AN IMPROVED TOTAL VARIATION REGULARIZED RPCA FOR MOVING OBJECT DETECTION WITH DYNAMIC BACKGROUND

XIANCHAO XIU AND YING YANG∗
State Key Lab for Turbulence and Complex Systems
Department of Mechanics and Engineering Science, College of Engineering
Peking University, China

WANQUAN LIU
Department of Computing
Curtin University, Australia

LINGCHEN KONG
Department of Applied Mathematics
Beijing Jiaotong University, China

MEIJUAN SHANG
College of Science
Shijiazhuang University, China

(Communicated by Cedric Yiu)

ABSTRACT. Dynamic background extraction has been a fundamental research topic in video analysis. In this paper, a novel robust principal component analysis (RPCA) approach for foreground extraction is proposed by decomposing video frames into three parts: rank-one static background, dynamic background and sparse foreground. First, the static background is represented by a rank-one matrix, which can avoid the computation of singular value decomposition because usually the dimensionality of a surveillance video is very large. Secondly, the dynamic background is characterized by \( \ell_{2,1} \)-norm, which can exploit the shared information. Thirdly, the sparse foreground is enhanced by total variation, which can preserve sharp edges that are usually the most important for clear object extraction. An efficient symmetric Gauss-Seidel based alternating direction method of multipliers (sGS-ADMM) is established with convergence analysis. Extensive experiments on real-world datasets show that our proposed approach outperforms the existing state-of-the-art approaches. In fact, to the best of our knowledge, this is the first time to integrate the joint sparsity and total variation into a RPCA framework, which has demonstrated the superiority of performance.

2010 Mathematics Subject Classification. Primary: 90C25, 90C90; Secondary: 65K10.
Key words and phrases. Dynamic background extraction, rank-one matrix, joint sparsity, total variation, alternating direction method of multipliers.

This work was supported in part by the National Natural Science Foundation of China (61633001, 11671029, 11601348) and the 111 Project of China (B16002).

∗ Corresponding author: Ying Yang.
1. Introduction. In surveillance, video signals are usually captured by spot cameras and transferred to a data processing center. The number of cameras has increased dramatically over the decade, which results in a huge amount of data. Moreover, there still exist many challenges for video analysis in our real life environment (e.g., irregular object movements, low contrast, high sensor noise, camouflage, shelters, bad weather, and so on). Therefore, how to extract the foreground (moving objects) in surveillance videos quickly and precisely has become a significant problem.

The traditional approaches for foreground extraction are developed with the help of using frame difference, such as the nonparametric models like kernel density estimation [12], neural network models [22] and mixture of Gaussian models [28]. However, all these approaches usually result in local misclassifications, and also cost plentiful time due to excessive processing or management for all pixels of surveillance videos. See, e.g., [1, 27] for comprehensive reviews.

Recently, the techniques based on Compressed Sensing (CS) have developed quickly due to their excellent performance. The foreground extraction in this framework was initially formulated as the following optimization problem:

$$\min_{L,S} \text{rank}(L) + \lambda \|S\|_0 \quad \text{s.t.} \quad L + S = M, \quad (1)$$

where $\|S\|_0$ is the cardinality function of $S$, which denotes the number of nonzero entries, and $\lambda > 0$ is a regularization parameter. The data matrix $M \in \mathbb{R}^{m \times n}$ can be decomposed (approximately) as the sum of a low rank matrix $L \in \mathbb{R}^{m \times n}$ modeling the background and a sparse matrix $S \in \mathbb{R}^{m \times n}$ modeling the foreground.

The background component has relatively static small changes over a period, so it is modeled as a low rank matrix $L$. The foreground component consists of moving objects, modeled as sparse matrix $S$ with most of its entries being zero or nearly zero. The low rank matrix can capture the background of the surveillance video, while the sparse matrix can capture the foreground.

In fact, the problem (1) is NP-hard due to the discontinuity and nonconvexity of rank($L$) and $\lambda \|S\|_0$. See, e.g., [5, 9, 10, 23]. In order to solve this problem in a tractable way, Candès et al. [4] proposed the robust principal component analysis (RPCA) model where they replaced the rank with nuclear norm and $\ell_0$ norm with $\ell_1$ norm, that is,

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad L + S = M, \quad (2)$$

in which $\|L\|_*$ is the nuclear norm (defined as the sum of all singular values) of $L$, $\|S\|_1$ is the $\ell_1$ norm in componentwise (defined as the sum of absolute values of all entries). This new model is an approximation for the original problem (1) and further, it is proved that under rather weak assumptions, the solutions of $L$ and $S$ in (1) can be recovered exactly via solving (2). Now the RPCA is one of the most popular approaches for foreground extraction [2].

