MACRO-BLOCK-LEVEL SELECTIVE BACKGROUND DIFFERENCE CODING FOR SURVEILLANCE VIDEO

Xianguo Zhang[1], Yonghong Tian[1], Luhong Liang[2], Tiejun Huang[1], Wen Gao[1]

[1]Institute of Digital Media, Peking University, Beijing, 100871, P. R. China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, P. R. China
{xgzhang, yhtian, lhliang, tjhuang, wgao}@jdl.ac.cn

***Abstract*— Utilizing the special properties to improve the surveillance video coding efficiency still has much room, although there have been three typical paradigms of methods: object-oriented, background-prediction-based and background-difference-based methods. However, due to the inaccurate foreground segmentation, the low-quality or unclear background frame, and the potential "foreground pollution" phenomenon, there is still much room for improvement. To address this problem, this paper proposes a macro-block-level selective background difference coding method (MSBDC). MSBDC selects the following two ways to encode each macro-block (MB): coding the original MB, and directly coding the difference data between the MB and its corresponding background. MSBDC also features at employs the classification of MBs to facilitate the selection, through which, prediction and motion compensation turns more accurate, both on foreground and background. Results show that, MSBDC significantly decreases the total bitrate and obtains a remarkable performance gain on foreground compared with several state-of-the-art methods.**

Keywords- surveillance video coding; background difference coding; background modeling; mode selection

I. INTRODUCTION

Video surveillance cameras are becoming ubiquitous for a wide range of applications in recent years. As networked high-definition cameras are widely adopted, one major challenge in building a video surveillance system is how to effectively reduce the bandwidth and storage costs. Therefore, it is desired to develop high-efficiency and low-complexity surveillance video encoders. Technologically, these methods should be able to utilize the special properties of surveillance video (e.g., the nearly invariant background in a short period) to boost the coding efficiency. Towards this end, periodically updated background modeling and prediction utilizing the generated background frame have become oft-used tools in surveillance video coding.

Basically, most of existing surveillance video coding methods follow two typical paradigms, namely, object-oriented video coding and background-prediction-based methods. Often, object-oriented methods employ background modeling and background subtraction techniques to separately compress the foreground and the background in surveillance video. Recently, by using some novel background modeling approaches (e.g., [1-3]) for foreground segmentation, several recent methods (e.g., [4-6]) based on MPEG-4 show the promising performance. However, it is widely recognized that video segmentation is a difficult problem in computer vision, especially on the surveillance videos with complex scenes (e.g., crowed streets). Therefore, instead of depending on accurate foreground segmentation, background-prediction-based methods [7-9] employ a long-term frame to predict the background regions under traditional hybrid coding framework (e.g. H.264/AVC). In [7-8], multiple quality-loss reconstructed frames are employed to model a background frame and use the background frame for the long-term reference, so the quality of the generated background frame cannot be guaranteed. Moreover, such a module should be embedded into the decoding process, leading to the inevitable increase of the decoding complexity. Therefore, Wiegand et al. [9] treat the high-quality encoded key frame as the long-term reference frame to avoid the problems in [7-8], and its efficiency has been proved by the anchor JM-OPT in [10]. However, without using background modeling, such a long-term frame is actually not a "clear" background frame, leading to inaccurate background prediction.

To avoid the inaccuracy foreground segmentation (as in [4-6]), the low-quality background frame (as in [7-8]), or the unclear background frame (as in [9]), Zhang et al. [10] introduced an efficient solution called background-difference-based coding (namely BDC here). BDC follows the traditional hybrid coding framework, but utilizes the original input frames to generate and encode the periodically updated background frame. After that, it calculates the difference frames by subtracting the reconstructed background frame from the input frames, and then codes these difference frames into the code stream.

As shown in Fig. 1, however, subtracting the macro-blocks (MBs) in the background frame from the MBs in the input frames might make the foreground lose their original texture. As a result, the dependency among the foreground MBs might be destroyed. In this paper, we call this phenomenon as "foreground pollution." We also conducted a simple experiment to evaluate the effect of the so-called foreground pollution. As shown in Table I, on the test sequences Bank, Office, and Crossroad in which there exist a large amount of foreground regions, the foreground coding performance of BDC is lower than the method in [10] which utilizes the key frame as long-term reference (namely KFLR). This suggests us that BDC may not apply to those

Dr. Yonghong Tian is the corresponding author.

