Fast and Efficient Transcoding Based on Low-Complexity Background Modeling and Adaptive Block Classification

Xianguo Zhang, *Student Member, IEEE*, Tiejun Huang, *Senior Member, IEEE*, Yonghong Tian, *Senior Member, IEEE*, Mingchao Geng, Siwei Ma, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*—It is in urgent need to develop fast and efficient transcoding methods so as to remarkably save the storage of surveillance videos and synchronously transmit conference videos over different bandwidths. Towards this end, the special characteristics of these videos, e.g., the relatively static background, should be utilized for transcoding. Therefore, we propose a fast and efficient transcoding method (FET) based on background modeling and block classification in this paper. To improve the transcoding efficiency, FET adds the background picture, which is modeled from the originally decoded frames in low complexity, into stream in the form of an intra-coded G-picture. And then, FET utilizes the reconstructed G-picture as the long-term reference frame to transcode the following frames. This is mainly because our theoretical analyses show that G-picture can significantly improve the transcoding performance. To reduce the complexity, FET utilizes an adaptive threshold updating model for block classification and then adopts different transcoding strategies for different categories. This is due to the following statistics: after dividing blocks into categories of foreground, background and hybrid ones, different block categories have different distributions of prediction modes, motion vectors and reference frames. Extensive experiments on transcoding high-bit-rate H.264/AVC streams to low-bit-rate ones are carried out to evaluate our FET. Over the traditional full-decoding-and-full-encoding methods, FET can save more than 35% of the transcoding bit-rate with a speed-up ratio of larger than 10 on the surveillance videos. On the conference videos which should be transcoded more timely, FET achieves more than 20 times speed-up ratio with 0.2 dB gain.**

***Index Terms*—Background modeling, classification, surveillance and conference videos, transcoding.**

I. Introduction

VIDEO surveillance and video teleconferencing systems are more and more widely used for safety and communication applications. These systems usually adopt common video codecs such as H.264/AVC with general settings to compress captured videos for weeks or months. Compared with general videos, surveillance and conference videos always own much lower coding bit-rate at the same quality. As a result, the deployed common video codecs always compress them at much higher bit-rates. For video surveillance applications, the high-bit-rate video streams greatly enlarge the video storage and retrieval cost. According to the statistics, if the more than 5 million cameras in UK are all High-Definition (HD) ones with general H.264/AVC video codecs, at least 8,000,000 Terabytes data will be produced in one-month storage time. Thereby for surveillance videos, high-efficiency and low-complexity bit-rate scaling techniques are in urgent need to transcode high bit-rate videos to low-bit-rate ones. Moreover, there should be remarkable bit saving compared with the traditional full-decoding-and-full-encoding (FDFE) transcoder. As for video teleconferencing applications, with the exponentially increasing usage of different teleconferencing clients (e.g., mobile devices), the high-bit-rate streams are required to be real-timely and simultaneously transcoded into multiple quality-maintained low-bit-rate conference videos for the various bandwidths of client devices. Thereby it is increasingly becoming an important issue to develop faster-than-real-time transcoders to broadcast multiple quality-maintained conference videos to various devices. Summarized from the requirements of surveillance and conference video transcoding, it is important to develop specially-designed methods to transcode surveillance videos with large bit saving and low complexity and transcode conference videos to quality-maintained streams as fast as possible.

One intuitive transcoder satisfying the basic requirement is to directly connect common video decoder and encoder. However, it is not very practical due to the transcoding complexity and sometime large degree of transcoding quality loss. To decrease the complexity, many motion estimation (ME) and mode decision (MD) simplification methods (e.g., [1]–[7]) have been proposed. But seldom were proposed specially for surveillance and conference videos. Whereas to improve the transcoding quality or efficiency, three methods can be utilized: the object-oriented transcoding methods based on object segmentation [8], [9], the region based methods which employed more bits on regions of interest [10], and the block-based background-prediction optimization methods [11]–[13]. Although some problems still exist in these methods, they enlightened us to utilize better and low-complexity modeled static background to optimize surveillance video transcoding.

In this paper, to analyze what kind of background can mostly improve the transcoding efficiency, a theoretical comparison using some conclusions in [14] is firstly carried out among three

Manuscript received December 14, 2012; revised March 12, 2013; accepted May 06, 2013. Date of publication August 29, 2013; date of current version November 13, 2013. This work was supported in part by grants from the National Basic Research Program of China under contract No. 2009CB320906, the Chinese National Natural Science Foundation under contract No. 61035001, 61121002 and 61176139. This paper is an extended version of the original paper which appeared in the Proceedings of IEEE ICME 2012 Conference and was among the top-rated 4% of ICME'12 submissions. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Sen-Ching Cheung.

The authors are with the Institute of Digital Media, National Engineering Laboratory for Video Technology, 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/TMM.2013.2280117

typical background frames, including the key frame in [11], background modeled from reconstructed frames in [12], [13] and the proposed G-picture modeled from the originally decoded frames. The comparison result shows G-picture is the optimal long-term reference frame. Furthermore, a theoretical analysis for how to quantize G-picture in transcoding shows that, G-picture can most significantly improve the rate-distortion (RD) performance when intra-encoded with the minimum quantization parameter (QP) of the input stream. To evaluate the complexity reduction, another experimental analysis is carried out to utilize G-picture to figure out blocks' motion characteristics. After blocks are divided into foreground units, background units and hybrid units by calculating their difference with the static background, the statistics based on such block classification show: different reference frame candidates should be used for different transcoding units; motion search range should be calculated differently from the difference between the decoded and predicted motion vectors; different set of prediction modes should be used for different categories.

Based on these analysis results, we propose a fast and efficient transcoding (FET) method based on low-complexity background modeling and adaptive block classification. In order to improve the transcoding efficiency, FET utilizes the G-picture, which is online trained by a low-complexity and high-quality background modeling using the originally decoded frames as input, as the long-term reference frames to transcode each decoded frame. Because the G-picture is very clean, encoded only using intra prediction and quantized with a smaller QP, such a better modeled and encoded G-picture will provide better long-term reference for the following frames. Even for conference videos, in which the background is usually covered by tightly-moved foreground in large areas, the unclean modeled G-picture can also enlarge the transcoding efficiency in some degree. Meantime, to reduce the complexity, FET employs G-picture to realize an adaptive threshold-updating model to achieve adaptive block classification and adopt different transcoding strategies for different block categories. These strategies are in forms of removing redundant prediction modes, simplifying motion estimation and reducing reference frames. Such adaptive block classification reduces the complexity dramatically by employing different ME&MD strategies on different block categories. In summary, as an extension of our work in [15], [16], besides the background model based high-efficiency transcoding in [15] and the block-classification based speed-up strategies in [16], this paper makes improvements on: the theoretical proof for the efficiency of transcoding with properly-quantized G-picture as reference, the low-complexity and high-efficiency background modeling algorithm to generate G-picture, the adaptive-threshold updating based block classification and extensive experiments for both surveillance and conference videos.

To assess the significant bit saving of our FET on surveillance videos and the remarkable complexity saving for both conference videos and surveillance videos, extensive experiments are conducted on eight surveillance videos from AVS (Audio and Video coding Standard) workgroup and eight conference videos from JCT-VC (Joint Collaborative Team on Video Coding). These experiments include the background modeling efficiency, block classification result and the final results for transcoding efficiency improvement and complexity reduction. These results are calculated during transcoding high-bit-rate H.264/AVC streams to low-bit-rate ones. To demonstrate the efficiency, the traditional FDFE method directly using H.264/AVC for re-encoding is chosen as the basic anchor.

The experimental results show that, for surveillance/conference videos, FET averagely saves more than 35%/5% bit saving, equivalent to more than 1.1/0.2 dB PSNR (peak signal-to-noise ratio) gains. Meanwhile, larger than 10/20 and 2/3 times speed-ups are obtained using full search ME and fast ME methods respectively. While compared with the more efficient transcoding with long-term key frame as background reference, the result is also very significant. Moreover, the block-classification based fast method in FET averagely achieves 0.5 times speed-up than the method not relying on block-classification, with the similar transcoding quality. To practically test our method, two real-time transcoding systems based on FET are designed to respectively transcode HD surveillance videos to much lower bit-rates and HD conference videos to different bit-rates for different bandwidths. In this way, FET is practically proved very efficient.

The rest of this paper is organized as follows. The related works for surveillance and conference video transcoding are discussed in Section II, and the theoretical analysis for the efficiency improvement with G-picture is presented in Section III. Section IV presents the framework and the methods, where the analyses for each block category's distributions of prediction modes, motion vectors and reference frames are included in the sub-sections to derive the speed-up methods. Experimental setup is given in Section V, and the extensive experimental results are shown in Sections VI and VII concludes this paper.

II. Related Works

Generally, FDFE is the simplest video transcoding approach without any change on the encoding process of decoded videos. However, due to the complexity and efficiency, FDFE is not applicable in practical transcoding systems. For complexity, several fast transcoding methods using motion vector refinement were proposed by [1]–[4] to decrease ME complexity, with comparable performance to FDFE. Meanwhile, methods for saving MD complexity [5]–[7] were also widely investigated in the past years. For example, a zero-block decision based scheme was introduced by Wu *et al.* [5], where the zero-block decision scheme was used to skip impossible inter and intra prediction modes, consequently leading to 93% saving of MD time, on average. Nevertheless, seldom methods specially focused on complexity reduction of surveillance and conference videos. In these videos, blocks with different proportion of foreground pixels have different motion characteristics, so simplifying the MD and ME processes for relative static regions will intuitively save the time cost with little quality loss.

For efficiency, because most of surveillance cameras are mounted to a fixed scene for a long-time and each frame can be subjectively divided into foreground and background objects, some pioneer works started to employ the static background to improve the efficiency. Intuitively, a reasonable solution following such idea is to transcode foreground objects and background separately. We denote it as object-oriented transcoding throughout this paper. Object-oriented methods were firstly

proposed in [8] and [9] to divide an input frame into foreground and background regions, and then transcode background with low bit-rate. However, object-oriented methods usually focused on subjectively measured "foreground objects." For surveillance and conference video transcoding, the subjective measurement is a debatable problem, especially considering various security requirements. Besides, the accurate automatic foreground segmentation is still an open problem, and it is also a great challenge to use few bits to compress the object representation information and the foreground prediction residual.

To avoid the challenging object segmentation and improve the transcoding efficiency, some efficient block-based coding methods can also be applied to the encoding procedure in transcoding. The methods include the region-based, long-term key frame and background prediction based coding. Among them, the region-based coding [10] mainly focused on achieving high compression efficiency and better subjective quality of foreground regions with low encoding complexity, but the total bit-rate was not decreased very much. The long-term key frame based coding utilized the high-quality encoded key frame as long-term [11] for follow-up frames, but there were still some so-called "exposed background" regions that appeared in the current frame and disappeared in the recent reference frames and the key frame. An example for the distributions of the exposed background can be seen in Fig. 1. As seen, there are some circled regions which can find better reference in the G-picture (although the background in conference video is usually not very clean). As a result, the transcoding efficiency for these regions could not be improved by using key frame and recently decoded frame as reference. To address this problem, background prediction based coding methods were proposed in [12] and [13]. Both H.120 in [12] and M. Paul *et al.* [13] featured at exploiting the reconstructed frames to model the background and employed the background as an additional reference for coding the following frames. However, quality of the generated background could not be guaranteed because significant quantization loss existed in the utilized reconstructed frames. Moreover, high-complexity background generation would be embedded in decoder to guarantee decoding match. Although there were some problems in the optimized methods above, they still enlightened us to improve the efficiency of the "exposed background regions" with better and low-complexity background frame as reference.

Following the above ideas for complexity and efficiency, it is very practical to improve transcoding efficiency using the better modeled G-picture and decrease the complexity according to the motion characteristics of the input blocks. Therefore, we propose to employ the long-term G-picture to facilitate more efficient background prediction and utilize block-classification based speed-up strategies for three categories of blocks.