However, the RPCA currently suffers from some prominent issues. For example, the extracted foreground always has small artifacts, which in fact belong to the background. To overcome this obvious shortcoming of the RPCA, much effort has been devoted in research community. One interesting work is to separate the foreground $S$ into dynamic background part $E$ and the intrinsic foreground part $F$,
and then the optimization model can be described as
\[
\min_{L,S,E,F} \|L\|_* + \lambda_1\|S\|_1 + \lambda_2\|E\|_1 + \lambda_3\|DF\|_1
\]
\[
\text{s.t. } L + S = M, E + F = S,
\]
which is called total variation regularized RPCA (TVRPCA) in [6]. Here, \(\lambda_1, \lambda_2, \lambda_3\) are the weights for balancing the corresponding terms, and \(\|DF\|_1\) is the three-dimensional total variation norm defined as
\[
\|DF\|_1 = \|D_h F\|_1 + \|D_v F\|_1 + \|D_t F\|_1,
\]
and the three difference operations at the voxel \((i,j,t)\) along the horizontal, vertical, and temporal directions are
\[
(D_h F)(i,j,t) = F(i,j,t) - F(i+1,j,t),
\]
\[
(D_v F)(i,j,t) = F(i,j,t) - F(i,j+1,t),
\]
\[
(D_t F)(i,j,t) = F(i,j,t) - F(i,j,t+1).
\]
As is shown in [25, 26], the added total variation term can remove small artifacts and make images much more clean.

Even though [6] established an alternating direction method of multipliers based algorithm, they didn’t present the convergence result. In fact, the convergence of the algorithm can not be guaranteed [7]. In this paper, we attempt to fill this gap and propose an improved total variation regularized RPCA model:
\[
\min_{u,S,E,F} \|S\|_1 + \lambda_1\|E\|_{2,1} + \lambda_2\|DF\|_1
\]
\[
\text{s.t. } u^T + S = M, E + F = S,
\]
where \(u \in \mathbb{R}^m\), \(1 \in \mathbb{R}^n\) are the decomposition of rank-one matrix \(L\), \(\|E\|_{2,1}\) is defined as the sum of absolute values of the \(\ell_2\)-norm of all rows of the matrix \(E\). The model is called the rank-one and joint sparsity matrix decomposition.
Compared to previous work, the main innovations of our paper are summarized below:

1. The joint sparsity [21, 31] is applied to exploit the structure of the dynamic background part $E$, see Figure 1 for detailed illustrations. Also we reduce the number of hyper-parameters, which brings great benefits to save the computational time.

2. The rank-one approximation [19, 29] is used to replace the low-rank part $L$. The motivation behind is that for a surveillance video with a static background, the static background matrix should consist of identical columns, instead of being a general low-rank matrix.

3. An efficient symmetric Gauss-Seidel based alternating direction method of multipliers (sGS-ADMM) with convergence guarantee is established to optimize our proposed model, which can take full advantage of the special structure. Furthermore, the step-length $\tau$ should be in $(0, (1 + \sqrt{5})/2)$, rather than $\tau = 1$ as discussed in [6]. In practice, we always choose $\tau = 1.618$.

The remaining of this paper is organized as follows. In Section 2, we summarize some preliminaries which are useful for further discussion. In Section 3, we propose an efficient algorithm for our proposed model, as well as the convergence result. Then, we report some numerical experiments on various surveillance video datasets to substantiate the superiority of the proposed model over the other existing ones in Section 4. Finally, we make some conclusions in Section 5.

2. Preliminaries. In this section, we summarize some preliminaries which are useful for further discussions. First, we briefly review on the ADMM, which is widely used in computer vision, image processing and statistical learning [3]. Let us start from a generic convex minimization problem with a separable objective function and linear constraints:

$$
\min f(X) + g(Y)
\text{s.t. } A(X) + B(Y) = Z,
$$

where $X, Y \in \mathbb{R}^{m \times n}$ are two variables, $Z \in \mathbb{R}^{m \times n}$ is a given matrix, $f, g$ are closed proper convex functions [24], and $A, B : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ are linear maps. The augmented Lagrangian function for problem (5) is

$$
\mathcal{L}_\beta(X, Y; \Lambda) := f(X) + g(Y) - \langle \Lambda, A(X) + B(Y) - Z \rangle + \frac{\beta}{2} \|A(X) + B(Y) - Z\|^2_F,
$$

where $\beta > 0$ is a penalty parameter, $\Lambda \in \mathbb{R}^{m \times n}$ is the Lagrangian multiplier, and $\langle \cdot, \cdot \rangle$ is the inner product of two matrices. To exploit the properties of $f$ and $g$ individually, the ADMM approach, which was originally proposed in [15], can be summarized in Algorithm 1. In most cases, the resulting subproblems admit closed-form solutions, which makes the ADMM particularly efficient. Under some mild conditions, the sequence $\{X^k, Y^k\}$ generated by the ADMM can be shown to converge to an optimal solution of (5), see, e.g., [13, 14, 17].

