Adaptive Rank Estimate in Robust Principal Component Analysis

Zhengqin Xu[1,3] Rui He[2,3] Shoulie Xie[4] Shiqian Wu[2,3]
[1]School of Machinery and Automation, Wuhan University of Science and Technology
[2]School of Information Science and Engineering, Wuhan University of Science and Technology
[3]Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology
[4]Signal Processing, RF & Optical Dept. Institute for Infocomm Research A*STAR, Singapore
{xuzhengqin,herui,shiqian.wu}@wust.edu.cn slxie@i2r.a-star.edu.sg

Abstract

Robust principal component analysis (RPCA) and its variants have gained wide applications in computer vision. However, these methods either involve manual adjustment of some parameters, or require the rank of a low-rank matrix to be known a prior. In this paper, an adaptive rank estimate based RPCA (ARE-RPCA) is proposed, which adaptively assigns weights on different singular values via rank estimation. More specifically, we study the characteristics of the low-rank matrix, and develop an improved Gerschgorin disk theorem to estimate the rank of the low-rank matrix accurately. Furthermore in view of the issue occurred in the Gerschgorin disk theorem that adjustment factor need to be manually pre-defined, an adaptive setting method, which greatly facilitates the practical implementation of the rank estimation, is presented. Then, the weights of singular values in the nuclear norm are updated adaptively based on iteratively estimated rank, and the resultant low-rank matrix is close to the target. Experimental results show that the proposed ARE-RPCA outperforms the state-of-the-art methods in various complex scenarios.

1. Introduction

The real world is full of high-dimensional data such as images and videos. The processing in high-dimensional space is computationally expensive and intractable. Fortunately, most data are not unstructured and randomly distributed over the high-dimensional space, and usually have patterns and distributed over low-dimensional manifolds. Principal component analysis (PCA) effectively proves this phenomenon, where most high-dimensional data lie around a low-dimensional subspace spanned by the principal components [1]. Hence PCA can be viewed as a low-rank modeling technique, and works well when the matrix data has no missing entries with normal errors/noise. However, the PCA often produces an undesired model when the data assumptions do not hold. Among these issues, the outlier is one of the most important limitations.

Robust PCA (RPCA) was proposed in [2] to address these issues by decomposing a data matrix into a low-rank matrix and a sparse matrix containing outliers as follows:

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

where $\boldsymbol{M}$ is the measurement matrix, $\boldsymbol{L}$ and $\boldsymbol{S}$ denote the decomposed low-rank matrix and sparse matrix, respectively, $rank(\boldsymbol{L})$ denotes the rank of matrix $\boldsymbol{L}$, $\|\boldsymbol{S}\|_0$ indicates the ℓ_0-norm which is the number of non-zero elements of $\boldsymbol{S}$, λ is a parameter balancing the rankness and sparsity. Hence, RPCA is more robust than PCA as the former considers outliers by employing a sparse term.

However, the above optimization problem is intractable because the rank operator and ℓ_0-norm are nonconvex. Fortunately, it can be relaxed to the following convex problem:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{2}$$

where $||\cdot||_*$ and $||\cdot||_1$ are the nuclear norm and the ℓ_1-norm, respectively, and $\|\boldsymbol{L}\|_* = \sum_i \sigma_i(\boldsymbol{L})$, where $\sigma_i(\boldsymbol{L})$ is the i-th singular value of matrix $\boldsymbol{L}$. As the nuclear norm and ℓ_1-norm are the convex surrogates of the rank function and ℓ_0-norm, respectively, a perfect recovery can be achieved by solving the above convex optimization [3].

RPCA is an efficient way to find the sparsity and low-rankness and has been gained wide applications [4-14], such as face recognition[4, 5], audio processing [6], depth image repair [7], background subtraction [8], and recovery models in vision process etc.

The optimization problem (2) is not solved immediately since the matrices $\boldsymbol{L}$ and $\boldsymbol{S}$ are coupled. Alternating direction method of multipliers (ADMM) algorithm is often employed to solve the RPCA problem for obtaining sparse and low-rank decomposition [2]. The key ingredient of ADMM based RPCA algorithm is the nuclear norm minimization (NNM) sub-problem, which is related to low-rank matrix recovery. The solution to NNM problem is the so-called singular value soft-thresholding operator [15, 16]:

$$\hat{\boldsymbol{L}} = \arg\min_{\boldsymbol{L}} \|\boldsymbol{M} - \boldsymbol{L}\|_F + \tau\|\boldsymbol{L}\|_* = \boldsymbol{U}\Sigma_\tau[\boldsymbol{\Sigma}]\boldsymbol{V}^T, \tag{3}$$

where τ is a parameter controlling the rankness, $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the singular value decomposition (SVD) of $\boldsymbol{M}$ with $\boldsymbol{\Sigma} = \mathrm{diag}(\{\sigma_i\}_{1\le i\le \min(m,n)})$, and $(\Sigma_\tau[\boldsymbol{\Sigma}])_{ii} =$

$\mathrm{sgn}(\sigma_i) \cdot \max(|\sigma_i| - \tau, 0)$ denotes the soft-thresholding on $\boldsymbol{\Sigma}$ with parameter τ. One can observe that the nuclear norm minimization reduces the singular values on the same scale for those larger than τ. Thus this approach is insensitive to outliers [17].

To improve the adaptation of NNM, Hu et al. [18] proposed a truncated nuclear norm regularization (TNNR) method, where only some special singular values are regularized. Later, Gu et al. [19, 20] proposed a weighted nuclear norm minimization (WNNM) that replaces the nuclear norm with a weighted one defined by

$$\|\boldsymbol{L}\|_{\boldsymbol{W}} = \sum_i \omega_i \sigma_i(\boldsymbol{L}), \quad (4)$$

where ω_i are non-negative weights. The weighting strategy greatly improves the performance of NNM method, but the weights are dependent on one constant regularization parameter that is empirically chosen [19, 20]. On the other hand, assuming that the rank of $\boldsymbol{L}$ is known in some practical applications, Oh et al. [21, 22] proposed a partial sum of singular values (PSSV) minimization defined by

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \sum_{i=N+1}^{\min(m,n)} \sigma_i(\boldsymbol{L}) + \lambda\|\boldsymbol{S}\|_1 \ \ s.t.\ \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \quad (5)$$

where N is the known rank of a low-rank matrix. For instance, one can set $N = 1$ for background subtraction, and $N = 3$ for photometric stereo. However, the rank of $\boldsymbol{L}$ cannot be known in most practical applications, PSSV is unable to recover the low-rank structure correctly in these cases.

To address this problem, a new method to estimate the rank of a low-rank matrix is presented in this paper. Inspired by the source number estimation in array signal processing [23], we propose an improved estimation method via Gerschgorin disks to estimate the rank of a low-rank matrix. Furthermore, an adaptive weighting strategy based on the iteratively estimated rank is developed to improve the performance of low-rank matrix recovery. Therefore, we can not only get an accurate approximation to the rank function, but also faultlessly recover the low-rank matrix. In summary, the main contributions of this paper are as follows:

- An improved method based on Gerschgorin disks is presented to estimate the rank of a low-rank matrix.
- A novel RPCA method with weight updating based on the iteratively estimated rank is proposed to recover the low-rank structure of a data matrix and the sparse representation from corrupted data.
- The proposed algorithm is applied to various scenarios to demonstrate the superior performance over the existing methods.

The organization of the paper is as follow. Section 2 presents the modified robust PCA which consists of rank estimation and rank-estimation based adaptive weighting. Section 3 reports the experimental results, and some conclusions are drawn in Section 4.

