Total Variation Regularized RPCA for Irregularly Moving Object Detection Under Dynamic Background

Xiaochun Cao, *Senior Member, IEEE*, Liang Yang, and Xiaojie Guo, *Member, IEEE*

***Abstract*—Moving object detection is one of the most fundamental tasks in computer vision. Many classic and contemporary algorithms work well under the assumption that backgrounds are stationary and movements are continuous, but degrade sharply when they are used in a real detection system, mainly due to: 1) the dynamic background (e.g., swaying trees, water ripples and fountains in real scenarios, as well as raindrops and snowflakes in bad weather) and 2) the irregular object movement (like lingering objects). This paper presents a unified framework for addressing the difficulties mentioned above, especially the one caused by irregular object movement. This framework separates dynamic background from moving objects using the spatial continuity of foreground, and detects lingering objects using the temporal continuity of foreground. The proposed framework assumes that the dynamic background is sparser than the moving foreground that has smooth boundary and trajectory. We regard the observed video as being made up of the sum of a low-rank static background, a sparse and smooth foreground, and a sparser dynamic background. To deal with this decomposition, i.e., a constrained minimization problem, the augmented Lagrangian multiplier method is employed with the help of the alternating direction minimizing strategy. Extensive experiments on both simulated and real data demonstrate that our method significantly outperforms the state-of-the-art approaches, especially for the cases with dynamic backgrounds and discontinuous movements.**

***Index Terms*—Irregularly moving object detection, lingering object, low-rank modeling, robust principal component analysis (RPCA), total variation regularization.**

I. Introduction

MOVING object detection [1], [2] is one of the most important and challenging tasks in computer vision, which plays a core role for various applications, such as object tracking [3], behavior recognition [4], scene understanding, and augmented reality [5], [6]. The approaches for moving object detection can be roughly divided into two categories: 1) supervised and 2) unsupervised. The supervised approaches required to build either a background model [7]–[12] or a foreground object model [13], [14] by learning from labeled data, which is obviously expensive. In other words, these approaches traditionally make restrictive assumptions on either the background or foreground. In reality, however, both foreground and background may be too complex to model since their appearances vary with the change of illumination and perspective. Therefore, the performance of supervised methods is very likely to degrade or even fail. Different from the supervised moving object detection methods, the unsupervised ones alternatively make use of motion information to directly separate foreground objects from background [15] instead of training background or foreground models. Most of them separate moving objects by detecting and modeling the changes between different frames. Optical flow that computes motion between two adjacent frames is a typical example. However, it is usually sensitive to illumination changes and dynamic backgrounds. Another classic example belonging to this category, named robust principal component analysis (RPCA) [16], [17], assumes that the backgrounds of different frames are linearly correlated and the moving objects appear to be sparse. By decomposing the observed data matrix into a low-rank background matrix and a sparse moving objects matrix, RPCA is able to handle moving object detection problem. Recently, RPCA has been widely studied and used in various computer vision problems [18]–[20] for its simple model [19], [21], [22], sound theory [16], [23], [24], and many efficient algorithms [25], [26]. However, it only can handle indoor and simple outdoor scenarios well, where there only exist few moving objects with (nearly) uniform movement, static background, and no shelter.

Unfortunately, the background is not always static, and the foreground movements are often nonuniform as shown in Fig. 1. On the one hand, difficulties caused by the dynamic background are inevitable in real scenarios. Since most surveillance cameras are mounted at the roadside or in the park,

Manuscript received November 30, 2014; revised March 20, 2015; accepted March 31, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61422213, Grant 61332012, and Grant 61402467, in part by the 100 Talents Programme of the Chinese Academy of Sciences, in part by the Excellent Young Talent Programme through the Institute Information Engineering, Chinese Academy of Sciences, in part by the Foundation for the Young Scholars by the Tianjin University of Commerce under Grant 150113, and in part by the National Training Programs of Innovation and Entrepreneurship for Undergraduates under Grant 201410069040. This paper was recommended by Associate Editor M. Shin. *(Corresponding author: Xiaojie Guo.)*

X. Cao and X. Guo are with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: xj.max.guo@gmail.com).

L. Yang is with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China, and also with the School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China.

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/TCYB.2015.2419737

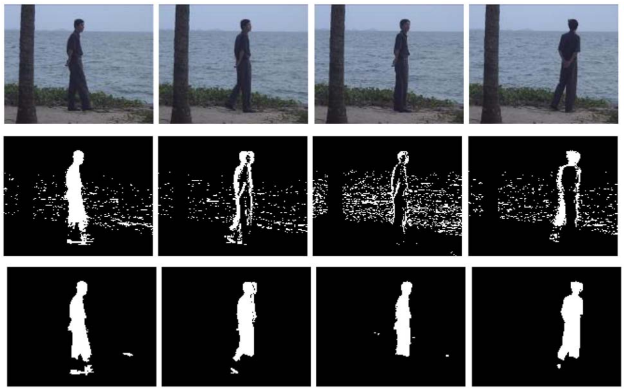

Fig. 1. Difficulties caused by dynamic background and lingering object. First row shows the observed video frames from *Watersurface* sequence, while the second row shows the results obtained by traditional RPCA. The results of our proposed TVRPCA, which separates dynamic background from moving objects and detects lingering objects using the spatial and temporal continuity of foreground, are shown in third row.

dynamic factors, such as fountains, ripples, and shaking leaves, are very common as shown in Fig. 7. Furthermore, moving object detection under bad weather conditions, such as rainfall or *SnowFall* as shown in Fig. 8, is particularly important for security protection, which can also be regarded as dynamic background. On the other hand, discontinuous movements, e.g., lingering objects as shown in Fig. 3, also make detection very difficult. In real life, lingering objects and discontinuous movements are not rare. However, most methods are unable to address this issue. For these two scenarios, most existing motion-based methods are unable to achieve good performance. They usually detect moving foreground as well as dynamic background and are unable to distinguish shelter regions, which decreases the detection precision. Even worse, they also treat the lingering objects as background and only detect their outlines as moving foreground, which seriously degrades performance, especially the recall.

To handle the challenges mentioned above, especially the one caused by lingering objects, we propose a novel unified framework to detect moving foreground objects by separating dynamic background from moving objects and detecting lingering objects using the spatial and temporal continuity of foreground. This is based on the observations that the dynamic backgrounds are typically sparser than the moving foreground objects. In addition, the intrinsic moving foreground objects should be temporally and spatially contiguous, which mathematically satisfies the definition of total variation. Based on these observations, we decompose the part disobeying the low-rank characterization of the video into two parts: 1) the dynamic background and 2) the foreground objects. To the best of our knowledge, we are the first to use total variation to explicitly describe the foreground continuity in foreground/background separation problem. Guyon *et al.* [27], [28] proved that the total variation regularized problem can be solved by iteratively reweighted least squares scheme and use total variation as the weighted mask for the matrix factorization problem.

The contributions of this paper are summarized as follows.

1) First, we present a new problem, i.e., how to detect irregular object movements (like lingering object), and analyze the reasons why most of the classic and contemporary algorithms degrade sharply when they are used in a real detection system. Then we determine what kind of objects people really want to detect, and discover what are their shared properties.
2) Second, we propose a novel unified framework to detect moving foreground objects. Dynamics background can be separated from moving objects using the spatial continuity of foreground, and lingering objects can be detected using the temporal continuity of foreground.
3) Third, we develop an efficient and effective algorithm to solve the total variation regularized RPCA (TVRPCA), which is a constrained minimization problem, by using the augmented Lagrange multiplier (ALM) with alternating direction minimizing (ADM) strategy.
4) Finally, we apply our proposed TVRPCA to the detection under bad weather condition, e.g., rainfall or *SnowFall*, and obtain satisfactory performance.

The rest of this paper is organized as follows. In Section II, we provide an overview of previous work on improving RPCA foreground detection performance. Section III presents our TVRPCA in details, including review of total variation regularization, problem formulation, solving algorithm, and computational complexity analysis. Extensive experiments on synthetic and real datasets are presented in Section IV. Finally, Section V conclude this paper.

II. Related Work

Moving object detection in dynamic scenes is an important and challenging task, since waving trees, spouting *Fountain*, and raindrops are common in real world. There is a lot of work on this topic [29]–[32]. Some of them solve this problem by design dynamic texture extraction algorithm based on different pixel descriptors, such as local dependency histogram descriptor [29], covariance matrix descriptor [31], and local binary pattern descriptor [30]. Others classify pixel based on samples of its neighbor pixels [32]. However, most of them are based on the local properties of pixels but ignore the global properties of dynamic background.

By considering the global low-rank property, RPCA has been widely used in moving object detection. Due to the dynamic background and lingering objects, original RPCA cannot achieve satisfactory performance in complex scenarios. To improve the performance of RPCA on moving object detection, two main methods have been proposed recently. Detecting contiguous outliers in the low-rank representation (DECOLOR) [33], [34] adopts the nonconvex penalty and Markov random field (MRF) to detect outliers, which prefers the regions that are relatively dense and contiguous. As we will see in Section IV, although it can alleviate some of the aforementioned problems, DECOLOR has three main disadvantages that could reduce the detection precision. First, due to its greedy property, it may detect some regions near the moving objects and lose the details of moving objects, e.g., the first row in Fig. 7. Second, when there are shelters in front of the moving object, this method cannot remove the shelters' region as shown in the third row in Fig. 7. Finally, because of computational complexity, only spatial MRF is usually considered, which makes it ignore the temporal relationship between frames. However, as we shown in the following sections, temporal constraints are critical for detection. The second method [35], [36] is based on block-sparsity [37] and a two-pass RPCA process, which first roughly detects the possible foreground regions via performing the first-pass RPCA on a low resolution video, and then carried out a motion saliency estimation by employing dense optical flow. The trajectories moving in some consistent directions are retained to suppress most of the changes from the background. Finally, by decreasing the regularizing parameter of the blocks, it obtains the consistent motions with the help of the second-pass RPCA. In this process, the moving object detection problem is divided into to three steps, and the first-pass RPCA and the dense optical flow can be regarded as the preprocessing of the video. The major shortcoming of this method is the high computational complexity. In addition, another approach, called Grassmannian robust adaptive subspace tracking algorithm (GRASTA) [38], proposes to update the subspace where the background should lie in, and separate the foreground in an online manner, which is designed to be flexible to slow changes of background. Although GRASTA can significantly cut the computational load, its performance would sharply degrade when the subspace is updated improperly, and it does not solve the dynamic background and lingering object problem at all. In conclusion, some of the existing RPCA-based methods directly impose spatial constraints on all moving objects, including foreground and dynamics background, but none of them separate dynamics background from moving objects and take into account the temporal constraints as well as the spatial constraints in an elegant way. In this paper, to accurately model moving object detection problem, we decompose the observed video into three components: 1) low-rank static background; 2) sparse and smooth foreground; and 3) sparser dynamic background, and formulate it as a TVRPCA problem.

