Robust Low-Rank Matrix Recovery via Hybrid Ordinary-Welsch Function

Zhi-Yong Wang, Hing Cheung So, *Fellow, IEEE*, and Abdelhak M. Zoubir, *Fellow, IEEE*

***Abstract*—As a widely-used tool to resist outliers, the correntropy criterion or Welsch function has recently been exploited for robust matrix recovery. However, it down-weighs all observations including uncontaminated data. On the other hand, its implicit regularizer (IR) cannot achieve sparseness, which is a desirable property in many practical scenarios. To address these two issues, we devise a novel M-estimator called hybrid ordinary-Welsch (HOW) function, which only down-weighs the outlier-contaminated data, and the IR generated by the HOW can attain sparseness. To verify the effectiveness of the HOW function, we apply it to robust matrix completion and principal component analysis. An efficient algorithm is developed and we prove that any generated limit point is a critical point. Finally, extensive experimental results based on synthetic and real-world data demonstrate that the proposed approach outperforms the state-of-the-art methods in terms of recovery accuracy and runtime.**

***Index Terms*—Low-rank matrix recovery, matrix factorization, sparsity, Welsch function, robust matrix completion, robust principal component analysis.**

I. Introduction

LOW-RANK matrix recovery (LRMR) refers to seeking the potential low-dimensional structure from a degraded matrix with possibly missing and/or noisy entries. LRMR has been widely used in numerous real-life applications such as picture recovery [1], [2], hyperspectral image restoration [3], shadow removal [4], [5] and video separation [6]. There are two main strategies to achieve LRMR, namely, rank minimization [4], [7] and factorization-based low-rank approximation [8], [9]. The former performs restoration via imposing a rank constraint on the recovered matrix. However, direct rank minimization is NP-hard since the rank is discrete and nonconvex. To alleviate this problem, similar to replacing the ℓ_0-norm with the ℓ_1-norm in sparse signal recovery, nuclear norm minimization is exploited, which is the tightest convex relaxation of rank minimization [4]. In addition, many efficient algorithms, including singular value thresholding (SVT) [10] and accelerated proximal gradient with linesearch (APGL) [11], have been developed. However, they require full singular value decomposition (SVD) at each iteration, which is computationally demanding, particularly for large-scale problems.

To avoid SVD, low-rank matrix factorization methods [8], [9] corresponding to the second strategy, have been proposed, whose idea is to utilize the product of two much smaller matrices to approximately substitute the degraded matrix so that the low-rank property is automatically fulfilled. Although the optimization problem based on factorization is nonconvex, empirical results in many works [12], [13] demonstrate that its low-rank solution can be as reliable as those obtained by solving the convex nuclear norm minimization. Recent theoretical studies [14], [15] have shown that the factorization-based LRMR has no spurious local minima and fulfills the strict saddle property which requires the cost function to have a directional negative curvature at all critical points but local minima. In other words, this means that global optimality can be achieved and thus we adopt the matrix factorization approach in this article.

While the derivation of the aforementioned schemes is based on the ℓ_2-norm, which is only optimal for Gaussian noise, non-Gaussian gross errors occur in many practical scenarios [16], calling for robust variants. Here, robustness signifies insensitivity to deviations from the model, which aims to decrease the impact of outliers on the data processing results [17]. A commonly-used strategy to combat outliers is to utilize the ℓ_1-norm, resulting in many optimization methods such as low-rank matrix fitting (LMaFit) [9], robust matrix factorization by majorization minimization (RMF-MM) [18], practical low-rank matrix approximation via robust ℓ_1-norm (RegL_1) [19] and semisoft go decomposition (SSGoDec) [20]. It is known that the ℓ_1-norm has a bias problem and is still vulnerable to large gross errors, even though it is favorable due to its convexity [21], [24]. Moreover, if the inherent model is very sparse, the sparse regularizer using the ℓ_1-norm can overestimate the number of non-zero coefficients [25], [26]. Thus, employing a nonconvex function [24], [27] as the loss function or a regularizer for LRMR when the observed data are contaminated by big sparse outliers has been recently suggested. For example, the ℓ_p-norm with $0 < p > 1$ is exploited and efficient algorithms including the ℓ_p-alternating direction method of multipliers (ℓ_p-ADMM) [28] and the joint Schatten p-norm and the ℓ_p-norm minimization ($S_p\ell_p$RMC) [29], are developed. Although they can resist outliers well, there are still two open issues. First, unlike the

Manuscript received 21 December 2022; revised 10 April 2023; accepted 20 June 2023. Date of publication 7 July 2023; date of current version 18 July 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiao Fu. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China under Project CityU 11207922. *(Corresponding author: Hing Cheung So.)*

Zhi-Yong Wang and Hing Cheung So are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: zhiyowang4-c@my.cityu.edu.hk; hcso@ee.cityu.edu.hk).

Abdelhak M. Zoubir is with the Signal Processing Group, Technische Universität Darmstadt, 64283 Darmstadt, Germany (e-mail: zoubir@spg.tu-darmstadt.de).

This article has supplementary downloadable material available at https://doi.org/10.1109/TSP.2023.3290353, provided by the authors.

Digital Object Identifier 10.1109/TSP.2023.3290353

1053-587X © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

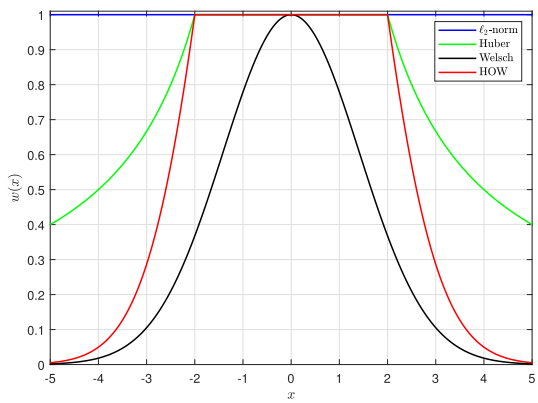

Fig. 1. Different weight functions with $\sigma = 2$ and $c = 2$.

ℓ_1-norm as the regularizer, which can be solved efficiently by soft thresholding, the nonconvex ℓ_p-norm minimization except for two special cases of $p = \frac{1}{2}$ [30] and $p = \frac{2}{3}$ [31], is difficult to handle since the closed-form expression of its proximity operator does not exist. The second issue is to determine an appropriate value of p, which is related to the intensity of impulsive noise. Although the p-Huber function [32] via modifying the ℓ_p-norm is devised to make the resultant optimization problem easy to solve, different values of p are needed through trial-and-error to yield excellent performance.

On the other hand, correntropy and adopting the Welsch function as the nonconvex error measure have achieved considerable success in image inpainting [33], [34], robust principal component analysis (RPCA) [27], [35], compressed sensing [36] and subspace clustering [37], [38]. It should be noted that maximizing correntropy with a Gaussian kernel is equivalent to minimizing the Welsch function. Based on the multiplicative form of half-quadratic optimization, He et al. [33] apply the Welsch function to robust matrix completion (RMC), and suggest two algorithms called half-quadratic power factorization (HQ-PF) and half-quadratic alternating steepest descent (HQ-ASD). Besides, RPCA methods based on the Welsch function via the additive form of half-quadratic optimization, namely, implicit regularizer (IR) and GoDec+, are developed in [27] and [38], respectively. Compared with the ℓ_1-norm, the Welsch function is insensitive to large outliers since its influence function is redescending [39]. Nevertheless, there are two limitations for the Welsch function: (i) a good loss function should reduce the weights of outliers while keeping those of 'normal' data fixed. Here, 'normal' data refer to observations without noise or with only Gaussian noise. However, the Welsch function down-weighs all data including those 'normal' values (See Fig. 1); and (ii) The IR produced by the Welsch function in [27], [40] cannot achieve sparseness.

In this article, we devise a novel loss function called **h**ybrid **o**rdinary-**W**elsch (HOW) in which 'ordinary' means the quadratic function or the ℓ_2-norm, which can overcome the above-mentioned drawbacks. That is to say, the proposed function only changes the weights of outlier-contaminated entries when processing the degraded matrix. In addition, the IR generated by HOW can attain sparseness. To the best of our knowledge, we are the first to provide a sparse regularization term corresponding to the correntropy/Welsch function, and the expression of its proximity operator is theoretically analyzed. In addition, the HOW function is exploited for robust LRMR via the matrix factorization formulation. Compared with existing loss functions, the effectiveness of HOW is demonstrated by performing two typical tasks of robust LRMR, namely, RMC and RPCA, and the corresponding algorithms are referred to as robust matrix completion via HOW (RMC-HOW) and robust principal component analysis via HOW (RPCA-HOW), respectively. The main contributions of our article are:

1) We devise the HOW function, which does not down-weigh the 'normal' data, and is not sensitive to gross errors particularly with big magnitudes. In fact, the Welsch function is a special case of the proposed function.
2) The IR of the HOW function can achieve sparseness, and its properties are theoretically proved. Furthermore, the closed-form expression of its proximity operator is analyzed.
3) By utilizing the Legendre-Fenchel transform, we also convert the complicated optimization formulation into a quadratic problem, resulting in two efficient algorithms based on matrix factorization for RMC and RPCA.
4) We conduct extensive experiments on both synthetic and real-life data to compare the proposed algorithms with the state-of-the-art methods. Results demonstrate that our approach has better low-rank and sparse recovery performance and is faster than the competitors by several or even ten orders.

The remainder of this article is organized as follows. In Section II, we introduce notations and related works. The new M-estimator and its properties are presented in Section III. In Section IV, we apply HOW to RMC and RPCA, and propose two efficient algorithms, whose convergences are guaranteed. Numerical simulation results based on synthetic data as well as real-world images and videos for RMC and RPCA are included in Section V. Finally, conclusions are drawn in Section VI.

II. Preliminaries

In this section, notations are provided and related works are reviewed.

A. Notations

Scalars and matrices are represented by italic letters and bold upper-case letters, respectively. The (i, j) entry of a matrix $\boldsymbol{A}$ is denoted by A_{ij}, and $(\cdot)^{-1}$ and $(\cdot)^T$ are the inverse and transpose operators, respectively. Let $\Omega \subset \{1, \ldots, m\} \times \{1, \ldots, n\}$ be the index set of the observed entries of $\boldsymbol{A}$ while Ω^c is the complement of Ω and $(\cdot)_\Omega$ is a projection operator, defined as:

$$[\boldsymbol{A}_\Omega]_{ij} = \begin{cases} A_{ij}, & \text{if } (i,j) \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

In addition, $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$ is the Frobenius norm of $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and $\mathrm{vec}(\boldsymbol{A})$ is a vector generated by vectorizing $\boldsymbol{A}$. The Frobenius inner product of $\boldsymbol{A}$ and $\boldsymbol{B}$ with the same dimensions is $\langle \boldsymbol{A}, \boldsymbol{B} \rangle = \mathrm{trace}(\boldsymbol{A}^T \boldsymbol{B})$, which is equal to the

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DIFFERENT LOSS FUNCTIONS AND THEIR CORRESPONDING WEIGHT FUNCTIONS

	Quadratic	Huber	Welsch	HOW
$g(x)$	$\frac{x^2}{2}$	$\begin{cases} x^2/2, & \lvert x\rvert \le c \\ c\lvert x\rvert - \frac{c^2}{2}, & \lvert x\rvert > c \end{cases}$	$\frac{\sigma^2}{2}\left(1 - e^{\frac{-x^2}{\sigma^2}}\right)$	$\begin{cases} x^2/2, & \lvert x\rvert \le c \\ \frac{\sigma^2}{2}\left(1 - e^{\frac{c^2-x^2}{\sigma^2}}\right) + \frac{c^2}{2}, & \lvert x\rvert > c \end{cases}$
$w(x) = \frac{g'(x)}{x}$	1	$\begin{cases} 1, & \lvert x\rvert \le c \\ c \cdot \mathrm{sign}(x)/x, & \lvert x\rvert > c \end{cases}$	$e^{\frac{-x^2}{\sigma^2}}$	$\begin{cases} 1, & \lvert x\rvert \le c \\ e^{\frac{c^2-x^2}{\sigma^2}}, & \lvert x\rvert > c \end{cases}$

inner product of $\langle \mathrm{vec}(\boldsymbol{A}), \mathrm{vec}(\boldsymbol{B})\rangle$. Unless stated otherwise, the matrix norm refers to the Frobenius norm, i.e., $\|\boldsymbol{A}\| = \|\boldsymbol{A}\|_F$. Finally, $|a|$ stands for the absolute value of a.

B. Related Works

1) Low-Rank Matrix Recovery: In this subsection, we mainly concentrate on recent works on matrix completion (MC), because PCA can be considered as a special case of MC when there are no missing entries. A direct formulation for MC is:

$$\min_{\boldsymbol{M}} \mathrm{rank}(\boldsymbol{M}), \text{ s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega \tag{1}$$

where $\boldsymbol{M}$ is the recovered matrix. Nevertheless, (1) is a NP-hard optimization problem. Thereby, nuclear norm as the tightest convex relaxation of the rank function is suggested, resulting in

$$\min_{\boldsymbol{M}} \|\boldsymbol{M}\|_*, \text{ s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega \tag{2}$$

where the nuclear norm $\|\boldsymbol{M}\|_*$ is the sum of singular values of $\boldsymbol{M}$. However, it still involves full SVD at each iteration, which means high computational requirement especially for large-size data. To circumvent this issue, matrix factorization [12] is proposed, leading to

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega\|_F^2 \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$ are two small size matrices with rank $r \ll \min(m,n)$. Apparently, the restored matrix $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}$ also has low rank. Nevertheless, its performance is vulnerable to outliers as the ℓ_2-norm is adopted.

