Robust Foreground Detection Using Smoothness and Arbitrariness Constraints

Xiaojie Guo[1], Xinggang Wang[2], Liang Yang[1], Xiaochun Cao[1] and Yi Ma[3]
xj.max.guo@gmail, xgwang@hust.edu.cn,
{yangliang, caoxiaochun}@iie.ac.cn, mayi@shanghaitech.edu.cn

[1]State Key Lab of Information Security, IIE, Chinese Academy of Sciences, China
[2]Department of EI, Huazhong University of Science and Technology, China
[3]School of Information Science and Technology, ShanghaiTech University, China

Abstract. Foreground detection is one of the most important and challenging problems in computer vision, which plays a core role in a wide spectrum of applications such as tracking and behavior analysis. It, especially for videos captured by fixed cameras, can be posed as a component decomposition problem, the background of which is typically assumed to lie in a low dimensional subspace. However, in real world cases, dynamic backgrounds like waving trees and water ripples violate the assumption. Besides, noises caused by the image capturing process and, camouflage and lingering foreground objects would also significantly increase the difficulty of accurate foreground detection. That is to say, simply imposing the correlation constraint on the background is no longer sufficient for such cases. To overcome the difficulties mentioned above, this paper proposes to further take into account foreground characteristics including 1) the smoothness: the foreground object should appear coherently in spatial domain and move smoothly in temporal, and 2) the arbitrariness: the appearance of foreground could be with arbitrary colors or intensities. With the consideration of the smoothness and the arbitrariness of foreground as well as the correlation of (static) background, we formulate the problem in a unified framework from a probabilistic perspective, and design an effective algorithm to seek the optimal solution. Experimental results on both synthetic and real data demonstrate the clear advantages of our method compared to the state of the art alternatives.

1 Introduction

Foreground detection is fundamental to numerous computer vision applications like tracking [26,17] and behavior analysis [4], as the foreground is usually of more interest and matters more than the background to further analysis. The problem of foreground detection can be viewed as a decomposition of a video into the foreground component and the background. From this view of point, it can be achieved through either foreground or background modeling, which mainly derives object detector based, motion based, and background construction based approaches. Object detectors are generally built by offline training [21] or online learning [1], which perform as classifiers to determine whether a

target region (often searched by a sliding window) belongs to the foreground or the background. But, most of offline trained detectors are based on separate datasets, which would be insufficiently discriminative for different cases and thus lead to poor performance. The online learned ones require an initialization by manually labeling at the start of a video, which limits the applicability in automated systems. As for motion-based methods [20,3,19], they avoid such training and learning phases by exploiting motion patterns to classify pixels into different groups. This kind of methods can deal with the cases in the presence of camera motion, but the objects are assumed to move regularly [20,3] in respective regions, which is often violated in practical situations.

Alternatively, constructing the background that is always present, seems to be more simple and easier than modeling the foreground that may be of diverse appearance and complex motion. Background subtraction [8] is probably the most straightforward method in this category. The difference image can be obtained by subtracting the reference background from the current frame in a pixel-wise manner, which is the etymology of background subtraction. If the absolute difference exceeds a threshold, the pixel in question is declared to belong to the foreground. Temporal average and median filtering are two of classical background subtraction methods. These approaches are simple and efficient, but extremely sensitive because it assumes a static background with well behaved objects. In practice, this is almost never the case. To remedy the sensitivity, simple Gaussian [24] is proposed to represent each background pixel using a Gaussian model, the pixel is determined to be the background if it falls into a deviation around the mean, otherwise the foreground. A more robust strategy [18] is to record the possible values of each pixel of the background image over time by a mixture of Gaussians. Instead of modeling the feature vectors of each pixel by a mixture of several Gaussians, Elgammal *et al.* [5] try to evaluate the probability of a background pixel using a nonparametric kernel density estimation based on very recent historical samples in the image sequence. In order to achieve the quick adaptation to changes in the scene and low false positive rates, they design a scheme to combine the results of the short-term and long-term background models for better updating decisions. Maddalena and Petrosino [12] propose an approach based on self organization through artificial neural networks, which claims to be robust to multiple situations such as gradual illumination changes.

Although the methods mentioned above provide promising progresses in foreground detection, they are rarely aware of the intensive global correlation of background across different frames. By considering the correlation (low rank) prior, it is natural to formulate the background as a linearly correlated model, which turns out to be a classic problem of learning a low dimensional linear model from high dimensional data. Mathematically, let $\boldsymbol{O} \in \mathcal{R}^{m \times n}$ be the observation matrix containing n frames. Each column of $\boldsymbol{O}$ corresponds to a vectorized frame that has m pixels. $\boldsymbol{O}$ can be decomposed into two components, *i.e.* $\boldsymbol{O} = \boldsymbol{B} + \boldsymbol{R}$, where $\boldsymbol{B} \in \mathcal{R}^{m \times n}$ and $\boldsymbol{R} \in \mathcal{R}^{m \times n}$ denote the background and the residual, respectively. Consequently, the objective can be designed as:

$$\underset{\boldsymbol{B},\boldsymbol{R}}{\operatorname{argmin}} \operatorname{rank}(\boldsymbol{B}) + \alpha \Upsilon(\boldsymbol{R}), \quad \text{s.t.} \quad \boldsymbol{O} = \boldsymbol{B} + \boldsymbol{R}, \tag{1}$$

where rank($\boldsymbol{B}$) computes the rank of $\boldsymbol{B}$ and is usually substituted by the nuclear norm, *i.e.* $\|\boldsymbol{B}\|_*$, to make it convex and computationally tractable, and α is the weight with respect to $\Upsilon(\boldsymbol{R})$ that acts as the regularizer on the residual. If $\Upsilon(\boldsymbol{R}) \doteq \|\boldsymbol{R}\|_0$ is employed, Eq. (1) becomes the problem of Robust PCA (RPCA) that is designed to be robust to sparse outliers with arbitrary magnitudes, where $\|\cdot\|_0$ denotes the ℓ^0 norm. But the problem is intractable and extremely difficult to approximate due to the non-convexity of ℓ^0. Alternatively, ℓ^1 norm can be employed as the convex surrogate of ℓ^0, which is optimal for Laplacian distribution. Based on the convex relaxation, many solutions have been investigated [2,28]. Moreover, the online extensions [7,25] broaden the applicable range of R-PCA for the tasks with the requirement of incremental processing. Based on the advanced solutions and extensions, plenty of interesting applications have been developed [14,27]. Equivalently, $\boldsymbol{B}$ can be replaced with $\boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U} \in \mathcal{R}^{m\times r}$ and $\boldsymbol{V} \in \mathcal{R}^{n\times r}$ (usually $r \ll \min\{m, n\}$). That is to say, the rank of $\boldsymbol{U}\boldsymbol{V}^T$ is guaranteed to be never over r, thus the rank term can be discarded. By casting the problem into probabilistic models, [22,13,23,6] have proven to be effective to solve this problem.