Compared with the previous work [39], there are four main differences. First, we present a new problem, i.e., how to detect irregular object movement (like lingering objects), in this paper. Then we investigate the reason why original RPCA fails to detect them in detail and solve this problem, while the previous work only focuses on the problem caused by dynamic background and noise from video capturing process. Second, in this paper, we regard the observed video as being made up of the sum of a low-rank static background, a sparse and smooth foreground and a sparser dynamic background instead of just the foreground and background. This makes it much easier and more accurate to model them. Third, they model moving object detection problem from different viewpoints, which results in different formulations and optimization algorithms. The previous work is from the viewpoint of Bayes, and formulates the moving object detection as a matrix factorization problem, i.e., $\boldsymbol{X} = \boldsymbol{AB}$. However, in this paper, we formulate it as a regularized RPCA problem, i.e., $\boldsymbol{X} = \boldsymbol{A}+\boldsymbol{B}$. Finally, and most importantly, in this paper, the rank of the static background can be automatically determined, while in the previous work it must be predefined by human.

III. Total Variation Regularized Robust PCA

We first provide an overview of total variation regularization and show how it is used to encode the spatial and temporal continuity in Section III-A. Then we introduce the TVRPCA that integrates the temporal and spatial smoothness into the traditional RPCA in Section III-B, and describe the optimization algorithm to solve the TVRPCA in Sections III-C and III-D. Finally, Section III-E presents the complexity analysis.

A. Continuity and Total Variation

To investigate the moving object detection problem, we must first determine what kind of objects people really want to detect. Intuitively, moving foreground to be detected should occupy a certain proportion of continuous region of the screen and exist salient movement. For one thing, although they exist salient movements, some very small objects, e.g., raindrops and snowflakes, are not the interesting thing for people. For another, if the movements are not salient and consistent, we usually do not need to detect them, e.g., ripples and shaking leaves. Combining these two points, moving object to be detected should make intensity change saliently and continuously. Taking the *Watersurface* sequence, in which a man walks by the river as shown in Fig. 1, as an example, what we should detect is the person, while the ripples, whose movements are slight and discontinuous, should be suppressed. So we can impose spatial continuity constraints on detected objects to suppress the slight and discontinuous movements.

In reality, however, it is much more complicated. Please consider a case that a man walks by the river, lingers for a period of time and looks around. Should he be detected in that period of time when his movement is not salient. The answer is YES. However, what about the case that a man only stand there and look around throughout the video. Traditional motion-based approaches treat both cases equally and only detect the outline of the object. To understand the reason for these cases, we plot the intensity changes of fixed space points over time. As shown in Fig. 2, we treat a video clip as a 3-D tensor and cut it into slides along the vertical and horizontal directions. From slide examples shown in Fig. 2(d), we find that the intensities of fixed space points change continuously over time. However, in the results of traditional motion-based methods as shown in Fig. 2(e), this continuity property is destroyed, and there are some gaps and holes in the detection areas. To alleviate this problem, we impose temporal continuity constraints on foreground to be detected. And as shown in Fig. 2(f), the continuity property is preserved in the results of our methods.

Having determined to impose spatial and temporal continuity constraints on moving foreground to be detected, the remaining issue is how to encode these continuity constraints. In mathematics, the derivative can be used to measure the sensitivity to change of a quantity function. For discrete functions, difference operators are the approximation to derivative. A video can be represented by a 3-D tensor $\boldsymbol{F} \in \mathbb{R}^{d_v \times d_h \times d_t}$.

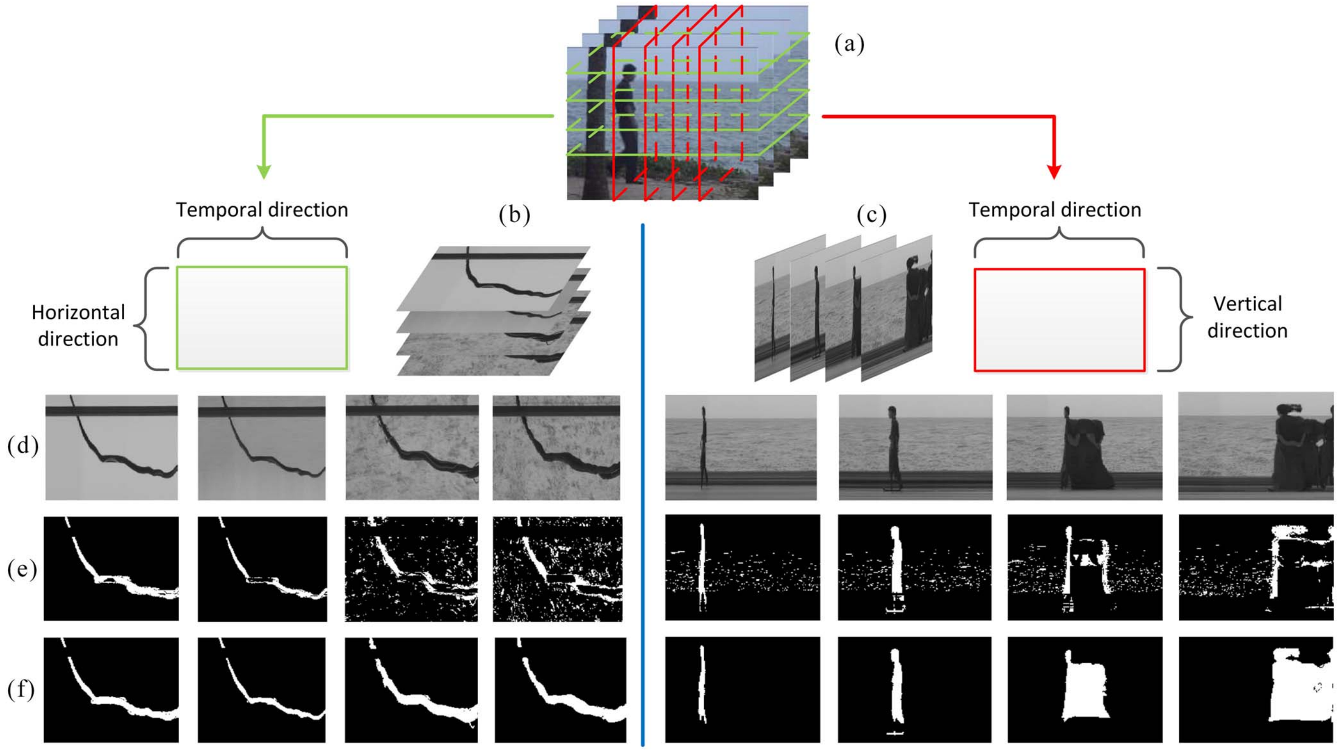

Fig. 2. Temporal continuity of moving object. (a) By stacking the video frames, video clip can be regarded as a 3-D tensor. We cut it into slides along (b) horizontal and (c) vertical directions. (d) Each slide can be seen as a 2-D image, one dimension of which is temporal direction. These images demonstrate the intensity changes of some fixed spatial points over time. We find that these changes are continuous over time. (e) and (f) Results from RPCA and our proposed TVRPCA, respectively. Without considering temporal continuity, the results of RPCA become discontinuous, and only the outline can be correctly detected. By introducing temporal continuity, our approach can overcome this difficulty and preserve the continuity property.

And we use $\boldsymbol{F}(x, y, t)$ to indicate the intensity of position (x, y) at time t, and use

$$\begin{aligned}\boldsymbol{F}_h(x, y, t) &= \boldsymbol{F}(x+1, y, t) - \boldsymbol{F}(x, y, t)\\ \boldsymbol{F}_v(x, y, t) &= \boldsymbol{F}(x, y+1, t) - \boldsymbol{F}(x, y, t)\\ \boldsymbol{F}_t(x, y, t) &= \boldsymbol{F}(x, y, t+1) - \boldsymbol{F}(x, y, t)\end{aligned}$$

to denote three difference operation results of position (x, y) at time t with periodic boundary conditions along the horizontal, vertical, and temporal directions, respectively, For the simplicity of numerical computation, we stack all the entries of $\boldsymbol{F}$ into a column vector $\boldsymbol{f} = \mathbf{vec}(\boldsymbol{F})$, in which $\mathbf{vec}()$ represents the vectorization operator, and use $\boldsymbol{D}_h\boldsymbol{f} = \mathbf{vec}(\boldsymbol{F}_h)$, $\boldsymbol{D}_v\boldsymbol{f} = \mathbf{vec}(\boldsymbol{F}_v)$, and $\boldsymbol{D}_t\boldsymbol{f} = \mathbf{vec}(\boldsymbol{F}_t)$ to represent the vectorizations of the three difference operation results, respectively, in which $\boldsymbol{D}_v$, $\boldsymbol{D}_h$, and $\boldsymbol{D}_t \in \mathbb{R}^{d_v d_h d_t \times d_v d_h d_t}$. And use $\boldsymbol{D}\boldsymbol{f} = [\boldsymbol{D}_h\boldsymbol{f}^T, \boldsymbol{D}_v\boldsymbol{f}^T, \boldsymbol{D}_t\boldsymbol{f}^T]^T$ to denote the concatenated difference operation, in which $\boldsymbol{D} \in \mathbb{R}^{3d_v d_h d_t \times d_v d_h d_t} = [\boldsymbol{D}_h^T, \boldsymbol{D}_v^T, \boldsymbol{D}_t^T]^T$. Since the ith element in $\boldsymbol{D}_h\boldsymbol{f}$, $\boldsymbol{D}_v\boldsymbol{f}$, and $\boldsymbol{D}_t\boldsymbol{f}$, i.e., $[\boldsymbol{D}_h\boldsymbol{f}]_i$, $[\boldsymbol{D}_v\boldsymbol{f}]_i$, and $[\boldsymbol{D}_t\boldsymbol{f}]_i$, describe the intensity changes of ith point in $\boldsymbol{f}$ along the horizontal, vertical, and temporal directions, we can use any vector norm of $[[\boldsymbol{D}_h\boldsymbol{f}]_i, [\boldsymbol{D}_v\boldsymbol{f}]_i, [\boldsymbol{D}_t\boldsymbol{f}]_i]^T$ to quantize the changes of intensity. Two widely used vector norms are ℓ_1 and ℓ_2 norms. By summing up all vector norms of different points, we obtain the definition of anisotropic total variation norm as

$$\|\boldsymbol{F}\|_{TV_1} = \sum_i \left(\left|[\boldsymbol{D}_h\boldsymbol{f}]_i\right| + \left|[\boldsymbol{D}_v\boldsymbol{f}]_i\right| + \left|[\boldsymbol{D}_t\boldsymbol{f}]_i\right|\right) \tag{1}$$

and the isotropic total variation norm as

$$\|\boldsymbol{F}\|_{TV_2} = \sum_i \sqrt{[\boldsymbol{D}_h\boldsymbol{f}]_i^2 + [\boldsymbol{D}_v\boldsymbol{f}]_i^2 + [\boldsymbol{D}_t\boldsymbol{f}]_i^2} \tag{2}$$

which are the ℓ_1 and $\ell_{2,1}$ norms of $[\boldsymbol{D}_v\boldsymbol{f}, \boldsymbol{D}_h\boldsymbol{f}, \boldsymbol{D}_t\boldsymbol{f}]^T$. By a slight abuse of notations, we use $\|\boldsymbol{D}\boldsymbol{f}\|_{2,1}$ to represent the isotropic total variation of $\boldsymbol{F}$. Total variation regularization have been widely used in image and video denoising [40]–[42] for its superior performance on suppressing discontinuous changes which are regarded as noises in image processing. And we adopt it to suppress the intensity changes caused by dynamic background and fill up the gaps caused by lingering objects.