2. Modified Robust PCA

2.1. Rank Estimation of Low-Rank Matrix

Since the rank of a low-rank matrix is a very important parameter in weighted NNM problem, we borrow the idea of Gerschgorin disk theorem [23] to identify the rank when it is unknown.

Assuming that there are N observations with M elements, and each observation is transformed into a row through the stretching process, the N samples thus form a $(N \times M)$ observation matrix $\boldsymbol{O}$. If each row of the observation matrix is regarded as a one-dimensional signal $o(t)$ with the snapshot number M, then N samples are viewed as N signal sensors. In this way, the multiple sample processing is transformed into an array signal processing.

Given a low-rank matrix $\tilde{\boldsymbol{L}}$ with rank r, it can be regarded as an array signal $\tilde{\boldsymbol{L}}(t) = [l_1(t), l_2(t), \cdots, l_N(t)]^T$, which includes r independent signals with M snapshots from N signal sensors. Thus the r independent signals can be defined as $\widetilde{\boldsymbol{F}}(t) = [f_1(t), f_2(t), \cdots, f_r(t)]^T$. The low-rank matrix $\tilde{\boldsymbol{L}}$ is decomposed by SVD as follows:

$$\begin{aligned}\tilde{\boldsymbol{L}} &= \boldsymbol{U}_{\tilde{\boldsymbol{L}}}\boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}}\boldsymbol{V}_{\tilde{\boldsymbol{L}}}^{\mathrm{T}} \\ &= \sum_{i=1}^{r} u_{\tilde{\boldsymbol{L}}i}\sigma_{\tilde{\boldsymbol{L}}i}v_{\tilde{\boldsymbol{L}}i}^{\mathrm{T}}\end{aligned} \quad (6)$$

where $\boldsymbol{U}_{\tilde{\boldsymbol{L}}} = (u_{\tilde{\boldsymbol{L}}1}, u_{\tilde{\boldsymbol{L}}2}, \cdots, u_{\tilde{\boldsymbol{L}}r})$ is the matrix consisting of left singular value vectors of $\tilde{\boldsymbol{L}}$ with $u_{\tilde{\boldsymbol{L}}i} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}} = diag(\sigma_{\tilde{\boldsymbol{L}}1}, \sigma_{\tilde{\boldsymbol{L}}2}, \cdots, \sigma_{\tilde{\boldsymbol{L}}r})$ is the singular value matrix, $\boldsymbol{V}_{\tilde{\boldsymbol{L}}} = (v_{\tilde{\boldsymbol{L}}1}, v_{\tilde{\boldsymbol{L}}2}, \cdots, v_{\tilde{\boldsymbol{L}}r})$ is the matrix consisting of right singular value vectors of $\tilde{\boldsymbol{L}}$ with $v_{\tilde{\boldsymbol{L}}i} \in \mathbb{R}^{M\times 1}$.

Let $\boldsymbol{A}_{\tilde{\boldsymbol{L}}} = \boldsymbol{U}_{\tilde{\boldsymbol{L}}}$, and $\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}} = \boldsymbol{\Sigma}_{\tilde{\boldsymbol{L}}}\boldsymbol{V}_{\tilde{\boldsymbol{L}}}^{\mathrm{T}}$, the observation matrix is $\boldsymbol{M} = \boldsymbol{A}_{\tilde{\boldsymbol{L}}}\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}}$ if the environment is noise-free. If outliers or corrupt noise $\tilde{\boldsymbol{S}}$ occur in the environment, the observation matrix can be defined as

$$\boldsymbol{M} = \boldsymbol{A}_{\tilde{\boldsymbol{L}}}\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}} + \tilde{\boldsymbol{S}}. \quad (7)$$

In view of Eq. (6), we define $\sigma_{\tilde{\boldsymbol{L}}i}v_{\tilde{\boldsymbol{L}}i}^{\mathrm{T}}$ to correspond to the ith independent signal, and $u_{\tilde{\boldsymbol{L}}i}$ to the ith signal's array manifold of N signal sensors. It is known from SVD principle that $v_{\tilde{\boldsymbol{L}}i}$ are independent each other. It can be seen from [13] that $\tilde{\boldsymbol{S}}$ is often sparse matrix and independent of the low-rank matrix, and each sample, i.e. each row in $\tilde{\boldsymbol{S}}$, is also independent each other. Thus Eq. (7) is equivalent to the array output signal in the array signal processing [24] given by

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{v}(t). \quad (8)$$

In this way, the rank estimation problem of low-rank matrix is transformed into the problem of estimating the number of sources in the array signal processing. The information corresponding to each rank in the low-rank matrix can be equivalent to the information of the signal sent by each source in the array signal processing.

The covariance matrix of the observation matrix $\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_M]$ with rank r can be defined as

$$\boldsymbol{R}_{\boldsymbol{M}} = \boldsymbol{M}\boldsymbol{M}^T. \quad (9)$$

Eigenvalue decomposition of $\boldsymbol{R_M}$ is

$$\boldsymbol{R_M} = \boldsymbol{U_{R_M}}\boldsymbol{\Sigma_{R_M}}\boldsymbol{U_{R_B}^H}, \quad (10)$$

where $\boldsymbol{U_{R_M}} = [\boldsymbol{u_1}, \boldsymbol{u_2}, \cdots, \boldsymbol{u_N}]$ is the eigenvector matrix, and $\boldsymbol{\Sigma_{R_M}} = diag(\sigma_1, \sigma_2, \ldots, \sigma_N)$ is the eigenvalue matrix. If there is no noise, the eigenvalues of $\boldsymbol{R_M}$ are

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_N = 0. \quad (11)$$

Due to the interference of the sparse matrix in the real environment, the eigenvalues of the covariance matrix $\boldsymbol{R_M}$ are

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq \sigma_{r+1} \geq \cdots \geq \sigma_N. \quad (12)$$

In order to accurately identify the rank of the low-rank matrix, the idea of the Gerschgorin's disk theorem is employed. First the covariance matrix $\boldsymbol{R_M}$ is partitioned as

$$\boldsymbol{R_M} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1N} \\ R_{21} & R_{22} & \cdots & R_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N1} & R_{N2} & \cdots & R_{NN} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R_{M1}} & \boldsymbol{R} \\ \boldsymbol{R^H} & R_{NN} \end{bmatrix}, \quad (13)$$

where matrix $\boldsymbol{R_{M1}} \in \mathbb{R}^{(N-1)\times(N-1)}$ is obtained by deleting the last column and row of $\boldsymbol{R_M}$. By defining each row of $\boldsymbol{A_{\tilde{L}}}$ in Eq. (7) as a vector, it can be rewritten as

$$\boldsymbol{A_{\tilde{L}}} = [b_1, \boldsymbol{b_2}, \cdots, b_N]^T. \quad (14)$$

It is noted that $\boldsymbol{R}$ in Eq. (13) can be expressed by

$$\begin{aligned} \boldsymbol{R} &= \left[R_{1N}, R_{2N}, \cdots, R_{(N-1)N}\right]^T \\ &= [\boldsymbol{b_1}, \boldsymbol{b_2}, \cdots, \boldsymbol{b_{N-1}}]^T \boldsymbol{R_F} b_N^* \\ &= \boldsymbol{A_1}\boldsymbol{R_F} b_N^*, \end{aligned} \quad (15)$$

where $\boldsymbol{R_F}$=$\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}}\widetilde{\boldsymbol{F}}_{\tilde{\boldsymbol{L}}}^{\ T}$ and $\boldsymbol{A_1} = [b_1, b_2, \cdots, b_{N-1}]^T$.