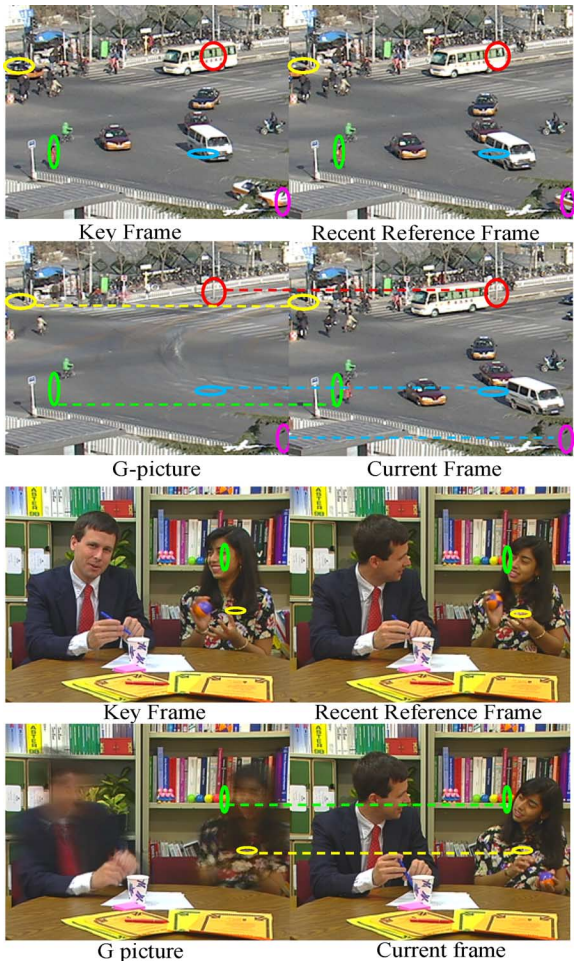


Fig. 1. The examples of the "exposed background regions" are shown in the current frame for surveillance and conference videos respectively. The circled regions in the "current frame" can only find good reference in the "G-picture" rather than the "key frames" and the "recent reference frames." Usually, there are more such regions in surveillance videos because objects in surveillance videos move more intensely.

TABLE I
SYMBOL DEFINITION

Symbol	Meaning
σ_Δ	Displacement error variance with one hypothesis, which usually denotes a reference frame in transcoding.
σ_n	Residual noise component that cannot be predicted by motion compensation under a hypothesis.
$\Phi(\Lambda)$	PSD (power spectral density) of the input 2-D and spatial-frequency signal $\Lambda=(\omega_x, \omega_y)$.
$\Phi_X(\Lambda)$	$\Phi_X(\Lambda)$ denotes the $\Phi(\Lambda)$ with X as long-term reference
$P(\Lambda)$	PDF (probability density function) of σ_Δ. $P(\Lambda) = e^{-2\pi\sigma_\Delta^2(\omega_x^2+\omega_y^2)}$ reflects the motion compensation accuracy.
$\Phi_{nn}(\Lambda)$	The power spectrum of residual noise component σ_n
$\Phi_{ss}(\Lambda)$	The non-negative signal power spectrum of the input video signal
OB	Sample matrix of the reconstructed result of the proposed G-picture
RB	Sample matrix of the background trained from the reconstructed result of transcoding each frame.
KB	Sample matrix of the high-quality transcoded key frame

III. EFFICIENCY ANALYSIS

To begin with our analysis and discussion, the symbols used in this paper are defined in Table I.

As discussed, the key to improve transcoding performance is to explore high-quality background data from decoded frames. Following this, the efficiency of exposed background regions will be improved with the help of better long-term background. Although the idea is very straightforward, there is no theoretical analysis so far on what is the optimal background. In this section, we firstly theoretically prove that the G-picture represented by *OB* is the optimal long-term reference frame for efficiently transcoding decoded frames. Secondly, we analyze that encoding G-picture into stream with the minimum decoded QP can guarantee the optimal RD performance for transcoding.

A. Why G-Picture is Optimal for Transcoding the Background

As stated, *RB* is trained from the reconstructed results of the decoded frames, *KB* is one of the originally decoded frames and *OB* is the background trained from the originally decoded frames. Therefore, *OB* could combine the advantages of *RB* and *KB* and it is probably a better long-term reference frame. In terms of prediction distortion, using *OB* as the long-term reference frame will achieve less distortion than that using *RB* or *KB* for any long surveillance or conference video. More formally, this conjecture can be expressed as Lemma 1.

Lemma 1: Let $D(OB,\Lambda)/D(RB,\Lambda)/D(KB,\Lambda)$ respectively denote the prediction distortion between a decoded long surveillance or conference sequence Λ and the long-term OB/RB/KB. Using the same motion search method, the following equation is satisfied in transcoding:

$$D(OB,\Lambda) \le \min\{D(KB,\Lambda), D(RB,\Lambda)\}. \tag{1}$$

Proof: For any transcoding unit at the position of (x,y) in an exposed background region (if any) of an input frame $I_i \in \Lambda$, we can find a matched block in *OB* but there is no such block in *KB*. *RB* may contain similar block but its quality is probably poorer than *OB* due to the quantization loss of the frames used to reconstruct it. Thus on the probability, we have (2), shown at the bottom of the page, where P denotes the set of exposed background regions and $|X|$ is the size of the set X. For surveillance and conference videos, as lots of such regions exist in each decoded frame I_i, we can get

$$\begin{aligned} D(I_i,OB) &= \sum_{x,y} D(I_i(x,y),OB) \\ &\le \sum_{x,y} \min\left(D(I_i(x,y),KB), D(I_i(x,y),RB)\right). \end{aligned} \tag{3}$$

Because

$$\begin{aligned} &\min\{D(I_i,RB), D(I_i,KB)\} \\ &\quad \ge \sum_{x,y} \min\left(D(I_i(x,y),KB), D(I_i(x,y),RB)\right), \end{aligned} \tag{4}$$

we can get

$$\begin{aligned} D(I_i,OB) &= \sum_{x,y} D(I_i(x,y),OB) \\ &\le \min\{D(I_i,RB), D(I_i,KB)\}. \end{aligned} \tag{5}$$

Thus for the decoded long sequence, we can obtain

$$\begin{aligned} D(\Lambda,OB) &= \sum_{i=1} D(I_i,OB) \\ &\le \sum_{i=1} \min\left(D(I_i,RB), D(I_i,KB)\right) \\ &\le \min\{D(\Lambda,RB), D(\Lambda,KB)\}. \end{aligned} \tag{6}$$

As stated in [14], for any two reference frames, the one providing smaller prediction distortion and smaller $\Phi(\Lambda)$ will lead to a better rate-distortion performance. The distortion relationship has been discussed in lemma 1. To regard the $\Phi(\Lambda)$s of *OB*, *RB* and *KB*, we have Lemma 2.

Lemma 2: Let $\Phi_{OB}(\Lambda)$, $\Phi_{RB}(\Lambda)$ and $\Phi_{KB}(\Lambda)$ denote the PSDs of a decoded surveillance or conference sequence with OB, RB and KB as the long-term reference frames respectively. With the same ME efforts, the following equation is satisfied in surveillance and conference video transcoding :

$$\Phi_{OB}(\Lambda) < \min\left\{\Phi_{RB}(\Lambda), \Phi_{KB}(\Lambda)\right\}. \tag{7}$$

Proof of Lemma 2 is given in the Appendix. By combining Lemma 1 and 2, we can get Theorem 1.

Theorem 1: Let $RD(OB,\Lambda)/RD(RB,\Lambda)/RD(KB,\Lambda)$ denote the rate-distortion performance between a decoded surveillance or conference sequence Λ and OB/RB/KB. Using the same motion search on the long-term reference frames OB, KB and RB, the following equation is satisfied in transcoding:

$$RD(OB,\Lambda) < \min\{RD(KB,\Lambda), RD(RB,\Lambda). \tag{8}$$

Proof: Again as stated in [14], between any two prediction reference frames, if using one can obtain smaller prediction distortion and smaller $\Phi(\Lambda)$, the reference frame can help to achieve a better RD result. Because $D(OB,\Lambda)$ is proved in Lemma 1 to be the minimum among $D(X,\Lambda)$s and $\Phi_{OB}(\Lambda)$ is also derived in Lemma 2 to be the minimum among $\Phi_{\mathrm{X}}(\Lambda)$s, $RD(OB,\Lambda)$ is also minimum.

In summary, by utilizing *OB* as the long-term reference frame, the transcoding efficiency of the decoded frames in surveillance and conference videos will be significantly improved. As stated in [14], the less prediction error variance (PEV) leads to less $\Phi_{nn\_l}(\Lambda)$, so using the long-term *OB* with less $\Phi_{nn\_l}(\Lambda)$ might produce less PEV. To validate this, we experimentally calculate the average PEV between each input frame and the long-term *OB*/*KB*/*RB*. Fig. 2 shows the results for two sequences, crossroad (352×288) and overbridge (352×288). We can see that, after several initial frames, PEV for *OB* becomes less than that of *KB* and *RB*. Moreover, the gap between *OB* and *KB*/*RB* becomes larger and larger as frame number increases. This is because *OB* contains more higher-quality background pixels and less noise or foreground pixels.

B. How to Quantize G-Picture for the Least RD Cost

For the decoding match of using G-picture as the long-term reference frame in transcoding, we should encode G-picture into stream. Thus another problem is how large should be the QP for quantizing it? As stated in [23], the Lagrange RDO theory calculates the Lagrange cost J for each sequence from the Lagrange cost of each frame by

$$J = \sum_{j=1}^{n} \sum_{k} RD(Q, I_{j,k}, P_{j,k}, V_{j,k}), \tag{9}$$

$$\frac{\left|\{(x,y)|\left(D(I_i(x,y),OB) \le \min\{D(I_i(x,y),DB), D(I_i(x,y),KB)\}\right)\}\right|}{|\{(x,y)|I_i(x,y)\in P\}|} \approx 1 \tag{2}$$

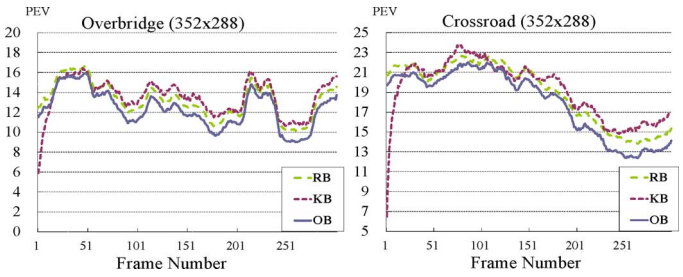


Fig. 2. Each frame's PEV curves for long-term *RB*, *KB* and *OB*.

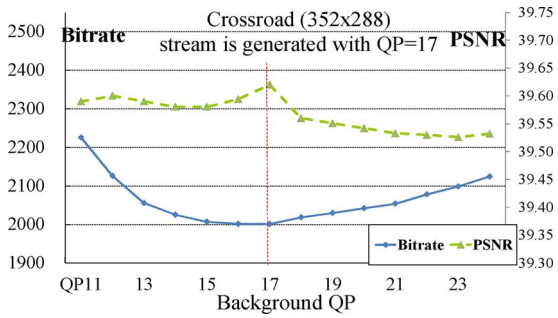


Fig. 3. The transcoding PSNR and bit-rates with different QPs for G-picture, where the minimum decoded QP $=$ 17 helps achieve the best bit-rate and PSNR.

where n is the total number of frames in the sequence, $I_{j,k}$ is the k-th transcoding unit of the j-th frame I_j, $V_{j,k}$ is the prediction motion vector, $P_{j,k}$ denotes the predicted data of the current transcoding unit and $RD(Q, I_{j,k}, P_{j,k}, V_{j,k})$ is the Lagrange cost for coding $I_{j,k}$ with Q as the QP. Particularly, while coding with G-picture as the long-term reference frame, $P_{j,k}$ can be described by

$$P_{j,k} = \Omega(I_{j,k}, OB, Q_B, R_{j,k}), \tag{10}$$

where Q_B is the QP for coding *OB* and $R_{j,k}$ is $I_{j,k}$'s set of reference frames excluding the long-term reference frame. Because G-picture is not an original input frame, (9) turns to be

$$\begin{aligned} J(Q_B) &= \lambda_B(Q_B) \times R_B(Q_B, OB) \\ &+ \sum_{j=1,k}^{n} RD(Q, I_{j,k}, \Omega(I_{j,k}, OB, Q_B, R_{j,k}), V_{j,k}), \end{aligned} \tag{11}$$

where function R_B calculates the bit cost for coding the *OB* with QP equal to Q_B, and λ_B is the Lagrange multiplier of R_B. As how *Theorem* 1 is derived, any prediction reference X' which provides less sufficient reference (i.e., larger prediction distortion) than X will lead to

$$RD(Q, I_{j,k}, X', V_{j,k}) \geq RD(Q, I_{j,k}, X, V_{j,k}). \tag{12}$$

Because the larger QP produces larger distortion, for any positive integer q, we further have

$$\begin{aligned} &RD(Q, I_{j,k}, \Omega(I_{j,k}, (Q_B + q), R_{j,k}), V_{j,k}) \\ &\quad \geq RD(Q, I_{j,k}, \Omega(I_{j,k}, Q_B, R_{j,k}), V_{j,k}), \\ &\text{where } RD(Q_B) - RD(Q_B + q) \\ &\quad = A(Q_B, q) - \sum_{j=1}^{n} B_j(Q_B, q), \\ &A(Q_B, q) = (\lambda_B(Q_B) \times R_B(Q_B, OB) \\ &\quad - \lambda_B(Q_B + q) \times R_B(Q_B + q, OB)), \\ &B_j(Q_B, q) = RD(Q+q, I_{j,k}, \Omega(I_{j,k}, OB, Q_B+q, R_{j,k}), V_{j,k}) \\ &\quad - RD(Q, I_{j,k}, \Omega(I_{j,k}, OB, Q_B, R_{j,k}), V_{j,k}). \end{aligned} \tag{13, 14}$$

