Selective Eigenbackground for Background Modeling and Subtraction in Crowded Scenes

Yonghong Tian, *Senior Member, IEEE,* Yaowei Wang, Zhipeng Hu, and
Tiejun Huang, *Senior Member, IEEE*

***Abstract*—Background subtraction is a fundamental preprocessing step in many surveillance video analysis tasks. In spite of significant efforts, however, background subtraction in crowded scenes remains challenging, especially, when a large number of foreground objects move slowly or just keep still. To address the problem, this paper proposes a selective eigenbackground method for background modeling and subtraction in crowded scenes. The contributions of our method are three-fold: First, instead of training eigenbackgrounds using the original video frames that may contain more or less foregrounds, a virtual frame construction algorithm is utilized to assemble clean background pixels from different original frames so as to construct some *virtual frames* as the training and update samples. This can significantly improve the purity of the trained eigenbackgrounds. Second, for a crowded scene with diversified environmental conditions (e.g., illuminations), it is difficult to use only one eigenbackground model to deal with all these variations, even using some online update strategies. Thus given several models trained offline, we utilize peak signal-to-noise ratio to adaptively choose the optimal one to initialize the online eigenbackground model. Third, to tackle the problem that not all pixels can obtain the optimal results when the reconstruction is performed at once for the whole frame, our method selects the best eigenbackground for each pixel to obtain an improved quality of the reconstructed background image. Extensive experiments on the TRECVID-SED dataset and the Road video dataset show that our method outperforms several state-of-the-art methods remarkably.**

***Index Terms*—Background subtraction, selective eigenbackground, selective model initialization, selective reconstruction, virtual frames.**

I. Introduction

BACKGROUND subtraction is a fundamental preprocessing step in many surveillance video analysis tasks. Its basic idea is to detect the foreground objects by calculating the difference between the current frame and a "background image" (or background model), and then thresholding the result to generate the objects of interest. The extracted foreground objects can then be used in the subsequent processing such as object detection/tracking and event detection. For example, in order to increase the processing efficiency, object detection and tracking only needs to focalize the foregrounds. More importantly, this is beneficial to the removal of false alarms that may be caused by background regions, consequently, resulting in an improved performance. Also, we can only extract spatial-temporal features of foreground regions to infer whether an abnormal event is happened. Obviously, more accurate the background model is, more benefits the video surveillance system will gain.

Background modeling and subtraction has been studied about two decades. There are many methods in the literature, e.g., frame differencing [1], temporal median filtering [2], [3], Gaussian mixture models (GMMs or MoGs) [7]–[10], Bayesian [12]–[14], kernel density estimation (KDE) [15], [16], codebook methods [18]–[20], and nonparametric models [32], [33]. In most of these methods, the background model is updated with the recent frames so that it can capture the scene's change without delay. In this case, most moving foregrounds can be easily detected. However, when used for the *crowded scenes* where the object density is pretty high and most of the objects move slowly or just keep still, the background model might be updated with features possibly extracted from foreground regions. As a consequence, some foregrounds would be absorbed in the reconstructed background. Such crowded scenes can be found in many real-world surveillance applications, such as airports and city crossroads. Therefore, how to perform robust background subtraction in crowded scenes poses a significant challenge.

Eigenbackground modeling [23], [24] is a frequently used method for detecting static or slowly moving foregrounds in a scene. In this method, the background is represented by M principal eigenvectors (i.e., eigenbackgrounds), which are obtained by singular value decomposition (SVD) or eigendecomposition on the covariance matrix of the training frames. Since the eigenbackgrounds are generated with randomly-selected training frames, slowly moving objects seldom appear at the same locations in these training frames and thus do not have a significant contribution to the background model. In this way, the eigenbackground method can avoid the foreground absorption effectively.

However, there are still some problems when the classic eigenbackground method is applied to crowded scenes. The first and most important one is the problem of dirty training

Manuscript received July 23, 2012; revised October 30, 2012; accepted January 10, 2013. Date of publication February 22, 2013; date of current version November 1, 2013. This work was supported in part by grants from the National Basic Research Program of China under Contract 2009CB320906, and the Chinese National Natural Science Foundation under Contract 61035001, Contract 61072095, and Contract 61171139. This paper was recommended by Associate Editor F. Lavagetto. (*Corresponding author: Y. Wang.*)

Y. Tian, Z. Hu, and T. Huang are with the School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: yhtian@pku.edu.cn; tjhuang@pku.edu.cn; zphu@jdl.ac.cn)

Y. Wang is with the Department of Electronic Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: ywwang@jdl.ac.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/TCSVT.2013.2248239

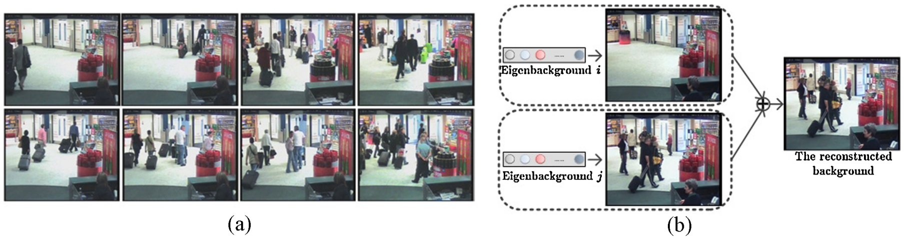


Fig. 1. Problem of dirty training samples. (a) Some frames are randomly selected to train eigenbackgrounds, but most of them contain more or less foregrounds. (b) "Noise" eigenbackgrounds are used for reconstruction, making the derived background full of some foregrounds.

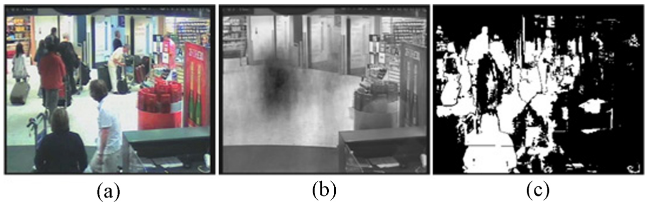


Fig. 2. Example where the frame-level eigenbackground method is applied in a crowded scene. (a) Original frame. (b) Reconstructed background. (c) Subtraction result.

samples. Consider Fig. 1 as an example. In a video where foreground objects may occlude the background for a fairly long time, it is difficult to find enough clean frames as the training samples. That is, most of the training samples may contain more or less foregrounds [as shown in Fig. 1(a)]. Accordingly, the eigenbackgrounds will be noise when trained by these dirty training samples, making the reconstructed background image also include some foregrounds [as shown in Fig. 1(b)]. A related problem is mainly due to the fact that the background reconstruction is often performed at once for the whole frame. In this case, not all pixels could get the best reconstruction results. This can be further illustrated by Fig. 2, which shows an example of the results by frame-level eigenbackground method. From Fig. 2(b), we can see that there are obviously some foregrounds (the black blobs) in the reconstructed background image, indicating that the pixels in these blobs do not obtain the optimal results via frame-based reconstruction. As a result, missing detections are occurred at the corresponding positions in the background subtraction results [as shown in Fig. 2(c)].

To address these problems, this paper proposes a selective eigenbackground method for background modeling and subtraction in crowded scenes. To keep robust in crowded scenes, our method extends the classic eigenbackground method by introducing three *selectivity* mechanisms, namely, selective training of eigenbackgrounds, selective model initialization, and pixel-level selective reconstruction. Instead of training eigenbackgrounds with the original video frames that may contain more or less foreground objects, a virtual frame construction algorithm is proposed to assemble clean background pixels from different original frames so as to construct some virtual frames as training and update samples. This selection process is implemented by frame selection map (FSM), a data structure with the same size to the video frame and each element of which records the frame index, where the corresponding location is a clean background pixel. In our method, a GMM is trained for each pixel location to identify whether a pixel at that location is clean or not. Using the constructed virtual frames as training samples, fewer foregrounds would be absorbed into the trained eigenbackgrounds and, consequently, their purity can be significantly improved. Note that virtual frames are not only constructed offline for eigenbackground training, but also updated online to refresh the eigenbackgrounds so as to capture the dynamic changes of the scene.

For a crowded scene with diversified environmental conditions (e.g., illuminations), it is difficult to use only one eigenbackground model to deal with all these variations. So we need to train several eigenbackground models offline for such a crowded scene, which are in turn used to initialize the online eigenbackground model (e.g., [24]) that will update the eigenbackgrounds incrementally to capture the temporally *local* changes in the input video. For any input video frame, peak signal-to-noise ratio (PSNR), which is originally used to evaluate the quality of the reconstructed image in visual compression, is utilized to choose the optimal one for initialization. This selective model initialization can make our method quickly adaptive to both the global and local variations of the scene. Moreover, instead of performing background reconstruction at once for the whole frame, our method selects the best eigenbackground for each pixel. Similarly, an eigenbackground selection map (ESM) is used to record the respective indices of the best eigenbackgrounds for all pixels. This can further improve the quality of the reconstructed background image. In addition, an adaptive threshold is calculated to binarize the difference between the background image and the input frame so as to generate the foreground objects.

Extensive experiments were performed on the TRECVID surveillance event detection (SED) corpus [5] and the Road video dataset that was collected from a city crossroad surveillance system. Experimental results show that on both indoor and outdoor crowded videos, the proposed method outperformed several state-of-the-art eigenbackground methods (e.g., the classic and block-based eigenbackground methods) and noneigenbackground methods (e.g., GMM [9], Bayes [12], CodeBook [18], PBAS [32], and ViBe [33]) remarkably.

The remainder of this paper is organized as follows. Section II briefly reviews the eigenbackground methods. The proposed method is presented in Section III. Experimental results are shown in Section IV. Section V concludes this paper.

A preliminary version of this paper has been published in [30]. The main extensions include selective model initialization, where PSNR is utilized to choose the optimal model trained offline to initialize the online eigenbackgrounds, an improved background reconstruction algorithm by introducing the influence of neighboring pixels, an adaptive thresholding for background subtraction, and more extensive experiments on the TRECVID-SED and Road video datasets.

II. Eigenbackground

In general, eigenbackground methods have two typical background modeling strategies, namely, offline and online. The offline strategy (e.g., Batch PCA [23]) tries to learn the globally optimal eigenbackgrounds from the training set and keep them fixed during the detection procedure, while the online one (e.g., CCIPCA [24]) mainly focuses on how to incrementally update the eigenbackgrounds. Typically, the two modeling strategies can be used in different scenarios in real-world surveillance applications.