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.136

foreground MBs. Thus, for each MB, a strategy should be introduced to selectively encode the original data or the corresponding data in the difference frame.

Towards this end, a macro-block-level selective background difference coding method (MSBDC) is proposed in this paper. MSBDC still follows the strategy of BDC in that original input frames are used to generate the background frame, but encodes the difference data in MB level rather than in frame level. That is, the basic coding unit for the difference data in MSBDC is the MB, instead of the overall difference frame in BDC. Moreover, MSBDC employs a threshold generation step in background modeling to divide MBs into three categories, which are used to select the available prediction modes. Most importantly, for each MB, instead of always coding the data in the difference frame (denoted by DM), MSBDC selectively encodes the original data or the DM according to its category and the rate-distortion performance.

To realize the selection, MSBDC adds a special group of prediction modes to encode DM, called background difference coding modes (BDCMs). BDCMs adopt the similar prediction methods employed in the traditional inter- and intra-prediction modes (namely traditional modes) in hybrid coding framework, but with DM as its input. In this case, a DM cannot be predicted by the original reference data used in the traditional modes. Instead, BDCMs use the difference data between the original reference data and their corresponding background data as the reference data. For simplicity, these reference data for DM are referred as DR.

Usually, foreground and background pixels in surveillance video have different motion characteristics. Thus for those MBs that contain background pixels, after removing the background pixels, it is more accurate to use the DR to predict the left foreground in DM than traditional prediction. As a result, less residual will be generated in MSBDC in these MBs, and the performance gain will be obtained both in the foreground and the overall frame.

Experimental results show that, the proposed method achieves an average PSNR gain of 0.56dB and 1.30dB on the foreground coding performance for CIF (352×288) and SD (720×576) sequences over BDC, and 0.96dB and 0.18dB over KFLR for CIF/SD squences. On the average, it obtains an overall bitrate decrease of 7.9%/19.1% compared with BDC, and 42.8%/36.4% compared with KFLR. Moreover, the complexity increases slightly.

The rest paper is organized as follows. The proposed MSBDC is described in Sec. 2. Then Sec. 3 presents the experimental results. We conclude this paper and discuss the future work in Sec. 4.

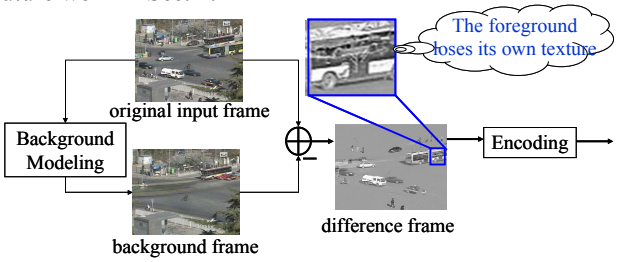


Figure 1. The foreground pollution problem in BDC [10].

TABLE I. BDC VS. KFLR ON FOREGROUND CODING PEFORMANCE

SD	***Crossroad***	*Overbridge*	***Bank***	***Office***	***average***
PSNR gain	**-0.15 dB**	1.10 dB	**-0.98 dB**	**-1.98 dB**	**-0.50 dB**
CIF	***Crossroad***	*Overbridge*	*Snowroad*	*Snowway*	*average*
PSNR gain	**-0.30 dB**	0.30 dB	0.66 dB	1.05 dB	0.43 dB

II. THE PRPOSED METHOD

A. The Framework

The overall framework of the proposed MSBDC is shown in Fig. 2. The "traditional framework" in the figure represents the traditional hybrid coding framework such as H.264/AVC, and it consists modules of the Reconstructed Frame Buffer, Encoding with Traditional Modes and Reconstructing with Traditional Modes. Besides, MSBDC also contains the following modules: (1) The Background Modeling module is used to generate the background frame using the original input frames; (2) the MB Classification module divides the input MBs into three categories; (3) the Mode Calculation module is used to estimate the available prediction modes for each MB; (4) the Background Encoding and Reconstructing modules are used to encode and reconstruct the background frame; (5) the BDCMs-based Coding and Reconstructing modules are employed for encoding and reconstructing DM; (6) the Mode Selection module selects the best prediction mode. In addition, some computing and selection operators are used to calculate DM, DR and the final reconstructed MBs.

As shown in Fig. 2, the encoding process of MSBDC can be described as follows:

1. The original input frames are utilized by the Background Modeling module to generate the background frame, which is then encoded by Background Encoding.

