# Adaptive Rank-One Matrix Completion using Sum of Outer Products

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

***Abstract*—Matrix completion refers to recovering a matrix from a small subset of its entries. It is an important topic because numerous real-world data can be modeled as low-rank matrices. One popular approach for matrix completion is based on low-rank matrix factorization, but it requires knowing the matrix rank, which is difficult to accurately determine in many practical scenarios. We propose a novel algorithm based on rank-one approximation that a matrix can be decomposed as a sum of outer products. The key idea is to find the basis vectors of the underlying matrix according to the observed entries, and gradually increase the vector number until an appropriate rank estimate is reached. In contrast to the conventional rank-one schemes that employ unchanging rank-one basis matrices, our algorithm performs completion from the vector viewpoint and is able to generate continuously updated rank-one basis matrices. Besides, we theoretically show that the developed method has a linear convergence rate and a smaller recovery error than existing rank-one based algorithms. Experimental results using both synthetic data and real-world images demonstrate that our solution has the best recovery performance among the competing algorithms when the observations are contaminated by Gaussian noise.**

***Index Terms*—Low-rank matrix completion, rank-one approximation, basis vector, linear convergence, alternating minimization**

## I. Introduction

MATRIX completion [1]–[3] aims to recover the missing entries given incomplete data and has been widely used in a burst of areas of science and engineering such as background initialization [4], [5], recommender system [6], [7], hyperspectral remote sensing [8], image inpainting [9], image super-resolution [10], multi-task learning [11] and deep learning [12], [13]. This is because many real-world data can be considered as low-rank matrices, whose rank is much smaller than the number of columns or rows. For example, the reason why matrix completion can recover the missing pixels of images is that their main information often concentrates on a subspace of much lower dimensionality [14], [15].

Matrix completion can be formulated as a constrained rank minimization problem [16]. Unfortunately, this problem is in general NP-hard since the rank is discrete and nonconvex. Meanwhile, nuclear norm minimization is exploited to solve matrix completion and theoretical guarantees can be found in [17]. Generally, nuclear norm minimization is recast as a semi-definite program (SDP) [18], [19], and hence can be solved by the interior-point method [20], [21], but its computational complexity is as high as $\mathcal{O}(p(m+n)^3+p^2(m+n)^2+p^3)$, where $p, m, n$ denote the numbers of observed entries, rows and columns of the target matrix with missing entries, respectively. On the other hand, more computationally efficient algorithms such as singular value thresholding (SVT) [22], fixed point continuation with approximating singular value decomposition (FPCA) [23], and accelerated proximal gradient with linesearch (APGL) [24], have been proposed. Cao *et al*. [10] utilize augmented Lagrange multiplier method to achieve image interpolation via nuclear norm minimization. However, these approaches still involve full singular value decomposition (SVD) at each iteration, resulting in high computational load, particularly for large-size matrices.

Another methodology, including singular value projection (SVP) [25], normalized iterative hard thresholding (NIHT) [26], and alternating projection (AP) [27], directly exploits the matrix completion problem with a rank constraint to attain a low-rank solution. In particular, both SVP and NIHT employ the gradient projection descent. Different from the nuclear norm based methods requiring full SVD calculation, SVP and NIHT only need truncated SVD to obtain the $r$ dominant singular values and the corresponding singular vectors, resulting in complexity reduction. Here, $r$ can be considered as an optimum rank value of the matrix, and although some methods [28], [38] aim to find a good estimation for $r$, their performance is sensitive to the choice of $r$.

To avoid performing SVD, low-rank matrix factorization approaches [29]–[34] have also been suggested. The key idea is to employ the product of two much smaller matrices to approximately fit the objective matrix so that the low-rank property is automatically fulfilled. Among them, low-rank matrix fitting (LMaFit) [32] is proposed, but it lacks the global convergence guarantees. Keshavan *et al*. [34] reformulate the LMaFit model into an SVD form and propose a gradient descent algorithm as the solver. In order to overcome the slow convergence of gradient descent, alternating minimization for matrix completion (AltMincomplete) [35] is proposed. Recently, Zhu *et al*. [30], [31] have shown that the low-rank matrix completion problem based on matrix factorization has no spurious local minima and obeys the strict saddle property which requires the cost function to have a directional negative curvature at all critical points but local minima, if the matrix completion cost function satisfies certain conditions. In other words, global optimality in low-rank matrix completion can be achieved with a well-conditioned cost function. Besides,

The work described in this paper was supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 11207922]. (Corresponding author: Xiao Peng Li and Hing Cheung So).

Z.-Y. Wang, X. P. Li and H. C. So are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China. A. M. Zoubir is with the Signal Processing Group at Technische Universität Darmstadt, 64283 Darmstadt, Germany. (E-mail: z.y.wang@my.cityu.edu.hk, x.p.li@my.cityu.edu.hk, hcso@ee.cityu.edu.hk, zoubir@spg.tu-darmstadt.de).

Li *et al*. [36] have analysed that the proximal alternating minimization (PAM) can converge to a second-order stationary point via random initialization under some mild conditions, which means that the PAM can avoid strict saddle points and attain the optimal solution. Furthermore, it is proved [37] that the optimal solution of low-rank matrix completion can be obtained even without the regularization as required in [30], [31]. However, these factorization based methods also require prior knowledge of rank $r$ which may not be available in many real-life situations.

The last type of methods utilizes successive rank-one based matrix completion [38]–[44], which can meet the low-rank property explicitly. Their key idea is to exploit the fact that any low-rank matrix can be expressed as a sum of $r$ rank-one matrices, and the most well-known representation is SVD where the left-singular vectors are orthonormal and so are right-singular vectors. Wang *et al*. [40], [41] propose the orthogonal rank-one matrix pursuit (OR1MP) algorithm and its computationally efficient variant, namely, the economic OR1MP (EOR1MP) algorithm by using orthogonal matching pursuit (OMP) [45], [46]. The main difference between OR1MP and EOR1MP is the way to refine the weights of the rank-one basis matrices after completing the current outer iteration. Compared with OR1MP, EOR1MP can reduce the storage complexity without increasing too much the recovery error, which is vital for large-scale data. However, these algorithms do not update the previously computed rank-one basis matrices at the end of each iteration, although they revise the weights during the process. Likewise, Shi *et al*. [38] propose another rank-one matrix completion method called L1MC-RF. Their first step is to estimate the rank of the original matrix by using $l_1$-norm regularization, and then perform truncated SVD at each iteration for matrix completion. However, the performance of [38] is sensitive to the matrix rank. If the rank estimated by the $l_1$-norm is higher or lower than the true rank, it will lead to a poor recovery performance. As the iteration is updated by computing the truncated SVD, expensive computation is also involved, especially for the large-size problems. Thirdly, this method is not robust to disturbance, and even a small noise level can give rise to a large degradation in the recovery accuracy.

In this paper, we devise a matrix completion algorithm using sum of outer products without the need of rank information, which is referred to as adaptive rank-one matrix completion (AROMC). The core of AROMC is to seek the hidden low-dimensional vector subspace that captures most of the information in the incomplete target matrix and then project the matrix orthogonally into that subspace to obtain the optimal weights of basis vectors so as to make the estimated matrix best fit the observed entries. By "best", we mean the smallest in the mean square error sense. In each iteration, one basis vector is added to increment the rank of estimated matrix, while alternating minimization and least squares (LS) [47] are adopted to compute the vectors and weights. Subsequently, we recover the matrix via summing the outer products of basis vectors and the corresponding weight vectors. An appealing feature of AROMC is that it converges linearly, contrasting to the OMP that has a sub-linear convergence rate for sparse vector recovery [46]. The main contributions of our paper are as follows:

1) We devise the AROMC algorithm via seeking a basis vector subspace that is perpendicular to the space spanned by the columns of the residual matrix, which is analogous to OMP. That means in the column vectors of residual matrix, all information associated with the basis vectors is orthogonally projected onto the basis vector subspace, thus resulting in a better recovery performance.
2) We theoretically prove that the convergence rate of AROMC is linear. As a result, we only require $\mathcal{O}\left(\log(1/\epsilon)\right)$ iterations to obtain an $\epsilon$-accuracy solution.
3) We theoretically show that the recovery error of AROMC is smaller than that of OR1MP. It is worth noting that the AROMC does not require refining the weights of rank-one matrices at the end of each iteration.