To achieve outlier resistance, other loss functions such as Huber function [2] and Welsch function [33] have been exploited. The corresponding RMC problem can be formulated as:

$$\min_{\boldsymbol{U},\boldsymbol{V}} g(\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega) \tag{4}$$

where $g(\cdot)$ stands for a robust loss function. Apparently, the ℓ_2-norm is sensitive to outliers because it assigns the same weights to all data including outlier-corrupted data. Compared to the ℓ_2-norm, the Huber and Welsch functions assign small weights to outlier-contaminated data. However, the Welsch function also changes the weights of normal data, while the Huber function may degrade for disturbances with larger magnitudes since it adopts the ℓ_1-norm to combat outliers. Table I and Fig. 1 show their weight functions and the corresponding curves. Here we assume that normal data have magnitudes less than c. It is seen that the weight of the Huber function remains unchanged when $|x| \le c$, while the Welsch function has reduced weights for all values of x except 0. In addition, the Welsch function assigns smaller weights to outlier-contaminated data compared to the Huber function. Nevertheless, our HOW function combines the advantages of Huber and Welsch functions, that is, it does not change the weights of normal data and is not sensitive to gross errors.

2) Proximity Operator: Consider the following penalized least squares problem:

$$\min_{y} \frac{1}{2}(x-y)^2 + \varphi_\lambda(y) \tag{5}$$

where $\varphi_\lambda(y)$ is the regularizer (also known as the regularization term), which penalizes the first quadratic term. The solution to y is given by the proximity operator:

$$p_\varphi(x) := \arg\min_{y} \frac{1}{2}(x-y)^2 + \varphi_\lambda(y) \tag{6}$$

As an illustration, the ℓ_1-norm based penalized least squares problem is:

$$\min_{y} \frac{1}{2}(x-y)^2 + \lambda|y|_1 \tag{7}$$

whose solution is:

$$y = p_\varphi(x) := \max\{0, |x| - \lambda\}\mathrm{sign}(x) \tag{8}$$

where $\lambda \ge 0$ is a thresholding parameter, and (8) is called the proximity operator of $|\cdot|_1$, also known as the soft-thresholding operator. The ℓ_1-norm is able to make the solution to (7) sparse because $p_\varphi(x) = 0$ for $|x| \le \lambda$.

Apart from the ℓ_1-norm, four common penalty functions/regularizers are tabulated in Table II and Fig. 2 plots the corresponding $p_\varphi(x)$. When $\varphi_\lambda(\cdot)$ is chosen as the ℓ_0-'norm', its proximity operator is known as the hard shrinkage thresholding. As stated in [21], a good penalty function should make the solution possess the following three properties: (i) Unbiasedness: the solution to y in (5) is nearly unbiased when x is large to avoid unnecessary modeling bias; (ii) Sparsity: the solution is 0 when x is sufficiently small to reduce model complexity; and (iii) Continuity: the solution is continuous to facilitate a stable model. In sparse signal recovery, the bias refers to the estimation error of sparse components [22], [23], which is given by the gap between the identity function and the proximity operator. Besides, the properties that the five regularizers satisfy are indicated in Table II. As seen in Fig. 2, although the $\ell_{\frac{1}{2}}$-'norm' and $\ell_{\frac{2}{3}}$-'norm' can mitigate the bias shrinking problem of the ℓ_1-norm, their proximity operator is discontinuous at $x = 1$. Following the definition of sparsity in [21], the IR [27], [40] generated by the Welsch function is not able to provide a sparse solution because

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE II
DIFFERENT REGULARIZATION FUNCTIONS AND THEIR CORRESPONDING SOLUTIONS AND PROPERTIES

	ℓ_0	$\ell_{\frac{1}{2}}$	$\ell_{\frac{2}{3}}$	ℓ_1	IR (Welsch)
$\varphi_\lambda(y)$	$\lambda\|y\|_0$	$\lambda\|y\|^{\frac{1}{2}}$	$\lambda\|y\|^{\frac{2}{3}}$	$\lambda\|y\|$	
$p_\varphi(x)$	$\begin{cases} 0, & \|x\| < \sqrt{2\lambda} \\ \{0, x\}, & \|x\| = \sqrt{2\lambda} \\ x, & \|x\| > \sqrt{2\lambda} \end{cases}$	$\begin{cases} \frac{2}{3}x(1+\cos(\frac{2\pi - 2\phi(x)}{3})), & \\ & \|x\| > \xi_1(\lambda) \\ 0, & \|x\| \le \xi_1(\lambda) \end{cases}$	$\begin{cases} ((w+\sqrt{2x/w - w^2})/2)^3, & \\ & \|x\| > \xi_2(\lambda) \\ 0, & \|x\| \le \xi_2(\lambda) \end{cases}$	$\begin{cases} x+\lambda, & x < -\lambda \\ 0, & \|x\| \le \lambda \\ x-\lambda, & x > \lambda \end{cases}$	$x - x \cdot \exp(-\frac{x^2}{\sigma^2})$
Unbiasedness	✓	✗	✗	✗	✓
Sparsity	✓	✓	✓	✓	✗
Continuity	✗	✗	✗	✓	✓

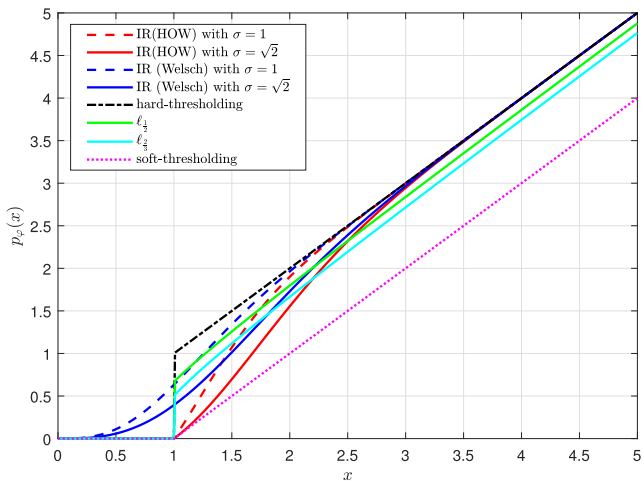

Fig. 2. $p_\varphi(x)$ for different regularizers.

Table II shows that its proximity operator $p_\varphi(x) = 0$ if and only if $x = 0$.

III. HYBRID ORDINARY-WELSCH FUNCTION

In this section, we devise a novel loss function, and propose a new regularizer. Importantly, the regularizer concurrently satisfies the desirable properties of unbiasedness, sparsity and continuity.

The expression of our designed HOW is:

$$l_{c,\sigma}(x) = \begin{cases} x^2/2, & |x| \le c \\ \frac{\sigma^2}{2}\left(1 - e^{\frac{c^2 - x^2}{\sigma^2}}\right) + \frac{c^2}{2}, & |x| > c \end{cases} \tag{9}$$

where $c \ge 0$ is a constant and σ is the kernel size. It is easy to find that when $c \to \infty$, $l_{c,\sigma}(x)$ is a quadratic function or ℓ_2-norm, and when $c = 0$, $l_{c,\sigma}(x)$ is the Welsch function. While for $0 < c < \infty$, it is a *redescending* M-estimator, whose analysis is given in Appendix A. Since $l_{c,\sigma}(x)$ is nonconvex, the Legendre-Fenchel transform is exploited. Given a function $f(x)$, its conjugate $f^*(y)$ is [42], [43]:

$$f^*(y) = \sup_x x \cdot y - f(x) \tag{10}$$

Similarly, the conjugate of $f^*(y)$ is:

$$(f^*(x))^* = \sup_y x \cdot y - f^*(y) \tag{11}$$

If $f(x)$ is a convex function, then $f(x) = (f^*(x))^*$. To obtain a convex $f(x)$, we define:

$$\begin{aligned} f(x) &= \frac{x^2}{2} - l_{c,\sigma}(x) \\ &= \begin{cases} 0, & |x| \le c \\ \frac{x^2}{2} - \frac{\sigma^2}{2}\left(1 - e^{\frac{c^2 - x^2}{\sigma^2}}\right) - \frac{c^2}{2}, & |x| > c \end{cases} \end{aligned} \tag{12}$$

Thus, we have:

$$\begin{aligned} f^*(y) &= \sup_x x \cdot y - \frac{x^2}{2} + l_{c,\sigma}(x) \\ &= \sup_x -\frac{(y-x)^2}{2} + l_{c,\sigma}(x) + \frac{y^2}{2} \\ &= \varphi(y) + \frac{y^2}{2} \end{aligned} \tag{13}$$

where

$$\varphi(y) = \sup_x -\frac{(y-x)^2}{2} + l_{c,\sigma}(x) \tag{14}$$

Since $f(x)$ is convex (See Appendix B), it is easy to get:

$$\begin{aligned} f(x) = (f^*(x))^* &= \sup_y y \cdot x - f^*(y) \\ &= \sup_y -\frac{(y-x)^2}{2} - \varphi(y) + \frac{x^2}{2} \end{aligned} \tag{15}$$

Combining (12) and (15) yields:

$$l_{c,\sigma}(x) = \inf_y \frac{(y-x)^2}{2} + \varphi(y) \tag{16}$$

where $\varphi(y)$ is the IR of HOW. In general, the exact expression of $\varphi(y)$ is unknown. As an illustration, we compute it numerically for several values of σ and plot the results in Fig. 3. In addition, the impact of c on $\varphi(y)$ is investigated in Fig. 4. There are 4 properties of $\varphi(y)$ as follows (See Appendix C):

1) $\varphi(y)$ is symmetric, i.e., $\varphi(y) = \varphi(-y)$.
2) $\varphi(y)$ is continuous, but nonsmooth at $y = 0$.
3) $\varphi(y)$ is nonnegative and increasing when $y > 0$.
4) If $\sigma \le \sqrt{2}c$, $\varphi(y)$ satisfies the triangle inequality, i.e., $\varphi(y_1 + y_2) \le \varphi(y_1) + \varphi(y_2)$ for any $y_1, y_2 \in \mathbb{R}$.

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

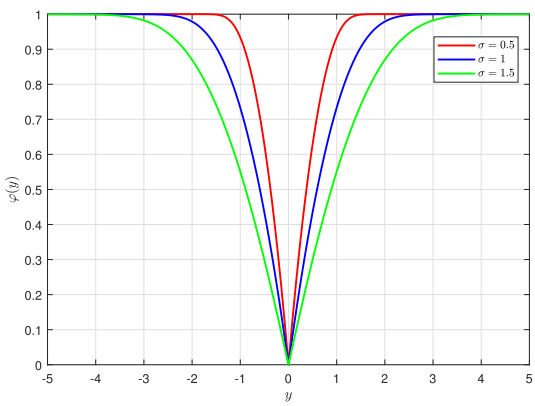

Fig. 3. Numerical curve of $\varphi(y)/\max(|\varphi(y)|)$ with $c = 1$.

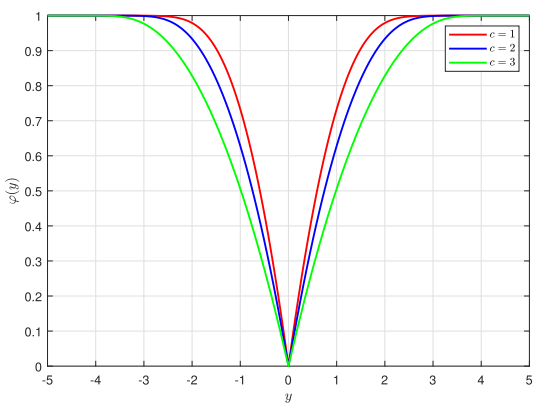

Fig. 4. Numerical curve of $\varphi(y)/\max(|\varphi(y)|)$ with $\sigma = 1$.

The proximity operator of (16) is given in the following proposition.

Proposition 1: The function $g(y) = \frac{(y-x)^2}{2} + \varphi(y)$ in (16) is convex, and the solution to y is (See Appendix D):

$$p_\varphi(x) := \max\{0, |x| - |x| \cdot e^{(c^2-x^2)/\sigma^2}\}\text{sign}(x) \tag{17}$$

In addition, $p_\varphi(x)$ is monotonic, namely, for any $x_1 \leq x_2$, $p_\varphi(x_1) \leq p_\varphi(x_2)$.

Moreover, the relation between c and σ in (9) is provided in Proposition 2, whose proof is given in Appendix E.

Proposition 2: The parameters c and σ control the robustness of $l_{c,\sigma}(x)$ and to decrease the bias introduced by the IR of HOW, they satisfy $c \geq \frac{\sigma}{\sqrt{2}}$.

It is worth mentioning that compared with the regularizers shown in Table II, the IR generated by HOW makes the solution sparse, continuous and approximately unbiased. The detailed analysis is found in Appendix F.