2. The Background Reconstructing module reconstructs the background frame. Then, the DM and DR are generated by subtracting MBs in this reconstructed background frame.

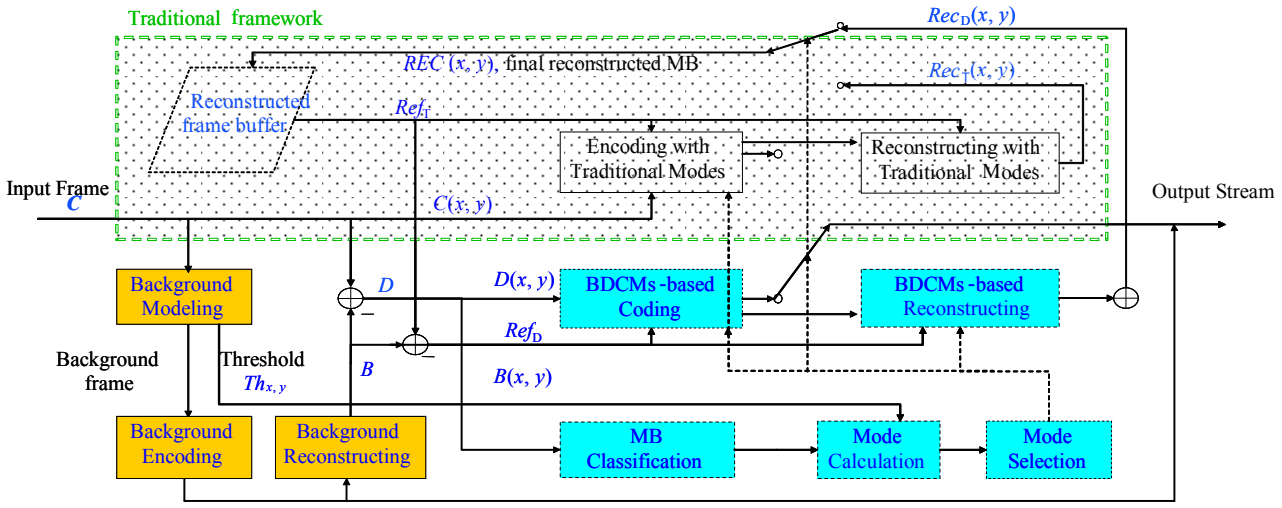
3. Each MB is then classified by the MB Classification module as Foreground, Background or Foreground Border MB, according to its corresponding DM. To this end, a threshold is generated from the Background Modeling.

4. For each MB, the Mode Calculation module estimates which traditional modes should be available and which modes should be reused in BDCMs.

5. Two groups of encoding modes are carried out for each MB simultaneously:

a) Traditional Modes: The original data of this MB are encoded with the inter- and intra-prediction modes in the traditional hybrid coding framework, and the data from the Reconstructed Frame Buffer are used as reference.

b) BDCMs: The DM corresponding to this MB is encoded through BDCMs with DR as the reference data. In the intra prediction of BDCMs, the DR are the difference data between the reconstructed neighboring MBs of this MB and their corresponding reconstructed background MBs; whereas in the inter prediction, the DR are generated by subtracting the reconstructed background frame from the original reference frames of this MB.



BDCMs : background difference coding modes;

Figure 2. The proposed framework

6. The better result between the above two groups of prediction modes, for each MB, is selected as the final coding result by the Mode Selection Module, according to the available prediction modes from the Mode Calculation module and the rate-distortion performance of these modes.

7. If one traditional mode is selected as the best mode for an MB, the reconstructed data of the MB from the Reconstructing with Traditional Modes module can be directly written into the Reconstructed Frame Buffer. Otherwise, if one BDCM is selected, we should firstly compensate the directly reconstructed DM with its corresponding reconstructed background MB. After that, the compensation result is treated as the final reconstructed MB and written into the Reconstructed Frame Buffer.

In the following, we will firstly present the formulation of MSBDC in Sec. B, and then describe two algorithms in Sec. C respectively for calculating the available prediction modes for each MB and selecting the best mode. Since the calculation of available prediction modes for each MB needs one threshold to classify the MB, Sec. D further presents how to calculate such a threshold..