However, in real world cases, only imposing the global correlation constraint on the background component is inadequate, as the dynamic background, like waving trees and water ripples, breaks the low rank assumption, and the noise caused by the image capturing process, and the camouflage also significantly increase the difficulty of accurately detecting foregrounds. Actually, some useful properties of foreground could be exploited jointly with the correlation of background for improving the performance. Specifically, the foreground should appear coherently in space and move smoothly in time. We call this property the smoothness. [29] and [23] utilize Markov Random Field (MRF) constraints directly on the foreground support to guarantee the spatial and temporal smoothness, which provide desirable results but with relatively high computational complexities. Instead, we propose a more efficient solution than MRF by using a total variation (TV) regularizer [15]. In addition, the appearance of foreground could be with any values. Even though the background is unknown in advance, the residual caused by foreground distributes the same as the foreground. That is to say, $\boldsymbol{R}$ caused by foreground is more like uniformly distributed than either Gaussian or Laplacian. We name this characteristic of foreground the arbitrariness. In this work, we focus on how to harness the arbitrariness and the smoothness of foreground and the global correlation of background for boosting the performance of foreground detection.

The main contributions of this paper can be summarized as follows:

- Our framework harnesses three priors, including the arbitrariness of foreground appearance, the spatial-temporal smoothness of foreground, and the correlation of background, in a unified fashion.
- We design an effective and efficient algorithm to seek the optimal solution of the associated optimization problem based on Augmented Lagrangian Multiplier with Alternating Direction Minimizing (ALM-ADM) strategy. Extensive experiments are conducted to demonstrate the efficacy of our method.

2 Our Method

2.1 Problem Formulation

Recall that each element $\boldsymbol{O}_{ij}(i = 1, 2, ..., m; j = 1, 2, ...n)$ of the observation matrix $\boldsymbol{O}$ can be modeled as $\boldsymbol{O}_{ij} = \boldsymbol{U}_i\boldsymbol{V}_j^T + \boldsymbol{R}_{ij}$, where $\boldsymbol{U}_i$ and $\boldsymbol{V}_i$ are the i^{th} row vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. Since a foreground pixel can be any value within a bounded range, the corresponding residual falls into $[-\boldsymbol{U}_i\boldsymbol{V}_j^T, 255 - \boldsymbol{U}_i\boldsymbol{V}_j^T]$ (the arbitrariness). Thus, we can assume they follow the uniform distribution $\frac{1}{256}$. As for the residuals caused by the other factors, we simply assume they (approximately) follow a Gaussian distribution $\mathcal{N}(0, \sigma^2)$[1]. Let π^u and π^g be the percentages of foreground and the other, respectively, which we actually do not know in advance. As a result, each $\boldsymbol{R}_{ij}$ can be seen as a sample from a mixture model of distributions with probability $p(\boldsymbol{R}_{ij}) = \pi^g\mathcal{N}(\boldsymbol{R}_{ij}|0, \sigma^2) + \pi^u\frac{1}{256}$, where $\pi^g + \pi^u = 1$. Then the likelihood of $\boldsymbol{O}$ can be written as:

$$p(\boldsymbol{O}|\boldsymbol{U}, \boldsymbol{V}, \Theta) = \prod_{i,j} p(\boldsymbol{O}_{ij}|\boldsymbol{U}_i\boldsymbol{V}_j^T, \Theta) = \prod_{i,j}\left(\pi^g\mathcal{N}(\boldsymbol{O}_{ij}|\boldsymbol{U}_i\boldsymbol{V}_j^T, \sigma^2) + \frac{\pi^u}{256}\right), \quad (2)$$

where $\Theta = \{\sigma^2, \pi^g, \pi^u\}$ is a parameter vector. The negative log-likelihood functional of Eq. (2) is given as follows:

$$\mathcal{L}(\boldsymbol{U}, \boldsymbol{V}, \Theta) = -\sum_{i,j}\log\left(\pi^g\mathcal{N}(\boldsymbol{O}_{ij}|\boldsymbol{U}_i\boldsymbol{V}_j^T, \sigma^2) + \frac{\pi^u}{256}\right). \quad (3)$$

By applying Lemma 1 on $\mathcal{L}(\boldsymbol{U}, \boldsymbol{V}, \Theta)$, we have:

$$\mathcal{C}(\boldsymbol{U}, \boldsymbol{V}, \Theta, \Phi) = \sum_{i,j}\Phi_{ij}^g(\log\Phi_{ij}^g - \log\pi^g\mathcal{N}(\boldsymbol{O}_{ij}|\boldsymbol{U}_i\boldsymbol{V}_j^T, \sigma^2)) + \Phi_{ij}^u(\log\Phi_{ij}^u - \log\frac{\pi^u}{256}), \quad (4)$$

with an additional variable (hidden parameter) Φ.

Lemma 1. *(Commutativity of Log-Sum operations. [11]) Given two functions* $\pi_k(x) > 0$ *and* $p_k(x) > 0$*, we have:*

$$-\log\sum_{k=1}^{K}\pi_k(x)p_k(x) = \min_{\Phi(x)\in\Delta_+} -\sum_{k=1}^{K}\Phi_k(x)\log(\pi_k(x)p_k(x)) + \sum_{k=1}^{K}\Phi_k(x)\log\Phi_k(x),$$

where $\Phi(x) = \{\Phi_1(x), \Phi_2(x), , ..., \Phi_K(x)\}$ *are hidden parameters, and* $\Delta_+ = \{\Phi(x) : 0 < \Phi_k(x) < 1, and \sum_{k=1}^{K}\Phi_k(x) = 1\}$ *is a convex relaxation of a characteristic function decomposition.*

As can be seen from Eq. (4), minimizing $\mathcal{C}(\boldsymbol{U}, \boldsymbol{V}, \Theta, \Phi)$ will give a minimizer of $\mathcal{L}(\boldsymbol{U}, \boldsymbol{V}, \Theta)$, which can be processed easily and efficiently as it becomes a quadratic function by interchanging the logarithm and summation operations. The following propositions further show good properties of $\mathcal{C}(\boldsymbol{U}, \boldsymbol{V}, \Theta, \Phi)$ that inspire the design of our method.

[1] Although these residuals may be not pure Gaussian, our method can effectively handle this issue thanks to the smoothness, which will be demonstrated in Sec. 3.

Proposition 1. *Both* $\mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi)$ *and* $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\Theta)$ *have the same global minimizer* $(\boldsymbol{U}^*,\boldsymbol{V}^*,\Theta^*)$ *if* $\Phi \in \Delta_+$.

Moreover, if the following Alternating Direction Minimizing (ADM) strategy is employed to minimize $\mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi)$:

$$\begin{aligned} \Phi^{t+1} &= \underset{\Phi\in\Delta_+}{\operatorname{argmin}} \mathcal{C}(\boldsymbol{U}^t,\boldsymbol{V}^t,\Theta^t,\Phi); \\ (\boldsymbol{U}^{t+1},\boldsymbol{V}^{t+1},\Theta^{t+1}) &= \underset{\boldsymbol{U},\boldsymbol{V},\Theta}{\operatorname{argmin}} \mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi^{t+1}), \end{aligned} \tag{5}$$

the energy of $\mathcal{L}(\boldsymbol{U},\boldsymbol{V},\Theta)$ will gradually decrease as the two steps iterate (Proposition 2). That is to say, the local convergence of the problem is guaranteed.