Note that, the $A(Q_B, q)$ in this equation is not less than zero because bit cost the intra-coded *OB* will turn smaller with a larger QP, and $B(Q_B, q)$ is also larger than zero because of (13). Moreover, supposing Q_D is the minimum QP of the decoded QPs from the input stream, we can derive

$$\begin{aligned} B_j(Q_B, q) = &RD(Q, I_{j,k}, \Omega(I_{j,k}, (Q_B + q), R_{j,k}), V_{j,k}) \\ &- RD(Q, I_{j,k}, \Omega(I_{j,k}, Q_B, R_{j,k}), V_{j,k}) \approx 0. \end{aligned} \tag{15}$$

This is because *OB* is trained from the original decoded frames which already had the QP_D-level quality loss, and quantizing *OB* could not make the quality loss less than QP_D-level. Therefore, we can get

$$\begin{aligned} &J(Q_B) - J(Q_B + q) \\ &= \begin{cases} A(Q_B, q), & Q_B + q \leq Q_D \\ A(Q_B, q) - \sum_{j=1}^{n} B_j(Q_B, q), & \text{otherwise.} \end{cases} \end{aligned} \tag{16}$$

In (16), because surveillance and conference videos always capture the same scene for long-time, one G-picture can long-termly predict large number of following frames. That means, n is very large and

$$\begin{cases} J(Q_B) - J(Q_B + q) = A(Q_B, q) > 0, & Q_B + q \leq Q_D \\ J(Q_B) - J(Q_B + q) = A(Q_B, q) - \sum_{j=1}^{n} B_j(Q_B, q) \leq 0, & \\ \qquad \text{otherwise.} \end{cases} \tag{17}$$

This means, Q_D is the best QP to quantize *OB* and achieve the minimum total rate-distortion cost.

$$\begin{aligned} J(Q_D) \leq \min\{&J(Q_B), Q_B \; ranges \; from \; the \; smallest \; QP \\ &to \; the \; largest\}. \end{aligned} \tag{18}$$

Thus in our FET, G-picture should be quantized with the minimum decoded QP. To verify the theory, we have employed different QPs for G-picture to obtain the total bits and the transcoding PSNR for an input stream. With an input stream of crossroad (CIF, 352 × 288) encoded it with QP $=$ 17, the coding bit-rate and PSNR curves utilizing the long-term *OB* quantized with QP $= 11 \sim 24$ can be seen from Fig. 3. As is seen, using QP $=$ 17 to quantize *OB* leads to the least bit cost and best PSNR.

IV. The Proposed Method

Besides the necessary of utilizing the long-term and properly-quantized G-picture improve the transcoding efficiency,

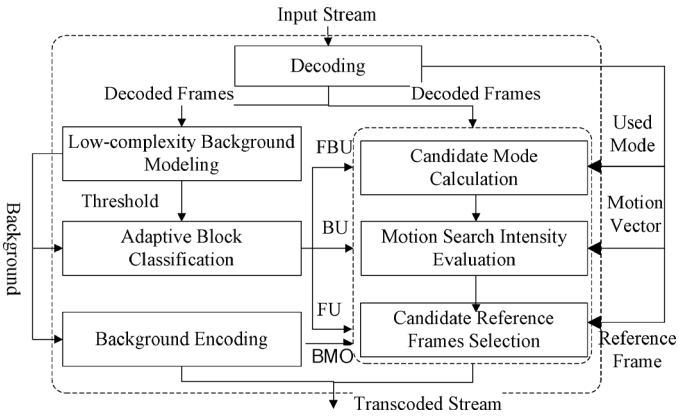


Fig. 4. Framework of the Proposed FET.

a low-complexity and high-efficiency background modeling algorithm should be embedded into FET to generate a high-quality G-picture. As for reducing the transcoding complexity, because the classified transcoding units, including background unit (BU), foreground unit (FU) and hybrid unit (FBU), always have different motion characteristics, FET should employ different MD and ME strategies for each category. Therefore, the generalized framework of the proposed FET is constructed as shown in Fig. 4. It works as follows: firstly, a background frame is generated from the originally decoded frames by *Low-complexity Background Modeling*, and then this background should be encoded into stream by *Background Encoding*. The reconstructed result of *Background Encoding* is used as a selective long-term reference frame for following decoded frames. After that, the *Adaptive Block Classification* utilizes an adaptively updated and automatically learning threshold to divide the blocks in current frame into FUs, FBUs and BUs. Thirdly, with the help of decoded data (i.e., reference frames, motion vectors and prediction modes), different ME and MD strategies (i.e., *Reference Frame Selection*, *Candidate Modes Calculation* and *Motion Estimation Intensity Evaluation*) will be respectively used for the three block categories.

In the following parts of this section, Section IV-A introduces the low-complexity and high-efficiency background modeling algorithm used in the *Low-complexity Background Modeling*; Section IV-B presents the algorithm of block classification based on threshold updating for the *Adaptive Block Classification*; The complexity analyses and summarized methods respectively for *Reference Frame Selection, Candidate Mode Calculation* and *Motion Estimation Intensity Evaluation* are introduced in Sections IV-C, IV-D and IV-E. In these complexity analyses, an H.264/AVC-transcoding is used to derive the distributions of the optimal reference frames, best prediction modes and motion search ranges for BUs, FUs and FBUs, and speed-up strategies are respectively summarized for them. The experiments are conducted on four representative surveillance and conference videos (surveillance ones of crossroad/overbridge and conference ones of mthr_dotr/paris, all of which can be seen from Section V), whose input H.264/AVC stream for transcoding is at about 1000 kbps. These videos are more representative because they contain different characteristics of bright/dark scenes, large/small moving objects and fast/slow motions.

TABLE II
MEMORY COST FOR EACH PIXEL (BYTE)

ITEM	RA	GMM-1	GMM-5	MS
Buffered frames	1× (char)	1× (char)	1×(char)	M× (char)
Mean values	1× (float)	2× (double)	2× (double)	1
Weight	0	1× (double)	1× (double)	0
Threshold/variance	0	1× (double)	1× (double)	0
Match points number	0	1× (char)	1× (char)	0
SUM of MEMORY	**5**	**34**	**34×5 = 170**	**M=120**

TABLE III
THE PSNR GAIN (dB) FROM BACKGROUND MODELING AND THE MODELING TIME (SECOND)

T-BP-x	crossroad		overbridge		snowgate		snowroad		average	
	Gain	Time	Gain	Time	Gain	Time	Gain	Time	Gain	Time
GMM-1	0.79	5.9	0.50	5.9	1.13	5.8	0.91	5.8	0.84	5.9
RA	0.93	1.6	0.80	1.7	1.75	1.7	1.34	1.7	1.21	1.7
MS	1.01	11.5	0.89	11.3	1.62	10.4	1.23	10.3	1.19	10.8
GMM-5	1.22	61.8	0.95	61.2	1.76	57.0	1.51	56.4	1.36	59.1

A. Background Modeling

Recently, background modeling has been utilized for efficient surveillance video coding and transcoding. In this section, we will firstly analyze and compare among existing oft-used background modeling methods, and then a background modeling method with low memory cost and computational complexity is proposed to generate G-picture for video transcoding. To evaluate the efficiency of different common modeling methods for video transcoding, four typical background modeling methods are implemented and embedded into the FDFE method with H.264/AVC baseline profile. We transcode the input H.264/AVC streams at 1000 kbps for four sequences (crossroad, overbridge, snowroad and snowgate) to the output streams at bit-rates of 64, 128, 256 and 512 kbps. The four methods are the Gaussian Mixed Models [17] using 1 or 5 models for each pixel (GMM-1 or GMM-5), the Mean-Shift (namely MS) proposed in [18], and the popularly used Gaussian running average (RA). For background modeling in surveillance and conference video transcoding, as is referred in Piccardi's [19], performance, memory cost and running time are the same important factors. The calculations for their memory cost in each pixel position are listed as follows. (1) RA: one current pixel with type of char and one float-precision mean value for each pixel should be buffered. (2) GMM-X: besides the buffered input pixel, a GMM model is required to be buffered. The model is composed of double-precision mean value, variance and weight. Moreover, an 8-bit value should be stored to count the number of matched points for each GMM model. (3) MS: Mean-shift based algorithms usually buffer all the training frames and very few additional temporal variables are used for the clustering and sorting operations.

Supposing the number of training frames is $M = 120$, the memory cost for each algorithm derived from the above analysis is listed in Table II. Then we implement transcoding methods respectively utilizing RA, MS, GMM-1 and GMM-5 to train G-picture as long-term reference on H.264/AVC baseline profile. The methods are correspondingly named T-BP-MS, T-BP-RA, T-BP-GMM-1 and T-BP-GMM-5, and the transcoding time and efficiency on different CIF sequences can be seen from Table III. In a brief summary, GMM-5 contributes largest to video coding performance gain but spares a

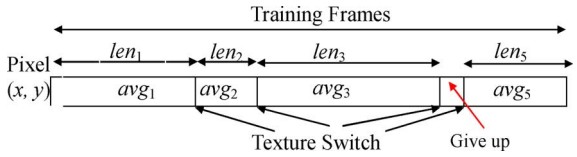


Fig. 5. Calculate the mean value and weight for each segment.

relative large memory and time cost. In practical system, especially in parallelism or hardware environment, such GMM-5 cannot meet the requirement for fast modeling and low memory cost. This inspires us to propose a method which can achieve higher performance with less memory and time cost.

To maintain or improve background quality, an ideal solution for background modeling is to calculate the mean value of all the pure background pixels in the training frames. However, it is very difficult in recent years to exactly justify which pixels belong to the background. Physically, "background" equals to the most frequently-appearing content. This inspires FET to utilize a novel segment-and-weight based running average (SWRA) to approximately calculate background by paying larger weight on the frequently-appearing values in the averaging process. Because SWRA is based on a running average procedure, there will not be large memory cost and computational complexity. Generally, SWRA divides the pixels at each position in the training frames into temporal segments with their own mean values and weights, and then calculates the running and weighted average result on the mean values of the segments. In the process, pixels in the same segment have the same background/foreground property and the longer segments have larger weights. This method ignores the foreground/background property of each segment, so foreground recognition is avoided. Meanwhile, low memory cost and no-delay modeling are guaranteed.

In detail, SWRA models a background value of pixels at position (x, y) by following five steps:

1) Initialize: Initialize background model value AVG and its weight W for the following weighted average procedure to 0, and then create first segment. Length of the first segment L equals to 0 and its mean value $avg = 0$. The model value before the current segment avg' is also set 0.

2) Calculate the Threshold for Segmenting: Supposing μ is the mean value and σ^2 is the mean square error, the probability of $|f(x) - \mu| > 2\sigma$ in normal distribution $f(x)$ is less than 4%. So we use 2σ as the threshold th to temporally segment a pixel in training frames. The threshold th is initialized to 14 and updated by 2 times the root square value of the mean of gap values not larger than the th before. The gap value is the difference between a pixel and its avg.

3) Create a New Segment or Widen the Current Segment: At arbitrary position, a new temporal segment will be created if $|I_i(x, y) - I_{i+1}(x, y)|$ is larger than Th_i. Otherwise, length of the current segment is widened. Through this procedure, temporally successive pixels can be divided into segments as shown in Fig. 5. Borders between segments stand for a *texture switch* on adjacent frames. Note that, if length of a segment is too short, the weight of for the segment is 0, and 1/20 of the length of training frames is used to judge whether a segment is too short.

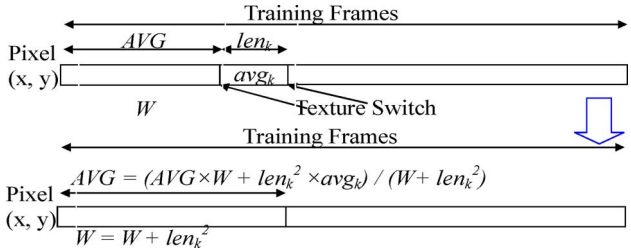


Fig. 6. The calculation of buffered AVG and W.