B. The Formulation of MSBDC

To begin with, we first define several notations as follows: C denotes the input frame, D denotes the difference frame, B is the recently reconstructed background frame, $R(x, y)$ is the encoding result of the MB at position (x, y), $R_T(x, y)$ is the MB coding result under the traditional modes, $R_D(x, y)$ denotes that under BDCMs, Ref_T denotes the original reference data used in traditional modes, and Ref_D denotes DR in BDCM. Suppose $\Phi(A, B)$ represents the following coding procedure: employ intra and inter prediction on matrix A with matrix B as reference, and use entropy coding on the transforming and quantifying result of the prediction residuals. Among them, $R(x, y)$ is calculated by

$$R(x,y)=\begin{cases} R_T(x,y)=\Phi(C(x,y),Ref_T), & J_T(x,y)\le J_D(x,y) \\ R_D(x,y)=\Phi(D(x,y),Ref_D), & J_T(x,y)>J_D(x,y) \end{cases}, \quad (1)$$

where

$$D(x, y) = C(x, y) - B(x, y) + 256, \quad (2)$$

$$Ref_D = Ref_T - B + 256, \quad (3)$$

and $J_T(x, y)$ is the minimum rate-distortion cost (*RDCost*) calculated from encoding the original MB at position (x, y) using the available traditional modes, and $J_D(x, y)$ is that of encoding the DM using the BDCMs. Eq. 1 shows that for each MB, MSBDC will make a selection between the following two processes through RDCost comparison: coding the residual data generated by predicting the original MB from the original reference data, and coding the residual data generated by predicting the DM using the DR. Note that in Eq. 2 and 3, a constant of 256 is added to ensure the subtraction result always positive. How to calculate $J_T(x, y)$ and $J_D(x, y)$ remains to be discussed in Sec. C.

To guarantee the match between video coding and decoding, Ref_T should be read from the Reconstructed Frame Buffer. This buffer is used to store the reconstructed result of each MB. Suppose that, $Rec(x, y)$ is the final reconstructed result of each MB, $Rec_T(x, y)$ is the directly reconstructing result of $R_T(x, y)$, and $Rec_D(x, y)$ is that of $R_D(x, y)$. Then $Rec(x, y)$ for each *MB* is calculated by

$$Rec(x,y)=\begin{cases} Rec_T(x,y), & J_T(x,y)\le J_D(x,y) \\ Clip(Rec_D(x,y) - 256 + B(x,y)), & J_T(x,y)>J_D(x,y) \end{cases}, \quad (4)$$

where the *Clip* function with any 16×16 Matrix I as input is

$$Clip(I_{i,j})=\begin{cases} I_{i,j}, & 0\le I_{i,j}\le 255 \\ 0, & I_{i,j}<0 \\ 255, & I_{i,j}>255 \end{cases}, \quad (5)$$

where $I_{i,j}$ denotes the element at position (i, j) in I, $0\le i, j\le 15$..

C. Selection between Traditional Modes and BDCMs

1) *Calculating the available prediction modes*

Because the foreground and background in surveillance video have different motion characteristics, different prediction modes should be employed in their coding process. Thus to remove computational redundancy, the

prediction modes for each MB should be pre-determined if it can be classified as foreground MBs (*FM*s), foreground border MBs (*FBM*s) and background MBs (*BM*s). According to the pixel values in the DM at position (x, y), the threshold $Th(x, y)$ of this MB, and the value $D_i(x, y)$ of the i-th pixel in $D(x, y)$, the category of the DM at position (x, y), $S(x, y)$, is calculated by

$$S(x,y)=\begin{cases} FM, & \left|\{i \mid D_i(x,y) < Th_{x,y}\}\right| < 20 \\ FBM, & 20 \le \left|\{i \mid D_i(x,y) < Th_{x,y}\}\right| < 200 \\ BM, & \left|\{i \mid D_i(x,y) < Th_{x,y}\}\right| \ge 200 \end{cases}, \quad (6)$$

where the $|\bullet|$ denotes the size of a set. This equation means that each *BM* has more than 200 background pixels, while the number of background pixels in each *FM* is less than 20; Otherwise, it is an *FBM*.

Generally speaking, the foreground pollution often happens on FMs rather than BMs. Therefore, BMs should be encoded only by BDCMs while FMs are encoded by the traditional modes. For FBMs, after removing the background pixels, the predictions in BDCMs may produce less or more residual than traditional modes. Thus both the traditional modes and BDCMs should be used in FBMs. As a result, the total bitrate may decrease through the selection.

