Double-Weighted Low-Rank Matrix Recovery Based on Rank Estimation

Zhengqin Xu[1,3,&] Huasong Xing[2,3,&] Shun Fang[2,3] ShiqianWu[2,3,*] Shoulie Xie[4]
[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
& First Author and Second Author contribute equally to this work.
{xuzhengqin,huasongxing,fangshun97,shiqian.wu}@wust.edu.cn slxie@i2r.a-star.edu.sg

Abstract

Robust principal component analysis (RPCA) has widely application in computer vision and data mining. However, the various RPCA algorithms in practical applications need to know the rank of low-rank matrix in advance, or adjust parameters. To overcome these limitations, an adaptive double-weighted RPCA algorithm is proposed to recover low-rank matrix accurately based on the estimated rank of the low-rank matrix and the reweighting strategy in this paper. More specifically, the Gerschgorin's disk theorem is introduced to estimate the rank of the low-rank matrix first. Then a double-weighted optimization model through two weighting factors for the low rankness and sparsity is presented. Finally an adaptive double weighted algorithm based on rank estimation is proposed, which can reweight the singular values of low-rank matrix and the sparsity of sparse matrix iteratively. Experimental results show that the proposed double-weighted RPCA algorithm outperforms the state-of-the-art RPCA methods.

1. Introduction

In the past decades, the word has entered the age of "Big Data". The information industry is now facing the fact that the size and the dimension of the data have reach an unprecedented scale and are still increasing at an unprecedented rate [1]. Although many kinds of data, such as image processing [2], video processing [3] and audio processing [4], are in high dimensions, their distributions still have low-dimensional manifolds. Low-rank matrix estimation has attracted increasing attention due to its overwhelming advantages in correctly and effectively acquiring and retaining such low-dimensional information [2].

One typical low-rank matrix estimation method is the low-rank matrix factorization (LRMF) [5], which factorizes the observed data $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ into two smaller ones $\boldsymbol{U} \in \mathbb{R}^{r\times m}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$, where $r \ll \min(m,n)$, such that $\boldsymbol{M} = \boldsymbol{U}^T\boldsymbol{V}$. A series of LRMF methods have been developed, such as the classical singular value decomposition (SVD) in $\ell_2 - norm$ [6], robust LRMF methods in $\ell_1 - norm$ [5] and the probabilistic method [7].

Another research focuses on the rank minimization, which decomposes the data matrix into a low-rank matrix $\boldsymbol{L} \in \mathbb{R}^{m\times n}$ and a small perturbation sparse matrix $\boldsymbol{S} \in \mathbb{R}^{m\times n}$, i.e., $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}$. Classical Principal Component Analysis (PCA) [8], which is the most widely tool to find the best approximation of the underlying low-rank structure of the observation data, can effectively estimate the low-rank matrix $\boldsymbol{L}$ by minimizing the rank of the matrix $\boldsymbol{M}$. However, the PCA processing is sensitive to outliers, which limits its applications. In [9], Candès et al. proposed Robust principal component analysis (RPCA) to address the robustness and offered a powerful framework for many practical applications [10, 11]. From an optimization viewpoint, RPCA is formulated as the following optimization problem:

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

where $\boldsymbol{M}, \boldsymbol{L}, \boldsymbol{S} \in \mathcal{R}^{m\times n}$, $rank(\boldsymbol{L})$ denotes the rank of matrix $\boldsymbol{L}$, $\|\boldsymbol{S}\|_0$ represents the $\ell_0 - norm$ which is the number of non-zero elements of $\boldsymbol{S}$, λ is the balance parameter between the two terms. Due to the penalty of sparse component, RPCA is more robust than PCA in the case of outliers. The minimization problem (1) is non-convex and is often relaxed to the following convex surrogate:

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

where $\|\boldsymbol{L}\|_* = \sum_1^{min(m,n)} \sigma_i(\boldsymbol{L})$ denotes the nuclear norm of $\boldsymbol{L}$, $\sigma_i(\boldsymbol{L})$ is the i-th singular value of matrix $\boldsymbol{L}$, $\|\boldsymbol{S}\|_1$

denotes the $\ell_1 - norm$ of matrix $\boldsymbol{S}$. Thus the nuclear norm minimization problem is used to replace the rank minimization problem, which can recover the low-rank matrix L efficiently by the Inexact Augmented Lagrange Multipliers (IALM) [12]. In this algorithm, the singular value soft threshold operator [13] is involved to solve the nuclear norm approximation problem as follows [14]:

$$\boldsymbol{L}^* = \arg\min_{\boldsymbol{L}} \|\boldsymbol{M} - \boldsymbol{L}\|_F^2/2 + \tau\|\boldsymbol{L}\|_* \tag{3}$$

where τ is the parameter maintaining the rankness of matrix $\boldsymbol{L}$.

Albeit the nuclear norm minimization (NNM) successfully obtained the low-rank matrix $\boldsymbol{L}^*$, it still has certain limitations. As can be seen, nuclear norm minimizes the rank of the matrix and inevitably reduces the singular values that exceed τ on the same scale. Therefore, one major limitation of this approach is that all the singular values are simultaneously and equally minimized. In other words, this ignores the prior knowledge in shrinking the signal values of the data matrix. More specifically, for background modelling, the first singular value represents the background and provides background information. Obviously, traditional NNM methods cannot deal with these similar problems.