Next the eigenvalue decomposition of the covariance matrix $\boldsymbol{R_{M1}}$ can be given by

$$\boldsymbol{R_{M1}} = \boldsymbol{U_{M1}}\boldsymbol{\Sigma_1}\boldsymbol{U_{M1}^H}, \quad (16)$$

where $\boldsymbol{U_{M1}}$ is an $(N-1)\times(N-1)$ unitary matrix composed of the eigenvectors of $\boldsymbol{R_{M1}}$ as

$$\boldsymbol{U_{M1}} = [\boldsymbol{q_1'}, \boldsymbol{q_2'}, \cdots, \boldsymbol{q_{N-1}'}], \quad (17)$$

and $\boldsymbol{\Sigma_1} = diag\{\sigma_1', \sigma_2', \cdots, \sigma_{N-1}'\}$ is a diagonal matrix of eigenvalues of $\boldsymbol{R_{M1}}$. Similar to Eq. (12), the eigenvalues can be expressed as

$$\sigma_1' \geq \sigma_2' \geq \cdots \geq \sigma_r' \geq \sigma_{r+1}' \geq \cdots \geq \sigma_{N-1}' \quad (18)$$

Following the idea in [25] that the eigenvalues in (12) and (18) satisfy the interlacing property:

$$\begin{aligned} \sigma_1 \geq \sigma_1' \geq \sigma_2 \geq \sigma_2' \geq \cdots \geq \sigma_r \geq \sigma_r' \geq \sigma_{r+1} \geq \sigma_{r+1}' \\ \geq \cdots \geq \sigma_{N-1} \geq \sigma_{N-1}' \geq \sigma_N \end{aligned} \quad (19)$$

One $N \times N$ unitary transformed matrix $\boldsymbol{U}$ $(\boldsymbol{U}\boldsymbol{U}^H = \boldsymbol{I})$ can be defined as

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U_{M1}} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{1} \end{pmatrix}. \quad (20)$$

Thus, the transformed covariance matrix is obtained by

$$\begin{aligned} \boldsymbol{R_T} = \boldsymbol{U}^H\boldsymbol{R_M}\boldsymbol{U} &= \begin{pmatrix} \boldsymbol{U_{M1}^H}\boldsymbol{R_{M1}}\boldsymbol{U_{M1}} & \boldsymbol{U_{M1}^H}\boldsymbol{R} \\ \boldsymbol{R^H}\boldsymbol{U_{M1}} & R_{NN} \end{pmatrix} \\ &= \begin{pmatrix} \boldsymbol{\Sigma_1} & \boldsymbol{U_{M1}^H}\boldsymbol{R} \\ \boldsymbol{R^H}\boldsymbol{U_{M1}} & R_{NN} \end{pmatrix} \\ &= \begin{pmatrix} \sigma_1' & 0 & 0 & \cdots & 0 & \rho_1 \\ 0 & \sigma_2' & 0 & \cdots & 0 & \rho_2 \\ 0 & 0 & \sigma_3' & \cdots & 0 & \rho_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{N-1}' & \rho_{N-1} \\ \rho_1^* & \rho_2^* & \rho_3^* & \cdots & \rho_{N-1}^* & R_{NN} \end{pmatrix} \end{aligned} \quad (21)$$

where

$$\rho_i = \boldsymbol{q_i'}^H\boldsymbol{R} = \boldsymbol{q_i'}^H\boldsymbol{A_1}\boldsymbol{R_F}b_N^* \quad (22)$$

for $i = 1,2,\cdots,N-1$.

The eigenvalues of $\boldsymbol{R_T}$ can be estimated by Gerschgorin's disk theorem [23]. The radii of the first $(N-1)$ Gerschgorin's disks can be expressed as

$$r_i = |\rho_i| = \left|\boldsymbol{q_i'}^H\boldsymbol{A_1}\boldsymbol{R_F}b_N^*\right| = \left|\boldsymbol{q_i'}^H\boldsymbol{R}\right| \quad (23)$$

for $i = 1,2,\cdots,N-1$.

By Cauchy-Schwartz inequality, we can obtain that

$$\begin{aligned} r_i = |\rho_i| &= \left|\boldsymbol{q_i'}^H\boldsymbol{A_1}\boldsymbol{R_F}b_N^*\right| \\ &\leq \left|\boldsymbol{q_i'}^H\boldsymbol{A_1}\right| \cdot |\boldsymbol{R_F}b_N^*| = \gamma\left|\boldsymbol{q_i'}^H\boldsymbol{A_1}\right|, \end{aligned} \quad (24)$$

where $\gamma = |\boldsymbol{R_F}b_N^*|$ is independent of i. Then the radius r_i of the ith Gerschgorin's disk actually depends on the size of $\boldsymbol{q_i'}^H\boldsymbol{A_1}$.

If $\boldsymbol{q_i'}$ is the eigenvector of noise, the radius of the ith Gerschgorin's disk will be significantly small and close to zero. If $\boldsymbol{q_i'}$ is the eigenvector of the low-rank part, the radius of the ith Gerschgorin's disk will be far from zero. In this work, the rank is identified by the heuristic decision rule as

$$\boldsymbol{GDE}(k) = r_k - \frac{D(M)}{N-1}\sum_{i=1}^{N-1} r_i \quad (25)$$

where $k = 1,2,\cdots,N-2$, and the adjustment factor $D(M)$ (between 0 to 1) is a constant related to M. The rank of the low-rank matrix is $r = k-1$ when $\boldsymbol{GDE}(k)$ is negative for the first time. This implies that the rank can be estimated by comparing the kth Gerschgorin's disk radius r_k with a threshold, which is equal to the product of the adjustment factor $D(M)$ and the arithmetic mean of all Gerschgorin's disk radius.

To further improve the accuracy of the rank estimation, we propose a new method to shrink the radius of the Gerschgorin's disk. The idea is to compress the radii of low-rank Gerschgorin's disks and sparse Gerschgorin's disks to different degrees, which benefits to discriminate between low-rank Gerschgorin's disks and sparse ones. In light of Eq. (18), σ_i' of the sparse Gerschgorin's disk is significantly smaller than that of the low-rank Gerschgorin's disk. Thus, the diagonal matrix $\boldsymbol{D}$ can be constructed as follows:

$$\boldsymbol{D} = diag(\sigma_1', \sigma_2', \cdots, \sigma_{N-1}', \sigma_N') \quad (26)$$

where $\sigma_N' = \sqrt{\sum_{i=1}^{N-1}\sigma_i'^2}$. The new transformed matrix $\boldsymbol{R_{TD}}$ can be obtained by

$$\boldsymbol{R_{TD}} = \boldsymbol{D}\boldsymbol{R_T}\boldsymbol{D}^{-1}$$

$$= \begin{pmatrix} \sigma_1' & 0 & \cdots & 0 & \frac{\sigma_1'}{\sigma_N'}\rho_1 \\ 0 & \sigma_2' & \cdots & 0 & \frac{\sigma_2'}{\sigma_N'}\rho_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{N-1}' & \frac{\sigma_{N-1}'}{\sigma_N'}\rho_{N-1} \\ \frac{\sigma_1'}{\sigma_N'}\rho_1^* & \frac{\sigma_2'}{\sigma_N'}\rho_2^* & \cdots & \frac{\sigma_{N-1}'}{\sigma_N'}\rho_{N-1}^* & R_{NN} \end{pmatrix} \quad (27)$$

It is noted that $\boldsymbol{R_{TD}}$ and $\boldsymbol{R_T}$ are similar matrices and their eigenvalues are the same. Given Eq. (27), the centers of the Gerschgorin's disks are not changed, but the radii are compressed to various degrees. The radii of the sparse Gerschgorin's disks are compressed more than those of the low-rank Gerschgorin's disks. Then, we can estimate the rank by the improved heuristic decision rule as

$$\boldsymbol{RANK}(k) = \frac{1}{\sqrt{\sum_{i=1}^{N-1}\sigma_i'^2}}\left[\left|\sigma_k'\right|r_k - \frac{D^{(k)}(M)}{N-1}\sum_{i=1}^{N-1}\left|\sigma_i'\right|r_i\right] \quad (28)$$

where $k = 1,2,\cdots,N-2$, and the adjustment factor $0 < D^{(k)}(M) < 1$. The rank $r = k-1$ if the first negative value of (28) is reached at k.