Batch PCA [23] requires that all the training images are available before the principal components can be estimated. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ denote N training samples, where $\mathbf{x}_n$ is the column vector representation of the n^{th} sample. That is, each sample is treated as one vector, simply by concatenating the rows of pixels in the original image. Then, the principal components, also known as eigenbackgrounds, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_M]$, can be obtained by SVD on the covariance matrix $\mathbf{C} = \frac{1}{N}\sum_{n=1}^{N}[\mathbf{x}_n - \bar{\mathbf{x}}][\mathbf{x}_n - \bar{\mathbf{x}}]^{\mathrm{T}}$, where $\bar{\mathbf{x}}$ is the mean vector of $\{\mathbf{x}_n\}_{n=1,\ldots,N}$. Let the SVD of $\mathbf{C}$ be $\mathbf{C} = \mathbf{W}\Sigma\mathbf{W}^{\mathrm{T}}$, then $\mathbf{U}$ is defined as the M eigenvectors in $\mathbf{W}$ that correspond to the M largest eigenvalues. Once a new image, $\mathbf{x}$, is available, the reconstructed background can be calculated as

$$\mathbf{b} = \mathbf{U}\mathbf{U}^{\mathrm{T}}(\mathbf{x} - \bar{\mathbf{x}}) + \bar{\mathbf{x}}. \tag{1}$$

The batch PCA looks like fairly simple, but in real-world applications this approach rarely works alone since the offline trained eigenbackgrounds are not able to capture any changes of a scene such as lighting changes, moving background objects and nonstationary scenes. To capture such dynamic changes, incremental PCA (IPCA) was proposed to compute the principal components for observations arriving sequentially, and then update the eigenbackground model online. Researchers have proposed many different variations of IPCA. Some variations try to obtain principal components with the original training samples and a newly added sample [21], [27]–[29], while some others avoid computing the covariance matrix as an intermediate result [24]–[26]. This is important since when the dimension of a video frame is high, both the computation and storage complexity of such a matrix grow dramatically. For example, the covariance matrix amounts to 8.5997e+10 elements for a standard-definition video frame of 720 × 576. To tackle this problem, Weng *et al.* [24] proposed the candid covariance-free IPCA (CCIPCA) algorithm that is fast in convergence rate and low in the computational complexity.

The basic idea of CCIPCA is to compute the eigenbackgrounds of a sequence of samples incrementally without estimating the covariance matrix (thus, covariance-free). Let $\mathbf{x}_n$ denote the nth sample vector that is acquired sequentially, then the mean vector $\bar{\mathbf{x}}^{(n)}$ can be computed incrementally

$$\bar{\mathbf{x}}^{(n)} = \frac{n-1}{n}\bar{\mathbf{x}}^{(n-1)} + \frac{1}{n}\mathbf{x}_n \tag{2}$$

where $\bar{\mathbf{x}}^{(n-1)}$ is the corresponding mean vector before $\mathbf{x}_n$ is available. Accordingly, the zero-mean form of $\mathbf{x}_n$ can be easily derived as $\mathbf{z}_n = \mathbf{x}_n - \bar{\mathbf{x}}^{(n)}$. In this case, the ith dominant principal component $\mathbf{u}_i^{(n)}$ can be obtained as follows [24]: Let $\mathbf{z}_1^{(n)} = \mathbf{z}_n$

$$\mathbf{u}_i^{(n)} = \frac{n-1-\zeta}{n}\mathbf{u}_i^{(n-1)} + \frac{1+\zeta}{n}\mathbf{z}_i^{(n)}(\mathbf{z}_i^{(n)})^{\mathrm{T}}\frac{\mathbf{u}_i^{(n-1)}}{\|\mathbf{u}_i^{(n-1)}\|} \tag{3}$$

$$\mathbf{z}_{i+1}^{(n)} = \mathbf{z}_i^{(n)} - (\mathbf{z}_i^{(n)})^{\mathrm{T}}\frac{\mathbf{u}_i^{(n)}}{\|\mathbf{u}_i^{(n)}\|}\frac{\mathbf{u}_i^{(n)}}{\|\mathbf{u}_i^{(n)}\|} \tag{4}$$

where ζ is the updating rate. After normalization, the final eigenvector and eigenvalue are respectively given by $\mathbf{u}_i = \frac{\mathbf{u}_i^{(n)}}{\|\mathbf{u}_i^{(n)}\|}$ and $\lambda_i = \|\mathbf{u}_i^{(n)}\|$. Note that (3) and (4) represent an iterative computation procedure, where the higher order eigenvectors are obtained in the complementary space of the lower order eigenvectors. The algorithm can run in real-time because it avoids the computation of covariance matrix and each eigenvector updates in only one step.

However, when applied in a video with crowded scenes, the trained eigenbackgrounds in CCIPCA may contain more or less foregrounds since samples available incrementally are likely to be dirty. Therefore, it is preferable to combine the two kinds of eigenbackground methods in a unified framework. In other words, a batch PCA is utilized to train several eigenbackground models offline using clean training samples that are selected or constructed from the training set, while CCIPCA is used to online update the current eigenbackground model. When initialized by one of the offline trained models, CCIPCA can even achieve higher performance by capturing the sharply changes of the scene.

In our previous work [17], we proposed a block-level eigenbackground algorithm, where the original video frame is divided into blocks and the eigenbackground training and background subtraction are performed for each block independently. This paper further extends the algorithm by performing the selection of the best eigenbackground for each pixel. This can further reduce the foreground information in the reconstructed background, consequently, leading to a superior performance when applied in video with crowded scenes.

III. Selective Eigenbackground Method

The main objective of this paper is to present an effective eigenbackground method that can keep robust in crowded

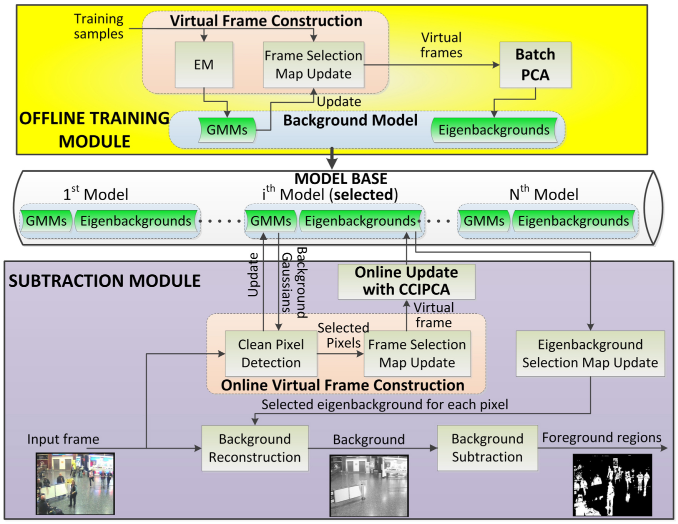


Fig. 3. Framework of the proposed method, which consists of two modules, i.e., the offline training module and the subtraction module. Note that the subtraction module involves online update of the background model with virtual frames.

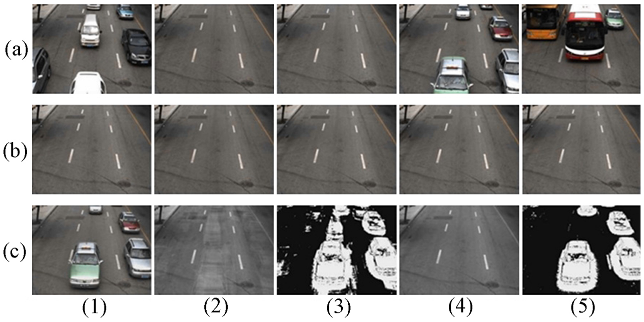


Fig. 4. Comparison of background modeling and subtraction results given different training samples. (a) Randomly selected frames. (b) Clean frames. (c) Background reconstruction and subtraction using the two kinds of training samples, where (c1) an original video frame, (c2) and (c3) the reconstructed background and the corresponding subtraction result with training frames shown in (a), and (c4) and (c5) those with training frames shown in (b).

scenes. Our method exploits three "selectivity" mechanisms for background modeling and subtraction, including automatically constructing virtual frames as the training and update samples of eigenbackgrounds (called selective training), adaptively choosing the optimal background model for initialization (called selective model initialization) and selecting the best eigenbackground for each pixel to reconstruct its background pixel (called pixel-level selective reconstruction). Using these mechanisms, our method can significantly increase the purity of the trained eigenbackgrounds and, consequently, obtain an improved quality of the reconstructed background image.

As shown in Fig. 3, our method mainly consists of two modules, namely, the offline training module and the subtraction module. In the offline training module, a GMM is first trained for each pixel through the Expectation Maximization (EM) algorithm [6]. In each GMM, some components represent the background in the scene (hereafter, referred to as "background Gaussians"), while the others characterize the foreground. Thus, for a given pixel, we can identify whether it is a clean background pixel by comparing its pixel value with the background Gaussians. Then a data structure with the same size to the video frame, FSM, is used to record the frame index of a training sample in which the corresponding location is a clean background pixel. When a FSM is crammed with the frame indices, all the corresponding pixels are assembled to construct a virtual frame. As such, the virtual frame does not contain any foreground pixels. Therefore, if these virtual frames are utilized by batch PCA to train the eigenbackground models, fewer foregrounds would be absorbed into the trained eigenbackgrounds and thus their purity can be significantly improved.

In the subtraction module, when a video frame is given, the optimal background model trained offline is first selected to initialize the GMMs and eigenbackgrounds using a PSNR-based algorithm. Then, we update the GMM for each pixel, and then use the updated GMM to identify whether the corresponding pixel is a clean background pixel. If so, the corresponding frame index will be used to update the FSM. After that, all the corresponding pixels in the updated FSM are assembled to construct a virtual frame, which is in turn used to online update the eigenbackgrounds by CCIPCA. Given the updated eigenbackgrounds, the next step is to reconstruct a background image for the input video frame. In our method, an ESM is used to record the respective index of the best eigenbackground for each pixel. Using the best eigenbackgrounds for all pixels indicated by the ESM, a background image with the improved quality can be reconstructed for the input video frame. Finally, the difference between the background image and the input frame is calculated and an adaptive threshold is used to binarize the difference so as to generate the foreground objects.

Note that our method employs two kinds of background models, namely, GMM and eigenbackground model. GMM is used in the virtual frame construction to identify a pixel as background or foreground. However, since it ignores the correlation of neighboring pixels, some random noises will be inevitably included into the constructed virtual frames. On the contrary, the eigenbackground model exploits the correlation of pixels to reduce the influence of random noises. Therefore, by using GMMs to generate the training and update samples for eigenbackgrounds, our method can make use of the complementarity of the two background modeling techniques.

A. *Selective Training With Virtual Frames*

Intuitively, the background model would be better when trained by clean samples than trained by randomly-selected samples, and accordingly the background subtraction results would have higher recall and precision. Note that here clean means that there are no foregrounds in the training sample. This conjecture can be validated by the example shown in Fig. 4, which illustrates a comparison of background modeling and subtraction results given the two kinds of training samples. We can see that the background subtraction result with clean training frames is much better than that with randomly selected frames.

In a real-world crowded video, however, it is often difficult to find enough clean frames as the training samples

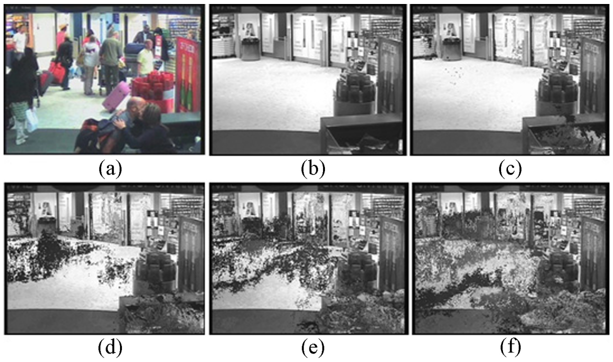


Fig. 5. Crowded scene and five images that depict the means of different Gaussian components for all its pixels.