When BDCMs are used in an MB, the prediction modes in BDCMs are calculated as follows: If the MB is classified as an BM, the high complexity prediction modes, e.g. I4×4, P4×4, P4×8, P8×4 in H.264/AVC, should not be contained in BDCMs because there are still little residual after removing the background. Otherwise, if the MB is classified as an FBM, the inter prediction in BDCMs for the remaining foreground pixels on foreground border region may produce less residual after removing the background pixels. For the blocks in the smaller inter prediction modes, e.g. P4×4, P4×8, P8×4 in H.264/AVC, there is a low probability to contain these foreground borders. Thus to remove the computational redundancy, only the larger inter prediction modes like P8×8, P16×16, P16×8 and P8×16 are used for BDCMs in the FBMs. Besides, the best traditional mode used in coding the original FBM is also included in the BDCMs to exclude accidental conditions (e.g., one foreground border is contained in a 4×8 block).

Compared with the traditional hybrid coding framework, there are less modes used in BMs, the same amount of modes used in FMs, and a few more modes used in FBMs. Therefore, there is only a slightly increase of complexity in the whole encoding process.

2) *Mode selection*

Given the available prediction modes, the remaining problem is how to select the best mode. As stated in Sec. A, the minimum RDCosts, $J_T(x, y)$ and $J_D(x, y)$ are used for the selection between the traditional modes and *BDCM*s. Let Θ and Ω denote the set of the traditional modes and the BDCMs respectively, D_k/R_k be the rate/distortion under the mode k in the traditional modes, and D_j'/R_j' be those under the mode j in the BDCMs. Then the minimum RDCost J_T and its corresponding mode M_T are calculated by

$$J_T = \min\{ J_k \mid k \in \Theta \}, \ where\ J_k = D_k + \lambda R_k, \quad (7)$$

$$M_T = \arg\min_k \{ J_k \mid k \in \Theta \}, \quad (8)$$

where J_k is the RDCost under the traditional mode k, and λ is the *Lagrangian* multiplier. Similarly, the minimum RDCost J_D and its corresponding prediction mode M_D are calculated from each RDCost J_j' of the mode j in BDCMs by

$$J_D = \min\{ J_j' \mid j \in \Omega \}, \ where\ J_j' = D_j' + \lambda R_j', \quad (9)$$

$$M_D = \arg\min_j \{ J'_j \mid j \in \Omega \}. \quad (10)$$

Then, the best mode $M^*(x, y)$ with the minimum *RDCost* among the traditional modes and *BDCM*s is calculated by

$$M^*(x,y) = \begin{cases} M_T & J_T \le J_D \ and\ S(x,y) \ne BM \\ M_D & J_T > J_D \ and\ S(x,y) \ne FM \end{cases} \quad (11)$$

From this equation, we can easily derive the following selection strategies: For BMs, the prediction mode with the minimum RDCost in the BDCMs is chosen; For FMs, the mode with the minimum RDCost in traditional modes is used; For FBMs, the mode with the minimum RDCost among traditional modes and BDCMs is selected.:

D. The Threshold Generation

As seen from Eq. 6, a threshold $Th_{x,y}$ is used to classify each MB at position (x, y). Such a threshold for each MB is calculated by the algorithm shown in Fig. 3. This algorithm can be divided into two steps: (1) the threshold that is generated for the MB at the same position in the last frame is used to determine and identify the potential background pixels in the current MB, and (2) the number of the potential background pixels is counted to calculate their root-mean-square deviation value, which is used as the new threshold.

Input:

$I(m, n)$: the pixel value at position (m, n) of the selected MB in the current frame.

$Bg(m, n)$: the background pixel corresponding to the $I(m, n)$,

Init:

$Th_{x,y}$ is initialized to the corresponding threshold in the previous frame, or is initialized to 14 for the first frame

Caluculating:

For each position (m, n)

1. Calculate the difference between $Bg(m, n)$ and $I(m, n)$

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

2. Mark $Cmp(m, n)$ to 1 for the potential background pixel.

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

3. Count the potential background pixel number

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

4. Calculate the root-mean-square deviation of the potential background pixels as T_i

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

where Round(A) denotes to round the value A.

Output: $Th_{x,y}$

Figure 3. The threshold calculation algorithm.

III. EXPERIMENTS

In experiments, H.264/AVC baseline encoder using key frames as long-term reference (KFLR) and the background difference based encoder (BDC) are used as the anchors for evaluating the performance of the MSBDC encoder.