To improve the flexibility of NNM, Gu et al. [15] proposed a weighted nuclear norm minimization (WNNM) model. The original nuclear norm is replaced by a weighted nuclear norm $\|\boldsymbol{L}\|_{\boldsymbol{W}} = \sum_1^{\min(m,n)} w_i\sigma_i(\boldsymbol{L})$, where w_i is the non-negative weight of the ith singular value. This method improves the accuracy of the recovered low rank matrix. But the parameters are manually adjusted for different applications, the generalization ability is limited in some applications. On the other hand, the rank of the target low-rank matrix $\boldsymbol{L}$ is assumed to be known in some applications. For example, in the most tasks of background separation, without considering the illumination, the rank of low-rank matrix is 1. Based on prior rank information, a partial sum of singular value (PSSV) minimization was given in [16] by

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_{p=r} + \lambda\|\boldsymbol{S}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{4}$$

where $\|\boldsymbol{L}\|_{\boldsymbol{W}} = \sum_1^p \sigma_i(\boldsymbol{L})$ denotes the sum of partial singular values after the rank. But it is not able to guarantee that the rank of the decomposed low-rank matrix is adaptively shrunk to the target rank. To this end, methods [17, 18] have been proposed to solve the rank estimation of low-rank matrix and adapatively reduce the singular value of the input matrix.

For the optimization to the sparse matrix $\boldsymbol{S}$, inferring by Hale et al. [19], the optimal solution $\boldsymbol{S}^*$ to the problem is

$$\boldsymbol{S}^* = \arg\min_{\boldsymbol{S}} \|\boldsymbol{M} - \boldsymbol{S}\|_F^2/2 + \tau\|\boldsymbol{S}\|_1 = \mathcal{S}_\tau[\boldsymbol{M}] \tag{5}$$

where the soft-thresholding operator $\mathcal{S}_\tau[\cdot]$ is proved to be a very effective operator that minimizes the $\ell_1 norm$ and guarantees that the solution $\boldsymbol{S}^*$ is the global minimum [16, 20]. However, similar to the deficiencies in the minimization of nuclear norms, there is a key difference τ, which is the dependence on magnitude, between the $\ell_1 - norm$ and the $\ell_0 - norm$. In other words, unlike the democratic penalization of the $\ell_0 - norm$, larger coefficients are penalized more heavily in the $\ell_1 - norm$ than the smaller coefficients. To address the imbalance, Candès et al. [21] proposed a weighted formulation of $\ell - 1 - norm$ designed to more democratically penalize nonzero coefficients. The weighted $\ell_1 - norm$ minimization function is defined as:

$$\arg\min_{\boldsymbol{S}} \|\boldsymbol{M} - \boldsymbol{L}\|_F^2/2 + \tau\|\boldsymbol{W_S} \circ \boldsymbol{S}\|_1 = \mathcal{S}_{\tau\boldsymbol{W_S}}[\boldsymbol{M}] \tag{6}$$

where $\boldsymbol{W_S}$ is the weight of the sparse matrix $\boldsymbol{S}$, $\circ$ denotes the element-wise product of the matrix. Peng et al. [22] proposed a nonuniform singular value thresholding (NSVT) operator to enhance low rank in NNM. By properly reweighting singular values for low-rank matrix and reweighting $\ell_1 - norm$ for sparse matrix, better matrix recovery performance can be obtained.

It is noted that the rank of the low-rank matrix $\boldsymbol{L}$ is always unknown in many practical applications. By considering the distribution of singular values of the low-rank matrix, we employ the Gerschgorin estimation method [23] to estimate the rank of the low-rank matrix. Inspired by the weighted $L1 - norm$ on the low-rank matrix [24, 17], we propose an adaptive weighting strategy to improve the performance of low-rank matrix restoration based on iteratively estimated rank. As the singular values contain the low-rank structure and the proportion of data information in the corresponding component direction, they are used to update the weights adaptively, which ensure that the rank of the recovered low-rank matrix is equal to the target rank. Motivated by reweighting the $L1 - norm$ of the weighted sparse signal to enhance the sparsity in [21], we introduced the reweighting strategy to the weighted $L1 - norm$ of the sparse matrix. By considering the two weighting factors for the low rankness and sparsity, we propose an adaptive dual-weighted low-rank matrix recovery optimization model with the estimated rank. This optimization problem is solved by employing alternating direction method of multipliers (ADMM) [12]. Experimental results that the proposed adaptive double-weighted RPCA approach significantly improves the accuracy and robustness of low-rank matrix recovery. In short, the main contributions of this work are summarized as follows:

- The Gerschgorin disc estimation method is introduced to effectively estimate the rank of the low-rank matrix without prior knowledge.

- A new double-weighted optimization model is proposed to recover the low-rank matrix and sparse matrix through incorporating two weighting factors for the rankness and sparsity.
- Also, an adaptive doube-weighted RPCA algorithm is proposed by employing ADMM method. This technique ensure the estimated rank to be equal to the target rank, and enhance the sparsity.
- A number of experiments are conducted to illustrate that the proposed method can be applied to various situations successfully, and outperforms the existing various RPCA approaches.

2. Double Weighted Low-Rank Matrix Recovery Based on Rank Estimation

2.1. Rank Estimation for Low-Rank Matrix

The rank of low-rank matrix is a vital parameter during the low-rank matrix recovery. For example, in background/foreground separation, the rank of background of subjects is always 1, while it is 2 sometimes when the illumination changes unevenly. Hence, the uncertainty of rank of low-rank matrix affects the robustness of the recovery algorithms. In this paper, Gerschgorin disk theorem [23] is employed to estimate the rank of a low-rank matrix.

According to [17], the received signal $\boldsymbol{M}$ from the sensor array in the noisy environment can be expressed as the sum of low-rank source signal matrix $\boldsymbol{L}$ and sparse noisy signal matrix $\boldsymbol{S}$.