[e.g., Fig. 1(a)]. Therefore, a feasible solution is to automatically construct some images by assembling background pixels from different original frames. Since these images are not directly selected from the original video, they are referred to as "virtual frames" in this paper. To construct such a virtual frame, there are two problems to be solved, i.e., how to determine whether a pixel is clean, and how to organize these clean pixels to form the virtual frame.

1) *Detection of Clean Pixels:* The key problem in detecting clean pixels is to derive a statistical model that can effectively characterize the background property of a pixel. Following the methods in [7], we employ the GMM model to describe which kind of pixel is really a background pixel

$$P(x) = \sum_{i=1}^{K} \omega_i \eta(x, \mu_i, \sigma_i) \tag{5}$$

$$\eta(x, \mu_i, \sigma_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\} \tag{6}$$

where K is the number of Gaussians ($K = 5$ in this paper), ω_i, μ_i, and σ_i are the weight, mean, and variance of the ith Gaussian, respectively. In each GMM, some components represent the background in the scene (i.e., "background Gaussians"), while the others characterize the foreground. This can be further illustrated by Fig. 5, which contains five images that depict the means of different Gaussian components for all pixels in a crowded scene. We can see that the first two components are likely to be background Gaussians, while the remaining components tend to contain more foreground information.

Statistically, the duration of the background occurred at a pixel location is often longer than that of the foreground. Moreover, the background usually remains unchanged in a short period while the foreground objects, whether they shade the background or not, may vary by time. As a result, the visual appearance of that pixel should be more stable if it is a background pixel. In a GMM, the occurrence duration of the ith Gaussian may influence its weight ω_i, while its stability can be characterized by the corresponding variance σ_i. As discussed above, compared with a foreground Gaussian, the ω for a background Gaussian should be bigger while its σ should be smaller. In this case, if we sort the Gaussians in a pixel's GMM by the ratio ω/σ in a descending order, then the top B Gaussians can be classified as background Gaussians, namely

$$\arg\max_B \left\{\sum_{i=1}^{B} \omega_i < T_\perp\right\} \tag{7}$$

where $T_\perp$ is an experimental threshold that denotes the upper bound of the duration proportion of a background pixel in the whole video ($T_\perp = 0.6$ in this paper). The statistical results on the TRECVID-SED dataset show that about 90% pixels take the first two Gaussians as background Gaussians.

After distinguishing the two kinds of Gaussian components, it is easy to identify whether a pixel in a frame is clean by comparing its pixel value with the background Gaussians. That is given by

$$|x - \mu_i| < 2.5\sigma_i, \ \exists\ i, \ 1 \le i \le B \tag{8}$$

where x is the value of a pixel to be determined, μ_i and σ_i are the mean and variance of the ith Gaussian component at the corresponding location. That is, if the absolute difference between the x's pixel value and the mean μ_i of the ith Gaussian component is less than 2.5 times the variance σ_i, then x is considered as a match with this Gaussian component. In this way, all background pixels in the training frames can be picked out to construct virtual frames.

2) *Construction of Virtual Frames:* Given the selected background pixels, the next problem is how to organize these pixels to form virtual frames. In our method, a matrix with the same size to the video frame, referred to as FSM, is used to record the latest frame index of a training sample, wherein the corresponding location is a clean background pixel.

Fig. 6 presents an example to visualize the virtual frame construction. For simplicity, here we suppose the frame size is 3 × 3. All elements of the FSM are initialized as "−1" and will be updated when a new background pixel is found from the original frames. For example, the $(0, 0)$ element in the FSM is crammed with the frame index "34" after scanning the 34th frame and if no clean background pixel is found at the $(0, 0)$ location in the following frames; similarly, the $(1, 0)$ element in the FSM is first crammed with the frame index "27," and then updated with the frame index "43," if the $(1, 0)$ pixels in both frames are identified as a clean background pixel. In the process, the elements in the FSM always remain fresh so as to capture the recent changes in the video. When all elements in the FSM are assigned, the corresponding pixels are then assembled to construct a virtual frame (e.g., the $(0, 0)$ pixel of the 34th frame, the $(1, 0)$ pixel of the 43th frame, and so on). After that, the FSM is reset to "−1" and the above process will be repeated. Algorithm 1 describes this virtual frame construction process.

Fig. 7 shows some examples of virtual frames constructed on the TRECVID-SED dataset. Obviously, these virtual frames are much cleaner than those frames in Fig. 1. Nevertheless, there are still some fragmentary foregrounds in some virtual frames (as shown in the second row of Fig. 7). This is because some foreground pixels really exist in all the training samples, and consequently these foregrounds are modeled as backgrounds by GMMs.

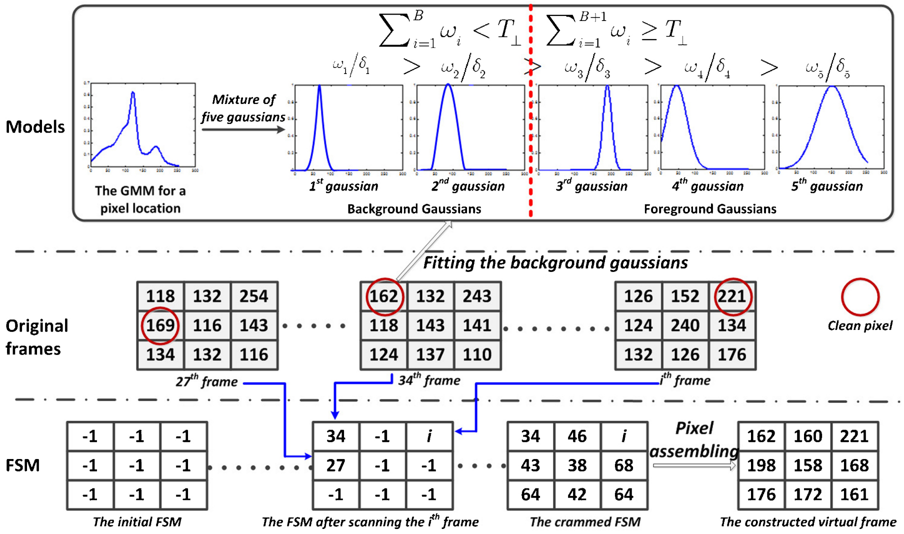


Fig. 6. Visualization of virtual frame construction. In this example, the $(1, 0)$ element in the FSM is first crammed with the frame index "27", and then updated with the frame index "43," if the $(1, 0)$ pixels in both frames are identified as a clean background pixel. When all elements in the FSM are assigned, the corresponding pixels are then assembled to construct a virtual frame (e.g., the $(0, 0)$ pixel of the 34th frame).

Algorithm 1: Virtual frames construction with FSM.

```
Input: The training frame set $\mathcal{I} = [\mathbf{I}_1, \ldots, \mathbf{I}_t, \ldots, \mathbf{I}_N]$, where $\mathbf{I}_t$
       is a $m \times n$ video frame; and the GMM models
       $\Phi = \{\Phi_{i,j}\}_{m\times n}$ for all locations in a frame, where
       $\Phi_{i,j} = \{\omega^k_{i,j}, \mu^k_{i,j}, \sigma^k_{i,j}\}_{k\in[1,K]}$, $K$ is the number of
       Gaussians and $B$ is the number of background
       Gaussians.
Output: A set of virtual frames $\mathcal{V} = [\mathbf{V}_1, \ldots, \mathbf{V}_l, \ldots, \mathbf{V}_L]$.
begin
    $l = 1; t = 1;$
    while $l <= L$ do
        Initialize the FSM $\mathbf{F} = [-1]_{m\times n}$ ;
        while $\exists(i, j), \mathbf{F}(i, j) == -1$ do
            if $\exists\ k,\ |\mathbf{I}_t(i, j) - \mu^k_{i,j}| < 2.5\sigma^k_{i,j},\ \ 1 \leq k \leq B,$
                $\mathbf{F}(i, j) = t;$
            else
                $t = t + 1;$
            if $t > N$ return;
        end
        for all $(i, j)$ do
            $\mathbf{V}_l(i, j) = \mathbf{I}_{\mathbf{F}(i,j)}(i, j)$ ;
        end
        $l = l + 1;$
    end
end
```

3) *Discussion:* Although the constructed virtual frames are visually similar to the background images for the given scenes, they cannot be directly used for background subtraction. This is due to the following two reasons. First, since a virtual frame is constructed by independently assembling clean pixels from different original frames, the correlation of neighboring pixels is ignored in this process. As a result, there are some random noises in the constructed virtual frames (e.g., the first row in Fig. 7). In this case, if these virtual frames are directly used for background subtraction, a large amount of random noises will also exist in the subtraction results. Instead, our method exploits these virtual frames as the training samples for eigenbackgrounds so that these low-frequency random noises can be removed. Second, a virtual frame may contain pixels from different training samples with diversified scene variations (e.g., illuminations). So if these virtual frames are directly used for background subtraction, it is impossible to find out the optimal one as the background image for any video frame. Instead, our method trains several eigenbackgrounds, each of which may capture one of the variations, from these virtual frames. Then they can be used to selectively reconstruct a more accurate background for a given input frame.

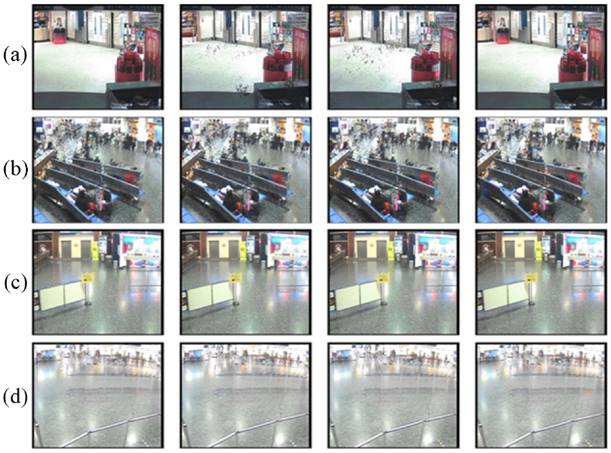


Fig. 7. Some examples of virtual frames constructed on the TRECVID-SED dataset for (a) Camera 1, (b) Camera 2, (c) Camera 3, and (d) Camera 5.

B. Selective Model Initialization and Update

Besides selective training, how to initialize and update the background models is another important issue. In our method, this issue is related to two kinds of background models, namely GMMs and eigenbackground models.

1) *Initialization of Background Models:* Given a set of training samples, we can use the EM algorithm described in [6] to train GMMs, and the Batch PCA [23] to train

the eigenbackground models. As discussed above, a set of background models should be trained offline for a crowded scene, each of which is used to characterize the background properties of the scene in a specific environmental condition. Then, one of the offline trained models can be used to initialize the online eigenbackground model, which is, in turn, devoted to capture some local changes in the same environmental condition. Thus, the problem becomes how to select the optimal background model trained offline for initialization.

