E.1 Candidate Partition Calculation

In FBs, most pixels belong to foreground with different motion characteristics. Thus all the prediction partitions in Θ for SRP are used to avoid reducing the prediction efficiency over the traditional methods such as that in AVC. In GBs, there are few foreground pixels, and almost all pixels share the same motion characteristics. Thus it is reasonable to only enable large inter prediction partitions in Θ_{GB} and Π_{GB} (e.g., 4×4, 4×8, 8×4 and 8×8 in AVC are not necessary). For HBs, BDP contributes much to the efficiency. In practice, for the smaller prediction partitions, e.g., 4×4, 4×8, 8×4 in AVC, there is a low probability for the corresponding blocks containing both foreground and background pixels. Thus only the larger inter prediction partitions (e.g., 8×8, 16×16, 16×8 and 8×16) should be included in Ω_{HB} and Π_{HB} for HBs. Let $\Theta_{i,j}/\Pi_{i,j}/\Omega_{i,j}$ respectively denotes the modes with size of $i \times j$ in the $\Theta/\Omega/\Pi$, then the candidate partitions can be calculated by

$$\begin{cases} \Theta_{FB} = \Theta, \\ \Theta_{GB} = \{\Theta_{i\times j} | \max\{i, j\} \ge 16\}, \\ \Pi_{GB} = \{\Pi_{i\times j} | \max\{i, j\} \ge 16\} \\ \Theta_{HB} = \Theta, \\ \Omega_{HB} = \Omega - \{\Omega_{i\times j} | \min\{i, j\} \le 4\}, \\ \Pi_{HB} = \Pi - \{\Pi_{i\times j} | \min\{i, j\} \le 4\} \end{cases} \tag{11}$$

As mentioned above, BRP is implemented by just replacing the last reference frame of SRP by BOF, so Π does not add any new candidate partition for FBs, GBs and HBs. For FBs, BMAP takes the same number of the candidate partitions with the traditional hybrid codec such as AVC. This is obvious since it only takes the candidate partition set Θ_{FB} that equals to Θ for AVC. For GBs, due to the BRP implementation, BMAP practically takes only the candidate partition set Θ_{GB}. Because Θ_{GB} only includes the larger partitions in Θ, BMAP thus takes less candidate partitions than AVC for GBs. For HBs, however, BMAP takes a larger number of the candidate partitions than AVC because not only Π_{HB} and Θ_{HB}, but Ω_{HB} are included. Overall speaking, there is a slight increase of prediction complexity in the encoding process. The next sub-section will further discuss how to reduce the encoding complexity for HBs.

E.2 Fast Mode Decision

Given the candidate prediction partitions, the left problem is how to select the best prediction mode for each category of blocks. Let $M_{FB}/M_{GB}/M_{HB}$ denote the best prediction mode for FB/GB/HB. From Eq. 7, the following strategy is used to achieve the minimal RDCost:

$$\begin{cases} M_{FB} = \arg\min\limits_{k} \{ J_k \mid k \in \Theta_{FB} \} \\ M_{GB} = \arg\min\limits_{k} \{ J_k \mid k \in \Theta_{GB} \cup \Pi_{GB} \} \\ M_{HB} = \arg\min\limits_{k} \{ J_k \mid k \in \Theta_{HB} \cup \Pi_{HB} \cup \Omega_{HB} \} \end{cases} \tag{12}$$

where J_k is the RDCost using the prediction partition k.

Note that for HBs, BMAP takes a larger number of the candidate partitions than AVC. To reduce the encoding complexity for HBs, a feasible approach is to exclude the partitions in

TABLE III
CANDIDATE PARTITION NUMBERS FOR HBS IN BMAP AND AVC

Sequence	HB number	AVC	BMAP	BMAP/ AVC
crossroad	4,638,238	32,467,666	26,940,097	**82.98%**
overbridge	3,962,775	27,739,425	23,318,964	**84.06%**

Ω_{HB} when there exists the serious foreground pollution problem in the current block. For example, a typical foreground pollution case happens when the prediction distortion using the root partition in Ω_{HB} (i.e., the largest partition in the current block to be coded, such as 16 × 16 in AVC) is significantly larger than that in Θ_{HB}. In this case, we can use Θ_{HB} rather than Ω_{HB} as the candidate partition for this block.

Let $R \times R$ denote the root partition of an input block, the optimization of M_{HB} in (12) can be re-written as in (13), shown at the bottom of the page, which is shown at the bottom of this page, where $SAD_{R\times R}$ is the prediction distortion using the $R \times R$ partition, and γ is set 10%. As such, the number of candidate partitions for HBs is less than AVC. Table III shows the candidate partition numbers of HBs in BMAP and AVC for the first 120 frames of crossroad and overbridge (SD).

V. EXPERIMENTS

A. Methodology

To evaluate the effectiveness of the proposed BMAP, extensive experiments are carried out on different kinds of surveillance videos. To compare the methods using the long-term reference, the first 1920 frames of eight long surveillance videos [4], [41] are used in the experiments. Among them, four sequences (Crossroad, Overbridge, Office and Bank) are in standard definition (SD) and four (Crossroad, Overbridge, Snowroad and Snowgate) are in CIF. These sequences cover different scenes, including bright and dusky lightness (BR/DU), large and small foreground (LF/SF), fast and slow motion (FM/SM). As shown in Fig. 10, Crossroad (SD), Overbridge (SD), Office (SD) and Crossroad (CIF) are brighter than others. Whereas in Crossroad (SD), Overbridge (SD), Office (SD) and Crossroad (CIF) and Overbridge(CIF), the foreground objects move fast and the proportion of foreground pixels is relatively large. In addition, four SD videos (with 6-8 Mbps) from TRECVID Surveillance Event Detection Task (TRECVID SED) and two HD sequences (with 30Mbps) from the Hisense traffic surveillance system are also used in our experiments.

