Complex Background Subtraction by Pursuing Dynamic Spatio-Temporal Models

Liang Lin, Yuanlu Xu, Xiaodan Liang, and Jianhuang Lai

***Abstract*— Although it has been widely discussed in video surveillance, background subtraction is still an open problem in the context of complex scenarios, e.g., dynamic backgrounds, illumination variations, and indistinct foreground objects. To address these challenges, we propose an effective background subtraction method by learning and maintaining an array of dynamic texture models within the spatio-temporal representations. At any location of the scene, we extract a sequence of regular video bricks, i.e., video volumes spanning over both spatial and temporal domain. The background modeling is thus posed as pursuing subspaces within the video bricks while adapting the scene variations. For each sequence of video bricks, we pursue the subspace by employing the auto regressive moving average model that jointly characterizes the appearance consistency and temporal coherence of the observations. During online processing, we incrementally update the subspaces to cope with disturbances from foreground objects and scene changes. In the experiments, we validate the proposed method in several complex scenarios, and show superior performances over other state-of-the-art approaches of background subtraction. The empirical studies of parameter setting and component analysis are presented as well.**

***Index Terms*— Background modeling, visual surveillance, spatio-temporal representation.**

I. Introduction

BACKGROUND subtraction (also referred as foreground extraction) has been extensively studied in decades [1]–[6], yet it still remains open in real surveillance applications due to the following challenges:

- Dynamic backgrounds. A scene environment is not always static but sometimes highly dynamic, *e.g.*, rippling water, heavy rain and camera jitter.

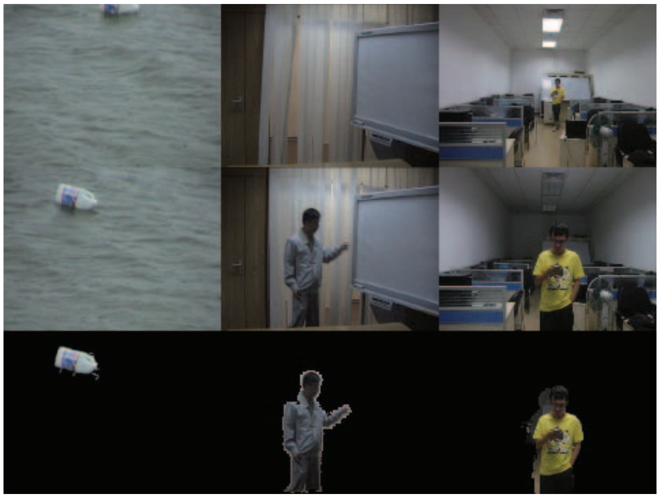


Fig. 1. Some challenging scenarios for foreground object extraction are handled by our approach: (i) a floating bottle with randomly dynamic water (in the left column), (ii) waving curtains around a person (in the middle column), and (iii) sudden light changing (in the right column).

- Lighting and illumination variations, particularly with sudden changes.
- Indistinct foreground objects having similar appearances with surrounding backgrounds.

In this paper, we address the above mentioned difficulties by building the background models with the online pursuit of spatio-temporal models. Some results generated by our system for the challenging scenarios are exhibited in Fig. 1. Prior to unfolding the proposed approach, we first review the existing works in literature.

A. Related Work

Due to their pervasiveness in various applications, there is no unique categorization on the existing works of background subtraction. Here we introduce the related methods basically according to their representations, to distinguish with our approach.

The **pixel-processing** approaches modeled observed scenes as a set of independent pixel processes, and they were widely applied in video surveillance applications [6], [7] . In these methods [1], [2], [8], [9], each pixel in the scene can be described by different parametric distributions (*e.g.* Gaussian Mixture Models) to temporally adapt to the environment changes. The parametric models, however, were not always compatible with real complex data, as they were defined based upon some underlying assumptions. To overcome this problem, some other non-parametric estimations [10]–[13]

Manuscript received October 15, 2013; revised January 25, 2014 and April 4, 2014; accepted May 17, 2014. Date of publication May 23, 2014; date of current version June 16, 2014. This work was supported in part by the Hi-Tech Research and Development (863) Program of China under Grant 2012AA011504, in part by the National Natural Science Foundation of China under Grant 61173082 and Grant 61173084, in part by the Guangdong Science and Technology Program under Grant 2012B031500006, in part by the Guangdong Natural Science Foundation under Grant S2013050014548, in part by the Special Project on Integration of Industry, Education and Research of Guangdong Province under Grant 2012B091000101, and in part by the Fundamental Research Funds for the Central Universities under Grant 13lgjc26. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Pierre-Marc Jodoin.

L. Lin is with the Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, Sun Yat-sen University, China, School of Advanced Computing, Sun Yat-sen University, Guangzhou 510006, China, and also with Sun Yat-sen University-Carnegie Mellon University Shunde International Joint Research Institute, Shunde 510006, China (e-mail: linliang@ieee.org).

Y. Xu, X. Liang, and J. Lai are with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China (e-mail: merayxu@gmail.com; xdliang328@gmail.com; stsljh@mail.sysu.edu.cn).

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

Digital Object Identifier 10.1109/TIP.2014.2326776

were proposed, and effectively improved the robustness. For example, Barnich et al. [13] presented a sample-based classification model that maintained a fixed number of samples for each pixel and classified a new observation as background when it matched with a predefined number of samples. Liao et al. [14] recently employed the kernel density estimation (KDE) technique to capture pixel-level variations. Some distinct scene variations, *i.e.* illumination changes and shadows, can be explicitly alleviated by introducing the extra estimations [15]. Guyon et al. [16] proposed to utilize the low rank matrix decomposition for background modeling, where the foreground objects constituted the correlated sparse outliers. Despite acknowledged successes, this category of approaches may have limitations on complex scenarios, as the pixel-wise representations overlooked the spatial correlations between pixels.

The **region-based** methods built background models by taking advantages of inter-pixel relations, demonstrating impressive results on handling dynamic scenes. A batch of diverse approaches were proposed to model spatial structures of scenes, such as joint distributions of neighboring pixels [11], [17], block-wise classifiers [18], structured adjacency graphs [19], auto-regression models [20], [21], random fields [22], and multi-layer models [23] etc. And a number of fast learning algorithms were discussed to maintain their models online, accounting for environment variations or any structural changes. For example, Monnet et al. [20] trained and updated the region-based model by the generative subspace learning. Cheng et al. [19] employed the generalized 1-SVM algorithm for model learning and foreground prediction. In general, methods in this category separated the spatial and temporal information, and their performances were somewhat limited in some highly dynamic scenarios, *e.g.* heavy rains or sudden illumination changes.

The third category modeled scene backgrounds by exploiting both spatial and temporal information. Mahadevan et al. [24] proposed to separate foreground objects from surroundings by judging the distinguished video patches, which contained different motions and appearances compared with the majority of the whole scene. Zhao et al. [25] addressed the outdoor night background modeling by performing subspace learning within video patches. Spatio-temporal representations were also extensively discussed in other vision tasks such as action recognition [26] and trajectory parsing [27]. These methods motivated us to build models upon the spatio-temporal representations, *i.e.* video bricks.

In addition, several **saliency-based** approaches provided alternative ways based on spatio-temporal saliency estimations [24], [28], [29]. The moving objects can be extracted according to their salient appearances and/or motions against the scene backgrounds. For example, Wixson et al. [28] detected the salient objects according to their consistent moving directions over time. Kim et al. [30] used a discriminant center-surround hypothesis to extract foreground objects around their surroundings.

Along with the above mentioned background models, a number of reliable image features were utilized to better handle the background noise [31]. Exemplars included the Local Binary Pattern (LBP) features [32]–[34] and color texture histograms [35]. The LBP operators described each pixel by the relative graylevels of its neighboring pixels, and their effectiveness has been demonstrated in several vision tasks such as face recognition and object detection [32], [36], [37]. The Center-Symmetric LBP was proposed in [34] to further improve the computational efficiency. Tan and Triggs [33] extended LBP to LTP (Local Ternary Pattern) by thresholding the graylevel differences with a small value, to enhance the effectiveness on flat image regions.

B. Overview

In this work, we propose to learn and maintain the dynamic models within spatio-temporal video patches (*i.e.* video bricks), accounting for real challenges in surveillance scenarios [7]. The algorithm can process $15 \sim 20$ frames per second in the resolution 352×288 (pixels) on average. We briefly overview the proposed framework of background modeling in the following aspects.