For fair comparison, a sequence structure dividing input frames into super group of pictures (*S-GOP*) is used for the three encoders (as shown in Fig.4). That is, the first frame is treated as the background frame of the initial group of training frames ($TrainSet_0$), and the background generated by $TrainSet_0$ is updated as the background frame for $S\text{-}GOP_1$, … In this way, each S-GOP can utilize the corresponding background frame to encode its frames. In our experiments, the number of training frames is set to 120 and the size of an *S-GOP* is set to 480. As in [10], a mean-shift background modeling method is used in BDC and MSBDC. Besides, to simplify the bit-allocation of the background frame or the key frame, the quantization parameter is equal to that of I frames minus 6, and only the intra predictions are utilized.

As stated in [11], all the encoders are implemented on the H.264/AVC baseline profile of the software JM17.2 configured, which is shown in Table II. For the data set, the first 1080 frames of SD surveillance sequences of *Crossroad*, *Overbridge*, *Office*, *Bank* and CIF sequences of *Crossroad*, *Overbridge*, *Snowroad*, *Snowway* [12] are used to evaluate the three encoders. Example frames of these sequences are shown in Fig. 5. We can see that among them, *Crossroad*, *Overbridge* and *Office* have relatively large foreground regions. The experimental results are shown from Table III to Table VI.

TABLE II. MAIN TEST CONDITION OF USED JM17.2 IN EXPERIMENTS

Item	Descr.	Item	Descr.	Item	Descr.
QP	22,27,32,37	B frames	Disable	Profile/Level	Baseline
Entropy	UVLC	SearchRange	32	Long-term	Enable
8x8Transform	0	RDO	Used	RDO Quant.	Used
RDO Quant	1	Ref Number	4	ME	UMH
SAD Method	hadamard	IntraPeriod	30	1/4-pel ME	Enable

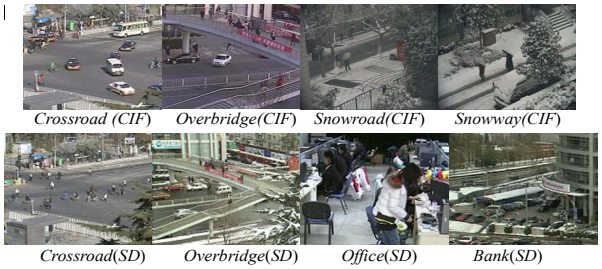


Figure 4. Example frames of tested surveillance sequences.

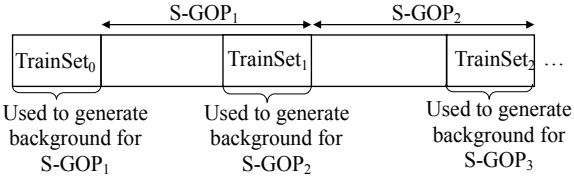


Figure 5. Sequence structure for background modeling.

TABLE III. MSBDC VS. BDC AND KFLR ON OVERALL BITRATE AND ENCODING TIME ON X86 PLATFORM (%)

SD	Crossroad		Overbridge		Bank		Office		**average**	
vs(%)	bitrate	time	bitrate	time	bitrate	time	bitrate	time	**bitrat**	**time**
KFLR	-33.8	8.4	-56.0	9.9	-50.0	4.4	-31.5	1.6	**-42.8**	**6.1**
BDC	-7.5	6.2	-3.5	7.7	-27.1	2.4	-38.5	0.6	**-19.1**	**4.2**
CIF	Crossroad		Overbridge		Snowroad		Snowway		**average**	
%	bitrate	time	bitrate	time	bitrate	time	bitrate	time	**bitrat**	**time**
KFLR	-22.0	10.2	-28.9	5.5	-45.5	5.7	-49.2	13.2	**-36.4**	**7.3**
BDC	-16.4	7.0	-7.2	2.5	-3.5	2.6	-4.4	10.2	**-7.9**	**4.2**

TABLE IV. MSBDC VS. BDC AND KFLR ON FOREGROUND PSNR GAIN

SD	Crossroad	Overbridge	Bank	Office	**average**
KFLR	0.37 dB	1.57 dB	0.95 dB	0.35 dB	**0.81 dB**
BDC	0.51 dB	0.47 dB	1.91 dB	2.31 dB	**1.30 dB**
CIF	Crossroad	Overbridge	Snowroad	Snowway	**average**
KFLR	0.64 dB	0.87 dB	1.06 dB	1.25 dB	**0.96 dB**
BDC	0.92 dB	0.58 dB	0.42 dB	0.31 dB	**0.56 dB**