Proposition 2. *The sequence* $(\boldsymbol{U}^t,\boldsymbol{V}^t,\Theta^t)$ *computed by (5) leads to*

$$\mathcal{L}(\boldsymbol{U}^{t+1},\boldsymbol{V}^{t+1},\Theta^{t+1}) \le \mathcal{L}(\boldsymbol{U}^t,\boldsymbol{V}^t,\Theta^t). \tag{6}$$

We observe that the foreground objects, such as cars and pedestrians, should appear to be spatially coherent and move smoothly in temporal. Thus, imposing a temporal-spatial smoothness constraint on foreground would boost the performance of foreground detection. Let us here revisit the model of the observed matrix $\boldsymbol{O}$ from another viewpoint, *i.e.* $\boldsymbol{O} = \mathcal{P}_\Omega(\boldsymbol{B}) + \mathcal{P}_{\Omega^\perp}(\boldsymbol{F})$, where $\mathcal{P}_\Omega(\cdot)$ is the orthogonal projection operator on the support $\Omega \in \{0,1\}^{m\times n}$, and $\Omega^\perp \in \{0,1\}^{m\times n}$ stands for the complementary support of Ω. Based on the above observation, it is intuitive to enforce the temporal-spatial smoothness on the support Ω (or $\Omega^\perp$ equivalently). But, the binary support is unknown in advance, that is to say, directly operating on the unknown binary support is extremely difficult. Moreover, the residual reflects the difference between the observation and the background, rather than the support of foreground object. Therefore, it is still improper to impose the temporal-spatial smoothness on the residual component $\boldsymbol{R}$ without assumptions.

Alternatively, as we have introduced, the hidden variable Φ can perform as the term with the smoothness property. In this work, we only take into account Φ^g because Φ^u performs actually the same as Φ^g due to $\Phi^g + \Phi^u = \mathbf{1}$. Please consider an extreme case that if the Gaussian function of mean 0 and variance σ^2 goes to 0 infinitesimally, each Φ^g_{ij} equals to 1 if $\boldsymbol{O}_{ij} = \boldsymbol{U}_i\boldsymbol{V}_j^T$, otherwise 0. For our problem, we relax the restrict binary requirement of support to a continuous value range $(0,1)$[2], in which σ^2 controls the width of the interface between 0 and 1. As a consequence, the regularization of smoothness on foreground can be achieved by imposing the anisotropic total variation on Φ^g, which is defined as:

$$\|\Phi^g\|_{tv} = \sum_{i,j} |[\boldsymbol{D}_h\Phi^g]_{ij}| + |[\boldsymbol{D}_v\Phi^g]_{ij}| + |[\boldsymbol{D}_t\Phi^g]_{ij}|, \tag{7}$$

[2] The reason why the range is $(0,1)$ instead of $[0,1]$ is to satisfy $\Phi \in \Delta_+$ introduced in Lemma 1. This can be easily done by adding a very small $\epsilon = 10^{-7}$ to Φ^u and Φ^g, then normalizing them by letting their summation be 1.

where $\boldsymbol{D}_h$, $\boldsymbol{D}_v$ and $\boldsymbol{D}_t$ are the forward finite difference operators in horizontal, vertical and temporal directions, respectively. By slightly transforming the form of (7), we have $\|\Phi^g\|_{tv} = \|\boldsymbol{D}\Phi^g\|_1$, where $\boldsymbol{D} = [\boldsymbol{D}_h^T, \boldsymbol{D}_v^T, \boldsymbol{D}_t^T]^T$.

By puting all the concerns aforementioned together, we can naturally formulate the problem of robust foreground detection in the following shape:

$$\min_{\boldsymbol{U},\boldsymbol{V},\Theta,\Phi} \mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi) + \lambda\|\boldsymbol{D}\Phi^g\|_1, \tag{8}$$

where λ is the weight of the smoothness regularizer.

2.2 Optimization

As can be seen from Eq. (8), it is difficult to directly optimize because the total variation regularizer breaks the linear structure of Φ^g. To efficiently and effectively solve the problem, we introduce two auxiliary variables to make the problem separable, which gives the following constraint minimizing problem:

$$\min_{\boldsymbol{U},\boldsymbol{V},\Theta,\Phi} \mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi) + \lambda\|\boldsymbol{T}\|_1, \quad \text{s.t.} \quad \boldsymbol{W} = \Phi^g, \boldsymbol{T} = \boldsymbol{D}\boldsymbol{W}. \tag{9}$$

For the above constraint minimizing problem, the penalty technique [10] can be adopted to change the constraint problem (9) into the unconstraint one in the following shape:

$$\begin{cases} \mathcal{Q}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi,\boldsymbol{T},\boldsymbol{W}) = \mathcal{C}(\boldsymbol{U},\boldsymbol{V},\Theta,\Phi) + \lambda\|\boldsymbol{T}\|_1 + \dfrac{\mu}{2}\|\boldsymbol{W} - \Phi^g\|_F^2 \\ \qquad\qquad + < \boldsymbol{X}, \boldsymbol{W} - \Phi^g > + \dfrac{\mu}{2}\|\boldsymbol{T} - \boldsymbol{D}\boldsymbol{W}\|_F^2 + < \boldsymbol{Y}, \boldsymbol{T} - \boldsymbol{D}\boldsymbol{W} >, \end{cases} \tag{10}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, $< \cdot, \cdot >$ represents matrix inner product, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are the Lagrangian multipliers, and μ is a positive penalty scalar. Below are the solutions of the sub-problems based on the ADM strategy.

Φ sub-problem: For computing $\hat{\Phi^u_{ij}}^{(t+1)}$, we take derivative of $\mathcal{Q}$ with respect to Φ^u_{ij} with the unrelated terms fixed and set it to zero, then obtain:

$$\hat{\Phi^u_{ij}}^{(t+1)} = \underset{\Phi^u_{ij}}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)}, \Theta^{(t)}, \Phi^u, \Phi^{g(t)}, \boldsymbol{T}^{(t)}, \boldsymbol{W}^{(t)}) = \frac{\pi^{u(t)}}{256\exp(1)}. \tag{11}$$