It is highlighted that the parameter $D(M)$ in Eq. (25) is manually set according to M. A very high or a very low adjustment factor yields inaccurate rank estimation. To this end, we define a new adjustment factor $D^{(k)}(M)$ by using the center of the Gerschgorin's disk, which is given by

$$D^{(k)}(M) = \frac{2\left|\sigma_{k+1}'\right|}{\sqrt{\sum_{i=k}^{N-1}\sigma_i'^2}}. \quad (29)$$

The value of $D^{(k)}(M)$ is completely determined by the matrix $\boldsymbol{R_{TD}}$ itself without heuristic selection. By using Eqs. (28) and (29), an automatic and improved method is accordingly developed for rank identification.

2.2. Adaptive RPCA based on Iterative Rank Estimate

In this subsection, a new adaptive RPCA, which updates the weights of singular values via iterative rank estimate, is proposed to recover the low-rank matrix from the corrupted measurements. Specifically, the low-rank recovery is achieved by solving the following optimization formulation:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \ ||\boldsymbol{L}||_{\boldsymbol{W}} + \lambda\|\boldsymbol{S}\|_1 \quad s.t. \ \ \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \quad (30)$$

where $\|\boldsymbol{L}\|_{\boldsymbol{W}} = \sum_i \omega_i\sigma_i(\boldsymbol{L})$ and ω_i are non-negative weights. Given the rank r estimated in the previous section, the proposed idea, different from the existing solutions [20], is to preserve the singular values within the target rank, i.e. $\sigma_{1\le i\le r}$ while minimizing the singular values outside the target rank, i.e. $\sigma_{r+1\le i\le N}$, such that the matrix $\boldsymbol{L}$ obtained by Eq. (30) is better close to the target low-rank matrix. Hence, we define the weight as

$$\omega_i = \begin{cases} 0, & i \le r \\ 1, & otherwise \end{cases} \quad (31)$$

where r is the rank of the low-rank matrix, which is estimated by Eq. (28). Thus only residual singular values are minimized, such that the recovered low-rank matrix has rank close to the estimated rank *r*.

In general, the solution to Problem (30) has to been performed via iterative technique by fixing rank r. Different from such processing, the proposed menthod updates the rank r according to Eq. (28) in the iterative procedure. This reveals that the weights shown in Eq. (31) updates in each iteration accordingly. Hence, the optimization via Eqs (28)-(31) is called adaptive rank estimate based RPCA (ARE-RPCA). In other word, Eq.(28) provides an initial estimation of rank r, but the rank r is updated iteratively in solving Eq.(30).

In this work, the alternating direction method of multipliers (ADMM) is employed to solve Problem (30). The augmented Lagrangian function of Eq. (30) can be written as

$$\mathcal{L}(\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}) = \|\boldsymbol{L}\|_{\boldsymbol{W}} + \lambda\|\boldsymbol{S}\|_1 + \langle\boldsymbol{Y},\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle + \tfrac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2 \quad (32)$$

where $\langle\cdot,\cdot\rangle$ represents matrix inner product, μ is a positive penalty scalar, and $\boldsymbol{Y}$ is the Lagrangian multiplier. As it is difficult to solve the minimization of Eq. (32), an alternativesolution is to optimize one variable while fixing the others. Accordingly, the optimization is divided into the following three sub-problems.

$\boldsymbol{S}$ *sub-problem*: While both $\boldsymbol{L}$ and $\boldsymbol{Y}$ are fixed, Eq. (32) is equal to the following optimization problem:

$$\begin{aligned}\boldsymbol{S}^* &= \arg\min_{\boldsymbol{S}} \ \lambda\|\boldsymbol{S}\|_1 + \langle\boldsymbol{Y},\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle + \tfrac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2 \\ &= \arg\min_{\boldsymbol{S}} \ \tfrac{\lambda}{\mu}\|\boldsymbol{S}\|_1 + \tfrac{1}{2}\|\boldsymbol{S}-(\boldsymbol{M}-\boldsymbol{L}+\mu^{-1}\boldsymbol{Y})\|_F^2 \quad (33)\end{aligned}$$

$\boldsymbol{L}$ *sub-problem*: Given $\boldsymbol{S}$ and $\boldsymbol{Y}$, Eq. (32) leads to the following optimization problem:

$$\begin{aligned}\boldsymbol{L}^* &= \arg\min_{\boldsymbol{L}} \ \|\boldsymbol{L}\|_{\boldsymbol{W}} + \langle\boldsymbol{Y},\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle + \tfrac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2 \\ &= \arg\min_{\boldsymbol{L}} \ \tfrac{1}{\mu}\|\boldsymbol{L}\|_{\boldsymbol{W}} + \tfrac{1}{2}\|\boldsymbol{L}-(\boldsymbol{M}-\boldsymbol{S}+\mu^{-1}\boldsymbol{Y})\|_F^2 \quad (34)\end{aligned}$$

$\boldsymbol{Y}$ *sub-problem*: $\boldsymbol{Y}$ is updated by

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{M}-\boldsymbol{L}_{k+1}-\boldsymbol{S}_{k+1}). \quad (35)$$

In order to solve the three sub-problems, a soft-thresholding operator is introduced:

$$S_\varepsilon[x] \doteq \begin{cases} x-\varepsilon, & if\ x > \varepsilon \\ x+\varepsilon, & if\ x < -\varepsilon \\ 0, & otherwise \end{cases} \quad (36)$$

where $x \in \mathbb{R}$ and $\varepsilon > 0$. $\boldsymbol{S}^*$ in Eq. (33) can be obtained by the well-known analysis [26]:

$$\boldsymbol{S}^* = S_{\frac{\lambda}{\mu}}[\boldsymbol{M}-\boldsymbol{L}+\mu^{-1}\boldsymbol{Y}] \quad (37)$$

with the operation being element-wise. In order to solve the optimization (34), we first give the following lemma and theorems

Lemma 1 If $\boldsymbol{C},\boldsymbol{D} \in \mathbb{R}^{m\times n}$ satisfy $\boldsymbol{C}^T\boldsymbol{D} = 0$, we have

$$\|\boldsymbol{C}+\boldsymbol{D}\|_{\boldsymbol{W}} \ge \|\boldsymbol{C}\|_{\boldsymbol{W}} \quad (38)$$

$$\|\boldsymbol{C}+\boldsymbol{D}\|_F \ge \|\boldsymbol{C}\|_F \quad (39)$$

where $\|\cdot\|_{\boldsymbol{W}}$ is defined in Eq. (4). Detailed proof of Lemma 1 is demonstrated in Supplementary Materials.

Theorem 1 Given $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ where $\boldsymbol{Q} = \boldsymbol{U_Q D_Q V_Q^T}$, the solution to the minimization problem

$$\arg\min_{\boldsymbol{P}} \tfrac{1}{2}\|\boldsymbol{P}-\boldsymbol{Q}\|_F^2 + \tau\|\boldsymbol{P}\|_{\boldsymbol{W}} \quad (40)$$

is $\widehat{\boldsymbol{P}} = \boldsymbol{U_Q}\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}}\boldsymbol{V_Q^T}$, where $\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}}$ is the solution of the following optimization problem:

$$\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}} = \arg\min_{\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}}} \tfrac{1}{2}\left\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}} - \boldsymbol{D_Q}\right\|_F^2 + \tau\left\|\widehat{\boldsymbol{\Sigma}}_{\boldsymbol{P}}\right\|_{\boldsymbol{W}}. \quad (41)$$

Based on Theorem 1, we obtain the following important result.