As usual, the BD-PSNR and BD-Rate [42] are used to evaluate the encoding performance. For comparison, AVC high profile reference encoder, JM17.2 with IPPP and IBBP structures, is used as the basic anchor (denoted by AVC). Besides, three typical state-of-the-art methods are also utilized as the state-of-the-art anchors, including:

Fig. 10. Example frames of the surveillance sequences in our experiments.

TABLE IV
CONFIGURATIONS OF AVC HIGH PROFILE IN OUR EXPERIMENTS

Item	Value	Item	Value	Item	Value
Long-term	Enable	Ref. Num.	5	Profile/Level	High
Entropy Coding	CABAC	SAD Method	hadamard	QP Gap between I/P/B	1
8x8Trans.	Enable	Intra Period	0	Loop Filter	Enable
RDOQ	Enable	Modes	All	Search Range	64
RDO	Enable	ME	UMH	1/4-pel ME	Enable

- BPC: the method [34] that uses the reconstructed frame to train a background frame as the long-term reference;
- LKC: the JM-OPT anchor in [4] that uses key-frames as the long-term reference;
- BDC: our previous work [4] that encodes the 9-bit difference frames generated by subtracting BOF from input frames.

All the encoders are also implemented by extending JM17.2. As in [43], the JM17.2 is configured as low-delay High Profile (shown in Table IV) for surveillance video coding. Note that, for each encoder configuration, the intra period is set 0 (i.e., only the first frame is encoded as an intra-frame except the updated key-frame or background frame), so the coding performance is evaluated on P/B frames. All experiments are performed on Genuine Intel(R) CPU@2.66 GHZ and 8GB 667MHz DDR2 FB-DIMM memory.

$$M_{HB} = \begin{cases} \arg\min\limits_{k} \{J_k | k \in \Theta_{HB} \cup \Pi_{HB} \cup \Omega_{HB}\}, & if \frac{SAD_{R\times R}(\Delta C, \Delta Ref)}{SAD_{R\times R}(C, Ref)} - 1 < \gamma; \\ \arg\min\limits_{k} \{J_k | k \in \Theta_{HB} \cup \Pi_{HB}\}, & Otherwise. \end{cases} \quad (13)$$

TABLE V
THE PERCENTAGES OF GBS, FBS AND HBS

SD	Bank	Office	Overbridge	Crossroad	average
GB	87.41%	77.72%	85.32%	66.69%	**79.28%**
FB	2.52%	5.11%	1.84%	6.78%	**4.06%**
HB	10.07%	17.16%	12.84%	26.54%	**16.65%**
CIF.	**Snowroad**	**Snowgate.**	**Overbridge**	**Crossroad**	**average**
GB	81.26%	83.00%	72.15%	66.53%	**75.73%**
FB	0.95%	0.53%	2.54%	5.82%	**2.46%**
HB	17.80%	16.47%	25.31%	27.65%	**21.81%**

TABLE VI
THE OVERALL BD-RATE (%) AND BD-PSNR (dB) OF BMAP VS. BDC/BPC/LKC ON X86 PLATFORM

	SD	Bank		Crossroad		Office		Overbridge		average	
	Vs.	dB	%	dB	%	dB	%	dB	%	dB	%
	AVC	1.59	-63.16	1.73	-41.39	1.05	-39.02	2.73	-66.40	**1.78**	**-52.49**
	LKC	1.57	-58.84	1.12	-36.44	0.97	-35.72	1.73	-56.45	**1.35**	**-46.86**
I	BPC	1.69	-62.97	1.79	-40.77	1.11	-38.94	2.88	-65.49	**1.87**	**-52.04**
P	BDC	0.93	-36.20	0.66	-20.54	1.54	-49.66	0.72	-20.59	**0.96**	**-31.75**
P	**CIF**	**Crossroad**		**Overbridge**		**Snowroad**		**Snowgate**		**average**	
P	Vs.	dB	%	dB	%	dB	%	dB	%	dB	%
	AVC	1.82	-41.41	1.42	-37.92	3.29	-76.45	2.16	-62.73	**2.17**	**-54.63**
	LKC	1.20	-34.52	1.18	-37.71	1.85	-60.74	1.09	-43.08	**1.33**	**-44.01**
	BPC	1.84	-40.58	1.44	-37.27	3.43	-75.67	2.36	-63.24	**2.27**	**-54.19**
	BDC	0.98	-28.27	0.42	-15.03	0.80	-10.14	1.32	-20.36	**0.88**	**-18.45**
	SD	**Bank**		**Crossroad**		**Office**		**Overbridge**		**average**	
	Vs.	dB	%	dB	%	dB	%	dB	%	dB	%
	AVC	1.38	-60.85	1.31	-37.63	0.87	-37.80	2.43	-63.85	**1.50**	**-50.03**
	LKC	1.50	-59.20	0.96	-34.46	0.87	-36.48	1.62	-54.78	**1.24**	**-46.23**
I	BPC	1.42	-59.58	1.34	-36.81	0.91	-37.27	2.47	-61.94	**1.54**	**-48.90**
B	BDC	1.21	-57.11	0.77	-25.67	1.46	-50.46	1.09	-38.18	**1.13**	**-42.86**
B	**CIF**	**Crossroad**		**Overbridge**		**Snowroad**		**Snowgate**		**average**	
P	Vs.	dB	%	dB	%	dB	%	dB	%	dB	%
	AVC	1.40	-38.22	1.11	-36.28	2.74	-71.39	1.84	-55.72	**1.77**	**-50.40**
	LKC	1.04	-33.51	1.04	-37.87	1.71	-58.55	1.02	-39.10	**1.20**	**-42.26**
	BPC	1.41	-36.91	1.14	-35.47	2.83	-70.10	2.01	-55.93	**1.85**	**-49.61**
	BDC	1.11	-33.34	0.75	-28.06	1.36	-46.96	2.03	-46.87	**1.31**	**-38.81**