Similarly, the problem corresponding to $\hat{\Phi^g_{ij}}$ turns out to be:

$$\begin{cases} \hat{\Phi^g_{ij}}^{(t+1)} = \underset{\Phi^g_{ij}}{\operatorname{argmin}} \mathcal{Q}(\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)}, \Theta^{(t)}, \Phi^{u(t+1)}, \Phi^g, \boldsymbol{T}^{(t)}, \boldsymbol{W}^{(t)}) \\ \qquad\qquad = \underset{\Phi^g_{ij}}{\operatorname{argmin}} -\Phi^g_{ij}\log\dfrac{\pi^{g(t)}}{\sqrt{2\pi\sigma^{2(t)}}}\exp\left(\dfrac{-(\boldsymbol{O}_{ij} - \boldsymbol{U}_i^{(t)}\boldsymbol{V}_j^{(t)T})^2}{2\sigma^{2(t)}}\right) \\ \qquad\qquad\qquad + \Phi^g_{ij}\log\Phi^g_{ij} + \dfrac{\mu(t)}{2}(\boldsymbol{W}_{ij}^{(t)} - \Phi^g_{ij})^2 + \boldsymbol{X}_{ij}^{(t)}\Phi^g_{ij}, \end{cases} \tag{12}$$

However, the quadratic term in (12), *i.e.* $\frac{\mu^{(t)}}{2}(\boldsymbol{W}_{ij}^{(t)} - \Phi_{ij}^{g})^2$, destroys the closed form solution of Φ_{ij}^{g}. For addressing this difficulty, we introduce pixel-wise weights $\eta_{ij}^{(t)}$ such that $\eta_{ij}^{(t)} = \frac{\boldsymbol{W}_{ij}^{(t)} - \Phi_{ij}^{g(t)}}{2}$ to approximate the original quadratic term. That is to say, we replace $\frac{\mu^{(t)}}{2}(\boldsymbol{W}_{ij}^{(t)} - \Phi_{ij}^{g})^2$ with $\mu^{(t)}\eta_{ij}^{(t)}(\boldsymbol{W}_{ij}^{(t)} - \Phi_{ij}^{g})$. Thus, the solution of (12) can be easily computed by:

$$\hat{\Phi_{ij}^{g}}^{(t+1)} = \frac{\pi^{g(t)} \exp\left(\frac{-(\boldsymbol{O}_{ij} - \boldsymbol{U}_i^{(t)}\boldsymbol{V}_j^{(t)T})^2}{2\sigma^{2(t)}}\right)}{\sqrt{2\pi\sigma^{2(t)}} \exp\left(1 - \mu^{(t)}\eta_{ij}^{(t)} - \boldsymbol{X}_{ij}^{(t)}\right)}. \tag{13}$$

The final $\Phi_{ij}^{u(t+1)}$ and $\Phi_{ij}^{g(t+1)}$ are obtained by enforcing their summation to be 1, thus we have:

$$\begin{aligned}
\Phi_{ij}^{u(t+1)} &= \frac{\sqrt{2\pi\sigma^{2(t)}}\pi^{u(t)} \exp\left(-\mu^{(t)}\eta_{ij}^{(t)} - \boldsymbol{X}_{ij}^{(t)}\right)}{\sqrt{2\pi\sigma^{2(t)}}\pi^{u(t)} \exp\left(-\mu^{(t)}\eta_{ij}^{(t)} - \boldsymbol{X}_{ij}^{(t)}\right) + 256\pi^{g(t)} \exp\left(\frac{-(\boldsymbol{O}_{ij} - \boldsymbol{U}_i^{(t)}\boldsymbol{V}_j^{(t)T})^2}{2\sigma^{2(t)}}\right)}, \\
\Phi_{ij}^{g(t+1)} &= \frac{256\pi^{g(t)} \exp\left(\frac{-(\boldsymbol{O}_{ij} - \boldsymbol{U}_i^{(t)}\boldsymbol{V}_j^{(t)T})^2}{2\sigma^{2(t)}}\right)}{\sqrt{2\pi\sigma^{2(t)}}\pi^{u(t)} \exp\left(-\mu^{(t)}\eta_{ij}^{(t)} - \boldsymbol{X}_{ij}^{(t)}\right) + 256\pi^{g(t)} \exp\left(\frac{-(\boldsymbol{O}_{ij} - \boldsymbol{U}_i^{(t)}\boldsymbol{V}_j^{(t)T})^2}{2\sigma^{2(t)}}\right)}.
\end{aligned} \tag{14}$$

Θ **sub-problem:** Here, we focus on updating the parameters of the mixed model including σ^2, π^g and π^u. The update can be directly calculated via setting the derivatives of $\mathcal{Q}$ with respect to σ^2, π^g and π^u, respectively, to be zero. More specifically, each of σ^2, π^g and π^u updates via:

$$\begin{aligned}
\pi^{g(t+1)} = \frac{\sum_{ij}\Phi_{ij}^{g(t+1)}}{mn}, \quad \pi^{u(t+1)} = \frac{\sum_{ij}\Phi_{ij}^{u(t+1)}}{mn}, \\
\sigma^{2(t+1)} = \frac{\sum_{ij}\Phi_{ij}^{g(t+1)}(\boldsymbol{O}_{ij} - \boldsymbol{U}_i\boldsymbol{V}_j,)^2}{\sum_{ij}\Phi_{ij}^{g(t+1)}}.
\end{aligned} \tag{15}$$

$\boldsymbol{T}$ **sub-problem:** By discarding the constant terms, its closed form solution can be found by:

$$\begin{cases}
\boldsymbol{T}^{(t+1)} = \underset{\boldsymbol{T}}{\operatorname{argmin}}\, \mathcal{Q}(\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)}, \Theta^{(t+1)}, \Phi^{(t+1)}, \boldsymbol{T}, \boldsymbol{W}^{(t)}) \\
\qquad = \underset{\boldsymbol{T}}{\operatorname{argmin}}\, \lambda\|\boldsymbol{T}\|_1 + \frac{\mu^{(t)}}{2}\|\boldsymbol{T} - \boldsymbol{D}\boldsymbol{W}^{(t)}\|_F^2 + < \boldsymbol{Y}^{(t)}, \boldsymbol{T} - \boldsymbol{D}\boldsymbol{W}^{(t)} > \\
\qquad = \mathcal{S}_{\frac{\lambda}{\mu^{(t)}}}(\boldsymbol{D}\boldsymbol{W}^{(t)} - \frac{\boldsymbol{Y}^{(t)}}{\mu^{(t)}}),
\end{cases} \tag{16}$$

where $\mathcal{S}_{\varepsilon>0}(\cdot)$ represents the shrinkage operator, the definition of which on scalars is: $\mathcal{S}_{\varepsilon}(x) = \operatorname{sgn}(x)\max(|x| - \varepsilon, 0)$. The extension of the shrinkage operator to vectors and matrices is simply applied element-wisely.