B. Problem Formulation

Suppose we are given an image sequence including d_t frames, then stack the vectorized frames as columns of a matrix $\boldsymbol{O} \in \mathbb{R}^{d_v d_h \times d_t}$, where d_v and d_h are the height and width of the frames, respectively. In ideal cases, the frames contain the static background component $\boldsymbol{B} \in \mathbb{R}^{d_v d_h \times d_t}$ and the residual $\boldsymbol{M} \in \mathbb{R}^{d_v d_h \times d_t}$. The observed matrix $\boldsymbol{O}$ may be decomposed as $\boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M}$. Due to the high correlation between the stationary backgrounds of frames, $\boldsymbol{B}$ has low rank. And $\boldsymbol{M}$ represents, for example, cars and pedestrians that usually occupy only a fraction of the image pixels and hence can be treated as sparse errors. Both $\boldsymbol{B}$ and $\boldsymbol{M}$ are of arbitrary magnitudes. Detecting the moving objects can be achieved through minimizing the following problem:

$$\min_{\boldsymbol{B},\boldsymbol{M}} \operatorname{rank}(\boldsymbol{B}) + \lambda\|\boldsymbol{M}\|_0, \qquad \text{s.t. } \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M} \tag{3}$$

where $\|\cdot\|_0$ denotes the ℓ^0 norm and λ is the coefficient controlling the weight of the sparse matrix $\boldsymbol{M}$. This problem is known as the RPCA.

The decomposition above works well under the assumption that the background is stationary, but degrades sharply when it is used in realistic situations. This is mostly due to the fact that, in real-world scenarios, the background is very likely to contain changes, i.e., dynamic factors, such as ripples on rivers, fountains and swaying branches of trees. Sudden and gradual illumination changes are another two common phenomenons in reality. Thus, it is very difficult to model the background well only relying on imposing low-rank constraint on background. In other words, the foreground would be detected inaccurately. Fortunately, the foreground has a high possibility to move smoothly and appear coherently in the frames. Based on this observation, the nonzero elements in $\boldsymbol{M}$ consists of two components, i.e., the foreground and the dynamic background. As a result, we further separate $\boldsymbol{M}$ of (3) into two terms $\boldsymbol{F} \in \mathbb{R}^{d_v d_h \times d_t}$ and $\boldsymbol{E} \in \mathbb{R}^{d_v d_h \times d_t}$, where $\boldsymbol{F}$ corresponds to the intrinsic foreground and $\boldsymbol{E}$ the dynamic background component. Thereby, we can naturally reformulate the problem as

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{M},\boldsymbol{F},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{B}) + \lambda_1\|\boldsymbol{M}\|_0 + \lambda_2\|\boldsymbol{E}\|_0 + \lambda_3\Psi(\boldsymbol{F}) \\ &\text{s.t. } \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M}, \qquad \boldsymbol{M} = \boldsymbol{F} + \boldsymbol{E} \end{aligned} \tag{4}$$

where λ_1, λ_2, and λ_3 are the weights for balancing the corresponding terms in (4). $\Psi(\cdot)$ is the function of regularizing the foreground to be spatially coherent and temporally smooth, which is done by total variation norm in this paper. So the final formulation of the problem is

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{M},\boldsymbol{F},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{B}) + \lambda_1\|\boldsymbol{M}\|_0 + \lambda_2\|\boldsymbol{E}\|_0 + \lambda_3\|\boldsymbol{F}\|_{TV} \\ &\text{s.t. } \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M}, \qquad \boldsymbol{M} = \boldsymbol{F} + \boldsymbol{E} \end{aligned} \tag{5}$$

where $\|\boldsymbol{F}\|_{TV}$ is the total variation norm as defined in (1) and (2).

Hence, we rewrite (5) in the following shape:

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{M},\boldsymbol{F},\boldsymbol{E}} \operatorname{rank}(\boldsymbol{B}) + \lambda_1\|\boldsymbol{M}\|_0 + \lambda_2\|\boldsymbol{E}\|_0 + \lambda_3\|\boldsymbol{Df}\|_q \\ &\text{s.t. } \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M}, \qquad \boldsymbol{M} = \boldsymbol{F} + \boldsymbol{E} \end{aligned} \tag{6}$$

where q can be either $\{1\}$ or $\{2, 1\}$ for representing ℓ_1 and $\ell_{2,1}$ norms, respectively. In this paper, we call the problem expressed in (6) the TVRPCA.

C. Solution of TVRPCA

In this section, we detail our proposed algorithm for solving the TVRPCA problem. Although the total variation has two different definitions, we do not distinguish them until necessarily. The objective function (6) is nonconvex due to the nonconvexity of the rank function and the ℓ_0 norm. It is NP-hard and hard to approximate. Alternatively, minimizing the natural convex surrogate for the objective function (6) can be employed to accomplish the task, which replaces $\operatorname{rank}(\cdot)$ and the ℓ_0 norm with the nuclear norm and the ℓ_1 norm, respectively. By putting everything together, the optimization problem turns out to be like

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{M},\boldsymbol{F},\boldsymbol{E}} \|\boldsymbol{B}\|_* + \lambda_1\|\boldsymbol{M}\|_1 + \lambda_2\|\boldsymbol{E}\|_1 + \lambda_3\|\boldsymbol{Df}\|_q \\ &\text{s.t. } \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{M}, \quad \boldsymbol{M} = \boldsymbol{F} + \boldsymbol{E}. \end{aligned} \tag{7}$$

For the optimization problem (7), the ALM with ADM strategy is an efficient and effective solver [25]. ALM with ADM is widely used to solve multivariable convex optimization problem as such ℓ_1-norm problem and low-rank problem. It minimizes dual form of original constrained optimization problem over one variable with others fixed at a time, and repeats this process with increasing positive penalty scalar until it converges.

The augmented Lagrangian function of (7) is given by

$$\begin{aligned} &\mathcal{L}_\mu(\boldsymbol{B},\boldsymbol{M},\boldsymbol{E},\boldsymbol{F},\boldsymbol{X},\boldsymbol{Y}) \\ &\quad = \|\boldsymbol{B}\|_* + \lambda_1\|\boldsymbol{M}\|_1 + \lambda_2\|\boldsymbol{E}\|_1 + \lambda_3\|\boldsymbol{Df}\|_q \\ &\qquad + \frac{\mu}{2}\|\boldsymbol{O}-\boldsymbol{B}-\boldsymbol{M}\|_F^2 + <\boldsymbol{X}, \boldsymbol{O}-\boldsymbol{B}-\boldsymbol{M}> \\ &\qquad + \frac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{F}-\boldsymbol{E}\|_F^2 + <\boldsymbol{Y}, \boldsymbol{M}-\boldsymbol{F}-\boldsymbol{E}> \end{aligned} \tag{8}$$

where $\boldsymbol{X} \in \mathbb{R}^{d_v d_h \times d_t}$ and $\boldsymbol{Y} \in \mathbb{R}^{d_v d_h \times d_t}$ are the Lagrange multiplier matrices, μ is a positive penalty scalar, $<\cdot,\cdot>$ denotes the matrix inner product, and $\|\cdot\|_F$ represents the Frobenius norm. Besides the Lagrange multipliers, there are four variables, i.e., $\boldsymbol{B}$, $\boldsymbol{M}$, $\boldsymbol{F}$, and $\boldsymbol{E}$. It is difficult to simultaneously optimize them. So we approximately solve it in the manner of minimizing one variable with others fixed at a time (ADM). To optimize (7), we sequentially optimize its dual form (8) over $\boldsymbol{B}$, $\boldsymbol{M}$, $\boldsymbol{F}$, and $\boldsymbol{E}$ with increasing μ. The details of ADM iteration is as follows.

Updating $\boldsymbol{B}$ with the other terms fixed

$$\boldsymbol{B}^{k+1} = \operatorname{argmin} \mathcal{L}_\mu\left(\boldsymbol{B}, \boldsymbol{M}^k, \boldsymbol{E}^k, \boldsymbol{F}^k, \boldsymbol{X}^k, \boldsymbol{Y}^k\right).$$

The solution of updating $\boldsymbol{B}$ is

$$\begin{aligned} (\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) &= \operatorname{svd}\left(\boldsymbol{O} - \boldsymbol{M}^k + \frac{1}{\mu^k}\boldsymbol{X}^k\right) \\ \boldsymbol{B}^{k+1} &= \boldsymbol{U}\mathcal{S}_{\frac{1}{\mu^k}}(\boldsymbol{\Sigma})\boldsymbol{V}^T \end{aligned} \tag{9}$$

where $\boldsymbol{U\Sigma V^T}$ is the singular value decomposition (SVD) of $(\boldsymbol{O}-\boldsymbol{M}^k+1/\mu^k\boldsymbol{X}^k)$, $\{\mu^k\}$ is a monotonically increasing positive sequence, and $\mathcal{S}[\cdot]$ represents the shrinkage operator, the definition of which on scalars is: $\mathcal{S}_{\varepsilon>0}(\cdot) = \operatorname{sgn}(x)\max(|x| - \varepsilon, 0)$. The extension of the shrinkage operator to vectors and matrices is applying it element-wisely.