B. Experimental Result

Several experiments are designed in this study. Firstly, we analyze the distribution of FBs, GBs and HBs so as to demonstrate the necessity of BRP and BDP. Then, the second experiment is to compare the total coding performance between BMAP and the anchors. Thirdly, we will report the gain in foreground performance and the saving in background bit-rate brought by BDP. At last, the fourth experiment is conducted on the TRECVID SED videos and Hisense HD videos, where each video contains a large proportion of foreground pixels.

B.1 Block Classification Results

In this experiment, a statistical analysis is made on the distribution of FBs, GBs and HBs. The result for each sequence is shown in Table V, and the block category distributions in example frames of Crossroad (CIF) and Overbridge (CIF) have been illustrated in Fig. 8. We can observe from these results that, GBs take the largest part in surveillance videos. Naturally, BRP will contribute a lot to the prediction efficiency in BMAP. Meanwhile, HBs take a much larger proportion than FBs. Thus BDP, which is specially designed for HBs, also has an important effect on the prediction efficiency.

B.2 Total Bit-Rate Saving and PSNR Gain

Table VI lists the total encoding performance gains of BMAP compared with AVC, LKC and BDC on each sequence. At the same PSNR, BMAP averagely decreases 52.49%/54.63% (IPPP) and 50.03%/50.40% (IBBP) bit-rate on SD/CIF sequences over AVC, and 46.86%/44.01% (IPPP) and 46.23%/42.26% (IBBP) over LKC. These results also correspond to 1.78/2.17dB (IPPP) and 1.50/1.77dB (IBBP) PSNR gains over AVC, whereas 1.35/1.33dB (IPPP) and 1.24/1.20dB (IBBP) PSNR gains over LKC at the same bit-rate. In addition, compared with BPC, BMAP decreases averagely 52.04%/54.19% (IPPP) and 48.90%/49.61% (IBBP) bit-rate on SD/CIF sequences. Compared with BDC, BMAP can also averagely decrease 31.75%/18.45% (IPPP) and 42.86%/38.81% (IBBP) bit-rate on SD/CIF sequences. Fig. 11 illustrates several example rate-distortion curves in IPPP and IBBP structures.

We also observe that, the less proportion of HBs and GBs a sequence has, the less total bit-rate saving will be obtained by BMAP. For example, Crossroad (SD) has the least proportion 66.69% and least bit-rate saving 20.54% on IPPP using BMAP over BDC. This is because the performance gain of BMAP is mostly from BRP and BDP on HBs and GBs. Moreover, on different kinds of surveillance video sequences, BMAP exhibits slightly different advantages over the anchors. Compared with LKC, BMAP can obtain more bit-saving on sequences with large background regions, demonstrating that BRP is efficient when coding the exposed background regions using BOF. Compared with BDC, BMAP obtains more bit-rate decrease on sequences with large foreground regions. This indicates that BMAP can effectively avoid the foreground pollution problem in a large extent. While compared with BPC and AVC, BMAP achieves much larger gain in coding performance. The main reason is that the BRF in BPC is modeled from the reconstructed frames, while AVC is not specially designed for surveillance videos.

B.3 Encoding and Decoding Time

Table VII shows the comparison results of software encoding and decoding time between BMAP and the anchors. Because BMAP reduces the candidate partitions in the predictions, the total encoding time only slightly increases over the anchors (overall, less than 11%). As mentioned above, the increase is mainly due to the additional ME time for BDP. On average, there are 7.15%/6.25% (IPPP) and 5.28%/4.79% (IBBP) increase in the encoding time over AVC on SD/CIF sequences, whereas 10.82%/7.64% (IPPP) and 7.20%/6.35% (IBBP) over LKC. In contrast to LKC, BMAP has less increase in the encoding time over both BPC and BDC, because they all involve the background modeling in the encoding procedure. The results are 3.73%/4.90% (IPPP) and 3.44%/3.28% (IBBP) when compared with BPC; while compared with BDC, the results are 9.54%/6.22% (IPPP) and 4.94%/3.27% (IBBP). BPC requires more time than BDC because its background modeling is carried out on the reconstructed data for both encoding and decoding processes.