The remainder of this paper is organized as follows. In Section II, we introduce notations and related works. The AROMC algorithm development, and its differences with the relevant schemes in the literature, are presented in Section III. In Section IV, the convergence of the proposed algorithm, comparative recovery errors, and computational complexity, are analyzed. Numerical simulation results based on synthetic data and real images are discussed 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, vectors and matrices are represented by italic, bold lower-case and bold upper-case letters, respectively. The $i$th element of a vector $\boldsymbol{a}$ is denoted by $a_i$, and the $(i,j)$ entry of a matrix $\boldsymbol{A}$ is denoted by $A_{ij}$. In particular, $\mathbf{0}$ and $\boldsymbol{O}$ are the vector and matrix with all entries being zeros, respectively. The $(\cdot)^\dagger$, $(\cdot)^{-1}$ and $(\cdot)^T$ are the pseudo-inverse operator, inverse operator and transpose operator, respectively. Let $\Omega \subset \{1,\cdots,m\} \times \{1,\cdots,n\}$ represent the index set of the observed entries of $\boldsymbol{A}$, 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, $\|\boldsymbol{A}\|_F = \sqrt{\sum_i^m \sum_j^n A_{ij}^2}$ 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 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., $\|\boldsymbol{A}\| = \|\boldsymbol{A}\|_F$, and we define $\langle \boldsymbol{A}, \boldsymbol{B}\rangle_\Omega = \langle \boldsymbol{A}_\Omega, \boldsymbol{B}_\Omega\rangle$ and $\|\boldsymbol{A}\|_\Omega^2 = \langle \boldsymbol{A}, \boldsymbol{A}\rangle_\Omega$. In particular, $\boldsymbol{u}_i \in \mathbb{R}^m$ and $\boldsymbol{v}_i \in \mathbb{R}^n$ for $i = 1, 2, \cdots, r$, are the column vectors of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, and $\overline{\boldsymbol{U}}$ and $\overline{\boldsymbol{u}}$ represent the basis matrix and vector, whose values do not change as the number of iterations increases. We denote $\sigma_i(\boldsymbol{A})$

as the $i$th largest singular value of $\boldsymbol{A}$ and let $\boldsymbol{A}_{j,:}$(resp. $\boldsymbol{A}_{:,j}$) represent its $j$th row (resp. column). Finally, $|\boldsymbol{A}|$, $|\boldsymbol{a}|$, $|\mathcal{I}|$ stand for the cardinality of the corresponding matrix, vector and set, respectively.

### B. Related Works

The task of matrix completion is to seek a low-rank matrix $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ to approximate an incomplete matrix $\boldsymbol{X}_\Omega$ and determine the missing entries. Mathematically, it is modeled as a rank minimization problem:

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

However, (1) is non-convex and NP-hard. In practice, the nuclear norm is exploited, leading to

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

where the nuclear norm $\|\boldsymbol{M}\|_*$ is the sum of singular values of $\boldsymbol{M}$. Although this is a convex optimization problem, it involves performing SVD at each iteration. To circumvent SVD, matrix factorization has been suggested, leading to the following optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \left\|\boldsymbol{U}\boldsymbol{V}^T - \boldsymbol{X}\right\|_\Omega^2, \ \text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega \tag{3}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, $\boldsymbol{V} \in \mathbb{R}^{n\times r}$, $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}^T$ and $r$ is the rank of $\boldsymbol{X}$. Apparently, $\boldsymbol{M}$ also has rank $r$. However, this approach needs to know the value of $r$, which is often an intractable task in many real settings. In order to address this problem, another low-rank matrix completion method is put forth and the basic idea is analogous to SVD, that is, any matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ with rank $r$ can be written as a linear combination of $r$ rank-one matrices:

$$\boldsymbol{X} = \sum_{i=1}^{r} \theta_i \boldsymbol{Y}_i \tag{4}$$

where $\boldsymbol{Y}_i \in \mathbb{R}^{m\times n}$ with $\|\boldsymbol{Y}_i\|_F = 1$, and $\theta_i$ are the $i$th rank-one basis matrix and its corresponding weight, respectively. Based on (4), Wang $et\ al.$ [40], [41] have developed the following rank-one matrix completion approach:

$$\min_{\theta_i} \ \left\|\boldsymbol{X} - \sum_{i=1}^{\tau} \theta_i \boldsymbol{Y}_i\right\|_\Omega^2 \tag{5}$$

where $\tau$ is the maximum allowable rank value. It starts with modeling $\boldsymbol{X}$ using a rank-one matrix, then the rank increments until a stopping criterion is reached. In their study, $\boldsymbol{Y}_i = \boldsymbol{l}_i\boldsymbol{r}_i^T$, where $\boldsymbol{l}_i$ and $\boldsymbol{r}_i$ are the top left and right singular vectors of the current residual matrix, respectively. The $\theta_i$ is uniquely determined by performing an orthogonal projection of $\boldsymbol{X}_\Omega$ onto the subspace spanned by $\boldsymbol{Y}_i$. Besides, in each iteration, one rank-one basis matrix is added so that the rank of the estimated matrix increases until algorithm termination. Hence, the matrix rank can be automatically determined, which is different from (3). However, all rank-one basis matrices in (5) are computed once and there are no further adjustments for them.

## III. Proposed Algorithm

To obtain a more flexible rank-one basis matrix representation, we first express $\boldsymbol{X}$ as a sum of $r$ outer products, that is:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V}^T = \boldsymbol{u}_1\boldsymbol{v}_1^T + \boldsymbol{u}_2\boldsymbol{v}_2^T + \cdots + \boldsymbol{u}_r\boldsymbol{v}_r^T \tag{6}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r]$, $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r]$, and the column vectors in $\boldsymbol{U}$ or $\boldsymbol{V}$ are not required to be orthogonal. It is worth mentioning that (6) is different from the SVD whose basis vectors are orthonormal. In this study, we give a new interpretation of (6) in that the column vectors, i.e., $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r$, are considered as basis vectors to be determined while $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_r$, are the corresponding weight vectors. Before proceeding, we first point out that (5) cannot update the previously computed rank-one basis matrices, via rewriting (5) as

$$\min_{\boldsymbol{w}_i} \ \left\|\boldsymbol{X} - \sum_{i=1}^{\tau} \boldsymbol{l}_i\boldsymbol{w}_i^T\right\|_\Omega^2, \ \text{s.t. } \boldsymbol{w}_i = \theta_i\boldsymbol{r}_i \tag{7}$$

where $\boldsymbol{w}_i$ refers to the weight vector of $\boldsymbol{l}_i$. It is easily observed that although $\theta_i$ can take any value, $\boldsymbol{w}_i$ is constrained by $\boldsymbol{r}_i$. In order to handle this problem, we aim to seek the basis vector subspace spanned by $\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_r$, instead of finding the rank-one basis matrices as in [40], [41]. Our idea is to modify (5) as