4) Calculate Mean Value and Weight for Each Segment: The weight of each segment is set square of its length, as shown in Fig. 5. Afterwards, denoting length and mean value of segment k as len_k and avg_k, a running average procedure will be employed to realize low computational complexity.

5) Generate and Output the Background Value: In a practical system, to satisfy low memory cost, we do not buffer the length and mean values of each segment. Instead, we just interactively buffer and calculate the total mean value AVG and its weight W from the first to the k-th segment by

$$AVG = \frac{(AVG \times W + len_k^2 \times avg_k)}{(W + len_k^2)}, \tag{19}$$

$$W = W + len_k^2. \tag{20}$$

Such calculation procedure is shown in Fig. 6. It indicates we only need to buffer and derive the AVG and W of the first k segments from the first $k-1$ segments. Following this, when the current segment reaches the end of training frames, we will calculate the final AVG and W. At last, we will obtain the required background by jointing the AVG of each pixel together.

From the above statement, we can see that the proposed SWRA works based on weights and running average. The additionally buffered data for each pixel position include: the avg_k/len_k for the current segment k, the AVG/W to summarize the previous segments and the updating threshold. Compared with the parametric methods like GMM, SWRA does not import multiple models for each pixel and never relies on the float precision calculation of proportion and variance, so both memory and time are saved; Compared with Mean-Shift, SWRA does not need to allocate large memory to buffer multiple training frames, so memory will be significantly reduced; This method is also different from non-parametric methods like codebook [20], although the codebook does not need to buffer multiple training frames, the management of the multiple codewords is very time consuming and memory sparing. In Section V, we will practically count the efficiency improvement and memory-and-time cost of SWRA.

B. Adaptive Block Classification

As discussed above, FET employs different transcoding strategies for different categories of transcoding units. Therefore, a low-complexity and scene-adaptive classification algorithm should be designed to classify units into BUs, FBUs and FUs. In our practice, an adaptive threshold Th_c is learned for each transcoding unit to judge the category S. Following

the idea that different units have different proportions of foreground, given the Th_c, S is calculated by

$$S=\begin{cases} FU, & \frac{\|\{(x,y)||C(x,y)-B(x,y)|<Th_c,0\le x,y<w\}\|}{w^2} < \alpha \\ FBU, & \alpha\le\frac{\|\{(x,y)||C(x,y)-B(x,y)|<Th_c,0\le x,y<w\}\|}{w^2} <\beta \\ BU, & \frac{\|\{(x,y)||C(x,y)-B(x,y)|<Th_c,0\le x,y<w\}\|}{w^2} \ge \beta, \end{cases} \tag{21}$$

where (x, y) is the pixel position in the current transcoding unit C, B is the reconstructed result of transcoding OB, $\|A\|$ is the number of elements in set A and w is the width of C. In practice, we usually set $\alpha = 5/64$ and $\beta = 50/64$. The remaining problem is how to adaptively calculate the threshold Th_c. To identify the foreground pixels in a new frame, a reasonable idea is to calculate a separate Th_c for each unit with help of the root-mean-square deviation σ. Following this, we propose an adaptively learning and updating algorithm as shown in Algorithm 1. The threshold calculating process for each unit can be divided into four steps: (1) Calculate the difference between the current unit and its background; (2) Utilize the threshold for the unit in the last frame to identify background pixels in the current unit; (3) Count the number of identified background pixels in the current unit; (4) Calculate the root-mean-square deviation value to update Th_c.

Algorithm 1. The Threshold Updating Model.

Input:

$I(m, n)$: the pixel value at position (m, n) of the current $w \times w$ coding unit in the current frame.

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

Initialization:

Th_c is initialized as the Th_p for co-located coding unit in the previous frame, or 14 for the first frame

Calculation:

1. For each $0 \le m, n \le w$, calculate $Diff(m, n) = |I(m, n) - B(m, n)|$,

2. For each (m, n) position, calculate

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

3. Count the potential background pixel number by

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

4. Calculate the root-mean-square deviation as the updated Th for the current coding unit

$$Th_c=2\sqrt{Round\left(\frac{\left(\sum_{m,n}(Cmp(m,n)\times Diff^2(m,n))\right)}{Sum}\right)},$$

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

Output: Th_c

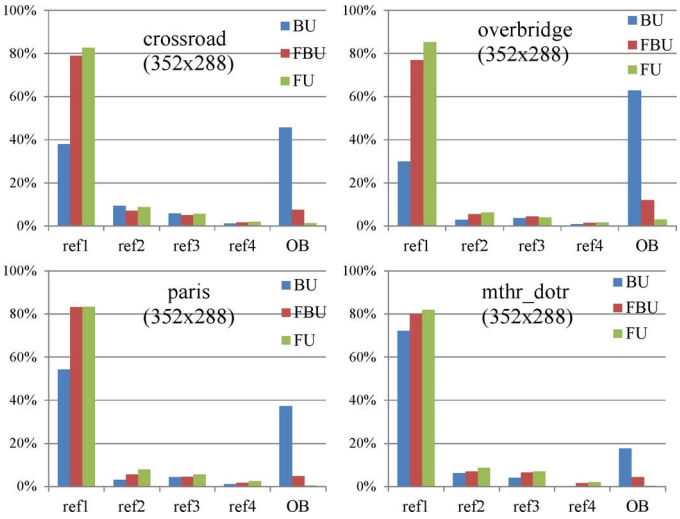


Fig. 7. The distributions of reference frames for crossroad, overbridge, mthr_dotr and pari, where ref1, ref2, ref3,ref4, OB respectively indicate the 1st, 2nd, 3rd, 4th and the long-term G-picture.

TABLE IV
THE SELECTED REFERENCE FRAME FOR EACH BLOCK CATEGORY

Three categories	BUs	FUs	FBUs
Candidate reference frame	*First, G-picture*	*First, Second, the decoded Reference frame*	*First, Second, G-picture, the decoded Reference*

C. Reference Frame Selection

To clearly and objectively analyze the distribution of the selected reference frames for different categories, the number of reference frames is set to 5 in experiments and G-picture is treated as the long-term reference frame. Respectively for BUs, FUs and FBUs, the percentage of one frame being selected as reference is calculated from the selected times of each reference frame for each category of units. Firstly as Fig. 7 shows, the first reference frame takes up more than 30%/50% for all the categories in surveillance/conference videos; the long-term G-picture takes up more than 40%/18% to predict the BUs, and more than 5% to predict FBUs. Secondly, the first two reference frames take up more than 90% to predict FUs; the first and G-picture can take 90% for BUs; the first two and G-picture together take up about 90% in BUs or FBUs.

From the statistics, we can conclude such rule for speeding up reference frame selection: only the first two reference frames are indispensable to FUs; whereas the first reference and the long-term G-picture can together provide sufficient reference for transcoding BUs; while adding the second reference, FBUs have sufficient reference. Moreover, to avoid exceptional cases, the decoded reference frame of current unit should also be utilized for FBUs and FUs. From this rule, the simplified candidate reference frame pool is shown as Table IV.

This means following selection mechanism: For BUs, only G-picture and the nearest reference frame should be added into the candidate reference frame set; For FBUs, the nearest, second nearest and G-picture should be used; For FUs, we should utilize the two nearest reference frames and the decoded reference frame. Due to the decrease of candidate reference frames in BUs/FUs/FBUs, the redundant computation in ME can be obviously reduced.

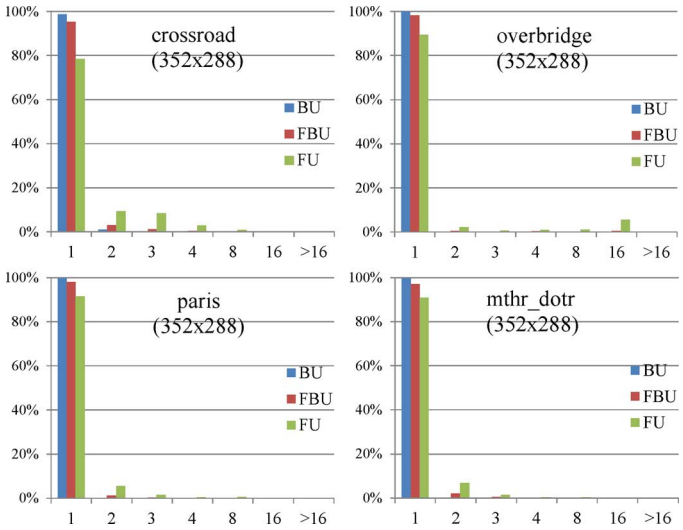


Fig. 8. The distributions of the Real MVDs for crossroad, overbridge, mthr_dotr and paris.

D. Motion Estimation Intensity Evaluation

To avoid performance loss in video transcoding, motion search range for each unit should be larger than "Real MVD". Here the so-called Real MVD means the difference between the predicted motion vector (PMV) from the neighboring units and the best matched motion vector. Fig. 8 shows the distribution of the Real MVDs for BUs, FUs and FBUs. As Fig. 8 shows, more than 99% of the Real MVDs are less than 1 integer pixel in BUs, so the transcoding integer motion search range can be set to 1. For FUs and FBUs, although the ratio of larger Real MVDs does not turn much larger (e.g., more than 10% Real MVDs is larger than 1 pixel), the increased proportion cannot be neglected because the transcoding bit-rate is more easily influenced by these larger Real MVDs.

From the statistics, we can conclude the rule for motion estimation intensity evaluation: in BUs, motion vector is close to the predicted motion vector; the motion search range should be a non-square window based on the difference between the predicted motion vector and the decoded MV; the search window should be narrowed in different degrees for FUs and FBUs. In this paper, for the i-th transcoding unit, denoting $PMVD(i,j)s$ as the difference between predicted motion vector $PMV(i,j)$ of the j-th prediction unit in the encoder and their corresponding decoded motion vector $MV_{dec}(i,j)$ from the decoder. However in the decoding process of the i-th unit, the decoded number of $MV_{dec}(i,j)$s for the prediction units is not the total prediction unit number p. Supposing there are k decoded motion vectors, we utilize $PMVD(i,j)$ to represent the largest value between $PMV(i,j)$ and $MV_{dec}(i,1) \sim MV_{dec}(i,k)$. That means, to maximum the motion estimation accuracy, we figure out the $PMVD(i,j)$ by

$$PMVD(i,j,x) = \max\{PMV(i,j,x) - MV_{dec}(i,m,x), m = 1 \sim k\} \quad (22)$$

$$PMVD(i,j,y) = \max\{PMV(i,j,y) - MV_{dec}(i,m,y), m = 1 \sim k\} \quad (23)$$

where $PMV(i,j,t)$ and $MV_{dec}(i,j,t)$ are the motion vector value of $PMV(i,j)$ and $MV_{dec}(i,j)$ in t coordinate.

According to the summarized rule, it will be enough to just employ sub-pixel motion estimation for BUs, and we should investigate on the search range calculation of FBUs and FUs, to reduce the accuracy of motion estimation in the least degree, it is intuitive that the search range in X and Y direction must be larger than the minimum value of all the $PMVD(i,1,\mathrm{x}) \sim PMVD(i,p,x)$ and $PMVD(i,1,y) \sim PMVD(i,p,y)$. Therefore, we calculate the search range $R_t(1) \sim R_t(p)$ for the total p prediction units in t coordinate by following algorithm in Algorithm 2. This algorithm can be summarized by 4 steps: (1) Calculate each prediction unit's category in {FU, FBU, BU}; (2) Fix every BU's search range R_t to 1; (3) Set the search range of FBU to be d_1 larger than the prediction unit $P(j)$'s $PMVD(i,j,t)$; (4) Set the search range of FU to be d_2 larger than $P(j)$'s $PMVD(i,j,t)$. Take a $w \times w/2$ prediction unit E as example, the search range (R_x, R_y) is shown in Fig. 9.

Algorithm 2. Search Range Calculating Algorithm.