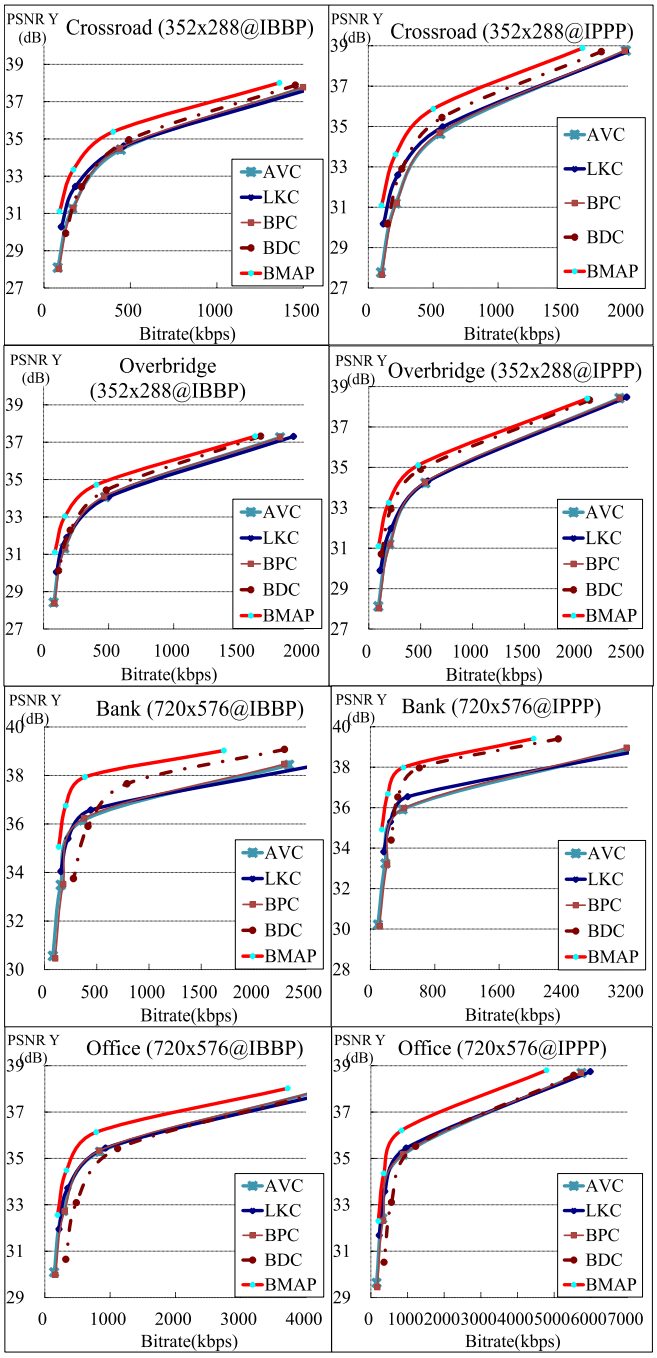

Fig. 11. The overall rate-distortion curves for four CIF and SD sequences coded in IPPP/IBBP.

On the other hand, since the BMAP decoder still only needs to decode each block once according to the decoded "mb_type," the decoding time also has a slight increase over the anchors (overall, less than 10%). The increase is mainly due to the calculation of ΔC, ΔR, Rec_f and background generation.

B.4 Foreground PSNR Gains and Background Bit-Saving

To validate that BMAP not only achieves the total performance gains, but also can obtain better foreground coding quality, we experimentally evaluate the foreground coding PSNR gains at the same bitrate. Let $Bits(A)$ denote bit number

TABLE VII
THE OVERALL ENCODING AND DECODING TIME INCREASE (ENI & DEI) OF BMAP VS. BDC/BPC/LKC ON X86 PLATFORM (%)

	SD	Bank		Crossroad		Office		Overbridge		average	
	Vs.	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI
IPPP	AVC	6.57	5.79	6.56	6.73	6.78	5.57	8.70	7.89	**7.15**	**6.50**
	LKC	9.45	8.10	9.13	9.13	10.84	8.54	13.88	12.03	**10.82**	**9.45**
	BPC	3.85	3.58	4.12	4.44	3.00	2.75	3.97	4.03	**3.73**	**3.70**
	BDC	8.32	6.62	7.44	7.23	10.79	8.49	11.63	9.01	**9.54**	**7.84**
	CIF	Crossroad		Overbridge		Snowroad		Snowgate		average	
	Vs.	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI
	AVC	5.27	5.79	10.20	10.26	4.64	4.80	4.90	4.82	**6.25**	**6.42**
	LKC	6.47	6.85	11.71	11.54	6.18	6.68	6.20	6.36	**7.64**	**7.86**
	BPC	4.10	4.74	8.73	9.01	3.14	2.98	3.63	3.33	**4.90**	**5.01**
	BDC	6.04	6.22	11.57	11.35	3.25	3.04	4.03	3.58	**6.22**	**6.05**
	SD	Bank		Crossroad		Office		Overbridge		average	
	Vs.	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI
	AVC	4.29	4.08	7.12	6.66	4.02	3.66	5.68	5.68	**5.28**	**5.02**
	LKC	5.75	5.65	10.56	9.64	4.78	4.53	7.72	8.00	**7.20**	**6.96**
IBBP	BPC	2.87	2.56	3.89	3.84	3.28	2.81	3.71	3.46	**3.44**	**3.17**
	BDC	3.57	3.07	8.45	6.93	3.43	2.91	4.29	3.87	**4.94**	**4.20**
	CIF	Crossroad		Overbridge		Snowroad		Snowgate		average	
	Vs.	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI	EnI	DeI
	AVC	3.52	3.59	9.02	7.70	3.46	3.61	3.15	3.41	**4.79**	**4.58**
	LKC	4.74	5.04	9.73	8.59	5.65	6.11	5.27	5.88	**6.35**	**6.40**
	BPC	2.34	2.19	8.31	6.82	1.35	1.24	1.12	1.05	**3.28**	**2.82**
	BDC	2.31	2.16	8.31	6.82	1.35	1.24	1.12	1.05	**3.27**	**2.81**