IV. Algorithms for Matrix Recovery

In this section, the HOW function is applied to LRMR in the presence of outliers. Adopting matrix factorization, our formulation is:

$$\min_{\boldsymbol{U},\boldsymbol{V}} l_{c,\sigma}\left(\boldsymbol{X}_\Omega - (\boldsymbol{UV})_\Omega\right) \tag{18}$$

where $l_{c,\sigma}(\boldsymbol{X}_\Omega - (\boldsymbol{UV})_\Omega)$ is separable, i.e., $l_{c,\sigma}(\boldsymbol{X}_\Omega - (\boldsymbol{UV})_\Omega) = \sum_{i,j\in\Omega} l_{c,\sigma}(\boldsymbol{X}_{i,j} - (\boldsymbol{UV})_{i,j})$. According to (16), we have:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \sum_{i,j\in\Omega} \frac{1}{2}\left(\boldsymbol{X}_{i,j} - (\boldsymbol{UV})_{i,j} - \boldsymbol{S}_{i,j}\right)^2 + \varphi(\boldsymbol{S}_{i,j}) \tag{19}$$

such that $\varphi(\boldsymbol{S}) = \sum_{i,j} \varphi(\boldsymbol{S}_{i,j})$, and $\boldsymbol{S}_{\Omega^c} = \boldsymbol{0}$. Then, (19) can be rewritten as:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \frac{1}{2}\left\|\boldsymbol{X}_\Omega - (\boldsymbol{UV})_\Omega - \boldsymbol{S}_\Omega\right\|_F^2 + \varphi(\boldsymbol{S}_\Omega) \tag{20}$$

It is easily seen that if $\boldsymbol{S}_\Omega$ is fixed, (20) becomes a quadratic optimization problem, implying that traditional optimization algorithms under Gaussian noise can be utilized. In our study, scaled alternating steepest descent (SASD) [44] is adopted to find $\boldsymbol{U}$ and $\boldsymbol{V}$ in (20). We first compute the solution to $\boldsymbol{S}$. In the $(k+1)$th iteration, given $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$, (20) is equal to:

$$\min_{\boldsymbol{S}} \frac{1}{2}\left\|\boldsymbol{D}_\Omega^k - \boldsymbol{S}_\Omega\right\|_F^2 + \varphi(\boldsymbol{S}_\Omega) \tag{21}$$

where $\boldsymbol{D}^k = \boldsymbol{X} - \boldsymbol{U}^k\boldsymbol{V}^k$. Since $\varphi(\boldsymbol{S})$ is separable, according to Proposition 1, the solution to (21) is:

$$\boldsymbol{S}_\Omega^{k+1} = p_\varphi(\boldsymbol{D}_\Omega^k) \tag{22}$$

Given $\boldsymbol{S}^{k+1}$, (20) amounts to:

$$\min_{\boldsymbol{U},\boldsymbol{V}} h(\boldsymbol{U},\boldsymbol{V}) := \frac{1}{2}\left\|\boldsymbol{H}_\Omega^{k+1} - (\boldsymbol{UV})_\Omega\right\|_F^2 \tag{23}$$

where $\boldsymbol{H}_\Omega^{k+1} = \boldsymbol{X}_\Omega - \boldsymbol{S}_\Omega^{k+1}$, and it can be efficiently solved by the SASD.

For a fixed $\boldsymbol{V}$, the gradient of $h(\boldsymbol{U},\boldsymbol{V})$ with respect to (w.r.t.) $\boldsymbol{U}$ is:

$$\nabla h_{\boldsymbol{V}}(\boldsymbol{U}) = -\left(\boldsymbol{H}_\Omega - (\boldsymbol{UV})_\Omega\right)\boldsymbol{V}^T \tag{24}$$

The corresponding step size $\mu_{\boldsymbol{U}}$ is calculated as:

$$\begin{aligned}\mu_{\boldsymbol{U}} :&= \arg\min_{\mu} \frac{1}{2}\left\|\boldsymbol{H}_\Omega - \left(\left(\boldsymbol{U} - \mu\nabla h_{\boldsymbol{V}}(\boldsymbol{U})\right)\boldsymbol{V}\right)_\Omega\right\|_F^2 \\ &= \|\nabla h_{\boldsymbol{V}}(\boldsymbol{U})\|_F^2 \,/\, \|(\nabla h_{\boldsymbol{V}}(\boldsymbol{U})\boldsymbol{V})_\Omega\|_F^2\end{aligned} \tag{25}$$

and we have:

$$\boldsymbol{U}^{k+1} = \boldsymbol{U}^k - \mu_{\boldsymbol{U}}^k \nabla h_{\boldsymbol{V}^k}(\boldsymbol{U}^k) \tag{26}$$

Similarly, given $\boldsymbol{U}$ and we can get:

$$\nabla h_{\boldsymbol{U}}(\boldsymbol{V}) = -\boldsymbol{U}^T\left(\boldsymbol{H}_\Omega - (\boldsymbol{UV})_\Omega\right) \tag{27a}$$

$$\mu_{\boldsymbol{V}} = \|\nabla h_{\boldsymbol{U}}(\boldsymbol{V})\|_F^2 \,/\, \|(\boldsymbol{U}\nabla h_{\boldsymbol{U}}(\boldsymbol{V}))_\Omega\|_F^2 \tag{27b}$$

and

$$\boldsymbol{V}^{k+1} = \boldsymbol{V}^k - \mu_{\boldsymbol{V}}^k \nabla h_{\boldsymbol{U}^{k+1}}(\boldsymbol{V}^k) \tag{28}$$

The steps from (24) to (28) are referred to as the alternating steepest descent (ASD) method. The accelerated version of ASD, SASD, obtained by scaling the gradient descent directions $\nabla h_{\boldsymbol{V}}(\boldsymbol{U})$ and $\nabla h_{\boldsymbol{U}}(\boldsymbol{V})$ with $(\boldsymbol{VV}^T)^{-1}$ and $(\boldsymbol{U}^T\boldsymbol{U})^{-1}$, respectively, has been verified to yield better performance than

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

ASD [44]. The gradient descent directions of SASD for our case are then:

$$\widetilde{\nabla} h_{\boldsymbol{V}}(\boldsymbol{U}) = (\boldsymbol{H}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega)\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{V}^T)^{-1} \tag{29a}$$

$$\widetilde{\nabla} h_{\boldsymbol{U}}(\boldsymbol{V}) = (\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T(\boldsymbol{H}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega) \tag{29b}$$

and the corresponding step sizes are:

$$\widetilde{\mu}_{\boldsymbol{U}} = \left\langle \nabla h_{\boldsymbol{V}}(\boldsymbol{U}), \widetilde{\nabla} h_{\boldsymbol{V}}(\boldsymbol{U}) \right\rangle \Big/ \left\| \left(\widetilde{\nabla} h_{\boldsymbol{V}}(\boldsymbol{U})\boldsymbol{V} \right)_\Omega \right\|_F^2 \tag{30a}$$

$$\widetilde{\mu}_{\boldsymbol{V}} = \left\langle \nabla h_{\boldsymbol{U}}(\boldsymbol{V}), \widetilde{\nabla} h_{\boldsymbol{U}}(\boldsymbol{V}) \right\rangle \Big/ \left\| \left(\boldsymbol{U}\widetilde{\nabla} h_{\boldsymbol{U}}(\boldsymbol{V}) \right)_\Omega \right\|_F^2 \tag{30b}$$

Finally, the SASD updates for $\boldsymbol{U}$ and $\boldsymbol{V}$ are:

$$\boldsymbol{U}^{k+1} = \boldsymbol{U}^k - \widetilde{\mu}_{\boldsymbol{U}}^k \widetilde{\nabla} h_{\boldsymbol{V}^k}(\boldsymbol{U}^k) \tag{31a}$$

$$\boldsymbol{V}^{k+1} = \boldsymbol{V}^k - \widetilde{\mu}_{\boldsymbol{V}}^k \widetilde{\nabla} h_{\boldsymbol{U}^{k+1}}(\boldsymbol{V}^k) \tag{31b}$$

Since c and σ control the robustness of (18), motivated by [33], [34], their values are set as:

$$\begin{aligned} c^k &= \min\left\{\xi_1\, d^k, c^{k-1}\right\} \\ \sigma^k &= \min\left\{\xi_2\, d^k, \sigma^{k-1}\right\} \end{aligned} \tag{32}$$

where $\xi_1 > 0$ and $\xi_2 > 0$ are constants, d^k is the robust normalized interquartile range of the vectorized $\boldsymbol{D}_\Omega$, which is computed as:

$$d^k = \mathrm{IQR}\left(\mathrm{vec}(\boldsymbol{D}_\Omega^k)\right)/1.349 \tag{33}$$

with $\mathrm{IQR}(\cdot)$ being the sample interquartile range operator [39]. Besides, in order to achieve stable recovery results, the solution to (3) is first computed to provide good initialization for $\boldsymbol{U}$ and $\boldsymbol{V}$, which is a similar strategy employed in [33]. In other words, (31a) and (31b) are first performed for a few iterations and then the optimization is switched to the RMC procedure with $\boldsymbol{S}$ being updated together.

The proposed RMC procedure is summarized in Algorithm 1. Defining $E_k = l_{c,\sigma}(\boldsymbol{X}_\Omega - (\boldsymbol{U}^k\boldsymbol{V}^k)_\Omega)$ and the relative error $rel_E^k = (E_k - E_{k-1})/E_{k-1}$, and if rel_E^k is less than a threshold, we say that the solutions satisfy the convergence condition. Besides, we define $\mathcal{C}_{c,\sigma}(\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}) = \frac{1}{2}\|\boldsymbol{X}_\Omega - (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{S}_\Omega\|_F^2 + \varphi(\boldsymbol{S}_\Omega)$, and the convergence of Algorithm 1 is analyzed in the following two theorems, whose proofs can be found in Appendix G and the supplementary material, respectively.

Theorem 1: The sequence $\{\mathcal{C}_{c^k,\sigma^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ generated by Algorithm 1 converges.

Theorem 2: Let $\{(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ be the sequence generated by Algorithm 1, and suppose that $(\boldsymbol{U}^k,\boldsymbol{V}^k)$ are of full rank. Given a bounded initialization, the generated $\{(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ is bounded and let $\{(\boldsymbol{U}^{k_j},\boldsymbol{V}^{k_j},\boldsymbol{S}^{k_j})\}$ be a subsequence of $\{(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ such that $\lim_{k_j\to\infty}\boldsymbol{U}^{k_j} = \boldsymbol{U}^\star$, $\lim_{k_j\to\infty}\boldsymbol{V}^{k_j} = \boldsymbol{V}^\star$ and $\lim_{k_j\to\infty}\boldsymbol{S}^{k_j} = \boldsymbol{S}^\star$. Then, $(\boldsymbol{U}^\star,\boldsymbol{V}^\star,\boldsymbol{S}^\star)$ is a critical point of (20).

The complexity of Algorithm 1 is dominated by the SASD, whose complexity is $\mathcal{O}(8|\Omega|r + 4(m+n)r^2)$ at each iteration. When we take the computation of $\boldsymbol{S}_\Omega$ into consideration, the total complexity becomes $\mathcal{O}(|\Omega|(8r+1) + 4(m+n)r^2)$ per iteration.

Algorithm 1: Robust Matrix Completion via HOW (RMC-HOW).

Input: Incomplete matrix $\boldsymbol{X}_\Omega$, rank r, index set Ω, ξ_1, ξ_2, maximum allowable iteration numbers I_1 and I_2, and tolerance parameters ζ_1 and ζ_2

Initialize: Generate a standard Gaussian matrix $\boldsymbol{V}^0 \in \mathbb{R}^{r\times n}$ and set $\boldsymbol{S} = \boldsymbol{0}$.

for $k = 1, 2, \ldots, I_1$ **do**
- Find the solution to $\boldsymbol{U}^k$ according to (31a)
- Find the solution to $\boldsymbol{V}^k$ according to (31b)
- Stop, if $rel_E^k \le \zeta_1$.

end for

Set $\boldsymbol{U}^0 = \boldsymbol{U}^k$ and $\boldsymbol{V}^0 = \boldsymbol{V}^k$

for $k = 1, 2, \ldots, I_2$ **do**
- Calculate c^k and σ^k according to (32)
- Find the solution to $\boldsymbol{S}^k$ according to (22)
- Find the solution to $\boldsymbol{U}^k$ according to (31a)
- Find the solution to $\boldsymbol{V}^k$ according to (31b)
- Stop, if $rel_E^k \le \zeta_2$.

end for

Output: $\boldsymbol{M} = \boldsymbol{U}^k\boldsymbol{V}^k$ and $\boldsymbol{S} = \boldsymbol{S}^k$.

Remark 1: Although Algorithm 1 is designed for RMC, it is clear that RPCA is the special case of RMC when all entries are observed. We refer the corresponding algorithm to as robust principal component analysis via HOW (RPCA-HOW).

Remark 2: Since the Welsch function is a special case of HOW when $c = 0$, Algorithm 1 can also realize the Welsch-based RMC and RPCA, referred to as RMC-Welsch and RPCA-Welsch, respectively.

V. Experimental Results

In this section, we perform experiments on RMC and RPCA, and all simulations are run on a computer with 3.0 GHz CPU and 16 GB memory. The recommended setting of parameters for the competing algorithms is adopted. If there are no parameter suggestions, we determine their appropriate values empirically. In Algorithm 1, we set $I_1 = 10$, $I_2 = 100$, $\zeta_1 = 10^{-3}$ and $\zeta_2 = 10^{-4}$.