***W* sub-problem:** As can be seen below, the $\boldsymbol{W}$ sub-problem is a classic least squares problem, thus the optimal $\boldsymbol{W}^{(t+1)}$ can be calculated easily.

$$\begin{cases} \boldsymbol{W}^{(t+1)} = \underset{\boldsymbol{W}}{\operatorname{argmin}}\, \mathcal{Q}(\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)}, \boldsymbol{\Theta}^{(t+1)}, \boldsymbol{\Phi}^{(t+1)}, \boldsymbol{T}^{(t+1)}, \boldsymbol{W}) \\ \quad = \underset{\boldsymbol{W}}{\operatorname{argmin}}\, \dfrac{\mu^{(t)}}{2}\|\boldsymbol{W} - \boldsymbol{\Phi}^{g(t+1)}\|_F^2 + < \boldsymbol{X}^{(t)}, \boldsymbol{W} - \boldsymbol{\Phi}^{g(t+1)} > \\ \qquad + \dfrac{\mu^{(t)}}{2}\|\boldsymbol{T}^{(t+1)} - \boldsymbol{D}\boldsymbol{W}\|_F^2 + < \boldsymbol{Y}^{(t)}, \boldsymbol{T}^{(t+1)} - \boldsymbol{D}\boldsymbol{W} > . \end{cases} \tag{17}$$

Traditionally, the optimal estimation of $\boldsymbol{W}^{(t+1)}$ can be simply obtained by computing $(\boldsymbol{I} + \boldsymbol{D}^T\boldsymbol{D})^{-1}(\boldsymbol{D}^T(\boldsymbol{T}^{(t+1)} + \frac{\boldsymbol{Y}^{(t)}}{\mu^{(t)}}) + \boldsymbol{\Phi}^{g(t+1)} - \frac{\boldsymbol{X}^{(t)}}{\mu^{(t)}})$. But, due to the size of matrix $(\boldsymbol{I} + \boldsymbol{D}^T\boldsymbol{D})$, it is computationally expensive to compute its inverse. Thanks to the block circulant structure of the matrix, it can be efficiently and exactly solved through applying 3D FFT on it, like:

$$\boldsymbol{W}^{(t+1)} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\boldsymbol{D}^T(\boldsymbol{T}^{(t+1)} + \frac{\boldsymbol{Y}^{(t)}}{\mu^{(t)}}) + \boldsymbol{\Phi}^{g(t+1)} - \frac{\boldsymbol{X}^{(t)}}{\mu^{(t)}})}{\mathbf{1} + |\mathcal{F}(\boldsymbol{D}_h)|^2 + |\mathcal{F}(\boldsymbol{D}_v)|^2 + |\mathcal{F}(\boldsymbol{D}_t)|^2}\right), \tag{18}$$

where $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$ stand for the 3D Fourier transform and the inverse 3D Fourier transform operators, respectively. $|\cdot|^2$ is the element-wise square and the division also performs element-wisely. Notice that the denominator in (18) only needs to be computed once.

***U*-*V* sub-problem:** In this sub-problem, we jointly seek the optimal solutions for $\boldsymbol{U}^{(t+1)}$ and $\boldsymbol{V}^{(t+1)}$. By keeping the elements related to $\boldsymbol{U}$ and $\boldsymbol{V}$ and dropping the others, the sub-problem can be rewritten as:

$$\begin{cases} (\boldsymbol{U}^{(t+1)}, \boldsymbol{V}^{(t+1)}) = \underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}}\, \mathcal{Q}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Theta}^{(t+1)}, \boldsymbol{\Phi}^{(t+1)}, \boldsymbol{T}^{(t+1)}, \boldsymbol{W}^{(t+1)}) \\ = \underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}} \sum_{ij} \dfrac{\boldsymbol{\Phi}_{ij}^{g(t+1)}}{2\sigma^{2(t+1)}}(\boldsymbol{O}_{ij} - \boldsymbol{U}_i\boldsymbol{V}_j^T)^2 = \underset{\boldsymbol{U},\boldsymbol{V}}{\operatorname{argmin}} \|\Omega \odot (\boldsymbol{O} - \boldsymbol{U}\boldsymbol{V}^T)\|_F^2, \end{cases} \tag{19}$$

where $\odot$ represents the Hadamard product of two matrices, Ω performs as a weight matrix[3] and $\Omega_{ij} = \sqrt{\boldsymbol{\Phi}_{ij}^{g(t+1)}/(2\sigma^{2(t+1)})}$. We can see that (19) is actually the weighted ℓ^2 low rank matrix factorization problem. The update of $\boldsymbol{U}$ and $\boldsymbol{V}$ can follow the rules introduced in existing methods, such as WLRA [16].

Besides, there are two multipliers and μ need to be updated, which can be simply done via:

$$\begin{aligned} \boldsymbol{X}^{(t+1)} &= \boldsymbol{X}^{(t)} + \mu^{(t)}(\boldsymbol{W}^{(t+1)} - \boldsymbol{\Phi}^{g(t+1)}), \\ \boldsymbol{Y}^{(t+1)} = \boldsymbol{Y}^{(t)} + \mu^{(t)}(\boldsymbol{T}^{(t+1)} - \boldsymbol{D}\boldsymbol{W}^{(t+1)}), &\ \mu^{(t+1)} = \min\{\rho\mu^{(t)}, \zeta\}, \end{aligned} \tag{20}$$

where $\rho > 1$ is a constant, and ζ is a predefined threshold (*e.g.* 10).

[3] By a slight abuse of notations, we reuse the notation Ω to represent the weight matrix for being consistent with the concept of support.

Algorithm 1: Robust Foreground Detection

Input: The observation $\boldsymbol{O}$
Initialization: Randomly initialize $\boldsymbol{U}^{(0)}$, $\boldsymbol{V}^{(0)}$, $\boldsymbol{\Phi}^{g(0)} = a\mathbf{1}, a \in (0,1)$, $\pi^{g(0)} \in (0,1)$ and $\sigma^2 > 0$. Set $\boldsymbol{T}^{(0)}$, $\boldsymbol{W}^{(0)}$, $\boldsymbol{X}^{(0)}$, $\boldsymbol{Y}^{(0)}$ to be zero matrices. $\boldsymbol{\Phi}^{u(0)} = \mathbf{1} - \boldsymbol{\Phi}^{g(0)}$, $\pi^{u(0)} = 1 - \pi^{g(0)}$. Compute $|\mathcal{F}(\boldsymbol{D}_h)|^2$, $|\mathcal{F}(\boldsymbol{D}_v)|^2$ and $|\mathcal{F}(\boldsymbol{D}_t)|^2$, $\mu^{(0)} = 0.1$, $\rho = 1.25$, $t = 0$.
while *not converged* **do**
  Update $\boldsymbol{\Phi}^{(t+1)}$ via Eq. (14);
  Update $\boldsymbol{\Theta}^{(t+1)}$ via Eq. (15);
  Update $\boldsymbol{T}^{(t+1)}$ via Eq. (16);
  Update $\boldsymbol{W}^{(t+1)}$ via Eq. (18);
  Update $\boldsymbol{U}^{(t+1)}$ and $\boldsymbol{V}^{(t+1)}$ via the rules introduced in [16];
  Update $\boldsymbol{X}^{(t+1)}$, $\boldsymbol{Y}^{(t+1)}$ and $\mu^{(t+1)}$ via Eq. (20);
  $t = t + 1$;
end
Output: Optimal solution $(\boldsymbol{U}^{(t)}, \boldsymbol{V}^{(t)}, \boldsymbol{\Phi}^{(t)}, \boldsymbol{\Theta}^{(t)})$.