Input value:

R_{org}: search range (SR) for the original FDFE; d_1/d_2: the extra SR for FBU/FU

Init value:

$R_t = R_{org}$, prediction unit j is namely $P(j)$, d_1 and d_2 are usually set to 2

Calculation procedure:

For $j = 1 \sim k$

$$S(j) = \begin{cases} FU, & \frac{\|\{(x,y)\|C(x,y)-B(x,y)|<Th, C(x,y)\in P(j)\}\|}{Sizeof(P(j))} < \alpha \\ FBU, & \alpha \le \frac{\|\{(x,y)\|C(x,y)-B(x,y)|<Th, C(x,y)\in P(j)\}\|}{Sizeof(P(j))} < \beta \\ BU, & \frac{\|\{(x,y)\|C(x,y)-B(x,y)|<Th, C(x,y)\in P(j)\}\|}{Sizeof(P(j))} \ge \beta \end{cases}$$

If $S(j) = BU$, **Then** $R_t(j) = 1$;

Else Begin

If $S(j) ==$ FBU, **then** $Flag = 0$; **Else**, $Flag = 1$;

If $(PMVD(i,j,t) == 0)$, **then** $R_{mod}(j) = 1 + 1 \times Flag$;

Else if $(PMVD(i,j,t) == 1)$, **then** $R_{mod}(j) = d_1 + d_2 \times Flag$;

Else if $(PMVD_{\max}(i,j,t) <= R_{org} - d_1 - d_2 \times Flag)$,

Then $R_t(j) = PMVD_{\max}(i,j,t) + d_1 + d_2 \times Flag$;

Else $R_t(j) = R_{org}$;

End

Output value: $R_t(1) \sim R_t(k)$

E. Candidate Mode Calculation

It is clearly that the used intra- and inter-prediction modes are entirely different among BUs, FUs and FBUs. Thereby the used prediction modes in transcoding units are counted to figure out

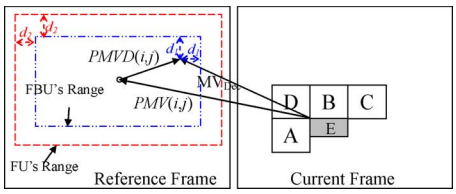


Fig. 9. PMVD and modified motion search range.

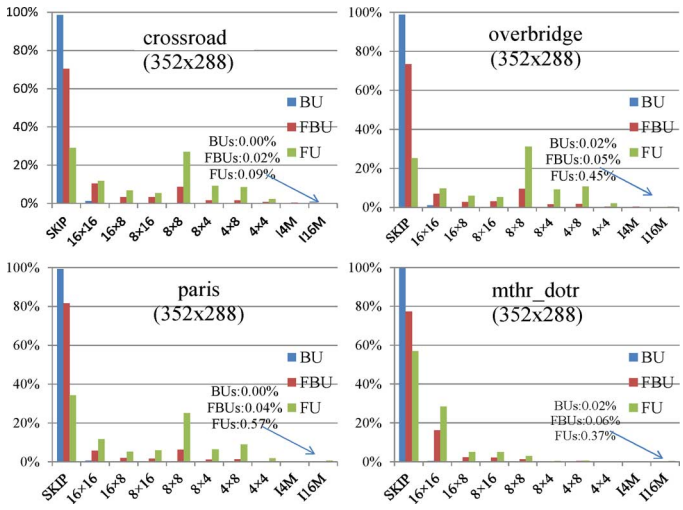


Fig. 10. The distributions of intra and inter modes for crossroad, overbridge, mthr_dotr and paris.

the proportion of each prediction mode. As Fig. 10 shows, SKIP and inter 16 × 16 prediction modes are selected almost 100% in BUs. Therefore, the intra, small and non-square modes are forbidden in BUs. For FUs and FBUs, however, although the small modes {8 × 8, 8 × 4, 4 × 8 and 4 × 4} are not used very much, there is still over 10% for them, on average. Another interesting discovery is that the Intra 16 × 16 (I16M) prediction mode is barely used in FBUs, because the flat I16M will produce large distortion for the background-and-foreground hybrid blocks.

These distributions indicate the rule for candidate mode calculation; the large and square inter-prediction modes are efficient enough for transcoding the BUs; the small-size modes for FBUs should be removed only when the decoded unit does not use any of them; the small-size modes for FUs should never be removed. According to this rule, for static regions, the large size prediction modes will be mostly selected, and smaller and non-square prediction modes like inter and intra 4 × 4(P4 × 4, I4M) and 8 × 8(P8 × 8, I8M, inter 4 × 8 and 8 × 4(P4 × 8, P4 × 8), inter 8 × 16 and 16 × 8(P8 × 16, P16 × 8) in H.264/AVC are forbidden in BUs. But these smaller inter modes should be enabled for FUs. Large intra prediction mode such as I16M in H.264/AVC should be always disabled and non-square small modes like P4 × 8 and P8 × 4 in H.264/AVC should be always enabled. The final mode decision refinement for H.264/AVC transcoding is clearly listed in Table V, where S denotes the lowest size of decode mode is equal or greater than 8 × 8 block size. As shown, the candidate prediction mode pool contains three levels, and each level has various sizes of modes.

TABLE V
THE SELECTED REFERENCE FRAME FOR EACH BLOCK CATEGORY

Decode Mode	FUs	FBUs	BUs
S	level1, I16M	level2	16×16, SKIP
P8×4, P4×8	level2, I16M		
P4×4, I4M	level3, I16M	level3	
*S: Decode Mode Size={ SKIP,P16×16, P16×8, P8×16, P8×8, I16M}, *level1={SKIP, P16×16, P16×8, P8×16, P8×8, I4M}, *level2=*level1 ∪ {P8×4, P4×8},*level3=*level2 ∪ {P4×4}			

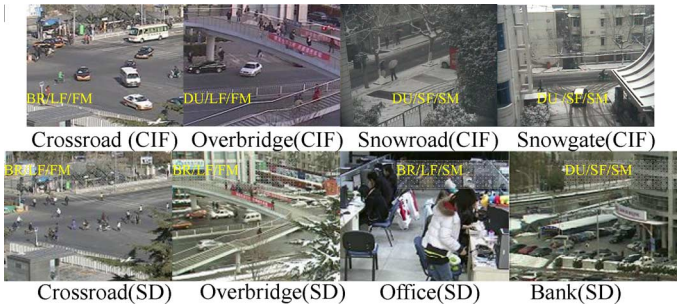


Fig. 11. Example frames of tested surveillance sequences.

Fig. 12. Example frames of tested conference sequences.

V. EXPERIMENTAL SETUP

A. Methodology

To evaluate the effectiveness and efficiency of the proposed FET, which transcodes high bit-rate input streams to low bit-rate streams in high efficiency and low complexity, extensive experiments are carried out on different kinds of surveillance and conference videos. For surveillance video, eight long ones are used, including four sequences (crossroad, overbridge, office and bank) in SD definition and four ones (crossroad, overbridge, snowroad and snowgate) in CIF. They 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. 11, 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 motion is very fast and the proportion of foreground pixels is relatively large. For conference video, eight JCT-VC videos including two CIF sequences (paris, mthr_dotr) and six 720p videos (vidyo1, vidyo3, vidyo4, johnny, KristenAndSara, FourPeople) are utilized to evaluate FET's efficiency and complexity. These conference videos can be seen from Fig. 12.

Note that, to calculate the efficiency of FET at different lower bit-rates, the input steams for all the sixteen videos above should

TABLE VI
CONFIGURATIONS OF THE USED JM17.2 HIGH PROFILE

Tools	Config.	Tools	Config.	Tools	Config.
Entropy Coding	CABAC	ME Range	64	Profile/Level	High
8x8Transform	Enable	RDO	Used	Long-term	Enable
RDO Quant.	Enable	Ref Number	5	RDO Quant.	Used
SAD Method	hadamard	Intra Period	0	ME	UMH
Modes	All Used	Deblock	Enable	1/4-pel ME	Enable

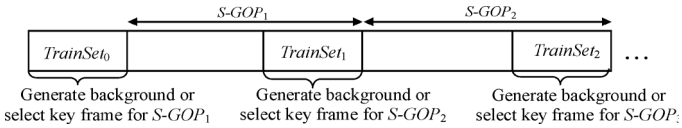


Fig. 13. Sequence structure for background generation.

TABLE VII
THE PROPORTION OF FUS, BUS AND FBUS IN TEST SEQUENCES

Surveillance Videos: Proportion for each block category	SD	bank	office	overbridge	crossroad	**Average**
	FBU	10.07%	17.16%	12.84%	26.54%	**16.65%**
	FU	2.52%	5.11%	1.84%	6.78%	**4.06%**
	BU	87.41%	77.72%	85.32%	66.69%	**79.28%**
	CIF	snowroad	Snowgate	overbridge	crossroad	**Average**
	FBU	17.80%	16.47%	25.31%	27.65%	**21.81%**
	FU	0.95%	0.53%	2.54%	5.82%	**2.46%**
	BU	81.26%	83.00%	72.15%	66.53%	**75.73%**
Conferencing Videos: Proportion for each block category	CIF	paris	mthr_dotr	**Average**	KristenAndSara	FourPeople
	FBU	37.5%	36.5%	**37.00%**	25.2%	23.2%
	FU	23.4%	13.1%	**18.25%**	12.6%	13.9%
	BU	39.1%	50.4%	**44.75%**	62.3%	62.8%
	720p	Vidyo1	Vidyo3	Vidyo4	Johnny	**Average**
	FBU	29.2%	19.5%	32.1%	19.8%	**24.83%**
	FU	11.2%	10.6%	10.1%	11.9%	**11.72%**
	BU	59.6%	69.9%	57.8%	68.2%	**63.43%**

be at high bit-rate, so these streams to be transcoded are all compressed by H.264/AVC High Profile using recommended configurations [21] with QP = 17. Besides, because low-delay characteristic is required for surveillance and conference video transcoding, an IPPP sequence structure without B frames is utilized. Moreover, the lower bit-rates refer to the following QPs: the eight surveillance videos are with QP = 22, 27, 32 and 37; the eight conference videos are with larger QP at 22, 24, 27 and 30 because the compression ratio of conference videos at similar QPs will be too large. All the methods are designed to transcode streams within H.264/AVC standard from higher bit-rates to lower bit-rates, since inside-standard transcoding will be more practical and import fewer problems for stream displaying and communication.

For an undisputed comparison, such above input streams are transcoded by following five high efficient or fast methods with the comparison tool of BD-PSNR in [22]: 1) T-AVC: In the encoding process of the transcoding procedure, it combines the decoder and encoder in the original H.264/AVC high profile of H.264/AVC test model JM17.2, which is configured as [21]. 2) T-KB: It transcodes with the high-quality key frame as the long-term reference frame. 3) FET-E: It is the FET with only the efficiency-improving techniques, that is, using proposed SWRA-based-modeling G-picture as the long-term reference frame, where G-picture is encoded by the minimum decoded QP. 4) FET-EF: It is the FET-E accelerated by the adaptive-block-classification based reference frame selection, candidate mode calculation and motion estimation intensity evaluation. 5) FET-ES: Based on FET-E, it only employs state-of-the-art fast transcoding methods to save MD and ME complexity, in forms of similar but block-classification independent speed-up strategies in FET-EF.

Through the comparison between FET-E and T-AVC/T-KB, we can figure out the transcoding performance gain or bit-rate-saving over the traditional FDFE and optimized FDFE. In further, by comparing between FET-EF and T-AVC/FET-ES, we can calculate the complexity saving of our proposed speed-up techniques with FET-E and the state-of-the-art methods. The common H.264/AVC test model JM17.2 for the transcoders is configured as Table VI.

B. Background or Key Frame Updating

Different kinds of background updating algorithms can be applied to our FET. Nevertheless, to highlight the transcoding efficiency, some factors such as the bit-allocation between background and input frames should not be taken into account in experiments, thus the background updating mechanism should be fixed and easy to implement. Therefore, the sequence structure in Fig. 13 is employed for background updating or key frame selection in all methods. In this structure, the background or key frame is updated periodically, and each background or key frame is transcoded by intra-prediction modes with the same quantization parameter. Moreover, each sequence is divided into super groups of pictures (*S-GOPs*). That is, an initial group of frames are utilized as $TrainSet_0$ to update the background frame or select a key frame for $S-GOP_1$, whereas the last group of frames in $S-GOP_1$ are utilized as $TrainSet_1$ for $S-GOP_2$, and those in $S-GOP_2$ are utilized as $TrainSet_2$ for $S-GOP_3, \ldots$ Note that, the first frame is treated as the background or key frame for $TrainSet_0$. In this way, each *S-GOP* owns the corresponding background frame for transcoding. In our experiments, the number of frames in each *TrainSet* is 120 and the length of an *S-GOP* is a function of the QP as

$$L = 300 \times \left(1 + \left\lfloor \frac{(QP-20)}{5} \right\rfloor\right). \tag{24}$$

VI. EXPERIMENTAL RESULTS

Several experiments are designed to validate the efficiency of FET. Firstly, we present the distribution of FUs, FBUs and BUs to show the effectiveness of transcoding with block classification in Section VI-A. Then, the performance gain, memory cost saving and time cost saving, brought by our proposed background modeling algorithm SWRA, are given in Section VI-B. Section VI-C introduces the total bit saving and complexity saving results for FET-E/FET-EF over the stat-of-the-art methods. At last, a practical transcoding system is implemented based on open-source X264 video codec, the appearance and efficiency of the system can be seen from Section VI-D.