A. Robust Matrix Completion

We compare our algorithm with the state-of-the-art RMC methods, namely, HQ-ASD [33], RegL_1 [19], $(\boldsymbol{S}+\boldsymbol{L})_{1/2}$ [30], $(\boldsymbol{S}+\boldsymbol{L})_{2/3}$ [31] and RMF-MM [18]. Besides, RMC-Welsch is also used to compare to the proposed method, which is a special case of RMC-HOW when $c = 0$. For the HQ-ASD, RMF-MM and RMC-HOW, which are monotonically decreasing algorithms, we stop them when the relative difference in the objective function is less than 10^{-4}, while for the ADMM-based methods, RegL_1, $(\boldsymbol{S}+\boldsymbol{L})_{1/2}$ and $(\boldsymbol{S}+\boldsymbol{L})_{2/3}$, we set their tolerance parameter to be 10^{-4}. Moreover, similar to RMC-HOW and RMC-Welsch, for the correntropy-based RMC algorithm, namely, HQ-ASD, its kernel size is calculated via (32).

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

1) Synthetic Data: The experimental strategy in [28], [33] is employed. We first generate the data matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}$, where the entries of $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$ satisfy the standard Gaussian distribution. Impulsive noise produced by Gaussian mixture model (GMM) is added into $\boldsymbol{X}$, and the positions of both impulsive noise and missing data are randomly chosen. The probability density function of GMM [28] is:

$$p_v(v) = \frac{1-\tau}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{v^2}{2\sigma_1^2}\right) + \frac{\tau}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{v^2}{2\sigma_2^2}\right) \quad (34)$$

where σ_1^2 and σ_2^2 are variances with $\sigma_1^2 \ll \sigma_2^2$, and τ controls the proportion of outliers. In our experiments, to model outliers, we set $\sigma_2^2 = 100\sigma_1^2$ and $\tau = 0.1$. The signal-to-noise ratio (SNR) of the impulsive noise is defined as [28]:

$$\mathrm{SNR} = \frac{\|\boldsymbol{X}_\Omega\|_F^2}{|\Omega|\left((1-\tau)\sigma_1^2 + \tau\sigma_2^2\right)} \quad (35)$$

Besides, the root mean square error (RMSE) is utilized to measure the performance of all algorithms:

$$\mathrm{RMSE} = \frac{\|\boldsymbol{X} - \boldsymbol{M}\|_F}{\sqrt{mn}} \quad (36)$$

We conduct experiments on data matrices with dimensions 300×200 and 1200×800. Except for matrix dimensions, there are two variables, i.e., SNR of impulsive noise and missing ratio. At each experiment, only one variable changes and the others keep fixed. The results based on 100 independent runs are tabulated in Table III. Ave and Std denote the average and the standard derivation of RMSE, respectively. Runtime is measured in seconds. The minimum Ave, Std and Runtime are shown in boldface. We can observe that our algorithm outperforms the remaining methods in term of Ave, Std and Runtime. Prior to explaining the reason why the proposed algorithm has the minimum average RMSE, we need to know that the impulsive noise generated by GMM contains small magnitude Gaussian noise and big outliers. All methods use different loss functions: our algorithm adopts the devised HOW function, which can be considered as using ℓ_2-norm to handle Gaussian noise and employing Welsch function to resist outliers, compared with the competing methods, which utilize Welsch function, $\ell_{\frac{1}{2}}$-norm, $\ell_{\frac{2}{3}}$-norm and ℓ_1-norm. That is to say, the competing algorithms employs only one loss function to resist impulsive noise, and our algorithm is equivalent to employing both ℓ_2-norm and Welsch function to achieve robustness, thus RMC-HOW is superior to the remaining methods. In addition, we can explain this phenomenon from another point. It is known that ℓ_2-norm does not change the weights of data, which cannot be used to resist outliers, while the Welsch function, $\ell_{\frac{1}{2}}$-norm, $\ell_{\frac{2}{3}}$-norm and ℓ_1-norm change the weights of all data and assign smaller weights to data with impulsive noise. Thus, for the data corrupted by both Gaussian noise and gross errors, it is desired that the weights of 'normal' data contaminated by Gaussian noise do not change, while the weights of outlier-contaminated data are assigned smaller values, which also indicates the superiority of the proposed method. Moreover, the Legendre-Fenchel transform and SASD are adopted to seek the solution, resulting in minimum runtime for our algorithm.

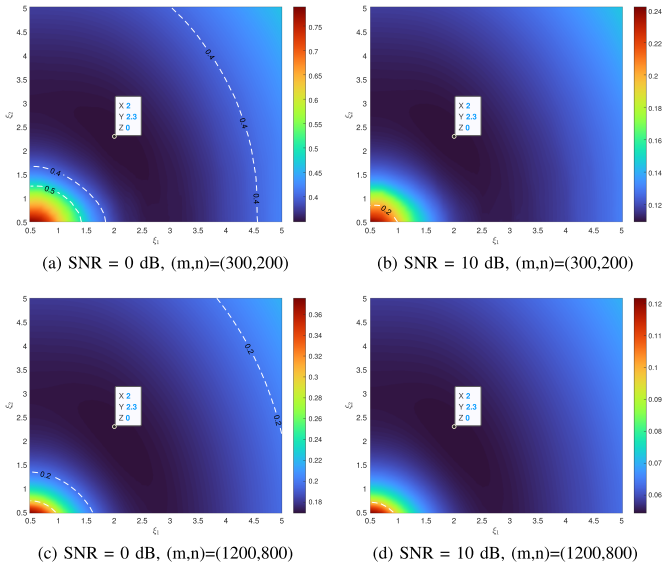

Fig. 5. RMSE versus ξ_1 and ξ_2.

Fig. 6. Original images from ZJU dataset.

For the proposed loss function, there are two parameters, i.e., c and σ, affecting its performance when resisting outliers. Thus, we analyze their sensitivity via conducting 100 independent runs. Fig. 5 depicts the RMSE map of the recovered matrix with different settings of $\{\xi_1, \xi_2\}$ according to (32). We see that for different SNRs and matrix dimensions, Algorithm 1 can work well for a wide range of $\{\xi_1, \xi_2\}$ and we set $\xi_1 = 2$ and $\xi_2 = 2.3$. When their values are too small, HOW will approach the Welsch M-estimator with very small kernel size, which causes the weight of normal data to drop rapidly, leading to a bad solution [33], [41].

2) Application to Image Inpainting: As mentioned in [18], [31], many natural images can be considered as low-rank matrices since their main information lies in a low-dimensional subspace. Thus, the proposed algorithm is employed to restore the degraded grayscale images, which are from the ZJU dataset[1] [1] and Rochester dataset[2] [46]. Fig. 6 shows several color images from the ZJU dataset where 10% pixels are randomly removed to generate the incomplete observations. In addition, as shown in Fig. 7, three other types of masks, including text mask, wash painting mask and block mask, are adopted to corrupt the images downloaded from Rochester dataset. To evaluate the robustness of all methods, Gaussian noise with variance 0.005 and salt-and-pepper noise, produced by the built-in Matlab function '$\mathrm{imnoise}(\boldsymbol{I}, \mathrm{salt}\,\&\,\mathrm{pepper}, \rho)$', where $\boldsymbol{I}$ is the incomplete matrix and ρ is the normalized noise intensity, are added into these images. The relationship between ρ and SNR is $\rho = 1/\mathrm{SNR}$ and $\mathrm{SNR} = 10\,\mathrm{dB}$. Moreover, to measure the recovery performance,

[1]https://sites.google.com/site/zjuyaohu/
[2]http://jiliu-ml.org/

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE COMPARISON ON SYNTHETIC MATRICES WITH DIFFERENT DIMENSIONS, SNRS AND MISSING RATIOS

$m = 300, n = 200$ $r = 10$		20% missing entries, SNR (dB) varies					SNR = 5 dB, % of missing entries varies				
		0	3	6	9	12	5%	10%	15%	20%	25%
RMC-HOW	Ave	**0.348**	**0.250**	**0.174**	**0.123**	**0.087**	**0.177**	**0.181**	**0.188**	**0.195**	**0.203**
	Std	0.0080	**0.0058**	**0.0040**	**0.0025**	**0.0020**	**0.0039**	**0.0040**	**0.0044**	**0.0048**	**0.0051**
	Runtime	**0.0328**	**0.0312**	**0.0298**	**0.0293**	**0.0294**	**0.0299**	**0.0310**	**0.0304**	**0.0308**	**0.0338**
RMC-Welsch	Ave	0.358	0.253	0.179	0.127	0.089	0.181	0.186	0.193	0.200	0.209
	Std	0.0081	0.0061	0.0041	0.0025	**0.0020**	**0.0039**	0.0041	0.0046	0.0050	0.0054
	Runtime	0.0412	0.0406	0.0400	0.0391	0.0395	0.0373	0.0393	0.0410	0.0406	0.0405
HQ-ASD	Ave	0.371	0.262	0.185	0.131	0.093	0.187	0.192	0.200	0.207	0.217
	Std	0.0085	0.0064	0.0042	0.0027	0.0021	**0.0039**	0.0043	0.0048	0.0052	0.0058
	Runtime	0.1129	0.1065	0.1012	0.0978	0.0955	0.0954	0.1024	0.1025	0.1032	0.1106
RegL_1	Ave	0.435	0.306	0.217	0.154	0.108	0.220	0.226	0.234	0.243	0.253
	Std	0.0097	0.0072	0.0052	0.0030	0.0023	0.0049	0.0047	0.0057	0.0059	0.0069
	Runtime	0.2058	0.2069	0.2029	0.2029	0.2042	0.2026	0.2130	0.2105	0.2048	0.2085
$S_{\frac{1}{2}}L_{\frac{1}{2}}$	Ave	0.571	0.401	0.285	0.201	0.142	0.293	0.301	0.308	0.327	0.319
	Std	0.0113	0.0113	0.0052	0.0042	0.0029	0.0067	0.0065	0.0066	0.0083	0.0078
	Runtime	0.7024	0.7111	0.6963	0.6980	0.7121	0.7225	0.7224	0.7192	0.7068	0.7247
$S_{\frac{2}{3}}L_{\frac{2}{3}}$	Ave	0.512	0.360	0.258	0.181	0.128	0.260	0.269	0.276	0.296	0.287
	Std	0.0098	0.0103	0.0058	0.0043	0.0027	0.0063	0.0060	0.0063	0.0079	0.0079
	Runtime	0.6255	0.6291	0.6163	0.6137	0.6248	0.6376	0.6346	0.6362	0.6358	0.6727
RMF-MM	Ave	0.440	0.309	0.220	0.154	0.109	0.220	0.229	0.235	0.254	0.245
	Std	**0.0078**	0.0090	0.0053	0.0038	0.0024	0.0054	0.0048	0.0054	0.0068	0.0067
	Runtime	28.422	27.106	32.444	32.657	30.663	21.567	33.993	24.796	26.580	27.579
$m = 1200, n = 800$ $r = 10$		20% missing entries, SNR (dB) varies					SNR = 5 dB, % of missing entries varies				
		0	3	6	9	12	5%	10%	15%	20%	25%
RMC-HOW	Ave	**0.171**	**0.121**	**0.086**	**0.061**	**0.043**	**0.088**	**0.091**	**0.094**	**0.096**	**0.099**
	Std	**0.00170**	**0.00125**	**0.00091**	**0.00069**	0.00054	**0.00091**	**0.00088**	**0.00097**	**0.00117**	**0.00110**
	Runtime	**0.425**	**0.411**	**0.403**	**0.401**	**0.414**	**0.427**	**0.422**	**0.420**	**0.402**	**0.446**
RMC-Welsch	Ave	0.175	0.123	0.087	0.062	0.044	0.090	0.092	0.095	0.098	0.101
	Std	0.00171	0.00126	0.00094	0.00073	0.00056	0.00093	0.00090	**0.00097**	0.00119	0.00112
	Runtime	0.581	0.541	0.537	0.526	0.526	0.569	0.558	0.555	0.527	0.535
HQ-ASD	Ave	0.178	0.126	0.089	0.063	0.045	0.092	0.094	0.097	0.100	0.103
	Std	0.0018	0.0013	0.00097	0.00075	0.00058	0.00095	0.00092	0.00100	0.00121	0.00113
	Runtime	1.312	1.262	1.220	1.196	1.207	1.305	1.283	1.264	1.213	1.228
RegL_1	Ave	0.214	0.151	0.107	0.076	0.054	0.110	0.113	0.117	0.120	0.124
	Std	0.0021	0.0016	0.0011	0.00088	0.00064	0.0011	0.0010	0.0012	0.0015	0.0014
	Runtime	4.089	4.083	3.992	3.986	4.171	4.119	4.114	4.140	3.955	4.079
$S_{\frac{1}{2}}L_{\frac{1}{2}}$	Ave	0.312	0.223	0.158	0.111	0.079	0.164	0.168	0.171	0.176	0.182
	Std	0.0038	0.0029	0.0019	0.0013	0.0008	0.0018	0.0019	0.0021	0.0023	0.0028
	Runtime	12.696	12.626	12.629	12.890	12.601	13.402	13.165	12.805	12.795	12.469
$S_{\frac{2}{3}}L_{\frac{2}{3}}$	Ave	0.264	0.188	0.133	0.094	0.067	0.137	0.141	0.144	0.149	0.154
	Std	0.0032	0.0024	0.0014	0.0011	0.0007	0.0014	0.0017	0.0018	0.0019	0.0023
	Runtime	11.688	11.769	11.797	11.893	11.760	12.385	12.051	11.869	11.747	11.392
RMF-MM	Ave	0.213	0.152	0.107	0.076	0.054	0.110	0.113	0.116	0.120	0.125
	Std	0.0027	0.0018	0.0013	0.0008	**0.00053**	0.0010	0.0015	0.0015	0.0015	0.0018
	Runtime	752.51	864.39	1981.1	2610.9	821.96	680.52	1018.6	822.58	865.37	1517.9