1) Spatio-Temporal Representations: We represent the observed scene by video bricks, *i.e.* video volumes spanning over both spatial and temporal domain, in order to jointly model spatial and temporal information. Specifically, at every location of the scene, a sequence of video bricks are extracted as the observations, within which we can learn and update the background models. Moreover, to compactly encode the video bricks against illumination variations, we design a brick-based descriptor, namely Center Symmetric Spatio-Temporal Local Ternary Pattern (CS-STLTP), which is inspired by the 2D scale invariant local pattern operator proposed in [14]. Its effectiveness is also validated in the experiments.

2) Pursuing Dynamic Subspaces: We treat each sequence of video bricks at a certain location as a consecutive signal, and generate the subspace within these video bricks. The linear dynamic system (*i.e.* Auto Regressive Moving Average, ARMA model [38]) is adopted to characterize the spatio-temporal statistics of the subspace. Specifically, given the observed video bricks, we express them by a data matrix, in which each column contains the feature of a video brick. The basis vectors (*i.e.* eigenvectors) of the matrix can be then estimated analytically, representing the appearance parameters of the subspace, and the parameters of dynamical variations are further computed based on the fixed appearance parameters. It is worth mentioning that our background model jointly captures the information of appearance and motion as the data (*i.e.* features of the video bricks) are extracted over both spatial and temporal domains.

3) Maintaining Dynamic Subspaces Online: Given the newly appearing video bricks with our model, moving foreground objects are segmented by estimating the residuals within the related subspaces of the scene, while the background models are maintained simultaneously to account for scene changes. The raising problem is to update parameters of the subspaces incrementally against disturbance from foreground objects and background noise. The new observation may include noise pixels (*i.e.* outliers), resulting in degeneration of model updating [20], [25]. Furthermore, one

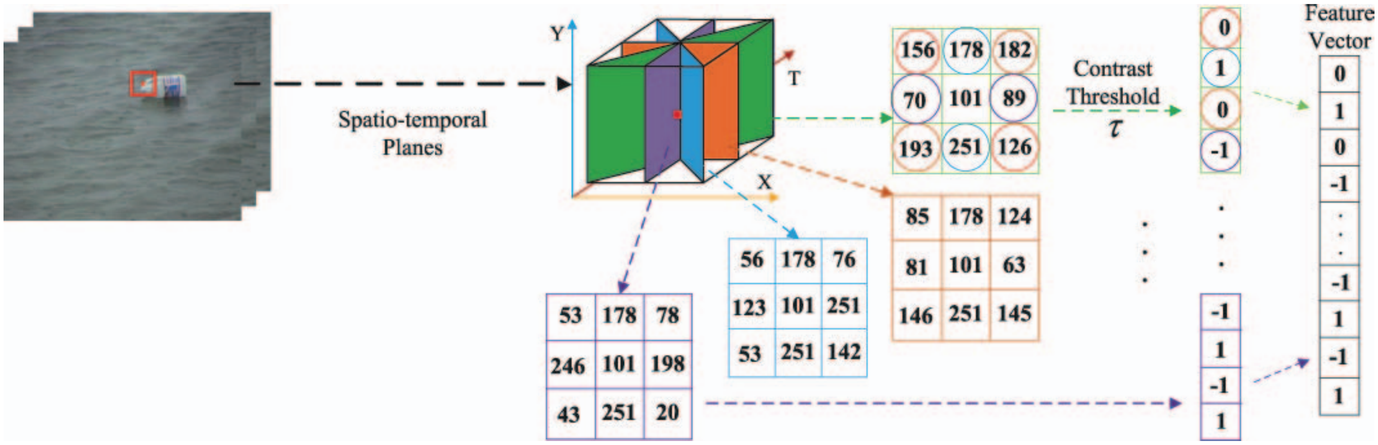


Fig. 2. An example of computing the CS-STLTP feature. For one pixel in the video brick, we construct four spatio-temporal planes. The center-symmetric local ternary patterns for each plane is calculated, which compares the intensities in a center-symmetric direction with a contrasting threshold τ. The CS-STLTP feature is concatenated by the vectors of the four planes.

video brick could be partially occluded by foreground objects in our representation, *i.e.* only some of pixels in the brick are true positives. To overcome this problem, we present a novel approach to compensate observations (*i.e.* the observed video bricks) by generating data from the current models. Specifically, we replace the pixels labeled as non-background by the generated pixels to synthesize the new observations. The algorithm for online model updating includes two steps: (i) update appearance parameters using the incremental subspace learning technique, and (ii) update dynamical variation parameters by analytically solving the linear reconstruction. The experiments show that the proposed method effectively improves the robustness during the online processing.

The remainder of this paper is arranged as follows. We first present the model representation in Section II, and then discuss the initial learning, foreground segmentation and online updating mechanism in Section III, respectively. The experiments and comparisons are demonstrated in Section IV and finally comes the conclusion in Section V with a summary.

II. Dynamic Spatio-Temporal Model

In this section, we introduce the background of our model, and then discuss the video brick representation and our model definition, respectively.

A. Background

In general, a complex surveillance background may include diverse appearances that sometimes move and change dynamically and randomly over time flying [39]. There is a branch of works on time-varying texture modeling [40]–[42] in computer vision. They often treated the scene as a whole, and pursued a global subspace by utilizing the linear dynamic system (LDS). These models worked well on some natural scenes mostly including a few homogeneous textures, as the LDS characterizes the subspace with a set of linearly combined components. However, under real surveillance challenges, it could be intractable to pursue the global subspace. In this work, we represent the observed scene by an array of small and independent subspaces, each of which is defined by the linear system, so that our model is able to handle better challenging scene variations. Our background model can be viewed as a mixed compositional model consisting of the linear subspaces. In particular, we conduct the background subtraction with our model based on the following observations.

Assumption 1: The local scene variants (*i.e.* appearance and motion changing over time) can be captured by the low-dimensional subspace.

Assumption 2: It is feasible to separate foreground moving objects from the scene background by fully exploiting spatio-temporal statistics.

B. Spatio-Temporal Video Brick

Given the surveillance video of one scene, we first decompose it with a batch of small brick-like volumes. We consider the video brick of small size (*e.g.*, $4 \times 4 \times 5$ pixels) includes relative simple content, which can be thus generated by few bases (components). And the brick volume integrates both spatial and temporal information, that we can better capture complex appearance and motion variations compared with the traditional image patch representations.

We divide each frame I_i , $(i = 1, 2, \ldots, n)$ into a set of image patches with the width w and height h. A number t of patches at the same location across the frames are combined together to form a brick. In this way, we extract a sequence of video bricks $V = \{v_1, v_2, \ldots, v_n\}$ at every location for the scene.

Moreover, we design a novel descriptor to describe the video brick instead of using RGB values. For any video brick v_i, we first apply the CS-STLTP operator on each pixel, and pool all the feature values into a histogram. For a pixel x_c, we construct a few 2D spatio-temporal planes centered at it, and compute the local ternary patterns (LTP) operator [33] on each plane. The CS-STLTP then encodes x_c by combining the LTP operators of all planes. Note that the way of splitting spatio-temporal planes little affects the operator's performance. To simplify the implementation, we make the planes parallel to the Y axis, as Fig. 2 shown.

We index the neighborhood pixels of x by $\{0, \dots, M\}$, the operator response of the j-th plane can be then calculated as:

$$F^j(x) = \biguplus_{m=0}^{\frac{M}{2}-1} s_\tau(p_m, p_{m+\frac{M}{2}}), \tag{1}$$

where pixel k and $k + M/2$ are two symmetric neighbors of pixel x_c. p_k and $p_{k+\frac{M}{2}}$ are the graylevels of the two pixels, respectively. The sign $\biguplus$ indicates stretching elements into a vector. The function s_τ is defined as follows:

$$s_\tau(p_m, p_{m+\frac{M}{2}}) = \begin{cases} 1, & \text{if } p_m > (1+\tau)p_{m+\frac{M}{2}}, \\ \text{-}1, & \text{if } p_m < (1-\tau)p_{m+\frac{M}{2}}, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where τ is a constant threshold for the comparing range.

Suppose that we take $M = 8$ neighborhood pixels for computing the operator in each spatio-temporal plane, and the number of planes is 4. The resulting CS-STLTP vector contains $M/2 \times 4 = 16$ bins. Fig. 2 illustrates an example of computing the CS-STLTP operator, where we apply the operator for one pixel on 4 spatial-temporal planes displayed with different colors (*e.g.*, green, blue, purple and orange).