Theorem 2 Given $\tau > 0$, $\boldsymbol{P}$, $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ where $\boldsymbol{Q} = \boldsymbol{U_Q D_Q V_Q^T}$, $\boldsymbol{D_Q} = diag\left(\delta_{\boldsymbol{Q}1}, \cdots, \delta_{\boldsymbol{Q}r}, \delta_{\boldsymbol{Q}(r+1)}, \cdots, \delta_{\boldsymbol{Q}\ell}\right)$ and $\ell = min(m,n)$. We can define $\boldsymbol{Q} = \boldsymbol{Q}_1 + \boldsymbol{Q}_2$, $\boldsymbol{Q}_1 = \boldsymbol{U_{Q1} D_{Q1} V_{Q1}^T}$ and $\boldsymbol{Q}_2 = \boldsymbol{U_{Q2} D_{Q2} V_{Q2}^T}$, where $\boldsymbol{D_{Q1}} = diag\left(\delta_{\boldsymbol{Q}1}, \cdots, \delta_{\boldsymbol{Q}r}, 0, \cdots, 0\right)$, $\boldsymbol{U_{Q1}}$ and $\boldsymbol{V_{Q1}}$ are the singular vector matrices corresponding to the rth largest singular values, $\boldsymbol{D_{Q2}} = diag\left(0, \cdots, 0, \delta_{\boldsymbol{Q}(r+1)}, \cdots, \delta_{\boldsymbol{Q}\ell}\right)$, $\boldsymbol{U_{Q2}}$ and $\boldsymbol{V_{Q2}}$ corresponding to the singular values from $(r+1)th$ to the last. $\|\cdot\|_{\boldsymbol{W}}$ is defined as shown in Eq. (30) and Eq. (31). The optimal solution to the minimization problem

$$\arg\min_{\boldsymbol{P}} \tfrac{1}{2}\|\boldsymbol{P}-\boldsymbol{Q}\|_F^2 + \tau\|\boldsymbol{P}\|_{\boldsymbol{W}} \quad (42)$$

can be expressed as

$$\begin{aligned}\boldsymbol{P}^* = \mathcal{D}_{\tau,\boldsymbol{W}}[\boldsymbol{Q}] &= \boldsymbol{U_Q}\left(\boldsymbol{D_{Q1}} + S_\tau\left[\boldsymbol{D_{Q2}}\right]\right)\boldsymbol{V_Q^T} \\ &= \boldsymbol{Q}_1 + \boldsymbol{U_{Q2}} S_\tau\left[\boldsymbol{D_{Q2}}\right]\boldsymbol{V_{Q2}^T}. \quad (43)\end{aligned}$$

Refer to Supplementary Materials for detailed proofs of Theorem 1 and Theorem 2 due to space limitation.

In light of Theorem 2, $\boldsymbol{L}^*$ in Eq. (34) can be obtained by

$$\boldsymbol{L}^* = \mathcal{D}_{\frac{1}{\mu},\boldsymbol{W}}[\boldsymbol{M} - \boldsymbol{S} + \mu^{-1}\boldsymbol{Y}]. \quad (44)$$

The entire procedure to solve problem (30) is summarized in Algorithm 1.

Algorithm 1 Adaptive rank estimate based RPCA (ARE-RPCA):

Input: $\boldsymbol{M} \in \mathbb{R}^{m\times n}, \lambda = 1/\sqrt{\max(m,n)}$;

1: **Initialization:** $\boldsymbol{S}_0 = \boldsymbol{Y}_0 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$, r is estimated by Eq. (28), $\mu = 1/\sigma_r$, and ω_i is defined by Eq. (31);
2: **while** not converged **do**
3: compute $\boldsymbol{L}_{k+1} = \mathcal{D}_{\frac{1}{\mu},\boldsymbol{W}}[\boldsymbol{M} - \boldsymbol{S}_k + \mu^{-1}\boldsymbol{Y}_k]$;
4: compute $\boldsymbol{S}_{k+1} = \Sigma_{\frac{\lambda}{\mu}}[\boldsymbol{M} - \boldsymbol{L}_{k+1} + \mu^{-1}\boldsymbol{Y}_k]$;
5: compute $\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1})$;
6: update r and ω_i according to Eq. (28) and Eq. (31), respectively;
7: **end while**
8: **output:** $\boldsymbol{L}, \boldsymbol{S}$.

Remark 1: In Algorithm 1, the parameter λ is set as $\lambda = 1/\sqrt{\max(m,n)}$, which is recommended in RPCA. The iteration is terminated when $||\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}||_F \leq 10^{-7}||\boldsymbol{M}||_F$.

Remark 2: It is noted that μ^{-1} occurs in singular value

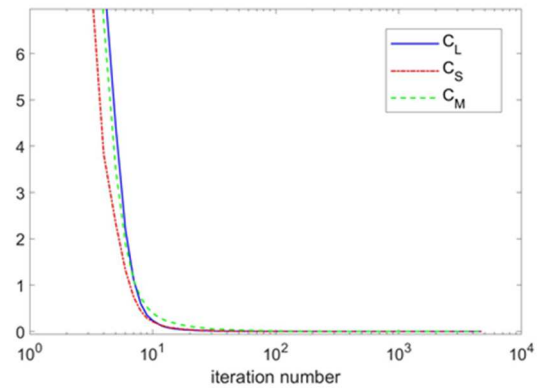

Figure 1 The value of C_L, C_S and C_M with the number of iterations. Assuming that $\boldsymbol{M} \in \boldsymbol{R}^{m\times n}$, $m = 10000$, $n = 20$, $rank(\boldsymbol{L}) = 3$, and the corrupted rate $p = 0.05$.

Table 1 Number of iterations, CPU time, $\boldsymbol{L}$ and $\boldsymbol{S}$ reconstruction error for LSD, LRSD, RPCA, SRPCP, WNNM, ARE-RPCA

	Number of iterations	CPU time	$\boldsymbol{L}$ reconstruction error	$\boldsymbol{S}$ reconstruction error
LSD	31	226s	0.1565	0.5772
LRSD	2813	15604s	0.0335	0.5143
RPCA	228	4.49s	0.0684	0.5038
SRPCP	18	355.78s	0.2148	0.6038
WNNM	171	3.2s	0.0062	0.5177
ARE-RPCA	145	2.55s	0.0051	0.5172

Table 2: The rank of low-rank matrix $\boldsymbol{L}$ decomposed by LSD, LRSD, RPCA, SRPCP, WNNM and ARE-RPCA for different value of corrupt rate p

	0.05	0.10	0.15	0.20	0.25
LSD	5	13	20	11	11
LRSD	5	5	20	20	20
RPCA	5	5	9	8	7
SRPCP	5	8	0	0	0
WNNM	5	5	5	5	5
ARE-RPCA	5	5	5	5	5