The covariance matrix $\boldsymbol{R_M}$ of the matrix $\boldsymbol{M}$ with rank r can be defined as:

$$\boldsymbol{R_M} = \boldsymbol{M}\boldsymbol{M}^T \tag{7}$$

Eigenvalue decomposition of $\boldsymbol{R_M}$ is

$$\boldsymbol{R_M} = \boldsymbol{U_{R_M}}\boldsymbol{\Sigma_{R_M}}\boldsymbol{U_{R_B}^H} \tag{8}$$

where $\boldsymbol{U_{R_M}} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m]$ is the eigenvector matrix, and $\boldsymbol{\Sigma_{R_M}} = diag(\sigma_1, \sigma_2, \cdots, \sigma_m)$ is the eigenvalue matrix. Assuming, the rank of $\boldsymbol{R_M}$ is r, which is far smaller than m, $\boldsymbol{R_M}$ can be transformed by Gerschgorin disk theorem as follows:

$$\begin{aligned}\boldsymbol{R_M} &= \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{22} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m1} & R_{m2} & \cdots & R_{mm} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{R_{M1}} & \boldsymbol{R} \\ \boldsymbol{R}^H & R_{mm} \end{bmatrix}\end{aligned} \tag{9}$$

where $\boldsymbol{R_{M1}} \in \mathbb{R}^{(m-1)\times(m-1)}$ is obtained by deleting the last row and column of $\boldsymbol{R_M}$. Then compute the covariance matrix $\boldsymbol{R_{M1}}$ based on eigenvalue decomposition:

$$\boldsymbol{R_{M1}} = \boldsymbol{U_{M1}}\boldsymbol{\Sigma_1}\boldsymbol{U_{M1}^H} \tag{10}$$

$\boldsymbol{R_{M1}}$ can be defined similarly as $\boldsymbol{R_M}$, then a unitary transformed matrix $\boldsymbol{U} \in \mathbb{R}^{m\times m}(\boldsymbol{U}\boldsymbol{U}^H = \boldsymbol{I})$ is calculated:

$$\boldsymbol{U} = \begin{pmatrix} \boldsymbol{U_{M1}} & \boldsymbol{0} \\ \boldsymbol{0}^T & \boldsymbol{1} \end{pmatrix} \tag{11}$$

The transformed covariance matrix is obtained by

$$\begin{aligned}\boldsymbol{R_T} = \boldsymbol{U}^H\boldsymbol{R_M}\boldsymbol{U} &= \begin{pmatrix} \boldsymbol{U_{M1}^H}\boldsymbol{R_{M1}}\boldsymbol{U_{M1}} & \boldsymbol{U_{M1}^H}\boldsymbol{R} \\ \boldsymbol{R^H}\boldsymbol{U_{M1}} & R_{mm} \end{pmatrix} \\ &= \begin{pmatrix} \sigma_1' & 0 & 0 & \cdots & 0 & \rho_1 \\ 0 & \sigma_2' & 0 & \cdots & 0 & \rho_2 \\ 0 & 0 & \sigma_3' & \cdots & 0 & \rho_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma_{m-1}' & \rho_{m-1} \\ \rho_1^* & \rho_2^* & \rho_3^* & \cdots & \rho_{m-1}^* & R_{mm} \end{pmatrix}\end{aligned} \tag{12}$$

where $\rho_i = \boldsymbol{q}_i'^H\boldsymbol{R}$. The eigenvalues of $\boldsymbol{R_T}$ can be estimated using Gerschgorin disk theorem. Thus, the radii of the first (m-1) Gerschgorin's disk can be written as:

$$r_i = |\rho_i| = |\boldsymbol{q}_i'^H\boldsymbol{R}| \tag{13}$$

where the radius r_i of the ith Gerschgorin's disk depends on the size of $\boldsymbol{q}_i'^H\boldsymbol{R}$. If $\boldsymbol{q}_i'$, is the eigenvector in the sparse space, r_i will be significantly small and close to zero. If $\boldsymbol{q}_i'$, is the eigenvector of the low-rank part, r_i will be far from zero. Hence, the rank can be estimated by the heuristic decision rule:

$$\boldsymbol{GDE}(k) = r_k - \frac{D(n)}{m-1}\sum_{i=1}^{m-1} r_i \tag{14}$$

where $k = 1, 2, \cdots, m-2$, the adjustment factor $D(n)$ is a constant related to n. In this paper, we define $D(n) = 2.3/log(n)$. The rank of low-rank matrix is $r = k - 1$ when $\boldsymbol{GDE}(k)$ is negative the first time. After estimating the rank based on Gerschgorin disk theorem, we propose a double weighted low-rank matrix factorization algorithm based on rank estimation in the next section.

2.2. Double Weighted Low-Rank Matrix Recovery

In this section, an adaptive weighting RPCA algorithm is proposed to recover the low-rank matrix and the corrupted measurements by solving the following double-weighted optimization problem:

$$\arg\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_{\boldsymbol{W_L}} + \lambda\|\boldsymbol{W_S}\circ\boldsymbol{S}\|_1 \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{15}$$

where $\|\boldsymbol{L}\|_{\boldsymbol{W_L}} = \sum_i \omega_{\boldsymbol{L},i}\sigma_i(\boldsymbol{L})$, $\boldsymbol{W_L} = diag(\{\omega_{\boldsymbol{L},i}\}_{1\le i\le min(m,n)})$ is the weight of low-rank matrix $\boldsymbol{L}$, $\boldsymbol{W_S}$ is the weight of sparse matrix $\boldsymbol{S}$, the operator $\circ$ denotes the element-wise multiplication. $\boldsymbol{W_L}$ and $\boldsymbol{W_S}$ are determined in the iterative recovery algorithm.

Before solving the optimization problem (15), we introduce the following important theorem.

Theorem 1 [17] Given $\boldsymbol{Z} \in \mathbb{R}^{m\times n}$, $\boldsymbol{W_L} = diag(\{\omega_i\}_{1\le i\le min(m,n)})$, where $0 < \omega_i \le \omega_{i+1}, \tau > 0$. $\boldsymbol{Z}$ is decomposed by SVD as $\boldsymbol{Z} = \boldsymbol{U\Sigma_Z V}^T$, where $\boldsymbol{\Sigma_Z} = diag(\{\omega_i(\boldsymbol{Z})\}_{1\le i\le min(m,n)})$. Then the optimization

$$\arg\min_{\boldsymbol{X}} \|\boldsymbol{X}-\boldsymbol{Z}\|_F^2/2 + \tau\|\boldsymbol{X}\|_{\boldsymbol{W_L}} \tag{16}$$