A. Block Classification Results

In the first experiment, we make a statistical analysis for the distribution of FUs, BUs and FBUs. The result for each se-

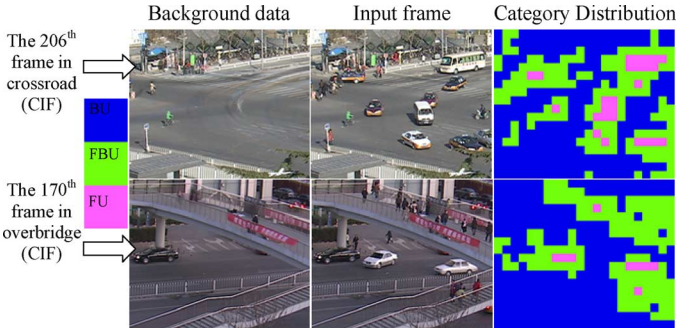


Fig. 14. Block category distribution for crossroad (CIF) and overbridge (CIF).

quence can be seen from Table VII and the block category distributions for example frames of crossroad (CIF) and overbridge (CIF) are shown in Fig. 14. From these results, we can observe the BUs take the largest part, so utilizing G-picture as long-term reference and designing specific speed-up techniques for BUs in FET will contribute a lot to the transcoding efficiency and complexity. Meanwhile, FBUs take much larger proportion than FUs. Thus after our FET saves the bit cost and complexity cost in BUs, transcoding the large amount of FBUs will consume a large percentage of the total bit cost and time cost. In such case, our employed speed-up techniques for FBUs will save the complexity in further and G-picture will also reduce the bit-rates of FBUs by providing more accurate background as reference. Moreover, although G-picture cannot provide more reference for FUs, we can design speed-up strategies to reduce the candidate reference frames for FUs. In summary, the statistics for block-category distributions indicates that designing category-adaptive speed-up strategies will be very effective for transcoding.

B. Experiment 2: Background Modeling Complexity and Efficiency of the Proposed SWRA

To evaluate the efficiency of the transcoding using SWRA, the FET-E using four state-of-the-art background modeling methods is employed as anchors for comparison, through their PSNR gains over T-AVC in high profile and IPPP structure on the first 620 frames of eight surveillance videos. The background modeling methods include the referred GMM-1, GMM-5, MS and RA. Transcoders using the methods are respectively namely FET-E-GMM-1, FET-E-GMM-5, FET-E-MS and FET-E-RA.

The transcoding performances of these anchors and our proposed FET-E-SWRA are firstly shown in Table VIII, together with their background modeling time. It indicates that FET-E-GMM-X transcoders seriously rely on the number of models utilized for each pixel. FET-E-GMM-1 achieves a much worse performance than other background modeling algorithms, and FET-E-GMM-5 achieves better performance than FET-E-RA, FET-E-MS and FET-E-GMM-1. On average, FET-E-SWRA achieves the best performance at 1.197/1.23 dB gains over T-AVC on CIF/SD sequences. FET-E-SWRA is slightly better than FET-E-GMM-5, which achieves 1.197/1.22 dB gains. Besides, FET-E-MS is proved more efficient than FET-E-RA in LF sequences, but less efficient in SF ones. As for the modeling time, The RA spares the least modeling time

TABLE VIII
BACKGROUND MODELING BASED FET-E (FET-E-X) VS. T-AVC ON PSNR GAIN (DB) AND MODELING TIME (SECOND) ON SURVEILLANCE VIDEOS

FET-E	crossroad		overbridge		bank		office		**SD Average**	
GMM-1	0.65	6.6	1.37	6.6	0.67	6.6	0.08	6.7	**0.694**	**6.6**
RA	0.93	24.4	1.73	24.5	1.24	24.4	0.30	24.5	**1.052**	**24.5**
MS	0.96	43.3	1.81	41.8	1.22	41.0	0.41	46.4	**1.097**	**43.1**
GMM-5	1.02	242.8	1.94	236.2	1.32	235.9	0.51	252.8	**1.197**	**241.9**
SWRA	1.07	10.7	1.93	10.4	1.33	10.3	0.46	10.5	**1.199**	**10.5**
FET-E	crossroad		overbridge		snowgate		snowroad		**CIF Average**	
GMM-1	0.55	1.6	0.26	1.7	1.26	1.7	0.81	1.7	**0.72**	**1.7**
RA	0.72	5.9	0.56	5.9	1.89	5.8	1.28	5.8	**1.11**	**5.9**
MS	0.78	11.5	0.61	11.3	1.77	10.4	1.17	10.3	**1.08**	**10.8**
GMM-5	0.93	61.8	0.65	61.2	1.89	57.0	1.41	56.4	**1.22**	**59.1**
SWRA	0.90	2.2	0.68	2.3	1.95	2.3	1.38	2.3	**1.23**	**2.3**

TABLE IX
MEMORY COST (BYTE) FOR EACH PIXEL IN BACKGROUND MODELING

ITEM	**RA**	**GMM-1**	**GMM-5**	**MS**	**SWRA**
Buffered pixel	1×(char)	1×(char)	1×(char)	M×(char)	**1×(char)**
Mean values	1×(float)	2×(double)	2×(double)	1	**2×(float)**
Weight	0	1×(double)	1×(double)	0	**2×(char)**
Threshold	0	1×(double)	1×(double)	0	**1×(float)**
Match points	0	1×(char)	1×(char)	0	**0**
Total	5	34	34×5 = 170	M=120	**14**

and MS spares the largest. Moreover, it shows that SWRA spares much less time than MS, GMM-1 and GMM-5 on all the sequences, only about 25% of the computing time spared by GMM-5. Moreover, SWRA is not sensitive to video content, which is quite different from GMM-X and MS.

Afterwards, another comparison for the memory cost of RA, MS, GMM-1 GMM-5 and proposed SWRA is shown in Table IX. Memory cost calculations for RA, MS, GMM-1 and GMM-5 have been referred in Section IV-A. In further for SWRA, the required memorized data for each pixel include: one current pixel with type of char and one float-precision mean value; two float-precision mean values avg'/avg and their corresponding char-type weights. In summary, the total memory cost is no more than 14 bytes for each pixel. Results show that SWRA helps to achieve better performance than other modeling algorithm, on average. Moreover, compared to the state-of-the-art GMM-5, SWRA only consumes 10% of the memory cost and spares 25% of GMM-5's modeling time.

C. Experiment 3: Total Transcoding Bit-Rate and Complexity

Table X lists the total PSNR gain and bit-rate saving of FET-E over T-AVC and T-KB for each sequence. This result can show the largest transcoding efficiency increase of the proposed FET. On average for surveillance videos, FET-E achieves bit-rate decreases of 39.84%/35.73% for SD/CIF sequences compared with T-AVC at the same PSNR, and 35.52%/27.50% over the state-of-the-art T-KB. These results correspond to 1.25/1.14 dB gains over T-AVC, whereas 0.90/0.73 dB PSNR gains over T-KB at the same bit-rate. For conference videos, FET-E achieves bit-rate decreases of 4.17%/5.85% for CIF/720p sequences compared with T-AVC at the same PSNR, and 3.34%/4.13% over the state-of-the-art T-KB. These results correspond to 0.20/0.20 dB gains over T-AVC, whereas 0.16/0.14 dB PSNR gains over T-KB at the same bit-rate. Firstly as we can see that, the less proportion of

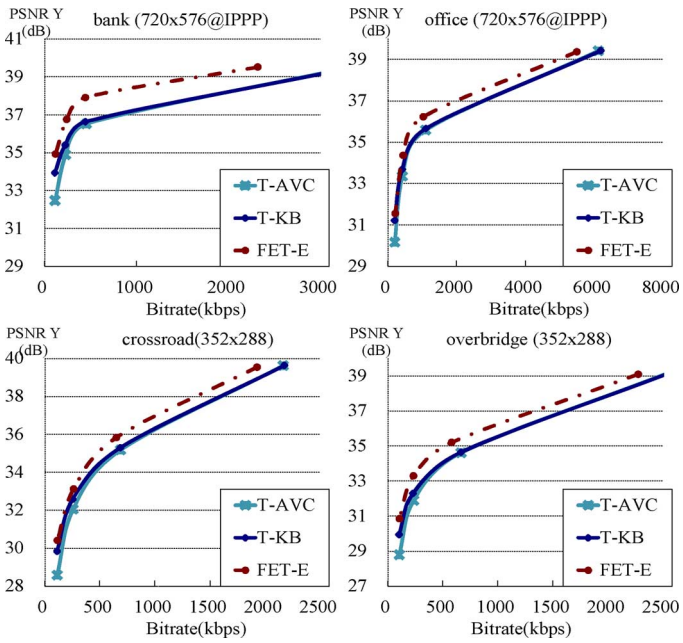


Fig. 15. The RD curves for four example surveillance videos.

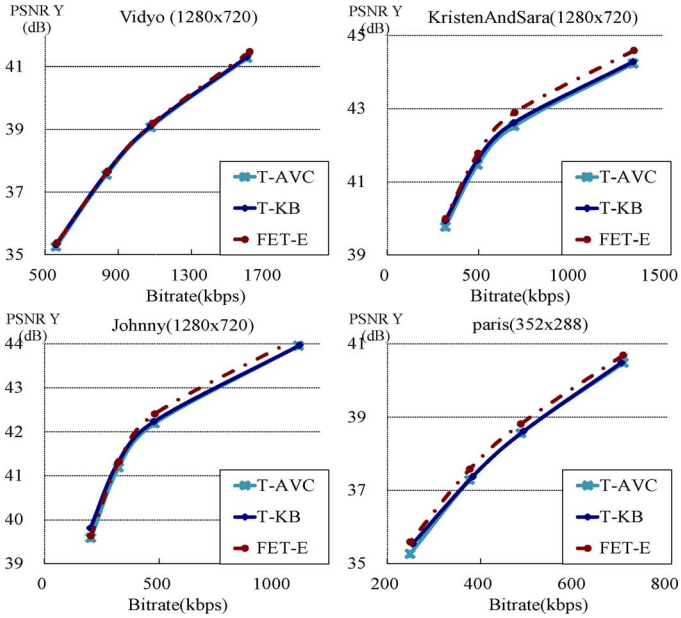


Fig. 16. The RD curves for four example conference sequences.

FBUs and BUs a sequence has, the less total bit-rate saving will be obtained (e.g., in surveillance videos, Crossroad (CIF) has the least proportion 66.53% and least bit-rate saving 17.06% over T-KB). This is because the performance gain of FET is mostly achieved on FBUs and BUs. Moreover, the transcoding efficiency increase of conference videos is much less than that of surveillance videos. This is because persons in conference videos move slightly, the exposed background regions will be much fewer than the surveillance videos in which cars or persons frequently cross the scene. Note that the transcoding efficiency RD curves of example surveillance and conference videos are shown in Figs. 15 and 16. In summary, for surveillance and conference video transcoding, FET-E saves more than 35%/5% of the bit-rate of T-AVC. Compared to the T-KB, FET-E also saves about 30%/4% of the bit-rate, on average.

The comparison results of transcoding time for FET-EF vs. FET-E and FET-ES are shown in Table XI. These results show the complexity decrease of our FET over the FET-E without speed-up techniques and the FET-ES with state-of-the-art transcoding techniques. Because we have designed specific speed-up strategies for different block classifications in the motion compensation for FET-EF, the total time decrease is very large over the anchors. Before the comparison of transcoding time, we can discover from PSNR gains in Tables X and XI that, both the PSNR decreases of the FET-EF and FET-ES compared with FET-E are less than 0.1 dB. This means the speed-up strategies still have similar PSNR gain with FET-E over T-AVC. Following this, as shown in Table XI, if we use Fast Full Search(FFS) for conference videos, FET-EF obtains as large as 15.4/7.5(CIF) and 22.3/12.0(720p) times speed up over FET-E/FET-ES, whereas the result is 16.1/7.9(CIF) and 10.0/5.3(SD) for surveillance videos. Otherwise, while Unsymmetrical-cross Multi-Hexagon grid Search(UMH) is used, for conference videos, the speed up is 2.5/2.0(CIF) and 5.3/2.1(720p) on FET-EF/FET-ES, whereas the result is 2.9/1.5(CIF) and 3.3/1.5(SD) for surveillance videos. The complexity saving in FFS is larger because motion search range reduction method will reduce the search points in full search in a large degree, but not for fast search algorithms. Table XII gives the results for search-point reduction. In summary, FET-EF saves more than 90% of the transcoding time at FS, and more than 60% at UMH. Compared to FET-ES, FET-EF saves about half the transcoding time.