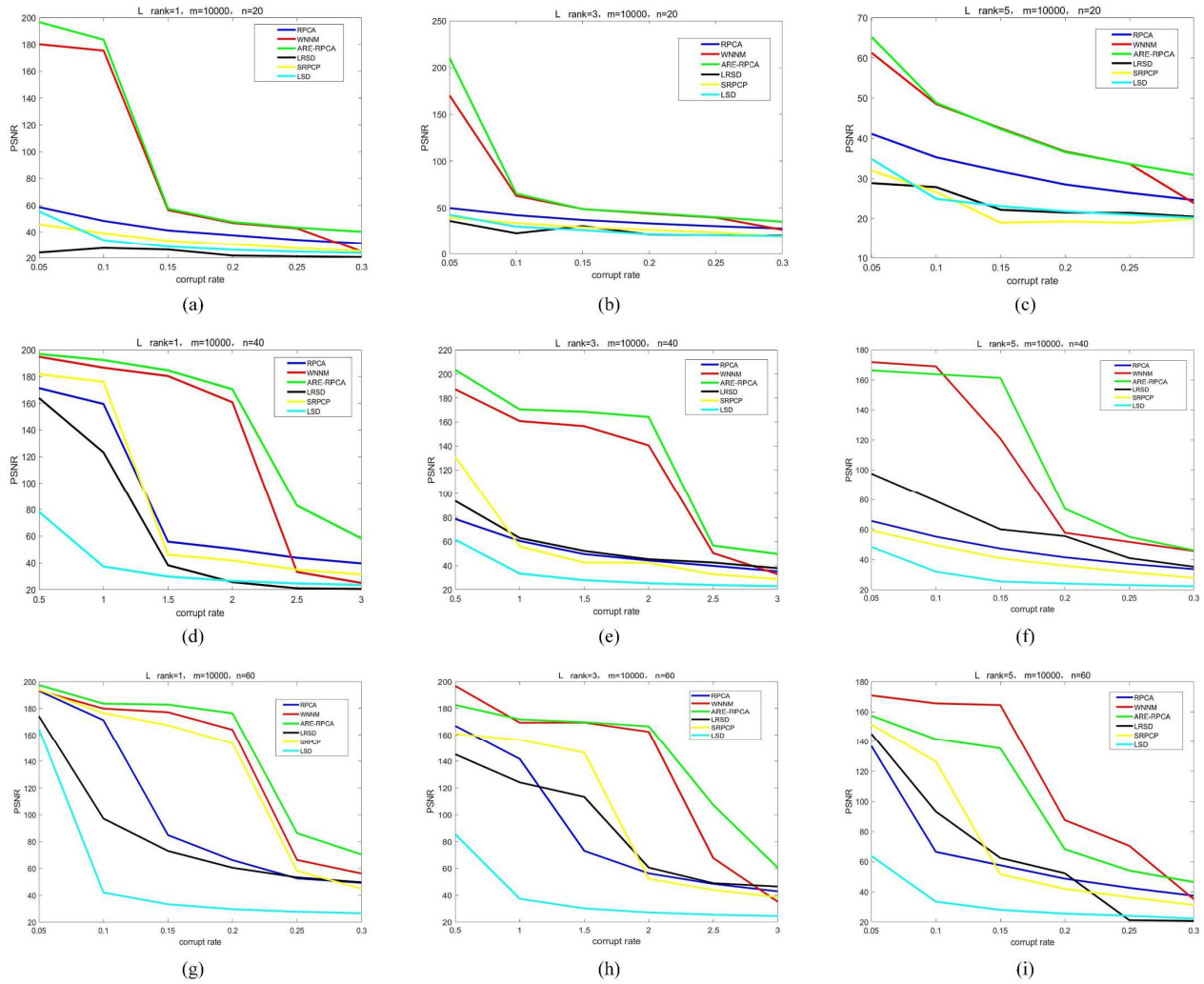

Figure 2 PSNR for various algorithms with different sample sizes n, matrix ranks r_0, and corrupt rate p.

thresholding operators. When μ^{-1} is small, a large proportion of singular values of $\boldsymbol{M}-\boldsymbol{S}_k+\mu^{-1}\boldsymbol{Y}_k$ would exceed the threshold and make the rank of $\boldsymbol{L}^*$ be too large. In classical RPCA, it is simply chosen as $\mu=\mathrm{m}\times\mathrm{n}/4\|\boldsymbol{M}\|_1$, which is not related with singular values. In this work, we select the size of μ by $\mu=1/\sigma_r$, where σ_r is the rth singular value of $\boldsymbol{M}$.

Remark 3: It should be noted that the proposed alternative Algorithm 1 follows the framework of inexact augmented Lagrangian multiplier (IALM) [27]. However, the weights in Eq. (30) are given by Eq. (31) and the underlying problem is usually nonconvex. Although mathematical proof of the convergence is challenging, the following empirical claim is provided.

Claim 1: The sequences $\{\boldsymbol{L}_k\}$ and $\{\boldsymbol{S}_k\}$ generated by Algorithm 1 satisfy:

$$C_{\boldsymbol{L}}=\lim_{k\to\infty}\|\boldsymbol{L}_{k+1}-\boldsymbol{L}_k\|_F=0$$
$$C_{\boldsymbol{S}}=\lim_{k\to\infty}\|\boldsymbol{S}_{k+1}-\boldsymbol{S}_k\|_F=0$$
$$C_{\boldsymbol{M}}=\lim_{k\to\infty}\|\boldsymbol{M}-\boldsymbol{L}_{k+1}-\boldsymbol{S}_{k+1}\|_F=0$$

Claim 1 has been proved by the experiment shown in Fig. 1.

3. Experimental Results

In this section, we report the experimental results of our adaptive rank estimate based RPCA (ARE-RPCA), and compare it with the state-of-the-art RPCA algorithms (RPCA [2], WNNM [20], SRPCP [28], LSD [29] and LRSD [30]). All the experiments are conducted on a laptop equipped with Windows 10, AMD Ryzen 7 4800H (8 Cores at 2.9 GHz) and 16GB DDR4-3200Mhz RAM, and running in MATLAB R2018b.

3.1. Synthetic Datasets

In this subsection, we test the algorithms on synthetic data. A matrix $\boldsymbol{L}\in\mathbb{R}^{m\times n}$ with rank r_0 is generated by sampling two matrices, $\boldsymbol{P}_s\in\mathbb{R}^{m\times r_0}$ and $\boldsymbol{Q}_s\in\mathbb{R}^{r_0\times n}$ with

entries belonging to normal distribution $\mathcal{N}(0,1)$, namely, $\boldsymbol{L} = \boldsymbol{P}_s \boldsymbol{Q}_s$. In this experiment, we set $m = 10000$. The ground truth matrix $\boldsymbol{L}$ is corrupted by sparse noise $\boldsymbol{S} \in \mathbb{R}^{m \times n}$, which has $p \times (m \times n)$ non-zero elements. The positions of non-zero elements in $\boldsymbol{S}$ are randomly selected, and its value is generated from a Gaussian distribution $\mathcal{N}(0,1)$. Therefore, we generate synthetic data $\boldsymbol{M}$ as follows:

$$\boldsymbol{M}_{i,j} = \boldsymbol{L}_{i,j} + \boldsymbol{S}_{i,j}.$$

We first evaluate the peak signal to noise ratio (PSNR) under different settings such as different sample sizes $n \in \{20, 40, 60\}$, different matrix ranks $r_0 \in \{1, 3, 5\}$, and different corrupt rate of p from 0.05 to 0.3. The simulation results are shown in Fig. 2. It is observed from Fig. 2 (a - c) that when the sample sizes n is small, both our proposed ARE-RPCA and WNNM outperform others, but ARE-RPCA is better than WNNM in the case of a lower corrupt rate ($p \leq 0.05$).