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE IV
IMAGE RESTORATION RESULTS FOR REAL IMAGES

Method		Image-01	Image-02	Image-03	Image-04	Image-05	Image-06	Image-07	Image-08	Image-09	Image-10	Image-11	Image-12
RMC-HOW	PSNR	**27.10**	**21.13**	**23.43**	**27.49**	**23.69**	**23.18**	**26.22**	**24.53**	**27.36**	**26.97**	**27.04**	**27.45**
	Runtime	**0.1436**	**0.1688**	**0.1649**	**0.1228**	**0.1530**	**0.1286**	**0.1638**	**0.1974**	**0.1786**	**0.1809**	**0.1738**	**0.1785**
RMC-Welsch	PSNR	26.66	20.93	23.29	27.33	23.53	22.94	26.09	24.46	27.19	26.81	26.90	27.31
	Runtime	0.1637	0.1827	0.1693	0.1524	0.1829	0.1389	0.1824	0.2215	0.1993	0.2082	0.2067	0.1850
HQ-ASD	PSNR	25.99	20.51	22.81	27.09	23.21	22.01	25.64	23.80	26.95	26.23	26.56	27.10
	Runtime	1.0589	1.1858	1.3388	0.4475	0.9284	1.0598	0.9723	2.2274	1.0964	0.9287	0.8180	1.1168
RegL_1	PSNR	25.94	20.87	22.80	26.06	22.93	22.67	25.09	23.91	26.04	25.71	25.79	26.32
	Runtime	0.3231	0.3148	0.30005	0.2977	0.2970	0.2897	0.2969	0.3163	0.2156	0.2210	0.2246	0.2070
$S_{\frac{1}{2}}L_{\frac{1}{2}}$	PSNR	24.44	19.94	21.99	24.94	22.10	21.38	24.13	23.02	24.86	24.67	24.72	25.09
	Runtime	1.2452	1.2977	1.2488	1.2131	1.2597	1.2184	1.2255	1.4923	1.4496	1.4583	1.4731	1.4597
$S_{\frac{2}{3}}L_{\frac{2}{3}}$	PSNR	25.03	20.32	22.41	25.50	22.51	21.96	24.60	23.45	25.39	25.20	25.25	25.66
	Runtime	1.0262	1.0782	1.0434	1.0097	1.0482	0.9969	1.0122	1.2321	1.2518	1.2273	1.2333	1.2381
RMF-MM	PSNR	25.92	20.85	22.76	26.03	22.83	22.67	24.92	23.79	26.00	25.58	25.78	25.65
	Runtime	85.531	85.733	94.559	77.744	98.999	70.574	97.494	112.13	115.53	121.38	111.85	108.46

Fig. 7. Images with different masks (from left to right: random mask, text mask, wash painting mask and block mask).

peak SNR (PSNR), is introduced, and we directly employ the built-in command 'psnr(recovered, original)' in MATLAB to calculate it. We set the rank of these images to 15 [18].

Table IV tabulates the image restoration results and Fig. 8 shows the recovered grayscale images by different algorithms for Image-09. We see that our algorithm achieves the best recovery, and is several times faster than HQ-ASD, $(\boldsymbol{S}+\boldsymbol{L})_{1/2}$ and $(\boldsymbol{S}+\boldsymbol{L})_{2/3}$, and more than 50 times faster than RMF-MM. Furthermore, the impact of percentage of observations on recovery performance is investigated. Fig. 9 plots the average PSNR versus the percentage of observations for Image-09. It is seen that our RMC-HOW and RMC-Welsch are superior to the competitors when the observed entries are no less than 50%.

3) Application to Hyperspectral Imaging: Hyperspectral imaging (HSI) has numerous applications such as earth climate, agriculture and urban planning [3], [47]. However, it may lose some data and be corrupted by Gaussian noise and impulsive noise, due to photon effects and calibration error. Thus, it is necessary to improve the HSI quality prior to its subsequent processing. In this section, we employ the proposed method to recover the degraded HSI in the presence of outliers. Indian Pines [3][3] dataset is used, whose dimensions are 145×145 pixels per band with 200 bands. Then, the data matrix $\boldsymbol{X} \in \mathbb{R}^{21025\times 200}$ is constructed of which columns comprise vectorized bands of the Indian Pines data. In addition, 20% of the pixels in $\boldsymbol{X}$ are randomly removed, $10\,\mathrm{dB}$ impulsive noise generated by GMM is added to the incomplete data. Fig. 10 displays the restoration

[3]http://lesun.weebly.com/hyperspectral-data-set.html

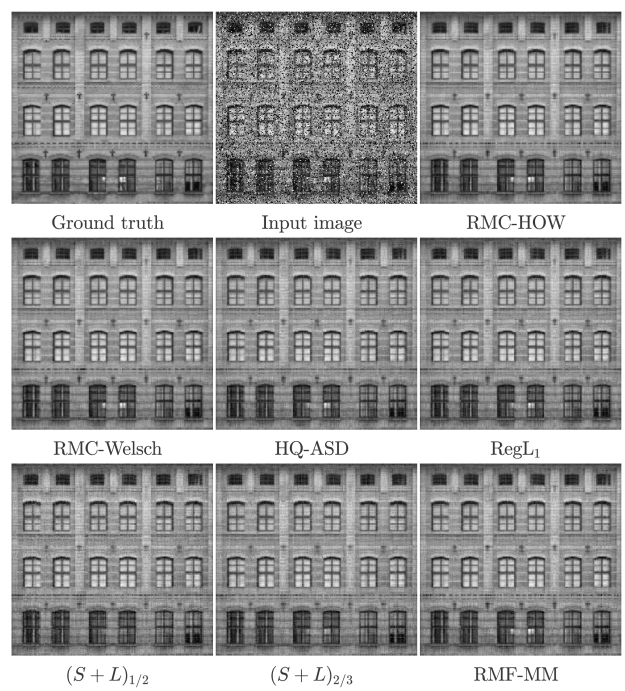

Fig. 8. Image inpainting results by different algorithms.

results of different methods. Comparing with the competing methods, the proposed algorithm achieves the highest PSNR and has the least computational time.

B. Robust Principal Component Analysis

In this section, extensive experiments are performed on synthetic data, shadow/specularity removal and video analysis. We compare the proposed method with several state-of-the-art RPCA algorithms, including IALM [48], SSGoDec [20] and NCRPCA [24]. Besides, RPCA-Welsch is employed as a competitor, which is a special case of the RPCA-HOW. Moreover, the parameters in the competing algorithms are adopted in accordance to their recommended settings.

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

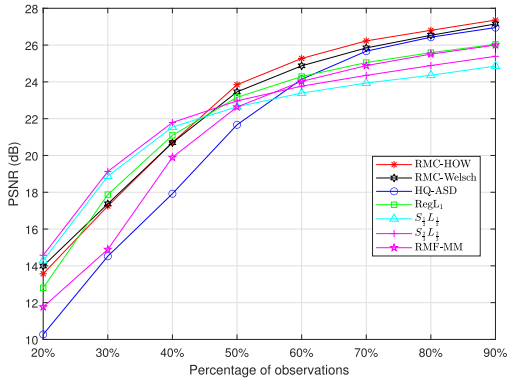

Fig. 9. PSNR versus the percentage of observations.

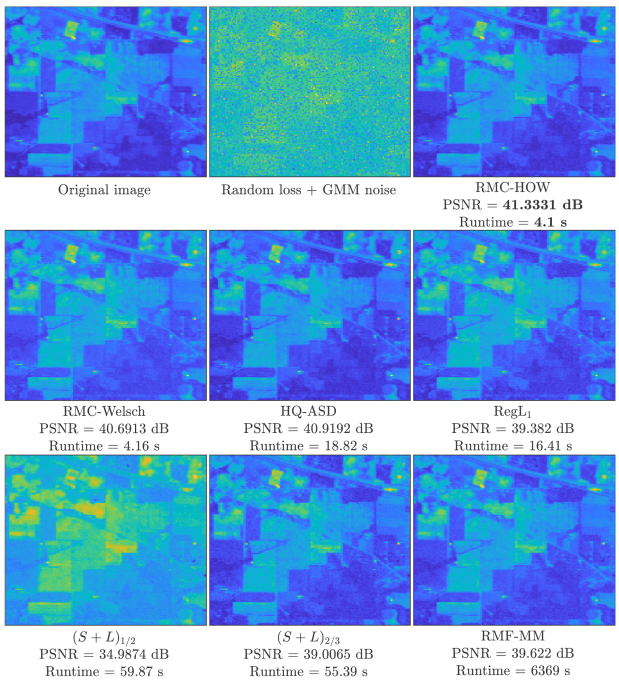

Fig. 10. Hyperspectral imaging restoration results for the 15th band.

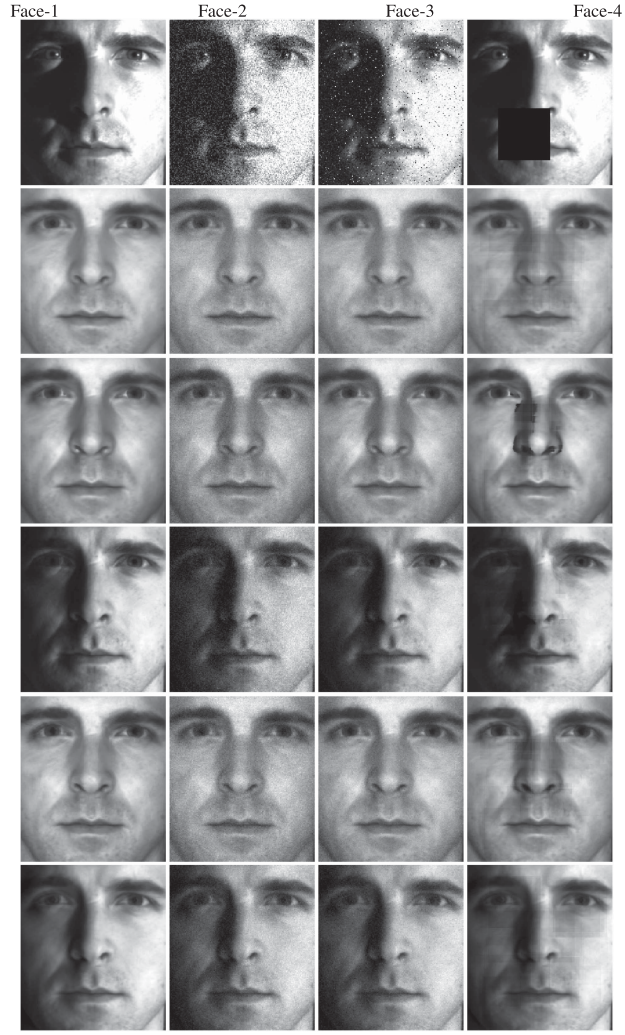

Fig. 11. Results of shadow and specularity removal. Face-1 to Face-4 are original images, corrupted images by 10 dB Gaussian noise, 10 dB GMM noise and occlusions, respectively. Images from the second row to the last are recovered images by RPCA-HOW, RPCA-Welsch, IALM, SSGoDec and NCRPCA, respectively.

1) Synthetic Data: The data matrix $\boldsymbol{X}$ is generated as $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S} + \boldsymbol{G}$, where $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$ with $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times m}$ whose entries are independently sampled from $\mathcal{N}(0,1)$ and $r = m/50$. $\boldsymbol{G}$ is the standard Gaussian noise matrix whose entries are sampled from $\mathcal{N}(0, 0.1)$. $\boldsymbol{S}$ is a sparse matrix and 1 percent of its elements are drawn from the uniform distribution with interval $[-50, 50]$. To evaluate the performance of different algorithms, the metrics, including the relative recovery errors related to $\boldsymbol{L}$ and $\boldsymbol{S}$, and the accuracy of capturing the sparse locations, are employed:

$$\mathrm{ERR_L} = \|\boldsymbol{L} - \hat{\boldsymbol{L}}\|_F / \|\boldsymbol{L}\|_F$$

$$\mathrm{ERR_S} = \|\boldsymbol{S} - \hat{\boldsymbol{S}}\|_F / \|\boldsymbol{L}\|_F$$

$$\mathrm{F_S} = \#\left\{\left(\hat{\boldsymbol{S}} \neq \boldsymbol{0}\right) \neq (\boldsymbol{S} \neq \boldsymbol{0})\right\} / m^2$$

where $\hat{\boldsymbol{L}}$ and $\hat{\boldsymbol{S}}$ are the estimates of $\boldsymbol{L}$ and $\boldsymbol{S}$, respectively. In the experiments, m varies in the range $\{500, 1000, 1500\}$ and the average results conducted via 100 independent runs are tabulated in Table V. It is observed that the RPCA-HOW outperforms the completing algorithms in terms of $\mathrm{ERR_L}$, $\mathrm{ERR_S}$, $\mathrm{F_S}$ and runtime. The proposed algorithm is almost 100 times faster than IALM and NCRPCA when $m = 1500$, because the latter methods require SVD calculation at each iteration.