has solution is $\hat{\boldsymbol{X}} = \boldsymbol{U}\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}}\boldsymbol{V}^T$ with $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{X}} = \mathcal{S}_{\tau\boldsymbol{W_L}}[\boldsymbol{\Sigma_Z}]$, where $\mathcal{S}_{\tau\boldsymbol{W_L}}[\boldsymbol{\Sigma_Z}]$ is the element-wise application of the soft-threshold operator:

$$\mathcal{S}(y) = sign(y)\max(|y|-\rho, 0) \tag{17}$$

Now we apply ADMM to solve the problem (15). The augmented Lagrangian function is:

$$\begin{aligned}\mathcal{L}(\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}) = &\|\boldsymbol{L}\|_{\boldsymbol{W_L}} + \lambda\|\boldsymbol{W_S}\circ\boldsymbol{S}\|_1 \\ &+ \langle \boldsymbol{Y}, \boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle \\ &+ \frac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2\end{aligned} \tag{18}$$

where $\langle .,.\rangle$ denotes matrix inner product, μ represents a positive penalty scalar, and $\boldsymbol{Y}$ is the Lagrangian multiplier.

In ADMM, an alternating method is used to solve the minimization of (18), which optimizes one variable while fixing the others. Thus, three sub-problems are formulated for three variables $\boldsymbol{L},\boldsymbol{S},\boldsymbol{Y}$.

$\boldsymbol{L}$ sub-problem: Fixed $\boldsymbol{S}$ and $\boldsymbol{Y}$, it follows from (18) that $\boldsymbol{L}$ can be obtained by

$$\begin{aligned}\boldsymbol{L}^* = &\arg\min_{\boldsymbol{L}} \|\boldsymbol{L}\|_{\boldsymbol{W_L}} + \langle \boldsymbol{Y}, \boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle \\ &+ \frac{\mu}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2 \\ &= \arg\min_{\boldsymbol{L}} \left\|(\boldsymbol{M}-\boldsymbol{S}+\mu^{-1}\boldsymbol{Y})-\boldsymbol{L}\right\|_F^2/2 \\ &+ \mu^{-1}\|\boldsymbol{L}\|_{\boldsymbol{W_L}}\end{aligned} \tag{19}$$

By defining $\boldsymbol{Y_L} = \boldsymbol{M}-\boldsymbol{S}+\mu^{-1}\boldsymbol{Y}$, the SVD of $\boldsymbol{Y_L}$ is $\boldsymbol{Y_L} = \boldsymbol{U_Y\Sigma_Y V_Y^T}$ with $\boldsymbol{\Sigma_Y} = diag\{\sigma_i(\boldsymbol{Y_L})\}$.

According to the theorem 1, the solution to (19) is $\boldsymbol{L^*} = \boldsymbol{U_Y\Sigma_Y^* V_Y^T}$, where $\boldsymbol{\Sigma_Y^*} = diag\{\sigma_i(\boldsymbol{L^*})\}$ with

$$\sigma_i(\boldsymbol{L^*}) = max(\sigma_i(\boldsymbol{Y_L}) - \mu^{-1}\omega_i, 0) \tag{20}$$

Weighted kernel norm in non-uniform soft threshold is $\mathcal{S}_{\mu^{-1}}[\boldsymbol{\Sigma}]_{ii} = \max(\sigma_i - \omega_i\mu^{-1}, 0)$. It is noted that the singular value sequence is arranged in non-negative descending order. In order to make the rank of the optimal low-rank matrix $\boldsymbol{L}^*$ equal to the target rank, we hope that the rth singularity of the low-rank matrix obtained after the solution of the soft threshold operator is greater than zero, and the $r+1$th singular value is less than zero. In the process of solving the optimal low-rank matrix $\boldsymbol{L}^*$ with the same rank as the target rank, we hope to reduce the interference of the sparse matrix to the low-rank matrix as much as possible.

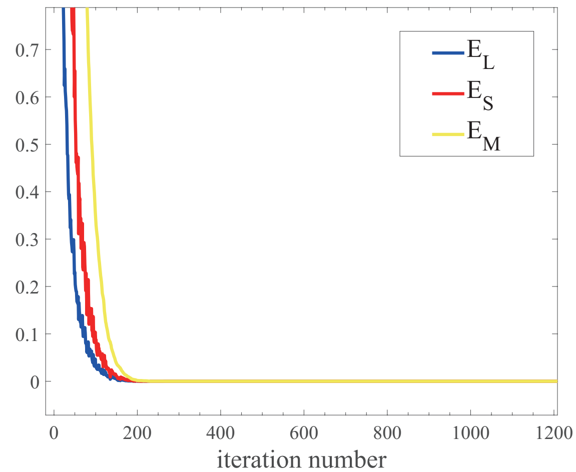


Fig. 1. The value of $\boldsymbol{E_L}$, $\boldsymbol{E_S}$ and $\boldsymbol{E_M}$ with the number of iterations. Assuming that $\boldsymbol{M} \in \boldsymbol{R}^{m\times n}$, $m = 10000$, $n = 20$, $rank(\boldsymbol{L}) = 3$, and the corrupted rate $\rho = 0.05$.

Thus, we define the threshold of the soft threshold operator by $\sigma'_{r+1} = \mathcal{S}_\rho(\sigma_{r+1}) = 0$.

If the estimated rank of the low-rank matrix is r in the last iteration, then the weight $\boldsymbol{W_L}$ is updated by

$$\omega_i = \frac{\mu\sigma_{r+1}(\boldsymbol{Y_L})^2}{\sigma_i(\boldsymbol{Y_L})} \tag{21}$$

By combining Eqs. (20) and (21), it is derived

$$\sigma_i(\boldsymbol{L^*}) = \begin{cases}\sigma_i(\boldsymbol{Y_L}) - \frac{\sigma_{r+1}(\boldsymbol{Y_L})^2}{\sigma_i(\boldsymbol{Y_L})}, & if \quad i\le r \\ 0, & if \quad i > r\end{cases} \tag{22}$$

Thus, the rank of the optimization solution $\boldsymbol{L^*} = \boldsymbol{U_Y\Sigma_Y^* V_Y^T}$ is conformed to the estimated rank r. We use the re-weighted strategy $\omega_i = \frac{\mu(\sigma_{r+1}(\boldsymbol{Y_L}))^2}{\sigma_i(\boldsymbol{Y_L})}$, but $(\sigma_{r+1}(\boldsymbol{Y_L}))^2$ and μ are fixed during every iteration. According to the reference in [21], our weighted stratege in Eq. (21) also makes the weighted kernel norm greatly approach the rank function. The optimization solution to the minimization (19) can be directly obtained by Eq. (22).