$$\min_{\boldsymbol{u}_p, \{\boldsymbol{v}_i\}} \ \left\|\boldsymbol{X} - \sum_{i=1}^{p-1} \overline{\boldsymbol{u}}_i\boldsymbol{v}_i^T - \boldsymbol{u}_p\boldsymbol{v}_p^T\right\|_\Omega^2 \tag{8}$$

where $p$ represents the $p$th outer iteration, $\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \cdots, \overline{\boldsymbol{u}}_{p-1}$, are the basis vectors that are kept fixed in the subsequent iterations, while $\boldsymbol{u}_p$ and $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_p$ are the $p$th basis vector and weight vectors, respectively, to be updated. Hence, there are $(p+1)$ unknown vectors, i.e., $\boldsymbol{u}_p, \boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_p$. As the weight vectors are updated at each iteration, the previously computed rank-one basis matrices can be continuously refined. In fact, compared with the result at the $(p-1)$th iteration that attains a rank-$(p-1)$ matrix, the $p$th outer iteration then produces a rank-$p$ matrix, and the relationship between them is that they have $(p-1)$ identical basis vectors, that is, $\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \cdots, \overline{\boldsymbol{u}}_{p-1}$. Apparently, the rank of the obtained matrix increases as the number of basis vectors increases.

To tackle (8), alternating minimization is adopted, whose nature is to optimize one of the factors in (8) while fixing the other in an alternate manner. To be more specific, in the $k$th ($k = 1, 2, \cdots$) inner iteration, $\boldsymbol{V}^k$ and $\boldsymbol{u}_p^k$ are alternately minimized via:

$$\boldsymbol{V}^k = \arg\min_{\boldsymbol{V}} \ \left\|\boldsymbol{X} - \boldsymbol{U}^{k-1}\boldsymbol{V}^T\right\|_\Omega^2 \tag{9}$$

$$\boldsymbol{u}_p^k = \arg\min_{\boldsymbol{u}_p} \ \left\|\boldsymbol{X} - \left[\overline{\boldsymbol{U}}_{p-1}, \boldsymbol{u}_p\right](\boldsymbol{V}^k)^T\right\|_\Omega^2 \tag{10}$$

where $\overline{\boldsymbol{U}}_{p-1} = [\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \cdots, \overline{\boldsymbol{u}}_{p-1}]$, $\boldsymbol{U}^{k-1} = \left[\overline{\boldsymbol{U}}_{p-1}, \boldsymbol{u}_p^{k-1}\right]$. We now focus on (9) for a fixed $\boldsymbol{U}^{k-1}$:

$$\min_{\boldsymbol{V}} \ f(\boldsymbol{V}) \ := \ \left\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\right\|_\Omega^2 \tag{11}$$

where the superscript $(\cdot)^k$ is dropped for notational simplicity. The problem (11) is equivalent to addressing the following $n$

independent subproblems because $f(\boldsymbol{V})$ can be decoupled with respect to (w.r.t.) $\boldsymbol{V}_{j,:}$ for $j = 1, 2, \cdots, n$,

$$\min_{\boldsymbol{V}_{j,:}} \ f(\boldsymbol{V}_{j,:}) \ := \ \left\|\boldsymbol{X}_{:,j} - \boldsymbol{U}\boldsymbol{V}_{j,:}^T\right\|_\Omega^2. \tag{12}$$

We first locate the row indices of the observed values in the $j$th column of $\boldsymbol{X}$, which are contained in $\mathcal{I}_j$, and hence the total index set $\mathcal{I}$ for $\boldsymbol{X}$ is $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \cdots, \mathcal{I}_n\}$. Apparently, $\sum_{j=1}^n |\mathcal{I}_j| = |\Omega|$. Defining $\boldsymbol{U}_{\mathcal{I}_j} \in \mathbb{R}^{|\mathcal{I}_j| \times p}$ which contains the $|\mathcal{I}_j|$ rows indexed by $\mathcal{I}_j$:

$$\boldsymbol{U}_{\mathcal{I}_j} = \begin{bmatrix} \boldsymbol{U}_{\mathcal{I}_{j_1},:} \\ \boldsymbol{U}_{\mathcal{I}_{j_2},:} \\ \vdots \\ \boldsymbol{U}_{\mathcal{I}_{j_{|\mathcal{I}_j|}},:} \end{bmatrix} \tag{13}$$

where $\mathcal{I}_{j\,i}$ stands for the $i$th element of $\mathcal{I}_j$, and the associated vector $(\boldsymbol{X}_{:,j})_{\mathcal{I}_j} = \left[\boldsymbol{X}_{\mathcal{I}_{j_1},j}, \cdots, \boldsymbol{X}_{\mathcal{I}_{j_{|\mathcal{I}_j|}},j}\right]^T \in \mathbb{R}^{|\mathcal{I}_j|}$, (12) is then rewritten as:

$$\min_{\boldsymbol{V}_{j,:}} \ f(\boldsymbol{V}_{j,:}) \ := \ \left\|(\boldsymbol{X}_{:,j})_{\mathcal{I}_j} - \boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{V}_{j,:}^T\right\|^2 \tag{14}$$

whose solution is simply the linear LS estimate:

$$\boldsymbol{V}_{j,:}^T = \left(\boldsymbol{U}_{\mathcal{I}_j}^T\boldsymbol{U}_{\mathcal{I}_j}\right)^{-1}\boldsymbol{U}_{\mathcal{I}_j}^T(\boldsymbol{X}_{:,j})_{\mathcal{I}_j} \tag{15}$$

and the computational complexity is $\mathcal{O}(|\Omega|p^2)$.

In a similar manner, (10) becomes:

$$\begin{aligned} \min_{\boldsymbol{u}_p} \ f(\boldsymbol{u}_p) \ &:= \ \left\|\left(\boldsymbol{X} - \overline{\boldsymbol{U}}_{p-1}\boldsymbol{V}_{p-1}\right) - \boldsymbol{u}_p\boldsymbol{v}_p^T\right\|_\Omega^2 \\ &= \left\|\left(\boldsymbol{X} - \overline{\boldsymbol{U}}_{p-1}\boldsymbol{V}_{p-1}\right)^T - \boldsymbol{v}_p\boldsymbol{u}_p^T\right\|_\Omega^2 \\ &= \left\|\boldsymbol{R}_p' - \boldsymbol{v}_p\boldsymbol{u}_p^T\right\|_\Omega^2 \end{aligned} \tag{16}$$

where $\boldsymbol{V}_{p-1} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{p-1}]$ and $\boldsymbol{R}_p' = (\boldsymbol{X} - \overline{\boldsymbol{U}}_{p-1}\boldsymbol{V}_{p-1})^T$. Problem (16) is the special case of (11) with single column and can be decomposed as the following $m$ subproblems:

$$\min_{(u_p)_i} \ f\left((u_p)_i\right) \ := \ \left\|\left(\boldsymbol{R}_p'\right)_{:,i} - \boldsymbol{v}_p\,(u_p)_i\right\|_\Omega^2 \tag{17}$$

which is equal to

$$\min_{(u_p)_i} \ f\left((u_p)_i\right) \ := \ \left\|\left(\left(\boldsymbol{R}_p'\right)_{:,i}\right)_{\mathcal{J}_i} - (\boldsymbol{v}_p)_{\mathcal{J}_i}\,(u_p)_i\right\|^2 \tag{18}$$

where $\mathcal{J}_i$ is the set of row indices of the observed values in the $i$th column of $\boldsymbol{R}_p'$ for $i = 1, 2, \cdots, m$. The LS solution of (18) is:

$$(u_p)_i = \left(\boldsymbol{v}_p^T\right)_{\mathcal{J}_i}\left(\left(\boldsymbol{R}_p'\right)_{:,i}\right)_{\mathcal{J}_i} / \left(\left(\boldsymbol{v}_p^T\right)_{\mathcal{J}_i}(\boldsymbol{v}_p)_{\mathcal{J}_i}\right) \tag{19}$$

and the corresponding computational complexity is $\mathcal{O}(|\Omega|)$.