Updating $\boldsymbol{M}$ with the other terms fixed

$$\boldsymbol{M}^{k+1} = \operatorname{argmin} \mathcal{L}_\mu\left(\boldsymbol{B}^{k+1}, \boldsymbol{M}, \boldsymbol{E}^k, \boldsymbol{F}^k, \boldsymbol{X}^k, \boldsymbol{Y}^k\right).$$

The solution of updating $\boldsymbol{M}$ is

$$\boldsymbol{M}^{k+1} = \mathcal{S}_{\frac{\lambda_1}{2\mu^k}}\left[\frac{\boldsymbol{O}-\boldsymbol{B}^{k+1}+\boldsymbol{E}^k+\boldsymbol{F}^k}{2} + \frac{\boldsymbol{X}^k-\boldsymbol{Y}^k}{2\mu^k}\right]. \tag{10}$$

Updating $\boldsymbol{E}$ with the other terms fixed

$$\boldsymbol{E}^{k+1} = \operatorname{argmin} \mathcal{L}_\mu\left(\boldsymbol{B}^{k+1}, \boldsymbol{M}^{k+1}, \boldsymbol{E}, \boldsymbol{F}^k, \boldsymbol{X}^k, \boldsymbol{Y}^k\right).$$

Algorithm 1: TV-RPCA

```
Input: $\lambda_1 > 0$, $\lambda_2 > 0$, $\lambda_3 > 0$, and the observation
        matrix $\boldsymbol{O}$.
Initialization:
$\boldsymbol{B}^0 = \boldsymbol{M}^0 = \boldsymbol{E}^0 = \boldsymbol{F}^0 = \boldsymbol{X}^0 = \boldsymbol{Y}^0 = \boldsymbol{0} \in \mathbb{R}^{d_v d_h \times d_t}$,
$\mu^0 > 0$, $\rho > 1$ and $k = 0$.
while not converged do
    Update $\boldsymbol{B}^{k+1}$ via (9);
    Update $\boldsymbol{M}^{k+1}$ via (10);
    Update $\boldsymbol{E}^{k+1}$ via (11);
    Update $\boldsymbol{F}^{k+1}$ via Algorithm 2;
    Update multipliers via (13);
    $\mu^{k+1} = \mu^k \rho$; $k = k + 1$;
end
Output: Optimal solution $(\boldsymbol{B}^k, \boldsymbol{F}^k, \boldsymbol{E}^k, \boldsymbol{M}^k)$
```

The solution of updating $\boldsymbol{E}$ is

$$\boldsymbol{E}^{k+1} = \mathcal{S}_{\frac{\lambda_2}{\mu^k}}\left[\boldsymbol{M}^{k+1} - \boldsymbol{F}^k + \frac{1}{\mu^k}\boldsymbol{Y}^k\right]. \tag{11}$$

Updating $\boldsymbol{F}$ with the other terms fixed

$$\begin{aligned}\boldsymbol{F}^{k+1} = \operatorname{argmin} \lambda_3 \|\boldsymbol{Df}\|_q &+ <\boldsymbol{Y}^k, \boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}> \\ &+ \frac{\mu}{2}\left\|\boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}\right\|_F^2.\end{aligned} \tag{12}$$

Since solving this subproblem involves an inner loop, we will discuss the inner loop later.

Updating multipliers with the other terms fixed

$$\begin{aligned}\boldsymbol{X}^{k+1} &= \boldsymbol{X}^k + \mu^k\left(\boldsymbol{O} - \boldsymbol{B}^{k+1} - \boldsymbol{M}^{k+1}\right) \\ \boldsymbol{Y}^{k+1} &= \boldsymbol{Y}^k + \mu^k\left(\boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}^{k+1}\right).\end{aligned} \tag{13}$$

The entire algorithm of solving the problem (7) has been summarized in Algorithm 1. The algorithm terminates when $\|\boldsymbol{O} - \boldsymbol{B}^k - \boldsymbol{F}^k - \boldsymbol{E}^k\|_F^2 \leq \delta \|\boldsymbol{O}\|_F^2$ with $\delta = 10^{-7}$, or the maximal number of iterations is reached.

D. Solver of the $\boldsymbol{F}$ Subproblem

To solve the subproblem (12), we introduce an auxiliary variable $\boldsymbol{K} \in \mathbb{R}^{3d_v d_h d_t \times 1}$ to replace $\boldsymbol{Df}$. Accordingly, $\boldsymbol{K} = \boldsymbol{Df}$ is as an additional constraint. Thus, we have

$$\begin{aligned}\boldsymbol{F}^{k+1} = \operatorname{argmin} \lambda_3 \|\boldsymbol{K}\|_q &+ <\boldsymbol{Y}^k, \boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}> \\ &+ \frac{\mu}{2}\|\boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}\|_F^2, \quad \text{s.t.} \quad \boldsymbol{K} = \boldsymbol{Df}.\end{aligned}$$

For brevity, we denote $\boldsymbol{A} \doteq \boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} - \boldsymbol{F}$, and omit the superscript $k+1$ of $\boldsymbol{F}$ in this subproblem, the augmented Lagrangian of which is

$$\begin{aligned}\mathcal{L}_\gamma(\boldsymbol{F}, \boldsymbol{K}, \boldsymbol{Z}) = \operatorname{argmin} \lambda_3 \|\boldsymbol{K}\|_q &+ <\boldsymbol{Y}^k, \boldsymbol{A}> + \frac{\mu}{2}\|\boldsymbol{A}\|_F^2 \\ &+ <\boldsymbol{Z}, \boldsymbol{K} - \boldsymbol{Df}> + \frac{\gamma}{2}\|\boldsymbol{K} - \boldsymbol{Df}\|_F^2\end{aligned}$$

where γ performs the same with μ and $\boldsymbol{Z}$ is the multiplier.

Updating $\boldsymbol{F}^{t+1}$ with the other terms fixed

$$\boldsymbol{F}^{t+1} = \operatorname{argmin} \mathcal{L}_\gamma\left(\boldsymbol{F}, \boldsymbol{K}^t, \boldsymbol{Z}^t\right).$$

Algorithm 2: Solver of the F Subproblem

```
Input: $\lambda_3 > 0$, $\boldsymbol{M}^{k+1}$, $\boldsymbol{E}^{k+1}$ and $\boldsymbol{Y}^k$.
Initialization: $\boldsymbol{Z}^0 = \boldsymbol{K}^0 = \boldsymbol{0} \in \mathbb{R}^{3d_v d_h d_t \times 1}$, $t = 0$,
Compute $|\mathcal{F}(\boldsymbol{D}_v)|^2$, $|\mathcal{F}(\boldsymbol{D}_h)|^2$, $|\mathcal{F}(\boldsymbol{D}_t)|^2$, and $\gamma^0 > 0$,
$\rho > 1$.
while not converged do
    Update $\boldsymbol{F}_{t+1}^{k+1}$ via (15);
    Update $\boldsymbol{K}_{t+1}^{k+1}$ via either (16) for the anisotropic total
    variation or (17) for the isotropic one;
    Update multipliers via (18);
    Update $\gamma^{t+1}$ via (19); $t = t + 1$;
end
Output: Optimal solution $(\boldsymbol{F}^t, \boldsymbol{K}^t)$
```

By considering its normal equation, we have

$$\left(\mu^k \boldsymbol{I} + \gamma^t \boldsymbol{D}^T \boldsymbol{D}\right)\boldsymbol{f} = \boldsymbol{Q}; \qquad \boldsymbol{F}^{t+1} = \text{reshape}(\boldsymbol{f}) \tag{14}$$

where $\boldsymbol{Q} = \mu^k \operatorname{vec}(\boldsymbol{M}^{k+1} - \boldsymbol{E}^{k+1} + \boldsymbol{Y}^k/\mu^k) + \gamma^t(\boldsymbol{D}^T\boldsymbol{K}^k + (\boldsymbol{D}^T\boldsymbol{Z}^k/\mu^k))$ and reshape$(\cdot)$ is to reshape the vector $\boldsymbol{f}$ back into its 3-D shape. Traditionally, the optimal estimation of $\boldsymbol{f}^{t+1}$ can be simply obtained via computing the Moore–Penrose pseudo-inverse of matrix $(\mu^k \boldsymbol{I} + \gamma^t \boldsymbol{D}^T \boldsymbol{D})$. However, due to the size of the matrix, the Moore–Penrose pseudo-inverse is computationally expensive. Thanks to the block-circulant structure of the matrix, it can be diagonalized by the 3-D-DFT matrix [43]. Therefore, $\boldsymbol{f}^{t+1}$ can be obtained exactly by

$$\mathcal{F}^{-1}\left(\frac{\mathcal{F}(\boldsymbol{Q})}{\mu^k \boldsymbol{1} + \gamma^t\left(|\mathcal{F}(\boldsymbol{D}_h)|^2 + |\mathcal{F}(\boldsymbol{D}_v)|^2 + |\mathcal{F}(\boldsymbol{D}_t)|^2\right)}\right) \tag{15}$$

where $\mathcal{F}(\cdot)$ denotes the 3-D Fourier transform operator, $|\cdot|^2$ is the element-wise square and the division also performs element-wisely. Note that the denominator in the equation can be precalculated outside the outer loop.

Updating $\boldsymbol{K}^{t+1}$ with the other terms fixed

$$\boldsymbol{K}^{t+1} = \operatorname{argmin} \mathcal{L}_\gamma\left(\boldsymbol{F}^{t+1}, \boldsymbol{K}, \boldsymbol{Z}^t\right).$$

For $q = \{1\}$ (anisotropic), the solution of updating $\boldsymbol{K}$ is

$$\boldsymbol{K}^{t+1} = \mathcal{S}_{\frac{\lambda_3}{\gamma^t}}\left[\boldsymbol{Df}^{t+1} - \frac{1}{\gamma^t}\boldsymbol{Z}^t\right]. \tag{16}$$

Before giving the solution of $\boldsymbol{K}$ when $q = \{2, 1\}$ (isotropic), we denote $\boldsymbol{p}_h = \boldsymbol{D}_h \boldsymbol{f} - (\boldsymbol{Z}_h^t/\gamma^t)$, $[\boldsymbol{K}_h^T, \boldsymbol{K}_v^T, \boldsymbol{K}_t^T]^T = \boldsymbol{K}$, and $[\boldsymbol{Z}_h^T, \boldsymbol{Z}_v^T, \boldsymbol{Z}_t^T]^T = \boldsymbol{Z}$. The definitions for $\boldsymbol{p}_v$ and $\boldsymbol{p}_t$ are analogous. $\boldsymbol{K}_h$ can be efficiently computed by

$$\begin{aligned}\boldsymbol{K}_h^{t+1} &= \max\left(\boldsymbol{p} - \frac{\lambda_3}{\gamma^t}, 0\right) \cdot \frac{\boldsymbol{p}_h}{\boldsymbol{p}} \\ \boldsymbol{p} &= \max\left(\sqrt{|\boldsymbol{p}_h|^2 + |\boldsymbol{p}_v|^2 + |\boldsymbol{p}_t|^2}, \epsilon\right).\end{aligned} \tag{17}$$

The operations in (17) are component-wise and ϵ is a small positive constant. Besides, the multiplier is updated via

$$\boldsymbol{Z}^{t+1} = \boldsymbol{Z}^t + \gamma^t\left(\boldsymbol{K}^{t+1} - \boldsymbol{Df}^{t+1}\right) \tag{18}$$

TABLE I
STATE-OF-THE-ART METHODS

Method	Description
RPCA[16]	Low-rank matrix decomposition without considering spatial and temporal smoothness
DECOLOR[33]	Low-rank matrix decomposition with L0 error norm instead of L1 error norm
PRMF[45]	Probabilistic matrix factorization with Laplace error and its online EM algorithm
GRASTA[38]	Low-rank subspace learning and its online algorithm
TVRPCA	Separating foreground into smooth component and sparse component using total variation regularization

in which γ^{t+1} is updated as suggested in [41]

$$\gamma^{t+1} = \begin{cases} \rho\gamma^t \text{ if } \left\|\boldsymbol{K}^{t+1} - \boldsymbol{D}\boldsymbol{f}^{t+1}\right\|_2 \geq \alpha\left\|\boldsymbol{K}^t - \boldsymbol{D}\boldsymbol{f}^t\right\|_2 \\ \gamma^t \quad \text{otherwise.} \end{cases} \tag{19}$$

The algorithm of the $\boldsymbol{F}$ subproblem is summarized in Algorithm 2. The stop criterion of Algorithm 2 is similar to those of Algorithm 1.