Then we build a histogram for each video brick by accumulating the CS-STLTP responses of all pixels. This definition was previously proposed by Guo et al [36].

$$H(k) = \Sigma_{x \in v_i} \Sigma_{j=1}^{4} \mathbf{1}(F^{\mathbf{j}}(\mathbf{x}), \mathbf{k}), \quad \mathbf{k} \in [\mathbf{0}, \mathbf{K}], \tag{3}$$

where $\mathbf{1(a, b)}$ is an indicator function, i.e. $\mathbf{1(a, b) = 1}$ only if $a = b$. To measure the operator response, we transform the binary vector of CS-STLTP into a uniform value that is defined as the number of spatial transitions (bitwise changes) following, as discussed in [36]. For example, the pattern (*i.e.* the vector of 16 bins) 0000000000000000 has a value of 0 and 1000000000000000 of 1. In our implementation, we further quantize all possible values into 48 levels. To further improve the capability, we can generate histograms in each color channel and concatenate them together.

The proposed descriptor is computationally efficient and compact to describe the video brick. In addition, by introducing a tolerative comparing range in the LTP operator computation, it is robust to local spatio-temporal noise within a range.

C. Model Definition

Let m be the descriptor length for each brick, and $V = \{v_1, v_2, \dots, v_n\}$, $v_i \in \mathbb{R}^m$ be a sequence of video bricks at a certain location of the observed background. We can use a set of bases (components) $\mathbf{C} = [C_1, C_2, \dots, C_d]$ to represent the subspace where V lies in. Each video brick v_i in V can be represented as

$$v_i = \sum_{j=1}^{d} z_{i,j} C_j + \omega_i, \tag{4}$$

where C_j is the j-th basis (j-th column of matrix $\mathbf{C}$) of the subspace, $z_{i,j}$ the coefficient for C_j, and ω_i the appearance residual. We denote $\mathbf{C}$ to represent appearance consistency of the sequence of video bricks. In some traditional background models by subspace learning, $z_{i,j}$ can be solved and kept as a constant, with the underlying assumption that the appearance of background would be stable within the observations. In contrast, we treat $z_{i,j}$ as the variable term that can be further phrased as the time-varying state, accounting for temporally coherent variations (*i.e.* the motions). For notation simplicity, we neglect the subscript j, and denote $Z = \{z_1, z_2, \dots, z_n\}$ for all the bricks. The dynamic model is formulated as,

$$z_{i+1} = Az_i + \eta_i, \tag{5}$$

where η_i is the state residual, and A is a matrix of $d \times d$ dimensions to model the variations. With this definition, we consider A representing the temporal coherence among the observations.

Therefore, the problem of pursuing dynamic subspace is posed as solving the appearance consistency $\mathbf{C}$ and the temporal coherence A, within the observations. Since the sequence states Z are unknown, we shall jointly solve $\mathbf{C}$, A, Z by minimizing an empirical energy function $\mathcal{F}_n(\mathbf{C}, A, Z)$:

$$\min \mathcal{F}_n(\mathbf{C}, A, Z) = \frac{1}{2n} \sum_{i=1}^{n} \|v_i - Cz_i\|_2^2 + \|z_i - Az_{i-1}\|_2^2. \tag{6}$$

Here $\mathcal{F}_n(\mathbf{C}, A, Z)$ is not completely convex but we can solve it by fixing either Z or $(\mathbf{C}, A)$. Nevertheless, its computation cost is expensive for learning the entire background online. Here we simplify the dynamic model in Equation (5) into a linear system, following the auto-regressive moving average (ARMA) process. In literature, Soatto et al. [40] originally associated the output of ARMA model with dynamic textures, and showed that the first-order ARMA model, driven by white zero-mean Gaussian noise, can capture a wide range of dynamic textures. In our approach, the difficulty of modeling the dynamic variations can be alleviated due to the brick-based representation, i.e. the observed scene is decomposed into video bricks. Thus, we consider the ARMA process a suitable solution to model the time-varying variables, which can be solved efficiently. Specifically, we introduce a robustness term (*i.e.* matrix) B, which includes a number d_ϵ of bases, and we set $\eta_i = B\epsilon_i$, where ϵ_i denotes the noise.

We further summarize the proposed dynamic model, where we add the subscript n to the main components, indicating they are solved within a number n of observations, as,

$$\begin{aligned} v_i &= \mathbf{C}_n z_i + \omega_i, \\ z_{i+1} &= A_n z_i + B_n \epsilon_i, \\ \omega_i &\overset{IID}{\sim} N(0, \Sigma_\omega), \quad \epsilon_i \overset{IID}{\sim} N(0, I_{d_\epsilon}). \end{aligned} \tag{7}$$

In this model, $\mathbf{C}_n \in \mathbb{R}^{m \times d}$ and $A_n \in \mathbb{R}^{d \times d}$ represent the appearance consistency and temporal coherence, respectively. $B_n \in \mathbb{R}^{d \times d_\epsilon}$ is the robustness term constraining the evolution of Z over time. $\omega_i \in \mathbb{R}^m$ indicates the residual corresponding to observation v_i, and $\epsilon_i \in \mathbb{R}^{d_\epsilon}$ the noise of state variations. During the subspace learning, ω_i and ϵ_i are assumed to follow the zero-mean Gaussian distributions. Given a new brick mapped into the subspace, ω_i and ϵ_i can be used to measure how likely the observation is suitable with the subspace, so that we utilize them for foreground object detection during online processing.

The proposed model is time-varying, and the parameters $\mathbf{C}_n, A_n, B_n$ can be updated incrementally along with the processing of new observations, in order to adapt our model with scene changes.

III. Learning Algorithm

In this section, we discuss the learning for spatio-temporal background models, including initial subspace generation and online maintenance. The initial learning is performed at the beginning of system deployment, when only a few foreground objects move in the scene. Afterwards, the system switches to the mode of online maintenance.

A. Initial Model Learning

In the initial stage, the model defined in Equation (7) can be degenerated as a non-dynamic linear system, as the n observations are extracted and fixed. Given a brick sequence $V = \{v_1, v_2, \ldots, v_n\}$, we present an algorithm to identify the model parameters $\mathbf{C}_n$, A_n, B_n, following the sub-optimal solution proposed in [40].

To guarantee the Equation (7) has an unique and canonical solution, we postulate

$$n \gg d, \;\; \mathrm{Rank}(\mathbf{C}_n) = d, \;\; \mathbf{C}_n^\top \mathbf{C}_n = I_d, \tag{8}$$

where I_d is the identity matrix of dimension $d \times d$. The appearance consistency term $\mathbf{C}_n$ can be estimated as,

$$\mathbf{C}_n = \arg\min_{\mathbf{C}_n} \mid W_n - \mathbf{C}_n [\, z_1 \; z_2 \; \cdots \; z_n \,] \mid \tag{9}$$

where W_n is the data matrix composed of observed video bricks $[v_1, v_2, \cdots, v_n]$. The equation (9) satisfies the full rank approximation property and can be thus solved by the singular value decomposition (SVD). We have,

$$\begin{aligned} W_n &= U \Sigma Q^\top, \\ U^\top U &= I, Q^\top Q = I, \end{aligned} \tag{10}$$

where Q is the unitary matrix, U includes the eigenvectors, and Σ is the diagonal matrix of the singular values. Thus, $\mathbf{C}_n$ is treated as the first d components of U, and the state matrix $[z_1 \; z_2 \; \cdots \; z_n]$ as the product of $d \times d$ sub-matrix of Σ and the first d columns of $Q^\top$.