The steps of AROMC are summarized in Algorithm 1. Note that there are two kinds of iterations, i.e. an inner iteration and an outer iteration. The inner iteration is mainly used to find the solutions to basis vectors and the associated weight vectors via alternating minimization. Defining $\varepsilon\left(\boldsymbol{U}_p^k, \boldsymbol{V}_p^k\right) = \left\|\boldsymbol{X} - \boldsymbol{U}_p^k\left(\boldsymbol{V}_p^k\right)^T\right\|_\Omega^2 / \|\boldsymbol{X}\|_\Omega^2$ and $\delta = \varepsilon\left(\boldsymbol{U}_p^k, \boldsymbol{V}_p^k\right) - \varepsilon\left(\boldsymbol{U}_p^{k+1}, \boldsymbol{V}_p^{k+1}\right)$, if $\delta$ is less than a threshold, we say that $\boldsymbol{u}_p^k$ and $\boldsymbol{V}_p^k$ satisfy the convergence condition and the inner iteration will stop. In this paper, we set the threshold as $10^{-5}$. The outer iteration aims to increase the dimension of the basis space, that is, increasing one basis vector at each outer iteration. Its stopping conditions are based on the maximum allowable number of outer iterations $I_m$, absolute residual error (RE) with $\mathrm{RE} = \|\boldsymbol{R}_{p-1}\|_F - \|\boldsymbol{R}_p\|_F$ or the relative gap (RG) of the Frobenius norm of the two most recent adjacent rank-one matrices, which is expressed as [48]:

$$\mathrm{RG} = \frac{\left\|\boldsymbol{u}_{p-1}\boldsymbol{v}_{p-1}^T\right\|_F - \left\|\boldsymbol{u}_p\boldsymbol{v}_p^T\right\|_F}{\left\|\boldsymbol{u}_{p-1}\boldsymbol{v}_{p-1}^T\right\|_F} \tag{20}$$

When the outer iteration number reaches $I_m$, $\mathrm{RE} \le \eta$ or $\mathrm{RG} \le \zeta$, where $I_m$, $\eta$ and $\zeta$ are user-defined tolerances, the algorithm terminates. We use RG as the termination condition because for the low-rank matrix as in images, which is contaminated by Gaussian noise, the gap of singular value above the "elbow" is bigger than that below the "elbow" [49], [50]. Compared with the matrix constructed by the singular values above the elbow and the associated singular vectors, the matrix that is generated by the singular values below the elbow and associated singular vectors are dominated by noise. Hence, the latter should not be involved in the estimated matrix. Thereby, the proposed algorithm attempts to find a sum of rank-one matrices with less noise by stopping the iterative procedure before the elbow. Since for the $p$th rank-one matrix, $\left\|\boldsymbol{u}_p\boldsymbol{v}_p^T\right\|_F$ is approximately

**Algorithm 1** Adaptive Rank-One Matrix Completion (AROMC) using Sum of Outer Products

**Input:** Incomplete matrix $\boldsymbol{X}_\Omega$, index set $\Omega$, and tolerance parameters $I_m$, $\eta$ and $\zeta$
**Initialize:** $\overline{\boldsymbol{U}}_0 = \emptyset$
**for** $p = 1, 2, \cdots, I_m$ **do**
  // In the $p$th outer iteration
  Randomly initialize $\boldsymbol{u}_p$ and define $\boldsymbol{U}_p^0 = [\overline{\boldsymbol{U}}_{p-1}, \boldsymbol{u}_p]$.
  **for** $k = 1, 2, \cdots$ **do**
    // In the $k$th inner iteration, use alternating LS to find $p$ outer products
    **for** $j = 1, 2, \cdots, n$ **do**
      // Fix $\left(\boldsymbol{U}_p^{k-1}\right)_{\mathcal{I}_j}$, optimize $\boldsymbol{V}_p$
      $\left(\boldsymbol{V}_p^k\right)_{j,:} \longleftarrow \arg\min_{\boldsymbol{V}_{j,:}} \left\|(\boldsymbol{X}_{:,j})_{\mathcal{I}_j} - \left(\boldsymbol{U}_p^{k-1}\right)_{\mathcal{I}_j}\boldsymbol{V}_{j,:}^T\right\|^2$
    **end for**
    **for** $i = 1, 2, \cdots, m$ **do**
      // Fix $\left(\boldsymbol{V}_p^k\right)_{\mathcal{J}_i}$, optimize $\boldsymbol{u}_p^k$
      $\left(u_p^k\right)_i \longleftarrow \arg\min_{(u_p)_i} \left\|\left(\left(\boldsymbol{R}_p'\right)_{:,i}\right)_{\mathcal{J}_i} - \left(\boldsymbol{v}_p^k\right)_{\mathcal{J}_i}(u_p)_i\right\|^2$
    **end for**
    Stop, if convergent.
  **end for**
  Set $\overline{\boldsymbol{u}}_p = \boldsymbol{u}_p^k$, $\overline{\boldsymbol{U}}_p = [\overline{\boldsymbol{U}}_{p-1}, \overline{\boldsymbol{u}}_p]$, and calculate $\boldsymbol{M} = \overline{\boldsymbol{U}}_p\boldsymbol{V}_p^k$.
  Stop, if a termination condition is satisfied.
**end for**
**Output:** $\boldsymbol{M}$.

equal to the $p$th singular value, hence the gap of singular values is calculated by RG in this study.

**Remark 1.** The proposed algorithm is different from (3), although they both need to find $\boldsymbol{V}$. We highlight the two main points to illustrate their differences. The first one is that when the matrix rank $r$ is known, given $\boldsymbol{V}$, all the $r$ column vectors in $\boldsymbol{U}$ need to be determined in (3), but for the proposed algorithm, the first $(r-1)$ column vectors are fixed and only the last column vector needs to be solved. The second point is that when $r$ is unknown, in contrast to (3) that is not applicable, AROMC can complete $\boldsymbol{X}_\Omega$ because it gradually increases the number of basis vectors until an appropriate rank estimate is reached. This point is extremely important because it is often difficult to determine the ranks of real-world data.

**Remark 2.** AROMC is also different from OR1MP, EOR1MP and L1MC-RF, although they are all based on the idea of rank-one matrices for matrix completion. The difference between AROMC and OR1MP or EOR1MP is that firstly, the bases they find are different. The OR1MP and EOR1MP compute rank-one basis matrices, while AROMC aims to seek basis vectors, and those basis vectors are used to update rank-one basis matrices. However, the previously computed basis matrices in OR1MP and EOR1MP remain unchanged, so AROMC has a more flexible basis subspace. Secondly, OR1MP and EOR1MP also need to refine the weights of different rank-one basis matrices after completing the current iteration, but AROMC only involves weight vectors and they are solved during the iterative procedure. L1MC-RF adopts the $\ell_1$-norm and truncated SVD to estimate the rank and achieve matrix completion, respectively. Nevertheless, AROMC can obtain an appropriate rank by gradually increasing its basis vectors and applies alternating minimization to find solutions.

## IV. Theoretical Analysis

In this section, we first show that the basis vectors in our outer product representation are independent of each other and AROMC achieves a linear convergence rate. Besides, we give a proof about the relationship of the recovery error among AROMC, OR1MP, and PAM in theory. Finally, the computational complexity of the proposed algorithm is examined.