However, when there exist more than two variables, the ADMM does not have a convergent guarantee. Recently, it is proved in [7] that the ADMM, when applied to a convex optimization problem with three variables, can be divergent. But, it really works well in [6] and other applications. This motivates the study of many provably convergent variants of the ADMM for convex problems with more than two variables, see [8, 16, 18, 20] for example. Let us consider the following convex
Algorithm 1: ADMM for model (5)
Let $\tau \in (0, (1 + \sqrt{5})/2)$ be a step-length and $\beta > 0$ be a given parameter. Choose initial point $(Y^0, \Lambda^0) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n}$.

while not converge do

Step 1. Compute $X^{k+1} = \arg \min_X L_\beta(X, Y^k; \Lambda^k)$;

Step 2. Compute $Y^{k+1} = \arg \min_Y L_\beta(X^{k+1}, Y; \Lambda^k)$;

Step 3. Update $\Lambda^{k+1} = \Lambda^k - \tau \beta (A(X^{k+1}) + B(Y^{k+1}) - Z)$;

end while

composite programming whose objective involves a nonsmooth term:

$$
\min_{X} \sum_{i=1}^{p} f_i(X_i) + \sum_{j=1}^{q} g_j(Y_j)
$$

s.t. $\sum_{i=1}^{p} A_i(X_i) + \sum_{j=1}^{q} B_j(Y_j) = Z,$

(6)

where $X_i, Y_j \in \mathbb{R}^{m \times n}$ are variables, $Z \in \mathbb{R}^{m \times n}$ is a given matrix, $f_1, g_1$ are nonsmooth convex functions, $f_i, g_j (i = 2, \ldots, p; j = 2, \ldots, q)$ are smooth convex functions, and $A_i, B_j : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ are linear maps. The augmented Lagrangian function for problem (6) is defined as

$$
L_\beta(X_i, Y_j; \Lambda) := \sum_{i=1}^{p} f_i(X_i) + \sum_{j=1}^{q} g_j(Y_j) - \langle \Lambda, \sum_{i=1}^{p} A_i(X_i) + \sum_{j=1}^{q} B_j(Y_j) - Z \rangle
+ \frac{\beta}{2} \sum_{i=1}^{p} \|A_i(X_i)\|_F^2 + \sum_{j=1}^{q} \|B_j(Y_j) - Z\|_F^2.
$$

With the help of symmetric Gauss-Seidel (sGS) technique, the sGS-ADMM for (6) was developed in [8, 20] and can be summarized in Algorithm 2. The idea behind is that, to make full use of these smooth convex functions, we should first do backward GS sweep and then do forward GS sweep. In this paper, we use $k + \frac{1}{2}$ and $k + 1$ to denote the backward GS sweep and forward GS sweep, respectively. We refer to [20] for the convergence of Algorithm 2. This gives us a theoretical guarantee for the sGS-ADMM with more than two variables.

For the sGS-ADMM type algorithm to be presently, their main subproblems at each iteration have closed-form solutions. We then review two operators which will help us express the closed-form solutions conveniently.

Lemma 2.1. The minimization problem

$$
\min_X \|X\|_1 + \frac{\beta}{2} \|X - T\|_F^2
$$

with $\beta > 0$ and $X, T \in \mathbb{R}^{m \times n}$ has a closed-form solution which is given by the soft-shrinkage operator defined as

$$
X^* = \text{shrink}_{1,2}(T, 1/\beta) := \text{sign}(T) \circ \max\{0, |T| - 1/\beta\},
$$

where $\circ$ and $\text{sign}$ represent, respectively, the pointwise product and signum function, and all operations are done componentwise. See [11] for more details.
Algorithm 2: sGS-ADMM for model (6)

Let $\tau \in (0, (1 + \sqrt{5})/2)$ be a step-length and $\beta > 0$ be a given parameter. Choose initial point $(X_0^i, Y_0^j, \Lambda^0) \in \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n} \times \mathbb{R}^{m \times n}$.

while not converge do

Step 1a. (Backward GS sweep) Compute for $i = p, \ldots, 2$,

$$X^k_{i+2} = \arg\min_{X_i} L_\beta(X_i, Y_j^k; \Lambda^k);$$

Step 1b. (Forward GS sweep) Compute for $i = 1, \ldots, p$,

$$X^k_{i+1} = \arg\min_{X_i} L_\beta(X_i, Y_j^k; \Lambda^k);$$

Step 2a. (Backward GS sweep) Compute for $j = q, \ldots, 2$,

$$Y^k_{j+2} = \arg\min_{Y_j} L_\beta(X^k_{i+1}, Y_j; \Lambda^k);$$

Step 2b. (Forward GS sweep) Compute for $j = 1, \ldots, q$,

$$Y^k_{j+1} = \arg\min_{Y_j} L_\beta(X^k_{i+1}, Y_j; \Lambda^k);$$

Step 3. Update $\Lambda^{k+1}$ via

$$\Lambda^{k+1} = \Lambda^k - \tau \beta (\sum_{i=1}^p A_i (X^k_{i+1}) + \sum_{j=1}^q B_j (Y^k_{j+1}) - Z);$$