By comparing Fig. 2 (a, d, g), Fig. 2 (b, e, h) and Fig. 2 (c, f, i), it can be seen that the performance of all algorithms is gradually enhanced with the increase of n under the same rank. However, the performance of the proposed ARE-RPCA algorithm is better than others in most cases except for the large number of samples and large rank where ARE-RPCA is slightly weaker than WNNM. However, WNNM algorithm has the disadvantage that it needs to adjust one regularization parameter C in weight updating $\omega_i = C/(\sigma_i(\boldsymbol{L}) + \epsilon)$ (see the details in [20], this parameter is manually adjusted to the optimal values according to different actual environments in this paper). ARE-RPCA algorithm has strong adaptability because it does not need to be adjusted for different application scenarios.

Next, we evaluate the number of iterations, running time, reconstruction error of low-rank matrix $\boldsymbol{L}$ and sparse matrix $\boldsymbol{S}$. Denote the solution as $\boldsymbol{L}_{sol}$ and $\boldsymbol{S}_{sol}$ in a certain algorithm and define the reconstruction error as $\boldsymbol{L}_{re} = ||\boldsymbol{L}_{sol} - \boldsymbol{L}||_F / ||\boldsymbol{L}||_F$ and $\boldsymbol{S}_{re} = ||\boldsymbol{S}_{sol} - \boldsymbol{S}||_F / ||\boldsymbol{S}||_F$. The test results are presented in Table 1. It can be seen that Although our ARE-RPCA is the fastest among all algorithms although it has more iterations than SRPCP and LRSD. It also can be seen that there is almost no difference in the reconstruction error of sparse matrix $\boldsymbol{S}$, but the reconstruction error of low-rank matrix $\boldsymbol{L}$ of our ARE-RPCA is obviously less than others.

Finally, we consider the rank of the low-rank matrix $\boldsymbol{L}$ decomposed by different algorithms under different corruption rates. Setting $n = 20$ and $r_0 = 5$, the test results are shown in Table 2. It can be observed that our ARE-RPCA and WNNM can obtain correct rank of low-rank matrix in all cases even with large corruption rates. Other algorithms can estimate the rank correctly only if the corruption rate is low. Overall, our proposed ARE-RPCA is fast and highly accurate.

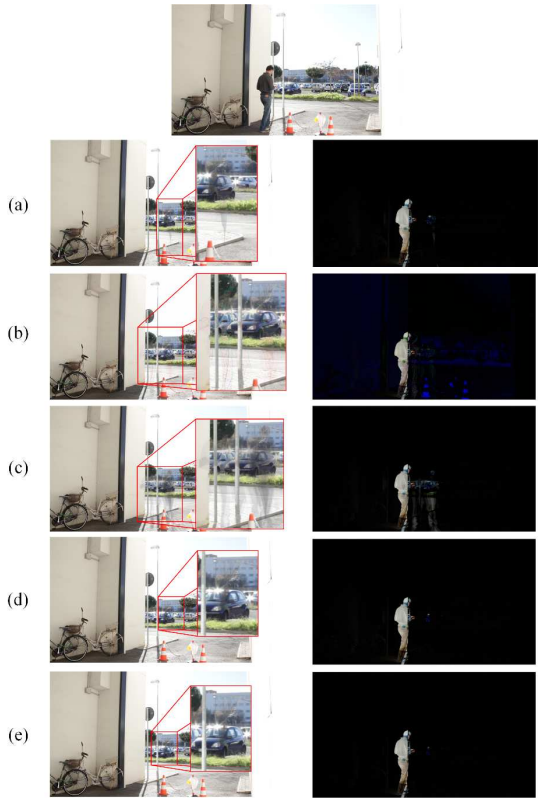

Figure 3 Video background subtraction: the top row corresponds to one frame from the video. The second to last rows are the separated background and foreground of FFP, MoG-RPCA, RPCA, WNNM and ARE-RPCA, respectively.

3.2. Real Datasets

In this subsection, we compare the performance of ARE-RPCA, WNNM, RPCA, FFP [31], MoG-RPCA [32] on two real world benchmark problems: video background subtraction and low dynamic range imaging.

The task of background subtraction is to separate the moving foreground object from the static background. We choose a Street dataset [1] that has a relatively static background and a walking person as a dynamic foreground. The size of each frame of the Street dataset is 1920×1080. The total number of frames are 48 in the Street dataset. The dataset can be represented by a matrix, where each column of the matrix is a vectorized frame of the video. Then we apply each algorithm to decompose the matrix into low-rank parts representing the static background of the video and sparse parts representing the moving objects in the video. The results are shown in Fig. 3. RPCA cannot well separate the foreground from the background as shown in Fig. 3 (c). It can be seen from Fig. 3 (a) that FFP has better

[1] The Street dataset is provided in the supplementary materials.

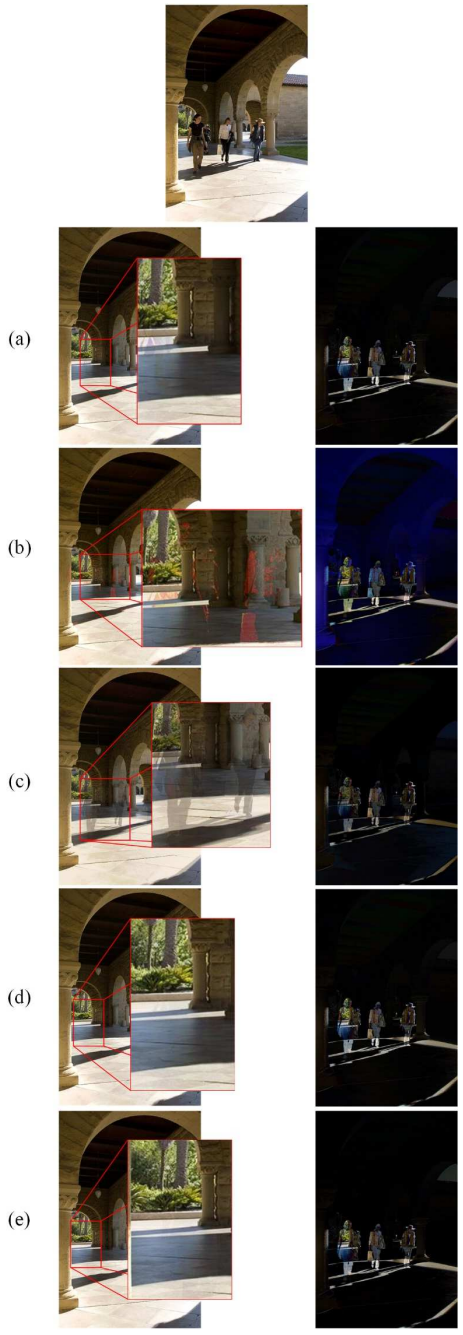

Figure 4 Low-Dynamic Range Imaging: the top row corresponds to one frame of a sequence with differently exposed changes. The second to last rows are the separated static part and dynamic part of FFP, MoG-RPCA, RPCA, WNNM and ARE-RPCA, respectively.

performance than RPCA, but it still cannot completely separate the foreground from the background. From Fig. 3 (b), it can be seen that MoG-RPCA has better performance on the blue and green components, but it has poorer performance on the red component. As shown in Fig. 3(d, e), one can see that both WNNM and ARE-RPCA can effectively separate the foreground and background, and our proposed ARE-RPCA is better than WNNM.

In order to obtain high-contrast scene images, low dynamic range (LDR) imaging technology needs to be used to remove out-of-focus blur and dynamic objects in pictures captured by low dynamic range cameras. We select the Arch dataset[1] [33] and stack each image as a column into a matrix. The size of each frame of the Arch dataset is 669×1024. The total number of frames are 5 in the Arch dataset. Then we can use each algorithm to decompose the matrix into a low-rank part representing the scene and sparse part representing dynamic objects. The experimental results are shown in Fig. 4. It can be seen from Fig. 4 (c) that RPCA cannot remove moving objects well. From Fig. 4. (a), FFP has a better effect, but still has a larger ghost image. It can be seen from Fig. 4(b) that MoG-RPCA cannot effectively remove the moving objects in the red component. As shown in Fig. 4 (d, e), one can observe that both WNNM and ARE-RPCA can effectively remove moving objects, and our proposed ARE-RPCA handles ghosting better.