### A. Basic Properties

In this subsection, we show that the basis vectors in AROMC are independent and also perpendicular to the space spanned by the column vectors of the residual matrix. In addition, Algorithm 1 achieves a linear convergence rate. The results are shown in the following propositions and theorems, and their corresponding proofs are provided in the Appendix.

**Proposition 1.** *All basis vectors $\overline{\boldsymbol{u}}_i$ in $\overline{\boldsymbol{U}} = [\overline{\boldsymbol{u}}_1, \overline{\boldsymbol{u}}_2, \cdots, \overline{\boldsymbol{u}}_p]$ are independent.*

We then prove that the basis vector subspace is perpendicular to the space constructed by the column vectors of the residual matrix, i.e., the column vector space of $\overline{\boldsymbol{U}}_p$ is perpendicular to that of $\boldsymbol{R}_{p+1}$.

**Proposition 2.** $\overline{\boldsymbol{U}}_p^T \boldsymbol{R}_{p+1} = \boldsymbol{O}$, *for* $p = 1, 2, \cdots$.

We next give a proof on the linear convergence rate of the proposed algorithm. Prior to presenting the proof, it is necessary to introduce two important lemmas.

Let $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ be a matrix with rank $r$ and it is well known that $\boldsymbol{X}$ can be decomposed as

$$\boldsymbol{X} = \sum_{i=1}^{r} \sigma_i \boldsymbol{\alpha}_i \boldsymbol{\beta}_i^T \tag{21}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ are the singular values, $\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_r \in \mathbb{R}^m$ and $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_r \in \mathbb{R}^n$ are the corresponding left singular and right singular vectors, respectively. Equation (21) gives us an approach to construct the best low-rank approximation for $\boldsymbol{X}$ by the following lemma.

**Lemma 1.** *(Eckart-Young-Mirsky Theorem) [51]–[54] Suppose $\boldsymbol{X}$ has the SVD as in (21). If $k \leq r$, then the matrix $\boldsymbol{M} = \sum_{i=1}^{k} \sigma_i \boldsymbol{\alpha}_i \boldsymbol{\beta}_i^T$ satisfies*

$$\|\boldsymbol{X} - \boldsymbol{M}\|_F^2 \leq \|\boldsymbol{X} - \boldsymbol{Y}\|_F^2 \tag{22}$$

*where $\boldsymbol{Y}$ is the set of $\mathbb{R}^{m\times n}$ matrices with rank at most $k$ over $\mathbb{R}$. Lemma 1 means that $\boldsymbol{M}$ is the unique optimal solution of the following optimization problem*

$$\min_{\boldsymbol{M}} \ \|\boldsymbol{M} - \boldsymbol{X}\|_F^2, \ \textit{s.t.} \ \operatorname{rank}(\boldsymbol{M}) = k \tag{23}$$

Another lemma is the latest theorem for global optimality in low-rank optimization [55]–[62]. Recently, Zhu *et al.* [30], [31] developed a new theory on the global optimality convergence in low-rank matrix factorization in order to tackle the problem (3). They recast (3) as a general matrix factorization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ f(\boldsymbol{U}\boldsymbol{V}^T) \ \text{s.t.} \ \boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}^T \tag{24}$$

where $\boldsymbol{U} \in \mathbb{R}^{m\times r}$, $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ and $f(\cdot)$ is a cost function that evaluates how well a candidate solution approximates the observations. Apparently, $f(\cdot)$ is the Frobenius norm in our study. Problem (24) is non-convex because of the factorization $\boldsymbol{U}\boldsymbol{V}^T$, even when $f(\cdot)$ is convex. It may have spurious local minima or "bad" saddle points at first sight. However, according to [30], [31], if the cost function satisfies some mild conditions, the global minimum of (24) is obtained. Now, we summarize their main results in Lemma 2.

**Lemma 2.** *(Global optimality theorem) [30], [31] If the function $f(\boldsymbol{M})$ satisfies the following two assumptions, the globally optimal solution of (24) is obtained.*

Assumption 1: *$f(\boldsymbol{M})$ has a critical point at $\boldsymbol{M}^\star \in \mathbb{R}^{m\times n}$ which has rank $r$.*

Assumption 2: *$f(\boldsymbol{M})$ is $(2r, 4r)$-restricted strongly convex and smooth, i.e., for any $m \times n$ matrices $\boldsymbol{M}$ and $\boldsymbol{G}$ with $\operatorname{rank}(\boldsymbol{M}) \leq 2r$ and $\operatorname{rank}(\boldsymbol{G}) \leq 4r$, the Hessian of $f(\boldsymbol{M})$ satisfies the following equation for some positive $a$ and $b$:*

$$a\,\|\boldsymbol{G}\|_F^2 \leq \left[\nabla^2 f(\boldsymbol{M})\right](\boldsymbol{G}, \boldsymbol{G}) \leq b\,\|\boldsymbol{G}\|_F^2 \tag{25}$$

*In addition, this theorem is not only applicable for the exact-parameterization case where $\operatorname{rank}(\boldsymbol{M}^\star) = r$, but also suitable*

*for the under-parameterization case where* $\mathrm{rank}(\boldsymbol{M}^\star) < r$ *and the over-parameterization case where* $\mathrm{rank}(\boldsymbol{M}^\star) > r$. Assumption 2 *makes sure that the function* $f(\boldsymbol{M})$ *has no spurious local minima and obeys the strict saddle property, so the critical point of* Assumption 1 *is the globally optimal solution.*

It is worth noting that given a low-rank matrix $\boldsymbol{X}$ with rank $r$, Lemmas 1 and 2 show two different schemes to find the optimal approximation of $\boldsymbol{X}$. Their main difference is that the vectors $\boldsymbol{\alpha}_i$ (or $\boldsymbol{\beta}_i$) in Lemma 1 are orthogonal, while those vectors in Lemma 2 are just independent. We may say that OR1MP takes the idea of Lemma 1, while AROMC and PAM are based on Lemma 2. Since $\boldsymbol{M}$ in Lemma 2 is fully observed, it is necessary to verify that when the entries in $\boldsymbol{M}$ are partially observed, Lemma 2 holds.

**Proposition 3.** *For an* $\mu$*-incoherent and incomplete matrix, Lemma 2 holds with* $a = (1-\delta_{4r})p$ *and* $b = (1+\delta_{4r})p$*, if* $\Omega$ *is drawn according to Bernoulli sampling with probability* $p \geq 16C\mu^2 r^2 \log n/\delta_{4r}^2 m$.

Now, we show that the proposed algorithm converges linearly.

**Theorem 1.** *The AROMC algorithm achieves a linear convergence rate, which satisfies*

$$\|\boldsymbol{R}_p\| \leq \beta^{p-1} \|\boldsymbol{X}\|_\Omega ,$$

*where* $0 \leq \beta < 1$*, and* $p \geq 1$.

### B. Comparison of Recovery Error

In this section, we show the relationship in the recovery error among AROMC, OR1MP and PAM. Since these are different types of algorithms, it is reasonable to compare the recovery error according to the rank information. When the matrix rank is unknown, the recovery error relationship between OR1MP and AROMC is shown in Theorem 2. When the matrix rank is given, the relationship of these three algorithms is described in Theorem 3. In order to present the process of the proof clearly, we denote $\{\overline{\boldsymbol{u}}_p, \overline{\boldsymbol{v}}_p\}$ and $\{\hat{\boldsymbol{u}}_p, \hat{\boldsymbol{v}}_p\}$ as the optimal solutions of AROMC and OR1MP in the $p$th iteration, respectively. Besides, $\boldsymbol{R}_p^a, \boldsymbol{R}_p^o$, and $\boldsymbol{R}_p^m$ represent the residual matrices of AROMC, OR1MP and PAM in the $(p-1)$th iteration, respectively.