2) Shadow and Specularity Removal From Faces: One interesting task of RPCA is to remove shadows and specularities from face images [4], [5]. Often times, the images from the same person are taken under varying illuminations, creating a challenge for face analysis. However, if we have enough images of the same person, RPCA can be used to decompose it into low-rank and sparse components, where face features are modeled as the low-rank components, while shadow and specularity are considered as the sparse components. We conduct experiments on the extended Yale-B dataset [49], where the dimensions of each image are 192×168 and there are a total of 64 illuminations. Each face image of the same subject

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE V
RPCA RESULTS USING SYNTHETIC DATA BY DIFFERENT ALGORITHMS

Method	$m = 500$				$m = 1000$				$m = 1500$			
	$\mathrm{ERR_L}$	$\mathrm{ERR_S}$	$\mathrm{F_S}$	Time	$\mathrm{ERR_L}$	$\mathrm{ERR_S}$	$\mathrm{F_S}$	Time	$\mathrm{ERR_L}$	$\mathrm{ERR_S}$	$\mathrm{F_S}$	Time
RPCA-HOW	$\mathbf{5.1 \pm 3.4}$	$\mathbf{0.75 \pm 1.5}$	**0.0025**	**0.213**	$\mathbf{3.2 \pm 0.06}$	$\mathbf{0.48 \pm 0.02}$	**0.0025**	**0.531**	$\mathbf{2.6 \pm 0.04}$	$\mathbf{0.47 \pm 0.01}$	**0.0025**	**1.173**
RPCA-Welsch	23.3 ± 8.7	8.95 ± 3.4	0.99	0.734	7.6 ± 3.5	4.4 ± 2.2	0.99	1.78	3.87 ± 1.6	2.7 ± 1.3	0.99	3.183
IALM	7.0 ± 0.25	2.0 ± 0.06	0.55	5.408	5.0 ± 0.11	2.1 ± 0.03	0.55	34.94	4.0 ± 0.06	2.2 ± 0.02	0.55	101.3
SSGoDec	7.5 ± 0.21	3.1 ± 0.08	0.99	0.558	5.4 ± 0.12	3.3 ± 0.04	0.99	2.350	4.3 ± 0.06	3.4 ± 0.03	0.99	5.153
NCRPCA	172.9 ± 4.8	7.7 ± 0.67	0.99	3.785	89.1 ± 3.2	7.3 ± 4.3	0.99	1.815	71.8 ± 0.04	21.9 ± 4.9	0.99	81.13

$\mathrm{ERR_L}$ *and* $\mathrm{ERR_S}$ *are scaled by* 10^{-3} *and* 10^{-2}, *respectively, and time is in seconds.*

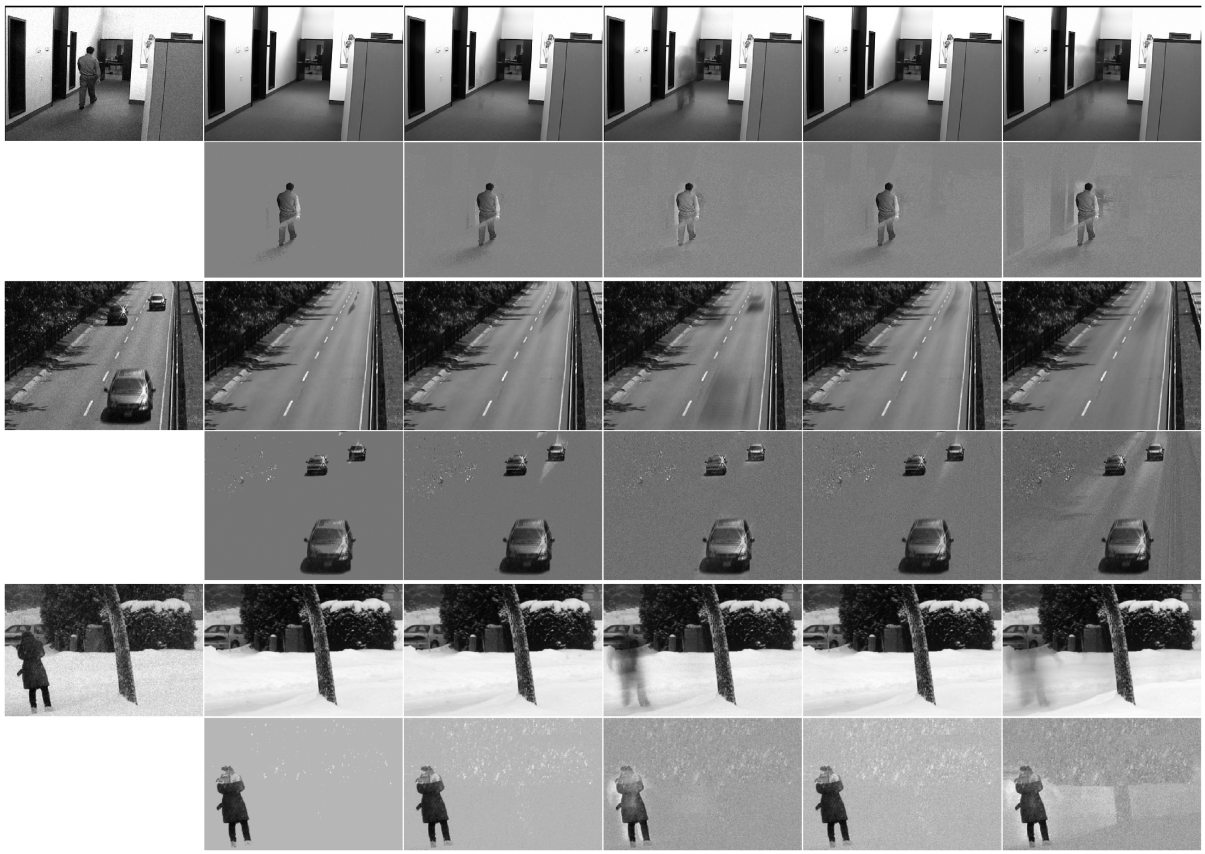

Fig. 12. Background and foreground separation results of different algorithms using real videos. Images from left to right are original images, recovered results by RPCA-HOW, RPCA-Welsch, IALM, SSGoDec and NCRPCA, respectively.

is vectorized to construct the data matrix $\boldsymbol{X} \in \mathbb{R}^{32256 \times 64}$. To compare the robustness of different algorithms, Gaussian noise, GMM noise and occlusions are added to $\boldsymbol{X}$. The results are shown in Fig. 11, and it is greatly recommended to enlarge the recovered faces to view their details. We denote the first row of Fig. 11 from left to right as Face-1, Face-2, Face-3 and Face-4, respectively. It is easy to see that the IALM and NCRPCA cannot remove the shadows well, compared with the remaining methods. In addition, the recovered images by our approach are better than those by RPCA-Welsch and SSGoDec, because there still exist some shadows in the noise part in the third row and fifth row. Moreover, RPCA-HOW is superior to RPCA-Welsch and SSGoDec when the face images are corrupted by Gaussian nosie. Table VI tabulates the average iteration number (Iter.) and runtime in seconds (Time) based on 20 independent runs, and it is observed that the RPCA-HOW is faster than the competing methods.

TABLE VI
RUNTIME COMPARISON BY DIFFERENT ALGORITHMS

Method	Face-1		Face-2		Face-3		Face-4	
	Iter.	Time	Iter.	Time	Iter.	Time	Iter.	Time
RPCA-HOW	**12.8**	**1.388**	**7.1**	**0.652**	**9.7**	**0.961**	**10.15**	**1.081**
RPCA-Welsch	25.05	2.756	14.55	1.605	21.15	2.355	30.05	3.375
IALM	247	17.786	189	13.193	211.6	14.665	229.3	16.271
SSGoDec	101	2.616	101	2.632	101	2.605	101	2.613
NCRPCA	59	4.853	100	8.431	100	8.991	59	4.926

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

TABLE VII
QUANTITATIVE RESULTS ON BACKGROUND AND FOREGROUND SEPARATION

Method	Cubicle				Highway				Skating			
	Iter.	Time	$\frac{\|\boldsymbol{S}\|_0}{mn}$	$\mathrm{F_m}$	Iter.	Time	$\frac{\|\boldsymbol{S}\|_0}{mn}$	$\mathrm{F_m}$	Iter.	Time	$\frac{\|\boldsymbol{S}\|_0}{mn}$	$\mathrm{F_m}$
RPCA-HOW	**9**	**6.89**	**0.050**	**0.7276**	**12**	**9.29**	**0.077**	**0.6191**	**8**	**3.35**	**0.060**	**0.6323**
RPCA-Welsch	13	10.7	1	0.7231	13	9.61	1	0.6031	12	5.68	1	0.6319
IALM	117	91.3	0.962	0.5372	126	97.3	0.933	0.6128	122	55.6	0.945	0.5791
SSGoDec	101	20.7	0.893	0.7255	101	19.0	0.922	0.6119	101	12.4	0.906	0.6313
NCRPCA	63	63.6	1	0.5838	100	88.1	1	0.5256	62	32.5	1	0.5794

$\boldsymbol{S}$ is *the foreground matrix.*

3) Background Modeling From Surveillance Video: We test all methods on background-foreground separation, which can be modeled as a RPCA problem since the background in videos and the moving objects in foreground can be considered as the low-rank and sparse components, respectively. Three videos from the CDnet dataset [50], namely, Cubicle (240×352 for each frame), Highway (240×320) and Skating (180×270), are employed. Besides, 200 frames for each video are chosen and then vectorized to generate the data matrix. We add zero-mean Gaussian noise with variance being 0.001 into the data matrix. For those videos, we assume that the background are approximately static, thus its rank is one. To numerically evaluate the performance, four metrics, namely, iteration number (Iter.), runtime in seconds, sparseness in foreground ($\|\boldsymbol{S}\|_0/mn$) and F-measure, are adopted. Given true positive (TP), false positive (FP), false negative (FN) and true negative (TN), F-measure is defined as:

$$\mathrm{F_m} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{37}$$

where $\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$ and $\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$.

The numerical results of different algorithms and the visual results of representative frames for the surveillance videos are shown in Table VII and Fig. 12, respectively. It is seen that obvious ghosting effect appears in the backgrounds obtained by IALM and NCRPCA in the first and third videos. Besides, in the second video, the shadows on the road are minimal in the background obtained by our method, compared to the remaining approaches. It is very easy to observe that RPCA-HOW obtains a more sparse foreground since the foreground generated by RPCA-Welsch, IALM, SSGoDec and NCRPCA still contain some apparent noise, which corresponds to the numerical results of $\|\boldsymbol{S}\|_0/mn$ in Table VII. Moreover, the results in Table VII demonstrate that RPCA-HOW outperforms the competing methods in terms of iteration number, running time, sparse foreground and F-measure.

VI. CONCLUSION

In this article, we propose a new M-estimator called hybrid ordinary-Welsch (HOW), and apply it to LRMR to combat outliers. Although HOW is nonconvex, the Legendre-Fenchel transform is employed to convert it into a sum of two convex problems and one of them is quadratic, thus traditional optimization methods can be applied. In addition, the IR of HOW is derived and the corresponding proximity operator satisfies three desirable properties, namely, unbiasedness, sparseness and continuity. We develop an efficient algorithm called RMC-HOW for the task of RMC, and three other algorithms named RPCA-HOW, RMC-Welsch and RPCA-Welsch, are the special cases of RMC-HOW. Moreover, we theoretically prove that any accumulation point of the iterates in the proposed algorithm is a critical point. Based on extensive numerical examples using synthetic and real-world datasets, our algorithm demonstrates superiority over the state-of-the-art robust methods in terms of restoration performance and computational time.

APPENDIX A

Proof: The following definitions are first introduced.

Definition 1 (M-estimator [39], [41]): The function $l(x)$ is called an M-estimator if it satisfies:

1) $l(x) \geq 0$;
2) $l(0) = 0$;
3) $l(x) = l(-x)$;
4) $l(x) \geq l(y)$ for $|x| \geq |y|$.

Definition 2 (Redescending M-estimator [39]): An M-estimator $l(x)$ is called redescending M-estimator if $\psi(x)$ where $\psi(x) = l'(x)$, is not monotone.

It is easy to check that $l_{c,\sigma}(x)$ is an M-estimator, and the derivative of $l_{c,\sigma}(x)$ is:

$$\psi(x) = \begin{cases} x, & |x| \leq c \\ x \cdot e^{\frac{c^2-x^2}{\sigma^2}}, & |x| > c \end{cases}$$

Besides, when $x > 0$, $\psi(x)$ increases with x first and then decreases with x, that is, $\psi(x)$ is not monotone, thus $l_{c,\sigma}(x)$ is a redescending M-estimator. ■

APPENDIX B

Proof: From the expression of $f(x)$, we know that $f(x)$ is a convex function if and only if $f(x)$ with $|x| > c$ is a convex function. Thus, when $|x| > c$,

$$f'(x) = x - x \cdot e^{\frac{c^2-x^2}{\sigma^2}}$$

and

$$f''(x) = 1 - \left(1 - \frac{2x^2}{\sigma^2}\right) e^{\frac{c^2-x^2}{\sigma^2}}$$

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

$$
\begin{aligned}
&= e^{\frac{c^2-x^2}{\sigma^2}}\left(e^{\frac{x^2-c^2}{\sigma^2}}+\frac{2x^2}{\sigma^2}-1\right)\\
&\geq e^{\frac{c^2-x^2}{\sigma^2}}\left(\frac{x^2-c^2}{\sigma^2}+1+\frac{2x^2}{\sigma^2}-1\right)\\
&= e^{\frac{c^2-x^2}{\sigma^2}}\left(\frac{3x^2-c^2}{\sigma^2}\right)\\
&> 0
\end{aligned} \tag{38}
$$

where the first inequality is obtained using $e^x > x+1$ for any $x \in \mathbb{R}$, and the last inequality is due to the prior condition $|x| > c$. Therefore, $f(x)$ is a convex function. ■

Appendix C

Proof: The proof is inspired by [32].