E. Complexity Analysis

In this section, we analyze the computational complexity of our proposed TVRPCA. For simplicity, we set $m = d_v d_h$ and $n = d_t$. From Algorithm 1, each outer iteration involves five updating rules with respect to $\boldsymbol{B}$, $\boldsymbol{M}$, $\boldsymbol{E}$, $\boldsymbol{F}$, and multipliers, respectively. Updating $\boldsymbol{B}$ needs to first compute the SVD of a $m \times n$ matrix, which requires $4m^2n + 8mn^2 + 9n^3$ floating point multiplications [45], and then multiplies the shrank singular value matrix with two singular vectors matrices as shown in (9), which costs $(m + n)r^2$ floating point operations, where $r \leq \min(m, n)$ is the rank of original matrix. To sum up, updating $\boldsymbol{B}$ needs $O(m^2n + mn^2 + n^3)$ floating operations. Updating $\boldsymbol{M}$, $\boldsymbol{E}$, and multipliers only requires element-wise addition and shrinkage operations of $m \times n$ matrices, say $O(mn)$. As shown in Algorithm 2, for $\boldsymbol{F}$ subproblem, it iteratively updates $\boldsymbol{f} \in \mathbb{R}^{mn\times 1}$ and $\boldsymbol{K} \in \mathbb{R}^{3mn\times 1}$. At each iteration of updating $\boldsymbol{f}$, the main computation is four FFTs (including three FFTs and one inverse fast Fourier transform), each is with $O(mn\log(mn))$ as shown in [43]. And updating $\boldsymbol{K}$ only requires $O(mn)$ element-wise shrinkage and addition operations. In summary, updating $\boldsymbol{F}$ requires $O(t(mn\log(mn)))$ floating operation where t is the inner iteration number and we fix it to ten in all our experiments. In conclusion, each outer iteration of our proposed algorithm requires $O(m^2n + mn^2 + n^3 + 3mn + 20(mn\log(mn))) = O(m^2n + mn^2 + n^3)$ floating point operations which is the same as the original RPCA.

IV. EXPERIMENTS

In this section, we conduct several experiments on both synthetic (Stuttgart artificial background subtraction (SABS) dataset) and real (perception test image sequences and change detection dataset) data. From different perspectives, we adopt two evaluation criteria on these datasets. In perception test image sequences, which is one of the most popular foreground detection benchmarks, we adopt receiver operating characteristic (ROC) curve to measure the performance of different methods. The definitions of recall and precision are as follows:

$$\text{recall} = \frac{\#\text{correctly classified foreground pixels}}{\#\text{foreground pixels in ground truth}} \tag{20}$$

$$\text{precision} = \frac{\#\text{correctly classified foreground pixels}}{\#\text{pixels classified as foreground}}. \tag{21}$$

To analyze the performance of different methods in details on SABS and change detection datasets, the F-measure is employed

$$F - \text{measure} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}. \tag{22}$$

The F-measure balances the recall and precision and gives an overall quantitative evaluation.

There are four state-of-the-art methods, including RPCA [16], DECOLOR [33], probabilistic robust matrix factorization (PRMF) [44], GRASTA [38], involved into the comparison as shown in Table I, and the codes of which are all downloaded from the authors' websites. As been proved by [16], RPCA can correctly separate the low rank and sparse components by setting the parameter λ as $1/\sqrt{mn}$, in which m and n are the width and height of every single video frame, respectively. Although the efficiency can be improved by setting the sampling ratio less than 1, it may also degrade the performance of GRASTA. To make the comparison as fair as possible, we use all the observations to process, say the sampling ratio is 1. For the algorithms participate in the comparison, the parameters are all set as default. Unless otherwise stated, the parameters of Algorithm 1 are fixed throughout the experiments empirically: $\lambda_1 = 0.4/\sqrt{mn}$ and $\lambda_2 = 2/\sqrt{mn}$. As for λ_3 that controls the weight of the total variation term, we set $\lambda_3 = 1/\sqrt{mn}$ for cases with dynamic background, while $\lambda_3 = 0.1/\sqrt{mn}$ for cases without dynamic background. In addition, the isotropic TV is adopted throughout the experiments, that is to say, $q = \{1, 2\}$.

A. I2R Dataset

I2R dataset[1] [9] contains nine video sequences, which has a variety of scenarios including static background (*Bootstrap* and lobby), dynamic background (*Campus* and *Fountain*), and slow movement with dynamic background (*Curtain* and *Watersurface*) as shown in Fig. 4. It is widely used as the benchmark in the tasks of tracking and foreground and background separation. The number of frames in each video ranges from 523 to 3584. Along with the video data, each video provides 20 frames foreground ground truth for evaluating the performance. We give the qualitative and quantitative results of our methods on this dataset, compared with the state-of-the-art.

First, to demonstrate the superiority of our method, Fig. 3 provides the visual difference between TVRPCA, RPCA,

[1] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

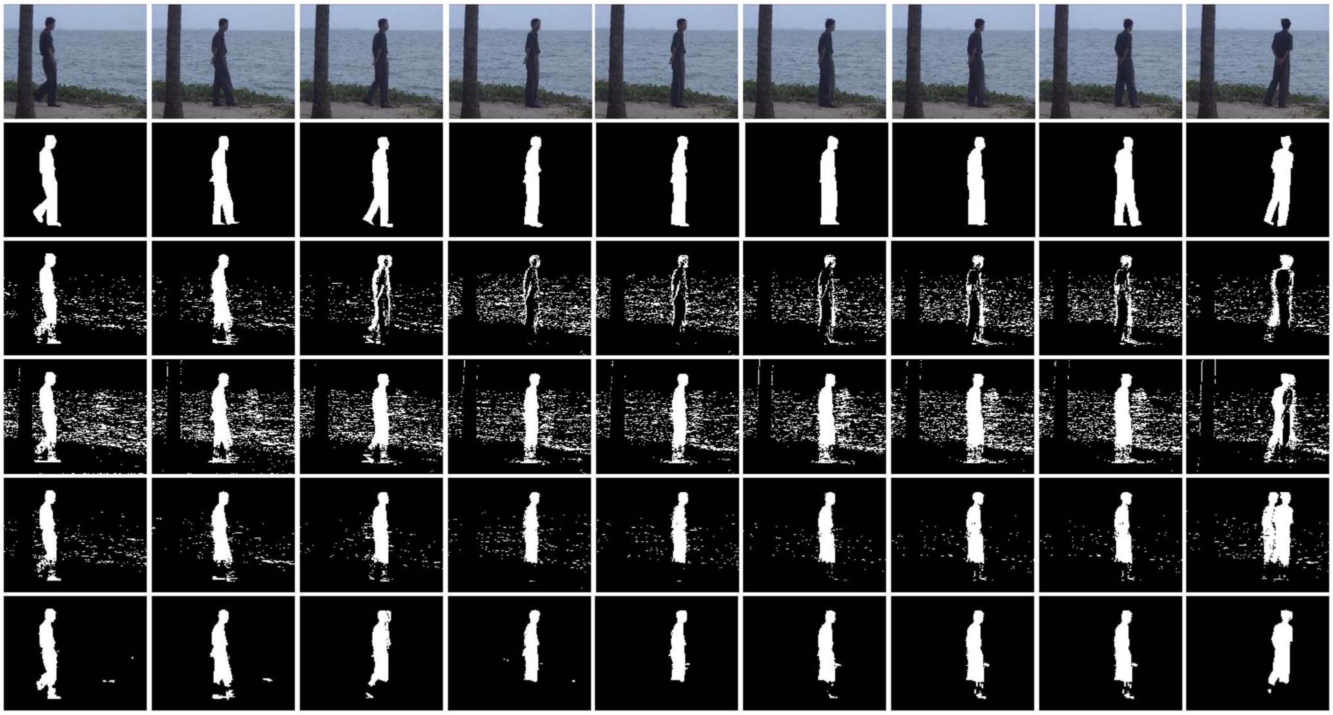

Fig. 3. Visual results of RPCA, PRMF, GRASTA, and TVRPCA on *Watersurface* sequence which contains lingering objects. First row is the original frames in the sequence. Second row shows the ground truth. Third row displays the results obtained by RPCA, which considers slowly moving objects as background while the ripples as foreground. The results of PRMF (fourth row) and GRASTA (fifth row) achieve better results than RPCA, but still degrade in the last two columns. As shown in sixth row, by considering the foreground to be contiguous in both space and time, TVRPCA can not only remove the dynamic backgrounds, but also recover the vast majority of foreground.

Fig. 4. Representative example frames of perception test image sequences dataset. From top, left to right: *Hall*, *Campus*, *Watersurface*, *Bootstrap*, *Fountain*, and *Curtain*. The order is the same as that shown in Fig. 5.

PRMF, and GRASTA, on the sequence of *Watersurface*. The dynamic backgrounds, such as water ripples, are regarded as foreground by all the other methods except for TVRPCA. At the same time as shown in the last two columns of Fig. 3, all the other methods are only able to detect only a small part of the foreground, i.e., the outline of the moving object, because the slowly moving objects are treated as the background by only considering the motion information. Thanks to the temporal and spatial continuity constraint on the foreground objects, our method can detect the vast majority of them. As shown in the last row of Fig. 2, different from original RPCA, TVRPCA preserves the continuity property along the temporal direction.

Second, to verify the efficacy of the proposed TVRPCA, ROC, and F-measure are employed as the quantitative metric to clearly show the performance difference. As can be seen in Fig. 5, the results in the first column correspond to two static background sequences without lingering object, i.e., shopping mall and *Bootstrap*, which reveal that TVRPCA performs similarly to the other methods with slight improvements. Since we take into account the spatial and temporal continuity, our method can alleviate the problem caused by the similarity between foreground and background to a certain but limited extent. In the second column, the results indicate that TVRPCA significantly outperforms the other methods for the sequences containing dynamic background. By suppressing the motion caused by dynamic background, TVRPCA achieves a high true positive with a small false positive, which significantly increases the area under the curve. As can be seen in the third column, the superiority of TVRPCA is quite obvious for the sequences with slowly moving foregrounds as well as dynamic backgrounds during a period of time. Most motion-base methods treat the interior region of the moving object as background, which substantially reduces the true positive, while our methods correctly detect these regions thanks to the temporal continuity constraint. To provide an overall comparison, we also compare TVRPCA with many state-of-the-art algorithms based on the F-measure. As shown in Table II, TVRPCA outperforms most of them.