Similar to the concept "groups of pictures" (GOPs) in video coding, we first divide each video sequence into different supergroups of pictures (S-GOPs). Each S-GOP contains a large amount of successive frames such that they can be initialized by the same offline trained model. In each S-GOP, we select a small number of initial frames (e.g., 30 frames) as the validate set to determine the optimal model for initialization. Let Ψ denote a set of offline trained models, our objective here is to select the best $\psi_k \in \Psi$ such that for each frame $\mathbf{I}_t$ in the validate set, $\mathcal{G}$, the background $\mathbf{B}_{k,t}$ reconstructed by ψ_k is closest to the ideal background $\mathbf{B}_t^*$.

According to [31], the observation of a video frame $\mathbf{I}_t$ can be modeled as the sum of the ideal background $\mathbf{B}_t^*$, the system noise $\mathbf{N}_t^{(\mathrm{Sys})}$, moving objects $\mathbf{M}_t^{(\mathrm{Obj})}$, and moving background $\mathbf{M}_t^{(Bg)}$, that is

$$\mathbf{I}_t = \mathbf{B}_t^* + \mathbf{N}_t^{(\mathrm{Sys})} + \mathbf{M}_t^{(\mathrm{Obj})} + \mathbf{M}_t^{(\mathrm{Bg})}. \tag{9}$$

Since the local background changes in each S-GOP will be captured by the online eigenbackground algorithm, we assume $\mathbf{M}_t^{(\mathrm{Bg})} = 0$ in $\mathcal{G}$. Without considering the system noise for simplicity, we have

$$\mathbf{B}_t^* = \mathbf{I}_t - \mathbf{M}_t^{(\mathrm{Obj})}. \tag{10}$$

Thus, given a set of offline trained models $\Psi = \{\psi_k\}$, the selection algorithm can be expressed as follows:

$$\begin{aligned} k^* &= \arg\max_k \left\{ \sum_{\mathbf{I}_t \in \mathcal{G}} J(\mathbf{B}_{k,t}, \mathbf{B}_t^*) \ \mid \psi_k \in \Psi \right\} \\ &= \arg\max_k \left\{ \sum_{\mathbf{I}_t \in \mathcal{G}} J(\mathbf{B}_{k,t}, \mathbf{I}_t - \mathbf{M}_t^{(\mathrm{Obj})}) \ \mid \psi_k \in \Psi \right\} \end{aligned} \tag{11}$$

where $J(\cdot,\cdot)$ is a monotonic function to evaluate the similarity between two images, $\mathbf{B}_{k,t}$ is the reconstructed background for $\mathbf{I}_t$ with ψ_k. Due to its monotonicity, we have $J(\mathbf{B}_{k,t}, \mathbf{I}_t - \mathbf{M}_t^{(Obj)}) \propto J(\mathbf{B}_{k,t}, \mathbf{I}_t) - J(\mathbf{B}_{k,t}, \mathbf{M}_t^{(Obj)})$. Then, (11) can be rewritten as follows:

$$k^* = \arg\max_k \left\{ \sum_{\mathbf{I}_t \in \mathcal{G}} J(\mathbf{B}_{k,t}, \mathbf{I}_t) - J(\mathbf{B}_{k,t}, \mathbf{M}_t^{(\mathrm{Obj})}) \ \mid \psi_k \in \Psi \right\} \tag{12}$$

If $\mathbf{M}_t^{(\mathrm{Obj})}$ for each $\mathbf{I}_t$ is known, namely all frames in $\mathcal{G}$ are manually labeled with the ground truth of foreground objects, we can directly apply (12) to choose the optimal initializing background model. However, manually labeling such a validation set for every S-GOP is very time-consuming, and thus impossible to be widely applied on large datasets or real-world surveillance systems.

We observe that for each $\mathbf{I}_t \in \mathcal{G}$, $J(\mathbf{B}_{k,t}, \mathbf{M}_t^{(Obj)}) \geq 0$, where the equality is satisfied if $\mathbf{I}_t$ is a clean frame. This enlightens us that we can only evaluate the similarity between the background regions of $\mathbf{I}_t$ (denoted by $\hat{\mathbf{I}}_t$) and the corresponding parts in $\mathbf{B}_{k,t}$ (denoted by $\hat{\mathbf{B}}_{k,t}$). This is reasonable since according to our assumption, each model $\psi_k \in \Psi$ is trained to characterize the background properties of the scene in a specific environmental condition, and thus $\hat{\mathbf{B}}_{k,t}$ are enough to evaluate whether the model ψ_k can well fit the background properties of $\mathbf{I}_t$. Following this idea, let $\dot{\mathbf{I}}_t$ denote a reference frame for $\mathbf{I}_t$ (e.g., a clean frame or an initial frame in $\mathcal{G}$), thus $\hat{\mathbf{I}}_t = |\mathbf{I}_t - \dot{\mathbf{I}}_t|_\Delta$ where Δ is the subtraction threshold. Then, (12) can be approximated by

$$k^* \approx \arg\max_k \left\{ \sum_{\mathbf{I}_t \in \mathcal{G}} J(\hat{\mathbf{B}}_{k,t}, \hat{\mathbf{I}}_t) \ \mid \psi_k \in \Psi \right\} \tag{13}$$

Thus, the remaining problem is how to design $J(\cdot,\cdot)$ that can accurately measure the similarity between $\hat{\mathbf{B}}_{k,t}$ and $\hat{\mathbf{I}}_t$. In image processing, PSNR is exactly such an evaluation function, which is originally used to evaluate the quality of the reconstructed image compared with the original image. In this paper, we introduce the PSNR to the task of background model selection. For each $\mathbf{I}_t \in \mathcal{G}$, the optimal initializing background model should be the one with the highest PSNR when it is used to reconstruct the background image for each $\mathbf{I}_t$. That is given by

$$\begin{aligned} k^* &\approx \arg\max_k \sum_{\mathbf{I}_t \in \mathcal{G}} \mathrm{PSNR}(\hat{\mathbf{B}}_{k,t}, \hat{\mathbf{I}}_t) \\ &= \arg\max_k \left\{ \sum_{\mathbf{I}_t \in \mathcal{G}} 20 \log_{10} \frac{255}{\sqrt{MSE(\hat{\mathbf{B}}_{k,t}, \hat{\mathbf{I}}_t)}} \right\} \end{aligned} \tag{14}$$

where

$$MSE(\hat{\mathbf{B}}_{k,t}, \hat{\mathbf{I}}_t) = \frac{1}{m \times n} \sum_{(i,j)} [\hat{\mathbf{I}}_t(i,j) - \hat{\mathbf{B}}_{k,t}(i,j)]^2. \tag{15}$$

A simple experiment was performed to verify the effectiveness of the PSNR-based model selection. From the TRECVID-SED corpus, we randomly selected 12 video sequences for training, and the other 12 sequences for testing. Note that each pair of training and testing sequences were captured with similar illumination conditions (e.g., respectively from the same time in two days). We trained 12 background models on these training sequences, and randomly extracted 30 frames as the validation set from each testing sequence. Given one of the models, PSNR scores were calculated for all frames in each validate set. The experimental results show that all test sequences could correctly find the most suitable initializing background models. As an example, Fig. 8 shows the selection results of the second and fifth test sequences. We can see that although some wrong decisions are made due to the approximation by (13) [e.g., the 21st, 22nd, and 26th frames in Fig. 8(a)], most of the frames can find the correct initializing background model.

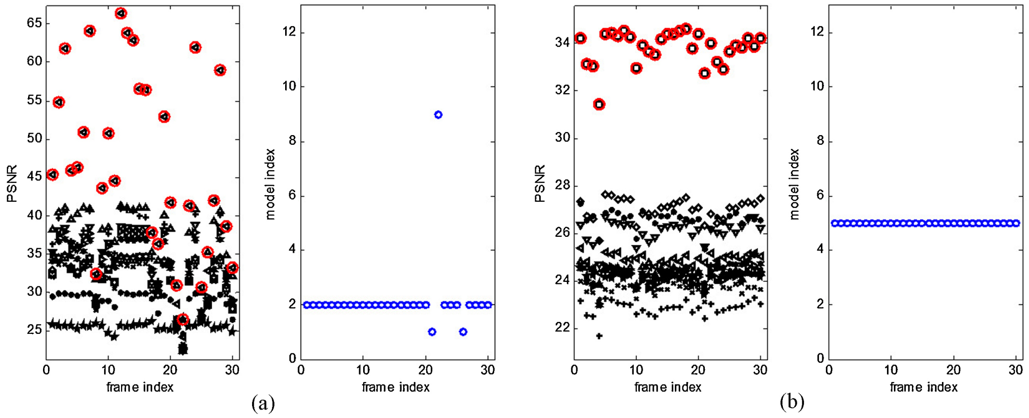


Fig. 8. Visualizing the result of PSNR-based model selection for (a) second and (b) fifth test sequences. In each figure, the left subfigure shows $PSNR(\hat{\mathbf{B}}_{k,t}, \hat{\mathbf{I}}_t)$, wherein the pink circle denotes the one with the maximal PSNR for the frame, while the right one shows the model index corresponding to the maximal PSNR for each testing frame.

2) *Update of Background Models:* The purpose of the model update is to make the background models capable to capture the temporally local changes in the video. In our method, it involves the update of a GMM and M eigenbackgrounds for each pixel location.

A selective update strategy is used in our method for GMMs. That is, when a video frame is input, the first thing is to determine whether each pixel is a clean background pixel; if so, a running average method can be used to update the parameters of its corresponding GMM. Let $\Phi^t_{i,j} = \{\omega^k_{i,j}, \mu^k_{i,j}, \sigma^k_{i,j}\}^{(t)}_{k\in[1,K]}$ be the parameters of the GMM for the (i, j) location when $\mathbf{I}_t$ is given, then the mean and variance of the kth Gaussian component can be updated as follows:

$$\mu^{k,t}_{i,j} = (1-\alpha)\mu^{k,t-1}_{i,j} + \alpha\mathbf{I}_t(i, j) \tag{16}$$

$$(\sigma^{k,t}_{i,j})^2 = (1-\alpha)(\sigma^{k,t-1}_{i,j})^2 + \alpha[\mathbf{I}_t(i, j) - \mu^{k,t}_{i,j}]^2 \tag{17}$$

where α is the learning rate ($\alpha = 0.01$ in this paper). For simplicity, the weight $\omega^k_{i,j}$ dose not change in this process.