$\boldsymbol{S}$ sub-problem: Fixed $\boldsymbol{L}$ and $\boldsymbol{Y}$, the following optimization problem is obtained from (16):

$$\begin{aligned}\boldsymbol{S}^* = &\arg\min_{\boldsymbol{S}} \lambda\|\boldsymbol{W_S}\circ\boldsymbol{S}\|_1 + \langle \boldsymbol{Y}, \boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\rangle \\ &+ \mu\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{S}\|_F^2/2 \\ &= \arg\min_{\boldsymbol{S}} \left\|(\boldsymbol{M}-\boldsymbol{L}+\mu^{-1}\boldsymbol{Y})-\boldsymbol{S}\right\|_F^2/2 \\ &+ \lambda\mu^{-1}\|\boldsymbol{W_S}\circ\boldsymbol{S}\|_1\end{aligned} \tag{23}$$

This problem can be solved by the element-wise soft-threshold operation:

$$\boldsymbol{S}^* = \mathcal{S}_{\lambda\mu^{-1}\boldsymbol{W_S}}[\boldsymbol{M}-\boldsymbol{L}+\mu^{-1}\boldsymbol{Y}] = \mathcal{S}_{\lambda\mu^{-1}\boldsymbol{W_S}}[\boldsymbol{Y_S}] \tag{24}$$

where $\boldsymbol{Y_S} = \boldsymbol{M}-\boldsymbol{L}+\mu^{-1}\boldsymbol{Y}$, and $\mathcal{S}_\rho(\cdot)$ is the soft-threshold function as in (15).

In order to enhance the sparsity and by adopting the reweighting skills in [25], the weight $\boldsymbol{W_S}$ can be updated by

$$\boldsymbol{W_{S,i,j}} = \frac{\mu}{\lambda(|\boldsymbol{S}_{i,j}| + \varepsilon)} \tag{25}$$

$\boldsymbol{W}_{\boldsymbol{S},i,j}$ is the ijth element of matrix $\boldsymbol{W_S}$.
$\boldsymbol{Y}$ sub-problem: The updated function of $\boldsymbol{Y}$ is written as:

$$\boldsymbol{Y}_{k+1} = \boldsymbol{Y}_k + \mu(\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1}) \tag{26}$$

The entire procedure to solve problem (15) is summarized in Algorithm 1.

Algorithm 1

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

Initialization: $\boldsymbol{S}_0 = \boldsymbol{Y}_0 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$, $k = 0$, k_{max}, r is estimated by Eq. (14), $\omega_{(\boldsymbol{S},0),i,j} = 1$, $\varepsilon = 10^{-7}$, $h = 0$, h_{max}.
while $\|\boldsymbol{M} - \boldsymbol{L} - \boldsymbol{S}\|_F/\|\boldsymbol{M}\|_F \geq \varepsilon$ or k is less than the maximum number of iterations **do**
  while $h < h_{max}$ **do**
    compute $\boldsymbol{Y}_{L,k} = \boldsymbol{M} - \boldsymbol{S}_{k-1} + \mu^{-1}\boldsymbol{Y}_{k-1}$ and its SVD $\boldsymbol{Y_{L,k}} = \boldsymbol{U_{Y,k}\Sigma_{Y,k}V^T_{Y,k}}$ with $\boldsymbol{\Sigma_{Y,k}} = diag\{\sigma_i(\boldsymbol{Y_{L,k}})\}$;
    update the weight $\boldsymbol{W}_{\boldsymbol{L},k}$ by $\omega_{i,k} = \frac{\mu\sigma_{r+1}(\boldsymbol{Y_{L,k}})^2}{\sigma_i(\boldsymbol{Y_{L,k}})}$;
    compute $\sigma_i(\boldsymbol{L}_k) = \max(\sigma_i(\boldsymbol{Y_{L,k}}) - \mu^{-1}\omega_i, 0)$ and $\boldsymbol{L}_k = \boldsymbol{U_{Y,k}\Sigma_{L,k}V^T_{Y,k}}$ with $\boldsymbol{\Sigma_{L,k}} = diag\{\sigma_i(\boldsymbol{L}_k)\}$;
    compute $\boldsymbol{Y_{S,k}} = \boldsymbol{M} - \boldsymbol{L}_k + \mu^{-1}\boldsymbol{Y}_{k-1}$;
    compute $\boldsymbol{S}_k = \mathcal{S}_{\lambda\mu^{-1}\boldsymbol{W_{S,h}}}[\boldsymbol{Y_{S,k}}]$;
    compute $\boldsymbol{Y}_k = \boldsymbol{Y}_{k-1} + \mu(\boldsymbol{M} - \boldsymbol{L}_k - \boldsymbol{S}_k)$;
    update r according to Eq. (14)
  end while
  update $\omega_{(\boldsymbol{S},h),i,j} = \frac{\mu}{\lambda(|\boldsymbol{S}_{k,i,j}|+\varepsilon)}$, $\boldsymbol{S}_k \to \boldsymbol{S}_0$, $\boldsymbol{Y}_k \to \boldsymbol{Y}_0$, $k = 0$
end while
Output: $\boldsymbol{L}^*, \boldsymbol{S}^*$

*Remark*1: Although the mathematical proof of the convergence of the proposed algorithm is challenging, the experimental results shown in Fig.1 lead to the following claim.