Finally, we investigate the robustness of TVRPCA to noise. Here, we consider four kinds of noise: 1) Gaussian noise; 2) salt and pepper noise; 3) speckle noise; and

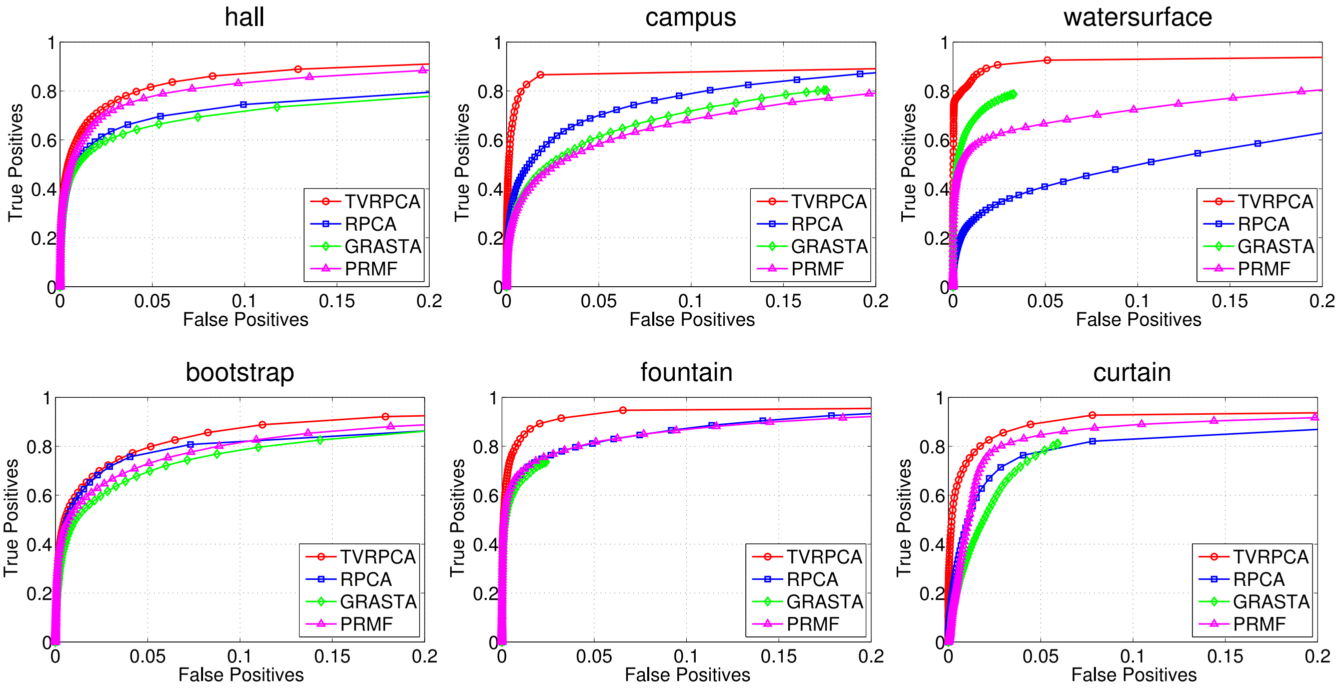

Fig. 5. Quantitative performance comparison between TVRPCA, RPCA, GRASTA, and PRMF. First column contains the sequences with static background. Second column contains the sequences with only dynamic background. Third column contains the most difficult sequences that have slow foreground movements during a period of time as well as dynamic background.

TABLE II
PERFORMANCE COMPARISON ON THE SEQUENCES OF I2R DATASET

Method	WaterSurface	Fountain	Hall	Campus	Lobby	Escalator	Bootstrap
GMM[12]	0.79	0.69	0.33	0.54	0.65	0.14	0.38
SOBS[47]	0.82	0.66	0.59	0.67	0.65	0.58	0.60
RPCA[16]	0.41	0.57	0.59	0.72	0.70	0.65	0.66
GRASTA[38]	0.73	0.38	0.58	0.71	0.56	0.47	0.61
SPDM[48]	0.79	0.77	0.64	**0.81**	0.58	0.65	-
SAC[39]	0.87	0.75	0.67	0.74	**0.80**	0.64	0.68
TVRPCA	**0.88**	**0.80**	**0.69**	0.77	0.75	**0.66**	**0.69**

4) Poisson noise. Gaussian noise (additive white noise) is the most common noise that arises during acquisition and transmission due to poor illumination and high temperature. Different from Gaussian noise, salt and pepper noise completely changes the pixel value to white or black, which makes the information in the pixel completely ruined. Speckle noise is the multiplicative noise to the image. Poisson noise is generated from image itself instead of an artificial noise. For the noises, except for Poisson noise, we set three different variance values to represent different noise levels. As shown in Fig. 6, TVRPCA is robust to these noises, even under a high noisy level, and the detection results form noisy videos are very similar to that from the clean video.

B. SABS Dataset

The SABS dataset[2] [48] is an artificial dataset for pixel-wisely evaluating the performance of background subtraction for surveillance videos. To demonstrate the performance improvement, we test the basic and dynamic background image sequences since our algorithm achieves limited performance improvement on videos, which are peripheral to our concern here as shown in the last section. To quantitatively show the performance difference between our method and DECOLOR [33] that directly generates foreground mask without tuning thresholds, we employ three metrics including recall, precision, and F-measure.

The results are shown in the first two rows in Fig. 7 and Table III. Notice that the case of SABS takes the whole picture of each frame into account, i.e., basic image sequence, while SABS-R only considers the region around the tree, i.e., rectangular region, as suggested in the default SABS experiment setup for evaluating the performance on dynamic background region. We can find that RPCA detects the inconsistent background movements caused by the moving leaves, which decreases the precision to 0.56. And due to the greedy property of DECOLOR, it may wrongly label the nearby background as the moving object regions by considering their

[2] http://www.vis.uni-stuttgart.de/forschung/informationsvisualisierung-und-visual-analytics/visuelle-analyse-videostroeme/sabs.html

TABLE III
Performance Comparison on the Sequences Shown in Fig. 7

Sequence	RPCA			DECOLOR			TVRPCA		
	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
SABS	0.82	0.56	0.66	0.88	0.51	0.64	0.69	0.80	**0.74**
SABS-R	0.82	0.48	0.49	0.97	0.32	0.49	0.66	0.81	**0.73**
boats	0.42	0.03	0.06	0.36	0.71	**0.47**	0.25	0.83	0.39
canoe	0.86	0.20	0.32	0.12	0.98	0.21	0.76	0.96	**0.86**
fall	0.70	0.15	0.25	0.78	0.84	**0.81**	0.47	0.55	0.51
fountain01	0.72	0.01	0.02	0.96	0.01	0.02	0.40	0.01	**0.12**
fountain02	0.74	0.31	0.43	1.00	0.47	0.64	0.56	0.98	**0.72**
overpass	0.81	0.32	0.46	0.98	0.85	**0.91**	0.65	0.95	0.77
Average	0.74	0.26	0.34	0.76	0.59	0.52	0.56	0.74	**0.61**

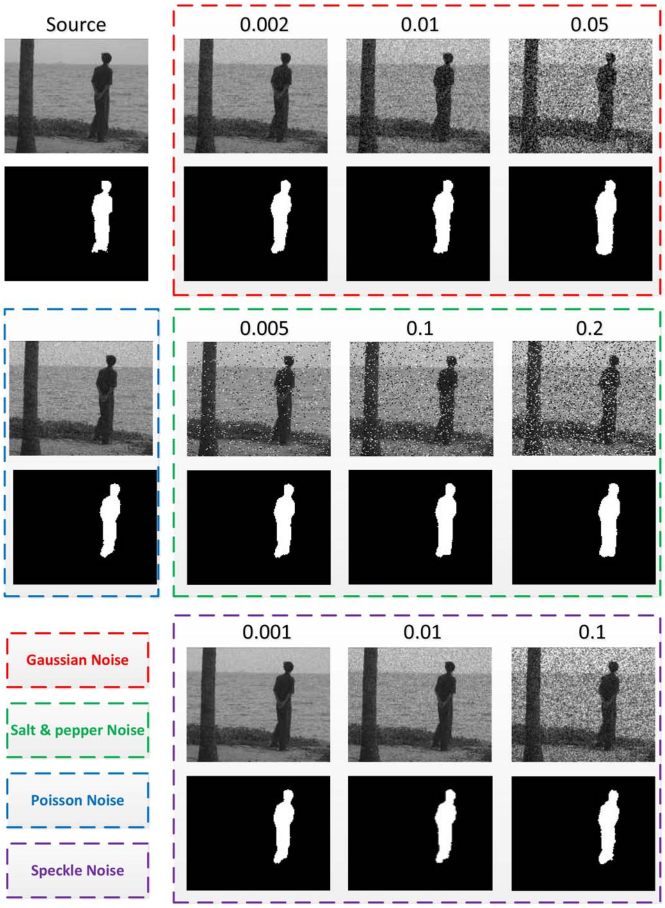

Fig. 6. Robustness to different noises. The source video frame and the result form our RPCA are displayed in the upper left corner of the figure. We add Gaussian noise, Poisson noise, salt and pepper noise, and speckle noise to the source video, respectively. We fix the noise mean to 0 and vary the noise variance, whose values are shown in the first row of each sub-figure, for three noise types except Poisson noise. We plot the noisy frame examples in the second row and the results from our proposed TVRPCA in third row. As the variance increases, the video gradually become unclear, and the detection task become difficult. Our method achieves good performance even in unclear videos with large noise variance.

spatial relationship. It improves the recall by 15% while reducing the precision by 16% in dynamic background region, which makes the F-measure stay or even decline. By taking into account the temporal relationship, TVRPCA removes these background regions that are inconsistent in time. Our method increases the precision by 33% and the F-measure by 24%. It is easy to conclude that TVPRCA outperforms original RPCA and DECOLOR on synthetic data sets from the results shown in Table III.

C. Change Detection Dataset

Change detection video database[3] [49] is considered as one of the most difficult tracking benchmarks, which consists of 31 real-world videos over 80 000 frames and spanning six categories including diverse motion and change detection challenges. To verify the performance of TVRPCA in complex scenarios, we only select the dynamic background category. This category is the most difficult among all the categories for mounted camera object detection [36], which contains six video sequences exhibiting dynamic background motions and shelters. These sequences are much more difficult than the previous datasets due to the following three factors.

1) One challenge comes from the significant dynamic background elements. For example, as shown in the fifth and sixth rows of Fig. 7, from the viewpoint of camera, the leaves shake heavily and the water flow of *Fountain* is intense, as the tree and *Fountain* in the fall and *Fountain01* sequences are close to the camera.
2) The camouflage, such as the motorboat in the boats sequence and the white car in the fall sequence, makes it more difficult to distinguish between the foreground and background.
3) In the sequences such as the *Fountain02*, the moving object is relative small (far from the camera), and may be partially occluded by the dynamic background, such as the *Fountain*.