The temporal coherence term A_n is calculated by solving the following linear problem:

$$A_n = \arg\min_{A_n} \mid [\, z_2 \; z_3 \; \cdots \; z_n \,] - A_n [\, z_1 \; z_2 \; \cdots \; z_{n-1} \,] \mid . \tag{11}$$

The statistical robustness term B_n is estimated by the reconstruction error E

$$\begin{aligned} E &= [\, z_2 \; z_3 \; \cdots \; z_n \,] - A_n [\, z_1 \; z_2 \; \cdots \; z_{n-1} \,] \\ &= B_n [\, \epsilon_1 \; \epsilon_2 \; \cdots \; \epsilon_{n-1} \,], \end{aligned} \tag{12}$$

where $B_n \cong \frac{1}{\sqrt{n-1}} E$. Since the rank of A_n is d and $d \ll n$, the rank of input-to-state noise d_ϵ is assumed to be much smaller than d. That is, the dimension of E can be further reduced by SVD: $E = U_\epsilon \, \Sigma_\epsilon \, Q_\epsilon^\top$, and we have

$$B_n = \frac{1}{\sqrt{n-1}} \left[U_\epsilon^1 \; \cdots \; U_\epsilon^{d_\epsilon} \right] \begin{bmatrix} \Sigma_\epsilon^1 & & \\ & \ddots & \\ & & \Sigma_\epsilon^{d_\epsilon} \end{bmatrix}. \tag{13}$$

The values of d, d_ϵ essentially imply the complexity of subspace from the aspects of appearance consistence and temporal coherence, respectively. For example, video bricks containing static content can be well described with a function of low dimensions while highly dynamic video bricks (*e.g.*, from an active fountain) require more bases to generate. In real surveillance scenarios, it is not practical to pre-determine the complexity of scene environments. Hence, in the proposed method, we adaptively determine d, d_ϵ by thresholding eigenvalues in Σ and Σ_ϵ, respectively.

$$\begin{aligned} d^* &= \arg\max_{d} \; \Sigma^d \; > \; T_d, \\ d_\epsilon^* &= \arg\max_{d_\epsilon} \; \Sigma_\epsilon^{d_\epsilon} \; > \; T_{d_\epsilon}, \end{aligned} \tag{14}$$

where Σ^d indicates the d-th eigenvalue in Σ and $\Sigma_\epsilon^{d_\epsilon}$ the d_ϵ-th eigenvalue in Σ_ϵ.

B. Online Model Maintenance

Then we discuss the online processing with our model that segments foreground moving objects and keeps the model updated.

1) Foreground Segmentation: Given one newly appearing video brick v_{n+1}, we can determine whether pixels in v_{n+1} belong to the background or not by thresholding their appearance residual and state residual. We first estimate the state of v_{n+1} with the existing $\mathbf{C}_n$,

$$z'_{n+1} = \mathbf{C}_n^\top v_{n+1}, \tag{15}$$

and further the appearance residual of v_{n+1}

$$\omega_{n+1} = v_{n+1} - \mathbf{C}_n z'_{n+1}. \tag{16}$$

As the state z_n and the temporal coherence A_n have been solved, we can then estimate the state residual ϵ_n according to Equation (7),

$$\begin{aligned} B_n \epsilon_n &= z'_{n+1} - A_n z_n \\ \Rightarrow \epsilon_n &= \mathrm{pinv}\,(B_n)\,(z'_{n+1} - A_n z_n), \end{aligned} \tag{17}$$

where pinv denotes the operator of pseudo-inverse.

With the state residual ϵ_n and the appearance residual ω_{n+1} for the new video brick v_{n+1}, we conduct the following criteria for foreground segmentation, in which two thresholds are introduced.

1) v_{n+1} is classified into background, only if all dimensions of ϵ_n are less than a threshold T_ϵ.
2) If v_{n+1} has been labeled as non-background, perform the pixel-wise segmentation by comparing ω_{n+1} with a threshold T_ω: the pixel is segmented as foreground if its corresponding dimension in ω_{n+1} is greater than T_ω.

2) Model Updating: During the online processing, the key problem for model updating is to deal with foreground disturbance, *i.e.* to avoid absorbing pixels from foreground objects or noise.

In this work, we develop an effective approach to update the model with the synthesized data. We first generate a video brick from the current model, namely noise-free brick, $\hat{v}_{n+1}$, as

$$\begin{aligned} \hat{z}_{n+1} &= A_n z_n, \\ \hat{v}_{n+1} &= \mathbf{C}_n \, \hat{z}_{n+1}. \end{aligned} \tag{18}$$

Then we extract pixels from $\hat{v}_{n+1}$ to compensate occluded (*i.e.* foreground) pixels in the newly appearing brick. Concretely, the pixels labeled as non-background are replaced by the pixels from the noise-free video brick at the same place. We can thus obtain a synthesized video brick $\bar{v}_{n+1}$ for model updating.

Given the brick $\bar{v}_{n+1}$, the data matrix W_n composed of observed video bricks is extended to W_{n+1}. Then we update the model $\mathbf{C}_{n+1}$ according to Equation (9).

Our algorithm of model updating includes two steps: (i) update parameters for appearance consistency $\mathbf{C}_{n+1}$ by employing the incremental subspace learning technique, and (ii) update parameters of state variations A_{n+1}, B_{n+1}.

(i) Step 1. For the d-dimension subspace, with eigenvectors $\mathbf{C}_n$ and eigenvalues Λ_n, its covariance matrix Cov_n can be approximated as

$$\mathrm{Cov}_n \approx \sum_{j=1}^{d} \lambda_{n,j} c_{n,j} c_{n,j}^{\top} = \mathbf{C}_n \Lambda_n \mathbf{C}_n^{\top}, \tag{19}$$

where $c_{n,j}$ and $\lambda_{n,j}$ denote the j-th eigenvector and eigenvalue, respectively. With the newly synthesized data $\bar{v}_{n+1}$, the updated covariance matrix Cov_{n+1} is formulated as

$$\begin{aligned} \mathrm{Cov}_{n+1} &= (1-\alpha)\,\mathrm{Cov}_n + \alpha\, \bar{v}_{n+1}\, \bar{v}_{n+1}^{\top} \\ &\approx (1-\alpha)\, \mathbf{C}_n \Lambda_n \mathbf{C}_n^{\top} + \alpha\, \bar{v}_{n+1}\, \bar{v}_{n+1}^{\top} \\ &= \sum_{i=1}^{d} (1-\alpha)\, \lambda_{n,i}\, c_{n,i}\, c_{n,i}^{\top} + \alpha\, \bar{v}_{n+1}\, \bar{v}_{n+1}^{\top}, \end{aligned} \tag{20}$$

where α denotes the learning rate. The covariance matrix can be further re-formulated to simplify computation, as,

$$\mathrm{Cov}_{n+1} = Y_{n+1} Y_{n+1}^{\top}, \tag{21}$$

where $Y_{n+1} = [y_{n+1,1}\, y_{n+1,2} \ldots y_{n+1,d+1}]$ and each column $y_{n+1,j}$ in Y_{n+1} is defined as

$$y_i = \begin{cases} \sqrt{1-\alpha\lambda_j}\, c_{n,i}, & \text{if } 1 < j < d, \\ \sqrt{\alpha}\, \bar{v}_{n+1}, & \text{if } j = d+1. \end{cases} \tag{22}$$

To reduce the computation cost, we can estimate $\mathbf{C}_{n+1}$ by a smaller matrix $Y_{n+1}^{\top} Y_{n+1}$, instead of the original large matrix Cov_{n+1}.

$$(Y_{n+1}^{\top}\, Y_{n+1}) e_{n+1,j} = \lambda_{n+1,j}\, e_{n+1,j} \quad j = 1, 2, \ldots, d+1, \tag{23}$$

where $e_{n+1,j}$ and $\lambda_{n+1,j}$ are the j-th eigenvector and eigenvalue of matrix $Y_{n+1}^{\top} Y_{n+1}$, respectively. Let $c_{n+1,j} = Y_{n+1} e_{n+1,j}$, and we re-write Equation (23) as

$$\begin{aligned} Y_{n+1}\, Y_{n+1}^{\top}\, Y_{n+1}\, e_{n+1,j} &= \lambda_{n+1,j}\, Y_{n+1}\, e_{n+1,j}, \\ \mathrm{Cov}_{n+1}\, c_{n+1,j} &= \lambda_{n+1,i}\, c_{n+1,j} \qquad j = 1, 2, \ldots, d+1. \end{aligned} \tag{24}$$

We thus obtain the updated eigenvectors $\mathbf{C}_{n+1}$ and the corresponding eigenvalues Λ_{n+1} of the new covariance matrix Cov_{n+1}. Note that the dimension of the subspace is automatically increased along with the newly added data $\bar{v}_{n+1}$. To guarantee the appearance parameters remain stable, we keep the main principal (*i.e.* top d) eigenvectors and eigenvalues while discarding the least significant components.

The above incremental subspace learning algorithm has been widely applied in several vision tasks such as face recognition and image segmentation [43]–[45], and also for background modeling in [4], [25] and [46]. However, the noise observations caused by moving objects or scene variations often disturb the subspace maintenance, *e.g.* the eigenvectors could change dramatically during the processing. Many efforts [47], [48] have been dedicated to improve the robustness of incremental learning by using statistical analysis. Several discriminative learning algorithms [49] were also employed to train background classifiers that can be incrementally updated. In this work, we utilize a version of Robust Incremental PCA (RIPCA) [50] to cope with the outliers in $\bar{v}_{n+1}$. Note that $\bar{v}_{n+1}$ consists of pixels either from the generated data $\hat{v}_{n+1}$ or real videos, where outliers may exist in some dimensions.