On the other hand, the FSM $\mathbf{F}$ will be updated synchronously when $\mathbf{I}_t$ is input. Similarly, when all elements in $\mathbf{F}$ are assigned, the corresponding pixels are assembled to construct a virtual frame. This virtual frame is then used to online update the eigenbackgrounds by CCIPCA [24]. Unlike the offline training case where all elements in $\mathbf{F}$ will be reset to "-1" after a virtual frame is constructed, here only the following elements will be reset to "-1":

$\mathbf{F}(i, j) \leftarrow -1$, only if $(t - \mathbf{F}(i, j)) > \varepsilon$

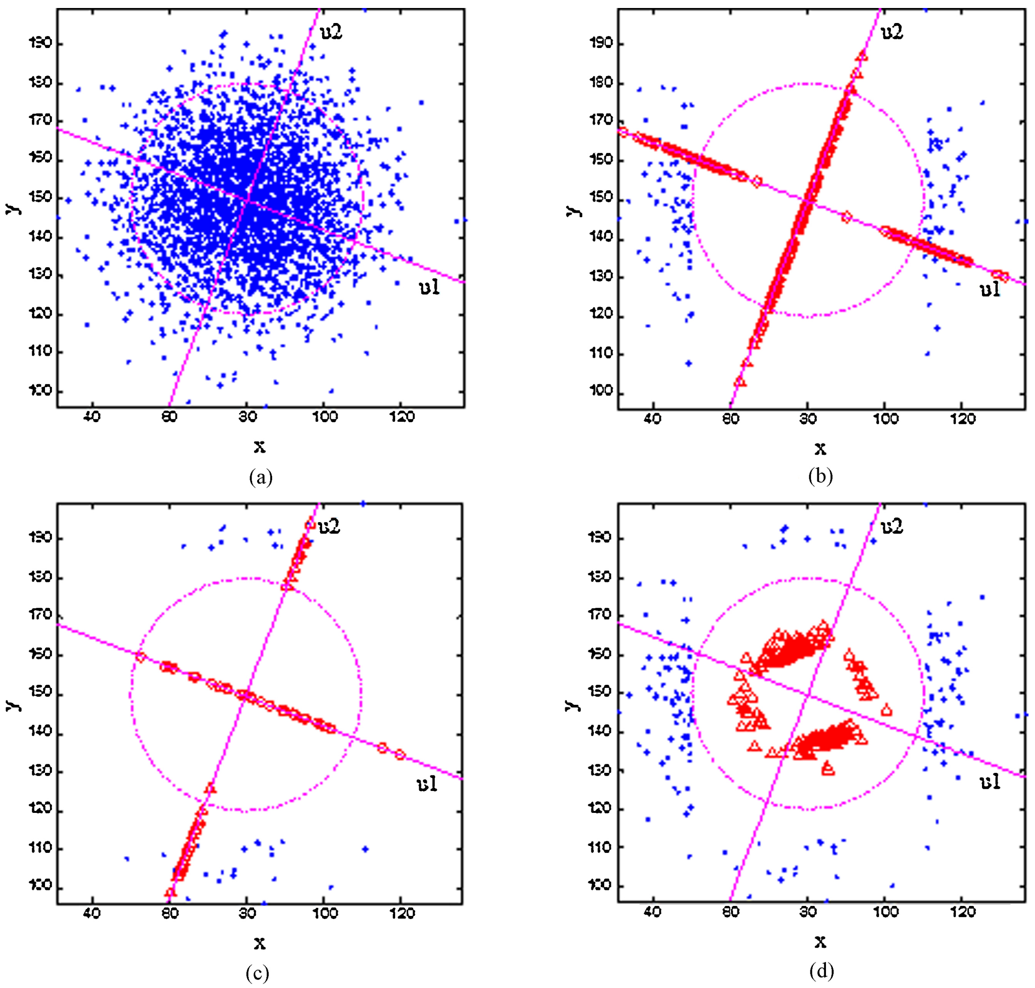
where ε is a predefined decay threshold ($\varepsilon = 1000$ in this paper).

C. Pixel-Level Selective Background Reconstruction

As shown in Fig. 2, when the background reconstruction is performed at once for the whole frame, not all pixels could get the optimal results. Instead, by selecting the best eigenbackground for each block in the reconstruction, the block-based method can effectively reduce the foreground information in the reconstructed background image. However, this is still not the optimal solution since the remarkable mosaic effects inevitably occur in the boundary regions between different blocks.

We note that eigenbackgrounds are in essence the main eigenvectors of the covariance matrix of the training frames; while in the covariance matrix, each element represents the variance of the pixel values at the corresponding position in the training frames or the covariance between two different pixel positions. Therefore, each eigenbackground represents a direction in which the frames differ from the mean image, and the absolute value of each element in an eigenbackground thus represents the scatter degree of the pixel values in that direction. That is, *the eigenbackground element with a smaller absolute value can better represent the "background" property of the corresponding pixel location* since the variance of a background pixel is often less than that of a foreground pixel.

Fig. 9 illustrates an intuitive interpretation of this conjuncture. To simplify the discussion, we suppose the samples are video frames with only two pixels, and then generate some samples using a 2-D Gaussian functions [as shown in Fig. 9(a)]. Samples inside the 2.5σ circle can be treated as background frames, while samples outside the circle are "outliers," where σ is the variance of all the samples. These outliers are used to simulate the video frames in which some foreground objects may occlude the background. Thus the task is to reconstruct the background images for these outliers with two eigenbackgrounds $\mathbf{u}_1 = [-0.9350, 0.3546]^T$ and $\mathbf{u}_2 = [0.3546, 0.9350]^T$. Fig. 9(b) and (c) depicts the results by utilizing the two eigenbackgrounds to reconstruct their background images from x-dimension or y-dimension, respectively. From Fig. 9(b), we can see that the x-dim reconstructed results using $\mathbf{u}_2$ are much better than those using $\mathbf{u}_1$, since most of $\mathbf{u}_2$'s results are inside the circle while most of $\mathbf{u}_1$'s results are not. This is due to the x-dim absolute value of $\mathbf{u}_2$ is much smaller than that of $\mathbf{u}_1$. The similar reason also results in that the y-dim reconstructed results using $\mathbf{u}_1$ are much better than those using $\mathbf{u}_2$, as shown in Fig. 9(c). Thus, to obtain the desirable reconstructed results for all outliers in both x and y dimensions, we should use $\mathbf{u}_2$ to calculate the x-dim pixel values and $\mathbf{u}_1$ to calculate the y-dim pixel values in



(a) (b) (c) (d)

Fig. 9. Intuitive interpretation of selective background reconstruction. (a) Samples distribution, wherein the solid points denote the simulated samples, straight lines represent two eigenbackgrounds, and the circle is the outlier boundary of 2.5σ. (b) and (c) Reconstruction results for the outliers in x-dimension or y-dimension, respectively, wherein the circle marks represent the reconstructed results using $\mathbf{u}_1$, and the triangle marks represent those using $\mathbf{u}_2$. (d) Reconstruction results for all outliers in both x-dimension and y-dimension.

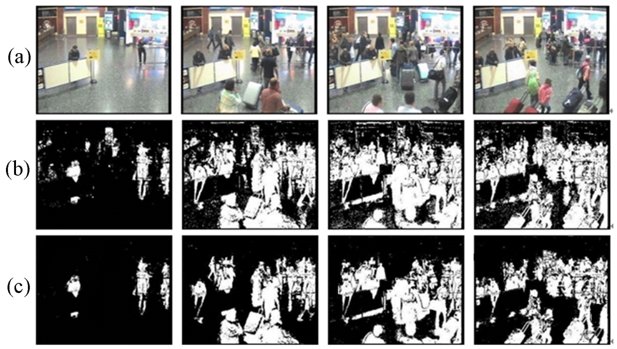


Fig. 10. Some visualized examples of background subtraction with a fixed or adaptive threshold. (a) Input frames. (b) Subtraction results with a fixed threshold. (c) Subtraction results with an adaptive threshold.

the construction. As shown in Fig. 9(d), all the reconstructed results are inside the circle, indicating that their background images can be correctly reconstructed using this method.

Motivated by this example, we propose a *pixel-level selective background reconstruction* algorithm to select an optimal eigenbackground for each pixel. Like the FSM, a matrix with the same dimensionality as the frame, called ESM, is introduced to record the indices of the optimal eigenbackgrounds for the corresponding pixel locations

$$\mathbf{E}(i,j) = \arg\min_k |\mathbf{u}_k(i,j)| \tag{18}$$

where $\mathbf{u}_k(i,j)$ is the (i,j) element of the kth eigenbackground. However, (18) ignores the correlation of neighboring pixels in the selection process. As a result, the reconstructed background might have some singularities that significantly distinguish them from all their neighboring pixels. This will lead to the occurrence of isolated noises in the subtraction result. To avoid this problem, we rewrite (18) by introducing the influence of neighboring pixels, as follows:

$$\mathbf{E}(i,j) = \arg\min_k \left\{ \alpha|\mathbf{u}_k(i,j)| + \frac{1-\alpha}{\|\mathcal{N}_{i,j}\|} \sum_{(i',j')\in\mathcal{N}_{i,j}} |\mathbf{u}_k(i',j')| \right\} \tag{19}$$

where $\mathcal{N}_{i,j}$ is the neighborhood of the (i,j) pixel, $\|\cdot\|$ is the size of a set, α is a weight that reflects the influence of neighborhood pixels ($\alpha = 0.8$ in this paper).

Given the crammed ESM, the background $\mathbf{B}_t$ for $\mathbf{I}_t$ can be reconstructed by

$$\mathbf{B}_t(i, j) = \psi_k(i, j) \tag{20}$$

$$\psi_k = \mathbf{u}_k\mathbf{u}_k^T(\mathbf{I}_t - \bar{\mathbf{I}}) + \bar{\mathbf{I}} \tag{21}$$

where $k = \mathbf{E}(i, j)$ and $\bar{\mathbf{I}}$ is the average image for the training samples.

D. Adaptive Thresholding for Background Subtraction

For background subtraction, a direct method is to threshold the difference between the current frame and its background image. However, it is very difficult to find an appropriate threshold that can be used to derive the best background subtraction results for all frames. Thus, motivated by the idea from [22], an adaptive threshold Δ_t is calculated for each frame $\mathbf{I}_t$ in this paper, namely

$$\Delta_t = \max_k \left\{ \sqrt{\frac{\sum\limits_{r=k}^{R} r^2 \times \mathfrak{b}_t(r)}{\sum_{r=k}^{R} \mathfrak{b}_t(r)} - \left[\frac{\sum_{r=k}^{R} r \times \mathfrak{b}_t(r)}{\sum_{r=k}^{R} \mathfrak{b}_t(r)}\right]^2} \right\}, \tag{22}$$

where

$$\mathfrak{b}_t(r) = \sum_{i=1}^{m}\sum_{j=1}^{n} s_{t,r}(i, j) \tag{23}$$

$$s_{t,r}(i, j) = \begin{cases} 1, |\mathbf{I}_t(i, j) - \mathbf{B}_t(i, j)| = r \\ 0, \text{otherwise} \end{cases} \tag{24}$$

where R is the maximal pixel value (e.g., 255).

Fig. 10 visualizes some examples of background subtraction with a fixed or adaptive threshold. We can see that compared with the results with a fixed threshold, the results with an adaptive threshold have much less noises and false alarms. That is to say, the adaptive thresholding method can effectively improve the performance of background subtraction. It also should be noted that this adaptive thresholding strategy is not only used in the proposed method, but also all other background subtraction methods.

IV. Experiments

Extensive experiments were conducted to verify the effectiveness of the proposed method. Some experimental results have been described in the previous section, mainly on evaluating the performance of different components in our method. Thus this section presents the experiments and results on its overall performance, with several state-of-the-art methods as baselines.

A. Experimental Settings

Two crowded datasets were used in our experiments. One is from the TRECVID-SED corpus [5] and the other is the Road video dataset. The TRECVID-SED dataset contains 145 h indoor surveillance video data collected by the U.K. Home Office from five camera views at a busy airport. We used the video from four cameras in our experiments and left the data from camera 4 out since this view is on the elevator closeup and not a crowded scene. The Road video dataset is 2 hour video corpus collected from a real-world city crossroad surveillance system. In our experiments, this dataset can be used to evaluate the performance of our method in outdoor, busy traffic surveillance videos.