**Proposition 4.**

$$\begin{aligned} &\min_{\boldsymbol{u}_p, \{\boldsymbol{v}_i\}} \left\| \boldsymbol{X} - \sum_{i=1}^{p-1} \overline{\boldsymbol{u}}_i \boldsymbol{v}_i^T - \boldsymbol{u}_p \boldsymbol{v}_p^T \right\|^2 \\ &= \min_{\boldsymbol{u}_p, \{\boldsymbol{v}_i'\}, \{\theta_i\}} \left\| \boldsymbol{X} - \sum_{i=1}^{p-1} \overline{\boldsymbol{u}}_i \theta_i (\boldsymbol{v}_i')^T - \boldsymbol{u}_p \theta_p (\boldsymbol{v}_p')^T \right\|^2 \end{aligned} \tag{26}$$

**Theorem 2.** *When the matrix rank is unknown, the relationship between OR1MP and AROMC in terms of recovery error is*

$$\left\|\boldsymbol{R}_p^a\right\|^2 \leq \left\|\boldsymbol{R}_p^o\right\|^2 . \tag{27}$$

Given the rank information of matrix, we will theoretically show that the recovery error of PAM is the infimum of that of AROMC and OR1MP in the following.

**Theorem 3.** *Given that the matrix rank* $r$ *is known, the relationship among AROMC , OR1MP and PAM in terms of recovery error is*

$$\|\boldsymbol{R}^m\|^2 \leq \left\|\boldsymbol{R}_{r+1}^a\right\|^2 \leq \left\|\boldsymbol{R}_{r+1}^o\right\|^2 . \tag{28}$$

**Remark 3.** When the rank of the estimated matrix obtained by different algorithms is the true rank of the matrix, we analyze that the recovery error among the three different algorithms satisfies (28). Although the PAM can attain the smallest recovery error, it requires knowing the matrix rank which is often difficult to estimate accurately. Therefore, the proposed algorithm is advantageous over many existing algorithms, because it does not need to know the matrix rank in advance, and the recovery error is not larger than that of OR1MP in theory.

### C. Computational Complexity

The AROMC computational complexity in the current iteration depends on the $p$th outer iteration and $|\Omega|$. For (14), we assume that there are $l_j$ observed entries in the $j$th column of $\boldsymbol{X}$, and hence the computational complexity of the LS solution is $\mathcal{O}(Q_1 l_j p^2)$, where $Q_1$ is the number of iterations in the alternating minimization. Generally, $Q_1$ with a value of several tens can satisfy the convergence condition [43]. Hence the computational complexity of the proposed algorithm in the current iteration is $\mathcal{O}(Q_1|\Omega|p^2)$, where $|\Omega| = \sum_{j=1}^n l_j$. Assuming that the matrix rank is $r$, the total complexity of the AROMC is $\mathcal{O}(Q_1|\Omega|r^3)$. Without loss of generality, suppose $m \geq n$, the computational complexity of full SVD is $\mathcal{O}(mn^2 + n^3)$. Therefore, for an incomplete low-rank matrix with rank $r \ll n$, the computational complexity of the proposed algorithm is smaller than that of the others requiring full SVD calculation.

## V. EXPERIMENTAL RESEULTS

In this section, we compare our proposed algorithm with the four types of matrix completion schemes in the literature. The first type of competing algorithms based on nuclear norm includes SVT [22], FPCA [23], and APGL [24]. The second one is based on projection onto nonconvex constraint sets, including SVP [25], NIHT [26], and AP [27]. Then, the third one performs matrix factorization directly to satisfy the low-rank property, and examples are LMaFit [32], OptSpace [34], and PAM [36]. The last category of competing algorithm makes use of rank-one matrix completion and no matrix rank knowledge is required, including L1MC-RF [38], OR1MP [40] and EOR1MP [41]. We perform comparisons using synthetic data as well as real images, and all simulations are run on a computer with 3.2 GHz CPU and 16 GB memory.

In our experiments, the recommended setting of parameters for the competing algorithms is adopted. If suggested parameters are not available, the parameters are selected as the values which possess the best performance among numerous trials.

In our proposed algorithm, we set $I_m = 100$, $\eta = 10^{-6}$ and $\zeta = 0.5$ if the data contain noise; otherwise, $\zeta = 0.001$.

### A. Results of Synthetic Random Data

The experimental setting strategy in [40], [41] is adopted, where the authors construct a synthetic matrix by first generating a random matrix whose entries satisfy the standard normal distribution, and then setting its $i$th singular value to $c^{r-i}$ for $i = 1, 2, \cdots r$, where $c = 2$. In our experiments, the dimensions of $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ are $m = 500$ and $n = 300$, and its rank $r$ is 10. Besides, the root mean square error (RMSE) is employed as the performance measure, which is expressed as:

$$\text{RMSE} = \sqrt{\text{E}\left\{\frac{\|\boldsymbol{X} - \boldsymbol{M}\|_F^2}{mn}\right\}} \tag{29}$$

and it is calculated based on 100 independent runs.

Fig. 1 plots the relationship between RMSE and percentage of observations without noise. It is seen that the PAM can obtain the optimal solution if the percentage of observations is larger than $20\%$, and the recovery error of the proposed algorithm is better than that of OR1MP, which aligns with (28). The results of the remaining algorithms have a poor performance compared with the PAM and proposed algorithm. While the SVP algorithm diverges when the percentage of observations is less than $40\%$.

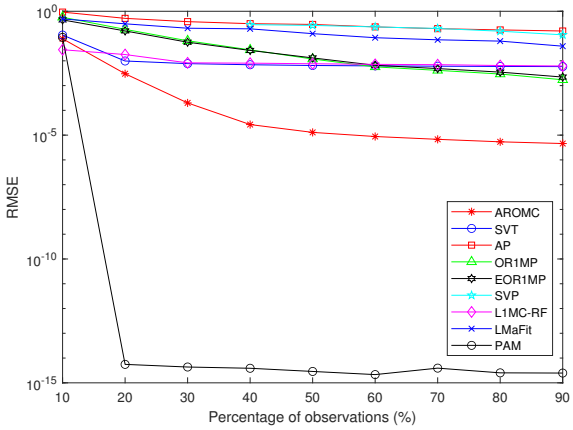



Fig. 1: RMSE versus percentage of observations in noise-free case

We then consider the noisy scenario where $\boldsymbol{N} \in \mathbb{R}^{m\times n}$ which contains zero-mean white Gaussian variables is added to the low-rank $\boldsymbol{X}$, and the signal-to-noise ratio (SNR) is defined as:

$$\text{SNR} = \frac{\|\boldsymbol{X}\|_\Omega^2}{\|\boldsymbol{N}\|_\Omega^2} \tag{30}$$

Fig. 2 plots the RMSE versus SNR for different algorithms when the percentage of observations is fixed at $50\%$. It is seen that AROMC has the minimum RMSE for all SNRs, compared with the other rank-one matrix completion schemes, namely, OR1MP, EOR1MP and L1MC-RF, and the remaining algorithms. Surprisingly, the AROMC can even yield a smaller RMSE than the PAM. Prior to explaining the superiority of AROMC over PAM in recovery error, it is necessary to know the fact that in the absence of noise and the rank of $\boldsymbol{X}$ is known, the PAM attains the global minimum recovery error in terms of RMSE. However, when $\boldsymbol{X}$ is contaminated by noise, PAM is only the optimal rank-$r$ approximation of $\boldsymbol{X}$ in the LS sense, which may be not the globally optimal approximation of $\boldsymbol{X}$. That is to say, it is possible that another rank-$r_0$ matrix with $r_0 < r$ is the best approximation in the presence of noise. This is because the AROMC will not take the signal components with stronger noise, and terminates before reaching the true matrix rank. Therefore, when the matrix is too noisy, the proposed method outperforms the PAM. Nevertheless, the gap of the RMSE between them decreases as the SNR increases.