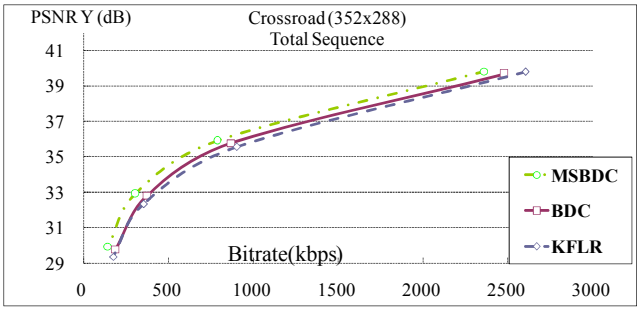


Figure 6. RD Curves for the Total Sequence of Crossroad(352x288)

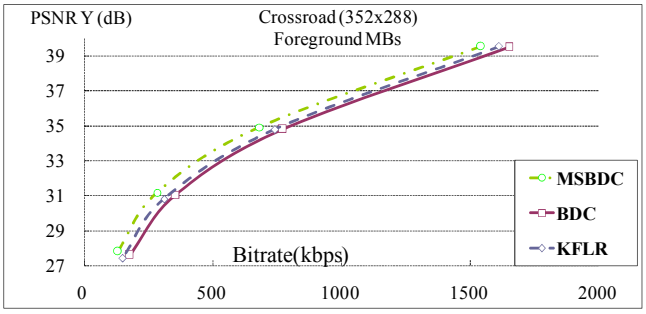


Figure 7. RD Curves for the Foreground MBs of Crossroad(352x288)

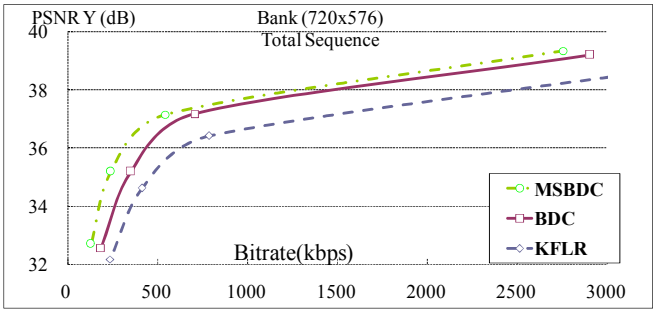


Figure 8. RD Curves for the Total Sequence of Bank(720x576)

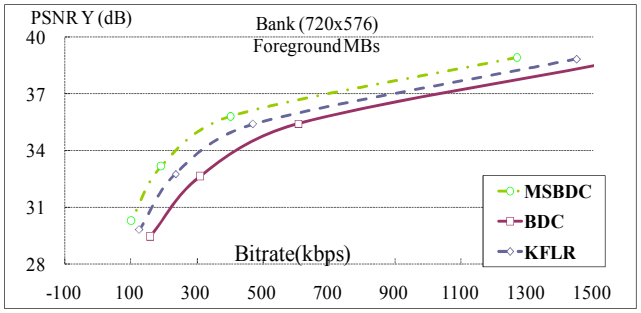


Figure 9. RD Curves for the Foreground MBs of Bank(720x576)

TABLE V. THE MSBDC USING THE LONG-TERM REFERENCE FRAME VS. BDC AND KFLR ON OVERALL BITRATE (%)

SD	Crossroad	Overbridge	Bank	Office	**average**
KFLR	-53.5	-36.3	-57.5	-32.7	**-45.0**
BDC	-11.0	-7.0	-32.2	-39.5	**-22.4**
CIF.	Crossroad	Overbridge	Snowroad	Snowway.	**average**
KFLR	-23.12	-29.7	-49.63	-50.39	**-38.23**
BDC	-17.63	-8.30	-10.77	-6.94	**-10.9**

TABLE VI. THE MSBDC USING LONG-TERM REFERENCE FRAME VS. BDC AND KFLR ON FOREGROUND CODING PERFORMANCE

SD	Crossroad	Overbridge	Bank	Office	**average**
KFLR	1.02 dB	0.47 dB	1.64 dB	0.36 dB	**0.87 dB**
BDC	1.99 dB	0.61 dB	0.54 dB	2.32 dB	**1.36 dB**
CIF.	Crossroad	Overbridge	Snowroad	Snowway.	**average**
KFLR	0.68 dB	0.86 dB	1.39 dB	1.69 dB	**1.16 dB**
BDC	0.96 dB	0.56 dB	0.74 dB	0.68 dB	**0.73 dB**

From Table III, we can see the total bitrate decrease in *MSBDC* compared with BDC and KFLR. On SD and CIF sequences, *MSBDC* achieves an average bitrate decrease of 19.1% and 7.9% compared with BDC, but 42.8% and 36.4% compared with KFLR, while encoding time increases only 6.1%/4.2% over BDC and 7.3%/6.1% over KFLR.