Three eigenbackground algorithms were included in our experiments as the baselines for the proposed method (denoted as **PS-EigenBg**).

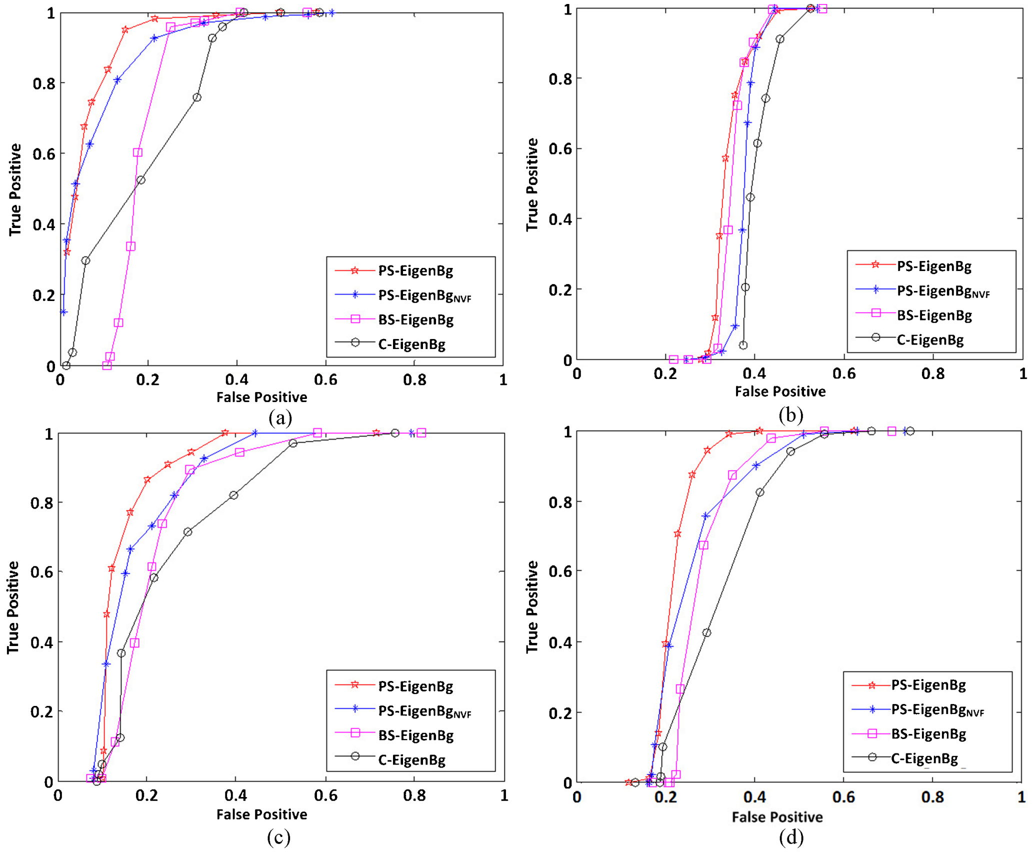
- **C-EigenBg**: The classic eigenbackground method that is trained on the whole frame with Batch PCA [23] and updated incrementally with CCIPCA [24]. It reconstructs the background image using all eigenbackgrounds.
- **BS-EigenBg**: The block-level selective eigenbackground algorithm [17] that divides each original frame into blocks and trains the eigenbackgrounds for each block independently. It selects the best eigenbackground for each block to reconstruct its corresponding background block.
- **PS-EigenBg$_{\text{NVF}}$**: One simplified version of **PS-EigenBg** without utilizing the offline constructed virtual frames for eigenbackground training.

Five noneigenbackground algorithms were also included in our experiments.

- **GMM**: Among different variants of GMMs (e.g., [7]–[10]), the model proposed in [9], which exploits the online EM to update the GMMs and has been implemented in OpenCV, was used as a baseline in our experiments.
- **Bayes**: A Bayesian framework that utilizes the most frequent features to characterize the background appearance and then uses a Bayes decision rule for background and foreground classification (e.g., [12]–[14]).
- **Codebook**: In **Codebook** [18]–[20], sample background values at each pixel are quantized into codebooks, a compressed form of the background model. In our experiments, totally 150 frames were randomly selected to train the codebooks for each video sequence.
- **PBAS**: A pixel-based nonparametric background model [32] that is constructed by *recently observed* pixel and magnitude values. In addition, two controllers with feedback loops are used for both the decision threshold as well as for the learning parameter, making the model adapting to the current video properties.
- **Vibe**: A pixel-based nonparametric background modeling method [33] that models the background with a set of *randomly selected* samples for each pixel. Note that some postprocessing operations such as the inhibition of propagation around internal borders or the distinction between the updating and segmentation masks are also used in **Vibe**.

For a fair comparison, the subtraction results were output by different algorithms without suffering any postprocessing except **Vibe**.

In most of the related literatures (e.g., [11]), the evaluation of background models is usually performed by visualizing some of the subtraction results and subjectively comparing the advantages and disadvantages of various algorithms. However,



(a) (b) (c) (d)

Fig. 11. ROC curves of different eigenbackground methods on the TRECVID-SED corpus. (a) Camera 1. (b) Camera 2. (c) Camera 3. (d) Camera 5.

such a subjective evaluation is argued sometimes as unfair and incomplete, since it cannot quantitatively compare the competing algorithms on a large dataset. Instead, the objective evaluation is performed by comparing the similarity between the subtraction results and the ground truth. Often, the ground truth should be manually labeled in the form of accurate contours of all foreground objects. However, labeling such ground truth on a large video dataset is too labor-intensive. A simplified method is proposed in [4], in which some random frames are extracted from the test data, and the ground truth of these frames are generate by labeling the bounding boxes of foreground objects. When at least 30% pixels in an object's bounding box (note that 25% was used in [4]) are determined as foregrounds by an algorithm, that object is considered as a correct detection. Then, the true positive *TP* is defined as the ratio of the number of correctly detecting objects versus the number of labeled objects in the ground truth, and the negative positive *FP* is measured as the percentage of pixels outside the bounding box that are incorrectly classified as foreground [4]. Thus, under an adaptive threshold or the optimal model parameter, we can quantitatively compare the performance of various methods using the **F-measure**, as follows:

$$\textbf{F-Measure} = \frac{\mathrm{TP}}{\mathrm{TP} + \gamma\mathrm{FN} + (1-\gamma)\mathrm{FP}} \tag{25}$$

where γ is a weight factor ($\gamma = 0.5$ in our experiments), *FN* denotes the false negative that is measured as the percentage of pixels inside the bounding box that are incorrectly classified as background. For eigenbackground methods, moreover, we can also plot the ROC curve for each method under different subtraction thresholds, and then calculate the area under the ROC curve (**AUC**). Different with **F-measure**, **AUC** measures the performance from different subtraction thresholds and then calculates a single overall score. It should be noted that in our experiments, 400 frames sampled from the TRECVID-SED dataset (100 frames *per* view) and 100 frames from the Road video dataset were manually labeled with the bounding boxes of foreground objects.

TABLE I

AUCs OF EIGENBACKGROUND METHODS ON THE TRECVID-SED AND ROAD VIDEO DATASETS

Algorithm	Camera 1	Camera 2	Camera 3	Camera 5	Road
C-EigenBg	0.820	0.599	0.755	0.677	0.851
BS-EigenBg	0.820	0.647	0.788	0.717	0.949
PS-EigenBg$_{\mathrm{NVF}}$	0.923	0.623	0.831	0.740	0.954
PS-EigenBg	**0.944**	**0.658**	**0.855**	**0.781**	**0.958**

B. Experimental Results

1) *Experiment on the TRECVID-SED Data:* The TRECVID-SED corpus is recognized as one of the most difficult indoor crowded video datasets, mainly due to heavy occlusions and significant illumination changes in the clutter scenes. Therefore, this experiment is to verify the performance of the proposed method in this complex dataset.

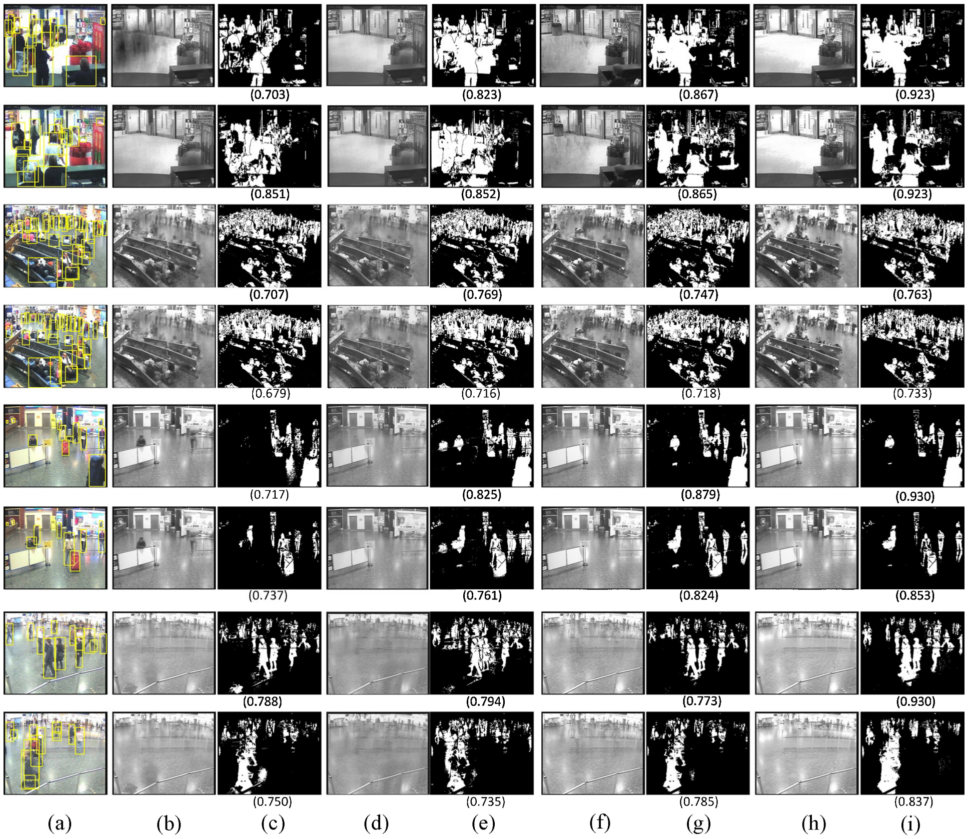


Fig. 12. Visualizing several backgrounds and subtraction results of different eigenbackground methods on the TRECVID-SED corpus. (a) Original frames. (b) and (c) **C-EigenBg**. (d) and (e) **BS-EigenBg**. (f) and (g) **PS-EigenBg$_{\text{NVF}}$**. (h) and (i) **PS-EigenBg**. Note that in this figure (and the following several figures), the decimal below each subtraction result is the corresponding **F-Measure**.