for encoding A, $Dis(A)$ denote the encoding distortion of A. Then the foreground coding gain can be evaluated by the foreground coding bit cost FC and the distortion FD for each the current frame I_i:

$$\begin{cases} FC = \sum\limits_{x,y} (Frate(I_i(x,y)), \\ FD = \sum\limits_{x,y} (FDis(I_i(x,y))), \end{cases} \tag{14}$$

$$\begin{cases} FDis(I_i(x,y)) \\ \quad = \begin{cases} 0, & S(I_i(x,y)) = GB. \\ SAD(I_i(x,y) - Rec(x,y)), & Otherwise. \end{cases} \\ Frate(I_i(x,y)) \\ \quad = \begin{cases} 0, & S(I_i(x,y)) = GB; \\ Bits(I_i(x,y)), & Otherwise. \end{cases} \end{cases} \tag{15}$$

In this equation, $Rec(x, y)$ is the final reconstructed result of the block $I_i(x, y)$ and $S(A)$ is the block category of A. With FC and FD as inputs, the foreground coding BD-PSNR [42] can be calculated. As for background coding, we often more care the bitrate saving at the same PSNR, thus the background BD-Rate can also be calculated in a similar way as foreground (namely, calculating the background coding cost and distortion).

Table VIII shows the foreground coding BD-PSNR and background coding BD-Rate of BMAP over BDC, BPC, LKC and AVC. Averagely, on SD/CIF sequences, BMAP achieves 1.31/0.89dB foreground coding PSNR gains and 63.12%/63.61% background bit-savings over BDC, 1.13/1.52dB and 28.39%/18.47% over BPC, 0.61/0.74dB and 66.06%/61.58% over LKC, 1.13/1.50dB and 63.68%/64.27%

TABLE VIII
THE FOREGROUND CODING BD-PSNR (F-BP, dB) AND BACKGROUND CODING BD-RATE (B-BR, %) OF BMAP VS. BDC/BPC/LKC

SD	Bank		Crossroad		Office		Overbridge		**average**	
Vs.	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR
AVC	1.32	-78.32	0.95	-45.11	0.53	-51.20	1.71	-80.07	**1.13**	**-63.68**
LKC	0.75	-78.66	0.53	-52.79	0.32	-54.52	0.84	-78.27	**0.61**	**-66.06**
BPC	1.29	-78.21	0.78	-43.64	2.27	-51.58	0.89	-79.06	**1.31**	**-63.12**
BDC	1.31	-30.22	0.95	-18.92	0.53	-41.64	1.72	-22.79	**1.13**	**-28.39**
CIF	Crossroad		Overbridge		Snowroad		Snowgate		**average**	
Vs.	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR	F-BP	B-BR
AVC	1.02	-56.28	0.84	-39.04	2.40	-90.01	1.75	-71.60	**1.50**	**-64.23**
LKC	0.60	-59.00	0.61	-47.05	1.06	-86.00	0.70	-54.28	**0.74**	**-61.58**
BPC	1.23	-54.90	0.67	-37.79	0.42	-89.26	1.25	-72.49	**0.89**	**-63.61**
BDC	1.02	-29.11	0.83	-5.60	2.42	-5.83	1.81	-13.32	**1.52**	**-13.47**

TABLE IX
THE TOTAL PERFORMANCE GAINS (BD-RATE) OF BMAP ON TRECVID AND HISENSE VIDEOS

	Vs.	MCTT E1101	MCTT E1102	MCTT E1103	MCTT E1105	**MCTTE Avg.**	Cross-road	Car-Road	**Hisense Avg.**
IPPP	AVC	-18.1%	-26.3%	-25.9%	-39.6%	**-27.5%**	-65.9%	-35.9%	**-50.9%**
	LKC	-17.2%	-22.4%	-25.5%	-30.5%	**-23.9%**	-50.0%	-25.0%	**-37.5%**
	BPC	-21.3%	-29.5%	-28.4%	-40.9%	**-30.0%**	-55.3%	-28.3%	**-41.8%**
	BDC	-35.8%	-24.0%	-47.5%	-17.5%	**-31.2%**	-45.8%	-17.5%	**-31.7%**
IBBP	AVC	-13.1%	-20.8%	-19.0%	-33.5%	**-21.6%**	-60.5%	-31.1%	**-45.8%**
	LKC	-15.6%	-21.3%	-22.8%	-27.6%	**-21.8%**	-43.3%	-26.2%	**-34.8%**
	BPC	-16.3%	-23.7%	-21.7%	-34.9%	**-24.1%**	-48.3%	-27.7%	**-38.0%**
	BDC	-32.6%	-24.3%	-43.8%	-14.8%	**-28.9%**	-41.4%	-16.2%	**-28.8%**

over AVC. We can observe that, although BDC saves more bits than LKC, BPC and AVC in the whole frame and background coding (i.e., the overall bit-rate decrease and background coding bit-saving of BMAP over BDC are the least), it produces a dramatic decrease of the foreground coding quality (i.e., the foreground coding gains of BMAP over BDC are nearly the largest). This is mainly due to the serious foreground pollution. On sequences with large foreground regions (e.g., Crossroad), the foreground coding loss of BDC, compared with BMAP, is much lower than the other sequences. Instead, BMAP can solve the foreground pollution problem to some extent while keeping the background coding capability of BDC. As a result, BMAP achieves remarkable foreground coding gain and background bit-saving.