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

$$E_{\boldsymbol{L}} = \lim_{k\to\infty} \|\boldsymbol{L}_{k+1} - \boldsymbol{L}_k\|_F = 0 \tag{27}$$

$$E_{\boldsymbol{S}} = \lim_{k\to\infty} \|\boldsymbol{S}_{k+1} - \boldsymbol{S}_k\|_F = 0 \tag{28}$$

$$E_{\boldsymbol{M}} = \lim_{k\to\infty} \|\boldsymbol{M} - \boldsymbol{L}_{k+1} - \boldsymbol{S}_{k+1}\|_F = 0 \tag{29}$$

3. Experimental Results

In this section, we report the performance of the proposed algorithm on the scene background initialization problem, and compare with several the state-of-the art RPCA algorithms: RPCA[9], WNNM[15], PSSV[16], and AccAltProj[26].

3.1. Visual Effects

Without considering the effects of illumination, in the background modeling task the rank is usually unchanged, so the rank $r = 1$ of low-rank matrix representing the background is known. In this experiment, Scene Background Initialization (SBI) dataset[1] [27] is considered. In particular, the sub-datasets CAVIAR1 and Hall & Monitor are studied, where the former contains 610 frames with size 384×256 and the latter contains 296 frames with size 352×240. Each data can be expressed by a matrix and we use various RPCA algorithms to decompose the matrix as the low-rank part representing the background and the sparse one representing the moving object.

CAVIAR1 dataset shows the scenario that persons walk slowly along with the corridor, whose initial state is shown in Fig.2. We show the background recovery results of 5-th, 127-th, 249-th, 371-th and 493-th in the dataset. The first column is the original video frame, the second one is the ground truth, and the other columns are the results of RPCA, PSSV, WNNM, AccAltProj and ours. As shown in Fig.2, RPCA can't restore the background well. PSSV performs better than RPCA, but it is still bad at separating the background. WNNM presents good results, on the condition that the parameters are adjusted continually. It is observed that our algorithm and AccAltProj can recover the background well, while AccAltProj has the problem that the rank of low-rank needs to be known in advance.

Hall&Monitor dataset describes the scenario that people are walking in the aisle and passers-by turn up in the same region during most frames, which is a big challenge to the low-rank matrix recovery. We show the recovered background results of 5-th, 64-th, 123-th, 182-th and 241-th frames in Fig.3. It is shown that RPCA and PSSV cannot obtain good background results. After adjusting the parameters, WNNM performs better than PSSV, but worse than AccAltProj and the proposed algorithm. According to the visual effects the proposed method looks better than AccAltProj.

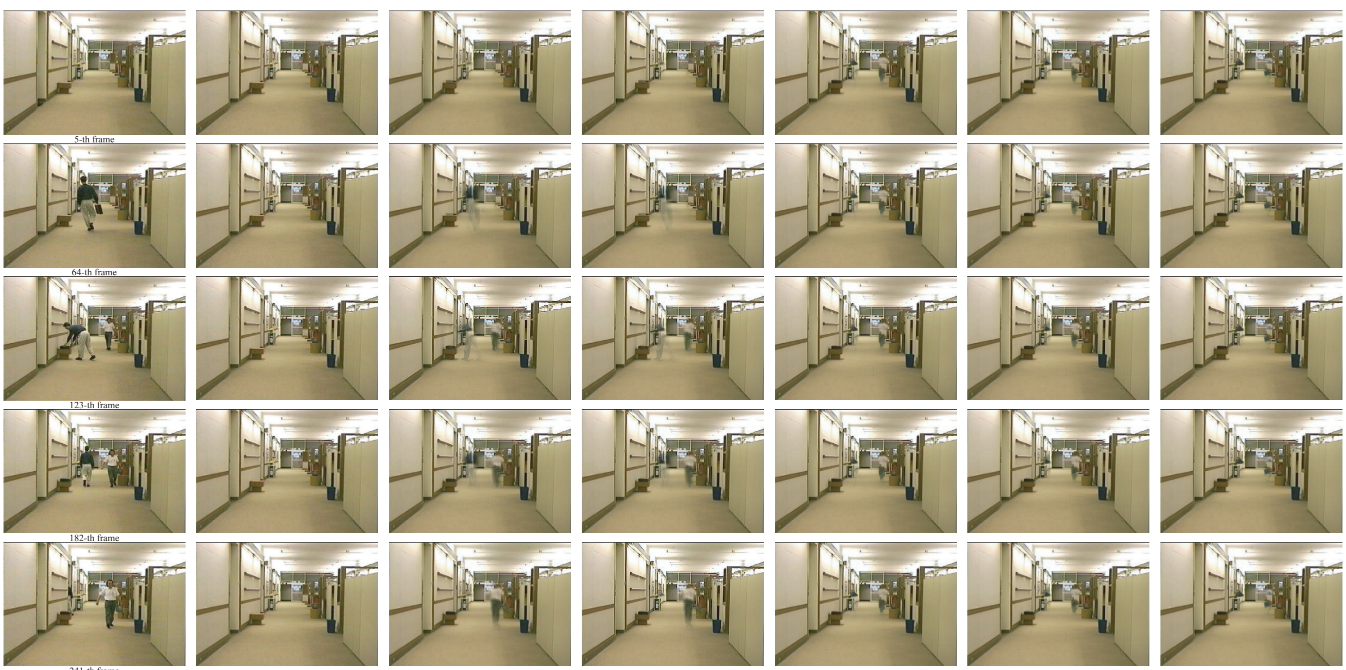
We compared more datasets shown in the supplementary materials. Similar to Hall & Monitor dataset which is a big challenge to the low-rank recovery, our algorithm achieves much better results than other algorithms on the other datasets.