end while

Lemma 2.2. The minimization problem

$$\min_X \|X\|_{2,1} + \frac{\beta}{2} \|X - T\|_F^2$$

with $\beta > 0$ and $X, T \in \mathbb{R}^{m \times n}$ has a closed-form solution which is given by

$$X^* = \text{shrink}_{2,2}(T_i, 1/\beta) := \frac{T_i}{\|T_i\|_2} \circ \max\{0, \|T_i\|_2 - 1/\beta\},$$

where $T_i$ represents the $i$-th row of $T$ for $i = 1, \ldots, m$.

3. Algorithm and convergence analysis. In this section, we apply the sGS-ADMM idea to derive an efficient algorithm with convergence analysis for our proposed model (4). We first introduce an auxiliary variable $K$ and then change (4) into the following equivalent form

$$\min_{u,F,S,E,K} \|S\|_1 + \lambda_1\|E\|_{2,1} + \lambda_2\|K\|_1$$

s.t. $u1^T + S = M$,

$E + F - S = 0$,

$K - DF = 0$.

(7)

The auxiliary variable $K$ is to liberate $F$ from the nonsmooth term $\| \cdot \|_1$. Let $L_\beta(u, F, S, E, K; \Lambda_1, \Lambda_2, \Lambda_3)$ be the augmented Lagrangian function of problem in (7), which is defined as

$$L_\beta(u, F, S, E, K; \Lambda_1, \Lambda_2, \Lambda_3) = \|S\|_1 + \lambda_1\|E\|_{2,1} + \lambda_2\|K\|_1$$

$$-\langle \Lambda_1, u1^T + S - M \rangle + \frac{\beta_1}{2}\|u1^T + S - M\|_F^2$$

$$-\langle \Lambda_2, E + F - S \rangle + \frac{\beta_2}{2}\|E + F - S\|_F^2$$

$$-\langle \Lambda_3, K - DF \rangle + \frac{\beta_3}{2}\|K - DF\|_F^2.$$
in which $\Lambda_1, \Lambda_2, \Lambda_3 \in \mathbb{R}^{m \times n}$ are the Lagrange multipliers and $\beta_1, \beta_2, \beta_3$ are the tuning penalty parameters. As is illustrated in Section 2, when there exist more than two variables, the convergence of traditional ADMM can not be guaranteed. Hence, the symmetric Gauss-Seidel based ADMM (sGS-ADMM) should be adopted.

By grouping the variables into $(u, F)$ and $(S, E, K)$, the iterative scheme is as follows

\begin{align*}
\left( \begin{array}{c}
   u^{k+\frac{1}{2}} \\
   F^{k+\frac{1}{2}}
\end{array} \right) & \leftarrow \arg \min_{u,F} L_\beta(u, F, S^k, E^k, K^k; \Lambda^k_1, \Lambda^k_2, \Lambda^k_3); \\
S^{k+1} & \leftarrow \arg \min_{S,E,K} L_\beta(u^{k+\frac{1}{2}}, F^{k+\frac{1}{2}}, S, E, K; \Lambda^k_1, \Lambda^k_2, \Lambda^k_3); \\
E^{k+1} & \leftarrow \arg \min_{S,E,K} L_\beta(u^{k+\frac{1}{2}}, F^{k+\frac{1}{2}}, S, E, K; \Lambda^k_1, \Lambda^k_2, \Lambda^k_3); \\
K^{k+1} & \leftarrow \arg \min_{u,F} L_\beta(u, F, S^{k+1}, E^{k+1}, K^{k+1}; \Lambda^{k+1}_1, \Lambda^{k+1}_2, \Lambda^{k+1}_3);
\end{align*}

In the following, we will show that the minimization with respect to each variable can be separated into the following subproblems.

**Step 1a.** For variable $u$, the subproblem of $L_\beta$ with respect to $u$ can be solved by

$$u^{k+\frac{1}{2}} = \text{Mean}(M - S^k + \Lambda^k_1/\beta_1),$$

where the operator $\text{Mean} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^m$ is defined as

$$\text{Mean}(X)_i := \frac{1}{n} \sum_{j=1}^{n} X_{i,j}, \quad i = 1, \ldots, m, \quad \forall X \in \mathbb{R}^{m \times n}.$$

Compared with low-rank decomposition, rank-one can avoid the computation of SVD, which saves a lot of time.