In the traditional PCA learning, the solution is derived by minimizing a least-squared reconstruction error,

$$\min |r_{n+1}|^2 = |\mathbf{C}_n \mathbf{C}_n^{\top} \bar{v}_{n+1} - \bar{v}_{n+1}|^2. \tag{25}$$

Following [50], we impose a robustness function $w(t) = \frac{1}{1+(t/\rho)^2}$ over each dimension of r_{n+1}, and the target can be re-defined as,

$$\min \sum_j (r_{n+1}^k)^2 \leftarrow w(r_{n+1}^k)(r_{n+1}^k)^2, \tag{26}$$

where the superscript k indicates the k-th dimension. The parameter ρ in the robustness function is estimated by

$$\begin{aligned} \rho &= [\rho^1, \rho^2, \ldots, \rho^{|\bar{v}_{n+1}|}]^{\top} \\ \rho^k &= \max_{i=1}^{d} \beta \sqrt{\lambda_{n,i}}\, |\, c_{n,j}^k\, |, \; j = 1, 2, \ldots, |\bar{v}_{n+1}| \end{aligned} \tag{27}$$

where β is a fixed coefficient. The k-th dimension of ρ is proportional to the maximal projection of the current eigenvectors on the k-th dimension, (*i.e.* ρ^k is weighted by their corresponding eigenvalues). Note that $w(r_{n+1}^k)$ is a function of the residual error which should be calculated for each vector dimension. And the computation cost for $w(r_{n+1}^k)$ can be neglected in the analytical solution.

Accordingly, we can update the observation $\bar{v}_{n+1}$ over each dimension by computing the function $w(r_{n+1}^k)$,

$$\tilde{v}_{n+1}^k = \sqrt{w(r_{n+1}^k)}\, \bar{v}_{n+1}^k. \tag{28}$$

That is, we treat $\tilde{v}_{n+1}$ as the new observation during the procedure of incremental learning.

(i) Step 2. With the fixed $\mathbf{C}_{n+1}$, we then update the parameters of state variations A_{n+1}, B_{n+1}. We first estimate the latest state z_{n+1} based on the updated $\mathbf{C}_{n+1}$ as,

$$z_{n+1} = \mathbf{C}_{n+1}^{\top} \tilde{v}_{n+1}. \tag{29}$$

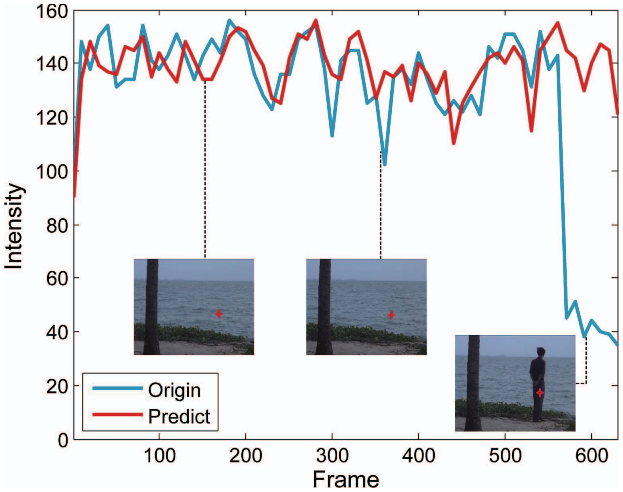


Fig. 3. An example to demonstrate the robustness of model maintenance. In the scenario of dynamic water surfaces, we visualize the original and predicted intensities for a fixed position (denoted by the red star), with the blue and red curves, respectively. With our updating scheme, when the position is occluded by a foreground object during from frame 551 to 632, the predicted intensities are not disturbed by foreground, *i.e.* the model remains stable.

A_{n+1} can be further calculated, by re-solving the linear problem of a fixed number of latest observed states,

$$A_{n+1}\,[\; z_{n-l+1} \;\cdots\; z_n \;] = [\; z_{n-l+2} \;\cdots\; z_{n+1} \;], \qquad (30)$$

where l indicates the number of latest observed states, *i.e.* the span of observations. And similarly, we update B_{n+1} by computing the new reconstruction error $E = [z_{n-l+2} \;\cdots\; z_{n+1}] - A_{n+1}\,[z_{n-l+1} \;\cdots\; z_n]$.

We present an empirical study in Fig. 3 to demonstrate the effectiveness of this updating method. The video for background modeling includes dynamic water surfaces. Here we visualize the original and predicted intensities for a fixed position (denoted by the red star), with the blue and red curves, respectively. We can observe that the model remains stable against foreground occlusion.

3) Time Complexity Analysis: We mainly employ SVD and linear programming in the initial learning. The time complexity of SVD is $O(n^3)$ and the learning time of linear programming is $O(n^2)$. For a certain location, the time complexity of initial learning is $O(n^3) + O(n^2) = O(n^3)$ for each subspace, where n denotes the number of video bricks for model learning. As for online learning, incremental subspace learning and linear programming are utilized. Given a d-dimension subspace, the time complexity for component updating (*i.e.* step 1 of the model maintenance) is $O(dn^2)$. Thus, the total time complexity for online learning is $O(dn^2) + O(l^2)$, where l is the number of states used to solve the linear problem.

We summarize the algorithm sketch of our framework in Algorithm 1.

IV. Experiments

In this section, we first introduce the datasets used in the experiments and the parameter settings, then present the experimental results and comparisons. The discussions of system components are proposed at last.

Algorithm 1: The Sketch of the Proposed Algorithm

```
Input: Video brick sequence $V = \{v_1, v_2, \ldots, v_n\}$ for
       every location for the scene.
Output: Maintained Background models and foreground
        regions
forall the locations for the scene do
    Given the observed video bricks $V$, extract the
    CS-STLTP descriptor;
    Initialize the subspace by estimating $\mathbf{C}_n, A_n, B_n$
    using Equation (8)-(14);
    for the newly appearing video brick $v_{n+1}$ do
        (1) Extract the CS-STLTP descriptor for $v_{n+1}$;
        (2) Calculate its state residual $\epsilon_n$ and appearances
        residual $\omega_{n+1}$ by Equation (16) and (17);
        (3) For each pixel of $v_{n+1}$, classify it into
        foreground or background by thresholding the
        two residuals with $\epsilon_n$, $\omega_{n+1}$;
        (4) Generate the noise-free brick $\hat{v}_{n+1}$ from the
        current model by Equation (18);
        (5) Synthesize video brick $\bar{v}_{n+1}$ for model
        updating;
        (6) Update $\bar{v}_{n+1}$ into $\tilde{v}_{n+1}$ by introducing a
        robustness function;
        (7) Update the new appearance parameter $\mathbf{C}_{n+1}$
        by calculating the covariance matrix $\mathrm{Cov}_{n+1}$
        with the learning rate $\alpha$;
        (8) Update the state variation parameters
        $A_{n+1}$, $B_{n+1}$ ;
    end
end
```

A. Datasets and Settings

We collect a number of challenging videos to validate our approach, which are publicly available or from real surveillance systems. Two of them (AirportHall and TrainStation) from the PETS database[1] include crowded pedestrians and moving cast shadows; five highly dynamic scenes[2] include waving curtain active fountain, swaying trees, water surface; the others contain extremely difficult cases such as heavy rain, sudden and gradual light changing. Most of the videos include thousands of frames, and some of the frames are manually annotated as the ground-truth provided by the original databases.

Our algorithm has been adopted in a real video surveillance system and achieves satisfactory performances. The system is capable of processing 15 ~ 20 frames per second in the resolution 352 × 288 pixels. The hardware architecture is an Intel i7 2600 (3.4 GHz) CPU and 8GB RAM desktop computer.

All parameters are fixed in the experiments, including the contrast threshold for CS-STLTP descriptor $\tau = 0.2$, the dimension threshold for ARMA model $T_d = 0.5$, $T_{d_\epsilon} = 0.5$, the span of observations for model updating $l = 60$, and the size of bricks $4 \times 4 \times 5$. For foreground segmentation, the

[1] Downloaded from http://www.cvg.rdg.ac.uk/slides/pets.html.

[2] Downloaded from http://perception.i2r.a-star.edu.sg

threshold of appearance residual $T_\omega = 3$, update threshold $T_\epsilon = 3$ and $T_\omega = 5$, $T_\epsilon = 4$ for RGB. In the online model maintenance, the coefficient $\beta = 2.3849$, the learning rate $\alpha = 0.05$ for RIPCA.