D. Two Practical Systems Based on FET

To practically assess the efficiency of our method, we also employ the proposed FET method to implement two practical real-time transcoding systems for high-definition surveillance and conference videos. The first is a surveillance video transcoding system for saving the video bit-rate for storage. The second is a conference video transmission system for transcoding the source video to four different lower bit-rate videos. The appearance of the systems can be seen from Figs. 17 and 18, where kinds of transcoding options, transcoding results and information are shown. For kinds of input high-definition surveillance and conference videos in Fig. 19, the summarized performance of these systems can be shown in Table XIII. Results show that, this system can also averagely save 36.6% of the input four long-time H.264/AVC streams.

VII. Conclusion

In this paper, we propose a fast and efficient transcoding method (FET) for surveillance and conference videos based on low-complexity background modeling and adaptive block classification. Results show that, FET averagely achieves more than 35% bit saving and larger than 10 times speed-up over the traditional FDFE on the surveillance videos. On the conference videos which should be transcoded to various devices with multiple bandwidths in real-time, FET can speed up more than 20 times and still achieve 0.2 dB transcoding performance gain

TABLE X
FET-E VS. T-AVC/T-KB ON OVERALL BIT-RATE AND PSNR (DB) ON X86 PLATFORM (%)

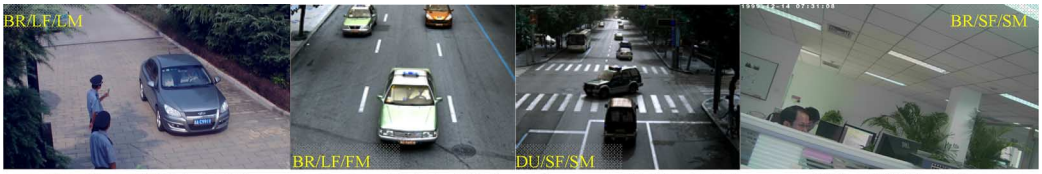
	FET-E	bank(SD)		office(SD)		overbridge(SD)		crossroad(SD)		**Average**	
Surveillance videos	**Vs. (%)**	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	**PSNR**	**bit-rate**
	T-AVC	1.38	-50.98%	0.73	-24.90%	1.66	-50.32%	1.22	-33.16%	**1.25**	**-39.84%**
	T-KB	1.06	-47.73%	0.52	-20.99%	1.17	-44.10%	0.85	-29.28%	**0.90**	**-35.52%**
	FET-E	snowroad(CIF)		snowgate(CIF)		overbridge(CIF)		crossroad(CIF)		**Average**	
	Vs. (%)	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	**PSNR**	**bit-rate**
	T-AVC	1.16	-41.93%	1.39	-48.03%	1.08	-29.71%	0.95	-23.23%	**1.14**	**-35.73%**
	T-KB	0.58	-26.31%	0.97	-39.08%	0.81	-27.56%	0.57	-17.06%	**0.73**	**-27.50%**
Conferencing videos	**FET-E**	paris		mthr_dotr		**Average for CIF**		KristenAndSara		FourPeople	
	Vs. (%)	PSNR	bit-rate	PSNR	bit-rate	**PSNR**	**bit-rate**	PSNR	bit-rate	PSNR	bit-rate
	T-AVC	0.13	-2.97%	0.27	-5.37%	**0.20**	**-4.17%**	0.32	-9.46%	0.30	-8.43%
	T-KB	0.08	-1.83%	0.24	-4.84%	**0.16**	**-3.34%**	0.22	-6.43%	0.22	-6.01%
	FET-E	Vidyo1		Vidyo3		Vidyo4		Johnny		**Average for 720p**	
	Vs. (%)	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	PSNR	bit-rate	**PSNR**	**bit-rate**
	T-AVC	0.08	-1.30%	0.19	-6.07%	0.12	-3.91%	0.17	-5.94%	**0.20**	**-5.85%**
	T-KB	0.03	-0.45%	0.16	-5.17%	0.10	-3.49%	0.10	-3.21%	**0.14**	**-4.13%**

TABLE XI
FET-EF VS. FET-E/FET-ES ON OVERALL TRANSCODING SPEED UP (TIMES) AND PSNR CHANGE (dB)

	FET-EF	bank(SD)		office(SD)		overbridge(SD)		crossroad(SD)		**Average**	
Surveillance videos	**Vs.**	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	**FFS/UMH**	**PSNR**
	T-AVC	176.3/2.5	1.30	11.5/1.7	0.66	126.8/1.9	1.58	25.0/2.3	1.12	**84.9/2.1**	**1.16**
	FET-ES	3.8/1.5	-0.01	2.3/1.5	0.00	3.4/1.5	-0.01	2.0/1.5	-0.02	**2.9/1.5**	**-0.01**
	FET-EF	snowroad(CIF)		snowgate(CIF)		overbridge(CIF)		crossroad(CIF)		**Average**	
	Vs.	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	**FFS/UMH**	**PSNR**
	T-AVC	129.3/1.6	1.09	112.8/1.8	1.32	18.0/1.3	0.98	19.5/1.7	0.86	**69.9/1.6**	**1.07**
	FET-ES	2.6/1.5	0.00	3.3/1.6	-0.02	1.8/1.0	0.00	2.0/1.7	-0.02	**2.4/1.6**	**-0.01**
Conferencing videos	**FET-EF**	paris		mthr_dotr		Average for CIF		KristenAndSara		**FourPeople**	
	Vs	FFS/UMH	PSNR	FFS/UMH	PSNR	**FFS/UMH**	**PSNR**	FFS/UMH	PSNR	FFS/UMH	PSNR
	T-AVC	31.7/2.7	0.14	73.1/2.1	0.27	**52.4/2.4**	**0.20**	23.9/1.7	0.28	40.6/2.0	0.26
	FET-ES	3.9/1.8	-0.01	3.2/1.8	0.00	**3.5/1.8**	**0.00**	2.9/1.6	0.00	3.3/1.7	0.00
	FET-EF	Vidyo1		Vidyo3		Vidyo4		Johnny		**Average for 720p**	
	Vs.	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	FFS/UMH	PSNR	**FFS/UMH**	**PSNR**
	T-AVC	70.5/1.4	0.05	65.6/2.1	0.14	33.5/1.2	0.09	26.1/1.7	0.14	**43.4/1.7**	**0.16**
	FET-ES	2.4/1.5	0.01	3.6/1.8	0.00	2.9/1.7	0.01	4.9/2.2	0.00	**3.3/1.8**	**0.00**

TABLE XII
SEARCH POINT REDUCTION PROPORTION USING FULL SEARCH AND UMHEXAGON

Surveillance	crossroad(CIF)	overbridge(CIF)	Snowroad	Snowgate	crossroad(SD)	overbridge(SD)	office	bank
Full Search	95.3%	94.9%	99.5%	99.3%	96.4%	99.5%	91.6%	99.6%
UMHexagon	65.8%	58.8%	71.9%	78.5%	65.8%	79.9%	65.7%	81.8%
Conference	mthr_dotr	FourPeople	paris	Vidyo3	KristenAndSara	Johnny	Vidyo4	Vidyo1
Full Search	92.6%	97.8%	99.1%	98.7%	96.0%	97.6%	97.2%	98.9%
UMHexagon	71.5%	78.3%	69.0%	81.6%	74.1%	79.4%	74.3%	75.8%



GateGuard(1920x1080) CarRoad(1600x1200) Crossroad(1600x1200) Office(1920x1080)

Fig. 17. Example frames of videos for testing the transcoding system.

over FDFE. The main contributions of the proposed FET are summarized as:

1) By theoretically analyzing what kind of background should be used and how the background should be quantized to improve the efficiency, FET transcoded the modeled G-pictures into stream using specially designed QP and intra prediction. And then, FET adopted the reconstructed G-pictures as long-term reference frames to significantly improve the transcoding efficiency of the following frames in surveillance and conference video. In our FET, G-picture was modeled from a low-complexity and high-efficiency background modeling algorithm.
2) Through analyzing the distributions of reference frames, motion vector and candidate prediction modes, FET

TABLE XIII
TRANSCODING PERFORMANCE OF OUR SYSTEMS ON ONE-THREAD OF GENUINE INTEL(R) CPU @ 2.66 GHZ

Vs. (%)	CarRoad	Crossroad	Office	GateGuard	Average
Source bitrate	30000 kbps	30000 kbps	4000 kbps	4000 kbps	----
frames	60321	55867	28993	38982	----
definition	1600x1200	1600x1200	1920x1080	1920x1080	----
Transcoding bitrate(kbps)	3221/616/315	3674/1012/473	692/440/282	1091/650/417	----
Transcoding speed(fps)	24.1/25.3/26.7	23.2/25.0/25.8	23.6/25.1/26.5	23.0/24.5/25.9	23.5/25.0/26.2
Bit saving vs. FDFE on x264	41.9%	35.8%	40.0%	28.6%	36.6%

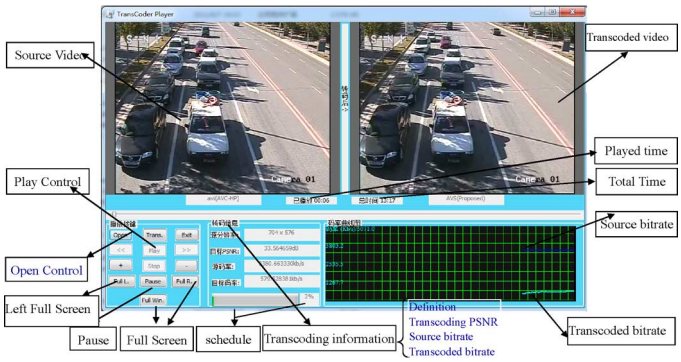


Fig. 18. Surveillance video transcoding system for saving storage.

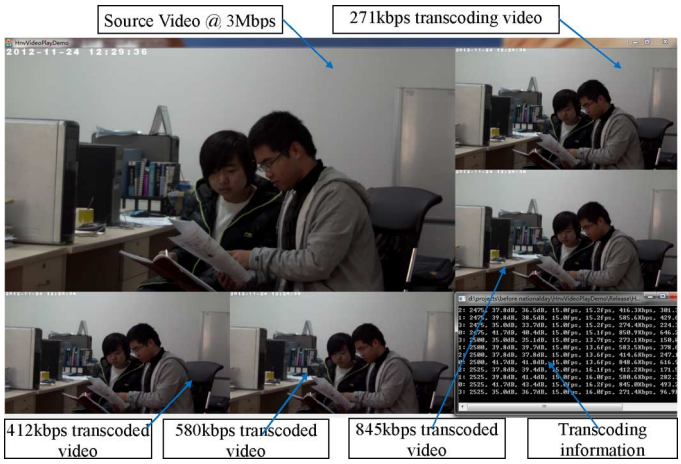


Fig. 19. Conferencing video transcoding system for video transmission.

proposed to classify blocks into three categories by an adaptive block classification based on adaptively updating thresholds. And then, FET employed different speed-up strategies for different categories to dramatically save the transcoding complexity. These strategies were in forms of reference frame selection, ME search range reduction and candidate mode calculation.

3) Extensive experiments on surveillance and conference videos were utilized to evaluate the performance of background modeling, block classification and the final transcoding efficiency and complexity. Moreover, FET was also implemented in two practical systems respectively to transcode HD surveillance videos in lower bit-rates for dramatic storage saving and real-timely transcode HD conference videos to various bit-rates for multiple-bandwidth transmission.

For future work, we will concentrate on accurate classification strategy and effective surveillance and conference video analysis technology.