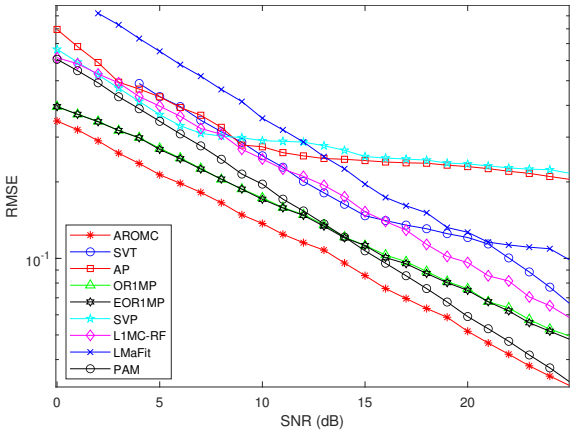



Fig. 2: RMSE versus SNR

Fig. 3 plots RMSE versus percentage of observations at SNR = 10 dB. Firstly, AROMC is significantly superior to all other algorithms for different percentages of observations. Except L1MC-RF, the RMSE of all algorithms decreases with the increase of the observation percentage. For L1MC-RF, it just updates the missing entries in each iteration and keeps the observed entries unchanged, so in the noisy case, the RMSE may increase although the percentage of observations increases. Furthermore, the PAM and LMaFit cannot converge if the percentage of observations is less than $10\%$.

TABLE I: Estimated rank by different algorithms

| Real rank | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|
| AROMC | 6 | 8 | 10 | 12 | 14 | 16 |
| SVT [22] | 4 | 6 | 8 | 10 | 12 | 118 |
| OR1MP [40] | 50 | 50 | 50 | 50 | 50 | 50 |
| EOR1MP [41] | 50 | 50 | 50 | 50 | 50 | 50 |
| SVP [25] | 12 | 16 | 20 | 24 | 28 | 32 |
| L1MC-RF [38] | 5 | 7 | 8 | 9 | 14 | 38 |

Table I tabulates the rank estimation results without noise. Note that the AP, LMaFit and PAM are not involved because they cannot estimate the matrix rank. It is seen that AROMC can find the rank accurately and the reason why the results of OR1MP and EOR1MP are both 50, is that their maximum outer iteration values are set to 50. In addition, the relationship between the matrix dimensions and RMSE as well as runtime are investigated. We set $m = n$, and $r = m/50$ [64]. Table II tabulates the results in the presence of white Gaussian

TABLE II: Average RMSE and runtime comparison.

| Size | 300 | | 400 | | 500 | | 600 | |
|---|---|---|---|---|---|---|---|---|
| | RMSE | Runtime | RMSE | Runtime | RMSE | Runtime | RMSE | Runtime |
| AROMC | **0.012** | 3.28 | **0.031** | 4.61 | **0.091** | 5.63 | **0.28** | 6.77 |
| SVT [22] | 0.013 | 11.30 | 0.094 | 22.23 | 0.31 | 41.61 | 0.85 | 75.90 |
| AP [27] | 0.0055 | 5.75 | 0.049 | 11.97 | 0.51 | 17.01 | 0.89 | 26.69 |
| OR1MP [40] | 0.016 | 2.25 | 0.042 | 8.83 | 0.12 | 11.26 | 0.37 | 16.12 |
| EOR1MP [41] | 0.016 | 1.65 | 0.041 | 8.41 | 0.12 | 10.44 | 0.36 | 14.55 |
| SVP [25] | 0.021 | 45.28 | 0.065 | 64.32 | 0.21 | 89.48 | 0.70 | 117.8 |
| L1MC-RF [38] | 0.034 | 1.51 | 0.080 | 14.47 | 0.26 | 29.50 | 0.84 | 49.58 |
| LMaFit [32] | 0.030 | **0.45** | 0.082 | **0.86** | 0.25 | **1.21** | 0.79 | **2.30** |
| PAM [36] | 0.013 | 0.81 | 0.040 | 1.27 | 0.13 | 2.11 | 0.44 | 3.25 |

noise at SNR = 10 dB. It is seen that the AROMC achieves better recovery performance, but it requires longer runtime than those of LMaFit and PAM. However, LMaFit and PAM need to know the rank. Fig. 4 compares the convergence of AROMC, OR1MP and EOR1MP. In order to obtain a smooth convergence curve, we modify two parameter settings, namely, $r = 30$ and $c = 1.2$. Both OR1MP and EOR1MP have been verified to have a linear convergence rate in [40], [41], and the proposed algorithm also converges linearly. It is shown that AROMC has a smaller RMSE than both OR1MP and EOR1MP. Moreover, the impact of the tolerance parameter $\zeta$ on the proposed algorithm is investigated in Appendix H.

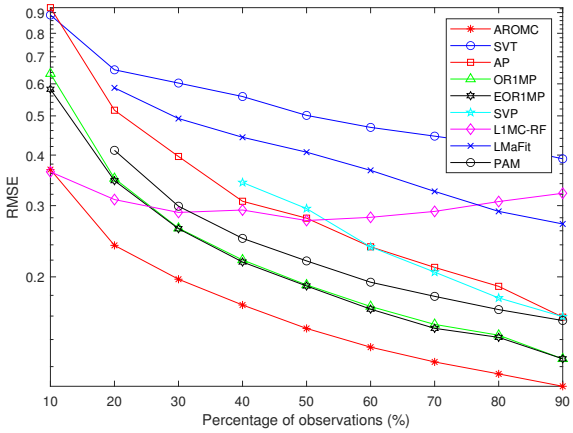



Fig. 3: RMSE versus percentage of observations at SNR = 10 dB

## B. Image Recovery

The proposed algorithm is applied on image inpainting where images in [65] and the ZJU dataset [49] are tested. In the experiments, the images are first converted into gray-scale to obtain the direct matrix forms, and we consider three different masks, namely, random mask, text mask and block mask, to generate the corrupted images. To measure the performance of the recovered images, peak signal-to-noise-ratio (PSNR) is adopted and we directly use the command 'psnr ( recoverd, original )' in MATLAB for its calculation. Given an incomplete image, we take a similar strategy in [49]

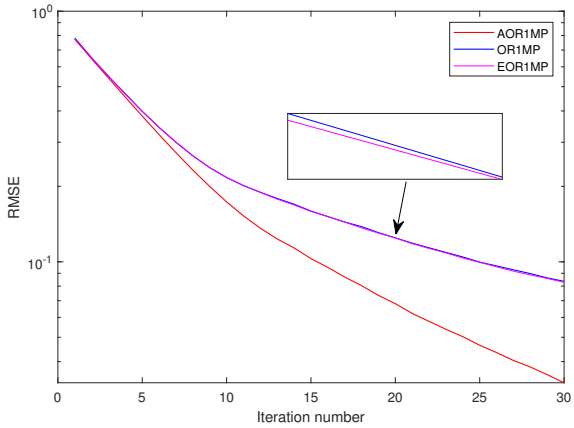



Fig. 4: Comparison of convergence between AROMC, OR1MP and EOR1MP

to determine the image rank, i.e., we try all values of rank and choose the one that yields the largest PSNR as the true rank.