We first compare the performance of different eigenbackground methods. Fig. 11 depicts their ROC curves, and the corresponding **AUC**s can be found in Table I. Moreover, Table II also lists their **F-measure**s under the adaptive subtraction threshold. Not surprisingly, **PS-EigenBg** outperforms all other eigenbackground methods remarkably, and even its simplified version **PS-EigenBg$_{\text{NVF}}$** can achieve higher **AUC**s and **F-measure**s than **BS-EigenBg** and **C-EigenBg** in most cases. This indicates that the proposed three selective eigenbackground mechanisms, in particular selective training, can effectively improve the performance of eigenbackground methods. We also notice that **AUC**s of different eigenbackground methods in Camera 2 are much lower that those in other cameras. This is because heavy occlusions exist almost in all training frames of this camera, and even many foreground objects remain nearly stationary in some areas (e.g., the seating area). This makes the trained eigenbackgrounds full of foregrounds. In particular, we find that **AUC** of **PS-EigenBg$_{\text{NVF}}$** in this camera is even lower than that of **BS-EigenBg**, despite their **F-Measure**s under the adaptive threshold are comparable. This means that in a crowded scene with heavy occlusions, we should explore more sophisticated methods to take into account the correlation of neighboring pixels in the reconstruction [e.g., using a scene-adaptive neighborhood in (18)].

Fig. 12 also visualizes some examples of their backgrounds and subtraction results on this dataset. We first note that in Camera 2, the quality of all these reconstructed backgrounds is totally low. This is consistent with the **AUC** and **F-measure** results listed in Tables I and II. Without considering results in Camera 2, it is obvious that the backgrounds of **PS-EigenBg** are much cleaner that those of the other methods, consequently, leading to higher **F-measure**s of its subtraction results. For example, some obvious ghost can be found in Fig. 12(b), (d), and (f1), (f2). Although worse than **PS-EigenBg**, the reconstructed backgrounds of **PS-EigenBg$_{\text{NVF}}$** are still slightly better than **BS-EigenBg** and much cleaner than **C-EigenBg**. Accordingly, most of its subtraction results are slightly better than those with **BS-EigenBg** [e.g., (e1) versus (g1), (e2) versus (g2), (e5) versus (g5)], while some results are worse [e.g., (e7) versus (g7)].

TABLE II

F-MEASURES OF DIFFERENT METHODS ON THE TRECVID-SED AND ROAD VIDEO DATASETS USING THE ADAPTIVE SUBTRACTION THRESHOLD OR THE OPTIMAL PARAMETER

	Camera 1	Camera 2	Camera 3	Camera 5	Road
GMM	0.417	0.394	0.505	0.659	0.858
CodeBook	0.838	0.590	0.715	0.685	0.648
Bayes	0.638	0.166	0.492	0.637	0.669
PBAS	0.636	0.116	0.493	0.606	0.690
ViBe	0.790	0.282	0.590	0.599	0.792
C-EigenBg	0.769	0.681	0.711	0.686	0.802
BS-EigenBg	0.841	0.717	0.728	0.744	0.932
PS-EigenBg$_{\text{NVF}}$	0.852	0.715	0.779	0.733	0.943
PS-EigenBg	**0.898**	**0.719**	**0.829**	**0.808**	**0.953**

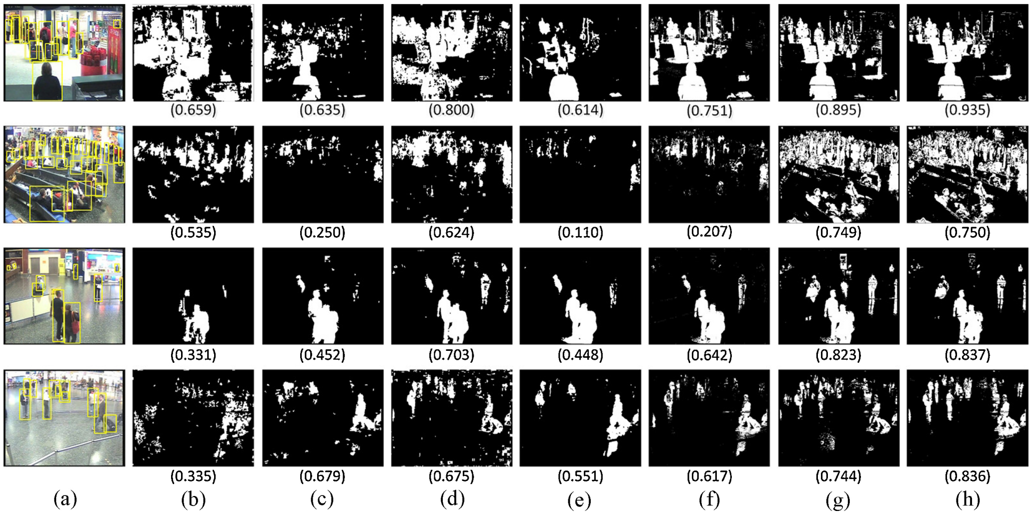


Fig. 13. Visualizing several subtraction results of noneigenbackground methods on the TRECVID-SED corpus, together with **PS-EigenBg** and **PS-EigenBg$_{\mathrm{NVF}}$**. (a) Original frames, (b) **GMM**, (c) **Bayes**, (d) **Codebook**, (e) **PBAS**, (f) **ViBe**, (g) **PS-EigenBg$_{\mathrm{NVF}}$**, and (h) **PS-EigenBg**.

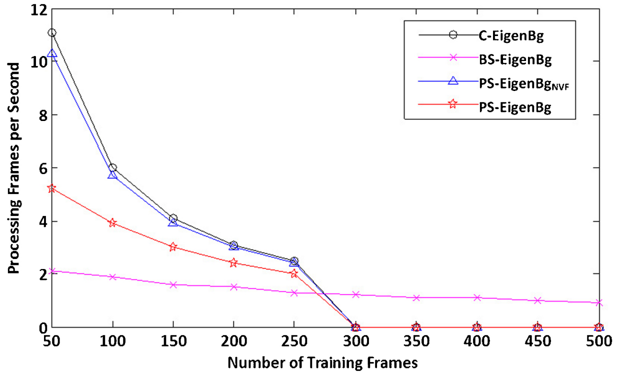


Fig. 14. Processing frames per second of various eigenbackground methods given different numbers of training frames.

We also compare the performance of **PS-EigenBg** and **PS-EigenBg$_{\mathrm{NVF}}$** with five noneigenbackground methods, including **GMM** [9], **Bayes** [12], **Codebook** [18], **PBAS** [32], and **ViBe** [33]. We can find their **F-measure**s in Table II under the corresponding optimal parameters, and some subtraction results in Fig. 13. Among them, **Codebook** shows a superior performance on this dataset (in particular on Camera 2), mainly due to the introduction of offline codebook training; **Bayes** and **PBAS** have very poor performance on all cameras in this dataset, exhibiting their inability to deal with the background modeling task in complex crowded scenes; in contrast, **Vibe** shows the medium performance on Cameras 1 and 3, while **GMM** also achieves the medium performance on Cameras 2 and 5. Totally speaking, although seems slightly unfair since **PS-EigenBg** is built on one of them (i.e., **GMM**), this comparison can validate a definite conclusion that these noneigenbackground methods do not work well on the complex crowded scenes.

In addition, Fig. 14 shows the processing frames per second **Fps** for different eigenbackground methods when given different numbers of training frames, on a four-core PC with 3.1 GHz CPU and 3 GB RAM. From the figure, we can see that the **Fps** of **C-EigenBg**, **PS-EigenBg$_{\mathrm{NVF}}$**, and **PS-EigenBg** all decrease to zero when the number of training frames is more than 300. In this case, the calculation is out of memory since they all perform eigendecomposition (i.e., Batch PCA) on the whole frame. When given less than 300 training frames, the training efficiency of **PS-EigenBg** is slightly lower than **C-EigenBg** and **PS-EigenBg$_{\mathrm{NVF}}$**, mainly due to the additional computation of virtual frame construction. We also notice that **BS-EigenBg** can be trained efficiently even with 500 training frames. This is because partitioning a whole frame into several blocks can effectively reduce the computational complexity during the computation of co-covariance matrix and eigendecomposition. Moreover, if these blocks are processed in parallel, the training efficiency of **BS-EigenBg** should go up by several times. The experimental results show that our algorithm does not increase the computational complexity significantly in the training stage.

For subtraction, the average processing efficiency of **PS-EigenBg** and **PS-EigenBg$_{NVF}$** is 3–4 **Fps**, slightly lower than that of **C-EigenBg** (4–5 **Fps**). The most time-consuming part of these eigenbackground methods is the online computation of CCIPCA; in contrast, online virtual frame construction and pixel-level optimal eigenbackground selection only takes less than 10% computation of **PS-EigenBg**. If without performing online update with CCIPCA, the average processing efficiency of **PS-EigenBg** will be up to 9–10 **Fps**. Thus, in order to deal with the real-time applications, some optimization techniques such as multithread or multiprocess programming should be used.

2) *Experiment on the Road Data:* Different with the TRECVID-SED corpus, the Road video dataset represents another typical scene from an outdoor, high-traffic crossroad surveillance system. Thus, this experiment is to evaluate the performance of the proposed method in this outdoor scene with crowded vehicles.

AUCs and **F-measure**s of different methods on this dataset can be found in Tables I and II, respectively. For

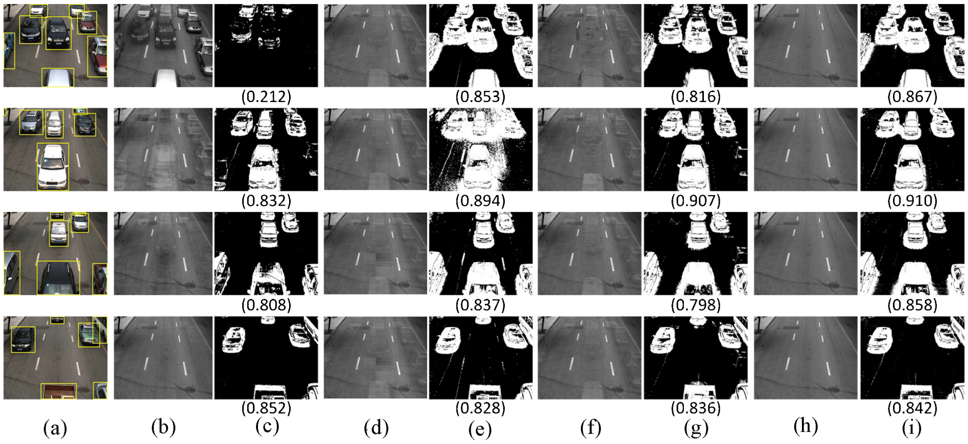


Fig. 15. Visualizing several backgrounds and subtraction results of different eigenbackground methods on the Road video corpus. (a) Original frames. (b) and (c) **C-EigenBg**. (d) and (e) **BS-EigenBg**. (f) and (g) **PS-EigenBg$_{\text{NVF}}$**. (h) and (i) **PS-EigenBg**.