In the experiments, we use the first 50 frames of each testing video to initialize our system (i.e. to perform the initial learning), and keep model updated in the rest of sequence. In addition, we utilize a standard post-processing to eliminate areas including less than 20 pixels. All other competing approaches are executed with the same setting as our approach.

We utilize the F-score as the benchmark metric, which measures the segmentation accuracy by considering both the recall and the precision. The F-score is defined as

$$F = \frac{2TP}{2TP + FP + FN}, \tag{31}$$

where TP is true positives (foreground objects), FN false negatives (false background pixels), FP false positive (false foreground pixels).

B. Experimental Results

Experimental results. We compare the proposed method (STDM) with six state-of-the-art online background subtraction algorithms including Gaussian Mixture Model (GMM) [1] as baseline, improved GMM [8][3], online auto-regression model [20], non-parametric model with scale-invariant local patterns [14], discriminative model using generalized Struct 1-SVM [19][4], and the Bayesian joint domain-range (JDR) model [11][5]. In the comparisons, for the methods [1], [8], [11], [19] we use their released codes, and implement the methods [14], [20] by ourselves. The F-scores (%) over all 10 videos are reported in Table I, where the last two columns report results of our method using either RGB or CS-STLTP as the feature. Note that for the result using the RGB feature we represent each video brick by concatenating the RGB values of all its pixels. We also exhibit the results and comparisons using the precision-recall (PR) curves, as shown in Fig. 4. Due to space limitation, we only show results on 5 videos. From the results, we can observe that the proposed method outperforms the other methods in most videos in general. For the scenes with highly dynamic backgrounds (*e.g.*, the #2 #5 and #10 scenes), the improvements made by our method are more than 10%. And the system enables us to well handle the indistinctive foreground objects (*i.e.* small objects or background-like objects in the #1, #3 scenes). Moreover, we make significant improvements (*i.e.* 15% $\sim$ 25%) in the scene #6 and #7 including both sudden and gradual lighting changes. A number of sampled results of background subtraction are exhibited in Fig. 5.

The benefit of using the proposed CS-STLTP feature is clearly validated by observing the results shown in Table I and Fig. 5. In general, our approach simply using RGB values can achieve satisfying performances for the common scenes, *e.g.*, with fair appearance and motion changes, while the CS-SILTP operator can better handle highly dynamic variations (*e.g.* sudden illumination changing, rippling water). In addition, we also compare CS-STLTP with the existing scale invariant descriptor SILTP proposed in [14]. We reserve all settings in our approach except replacing the feature by SILTP, and achieve the average precision over all 10 videos: 69.70%. This result shows that CS-STLTP is very suitable and effective for the video brick representation.

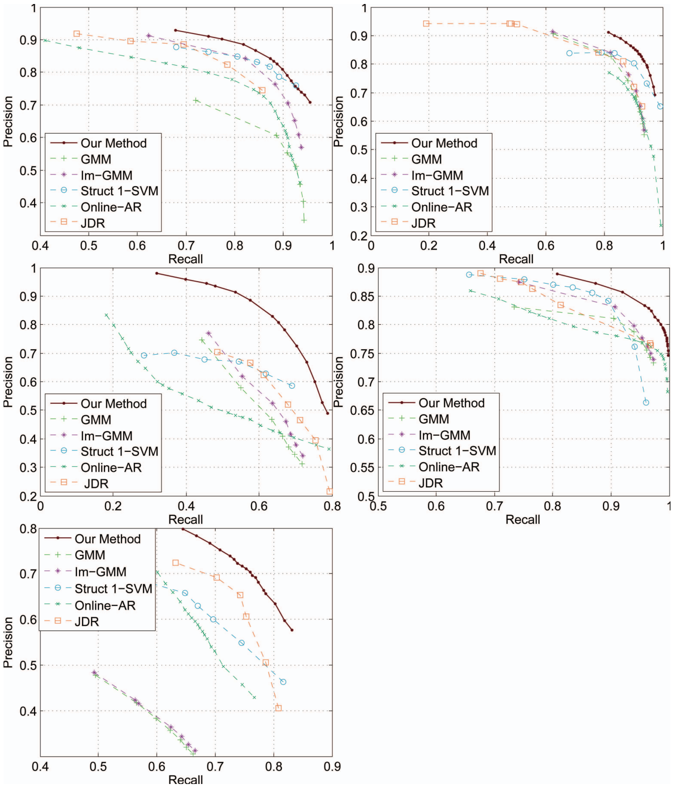


Fig. 4. Experimental results generated by our approach and competing methods on 5 videos: first row left, the scene including a dynamic curtain and indistinctive foreground objects (*i.e.* having similar appearance with backgrounds); first row right, the scene with heavy rain; second row left, an indoor scene with the sudden lighting changes; second row right, the scene with dynamic water surface; third row, a busy airport. The precision-recall (PR) curve is introduced as the benchmark measurement for all the 6 algorithms.

C. Discussion

Furthermore, we conduct the following empirical studies to justify the parameter determinations and settings of our approach.

a) Efficiency: Like other online-learning background models, there is a trade-off between the model stability and maintenance efficiency. The corresponding parameter in our method is the learning rate α. We tune α in the range of $0 \sim 0.3$ by fixing the other model parameters and visualize the quantitative results of background subtraction, as shown in Fig. 6(a). From the results, we can observe this parameter is insensitive in range $0 \sim 0.1$ in our model. In practice, once the scene is extremely busy and crowded, it could be set as a relative small value to keep the model stable.

b) Feature effectiveness: The contrast threshold τ is the only parameter in CS-STLTP operator, which affects the power of feature to character spatio-temporal information within video bricks. From the empirical results of parameter tuning,