The sequences and their corresponding results are presented in Fig. 7 (from third to eighth rows) and Table III (from third to eighth rows). RPCA inevitably detects a lot of dynamic backgrounds as moving foreground objects, which makes its precision very low as shown in boat and *Fountain02* sequence. When there are some shelters in front of the moving object, DECOLOR cannot remove these shelters' regions, as shown

[3] http://www.changedetection.net

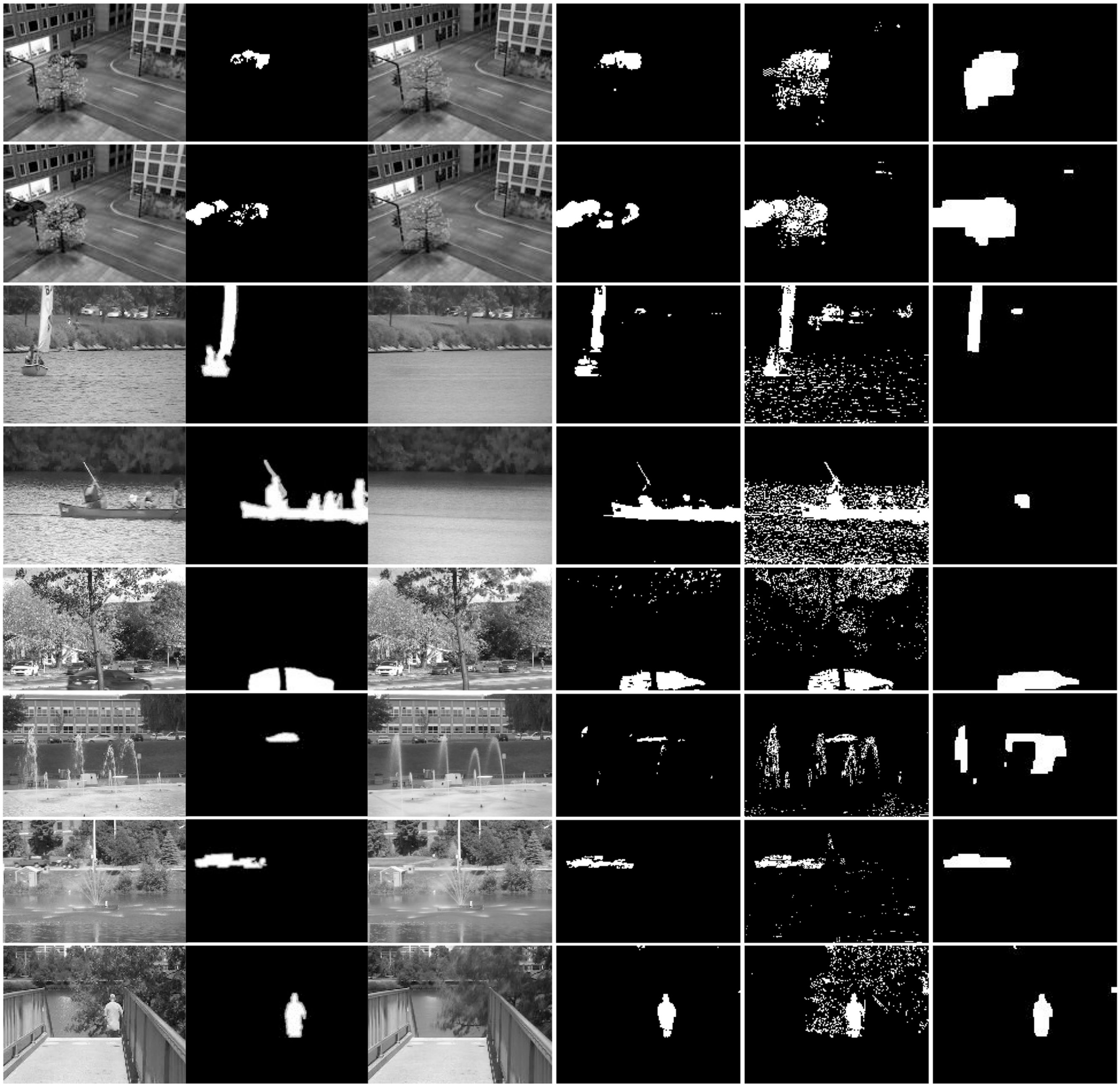

Fig. 7. Sample frames of surveillance videos in SABS and change detection datasets. First and second columns show the original frames and the ground truth foregrounds. The statics background and moving foreground objects recovered by TVRPCA are shown in third and fourth columns, respectively. The last two columns correspond to the results from RPCA and DECOLOR, respectively.

in the fifth row in Fig. 7. One reason is that although its greedy property is very useful to handle the camouflage problem, it also considers these shelters' regions as the inner parts of the objects. So, as shown in Table III, DECOLOR loses the precision sharply, especially in the condition where there are relatively small amounts of foreground pixels. Besides, DECOLOR misses some details of the detected objects such as the outline of the car in *Fountain02* sequence, which also reduces the precision. Different from above two methods, although TVRPCA drops recalls slightly, it intensively improves the precision and F-measure by removing the inconsistent movements. The reason why DECOLOR achieves better performance than TVRPCA in fall and overpass sequences is that in these two sequences, the camouflage is more important to F-measure than dynamic background. For example, in the overpass sequence, the color of the person's cloth is similar to that of the overpass.

Besides, we provide a comparison of running time between our proposed TVRPCA and DECOLOR which are both impose smoothness constraint on foreground. TVRPCA is implemented in MATLAB, while the core part of DECOLOR is implemented in C++. This experiment is conducted on a single PC with Intel Core i7-2600 3.4 GHz. CPU and 16.0 GB RAM. To separate foreground from a video with 633 frames and 128 × 160 pixels per frame, TVRPCA spends about 570 s while DECOLOR costs about 790 s.

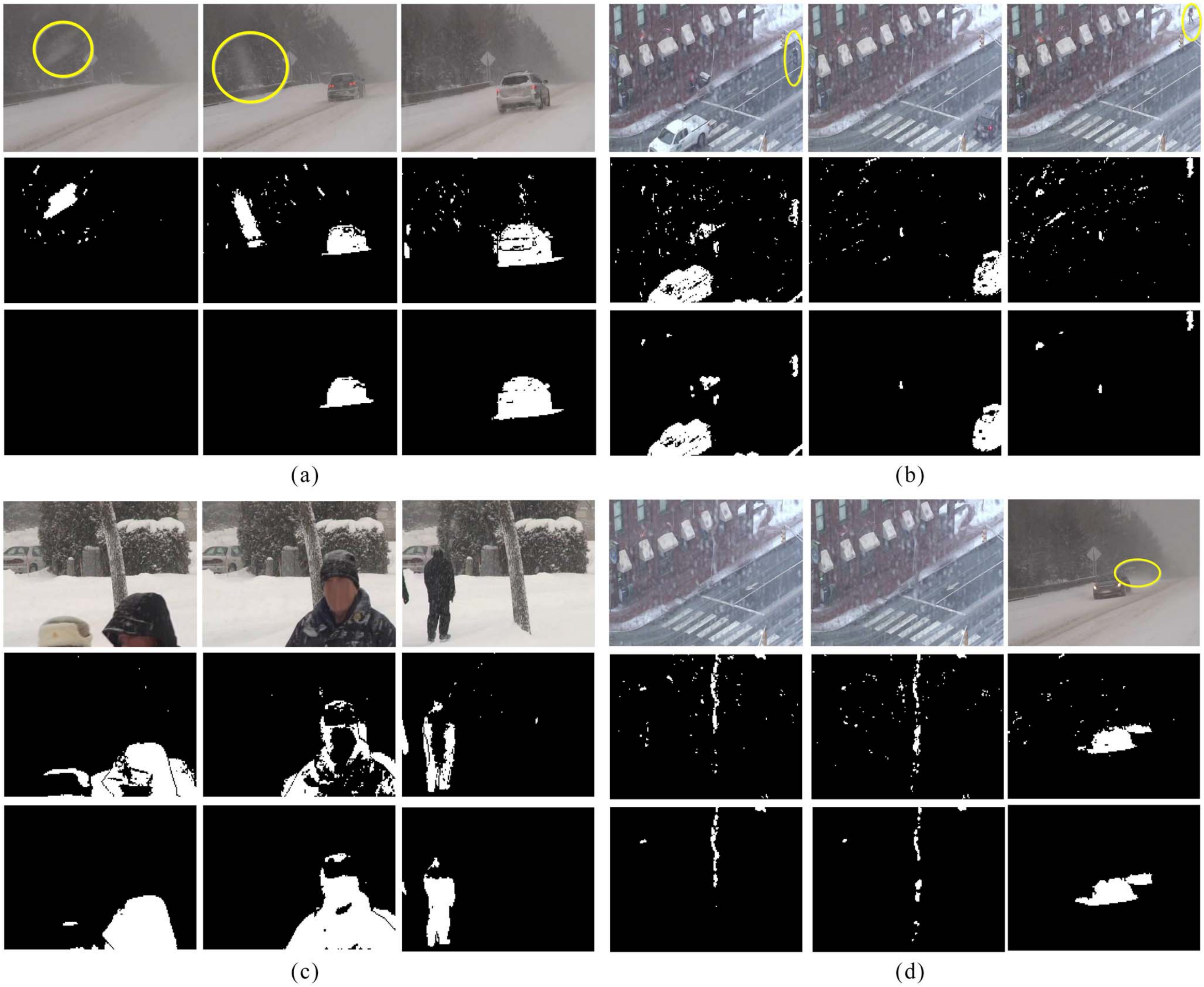

Fig. 8. Visual results of RPCA and TVRPCA on the *BadWeather* category on (a) *SnowFall*, (b) *WetSnow*, and (c) *Skating* sequences. Example frames are shown in first row, while the results from RPCA and TVRPCA are shown in second and third rows, respectively. From (a)–(c), we find that TVRPCA suppresses most of the snowflakes and raindrops from the detected foreground, and makes the detected area more continuous. (d) Two failure cases on the above sequences. The first case is caused by the rolling drops on the window, while the second case is caused by the snow brought up by the passing car.

D. Bad Weather Sequences

In addition to common camera mounted surveillance, TVRPCA can also be applied to video surveillance under bad weather, which has great significance to security protection. Please consider a scenario that the border lines of many countries are at high-altitude areas, it is important to robustly monitor the situation with bad weather. Another instance is that, according to the surveys [50]–[52], many serious incidents, such as traffic accidents and crimes, are happened under bad weather. Bad weather makes the moving foreground detection much more difficult. For one thing, snow and rain reduce the screen contrast of surveillance video, which makes distinction between foreground and background difficult as shown in Fig. 8(a). For another, the movements of snowflakes and raindrops are detected as moving foreground, which draws down the ability to identify small objects as shown in Fig. 8(b). Finally, the rain drops on the lens of monitor equipments disturb the accurate detection as shown in Fig. 8(b) and (d). As a result, moving object detection under bad weather has become a hot topic in computer vision. We conduct experiments on *BadWeather* category published in ChangeDetection.NET 2014 dataset, which consists of four typical bad weather sequences blizzard, *Skating*, *SnowFall*, and *WetSnow*.