B.5 Results on TRECVID SD and Hisense HD Videos

To evaluate the performance of BMAP on sequences with lots of foreground objects, the last experiment is conducted on the first 1920 frames of four long in-door surveillance sequences originally from TRECVID SED and two out-door Hisense HD videos. Note that TRECVID provides video sequences from five cameras, in which videos captured in Camera 4 are almost stationary videos without any moving foreground objects. So in this experiment, we use video sequences from the other four cameras, including MCTTE1101 (7.3Mbps from Camera 1), MCTTE1102 (7.3Mbps from Camera 2), MCTTE1103 (7.8Mbps from Camera 3) and MCTTE1105 (8.6Mbps from Camera 5). The Hisense HD videos are CrossRoad and CarRoad (30Mbps, 1600 $\times$ 1200).

Table IX shows the total performance gains (BD-Rate) of BMAP vs. BDC, BPC, LKC and AVC on the two kinds of sequences On average, BMAP achieves 27.5/21.6% (IPPP/IBBP) bitrate saving over AVC on the whole frames on the TRECVID videos, whereas 50.9/45.8% on the two Hisense videos. Over LKC/BPC/BDC, the results are 23.9/30.0/31.2% and 21.8/24.1/28.9% on the TRECVID videos; while 37.5/41.8/31.7% and 34.8/38.0/28.8% on the Hisense videos. We can see that, although a large number of irregular foreground objects exist in these sequences, BMAP can still outperform these anchors remarkably. This is mainly due to the high-efficient prediction algorithm BDP which is especially designed for HBs. Meanwhile, on the crowded in-door TRECVID videos, the performance improvement for BMAP LKC BPC and BDC over AVC is not as significant as that on other datasets On these crowded videos it is difficult for LKC to find a clean key-frame, and also difficult for BMAP, BDC and BPC to generate a clean background, whatever from the original input frames or from the low-quality reconstructed frames. It should be noted that, even in this case, BMAP still achieves remarkable bit-saving.

VI. CONCLUSION

This paper proposes a background-modeling based adaptive prediction (BMAP) method for surveillance video coding. The key idea of BMAP is to adaptively adopt the short-term reference prediction (SRP), the background reference prediction (BRP) and the background difference prediction (BDP) to encode the current data according to the block classification results. Extensive experiments on surveillance video sequences with different definitions and foreground distributions show that, the proposed BMAP can averagely save half of the total bit-rate and achieve a significant gain in foreground coding performance over AVC high profile. Moreover, it also outperforms several state-of-the-art methods remarkably.

Compared with model-based and object-oriented coding methods, BMAP improves the prediction efficiency without depending on pixel-level accurate background modeling and foreground segmentation. This makes it more practically applicable for standardization and a wide range of surveillance video systems. Moreover, the high-quality background frames that are encoded into the stream can well support the scene analysis, object detection and recognition. In this sense, BMAP can be treated as a useful attempt to integrate some recognition-friendly functionality into the video coding framework.

In the future work, we will further investigate on background modeling of videos under complex weather and illumination conditions, and then explore higher-efficient surveillance video coding methods that can utilize both the static background (e.g., the street light poles) and periodic background (e.g., the scene changes each day) to improve the coding performance.

APPENDIX: PROOF OF THEOREM 1

For each current block C and the buffered reference data *Ref*, after subtracting their background BC and BG (*BC is the corresponding data of C in BG*), the difference data of the current block, denoted by ΔC, and the difference data of the reference frame, denoted by ΔRef, can be generated by

$$\Delta C = C - BG, \quad \Delta Ref = Ref - BG. \tag{16}$$

In ΔRef, the process of searching ΔR for each ΔC can be expressed as

$$\Delta R = \arg\min_{\Delta W} \left\{ SAD(\Delta C - \Delta W) | \Delta W = W_{x,y} - BG_{x,y} \right\}, \tag{17}$$

where $BG_{x,y}$ is the block at position (x, y) in BG. From Eq. 17, we have

$$\begin{aligned} &SAD(\Delta C - \Delta R) \\ &\quad = \min \left\{ SAD\left((C - BC) - \left(W_{x,y} - BG_{x,y} \right) \right) \right\}. \end{aligned} \tag{18}$$

Then from Eq. 17 and Eq. 18, $SAD(\Delta C - \Delta R)$ can be rewritten as

$$\begin{aligned} &SAD(\Delta C - \Delta R) \\ &= \min \left\{ SAD \begin{pmatrix} ((F+B) - (BCF + BCB)) \\ -\left((WF_{x,y} + WB_{x,y}) - (BGF_{x,y} + BGB_{x,y}) \right) \end{pmatrix} \right\} \\ &= \min \left\{ SAD \begin{pmatrix} (F - WF_{x,y}) + (B - BCB) \\ -\left((BCF - BGF_{x,y}) + (WB_{x,y} - BGB_{x,y}) \right) \end{pmatrix} \right\}. \end{aligned} \tag{19}$$