[3] Available at http://dparks.wikidot.com/background-subtraction

[4] Available at http://www.cs.mun.ca/~gong/Pages/Research.html

[5] Available at http://www.cs.cmu.edu/~yaser/

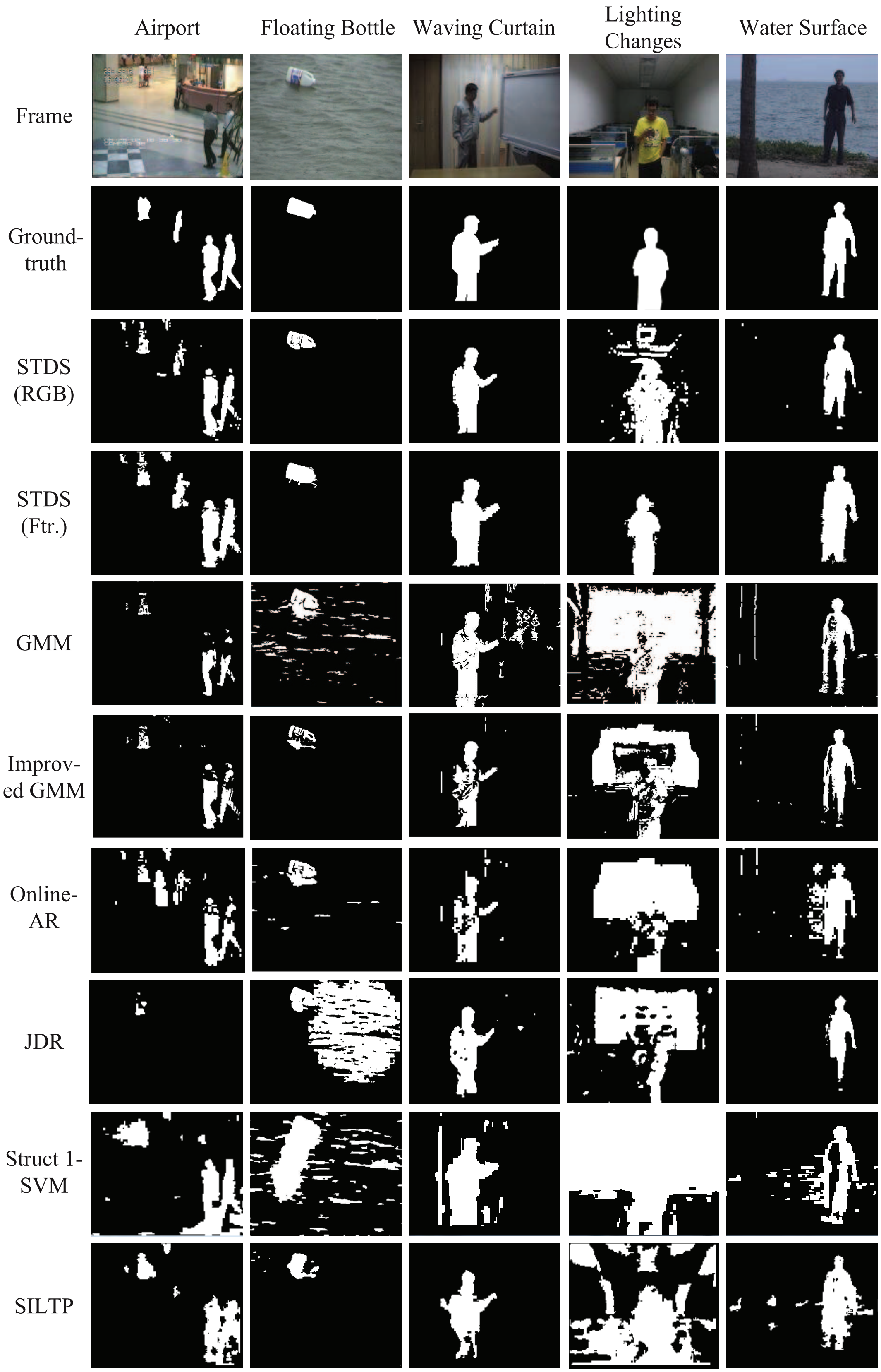


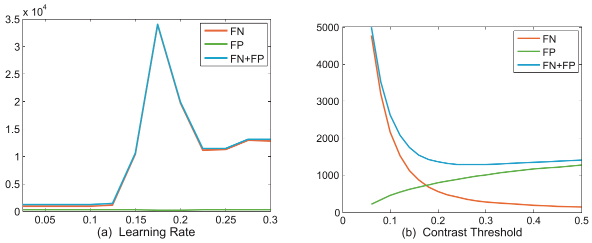
Fig. 5. Sampled results of background subtraction generated by our approach (using RGB or CS-STLTP as the feature and RIPCA as the update strategy) and other competing methods.

as shown in Fig. 6 (b), we can observe that the appropriate range for τ is $0.15 \sim 0.25$. In practice, the model could become sensitive to noise by setting a very small value of τ (say $\tau < 0.15$), and too large τ (say $\tau > 0.25$) might reduce the accuracy on detecting foreground regions with homogeneous appearances.

TABLE I

QUANTITATIVE RESULTS AND COMPARISONS ON THE 10 COMPLEX VIDEOS USING THE F-SCORE (%) MEASUREMENT. THE LAST TWO COLUMNS REPORT THE RESULTS OF OUR METHOD USING EITHER RGB OR CS-STLTP AS THE FEATURE

Scene	GMM[1]	Im-GMM[8]	Online-AR[20]	JDR[11]	SVM[19]	PKDE[14]	STDM(RGB)	STDM(Ftr.)
1# Airport	46.99	47.36	62.72	60.23	65.35	68.14	**70.52**	66.40
2# Floating Bottle	57.91	57.77	43.79	45.64	47.87	59.57	69.04	**78.17**
3# Waving Curtain	62.75	74.58	77.86	72.72	77.34	78.01	**79.74**	74.93
4# Active Fountain	52.77	60.11	70.41	68.53	74.94	76.33	76.85	**85.46**
5# Heavy Rain	71.11	81.54	78.68	75.88	**82.62**	76.71	79.35	75.29
6# Sudden Light	47.11	51.37	37.30	52.26	47.61	52.63	51.56	**74.57**
7# Gradual Light	51.10	50.12	13.16	47.48	62.44	54.86	54.84	**77.41**
8# Train Station	65.12	68.80	36.01	57.68	61.79	67.05	**73.43**	66.35
9# Swaying Trees	19.51	23.25	63.54	45.61	24.38	42.54	43.71	**75.89**
10# Water Surface	79.54	86.01	77.31	84.27	83.13	74.30	88.54	**88.68**
Average	55.39	59.56	57.02	60.23	59.79	63.08	68.75	**76.31**



(a) Learning Rate

(b) Contrast Threshold

Fig. 6. Discussion of parameter selection: (i) learning rate α for model maintenance (in (a)) and (ii) the contrast threshold of CS-STLTP feature τ (in (b)). In each figure, the horizontal axis represents the different parameter values; the three lines in different colors denote, respectively, the false positive (FP), false negative (FN), and the sum of FP and FN.

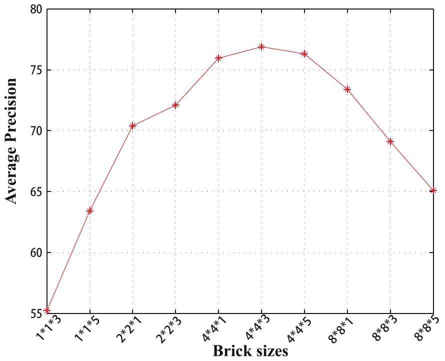


Fig. 7. Empirical study for the size of video brick in our approach. We carry on the experiments on the 10 videos with different brick size while keeping the rest settings. The vertical axis represents the average precisions of background subtraction and the horizontal represents the different sizes of video bricks with respect to background decomposition.

c) Size of video brick: One may be interested in how the system performance is affected by the size of video brick for background decomposition, so that we present an empirical study on different sizes of video bricks in Fig. 7. We observe that the best result is achieved with the certain brick size of $4 \times 4 \times 3$, and the results with the sizes of $4 \times 4 \times 1$ and $4 \times 4 \times 5$ are also satisfied. As of very small bricks (*e.g.* $1 \times 1 \times 3$), few spatio-temporal statistics are captured and the models may have problems on handling scene variations. The bricks of large sizes (*e.g.* $8 \times 8 \times 5$) carry too much information, and their subspaces cannot be effectively generated by the linear ARMA model. The experimental results are also accordant with our motivations in Section I. In practice, we can flexibly set the size according to the resolutions of surveillance videos.

d) Model initialization: Our method is not sensitive to the number of observed frames in the initial stage of subspace generation. We test the different numbers, say 30, 40, 60, on two typical surveillance scenes, i.e. the Airport Hall (scene #1) and the Train Station (scene #8). The F-score outputs show the deviations with different numbers of initial frames are very small, e.g. less than 0.2. In general, we require the observed scenes to be relatively clean for initialization, although a few objects that move across are allowed.

V. CONCLUSION

This paper studies an effective method for background subtraction, addressing the all challenges in real surveillance scenarios. In the method, we learn and maintain the dynamic texture models within spatio-temporal video patches (*i.e.* video bricks). Sufficient experiments as well as empirical analysis are presented to validate the advantages of our method.

In the future, we plan to improve the method in two aspects. (1) Some efficient tracking algorithms can be employed into the framework to better distinguish the foreground objects. (2) The GPU-based implementation can be developed to process each part of the scene in parallel, and it would probably significantly improve the system efficiency.

REFERENCES

[1] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Conf. CVPR*, Jun. 1999.

[2] T. Bouwmans, F. E. Baf, and B. Vachon, "Background modeling using mixture of Gaussians for foreground detection-a survey," *Recent Patents Comput. Sci.*, vol. 1, no. 3, pp. 219–237, 2008.

[3] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, Jul. 2008.

[4] D.-M. Tsai and S.-C. Lai, "Independent component analysis-based background subtraction for indoor surveillance," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 158–167, Jan. 2009.

[5] H. Chang, H. Jeong, and J. Choi, "Active attentional sampling for speed-up of background subtraction," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 2088–2095.

[6] X. Liu, L. Lin, S. Yan, H. Jin, and W. Tao, "Integrating spatio-temporal context with multiview representation for object recognition in visual surveillance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 393–407, Apr. 2011.

[7] L. Lin, Y. Lu, Y. Pan, and X. Chen, "Integrating graph partitioning and matching for trajectory analysis in video surveillance," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4844–4857, Apr. 2012.

[8] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th IEEE ICPR*, Aug. 2004, pp. 28–31.

[9] D. Lee, "Effective Gaussian mixture learning for video background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 827–832, May 2005.

[10] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, Jun. 2002.

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

[12] C. Benedek and T. Sziranyi, "Bayesian foreground and shadow detection in uncertain frame rate surveillance videos," *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 608–621, Apr. 2008.

[13] O. Barnich and M. Droogenbroeck, "Vibe: A universal background subtraction algorithm for video sequences," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, Jun. 2011.

[14] S. Liao, G. Zhao, V. Kellokumpu, M. Pietikainen, and S. Li, "Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2010, pp. 1301–1306.