For clarity, we summarize the whole procedure of optimization in Algorithm 1. The algorithm terminates when $\|\boldsymbol{\Phi}^{g(t+1)} - \boldsymbol{\Phi}^{g(t)}\|_F \leq \delta \|\boldsymbol{\Phi}^{g(t)}\|_F$ with $\delta = 10^{-3}$, or the maximal number of iteration is reached. Please notice that, as the elements in $\boldsymbol{\Phi}^u$ are real numbers rather than binaries, but very close to either 0 or 1. For foreground detection, the binary mask is required. To this end, we simply predefine a threshold (0.9 for all of experiments) to binarize $\boldsymbol{\Phi}^u$.

3 Simulations and Experiments

To quantitatively and qualitatively evaluate the performance of our proposed method, we conduct simulations on synthetic data and experiments on real sequences in this section. The simulations concentrate on revealing the effect of parameters involved in Algorithm 1, the convergence with random initializations and the robustness to different types of noise, such as Speckle, Gaussian, Salt & Pepper and Poisson. We also perform experiments on real videos, allowing us to compare against a large number of alternative approaches.

Parameter Effect. There are two parameters, *i.e.* λ and r, involved in our algorithm. In this part, we focus on evaluating the effect of r with $\lambda = 5$ fixed, while the effect of λ is tested latter together with the robustness to different noises. To better visualize the data, the observation is composed of 1D images. The background matrix is generated via $\boldsymbol{B}_0 = \boldsymbol{U}_0 \boldsymbol{V}_0^T$, where both $\boldsymbol{U}_0$ and $\boldsymbol{V}_0$ are $1000 \times Rank$ (an example background matrix is shown in Fig. 1 (f) corresponding to the case bounded by the dashed box in Fig. 1 (b)), and the observation (Fig. 1 (d)) is obtained by adding the foreground with a mask (Fig. 1 (e)) to the background. The foreground on each column shifts downward for 1 pixel per column. The foreground ratio controls the foreground width, taking the case with foreground ratio 0.4 for example (Fig. 1 (e)), the maximal width in the observation is actually 0.8. In addition, the entries in both the background and the

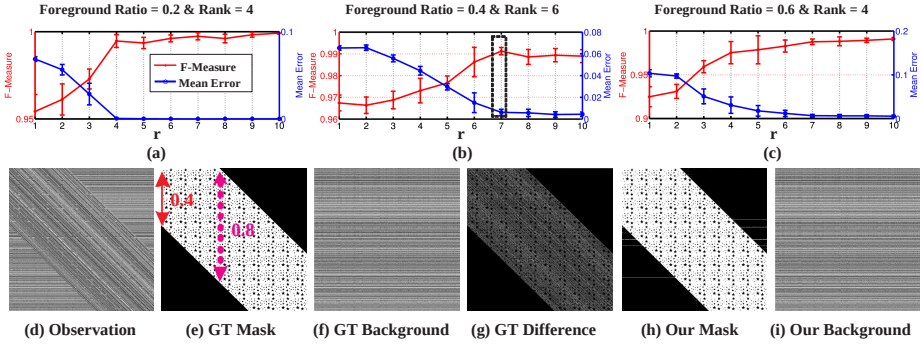

Fig. 1. The effect of parameter r. (a)-(c) are the results with respect to (Foreground Ratio: 0.2, Ground Truth Rank: 4), (0.4, 6) and (0.6, 4), respectively. (d)-(i) correspond to one trial of the case bounded by the dashed box in (b). The recovered background (i) is with only 0.0083 Mean Error and the estimated mask (g) is with 0.9906 F-Measure.

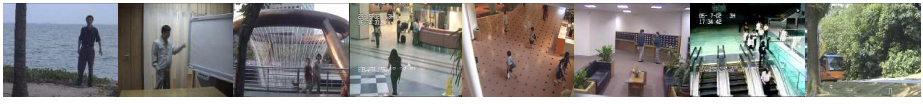

Fig. 2. Sample images from each sequence of the Star dataset.

foreground are sampled from the uniform distribution $[0, 255]$. To quantitatively reveal the recovery performance of both the background and the foreground mask, we employ Mean Error[4] and F-Measure[5] as our metrics. For each certain foreground ratio and r, we independently execute the algorithm for 10 times, and the average results are reported in Fig. 1 (a)-(c), as can be seen from which, we find that as r increases to the ground truth rank, the F-Measure and the Mean Error sharply increases and decreases, respectively. After that, the performance changes very smoothly in a relatively large range, which indicates the insensitivity of r. For the rest experiments, we set $r = 7$.

Performance Comparison on Real Sequences. We here compare our method against numerous state of the art approaches, *i.e.* GMM [18], SOBS [12], DP-GMM [6], PCP [2], DECOLOR [29] and GRASTA [7], on the dataset Star [9]. The Star contains 9 real world videos, which has a variety of scenarios including WaterSurface (*WS: outdoor, dynamic background, lingering person*), Curtain (*Cur: indoor, light switch, people*), Fountain (*Fou: outdoor, dynamic background, people*), Hall (*Hal: indoor, static background, crowd*), Lobby (*Lob: indoor, light switch, people*), Escalator (*Esc: indoor, dynamic background, crowd*), BootStrap (*BS: indoor, static background, crowd*) and Trees (*Tre: outdoor, dynamic back-*

[4] Mean Error is computed via computing the mean of the absolute sum of pixel-wise differences between the recovered background and the ground truth.

[5] F-Measure is defined as 2·precision·recall/(precision+recall).

method	*WS*	*Cur*	*Fou*	*Hal*	*SM*	*Lob*	*Esc*	*BS*	*Tre*	*mean*
GMM [18]	.7948	.7580	.6854	.3335	.5363	.6519	.1388	.3838	.0757	.4842
SOBS [12]	.8247	.8178	.6554	.5943	.6677	.6489	.5770	.6019	.6960	.6760
DP-GMM [6]	**.9090**	.8203	.7049	.5484	.6522	.5794	.5055	.6024	.7567	.6754
PCP [2]	.4137	.6193	.5679	.5917	.7234	.6989	.6728	.6582	.3406	.5874
DECOLOR [29]	.8866	.8255	**.8598**	.6424	.6525	.6149	**.6994**	.5869	**.8096**	.7308
GRASTA [7]	.7310	.6591	.3786	.5817	.7142	.5550	.4697	.6146	.2504	.5505
Our Method	.8796	**.8976**	.7544	**.6673**	**.7407**	**.8029**	.6353	**.6841**	.6779	**.7489**

Table 1. Performance comparison in terms of F-Measure.