**Step 1b.** For variable $F$, the subproblem of $L_\beta$ with respect to $F$ can be solved by the following equation

$$(\beta_2 I + \beta_3 D^T D)F^{k+\frac{1}{2}} = \beta_2 (S^k - E^k + \Lambda^k_2/\beta_2) + \beta_3 D^T (K^k - \Lambda^k_3/\beta_3).$$

Notice that the coefficient matrix $\beta_2 I + \beta_3 D^T D$ is nonsingular whenever $\beta_2, \beta_3 > 0$. As we all know, the Moore-Penrose pseudo-inverse can be computed via common linear system solvers, such as the Cholesky decomposition and the conjugate gradient method, but it will be extremely slow. Under the periodic boundary conditions for $F$, $D^T D$ is a block circulant matrix with circulant blocks and thus is diagonalizable by the 3D discrete Fourier transforms (DFTs). As a result, the $F^{k+\frac{1}{2}}$ can be computed by one forward DFT and one inverse DFT. Let $H = \beta_2 (S^k - E^k + \Lambda^k_2/\beta_2) + \beta_3 D^T (K^k - \Lambda^k_3/\beta_3)$, we have

$$F^{k+\frac{1}{2}} = \mathcal{F}^{-1} \left( \frac{\mathcal{F}(H)}{\beta_2 I + \beta_3 (|\mathcal{F}(D_h)|^2 + |\mathcal{F}(D_v)|^2 + |\mathcal{F}(D_l)|^2)} \right),$$

where $\mathcal{F}$ is the 3-dimensional DFT, $|\cdot|^2$ is the element-wise squares and the division also performs in element-wise.
Step 2a. For variable $S$, the optimization subproblem of $\mathcal{L}_\beta$ with respect to $S$ can be transformed as
\[
S^{k+1} = \arg \min_S \left\{ \|S\|_1 + \frac{\beta_1}{2}\|S - T_1^k\|_F^2 + \frac{\beta_2}{2}\|S - T_2^k\|_F^2 \right\}
\]
\[
= \arg \min_S \left\{ \|S\|_1 + \frac{\beta_1 + \beta_2}{2}\left\|S - \frac{\beta_1}{\beta_1 + \beta_2}T_1^k - \frac{\beta_2}{\beta_1 + \beta_2}T_2^k\right\|_F^2 \right\},
\]
where $T_1^k = M - u^{k+\frac{1}{2}}1^T + \Lambda_1^k/\beta_1$, and $T_2^k = E^k + F^{k+\frac{1}{2}} - \Lambda_2^k/\beta_2$. The solution is
\[
S^{k+1} = \text{shrink}_{1,2} \left( \frac{\beta_1}{\beta_1 + \beta_2}T_1^k + \frac{\beta_2}{\beta_1 + \beta_2}T_2^k, \frac{1}{\beta_1 + \beta_2} \right), \quad (10)
\]
which can be solved by the soft-shrinkage operator defined in Lemma 2.1.

Step 2b. For variable $E$, the subproblem of $\mathcal{L}_\beta$ with respect to $E$ can be simplified to
\[
E^{k+1} = \arg \min_E \left\{ \lambda_1\|E\|_{2,1} + \frac{\beta_2}{2}\|E + F^{k+\frac{1}{2}} - S^{k+1} - \Lambda_2^k/\beta_2\|_F^2 \right\}.
\]
From Lemma 2, the solution is
\[
(E^{k+1})_i = \text{shrink}_{2,2}((S^{k+1} - F^{k+\frac{1}{2}} + \Lambda_2^k/\beta_2)_i, \lambda_1/\beta_2). \quad (11)
\]

Step 2c. For variable $K$, the subproblem of $\mathcal{L}_\beta$ with respect to $K$ can be solved by
\[
K^{k+1} = \text{shrink}_{1,2}(DF^{k+1} + \Lambda_3^k/\beta_3, \lambda_2/\beta_3). \quad (12)
\]

Step 3a. Similarly to Step 1a, it is
\[
u^{k+1} = \text{Mean}(M - S^{k+1} + \Lambda_1^k/\beta_1). \quad (13)
\]

Step 3b. Similarly to Step 1b, it can be solved by the following equation
\[
(\beta_2I + \beta_3D^TD)F^{k+1} = \beta_2(S^{k+1} - E^{k+1} + \Lambda_2^k/\beta_2) + \beta_3D^T(K^{k+1} - \Lambda_3^k/\beta_3). \quad (14)
\]

Finally, we present our sGS-ADMM based algorithm for solving model (7) (equivalently (4)) in Algorithm 3.