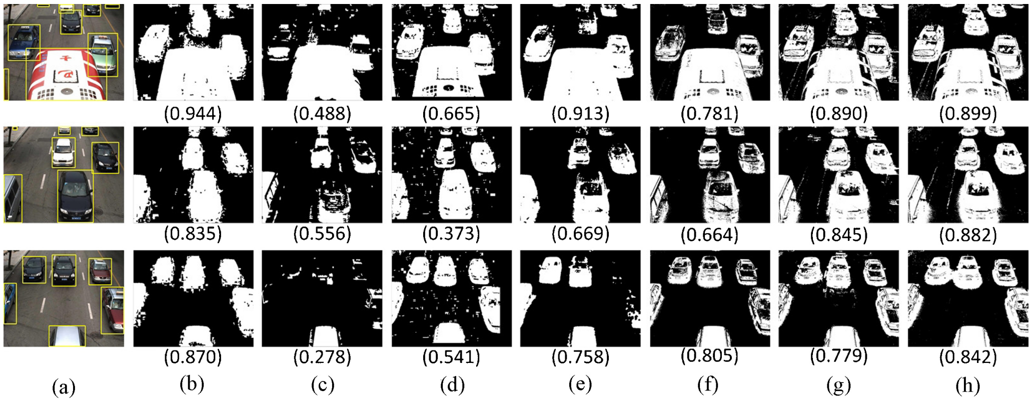


Fig. 17. Visualizing several subtraction results of noneigenbackground methods, together with **PS-EigenBg** and **PS-EigenBg$_{\text{NVF}}$** on the Road video corpus. (a) Original frames. (b) **GMM**. (c) **Bayes**. (d) **Codebook**. (e) **PBAS**. (f) **ViBe**. (g) **PS-EigenBg$_{\text{NVF}}$**. (h) **PS-EigenBg.**

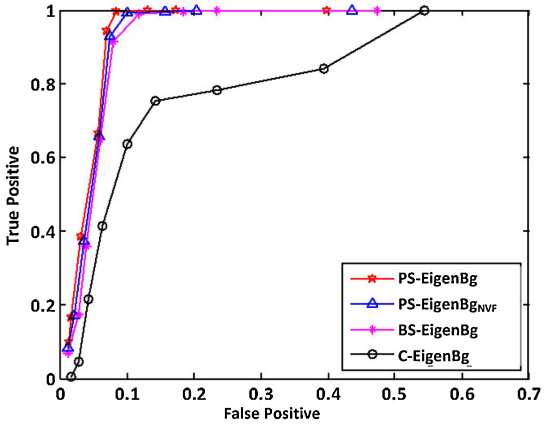


Fig. 16. ROC curves of different eigenbackground methods on the Road video dataset.

eigenbackground methods, Fig. 15 shows several examples of backgrounds and subtraction results, and Fig. 16 illustrates the corresponding ROC curves. From these results, we can see that **PS-EigenBg**, **PS-EigenBg$_{\text{NVF}}$**, and **BS-EigenBg** all can achieve very good performance. In addition, the performance gap among them becomes less insignificant since more clean frames can be found in this dataset. In Fig. 15, despite there still exist some nonsalient ghost effects, the reconstructed backgrounds of **PS-EigenBg$_{\text{NVF}}$** and **BS-EigenBg**, even some backgrounds of **C-EigenBg**, are visually very close to those of **PS-EigenBg**.

Fig. 17 also shows several subtraction results of noneigenbackground methods. We can see that **GMM** shows very good performance on this dataset, with **F-measure** of 0.858; **Vibe** also exhibits a comparable performance, with **F-measure** of 0.792. In some cases [e.g., Fig. 17 (b1) and (b3)], the background subtraction results of **GMM** are even better than those of **PS-EigenBg**. This is because there are some holes near the windows of the cars in the detection results of **PS-EigenBg$_{\text{NVF}}$** or **PS-EigenBg** (also by **C-EigenBg** or **BS-EigenBg**). Thus, we should utilize a postprocessing step to address this "window hole" problem for eigenbackground methods.

V. Conclusion

This paper proposed a selective eigenbackground modeling and subtraction method that could keep robust in crowded scenes. Three selectivity mechanisms were integrated in our methods, including selective training, selective model initialization, and pixel-level selective reconstruction. Using these mechanisms, our method could significantly increase the purity of the trained eigenbackgrounds and could obtain an improved quality of the reconstructed background image, consequently, lead to a better subtraction performance in crowded scenes. Extensive experiments on the TRECVID-SED and Road video datasets showed that our method outperformed several eigenbackground and noneigenbackground methods remarkably. Nevertheless, there are still much room to improving the proposed method on the very crowded scenes such as Camera 2 of the TRECVID-SED dataset. Meanwhile, the residual shadows and window holes should also be addressed in the future work.

References

[1] M. Y. Weng, G. Huang, and X. Y. Da, "A new interframe difference algorithm for moving target detection," in *Proc. 3rd Int. Congr. Image Signal Process.*, 2010, pp. 285–289.

[2] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Detecting moving objects, ghosts and shados in video streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[3] J. Ma and S. T. Li, "Moving target detection based on background modeling by multi-level median filter," in *Proc. World Congr. Intell. Control Autom.*, 2006, pp. 9974–9978.

[4] B. Klare and S. Sarkar, "Background subtraction in varying illuminations using an ensemble based on an enlarged feature set," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun. 2009, pp. 66–73.

[5] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and TRECVID," in *Proc. 8th ACM Int. Workshop Multimedia Inf. Retrieval*, 2006, pp. 321–330.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.

[8] R. H. Evangelio, T. Senst, and T. Sikora, "Detection of static objects for the task of video surveillance," in *Proc. IEEE Workshop Appl. Comput. Vis.*, Jan. 2011, pp. 534–540.

[9] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. 2nd Eur. Workshop Adv. Video-Based Surveillance Syst.*, 2002, pp. 135–144.

[10] F.-C. Cheng, S.-C. Huang, and S.-J. Ruan, "Advanced background subtraction approach using Laplacian distribution model," in *Proc. Int. Conf. Multimedia Expo.*, 2010, pp. 754–759.

[11] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1305–1312.

[12] L. Y. Li, W. M. Huang, I. Y.-H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proc. ACM Multimedia*, 2003, pp. 2–10.

[13] N.M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[14] O. Tuzel, F. Porikli, and P. Meer, "A Bayesian approach to background modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 3, Jun. 2005, pp. 58–61.

[15] A. Elgammal, D. Hanvood, and L. S. Davis, "Nonparametric model for background subtraction," in *Proc. Eur. Conf. Comput. Vis.*, 2000, pp. 751–767.

[16] B. Zhang, Y. Gao, S. Zhao, and B. Zhong, "Kernel similarity modeling of texture pattern flow for motion detection in complex background," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 29–38, Jan. 2011.

[17] Z. P. Hu, G. N. Ye, G. C. Jia, X. B. Chen, Q. Hu, K. H. Jiang, Y. W. Wang, L. Qing, Y. H. Tian, X. H. Wu, and W. Gao, "PKU@TRECVID2009: Single-actor and pair-activity event detection in surveillance video," in *Proc. TRECVID Workshop*, 2009. http://www-nlpir.nist.gov/projects/tvpubs/tv9.papers/pku-idm.pdf

[18] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground background segmentation using codebook model," *Real-Time Imaging*, vol. 11, no. 3, pp. 172–185, 2005.

[19] M. H. Sigari and M. Fathy, "Real-time background modeling/subtraction using two-layer codebook model," in *Proc. Int. Conf. Eng. Comput. Sci.*, 2008, pp. 717–720.

[20] A. Pal, G. Schaefer, and M. E. Celebi, "Robust codebook-based video background subtraction," in *Proc. Int. Conf. Acoust. Speech Signal Process*, Mar. 2010, pp. 1146–1149.

[21] Y. Yuan, Y. W. Pang, J. Pan, and X. L. Li, "Scene segmentation based on IPCA for visual surveillance," *Neurocomputing*, vol. 72, nos. 10–12, pp. 2450–2454, 2009.

[22] P. Rosin, "Thresholding for change detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jan. 1998, pp. 274–279.

[23] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Rev.: Comput. Statist.*, vol. 2, no. 4, pp. 433–459, 2010.

[24] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.

[25] Y.-F. Guo, X. D. Lin, Z. Teng, X. Y. Xue, and J. P. Fan, "A covariance-free iterative algorithm for distributed principal component analysis on vertically partitioned data," *Pattern Recognit.*, vol. 45, no. 3, pp. 1211–1219, 2012.

[26] E. Oja and J. Karhunen, "On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix," *J. Math. Anal. Appl.*, vol. 106, no. 1, pp. 69–84, 1985.

[27] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2. Aug. 2003, pp. 1494–1500.

[28] Y. Li, "On incremental and robust subspace learning," *Pattern Recognit.*, vol. 37, no. 7, pp. 1509–1518, 2004.

[29] D. Ross, J. Lim, and M. H. Yang, "Adaptive probabilistic visual tracking with incremental subspace update," in *Proc. 8th Eur. Conf. Comput. Vis.*, 2004, pp. 470–482.

[30] Z. P. Hu, Y. W. Wang, Y. H. Tian, and T. J. Huang, "Selective eigenbackgrounds method for background subtraction in crowded scenes," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3277–3280.

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

[32] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Proc. IEEE Comput. Vis. Pattern Recognit. Workshop Change Detect.*, Jun. 2012, pp. 38–43.

[33] M. Van Droogenbroeck and O. Paquot, "Background subtraction: Experiments and improvements for ViBe," in *Proc. IEEE CVPR Workshop Change Detect.*, Jun. 2012, pp. 32–37.

Yonghong Tian (M'05–SM'10) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently an Associate Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He is the author or co-author of more than 80 technical articles in refereed journals and conferences. His current 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 task during 2010–2011; the Top Performer in the TRECVID retrospective SED task during 2009–2012; and the Winner of the WikipediaMM Task in ImageCLEF 2008.

Yaowei Wang received the M.Sc. degree from the Department of Computer Science, Heibei University of Technology, Tianjin, China, in 2000, and the Ph.D. degree from the Graduate School of the Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently a Lecturer in the School of Information and Electronics, Beijing Institute of Technology, Beijing, China. He is the author or coauthor of more than 20 technical articles in refereed journals and conferences. His current research interests include multimedia analysis and surveillance video analysis.

Zhipeng Hu received the the M.Sc. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011.

From September 2009 to July 2011, he was a Visiting Student in the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, where he is currently in one senior global Research and Development Team in electromagnetic capability. He is currently involved in research on multicore raid technology to improve scalability of backend storage systems.

Tiejun Huang (M'01–SM'12) received the Ph.D. degree from the Institute for Image Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1998.

He was a Post-Doctoral Research Fellow from 1999 to 2001 and a Research Faculty Member with the Institute of Computing Technology, Chinese Academy of Sciences. He was also the Associated Director from 2001 to 2003 and the Director from 2003 to 2006 of the Research Center for Digital Media, Graduate School of the Chinese Academy of Sciences. He is currently a Professor and the Director of the Institute for Digital Media Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. Since 2002, he has also been the Secretary General of the Audio and Video coding Standards Working Group of China. He has published two books, and more than 120 technical papers in refereed journals and conferences. His research interests include video coding, image understanding, digital right management, and digital library.

Dr. Huang is the member of the Board of Director Digital Media Project, Advisory Board of the IEEE Computing Now and Editorial Board of the Springer *3-D Research Journal*.