4. Conclusions

Robust principal component analysis (RPCA), due to its powerful capability in dealing with outliers, has been gained wide applications in computer vision. To cope with the issues that some RPCA variants need predefine the rank of low-rank matrix and manually adjust some parameters, an adaptive rank estimate based RPCA (ARE-RPCA) is proposed in this paper. Specifically, the rank of a low-rank matrix is identified via Gerschgorin disk method. To avoid setting adjustment factor in Gerschgorin disk method, an improved rank estimation algorithm is proposed. On the other hand, a novel RPCA method with weight updating based on the iteratively estimated rank is proposed to recover the low-rank structure of a data matrix and the sparse representation from corrupted data, which makes our improved algorithm accurate and effective. Experimental results on synthetic data demonstrate that the identified rank is close to the ground truth, and the results on real data indicate that the proposed ARE-RPCA outperforms the state-of-the-art methods in terms of efficiency and accuracy. The proposed method will greatly facilitate RPCA in real applications.

5. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grants 61775172, 61371190, and Hubei Key Technical Innovation Project under Grant ZDCX2019000025.

[1] http://alumni.soe.ucsc.edu/~orazio/deghost.html

References

[1] I. Jolliffe. Principal Component Analysis. *Springer*, 2002.

[2] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John A Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1-37, 2011.

[3] Chengjin Li. A new approximation of the matrix rank function and its application to matrix rank minimization. *Journal of Optimization Theory & Applications*, 163(2):569-594, 2014.

[4] Niannan Xue, Jiankang Deng, Shiyang Cheng, Yannis Panagakis, and Stefanos Zafeiriou. Side information for face completion: a robust PCA approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(1):2349-2364, 2019.

[5] Lei Wang, Bangjun Wang, Zhao Zhang, Qiaolin Ye, Liyong Fu, Guangcan Liu, and Meng Wang. Robust auto-weighted projective low-rank and sparse recovery for visual representation. *Neural Networks*, 117(1):201-215, 2019.

[6] Sait Melih Doğan, and Özgül Salor. Music/Singing voice separation based on repeating pattern extraction technique and robust principal component analysis. *In 2018 5th International Conference on Electrical and Electronic Engineering (ICEEE)*, pages 482-487, 2018.

[7] Hongyang Xue, Shengming Zhang, and Deng Cai. Depth image inpainting: Improving low rank matrix completion with low gradient regularization. *IEEE Trans. on Image Processing*, 26(9):4311-4320, 2017.

[8] Zhuorui Yang, Marco F. Duarte, and Aura Ganz. A novel crowd-resilient visual localization algorithm via robust PCA background extraction. *in 2018 International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

[9] Paris V Giampouras, Athanasios A. Rontogiannis, and Konstantinos D. Koutroumbas. Robust PCA via alternating iteratively reweighted low-rank matrix factorization. *in 2018 International Conference on Image Processing (ICIP)*, 2018.

[10] Linhao Li, Qinghua Hu, and Xin Li. Moving object detection in video via hierarchical modeling and alternating optimization. *IEEE Trans. on Image Processing*, 28(1):2021-2036, 2018.

[11] Thierry Bouwmans, Sajid Javed, Hongyang Zhang, Zhouchen Lin, and Ricardo Otazo. On the applications of robust PCA in image and video Processing. *Proceedings of the IEEE*, 106(1):1427-1457, 2018.

[12] Namrata Vaswani, Thierry Bouwmans, Sajid Javed, and Praneeth Narayanamurthy. Robust subspace learning: Robust PCA, robust subspace tracking and robust subspace recovery. *IEEE Signal Processing Magazine*, 35(4):32-55, 2018.

[13] Thierry Bouwmans, Andrews Sobral, Sajid Javed, Soon Ki Jung, and El Hadi Zahzah. Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset. *Computer Science Review*, 23(1):1-71, 2017.

[14] Thierry Bouwmans and El Hadi Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122(1):22–34, 2014.

[15] Kim Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pacific Journal of Optimization*, 6(3):615-640, 2010.

[16] Jian Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completeon. *SIAM Journal on Optimization*, 20(4):1956-1982, 2010.

[17] Paul Rodriguez and Brendt Wohlberg. Performance comparison of iterative reweighting methods for total variation regularization. *in 2014 International Conference on Image Processing (ICIP)* 2014.

[18] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 35(9):2117–2130, 2013.

[19] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. *in 2014 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862-2869, 2014.

[20] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization and its applications to low level vision. *International Journal of Computer Vision*, 121(2):183-208, 2017.

[21] Tae-Hyun Oh, Yu-Wing Tai Tai, Jean-Charles Bazin, and Hyeongwoo Kim. Partial sum minimization of singular values in robust PCA: algorithm and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 38(4):744-758, 2016.

[22] Tae-Hyun Oh, Hyeongwoo Kim, Yu-Wing Tai, Jean-Charles Bazin, and In So Kweon. Partial sum minimization of singular values in RPCA for low-level vision. *in 2013 International Conference on Computer Vision (ICCV)*, pages 145-152, 2013.

[23] Yuanming Guo, Wei Li, Junyuan Shen, Jinjun Zhang, and Chengguang Shao. Source number estimation with Gerschgorin radii by reduced-rank covariance matrix. *in 2014 International Conference on Information & Communications Technologies (ICT)*, 2014.

[24] Payal Gupta and Monika Agrawal. Design and analysis of the sparse array for DOA estimation of noncircular signals. *IEEE Trans. on Signal Processing*, 67(2):460-473, 2019.

[25] Zhouchen Lin, Risheng. Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low rank representation. *in Neural Information Processing Systems (NIPS)*, pages 612-620, 2011.

[26] Elaine T. Hale, Wotao Yin, and Yin Zhang. Fixed-point continuation for ℓ_1 -minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[27] Fanhua Shang, Yuanyuan Liu, James Cheng, and Hong Cheng. Recovering low-rank and sparse matrices via robust bilateral factorization. *in 2014 International Conference on Data Mining (ICDM)*, pages 965-970, 2014.

[28] Jing Liu and Bhaskar D. Rao. Robust PCA via ℓ_0 - ℓ_1 regularization. *IEEE Trans. on Signal Processing*, pages 1-15, 2018.

[29] Xin Liu, Guoying Zhao, Jiawen Yao, and Chun Qi. Background subtraction based on low-rank and structured sparse decomposition. *IEEE Trans. on Image Processing*, 28(8):2502-2514, 2015.

[30] Zhichao Xue, Jing Dong, Yuxin Zhao, Chang Liu, and Ryad Chellali. Low-rank and sparse matrix decomposition via the truncated nuclear norm and a sparse regularizer. *Visual Computer*, 39(1):1549–1566, 2019.

[31] Chong Peng, Zhao Kang, and Qiang Cheng. A fast factorization-based approach to robust PCA. *in 2016 16th International Conference on Data Mining (ICDM)*. 2017.
[32] Qian Zhao, Deyu. Mengt, Zhongben Xu, Wangmeng Zuo, and Lei Zhang. Robust principal component analysis with complex noise. *in 2014 31st International Conference on Machine Learning (ICML)*, 2(1):1216-1226, 2017.
[33] Orazio Gallo, Natasha Gelfand, Wei-Chao Chen, Marius Tico, and Kari Pulli. Artifact-free high dynamic range imaging. *in 2009 International Conference on Signal and Image Processing (ICCP)*, 2009.