As shown in Table IV, on SD and CIF sequences, an average PSNR gain of 1.08dB and 0.56dB is achieved on foreground MBs compared with BDC, and 0.81dB and 0.96dB over KFLR. Rate-Distortion curves of Bank (SD) Crossroad (CIF) in the overall frame and the foreground regions are shown in Fig. 6~9. From these results, we can safely conclude that MSBDC can solve the foreground pollution problem to some extent. As for the decoding time, the decoder always needs to decode only once for each MB according to the decoded "mb_type." Therefore, the complexity increases slightly due to the calculation of DR.

Moreover, we can also utilize the reconstructed background frame as the long-term reference frame in MSBDC. In this context, we also conduct an experiment to evaluate the performance of MSBDC. As shown in Table V and VI, MSBDC further achieves an average bitrate decrease of 45.0%/38.23% over KFLR on SD/CIF sequences and 22.4%/10.9% over BDC. For the foreground performance gain, the result is 0.87dB/1.16dB over KFLR on SD/CIF sequences and 1.36dB/0.73dB over BDC.

IV. CONCLUSION

In this paper, we proposed a macro-block-level selective background difference coding method (MSBDC). The main contribution of this method is to selectively encode the original MB or the DM for each MB, making motion estimation in hybrid coding more accurate and producing less residual. Through this selection mechanism, moreover, this method avoids the "foreground pollution" existed in the BDC [10] to some extent. As a result, a total performance gain is achieved on both the foreground and the overall frame, with slight increase of the encoding and decoding complexity. In the future, we will engage to develop an adaptive quantization-rate-distortion rate allocation for encoding the background frame.

ACKNOWLEDGMENT

This work is partially supported by grants from National Basic Research Program of China under contract No. 2009CB320906, the Chinese National Natural Science Foundation under contract No. 61035001 and No. 61072095, and Fok Ying Dong Education Foundation under contract No. 122008.

REFERENCES

[1] A. Elgammal, "Efficient nonparametric kernel density estimation for real time computer vision," Ph.D. Thesis, Rutgers, The State University of New Jersey, 2002.

[2] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," IEEE Trans. Image Process., vol. 13, no. 11, 2004.

[3] L. Cheng, M. Gong, D. Schuurmans, T.Caelli, "Real-Time Discriminative Background Subtraction," IEEE Trans. Image Process., vol. 13, no. 11, 2009.

[4] R.V. Babu and A. Makur, "Object-based surveillance video compression using foreground motion compensation," ICARCV, 2006.

[5] A. Hakeem, K. Shafique and M. Shah, "An object-based video coding framework for video sequences obtained from static cameras," Proc. ACM MM, pp.608-617, 2005.

[6] D. Venkatraman and A. Makur, "A compressive sensing approach to object-based surveillance video coding," Proc. IEEE ICASSP, 2009

[7] ITU-T Recommendation H.120, Codec for videoconferencing using primary digital group transmission, 1984;

[8] M. Paul, W. Lin, C. T. Lau, , et al., "Video coding using the most common frame in scene," IEEE ICASSP, 2010.

[9] T. Wiegand, X. Zhang, B. Girod, "Long-term memory motion-compensated prediction" Circuits and Systems for Video Technology, IEEE Transactions on, 1999

[10] X. Zhang, L. Liang, Q. Huang, et al., "An Efficient Coding Scheme for Surveillance Videos Captured by Stationary Cameras," in Proc. Visual Commun. Image Process., 2010.

[11] TK Tan, G. Sullivan, T. Wedi, "Recommended Simulation Common Conditions for Coding Efficiency Experiments", ITU-T Q.6/SG16, doc. VCEG-AA10, October 2005

[12] ftp://124.207.250.92/public/seqs/video/ (accessed by AVS member).