APPENDIX
PROOF FOR LEMMA 2

From the rate-distortion analysis for motion compensation in [24], the $\Phi(\Lambda)$ with two hypotheses (i.e., two reference frames) is related to the accuracy of motion compensation $P(\Lambda)$ by

$$\Phi(\Lambda) = \frac{(\Phi_{nn\_1}(\Lambda) + \Phi_{nn\_2}(\Lambda))}{4} + \frac{(\Phi_{ss}(\Lambda) \times (3 + P_1(\Lambda)P_2(\Lambda) - 2P_1(\Lambda) - 2P_2(\Lambda)))}{2}. \tag{25}$$

In this equation, $\Phi_{nn\_1}(\Lambda)$ and $\Phi_{nn\_2}(\Lambda)$ are the $\Phi_{nn}(\Lambda)$s for the two prediction hypotheses, whereas $P_1(\Lambda)$ and $P_2(\Lambda)$ are their corresponding $P(\Lambda)$s. Let $\Phi_i(\Lambda)$ represent the $\Phi(\Lambda)$ using any long-term reference frame i, we can derive:

$$\Phi_i(\Lambda) = \frac{(\Phi_{nn\_s\_i}(\Lambda) + \Phi_{nn\_l\_i}(\Lambda))}{4} + \frac{(\Phi_{ss\_i}(\Lambda) \times (3 + P_{s\_i}(\Lambda)P_{l\_i}(\Lambda) - 2P_{s\_i}(\Lambda) - 2P_{l\_i}(\Lambda)))}{2}, \tag{26}$$

where $P_{s\_i}(\Lambda)$ and $P_{l\_i}(\Lambda)$ denote the $P(\Lambda)$ for the combination of short-term hypotheses and the long-term hypothesis, and $\Phi_{nn\_l\_i}(\Lambda)$ and $\Phi_{nn\_s\_i}(\Lambda)$ are corresponding $\Phi_{nn}(\Lambda)$s. Because $\Phi_i(\Lambda)$ and $\Phi_j(\Lambda)$ use the same motion search, $P_{s\_i}(\Lambda) = P_{s\_j}(\Lambda)$, $\Phi_{nn\_s\_i}(\Lambda) = \Phi_{nn\_s\_j}(\Lambda)$ and $\Phi_{ss\_i}(\Lambda) = \Phi_{ss\_j}(\Lambda)$. Therefore, the difference $\Delta\Phi_{i,j}(\Lambda)$ between $\Phi_i(\Lambda)$ and $\Phi_j(\Lambda)$ is:

$$\Delta\Phi_{i,j}(\Lambda) = \frac{(\Phi_{nn\_l\_i}(\Lambda) - \Phi_{nn\_l\_j}(\Lambda))}{4} + \left(\frac{\Phi_{ss\_i}(\Lambda) \times P_{s\_i}(\Lambda)}{2} + 1\right) \times (P_{l\_i}(\Lambda) - P_{l\_j}(\Lambda)). \tag{27}$$

According to Girod *et al.* [25], $P_{l\_x}(\Lambda)$ is determined by the displacement error variance σ_Δ of the long-term reference frame x and reflects the inaccuracy of the displacement vector used for the motion compensation. Therefore, when employing the same ME method,

$$P_{l\_i}(\Lambda) \approx P_{l\_j}(\Lambda). \tag{28}$$

From (27) and (28), we have

$$\Delta\Phi_{i,j}(\Lambda) \approx \frac{(\Phi_{nn\_l\_i}(\Lambda) - \Phi_{nn\_l\_j}(\Lambda))}{4}. \tag{29}$$

As pointed out by [14], $\Phi_{nn\lrcorner}(\Lambda)$ is determined by the prediction error variance (PEV) of the residual noise in a monotone-increasing manner. For each block at position (x, y) in k-th frame I_k among the total n frames, let $\Gamma(I_k(x, y), L_i)$ denote its PEV with L_i as long-term reference and utilize a monotone increasing function $\Psi(\Gamma(I_k(x, y), L_i))$ to represent the $\Phi_{nn}(\Lambda)$ with $\Gamma(I_k(x, y), L_i)$ as input. Then we re-write (29) as

$$\begin{aligned}&\Delta\Phi_{i,j}(\Lambda)\\&=\Phi_i(\Lambda)-\Phi_j(\Lambda)\\&\approx\frac{1}{4}\times\sum_{k=1}^{n}\sum_{x,y}\left(\Psi\left(\Gamma\left(I_k(x,y),L_i\right)\right)-\Psi\left(\Gamma\left(I_k(x,y),L_j\right)\right)\right).\end{aligned}\tag{30}$$

In *OB*, noise and foreground pixels are much fewer than the *KB* because of background generation. Besides, *OB* also has much less quality loss than the *RB* modeled from reconstructed frames. From the definition of PEV, we have

$$\begin{aligned}&\mathrm{Z}\left(I_k(x,y),RB,KB\right)\\&\quad=\min\{\Gamma\left(I_k(x,y),RB\right),\Gamma\left(I_k(x,y),KB\right)\}\\&\quad\begin{cases}\Gamma\left(I_k(x,y),OB\right)<\mathrm{Z}\left(I_k(x,y),RB,KB\right),\\\qquad\qquad I_k(x,y)\text{ is background}\\\Gamma\left(I_k(x,y),OB\right)\le\mathrm{Z}\left(I_k(x,y),RB,KB\right),\\\qquad\qquad\text{otherwise.}\end{cases}\end{aligned}\tag{31}$$

From the monotone increasing property of $\Psi(\Gamma(I_i(x, y), L_i))$, we can further derive

$$\begin{cases}\Psi\left(\Gamma\left(I_k(x,y),OB\right)\right)<\Psi\left(\mathrm{Z}\left(I_k(x,y),RB,KB\right)\right),\\\qquad\qquad I_k(x,y)\text{ is background}\\\Psi\left(\Gamma\left(I_k(x,y),OB\right)\right)\le\Psi\left(\mathrm{Z}\left(I_k(x,y),RB,KB\right)\right),\\\qquad\qquad\text{otherwise.}\end{cases}\tag{32}$$

For a decoded surveillance or conference sequence, there are lots of background pixels in each I_k. Combining the cases in (32), we have

$$\begin{aligned}&\sum_{x,y}\Psi\left(\Gamma\left(I_k(x,y),OB\right)\right)\\&<\Psi\Big(\min\Big\{\sum_{x,y}\Gamma\left(I_k(x,y),RB\right),\sum_{x,y}\Gamma\left(I_k(x,y),KB\right)\Big\}\Big).\end{aligned}\tag{33}$$

From (30) and (33), we can get

$$\begin{aligned}&\Delta\Phi_{OB,KB}(\Lambda)\\&=\Phi_{OB}(\Lambda)-\Phi_{KB}(\Lambda)\\&\approx\sum_{k=1}^{n}\Big(\Psi(\Gamma(I_k(x,y),OB))-\Psi(\sum_{x,y}(\Gamma(I_k(x,y),KB)))\Big)<0,\end{aligned}\tag{34}$$

$$\begin{aligned}&\Delta\Phi_{OB,RB}(\Lambda)\\&=\Phi_{OB}(\Lambda)-\Phi_{RB}(\Lambda)\\&\approx\sum_{k=1}^{n}\Big(\Psi(\Gamma(I_k(x,y),OB))-\Psi(\sum_{x,y}(\Gamma(I_k(x,y),RB)))\Big)<0.\end{aligned}\tag{35}$$

Therefore, we have $\Phi_{OB}(\Lambda)<\min\{\Phi_{RB}(\Lambda),\Phi_{KB}(\Lambda)\}$.

REFERENCES

[1] J. Youn, M. T. Sun, and C. W. Lin, "Motion vector refinement for high performance transcoding," *IEEE Trans. Multimedia*, vol. 1, no. 1, pp. 30–40, 1999.

[2] Y. Shin, N. Son, N. D. Toan, and G. Lee, "Low-complexity heterogeneous video transcoding by motion vector clustering," *Inf.. Sci. Applicat.*, Apr. 2010.

[3] K.-T. Fung and W.-C. Siu, "Low complexity H.263 to H.264 video transcoding using motion vector decomposition," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 908–911.

[4] H. Kalva and P. Kunselmann, "Dynamic motion estimation for transcoding P frames in H.264 to MPEG-2 transcoders," *IEEE Trans. Consum. Electron.*, vol. 54, pp. 657–662, May 2008.

[5] C. Wu and Y. Lin, "Efficient inter/intra mode decision for H.264/AVC inter frame transcoding," in *Proc. IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 3697–3700.

[6] P. Zhang, Q.-M. Huang, and W. Gao, "Key techniques of bit-rate reduction for H.264 streams," in *Proc. Pacific-Rim Conf. Multimedia*, Oct. 2004, pp. 278–281.

[7] X. Lu, A. M. Tourapis, P. Yin, and J. Boyce, "Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 1246–1249.

[8] A. Vetro, H. Sun, and Y. Wang, "Object-based transcoding for scalable quality of service," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 2000, pp. IV-17–IV-20.

[9] T. Hata *et al.*, "Surveillance system with object-aware video transcoder," in *Proc. IEEE Int. Workshop Multimedia Signal Process.*, Oct. 2005.

[10] 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.

[11] 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, 1999.

[12] Codec for Videoconferencing Using Primary Digital Group Transmission, 1984, ITU-T H.120.

[13] 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. Control, Automat., Robotics and Vision*, Mar. 2010, pp. 734–737.

[14] D. Liu, D. Zhao, X. 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, 2010.

[15] X. Zhang *et al.*, "A background model based method for transcoding surveillance videos captured by stationary camera," in *Proc. Int. Picture Coding Symp.*, Dec. 2010, pp. 78–81.

[16] M. Geng *et al.*, "A fast and performance-maintained transcoding method based on background modeling for surveillance video," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, Jul. 2012, pp. 61–66.

[17] M. Haque, M. Murshed, and M. Paul, "Improved Gaussian mixtures for robust object detection by adaptive multi-background generation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Dec. 2008.

[18] Y. Liu *et al.*, "Nonparametric background generation," *J. Vis. Commun. Image Represent.*, vol. 18, no. 3, pp. 253–263, 2007.

[19] M. Piccardi, "Background subtraction techniques: A review," in *Proc. IEEE Syst., Man and Cybern.*, Oct. 2004, pp. 3099–3104.

[20] K. Kim *et al.*, "Real-time foreground–background segmentation using codebook model," *Real-Time Imag.*, vol. 20, no. 11, pp. 172–185, 2005.

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

[22] G. Bjontegaard, "Improvements of the BD-PSNR model," in *ITU-T SC16/Q6, 35th VCEG Meeting*, Berlin, Germany, Jul. 2008, Doc.VCEG-AI11.

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

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

[25] 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.

Xianguo Zhang (S'12) received the B.S. degree in computer science and technology from Peking University, Beijing, China, in 2007. From Sep. 2007, he is working toward the Ph.D. degree in computer application technology from the Department of Computer Science and technology. Currently, he is a student in the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University. His research interests include video coding, transcoding and processing.

Tiejun Huang (M'01–SM'12) is a professor of the School of Electronic Engineering and Computer Science, Peking University, and the vice director of the National Engineering Laboratory for Video technology of China. Prof. Huang received Ph.D. degree on Pattern Recognition and Image Analysis from Huazhong (Central China) University of Science and Technology in 1998 and master and bachelor degree on computer science from Wuhan University of Technology in 1995 and 1992. His research area includes video coding, image understanding, digital right management (DRM) and digital library. He published more than sixty peer-reviewed papers and three books as author or co-author. He is the member of the Board of Director for Digital Media Project, Advisory Board of IEEE Computing Now, Editorial Board of Springer Journal on 3D Research and the Board of Chinese Institute of Electronics.

Yonghong Tian (M'05–SM'10) received the Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, China, in 2005. He is currently a Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He is the author or coauthor of over 80 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. He was the recipient of the Second Prize of National Science and Technology Progress Awards in 2010; the best performer in the TRECVID content-based copy detection (CCD) task (2010–2011); the top performer in the TRECVID retrospective surveillance event detection (SED) task (2009–2012); the winner of the WikipediaMM task in ImageCLEF 2008.

Mingchao Geng received his B.S. degree in electronic technology from Shandong University, Weihai, China, in 2009, and the M.S. degree in electronic science and technology from Peking University, Beijing, China, in 2012. When he worked as a student in the National Engineering Laboratory for Video Technology, he focused on fast surveillance video transcoding methods and practical transcoding systems.

Siwei Ma (S'03–M'12) received the B.S. degree from Shandong Normal University, Jinan, China, in 1999, and the Ph.D. degree in computer science from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. From 2005 to 2007, he was a Post-Doctorate with the University of Southern California. Then he joined the Institute of Digital Media, the School of Electronic Engineering and Computer Science, Peking University, where he is currently an Associate Professor. He published over 100 technical articles in refereed journals and proceedings in the areas of image and video coding, video processing, video streaming, and transmission.

Wen Gao (M'92–SM'05–F'09) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991. Currently, he is a Professor of the School of Electronic Engineering and Computer Science at Peking University, Beijing, China. Before joining Peking University, he was a Professor of computer science with the Harbin Institute of Technology, Harbin, China, from 1991 to 1995, and a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. 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, pattern recognition, multimedia information retrieval, multimodal interface, and bioinformatics. He has served or serves on the editorial boards of several journals, such as the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Multimedia, the IEEE Transactions on Autonomous Mental Development, the EURASIP Journal of Image Communications, and the Journal of Visual Communication and Image Representation. He has chaired a number of prestigious international conferences on multimedia and video signal processing, such as IEEE ICME and ACM Multimedia, and also has served on the advisory and technical committees of numerous professional organizations.