**Algorithm 3: sGS-ADMM for solving (4)**

Let $\tau \in (0, (1 + \sqrt{5})/2)$ be a step-length, $\beta_1, \beta_2, \beta_3 > 0$ be given parameters.
Choose initial point $(u^0, v^0, S^0, E^0, K^0, \Lambda_1^0, \Lambda_2^0, \Lambda_3^0)$.

while not converge do
  Step 1 Compute $\{u^{k+\frac{1}{2}}, F^{k+\frac{1}{2}}\}$ according to (8) and (9);
  Step 2 Compute $\{S^{k+1}, E^{k+1}, K^{k+1}\}$ according to (10), (11) and (12);
  Step 3 Compute $\{u^{k+1}, F^{k+1}\}$ according to (13) and (14);
  Step 4 Update $\{\Lambda_1^{k+1}, \Lambda_2^{k+1}, \Lambda_3^{k+1}\};$
end while

It is not hard to determine that the computational complexity of the Algorithm 3 is $O(m^2(m + n))$. Since our method is basically an application of sGS-ADMM, its convergence is guaranteed by the results in the literature, see, e.g., [8, 20] for more details. We summarize the convergence of Algorithm 3 in the following theorem.
Theorem 3.1. For $\beta_1, \beta_2, \beta_3 > 0$ and $\tau \in (0, (1+\sqrt{5})/2)$, the sequence $\{(u^k, F^k, S^k, E^k, K^k, \Lambda_1^k, \Lambda_2^k, \Lambda_3^k)\}$ generated by Algorithm 3 converges to $(u^*, F^*, S^*, E^*, K^*; \Lambda_1^*, \Lambda_2^*, \Lambda_3^*)$, where $(u^*, F^*, S^*, E^*, K^*)$ is a solution of (4), and $(\Lambda_1^*, \Lambda_2^*, \Lambda_3^*)$ is a solution of the dual problem associated with (4).

4. Numerical experiments. In this section, we will conduct experiments on synthetic and real-world video data sets to demonstrate the superiority of our proposed model over the existing state-of-the-art approaches for foreground extraction. All the experiments are performed using MATLAB (R2017a) on a desktop computer with an Intel Core i5-3570M CPU with 3.4 GHz and 8 GB of memory.

4.1. Data sets. We test our model on two popular foreground extraction data sets, namely, the SABS\textsuperscript{1} data set and the CDnet\textsuperscript{2} data set. In particular, we choose 6 representative sequences from these data sets, including both synthetic and real-world videos.

4.1.1. Synthetic Data. The SABS data set is an artificial data set for pixelwise evaluation of background models. This data set is synthetic and provides high quality ground-truths. Therefore, it is easy to check out the ground-truth of foregrounds. The data set consists of video sequences for nine different challenges of background subtraction, and we select the noncamouflage and noisynight sequences to evaluate our proposed approach. These two videos exist dynamic periodically swaying leaves in the background component and a moving car in the foreground component. See the left two columns of Figure 2.

4.1.2. Real-world Data. The CDnet data set is considered as one of the most difficult tracking benchmarks, which consists of 31 real-world videos over 8000 frames and spanning six categories including diverse motion and change detection changes. To verify the performance in different scenarios, we choose two categories of videos to test our approach. The first category is bad weather, i.e. snowfall and skating. This category is much more difficult. On the one hand, snow reduces the contrast

\textsuperscript{1}https://www.vis.uni-stuttgart.de/forschung/visual_analytics/visuelle_analyse_videostreeme/stuttgart_artificial_background_subtraction_dataset/index.en.html

\textsuperscript{2}http://www.changedetection.net
of surveillance videos, which makes distinction between foreground and background difficult. On the other hand, the movements of snowflakes are detected as foreground, which reduces the capacity to identify small objects. See the middle two columns of Figure 2. The second category is dynamic background, i.e., fall and fountain. This category is the most difficult. One challenge comes from the significant dynamic background, such as the tree and water flow. The other challenge is that the moving object is relatively small, and may be occluded by dynamic background. See the right two columns of Figure 2.

4.2. Data settings. We compare our model with two existing popular methods: RPCA\(^3\) and TVRPCA\(^4\). For the RPCA and TVRPCA, the results are generated from the source codes released by their authors, and the parameters are all set as default. For our proposed method, we empirically set the joint sparse parameter \(\lambda_1 = 5/\sqrt{mn}\) and the total variation parameter \(\lambda_2 = 1/\sqrt{mn}\). To accelerate the efficiency of our proposed, we set \(\tau = 1.618\). For parameters \(\beta_1, \beta_2, \beta_3 > 0\), we apply self-adaptive adjustment rules such as \(\beta_{k+1}^i = \sigma \beta_k^i (i = 1, 2, 3)\) with given \(\sigma = 1.01\). In addition, the stopping criterion is set as \(\text{Tol} < 10^{-3}\), that is,

\[
\text{Tol} = \max \left\{ \frac{\| u_{k+1}^T - u_k^T \|_F}{\| u_k^T \|_F + 1}, \frac{\| F_{k+1}^F - F_k^F \|_F}{\| F_k^F \|_F + 1} \right\}.
\]

4.3. Comparison results. In this subsection, we will divide the comparison results into two parts: visual results and quantitative results. Note that in comparison results, the white represents correctly detected foreground, red missing pixels, and blue false alarm. In addition, some obvious differences are labeled by the yellow box (best seen in the zoom-in version of pdf).