When the environment exists obvious illumination

[1] https://sbmi2015.na.icar.cnr.it/SBIdataset.html

(a)Original Frame (b)Ground Truth (c)RPCA (d)PSSV (e)WNNM (f)AccAltProj (g)Ours

Fig. 2. Comparison of different algorithms with different frames on CAVIAR1 dataset



(a)Original Frame (b)Ground Truth (c)RPCA (d)PSSV (e)WNNM (f)AccAltProj (g)Ours

Fig. 3. Comparison of different algorithms with different frames on Hall&Monitor dataset

change, such as the Arch dataset[2], the rank of the background low-rank matrix $r \neq 1$. At this time, the rank of low-rank matrix $r = 2$. Both PSSV and AccAltProj need to input the rank information in advance. The accuracy of the prior rank on the performances of PSSV, AccAltProj and our algorithm through the Archdataset with illumination changes are shown in Fig.4.

We find that when PSSV and AccAltProj are affected by

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

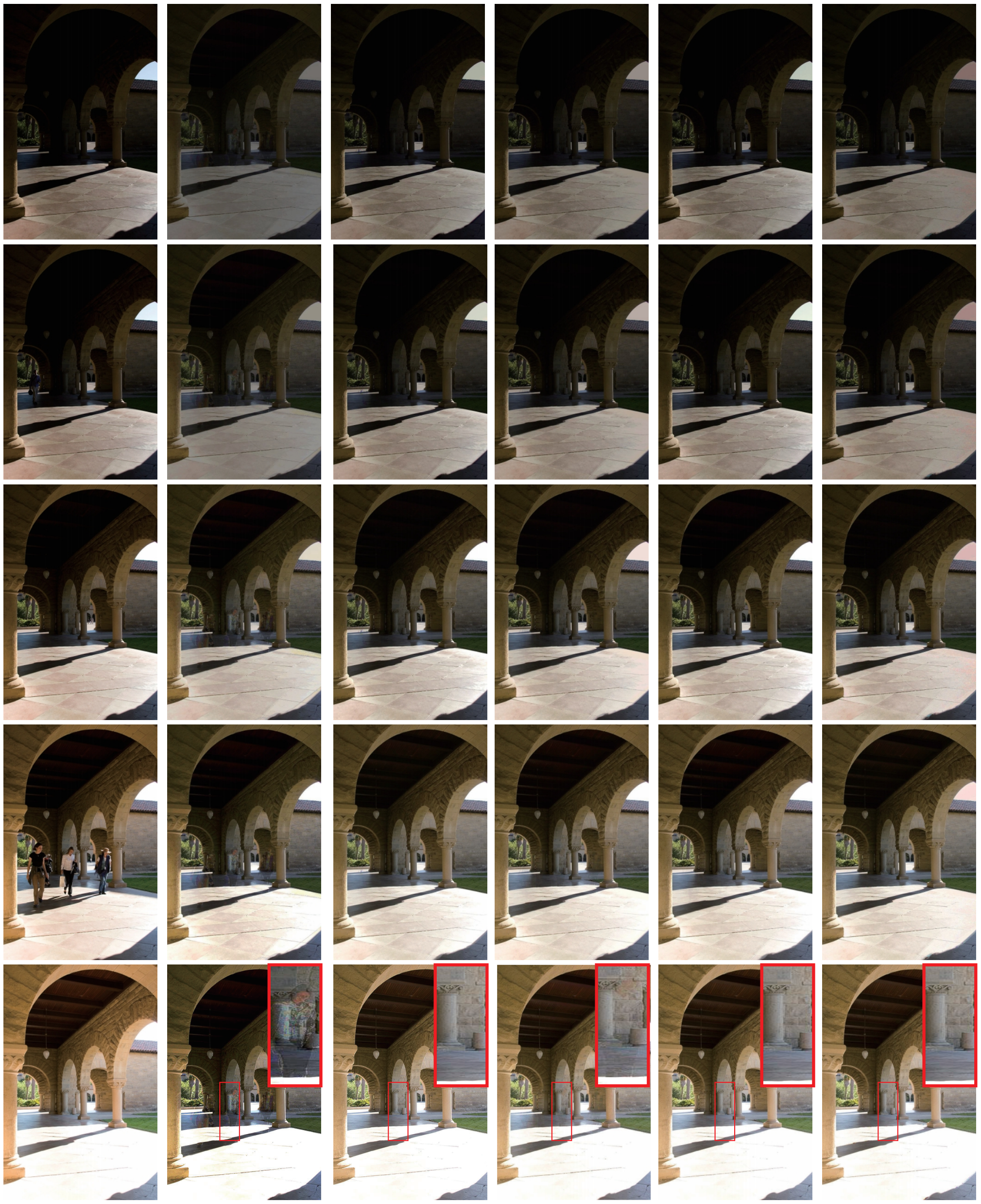


Fig. 4. Comparison of different algorithms with different frames on Arch dataset

CAVIAR1 dataset	AGE	pEPs%	pCEPS%	MSSSIM	PSNR	CQM
RPCA	5.9288	4.8319	4.0517	0.8515	24.5148	23.8925
PSSV	5.8004	4.718	3.9408	0.8566	24.8261	24.1782
WNNM	2.6332	0.353	0.2502	0.9906	34.689	33.5397
AccAltProj	**2.1352**	0.3225	0.2177	**0.9913**	35.2311	34.0159
Ours	2.1507	**0.3113**	**0.2014**	0.9912	**35.2972**	**34.0723**

Table 1. Performance of one frame from CAVIAR1 dataset among different algorithms

the wrong rank prior, their accuracies of recovered low-rank matrix are not good. As shown in the areas with red squares in Fig.5, when the rank $r = 1$ is used as the prior rank of Arch dataset, obvious shadows exist in AccAltProj and

Hall&Monitor dataset	AGE	pEPs%	pCEPS%	MSSSIM	PSNR	CQM
RPCA	4.1655	2.5829	1.3767	0.9384	28.0271	28.0838
PSSV	4.0974	2.3828	1.2157	0.9425	28.3952	28.4093
WNNM	3.3357	1.2109	0.5575	0.9623	29.9583	29.8644
AccAltProj	3.2426	1.2334	0.5942	0.9634	30.1149	29.9813
Ours	**3.1563**	**0.9387**	**0.3007**	**0.9666**	**30.2977**	**30.1575**

Table 2. Performance of one frame from Hall&Monitor dataset among different algorithms

shallow shadows in PSSV. If the accurate rank information is given, we can see that PSSV and AccAltProj have good results. By comparing the second and third columns and the fourth and fifth columns, it can be seen that AccAltProj is more dependent on the accuracy of the prior rank than PSSV. Our algorithm doesn't need the prior rank and is not affected by the accuracy of the prior rank, which verifies that our algorithm has better robustness.