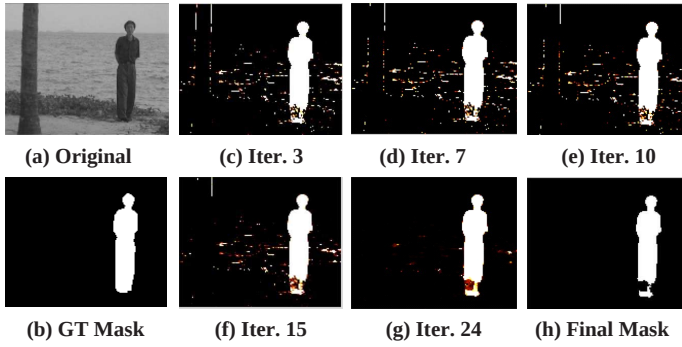

Fig. 3. Mask Evolution. (a) the original frame. (b) the Ground Truth (GT) mask. (c)-(g) are the estimations of the 3^{th}, 7^{th}, 10^{th}, 15^{th} and 24^{th} iterations, respectively. (h) is the final mask. Lighter colors in (c)-(g) indicate higher possibilities of being foreground, while darker ones stand for lower.

ground, cars). Sample images from each sequence of Star are provided in Fig. 2. In the comparison, we fix $\lambda = 5$, which is suggested by the Robustness analysis detailed latter. As for the other competitors, the codes are downloaded from the authors' websites and the parameters are all set as default. Please note that, for PCP and GRAST, the raw results are the residuals instead of the masks, so the post processing of binarization is essential to give the final foreground masks. To this end, we assume the absolute residuals in each frame satisfy a Gaussian distribution $\mathcal{N}(\tilde{\eta}, \tilde{\sigma})$. Therefore, we adaptively compute the threshold $t = \tilde{\eta} + \tilde{\sigma}$. With the threshold, the estimation of the mask is done via: if the absolute residual is larger than t, then the mask is 1, otherwise 0. The performance comparison in terms of F-Measure is given in Table 1, as can be viewed from which, our method performs stably and robustly for every sequence with high F-Measure. The average F-Measure of our method over the whole dataset is the clear evidence demonstrating the proposed method significantly outperforms the others. Figure 3 displays the changes of Φ^u (mask), without loss of generality, on the sequence of WaterSurface, as our algorithm iterates. Figure 3 (c)-(g) reveal that the estimation gradually eliminates the noise and becomes

stable. From Fig. 3 (h), we observe that the final mask obtained by our method is very close to the GT mask. Note that, although the F-Measure of DECOLOR is slightly behind our method, its computational cost is much higher than ours, please see the comparison in the next paragraph.

Time Comparison. We here provide a comparison of time among our method, PCP [2] and DECOLOR [29], which are all implemented in Matlab (the core part of DECOLOR is implemented in C++). The time comparison is conducted on a PC running Windows 7 32bit operating system with Intel Core i7 3.4 GHz CPU and 4.0 GB RAM. As can be seen from Fig. 4, as the matrix size increases, the time required by DECOLOR quickly grows up. In contrast with DECOLOR, PCP and our method are much lighter. For instance, in the case of the matrix size with 1000×1000, both PCP and our method spend about $24s$ while DECOLOR costs about $90s$. Please notice that, the computational time of DECOLOR with respect to 1500×1500 is not provided in Fig. 4, since DECOLOR runs out of memory. In other words, our method requires less memories than DECOLOR. For the larger matrix (1500×1500), as shown in the picture, the time load of PCP becomes much heavier than that of our method.

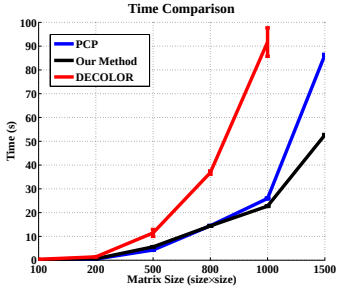

Fig. 4: Time Comparison

Convergence. The initialization of $\boldsymbol{U}^{(0)}$, $\boldsymbol{V}^{(0)}$, $\Phi^{g(0)}$, $\pi^{g(0)}$ and σ^2 in Algorithm 1 is random. To reveal the proposed algorithm can converge efficiently and stably with random initializations, we test 5 different initializations on, without loss of generality, the *WS* sequence. It is hard to visualize data with dimensions more than 3. Therefore, in Fig. 5, we choose π^u, σ^2 and iteration number to show, which is also capable to reflect the converging trend and speed of Algorithm 1. The initializations are indeed scattered in the σ^2-π^u plane at the beginning of loop, as shown in Fig. 5. Rapidly, within only 3 iterations, the 5 curves are gathering as shown in the upper ($4\times$) zoomed in patch. At the 5^{th} iteration, we can see, from the lower ($16\times$) zoomed in patch, that the algorithm gives almost the same estimations of σ^2 and π^u for different cases. In all the experiments present in this paper, we set the maximal number of iteration to 35, which works sufficiently well.

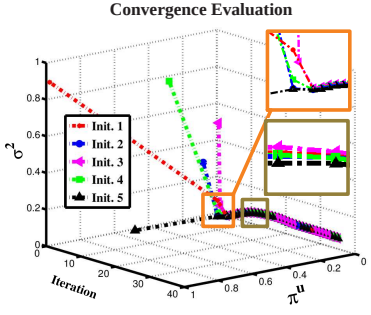

Fig. 5: Convergence Evaluation

Robustness to Noise. Although we simply assume the residual caused by the factors expect for the foreground satisfies a Gaussian distribution, the smoothness can make the algorithm survives from various noise types. This part aims to demonstrate the robustness of our method to different types of noise including Speckle, Gaussian, Salt & Pepper and Poisson with different levels. The

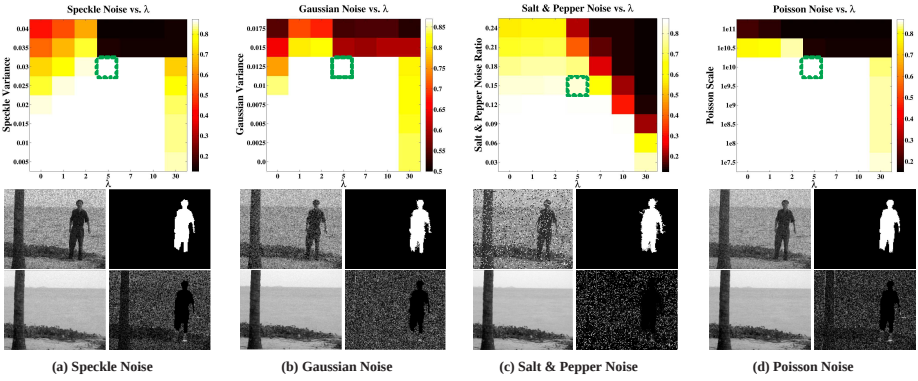

Fig. 6. Robustness to different noise types with varying λ values. The upper row gives the F-Measures, while the lower row shows four groups of visual results corresponding to the cases marked by green dashed boxes. Each group has four sub-images: the sample input (top-left), the estimated foreground mask (top-right), the recovered background (bottom-left) and the noise component (bottom-right). The noise level differs for different types of noises, please refer to their definitions.