4.3.1. Visual Results. Figure 3 shows the visual results of surveillance videos from SABS data set and CDnet data set. In the figure, the first column is the test samples, the second column is the extracted true foreground, the following three are extracted foregrounds by RPCA, TVRPCA and our proposed model.

It can be seen that the RPCA wrongly labels the nearby background as the moving object regions by considering their spatial relationship. The TVRPCA and our model successfully removes the swaying leaves and watersurfaces with some different degrees. This is because that the total variation can preserve sharp edges and object boundaries by removing small artifacts. It is worth noting that, compared with the TVRPCA, our model can filter out more dynamic textures and obtain more accurate foreground. The superior is due to the joint sparsity, which exploits the structural property.

4.3.2. Quantitative Results. In this subsection, we present some quantitative results. In order to evaluate the performance of foreground extraction, we compare the support of the recovered foreground with the support of the ground-truth by computing the following F-measure:

\[
F = 2 \frac{R \cdot P}{R + P},
\]

where \(R\) and \(P\) are short for recall and precision, respectively, which are defined as

\[
R = \frac{TP}{TP + FN}, \quad P = \frac{TP}{TP + FP}.
\]

---

\(^3\)http://perception.csl.illinois.edu/matrix-rank/sample_code.html

\(^4\)yangliang.github.io
Figure 3. Visual Comparisons on both Synthetic and Real-world Data Sets.

- TP stands for true positives: the number of true foreground pixels that are recovered;
- FP stands for false positives: the number of background pixels that are mis-detected as foreground;
- FN stands for false negatives: the number of true foreground pixels that are missed.

The support of the recovered foreground is obtained by thresholding entry-wise with a threshold value. We would like to point out that the F-measure varies between zero and one according to the similarity of the support of the recovered foreground and the ground-truth. The higher the F-measure value, the better the recovery accuracy. The F-measure approaches the maximum value one if the supports of of the recovered foreground and the ground-truth are the same, which means the foreground is recovered completely.

Table 1 summarizes the quantitative results on both synthetic and real-world data sets. Notice that the F-measure here indicates the averaged F-measure on all frames. From the table, one can observe that our proposed model improves the recall and precision, which makes F-measure increase in all cases. This can be attributed to the integration of the joint sparsity and total variation, where the joint sparsity
Table 1. Quantitative Results on both Synthetic and Real-world Data Sets

| Videos       | RPCA | Ours | Ours |
|--------------|------|------|------|
|              | R    | P    | F    | R    | P    | F    | R    | P    | F    |
| noncamouflage| 0.64 | 0.35 | 0.45 | 0.65 | 0.36 | 0.47 | 0.69 | 0.78 | 0.73 |
| noisynight   | 0.50 | 0.20 | 0.29 | 0.40 | 0.19 | 0.26 | 0.45 | 0.67 | 0.54 |
| snowfall     | 0.74 | 0.37 | 0.49 | 0.75 | 0.41 | 0.53 | 0.87 | 0.92 | 0.89 |
| skating      | 0.72 | 0.51 | 0.60 | 0.75 | 0.66 | 0.70 | 0.74 | 0.86 | 0.80 |
| fall         | 0.68 | 0.13 | 0.21 | 0.69 | 0.13 | 0.21 | 0.83 | 0.94 | 0.88 |
| fountain     | 0.84 | 0.02 | 0.04 | 0.74 | 0.03 | 0.06 | 0.74 | 0.08 | 0.14 |

can exploit the shared information and the total variation can preserve sharp edges and object boundaries by removing small artifacts.

5. Conclusion. In this paper, we propose a novel total variation based RPCA approach for foreground extraction, in which the total variation is utilized to model the spatio-temporal correlation of the dynamic foreground, and the joint sparsity is employed to characterize the dynamic background. Furthermore, we develop an efficient sGS-ADMM algorithm for our proposed RPCA model, and establish the convergence of the generated sequence. Finally, we perform numerical experiments on real-world datasets to demonstrate the effectiveness of our proposed approach.

Acknowledgments. We would like to thank the anonymous referees for their numerous insightful comments and suggestions, which have greatly improved the paper.