Fig. 8 shows the results of RPCA and TVRPCA on *BadWeather* category, in which Fig. 8(a)–(c) is the results on sequences *SnowFall*, *WetSnow*, and *Skating*, respectively. In Fig. 8(a) and (b), we find that the snowflakes and raindrops are all be detected as foreground by RPCA, especially the two large snowflakes near the camera in the first two column of Fig. 8(a). Furthermore, the raindrops make the detected pedestrians not obvious in the first and third columns of Fig. 8(b). In the first column of Fig. 8(b), the pedestrian crosses the street as shown in the yellow circle of the figure, while in the third column of Fig. 8(b) pedestrian walks on the sidewalk as shown in the top right circle of the figure. Thus, we can find out that TVRPCA can correctly detection small objects, such as pedestrian. The main reason is that although these small objects are sparse in spatial directions, they are continuous and not sparse in temporal direction. In summary, TVRPCA can correctly detect small objects with continuous movements. Our method regards the fast moving small object, such as raindrops, as dynamics background since it is spare in both temporal and spatial directions. Note that the detected regions in the left top corner of all three frames in Fig. 8(b) are caused by the continuous movements of a national flag. From the results, we find that compared with the original RPCA, our proposed TVRPCA largely alleviates these problems, besides the discontinuity problem in Fig. 8(c) caused by lingering objects. It fully demonstrates the superiority of TVRPCA on surveillance under bad weather.

However, there exist some problems that cannot be properly solved by TVRPCA. In Fig. 8(d), we show two failure cases on the three sequences. The first case is caused by the rolling drops on the window, and the second case is caused by the snow brought up by the passing car. The reasons why TVRPCA fails are that their size is not small and their movements are continuous over space and time.

V. Conclusion

In this paper, we have proposed a novel framework named TVRPCA to handle the scenarios with complex dynamic backgrounds, and slowly moving or lingering objects, based on the assumption that the moving foreground objects should be contiguous in both space and time and dynamic background is sparser than real foreground object. We have formulated the target problem in a unified objective function by introducing spatial and temporal continuity into the original RPCA using total variation regularization, while the proposed algorithm can effectively seek its optimal solution. The experimental results on synthetic and real datasets have demonstrated the clear advantages of TVRPCA compared with the state-of-the-art methods, which indicates that our method has wider applicable range than the others, especially in the presence of the bad weather or complex background.

There remain some problems that cannot be perfectly solved by our proposed TVRPCA. First, if the foreground objects have the similar appearance with the background, our framework cannot separate them as foreground. For example, in *Watersurface* sequence, man's trousers and the bush have the similar color. Second, the decomposition of the sparser component and the smooth component may probably miss some objects that are both small and fast moving, such as the fast-moving cars which are far from the camera. In the future, we want to investigate how to integrate the structural priori information into the background subtraction to solve this problem. Besides, we will conduct research on how to make our algorithm more efficient and how to design its online version.

References

[1] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014.

[2] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, Dec. 2006, Art. ID 13.

[4] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[5] R. T. Azuma, "A survey of augmented reality," *Presence Teleoperators Virtual Environ.*, vol. 6, no. 4, pp. 355–385, 1997.

[6] D. van Krevelen and R. Poelman, "A survey of augmented reality technologies, applications and limitations," *Int. J. Virtual Reality*, vol. 9, no. 2, pp. 1–20, 2010.

[7] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, vol. 1. Corfu, Greece, 1999, pp. 255–261.

[8] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Comput. Vis. Image Understand.*, vol. 104, no. 2, pp. 90–126, 2006.

[9] L. Li, W. Huang, I.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[10] T. Veit, F. Cao, and P. Bouthemy, "A maximality principle applied to a contrario motion detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 1. Genoa, Italy, Sep. 2005, pp. 1061–1064.

[11] S. Huwer and H. Niemann, "Adaptive change detection for real-time surveillance applications," in *Proc. 3rd IEEE Int. Workshop Visual Surveill.*, Dublin, Ireland, 2000, pp. 37–46.

[12] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Fort Collins, CO, USA, 1999, pp. 1–7.

[13] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2. Nice, France, 2003, pp. 734–741.

[14] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[15] A. Mittal and N. Paragios, "Motion-based background subtraction using adaptive kernel density estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Washington, DC, USA, Jun. 2004, pp. 302–309.

[16] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[17] T. Zhou and D. Tao, "Shifted subspaces tracking on sparse outlier for motion segmentation," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, Beijing, China, 2013, pp. 1946–1952.

[18] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, "TILT: Transform invariant low-rank textures," *Int. J. Comput. Vis.*, vol. 99, no. 1, pp. 1–24, 2012.

[19] J. Chen and J. Yang, "Robust subspace segmentation via low-rank representation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1432–1445, Aug. 2014.

[20] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[21] G. Liu *et al.*, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[22] G. Liu and S. Yan, "Latent low-rank representation for subspace segmentation and feature extraction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 1615–1622.

[23] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. La Jolla, CA, USA: Curran Assoc. Inc., 2009, pp. 2080–2088.

[24] A. Ganesh, J. Wright, X. Li, E. Candes, and Y. Ma, "Dense error correction for low-rank matrices via principal component pursuit," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Austin, TX, USA, Jun. 2010, pp. 1513–1517.

[25] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[26] A. Ganesh *et al.*, "Fast algorithms for recovering a corrupted low-rank matrix," in *Proc. 3rd IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Aruba, The Netherlands, 2009, pp. 213–216.

[27] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint," in *Proc. 11th Int. Conf. Comput. Vis. (ACCV)*, Daejeon, Korea, 2013, pp. 315–320.

[28] C. Guyon, T. Bouwmans, and E. Zahzah, "Foreground detection via robust low rank matrix factorization including spatial constraint with iterative reweighted regression," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Tsukuba, Japan, 2012, pp. 2805–2808.

[29] S. Zhang, H. Yao, and S. Liu, "Dynamic background subtraction based on local dependency histogram," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 23, no. 7, pp. 1397–1419, 2009.

[30] S. Zhang, H. Yao, and S. Liu, "Dynamic background modeling and subtraction using spatio-temporal local binary patterns," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, San Diego, CA, USA, Oct. 2008, pp. 1556–1559.

[31] S. Zhang, H. Yao, S. Liu, X. Chen, and W. Gao, "A covariance-based method for dynamic background subtraction," in *Proc. 19th Int. Conf. Pattern Recognit. (ICPR)*, Tampa, FL, USA, Dec. 2008, pp. 1–4.

[32] S. Zhang, H. Yao, and S. Liu, "Spatial-temporal nonparametric background subtraction in dynamic scenes," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, New York, NY, USA, Jun. 2009, pp. 518–521.

[33] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[34] X. Zhou, C. Yang, and W. Yu, "Automatic mitral leaflet tracking in echocardiography by outlier detection in the low-rank representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, Jun. 2012, pp. 972–979.

[35] Z. Gao, L.-F. Cheong, and M. Shan, "Block-sparse RPCA for consistent foreground detection," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 690–703.

[36] Z. Gao, L. Cheong, and Y. Wang, "Block-sparse RPCA for salient motion detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1975–1987, Oct. 2014.

[37] K. Rosenblum, L. Zelnik-Manor, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," in *Proc. AAAI Fall Symp. Manifold Learn.*, Arlington, VA, USA, 2010, pp. 50–58.

[38] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2012, pp. 1568–1575.

[39] X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma, "Robust foreground detection using smoothness and arbitrariness constraints," in *Computer Vision–ECCV*. Cham, Switzerland: Springer, 2014, pp. 535–550.

[40] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D Nonlin. Phenom.*, vol. 60, no. 1, pp. 259–268, 1992.

[41] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[42] A. Chambolle, "An algorithm for total variation minimization and applications," *J. Math. Imag. Vis.*, vol. 20, nos. 1–2, pp. 89–97, 2004.

[43] M. Tao and J. Yang, "Alternating direction algorithms for total variation deconvolution in image reconstruction," Dept. Math., Nanjing Univ., Nanjing, China, Tech. Rep. TR0918, 2009.

[44] N. Wang, T. Yao, J. Wang, and D.-Y. Yeung, "A probabilistic approach to robust matrix factorization," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 126–139.

[45] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, no. 5, pp. 403–420, 1970.

[46] 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.

[47] S. Zhang, S. Kasiviswanathan, P. C. Yuen, and M. Harandi, "Online dictionary learning on symmetric positive definite manifolds with vision applications," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin, TX, USA, Jan. 2015, pp. 1–9.

[48] S. Brutzer, B. Hoferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Providence, RI, USA, 2011, pp. 1937–1944.

[49] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Providence, RI, USA, 2012, pp. 1–8.

[50] E. G. Cohn, "Weather and crime," *Brit. J. Criminol.*, vol. 30, no. 1, pp. 51–64, 1990.

[51] S. J. Garzino, "Lunar effects on mental behavior a defense of the empirical research," *Environ. Behav.*, vol. 14, no. 4, pp. 395–417, 1982.

[52] E. G. Cohn and J. Rotton, "Weather, seasonal trends and property crimes in Minneapolis, 1987–1988. A moderator-variable time-series analysis of routine activities," *J. Environ. Psychol.*, vol. 20, no. 3, pp. 257–272, 2000.

Xiaochun Cao (SM'14) received the B.E. and M.E. degrees from Beihang University, Beijing, China, both in computer science, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, FL, USA.

He was a Research Scientist with ObjectVideo Inc., Reston, VA, USA, for three years. From 2008 to 2012, he was a Professor with Tianjin University, Tianjin, China. He is a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. He has authored and co-authored over 80 journal and conference papers.

Prof. Cao was a recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition in 2004 and 2010 and the University Level Outstanding Dissertation Award nomination for his dissertation.

Liang Yang received the B.E. and M.E. degrees from Nankai University, Tianjin, China, in 2004 and 2007, respectively, both in computational mathematics. He is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

He is an Assistant Professor with the School of Information Engineering, Tianjin University of Commerce, Tianjin. His current research interests include community detection, semi-supervised learning, low-rank modeling, and deep learning.

Xiaojie Guo (M'13) received the B.E. degree in software engineering from the School of Computer Science and Technology, Wuhan University of Technology, Wuhan, China, in 2008, and the M.S. and Ph.D. degrees in computer science from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2010 and 2013, respectively.

He is currently an Assistant Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

Dr. Guo was a recipient of the Piero Zamperoni Best Student Paper Award at the International Conference on Pattern Recognition (International Association on Pattern Recognition) in 2010.