Because

$$B \approx BCB \text{ and } WB_{x,y} \approx BGB_{x,y}, \tag{20}$$

we have

$$\begin{aligned} &SAD(\Delta C - \Delta R) \\ &\quad \approx \min \left\{ SAD\left((F - WF_{x,y}) - (BCF - BGF_{x,y}) \right) \right\}. \end{aligned} \tag{21}$$

Let (m, n) denote the best matched position of (x, y) when using $W_{x,y}$ to predict C, then we have $R = W_{\mathrm{m,n}}$ from Eq. 2. Then we further derive the following equation from Eq. 3.

$$\begin{aligned} &SAD(C - R) \\ &\quad = SAD(C - W_{m,n}) = SAD((F - WF_{m,n}) + (B - WB_{m,n})) \\ &\quad = SAD(F - WF_{m,n}) + SAD(B - WB_{m,n}). \end{aligned} \tag{22}$$

Instead, when using $\Delta W_{x,y}$ to predict ΔC, the (m, n) may not be the best matched position. Thus we have

$$\begin{aligned} SAD(\Delta C - \Delta R) &\le SAD(\Delta C - \Delta W_{m,n}) \\ &\approx SAD\left((F - WF_{m,n}) - (BCF - BGF_{m,n}) \right). \end{aligned} \tag{23}$$

Besides, since ΔR is a searched block that matches ΔC best, we have

$$\begin{aligned} &SAD(\Delta C - \Delta R) \\ &\quad \le SAD(\Delta C - 0) \approx SAD\left((F - BCF) + (B - BCB) \right) \\ &\quad = SAD(F - BCF) + SAD(B - BCB). \end{aligned} \tag{24}$$

By subtracting $SAD(\Delta C - \Delta R)$ from $SAD(C\text{-}R)$, we can obtain:

(1) If $F \approx WF_{m,n}$ and $SAD(BCF - BGF_{m,n}) < SAD(B - WB_{m,n})$, then we can get from Eq. 22 and Eq. 23

$$\begin{aligned} &SAD(\Delta C - \Delta R) - SAD(C - R) \\ &\quad \le SAD\left((F - WF_{m,n}) - (BCF - BGF_{m,n}) \right) \\ &\qquad - SAD(F - WF_{m,n}) - SAD(B - WB_{m,n}) \\ &\quad \approx SAD\left((BCF - BGF_{m,n}) \right) - SAD(B - WB_{m,n}) < 0. \end{aligned} \tag{25}$$

(2) If $B \approx WB_{\mathrm{m,n}}$ and $SAD(F - BCF) < SAD(F - WF_{m,n})$, then we can get also from Eq. 22 and Eq. 24

$$\begin{aligned} &SAD(\Delta C - \Delta R) - SAD(C - R) \\ &\quad \le SAD(F - BCF) - SAD(F - WF_{m,n}) \\ &\qquad - SAD(B - WB_{m,n}) \\ &\quad \approx SAD(F - BCF) - SAD(F - WF_{m,n}) < 0. \end{aligned} \tag{26}$$

(3) If $SAD((F - WF_{m,n}) - (BCF - BGF_{m,n})) < SAD(F - WF_{m,n})$, then from Eq. 23 and Eq. 24, we can get

$$\begin{aligned} &SAD(\Delta C - \Delta R) - SAD(C - R) \\ &\quad \le SAD\left((F - WF_{m,n}) - (BCF - BGF_{m,n}) \right) \\ &\qquad - SAD(F - WF_{m,n}) - SAD(B - WB_{m,n}) \\ &\quad \le SAD\left((F - WF_{m,n}) - (BCF - BGF_{m,n}) \right) \\ &\qquad - SAD(F - WF_{m,n}) < 0. \end{aligned} \tag{27}$$

(4) If there are no background pixels in $C (C \approx F \approx WF_{m,n})$, we have

$$\begin{aligned} &SAD(\Delta C - \Delta R) - SAD(C - R) \\ &\quad \ge SAD((BGF_{m,n} - BCF) + (F - WF_{m,n})) \\ &\qquad - SAD(F - WF_{m,n}) \\ &\quad \approx SAD((BGF_{m,n} - BCF) > 0. \end{aligned} \tag{28}$$

(5) If all the four conditions are not satisfied, we should practically calculate and compare the $SAD(C - R)$ and $SAD(\Delta C - \Delta R)$.

Because (4) and (5) are the cases that BDP may produce larger distortion, so we can get Eq. 6 by assembling (1)∼(3). ■

REFERENCES

[1] J. Gantz and D. Reinsel. (2012, Dec.). *The IDC Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East* [Online]. Available: http://www.emc.com/leadership/digital-universe/index.htm

[2] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. Chichester, U.K.: Wiley, 2003, pp. 136–138.

[3] K. Ugur, K. Andersson, A. Fuldseth, G. Bjontegaard, L. P. Endresen, J. Lainema, *et al.*, "High performance, low complexity video coding and the emerging HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 12, pp. 1688–1697, Dec. 2010.

[4] X. G. Zhang, L. H. Liang, Q. Huang, and W. Gao, "An efficient coding scheme for surveillance videos captured by stationary cameras," *Proc. SPIE Visual Commun. Image Process.*, vol. 7744, pp. 77442A-1–77442A-10, Jul. 2010.

[5] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 4. Oct. 2004, pp. 3099–3104.