REFERENCES

[1] T. Bouwmans, Traditional and recent approaches in background modeling for foreground detection: An overview, Computer Science Review, 11/12 (2014), 31–66.
[2] T. Bouwmans, A. Sobral, S. Javed, S. Jung and E. Zahzah, Decomposition into low-rank plus additive matrices for background/foreground separation, Computer Science Review, 23 (2017), 2–71.
[3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine learning, 3 (2010), 1–122.
[4] E. Candès, X. Li, Y. Ma and J. Wright, Robust principal component analysis?, Journal of the ACM, 58 (2009), Art. 11, 37 pp.
[5] E. Candès and T. Tao, Decoding by linear programming, IEEE Transactions on Information Theory, 51 (2005), 4203–4215.
[6] X. Cao, L. Yang and X. Guo, Total variation regularized RPCA for irregularly moving object detection under dynamic background, IEEE Transactions on Cybernetics, 46 (2016), 1014–1027.
[7] C. Chen, B. He, Y. Ye and X. Yuan, The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent, Mathematical Programming, 155 (2016), 57–79.
[8] L. Chen, D. Sun and K. Toh, An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming, Mathematical Programming, 161 (2017), 237–270.
[9] Y. Chen, Z. Luo and N. Xiu, Half thresholding eigenvalue algorithm for semidefinite matrix completion, Science China Mathematics, 58 (2015), 2015–2032.
[10] Y. Chen, N. Xiu and D. Peng, Global solutions of non-Lipschitz $S_2−S_p$ minimization over the positive semidefinite cone, Optimization Letters, 8 (2014), 2051–2064.
[11] D. Donoho, De-Noising by soft thresholding, IEEE Transactions on Information Theory, 41 (1995), 613–627.
[12] A. Elgammal, R. Duraiswami, D. Harwood and L. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, Proceedings of the IEEE, 90 (2002), 1151–1163.
[13] M. Fazel, T. Pong, D. Sun and P. Tseng, Hankel matrix rank minimization with applications to system identification and realization, SIAM Journal on Matrix Analysis and Applications, 34 (2013), 946–977.
[14] D. Gabay, Applications of the method of multipliers to variational inequalities, In: M. Fortin, R. Glowinski (Eds.), Augmented Lagrange Methods: Applications to the Solution of Boundary-valued Problems. North-Holland, Amsterdam, The Netherlands, (1983), 299–331.
[15] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers and Mathematics with Applications, 2 (1976), 17–40.
[16] B. He, M. Tao and X. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, SIAM Journal on Optimization, 22 (2012), 313–340.
[17] B. He and X. Yuan, On the $O(1/n)$ convergence rate of the Douglas-Rachford alternating direction method, SIAM Journal on Numerical Analysis, 50 (2012), 700–709.
[18] M. Li, D. Sun and K. Toh, A convergent 3-block semi-proximal ADMM for convex minimization problems with one strongly convex block, Asia-Pacific Journal of Operational Research, 32 (2015), 1550024, 19pp.
[19] X. Li, M. Ng and X. Yuan, Median filtering-based methods for static background extraction from surveillance video, Numerical Linear Algebra with Applications, 22 (2015), 845–865.
[20] X. Li, D. Sun and K. Toh, A block symmetric Gauss-Seidel decomposition theorem for convex composite quadratic programming and its applications, Mathematical Programming, (2017), 1–24.
[21] J. Liu, S. Ji and J. Ye, Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, (2012), 339–348.
[22] L. Maddalena and A. Petrosino, A self-organizing approach to background subtraction for visual surveillance applications, IEEE Transactions on Image Processing, 17 (2008), 1168–1177.
[23] B. Natarajan, Sparse approximate solutions to linear systems, SIAM Journal on Computing, 24 (1995), 227–234.
[24] R. Rockafellar, Convex Analysis, Princeton University Press, 1970.
[25] L. Rudin and S. Osher, Total variation based image restoration with free local constraints, Proceedings of IEEE International Conference on Image Processing, 1 (1994), 31–35.
[26] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60 (1992), 259–268.
[27] A. Sobral and A. Vacavant, A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos, Computer Vision and Image Understanding, 122 (2014), 4–21.
[28] C. Stauffer and W. Grimson, Adaptive background mixture models for real-time tracking, IEEE Conference on Computer Vision and Pattern Recognition, 2 (1999), 2246.
[29] X. Xiu and L. Kong, Rank-min-one and sparse tensor decomposition for surveillance video, Pacific Journal of Optimization, 11 (2015), 403–418.
[30] L. Yang, T. Pong and X. Chen. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. SIAM Journal on Imaging Sciences, 10 (2017), 74–110.
[31] X. Zhang, D. S. Pham, S. Venkatesh, W. Liu and D. Phung. Mixed-norm sparse representation for multi view face recognition, *Pattern Recognition, 48* (2015), 2935–2946.

Received May 2018; 1st revision December 2018; final revision January 2019.

*E-mail address:* xcxiu@bjtu.edu.cn
*E-mail address:* yy@pku.edu.cn
*E-mail address:* W.Liu@curtin.edu.au
*E-mail address:* lchkong@bjtu.edu.cn
*E-mail address:* meijuan_shang@163.com