1) According to (14), we have:

$$
\begin{aligned}
\varphi(-y) &= \sup_{x\in\mathbb{R}} -\frac{(-y-x)^2}{2}+l_{c,\sigma}(x)\\
&\overset{t=-x}{=} \sup_{t\in\mathbb{R}} -\frac{(-y+t)^2}{2}+l_{c,\sigma}(-t)\\
&= \sup_{t\in\mathbb{R}} -\frac{(y-t)^2}{2}+l_{c,\sigma}(t)\\
&= \varphi(y)
\end{aligned} \tag{39}
$$

where the penultimate equation is obtained because $l_{c,\sigma}(x)$ is an even function. Thus, $\varphi(y)$ is symmetric.

2) According to the inversion rule of subgradient relations for the Legendre-Ferchel transform and (13), we obtain:

$$\arg\sup_x x\cdot y - f(x) = \partial f^*(y) = y + \partial\varphi(y) \tag{40}$$

First, we define $q(x) = x\cdot y - f(x)$, which is concave because $f(x)$ is convex, thus the solution $x^\star$ to (40) satisfies $\nabla q(x^\star) = 0$, and we have:

$$y = \nabla f(x^\star) = \begin{cases} 0, & |x^\star| \leq c\\ x^\star - x^\star\cdot e^{\frac{c^2-(x^\star)^2}{\sigma^2}}, & |x^\star| > c\end{cases} \tag{41}$$

That is to say, when $|y| > 0$, $x^\star \in (-\infty, -c)\cup(c,\infty)$, while when $|y| = 0$, $x^\star \in [-c, c]$. Besides, we can conclude that the solution $x^\star$ is unique when $|y| > 0$ because $f'(x)$ is monotonic for $|x| > c$ due to (38), while it is not unique when $y = 0$. Hence $\varphi(y)$ is differentiable except the zero point, and it has a set of subgradients at 0, i.e., $\partial\varphi(0) \in [-c, c]$. Thus, $\varphi(y)$ is nonsmooth at the zero point.

In addition, when $y = 0$, we conclude that $0 = \varphi(0) + 0$ according to (13). Thus, we have $\varphi(0) = 0$. For $|y| > 0$, i.e., $|x| > c$, as $|x| \to c$, $y \to 0$ and $\varphi(y) \to 0$ as well. Therefore, $\varphi(y)$ is continuous.

3) According to (41), when $y > 0$, the solution to $\arg\max_x y\cdot x - f(x)$ is unique, and it satisfies:

$$y = x^\star - x^\star\cdot e^{\frac{c^2-(x^\star)^2}{\sigma^2}} \tag{42}$$

implying that $y < x^\star$, and by (40) ($x^\star = y + \varphi'(y)$, note that we replace $\partial\varphi(y)$ with $\varphi'(y)$ because $\varphi(y)$ is differentiable for $|y| > 0$), we obtain $\varphi'(y) > 0$. Therefore, $\varphi(y)$ is increasing for $y > 0$ and $\varphi(y)$ is nonnegative because $\varphi(0) = 0$.

4) To verify $\varphi(y_1+y_2) \leq \varphi(y_1)+\varphi(y_2)$ for any $y_1, y_2 \in \mathbb{R}$, we divide it into three cases.

If $y_1\cdot y_2 = 0$ and we assume that $y_1 \neq 0$ and $y_2 = 0$, it is easy to check $\varphi(y_1+y_2) = \varphi(y_1) \leq \varphi(y_1)+\varphi(0) \leq \varphi(y_1)+\varphi(y_2)$ due to $\varphi(0) = 0$.

If $y_1\cdot y_2 > 0$ and we first suppose that $y_1 > 0$ and $y_2 > 0$. For any $y > 0$ (implying $x > c$), we have (42) and by (38), we draw a conclusion that y increases with $x^\star$. Combining $x^\star = y + \varphi'(y)$ and (42), we obtain $\varphi'(y) = x^\star\cdot e^{\frac{c^2-(x^\star)^2}{\sigma^2}}$. We define $r(x) = x\cdot e^{\frac{c^2-x^2}{\sigma^2}}$ and $r'(x) = (1-\frac{2x^2}{\sigma^2})\cdot e^{\frac{c^2-x^2}{\sigma^2}} < 0$ because $\frac{2x^2}{\sigma^2} > 1$ when $x > c$, namely, $\sigma \leq \sqrt{2}c$, which will be analyzed in Appendix E. Thus, we know that $\varphi'(y)$ is a decreasing function of y. Then, we obtain:

$$
\begin{aligned}
&\int_0^{y_2} \varphi'(y_1+y) - \varphi'(y)dy\\
&= \varphi(y_1+y_2) - \varphi(y_2) - \varphi(y_1) + \varphi(0)\\
&\leq 0
\end{aligned}
$$

Thus, $\varphi(y_1+y_2) \leq \varphi(y_1)+\varphi(y_2)$. On the other hand, if $y_1\cdot y_2 > 0$, $y_1 < 0$ and $y_2 < 0$, we employ (39) to yield:

$$
\begin{aligned}
\varphi(y_1+y_2) &= \varphi(-y_1-y_2)\\
&\leq \varphi(-y_1)+\varphi(-y_2)\\
&= \varphi(y_1)+\varphi(y_2)
\end{aligned}
$$

If $y_1\cdot y_2 < 0$ and assuming that $y_1 > 0$ and $y_2 < 0$, we can draw the conclusion immediately from $\varphi(y_1+y_2) \leq \varphi(y_1+|y_2|) \leq \varphi(y_1)+\varphi(|y_2|) = \varphi(y_1)+\varphi(y_2)$. ■

Appendix D
Proof of Proposition 1

Proof: Since $f^*(y)$ is convex in (13), we know that $yx - f^*(y) = yx - \varphi(y) - \frac{y^2}{2}$ and $-g(y) = -\frac{(y-x)^2}{2} - \varphi(y)$ in (15) w.r.t. y are concave, thus $g(y)$ is convex. In addition, the solution to (16) is equal to that to (15), and we have by the inversion rule for subgradient relations:

$$\partial f(x) = \arg\sup_y y\cdot x - f^*(y)$$

thus,

$$y = \nabla f(x) = \begin{cases} 0, & |x| \leq c\\ x - x\cdot e^{(c^2-x^2)/\sigma^2}, & |x| > c\end{cases}$$

Moreover, it is easy to check that $\nabla f(x)$ is an odd function and increases with x when $|x| > c$ according to (38), thus we conclude that $\nabla f(x)$ is monotonic, that is, $p_\varphi(x)$ is nondecreasing. ■

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

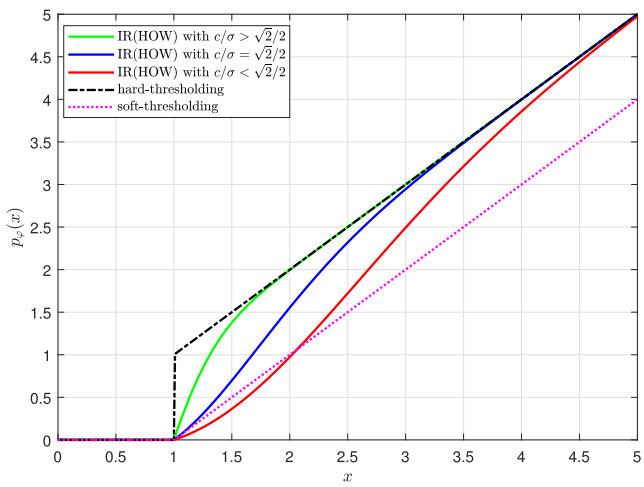

Fig. 13. Comparison of $p_\varphi(x)$ for different regularizers.

Appendix E
Proof of Proposition 2

Proof: The HOW function is a hybrid of quadratic and Welsch functions, and c and σ are used to control its robustness. The value of c determines whether an entry is contaminated by an outlier or not, while the kernel size σ is associated with the robustness of the Welsch function [33].

For convenience, only $x \geq 0$ is considered, and $\Delta b = x - p_\varphi(x)$ is used to measure the bias. As shown in Fig. 13, ℓ_0-'norm' as the regularizer is unbiased while ℓ_1-norm has a constant bias for $x > 1$, that is to say, $\Delta b = 0$ for the ℓ_0-'norm' and Δb is a constant for the ℓ_1-norm. To bridge the gap between them, it is necessary to ensure that Δb decreases with x ($x > c$, $c = 1$ in Fig. 13). Thus, by Proposition 1, we have:

$$\begin{aligned}\Delta b &= x - \left(x - x \cdot e^{\frac{c^2 - x^2}{\sigma^2}}\right) \\ &= x \cdot e^{\frac{c^2 - x^2}{\sigma^2}}\end{aligned} \tag{43}$$

and we need:

$$\frac{d(\Delta b)}{dx} = \left(1 - \frac{2x^2}{\sigma^2}\right) \cdot e^{\frac{c^2 - x^2}{\sigma^2}} < 0 \tag{44}$$

Therefore, for any $x > c$, we obtain:

$$1 - \left(2x^2\right)/\sigma^2 < 0 \tag{45a}$$

$$x > \frac{\sigma}{\sqrt{2}} \tag{45b}$$

Since $x > c$, we get $c \geq \frac{\sigma}{\sqrt{2}}$. This conclusion makes sure that $\varphi(y)$ satisfies the triangle inequality in Appendix C. It can be seen in Fig. 13 that when $c < \frac{\sigma}{\sqrt{2}}$, Δb increases first and then decreases with x, while when $c \geq \frac{\sigma}{\sqrt{2}}$, Δb decreases with x. This completes the proof. ■

Appendix F

Proof: We first rewrite the optimization problem:

$$\min_y \frac{1}{2}(x - y)^2 + \varphi_\lambda(y) \tag{46}$$

As stated in [21], a good penalty function makes the solution to (46) possess the following three properties: (i) Unbiasedness: the solution to y in (46) is nearly unbiased when x is large to avoid unnecessary modeling bias; (ii) Sparsity: the solution is 0 when x is sufficiently small to reduce model complexity; and (iii) Continuity: the solution is continuous to facilitate a stable model.

The solution to (46) is given by the proximal operator:

$$p_\varphi(x) := \arg\min_y \frac{1}{2}(x - y)^2 + \varphi_\lambda(y)$$

For the IR of HOW, its proximity operator is:

$$\begin{aligned}p_\varphi(x) &:= \max\left\{0, |x| - |x| \cdot e^{(c^2 - x^2)/\sigma^2}\right\} \operatorname{sign}(x) \\ &= \begin{cases} 0, & |x| \leq c \\ x - x \cdot e^{\frac{c^2 - (x)^2}{\sigma^2}}, & |x| > c \end{cases}\end{aligned} \tag{47}$$

We first check that the solution to (46) is sparse via (47) since $p_\varphi(x) = 0$ when $|x| \leq c$. Besides, the proximity operator is continuous at $x = c$ because $\lim_{x \to c^-} p_\varphi(x) = 0 = c - c = \lim_{x \to c^+} p_\varphi(x)$. To reduce the bias generated by regularizers, their proximity operators should approach the identity function ($y = x$) asymptotically. For example, the hard-thresholding is unbiased since it coincides with the identity function asymptotically. To prove that our proximity operator is approximately unbiased, we need to prove that the gap between our proximity operator and the identity function is approximately zero. When $|x| > c$, $p_\varphi(x) = x\left(1 - e^{\frac{c^2 - (x)^2}{\sigma^2}}\right)$, thus we only need to assure that $m(x) = e^{\frac{c^2 - (x)^2}{\sigma^2}} \to 0$ for a sufficiently large value of x (we consider $m(x) \approx 0$ if $m(x) < 10^{-4}$). By Proposition 2, we have $c \geq \sigma/\sqrt{2}$, resulting in $\frac{c^2 - (x)^2}{\sigma^2} \geq \frac{c^2 - (x)^2}{2c^2}$. Using $e^{-10} = 4.54 \times 10^{-5}$ and $\frac{c^2 - (x)^2}{2c^2} \leq \frac{c^2 - (x)^2}{\sigma^2} \leq -10$, we get $x \geq \sqrt{21}c \approx 4.6c$. That is, for our proximity operator, it is approximately unbiased when $x \geq 4.6c$. This completes the proof. ■

Appendix G
Proof of Theorem 1

Proof: There are five parameters, i.e., c, σ, $\boldsymbol{S}$, $\boldsymbol{U}$ and $\boldsymbol{V}$, in Algorithm 1, which will be theoretically analyzed that updating each of them in the proposed algorithm leads to the decrease of objective function.