[15] J. Pilet, C. Strecha, and P. Fua, "Making background subtraction robust to sudden illumination changes," in *Proc. ECCV*, 2008, pp. 567–580.

[16] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint," in *Proc. Comput. Vis. ACCV Workshops*, 2013, pp. 315–320.

[17] M. Wu and X. Peng, "Spatio-temporal context for codebook-based dynamic background subtraction," *AEU Int. J. Electron. Commun.*, vol. 64, no. 8, pp. 739–747, 2010.

[18] H.-H. Lin, T.-L. Liu, and J.-H. Chuang, "Learning a scene background model via classification," *IEEE Trans. Signal Process.*, vol. 57, no. 5, pp. 1641–1654, May 2009.

[19] L. Cheng and M. Gong, "Realtime background subtraction from dynamic scenes," in *Proc. 12th IEEE ICCV*, Sep./Oct. 2009, pp. 2066–2073.

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

[21] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proc. 9th IEEE ICCV*, Oct. 2003, pp. 44–50.

[22] Y. Wang, K.-F. Loe, and J.-K. Wu, "A dynamic conditional random field model for foreground and shadow segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 2, pp. 279–289, Feb. 2007.

[23] K. A. Patwardhan, G. Sapiro, and V. Morellas, "Robust foreground detection in video using pixel layers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 746–751, Apr. 2008.

[24] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, Jan. 2010.

[25] Y. Zhao, H. Gong, L. Lin, and Y. Jia, "Spatio-temporal patches for night background modeling by subspace learning," in *Proc. IEEE ICPR*, Dec. 2008, pp. 1–4.

[26] X. Liang, L. Lin, and L. Cao, "Learning latent spatio-temporal compositional model for human action recognition," in *Proc. ACM Int. Conf. Multimedia*, Oct. 2013, pp. 263–272.

[27] X. Liu, L. Lin, and H. Jin, "Contextualized trajectory parsing with spatio-temporal graph," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 3010–3024, Dec. 2013.

[28] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 774–780, Aug. 2000.

[29] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, and A. K. Jain, "A background model initialization algorithm for video surveillance," in *Proc. IEEE ICCV*, Jul. 2001, pp. 733–740.

[30] W. Kim, C. Jung, and C. Kim, "Spatiotemporal saliency detection and its applications in static and dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 446–456, Apr. 2011.

[31] L. Lin, X. Liu, and S.-C. Zhu, "Layered graph matching with composite cluster sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 8, pp. 1426–1442, Aug. 2010.

[32] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[33] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 168–182, Oct. 2007.

[34] M. Heikkila, M. Pietikainen, and C. Schmid, "Description of interest regions with local binary patterns," *Pattern Recognit.*, vol. 42, no. 3, pp. 425–436, Mar. 2009.

[35] J. Yao and J. Odobez, "Multi-layer background subtraction based on color and texture," in *Proc. IEEE CVPR*, Jun. 2007, pp. 1–8.

[36] Z. Guo, L. Zhang, and D. Zhang, "Rotation invariant texture classification using LBP variance (LBPV) with global matching," *Pattern Recognit.*, vol. 43, no. 3, pp. 706–719, Mar. 2010.

[37] L. Lin, P. Luo, X. Chen, and K. Zeng, "Representing and recognizing objects with massive local image patches," *Pattern Recognit.*, vol. 45, no. 1, pp. 231–240, Jan. 2012.

[38] E. Hannan and M. Deistler, *Statistical Theory Of Linear Systems* (Probability and Mathematical Statistics). New York, NY, USA: Wiley, 1988.

[39] L. Lin, T. Wu, J. Porway, and Z. Xu, "A stochastic graph grammar for compositional object representation and recognition," *Pattern Recognit.*, vol. 42, no. 7, pp. 1297–1307, Jul. 2009.

[40] S. Soatto, G. Doretto, and Y. Wu, "Dynamic textures," *Int. J. Comput. Vis.*, vol. 52, no. 2, pp. 91–109, 2003.

[41] P. Saisan, G. Doretto, Y. Wu, and S. Soatto, "Dynamic texture recognition," in *Proc. IEEE CVPR*, Jun. 2001, pp. 58–63.

[42] G. Doretto, D. Cremers, P. Favaro, and S. Soatto, "Dynamic texture segmentation," in *Proc. 9th IEEE ICCV*, Oct. 2003, pp. 1236–1242.

[43] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *Proc. 9th IEEE ICCV*, Oct. 2003, pp. 1494–1501.

[44] M. Artac, M. Jogan, and A. Leonardis, "Incremental PCA for on-line visual learning and recognition," in *Proc. 16th ICPR*, 2002, pp. 781–784.

[45] A. Levy and M. Lindenbaum, "Sequential karhunen-loeve basis extraction and its application to images," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1371–1374, Aug. 2000.

[46] L. Wang, L. Wang, M. Wen, Q. Zhuo, and W. Wang, "Background subtraction using incremental subspace learning," in *Proc. IEEE ICIP*, Sep./Oct. 2007, pp. V-45–V-48.

[47] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

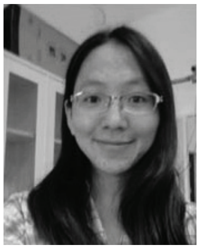
[48] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3419–3430, Dec. 2011.

[49] D. Farcas, C. Marghes, and T. Bouwmans, "Background subtraction via incremental maximum margin criterion: A discriminative subspace approach," *Mach. Vis. Appl.*, vol. 23, no. 6, pp. 1083–1101, Nov. 2012.

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

Liang Lin is a Full Professor with the School of Advanced Computing, Sun Yat-sen University, Guangzhou, China. He received the B.S. and Ph.D. degrees from the Beijing Institute of Technology, Beijing, China, in 1999 and 2008, respectively, and the Ph.D. degree from the Department of Statistics, University of California at Los Angeles (UCLA), Los Angeles, CA, USA, in 2007. His Ph.D. dissertation was achieved the China National Excellent Ph.D. Thesis Award Nomination in 2010. He was a Post-Doctoral Research Fellow with the Center for Vision, Cognition, Learning, and Art at UCLA. His research focuses on new models, algorithms, and systems for intelligent processing and understanding of visual data such as images and videos. He has authored more than 50 papers in top tier academic journals and conferences, including the PROCEEDINGS OF THE IEEE, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE PATTERN RECOGNITION, Computer Vision and Pattern Recognition Conference, the International Conference on Computer Vision, the European Conference on Computer Vision, the ACM Multimedia Conference, and the Conference on Neural Information Processing Systems. He was supported by several promotive programs or funds for his works, such as the Program for New Century Excellent Talents of the Ministry of Education, China, in 2012, the Program of Guangzhou Zhujiang Star of Science and Technology in 2012, and the Guangdong Natural Science Funds for Distinguished Young Scholars in 2013. He was a recipient of the Best Paper Runners-Up Award in ACM NPAR 2010 and the Google Faculty Award in 2012.

Yuanlu Xu received the master's degree from the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China. He is currently pursuing the Ph.D. degree with the University of California at Los Angeles, Los Angeles, CA, USA. His current advisor is Prof. L. Lin, and they have cooperated on publishing a couple of papers on computer vision. He received the B.E. (Hons.) degree from the School of Software, Sun Yat-sen University. His research interests are in video surveillance, image matching, and statistical modeling and inference.



Xiaodan Liang received the B.B.A degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2010, where she is currently pursuing the Ph.D. degree with the School of Information Science and Technology. She has authored several research papers in top-tier academic conferences and journals. Her research focuses on structured vision models and multimedia understanding.

Jianhuang Lai received the M.Sc. degree in applied mathematics and the Ph.D. degree in mathematics from Sun Yat-sen University, Guangzhou, China, in 1989 and 1999, respectively. He joined Sun Yat-sen University in 1989 as an Assistant Professor, where he is currently a Professor with the Department of Automation, School of Information Science and Technology, and the Dean of the School of Information Science and Technology. His current research interests are in the areas of digital image processing, pattern recognition, multimedia communication, wavelet, and its applications. He has authored over 100 scientific papers in the international journals and conferences on image processing and pattern recognition, e.g., the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, PART B, the IEEE PATTERN RECOGNITION, the International Conference on Computer Vision, the Computer Vision and Pattern Recognition Conference, and the International Conference on Data Minin. He serves as a Standing Member of the Image and Graphics Association of China and also serves as a Standing Director of the Image and Graphics Association of Guangdong.