Through the above two experiments, it can be seen that when the accurate prior rank is known, AccAltProj can achieve good results in both large-sample dataset Hall & Monitor dataset and small-sample Archdataset. But when an accurate prior rank cannot be obtained, the performance of AccAltProj will be severely affected. PSSV has superior performance in the case of small samples, and it can still maintain a certain performance in the case of inaccurate prior rank. But in large sample dataset, the performance of PSSV algorithm is insufficient. Our algorithm can guarantee superior performance in both large dataset and small one, and is not affected by the prior rank accuracy, making the algorithm more robust.

3.2. Performance Evaluation

Six metrics for background estimation are used for evaluating results provided by SBI dataset: AGE, pEPs, pCEPs, PSNR, MS-SSIM and CQM. We select a frame picture hardest to be recovered to compare with the GT from the dataset, whose results for two datasets are shown in Table 1 and 2 and the best ones are marked boldly. For some of pictures, the visual effects between our algorithm and others differ little, but it is observed that ours is best among the evaluation metrics.

4. Conclusion

In this paper, we propose an adaptive double-weighted RPCA algorithm based on the rank estimation. The proposed algorithm can estimate the rank of low-rank matrix according to the Gerschgorin's disk. In addition, based on the estimated rank estimation, an adaptive double-weighting strategy is presented to recover the low-rank and sparse matrices accurately and efficiently. Experimental results over the background initialization shows that the proposed adaptive double-weighted RPCA algorithm outperforms other existing RPCA approaches.

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

References

[1] J Wright and Y Ma. High-dimensional data analysis with low-dimensional models: Principles, computation, and applications. *Cambridge University Press*, 3:12, 2021.

[2] Zhiyuan Zha, Xin Yuan, Bihan Wen, Jiantao Zhou, Jiachao Zhang, and Ce Zhu. From rank estimation to rank approximation: Rank residual constraint for image restoration. *IEEE Transactions on Image Processing*, 29:3254–3269, 2019.

[3] Majid Moradikia, Sadegh Samadi, Hamid Reza Hashempour, and Mujdat Cetin. Video-sar imaging of dynamic scenes using low-rank and sparse decomposition. *IEEE Transactions on Computational Imaging*, 7:384–398, 2021.

[4] Jie Pu, Yannis Panagakis, and Maja Pantic. Learning differentiable sparse and low rank networks for audio-visual object localization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8668–8672. IEEE, 2020.

[5] Qian Zhao, Deyu Meng, Zongben Xu, Wangmeng Zuo, and Yan Yan. l_1-norm low-rank matrix factorization by variational bayesian method. *IEEE transactions on neural networks and learning systems*, 26(4):825–839, 2015.

[6] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 720–727, 2003.

[7] Yu Kong, Ming Shao, Kang Li, and Yun Fu. Probabilistic low-rank multitask learning. *IEEE transactions on neural networks and learning systems*, 29(3):670–680, 2017.

[8] Ian Jolliffe. Principal component analysis. *Encyclopedia of statistics in behavioral science*, 2005.

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

[10] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. Domain generalization and adaptation using low rank exemplar svms. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1114–1127, 2017.

[11] Shijie Xiao, Mingkui Tan, Dong Xu, and Zhao Yang Dong. Robust kernel low-rank representation. *IEEE transactions on neural networks and learning systems*, 27(11):2268–2281, 2015.

[12] Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *arXiv preprint arXiv:1109.0367*, 2011.

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

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

[15] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision*, 121(2):183–208, 2017.

[16] Tae-Hyun Oh, Yu-Wing Tai, Jean-Charles Bazin, Hyeongwoo Kim, and In So Kweon. Partial sum minimization of singular values in robust pca: Algorithm and applications. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):744–758, 2015.

[17] Zhengqin Xu, Yang Lu, Jiaxing Wu, Rui He, Shiqian Wu, and Shoulie Xie. Adaptive weighted robust principal component analysis. In *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 19–24. IEEE, 2020.

[18] Zhengqin Xu, Rui He, Shoulie Xie, and Shiqian Wu. Adaptive rank estimate in robust principal component analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6577–6586, 2021.

[19] Elaine T Hale, Wotao Yin, and Yin Zhang. Fixed-point continuation for ell_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

[20] David L Donoho and Iain M Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

[21] Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ 1 minimization. *Journal of Fourier analysis and applications*, 14(5):877–905, 2008.

[22] Yigang Peng, Jinli Suo, Qionghai Dai, and Wenli Xu. Reweighted low-rank matrix recovery and its application in image restoration. *IEEE transactions on cybernetics*, 44(12):2418–2430, 2014.

[23] Yuanming Guo, Wei Li, Junyuan Shen, Jinjun Zhang, and Chengguang Shao. Source number estimation with gerschgorin radii by reduced-rank covariance matrix. 2014.

[24] Hengyou Wang, Yigang Cen, Zhiquan He, Zhihai He, Ruizhen Zhao, and Fengzhen Zhang. Reweighted low-rank matrix analysis with structural smoothness for image denoising. *IEEE Transactions on Image Processing*, 27(4):1777–1792, 2017.

[25] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[26] HanQin Cai, Jian-Feng Cai, and Ke Wei. Accelerated alternating projections for robust principal component analysis. *The Journal of Machine Learning Research*, 20(1):685–717, 2019.

[27] Lucia Maddalena and Alfredo Petrosino. Towards benchmarking scene background initialization. In *International conference on image analysis and processing*, pages 469–476. Springer, 2015.