First, when only updating c from c_1 to c_2 ($c_2 < c_1$) and fixing the remaining parameters, we divide $x \in \mathbb{R}$ into $|x_1| \leq c_2$, $c_2 < |x_2| \leq c_1$ and $|x_3| > c_1$, then we have

$$\begin{aligned}& l_{c_1,\sigma}(x) - l_{c_2,\sigma}(x) \\ =\ & l_{c_1,\sigma}(x_1) + l_{c_1,\sigma}(x_2) + l_{c_1,\sigma}(x_3) \\ & - l_{c_2,\sigma}(x_1) - l_{c_2,\sigma}(x_2) - l_{c_2,\sigma}(x_3) \\ =\ & a_1 + a_2 + a_3\end{aligned}$$

where $a_1 = l_{c_1,\sigma}(x_1) - l_{c_2,\sigma}(x_1)$, $a_2 = l_{c_1,\sigma}(x_2) - l_{c_2,\sigma}(x_2)$ and $a_3 = l_{c_1,\sigma}(x_3) - l_{c_2,\sigma}(x_3)$. According to the definition of

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

$l_{c,\sigma}(x)$ in (9), we get:

$$l_{c_1,\sigma}(x_1) = l_{c_2,\sigma}(x_1) = x_1^2/2$$
$$l_{c_1,\sigma}(x_2) = x_2^2/2$$
$$l_{c_1,\sigma}(x_3) = \frac{\sigma^2}{2}\left(1 - e^{\frac{c_1^2 - x_3^2}{\sigma^2}}\right) + \frac{c_1^2}{2}$$
$$l_{c_2,\sigma}(x_2) = \frac{\sigma^2}{2}\left(1 - e^{\frac{c_2^2 - x_2^2}{\sigma^2}}\right) + \frac{c_2^2}{2}$$
$$l_{c_2,\sigma}(x_3) = \frac{\sigma^2}{2}\left(1 - e^{\frac{c_2^2 - x_3^2}{\sigma^2}}\right) + \frac{c_2^2}{2}$$

Thus, $a_1 = 0$. Besides, according to Appendix B, for any $c_2 > 0$, a_2 increases with x_2, thus we have

$$a_2 > l_{c_1,\sigma}(c_2) - l_{c_2,\sigma}(c_2) = 0 \tag{48}$$

When $|x_3| > c_1$, $\frac{\partial l_{c,\sigma}(x)}{\partial c} = c \cdot (1 - e^{(c^2 - x^2)/\sigma^2}) > 0$, thus $l_{c_1,\sigma}(x_3) > l_{c_2,\sigma}(x_3)$ due to $c_1 > c_2$, resulting in $a_3 > 0$. Therefore, we get:

$$l_{c_1,\sigma}(x) - l_{c_2,\sigma}(x) > 0$$

Hence, $l_{c,\sigma}(x)$ increases with c.

Second, when just updating σ, since σ is not in the range, we take the derivative of $l_{c,\sigma}(x)$ w.r.t. σ. If $|x| \le c$, $\frac{\partial l_{c,\sigma}(x)}{\partial \sigma} = 0$, while if $|x| > c$, $\frac{\partial l_{c,\sigma}(x)}{\partial \sigma} = \sigma + (\frac{c^2 - x^2}{\sigma} - \sigma)e^{\frac{c^2 - x^2}{\sigma^2}} = \sigma(1 - (\frac{x^2 - c^2}{\sigma^2} + 1)e^{\frac{c^2 - x^2}{\sigma^2}}) \overset{d}{\ge} \sigma(1 - e^{\frac{x^2 - c^2}{\sigma^2}} e^{\frac{c^2 - x^2}{\sigma^2}}) = 0$, where d holds because $e^y \ge y + 1$ for any $y \in \mathbb{R}$, thus we obtain $\frac{\partial l_{c,\sigma}(x)}{\partial \sigma} \ge 0$.

Therefore, for the non-increasing c^{k+1} and σ^{k+1}, i.e., $c^{k+1} \le c^k$ and $\sigma^{k+1} \le \sigma^k$, $l_{c^{k+1},\sigma^{k+1}}(x) \le l_{c^k,\sigma^k}(x)$ holds. Note that the impact of c and σ on (18) and (20) is the same. Thus, we have:

$$\mathcal{C}_{c^{k+1},\sigma^{k+1}}\left(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k\right) \le \mathcal{C}_{c^k,\sigma^k}\left(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k\right) \tag{49}$$

For fixed c^{k+1} and σ^{k+1}, based on the property of alternating minimization, we have:

$$\begin{aligned}
&\mathcal{C}_{c^{k+1},\sigma^{k+1}}\left(\boldsymbol{U}^{k+1}, \boldsymbol{V}^{k+1}, \boldsymbol{S}^{k+1}\right) \\
&\le \mathcal{C}_{c^{k+1},\sigma^{k+1}}\left(\boldsymbol{U}^{k+1}, \boldsymbol{V}^{k}, \boldsymbol{S}^{k+1}\right) \\
&\le \mathcal{C}_{c^{k+1},\sigma^{k+1}}\left(\boldsymbol{U}^{k}, \boldsymbol{V}^{k}, \boldsymbol{S}^{k+1}\right) \\
&\le \mathcal{C}_{c^{k+1},\sigma^{k+1}}\left(\boldsymbol{U}^{k}, \boldsymbol{V}^{k}, \boldsymbol{S}^{k}\right)
\end{aligned} \tag{50}$$

Combining (49) and (50), for the non-increasing c and σ, which is satisfied owing to (32), the sequence $\{\mathcal{C}_{c^k,\sigma^k}(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k),\ k = 1, 2, \cdots\}$ converges since $\mathcal{C}_{c^k,\sigma^k}(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k)$ is bounded below. This completes the proof. ■

References

[1] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.

[2] M. Muma, W.-J. Zeng, and A. M. Zoubir, "Robust M-estimation based matrix completion," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2019, pp. 5476–5480.

[3] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2016.

[4] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.

[5] F. Nie, Z. Li, Z. Hu, R. Wang, and X. Li, "Robust matrix completion with column outliers," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12042–12055, Nov. 2022.

[6] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.

[7] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, Dec. 2009.

[8] T. Zhou and D. Tao, "Greedy bilateral sketch, completion & smoothing," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2013, pp. 650–658.

[9] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optim. Methods Softw.*, vol. 29, no. 2, pp. 239–263, 2014.

[10] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Opt.*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[11] K. C. Toh and S. W. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Optim.*, vol. 6, no. 15, pp. 615–640, Sep. 2010.

[12] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, Dec. 2012.

[13] J. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using power factorization," *IEEE Signal Process. Lett.*, vol. 16, no. 7, pp. 584–587, Jul. 2009.

[14] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "Global optimality in low-rank matrix optimization," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3614–3628, Jul. 2018.

[15] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "The global optimization geometry of low-rank matrix optimization," *IEEE Trans. Inf. Theory*, vol. 67, no. 2, pp. 1308–1331, Feb. 2021.

[16] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge, MA, USA: Cambridge Univ. Press, 2018.

[17] P. J. Huber and E. M. Ronchetti, *Robust Statistics*. Hoboken, NJ, USA: Wiley, 2009.

[18] Z. Lin, C. Xu, and H. Zha, "Robust matrix factorization by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 208–220, Jan. 2018.

[19] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust ℓ_1-norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1410–1417.

[20] T. Zhou and D. Tao, "Shifted subspaces tracking on sparse outlier for motion segmentation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1946–1952.

[21] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, Dec. 2001.

[22] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, 2010.

[23] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *Ann. Statist.*, vol. 37, no. 1, pp. 246–270, Feb. 2009.

[24] F. Wen, R. Ying, P. Liu, and R. C. Qiu, "Robust PCA using generalized nonconvex regularization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1497–1510, Jun. 2020.

[25] G. Marjanovic and V. Solo, "On ℓ_q optimization and matrix completion," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5714–5724, Nov. 2012.

[26] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, pp. 2287–2322, Aug. 2010.

[27] R. He, T. Tan, and L. Wang, "Robust recovery of corrupted low-rankmatrix by implicit regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 770–783, Apr. 2014.

[28] W. J. Zeng and H. C. So, "Outlier-robust matrix completion via ℓ_p-minimization," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1125–1140, Mar. 2018.

[29] F. Nie, H. Wang, X. Cai, H. Huang, and C. Ding, "Robust matrix completion via joint Schatten p-norm and ℓ_p-norm minimization," in *Proc. IEEE Int. Conf. Data Mining*, 2012, pp. 566–574.

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.

[30] F. Shang, Y. Liu, and J. Cheng, "Scalable algorithms for tractable Schatten quasi-norm minimization," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2016–2022.
[31] F. Shang, J. Cheng, Y. Liu, Z. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2066–2080, Sep. 2018.
[32] R. Chartrand, "Nonconvex splitting for regularized low-rank sparse decomposition," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5810–5819, Nov. 2012.
[33] Y. He, F. Wang, Y. Li, J. Qin, and B. Chen, "Robust matrix completion via maximum correntropy criterion and half-quadratic optimization," *IEEE Trans. Signal Process.*, vol. 68, pp. 181–195, 2020.
[34] Y. He and G. K. Atia, "Coarse to fine two-stage approach to robust tensor completion of visual data," *IEEE Trans. Cybern.*, early access, Sep. 5, 2022, doi: 10.1109/TCYB.2022.3198932..
[35] R. He, B.-G. Hu, W.-S. Zheng, and X.-W. Kong, "Robust principal component analysis based on maximum correntropy criterion," *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1485–1494, Jun. 2011.
[36] S. Seth and J. Prýncipe, "Compressed signal reconstruction using the correntropy induced metric," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2008, pp. 3845–3848.
[37] R. He, Y. Zhang, Z. Sun, and Q. Yin, "Robust subspace clustering with complex noise," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4001–4013, Nov. 2015.
[38] K. Guo, L. Liu, X. Xu, D. Xu, and D. Tao, "GoDec : Fast and robust low-rank matrix decomposition based on maximum correntropy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2323–2336, Jun. 2018.
[39] R. A. Maronna, D. R. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods.* New York, NY, USA: Wiley, 2006.
[40] R. He, W.-S. Zheng, T. Tan, and Z. Sun, "Half-quadratic based iterative minimization for robust sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 261–275, Feb. 2014.
[41] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.
[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge Univ. Press, 2004.
[43] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*, 2nd ed. Berlin, Germany: Springer, 2004.
[44] J. Tanner and K. Wei, "Low rank matrix completion by alternating steepest descent methods," *Appl. Comput. Harmon. Anal.*, vol. 40, no. 2, pp. 417–429, Mar. 2015.
[45] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality," *Math. Oper. Res.*, vol. 35, no. 2, pp. 438–457, May 2010.
[46] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
[47] Y. Chang, L. Yan, X.-L. Zhao, H. Fang, Z. Zhang, and S. Zhong, "Weighted low-rank tensor recovery for hyperspectral image restoration," *IEEE Trans. Cybern.*, vol. 50, no. 11, pp. 4558–4572, Nov. 2020.
[48] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.
[49] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 6, pp. 643–660, Jun. 2001.
[50] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 393–400.
[51] Q. Yao, J. T. Kwok, T. Wang, and T.-Y. Liu, "Large-scale low-rank matrix learning with nonconvex regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2628–2643, Nov. 2019.
[52] R. G. Bartle and D. R. Sherbert, *Introduction to Real Analysis*, 4th ed. Hoboken, NJ, USA: Wiley, 2011.

Zhi-Yong Wang received the B.S. degree in mechanical engineering from Zhengzhou University, Zhengzhou, China, in 2017, and the M.S. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2020. He is currently working toward the Ph.D. degree with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China. His research interests include nonconvex optimization, sparse recovery, matrix/tensor completion, and robust signal processing.

Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, Hong Kong, in 1990, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 1995. From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. During 1995–1996, he was a Postdoctoral Fellow with The Chinese University of Hong Kong, Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, fast and adaptive algorithms, multidimensional harmonic retrieval, robust signal processing, source localization, and sparse approximation. He has been on the editorial boards of *IEEE Signal Processing Magazine* during 2014–2017, IEEE TRANSACTIONS ON SIGNAL PROCESSING during 2010–2014, *Signal Processing* in 2010, and *Digital Signal Processing* in 2011. In 2017, he was also the Lead Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, special issue on Advances in Time/Frequency Modulated Array Signal Processing. In addition, he was an elected member in Signal Processing Theory and Methods Technical Committee during 2011–2016 of the IEEE Signal Processing Society where he was the Chair in the awards subcommittee during 2015–2016.

Abdelhak M. Zoubir (Fellow, IEEE) has been a Professor of signal processing and the Head of the Signal Processing Group with Technische Universität Darmstadt, Darmstadt, Germany, since 2003. He has authored or coauthored more than 500 journal and conference papers on the above areas. His research interests include statistical methods for signal processing with emphasis on bootstrap techniques, robust detection and estimation and array processing applied to radar, sonar, telecommunications, automotive monitoring and safety, and biomedicine. He was the recipient of several Best Paper awards, and 2018 IEEE Leo L. Beranek Meritorious Service Award. He was the General Chair and Technical Chair of numerous international IEEE conferences and workshops, most notably as Technical Co-Chair of ICASSP-14. He was on Publication Boards or as the Guest Editor of various journals, and the Editor-in-Chief of the *IEEE Signal Processing Magazine* duirng 2012–2014. He was the Chair of the IEEE Signal Processing Society (SPS) Technical Committee Signal Processing Theory and Methods (SPTM) during 2010–2011 and was on the Board of Governors of the IEEE SPS as a Member-at-Large during 2015–2017. From 2017 to 2018, he was the President of the European Association of Signal Processing. He is an IEEE Distinguished Lecturer (Class 2010–2011).

Authorized licensed use limited to: LAHORE UNIV OF MANAGEMENT SCIENCES. Downloaded on October 25,2023 at 11:15:24 UTC from IEEE Xplore. Restrictions apply.