[6] R. Forchheimer and O. Fahlander, "Low bit-rate coding through animation," in *Proc. Int. Picture Coding Symp.*, 1983, pp. 113–114.

[7] R. Forchheimer, O. Fahlander, and T. Kronander, "A semantic approach to the transmission of face images," in *Proc. Int. Picture Coding Symp.*, no. 10.5, Cesson-Sevigne, France, 1984.

[8] W. J. Welsh, "Model-based coding of moving images at very low bit rates," in *Proc. Int. Picture Coding Symp.*, paper 3.9, Stockholm, Sweden, 1987.

[9] K. Aizawa, "Model-based analysis-synthesis image coding system for very low-rate image transmission," in *Proc. Int. Picture Coding Symp.*, paper 4.3, Turin, Italy, 1988.

[10] K. Aizawa, H. Harashima, and T. Saito, "Model-based analysis synthesis image coding (MBASIC) system for a person's face," *Signal Process., Image Commun.*, vol. 1, no. 2, pp. 139–152, 1989.

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

[12] J. Y. Wang and E. H. Adelson, "Representing moving images with layers," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 625–638, Sep. 1994.

[13] D. Chai and K. Ngan, "Foreground/background video coding scheme," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 2. Hong Kong, Jun. 1997, pp. 1448–1451.

[14] I. Martins and L. Corte-Real, "A video coder using 3-D model based background for video surveillance applications," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 1998, pp. 919–923.

[15] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 255–261.

[16] I. Haritaoglu, D. Harwood, and L. Davis, "W^4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[17] A. Elgammal, "Efficient nonparametric kernel density estimation for real time computer vision," Ph.D. thesis, Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, 2002.

[18] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. 6th Eur. Conf. Comput. Vis.*, vol. 2. 2000, pp. 751–767.

[19] Y. Heikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.

[20] L. Cheng, M. Gong, D. Schuurmans, and T. Caelli, "Real-time discriminative background subtraction," *IEEE Trans. Image Process.*, vol. 20, no. 5, pp. 1401–1414, May 2011.

[21] J. K. Suhr, H. G. Jung, G. Li, and J. Kim, "Mixture of Gaussians-based background subtraction for bayer-pattern image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 3, pp. 365–370, Mar. 2011.

[22] J. Ding, M. Li, K. Huang, and T. Tan, "Modeling complex scenes for accurate moving objects segmentation," in *Proc. 10th Asian Conf. Comput. Vis.*, 2010, pp. 592–604.

[23] E. Francois, 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.

[24] 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.

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

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

[27] 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.

[28] 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.

[29] L. Liu, Z. Li, and E. J. Delp, "Efficient and low-complexity surveillance video compression using backward-channel aware Wyner-Ziv video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 453–465, Apr. 2009.

[30] 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.

[31] 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.

[32] M. Tiwari and P. C. Cosman, "Selection of long-term reference frames in dual-frame video coding using simulated annealing," *IEEE Signal Process. Lett.*, vol. 15, pp. 249–252, Feb. 2008.

[33] ITU, "Codecs for videoconferencing using primary digital group transmission," document ITU-T H.120, 1984.

[34] 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.

[35] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 7, pp. 1140–1154, Aug. 1987.

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

[37] D. Liu, D. B. Zhao, X. Y. Ji, and W. Gao, "Dual frame motion compensation with optimal long-term reference frame selection and bit allocation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 3, pp. 325–339, Mar. 2010.

[38] A. Leontaris and P. C. Cosman, "Compression efficiency and delay tradeoffs for hierarchical B-pictures and pulsed-quality frames," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1726–1740, Jul. 2007.

[39] 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 Recgnit.*, Dec. 2008, pp. 1–4.

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

[41] X. G. Zhang, T. J. Huang, Y. H. Tian, M. C. Geng, S. W. Ma, and W. Gao, "Fast and efficient transcoding based on low-complexity background modeling and adaptive block classification," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1769–1785, Dec. 2013.

[42] G. Bjontegaard, "Improvements of the BD-PSNR model," document ITU-T SC16/Q6, Doc.VCEG-AI11, 2008.

[43] T. K. Tan, G. Sullivan, and T. Wedi, "Recommended simulation common conditions for compression efficiency experiments," document ITU-T Q.6/SG16, Doc. VCEG-AA10, Nice, France, Oct. 2005.

[44] R. Ding, Q. Dai, W. Xu, D. Zhu, and H. Yin, "Background-frame based motion compensation for video compression," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2004, pp. 1487–1490.

[45] X. G. Zhang, Y. H. Tian, T. J. Huang, and W. Gao, "Low-complexity and high-efficiency background modeling for surveillance video coding," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process.*, Nov. 2012, pp. 1–6.

[46] X. G. Zhang, Y. H. Tian, L. H. Liang, T. J. Huang, and W. Gao, "Macro-block-level selective background difference coding for surveillance video," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2012, pp. 1067–1072.

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. Currently, he is a Researcher of video processing and compression techniques with MediaTek Inc., Beijing. His research interests include video coding, transcoding, and processing.

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

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

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991. He is a Professor of computer science with Peking University. Before joining Peking University, he was a Professor with the Harbin Institute of Technology from 1991 to 1995, and a Professor with the Institute of Computing Technology of Chinese Academy of Sciences 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. He served on the editorial board for 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 the IEEE ISCAS 2013, and served on the advisory and technical committees of numerous professional organizations. He is a member of the Chinese Academy of Engineering.