upper row of Figure 6 shows the F-Measures with respect to different types of noise. As can be seen in Fig. 6 (a), as λ increases to 10, the speckle noise variance that our method can handle reaches 0.03. But when $\lambda = 30$, the performance drops because the results are over smoothed, which is further confirmed by the rest cases. Figure 6 (b) corresponds to the white Gaussian noise, from which we can find that the proposed algorithm robustly processes the noise on the 0.0125 level using a large value range of λ. Even though Salt & Pepper noise heavily deviates from the Gaussian assumption, Figure 6 (c) shows that, by setting $\lambda = 5$, our method is robust to process the case with 15% pixels polluted by such noise. The reason why the F-Measure decreases as the λ increases from 5 is that the smoothness makes masks dilated or eroded. For the Poisson noise, the results given in Fig. 6 (d) demonstrate the robustness of our method to the 10^{10} scale with λ under 10. Since the definitions of these noises are different, so are the noise levels. The lower row of Fig. 6 offers the visual results corresponding to the cases marked by the dashed green boxes in the upper row, from which we can see that even though the input images are severely perturbed by noises, our method can robustly recover the backgrounds, the foreground masks and the noise components. Please zoom in to see more details.

4 Conclusion and Future Work

In real world scenarios, foreground detection is difficult as videos may contain not only static backgrounds, but also noises, dynamic backgrounds and camouflage foregrounds. In this paper, we have shown how to harness three structural priors of the background and foreground, including the arbitrariness of foreground

appearance, the spatial-temporal smoothness of foreground, and the correlation of background, for robustly detecting foreground objects, which are achieved by assuming the foreground satisfies the uniform distribution, smoothing the foreground via total variation regularization, and imposing the correlation constraint on the background, respectively. We have formulated the problem in a unified optimization framework and proposed a novel ALM-ADM based algorithm to effectively and efficiently seek the optimal solution. The effect of parameters, the convergence and the robustness to noises of the proposed algorithm have been analyzed. Besides, compared to the state of the art alternatives, the experimental results on real sequences have demonstrated the clear advantages of the proposed method in terms of accuracy, robustness and efficiency.

Currently, our algorithm processes sequences in a batch fashion. For applications with the online requirement, it is more desirable to design its online version, which actually can be done by reconstructing a new coming frame using the existing background model, referring to the foreground mask of the latest frame to achieve the temporal smoothness, and then updating the model in an incremental way. We leave this extension as our future work.

Acknowledgment

X. Guo was supported by Excellent Young Talent of the Institute Information Engineering, Chinese Academy of Sciences. X. Wang was supported by Microsoft Research Asia Fellowship 2012. X. Cao was supported by National Natural Science Foundation of China (No.61332012), National Hightech R&D Program of China (2014BAK11B03), and 100 Talents Programme of The Chinese Academy of Sciences.

References

1. Babenko, B., Yang, M., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE TPAMI 33(8), 1619–1632 (2011) 1
2. Candès, E., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM 58(3), 1–37 (2011) 3, 10, 11, 12
3. Cremers, D., Soatto, S.: Motion competition: A variational approach to piecewise parametric motion segmentation. IJCV 62(3), 249–265 (2005) 2
4. Cristani, M., Raghavendra, R., Bue, A.D., Murino, V.: Human behavior analysis in video surveillance: a social signal processing perspective. Neurocomputing 100, 86–97 (2013) 1
5. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of IEEE 90(7), 1151–1163 (2002) 2
6. Haines, T., Xiang, T.: Background subtraction with dirichlet process mixture models. IEEE TPAMI (2014) 3, 10, 11
7. He, J., Balzano, L., Szlam, A.: Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In: CVPR. pp. 1568–1575 (2012) 3, 10, 11

8. Lee, D.: Effective gaussian mixture learning for video background subtraction. IEEE TPAMI 27(5), 827–832 (2005) 2
9. Li, L., Huang, W., Gu, I., Tian, Q.: Statistical modeling of complex backgrounds for foreground object detection. IEEE TIP 13(11), 1459–1472 (2004) 10
10. Lin, Z., Chen, M., Wu, L., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Tech. Rep. UILU-ENG-09-2215, UIUC Technical Report (2009) 6
11. Liu, J., Tai, X., Huang, H., Huan, Z.: A weighted dictionary learning model for denoising images corrupted by mixed noise. IEEE TIP 22(3), 1108–1120 (2013) 4
12. Maddalena, L., Petrosino, A.: A self-organizing approach to background subtraction for visual surveillance applications. IEEE TIP 17(7), 1168–1177 (2008) 2, 10, 11
13. Meng, D., De la Torre, F.: Robust matrix factorization with unknown noise. In: ICCV. pp. 1337–1344 (2013) 3
14. Peng, Y., Ganesh, A., Wright, J., Xu, W., Ma, Y.: Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. IEEE TPAMI 34(11), 2233 – 2246 (2012) 3
15. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena 60(1), 259–268 (1992) 3
16. Srebro, N., Jaakkola, T.: Weighted low-rank approximations. In: ICML. pp. 720–727 (2003) 8, 9
17. Stalderand, S., Grabner, H., Van Gool, L.: Cascaded confidence filtering for improved tracking-by-detection. In: ECCV. pp. 269–382 (2010) 1
18. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: CVPR. pp. 246–252 (1999) 2, 10, 11
19. Vidal, R.: Subspace clustering. IEEE Signal Processing Magazine 28(2), 52–68 (2011) 2
20. Vidal, R., Ma, Y.: A unified algebraic approach to 2-d and 3-d motion segmentation. In: ECCV. pp. 1–15 (2004) 2
21. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. IJCV 63(2), 153–161 (2005) 1
22. Wang, N., Yao, T., Wang, J., Yeung, D.: A probabilistic approach to robust matrix factorization. In: ECCV. pp. 126–139 (2012) 3
23. Wang, N., Yeung, D.: Bayesian robust matrix factorization for image and video processing. In: ICCV. pp. 1785–1792 (2013) 3
24. Wren, C., Azarbayejani, A., Darrell, T., Pentland, A.: Pfinder: Realtime tracking of the human body. IEEE TPAMI 19(7), 780–785 (1997) 2
25. Xu, J., Ithapu, V., Mukherjee, L., Rehg, J., Singh, V.: GOSUS: Grassmanian online subspace updates with strutured-sparsity. In: ICCV. pp. 3376–3383 (2013) 3
26. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4), 1–45 (2006) 1
27. Zhang, Z., Ganesh, A., Liang, X., Y.Ma: Tilt: Transform invariant low-rank textures. IJCV 99(1), 1–24 (2012) 3
28. Zhou, T., Tao, D.: Godec: Randomized low-rank & sparse matrix decomposition in noisy case. In: ICML. pp. 33–40 (2011) 3
29. Zhou, X., Yang, C., Yu, W.: Moving object detection by detecting contiguous outliers in the low-rank representation. IEEE TPAMI 35(3), 597–610 (2013) 3, 10, 11, 12