*1) Random Mask:* Firstly, we address a relatively easy matrix completion problem where the missing data are randomly distributed. During the experiments, we first cover 50% pixels of the image, and do not add any noise. Then, the recovery performance of different algorithms is compared in terms of PSNR and the results are shown in Fig. 5. It is observed that the AROMC achieves much higher PSNR values compared with the other algorithms. In addition, we conduct experiments when the image is contaminated by zero-mean white Gaussian noise. Fig. 6 plots PSNR versus SNR for different algorithms with 50% of observations. It is seen that the proposed algorithm has the best performance for all SNRs, and OR1MP as well as EOR1MP obtain a better performance compared with other methods. Furthermore, we study the relationship between recovery performance and observation ratio, and the results are shown in Fig. 7, which plots the PSNR versus percentage of observations at SNR = 10 dB. Again, AROMC attains the best recovery performance among all competing schemes. Finally, the images from the ZJU dataset shown in Fig. 8, are tested, and the results are tabulated in Table III. We easily observe that the proposed algorithm attains the best recovery performance at SNR = 10 dB.

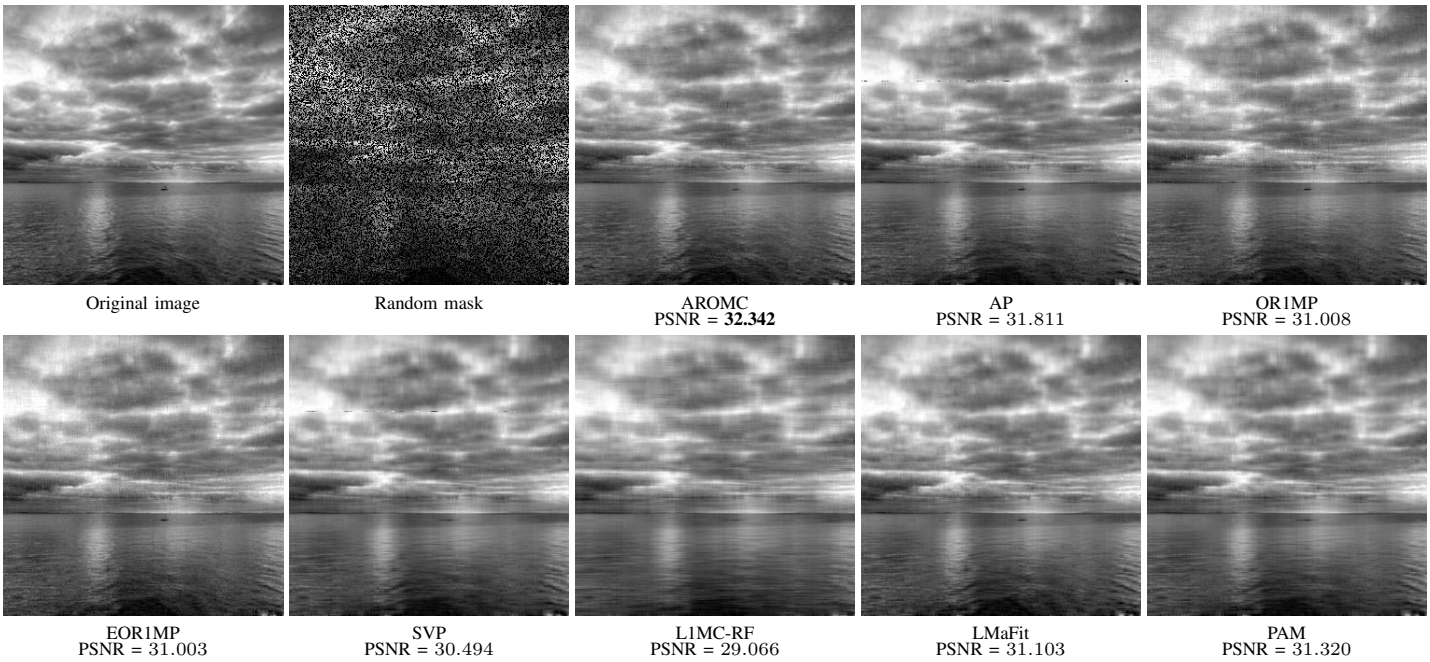



Fig. 5: Image recovery results with random mask by different algorithms

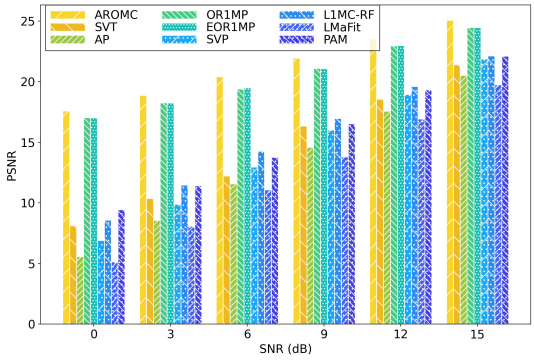



Fig. 6: PSNR versus SNR for random mask

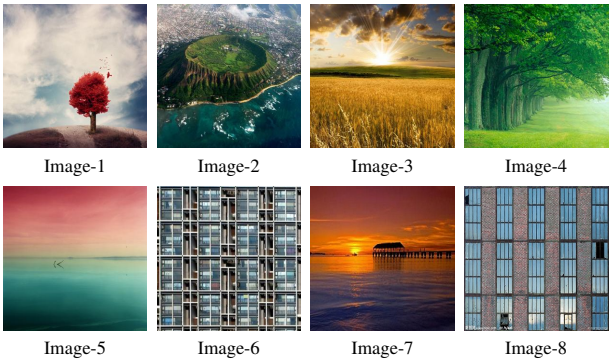



Fig. 8: Images from ZJU dataset

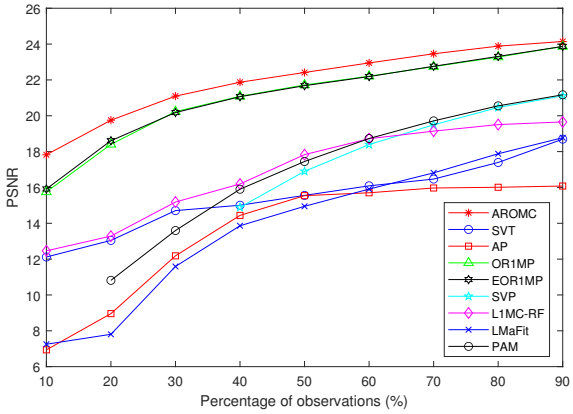



Fig. 7: PSNR versus percentage of observations at SNR = 10 dB

*2) Text Mask:* Text mask is another type of matrix completion problem which is not easy to tackle since the corresponding missing entries are not randomly distributed and some important texture information of images may be covered by text. As shown in Fig. 9, the image is covered by "Matrix", "completion" and "2021". We first conduct matrix completion experiments without noise, and the results are shown in Fig. 9, while Fig. 11 plots the PSNR versus SNR for different algorithms. It is seen that the proposed algorithm has superior recovery performance over other matrix completion methods. Besides, the same mask is utilized to cover the images in Fig. 8 and the restoration results by different methods are also included in Table III. It is seen that our algorithm outperforms the competing methods.

*3) Block Mask:* In some situations, images are contaminated by some large missing blocks. In our experiments, the image is covered by blocks of different shapes, making the recovery of missing entries challenging. As shown in Fig. 10, the image is contaminated by 12 different shapes of blocks, together with the image recovery results. It is observed that AROMC achieves the best image completion results than other matrix completion algorithms. In addition, we conduct experiments for the noisy case, and the results are shown in Fig. 12. We again see the superiority of our proposed AROMC.

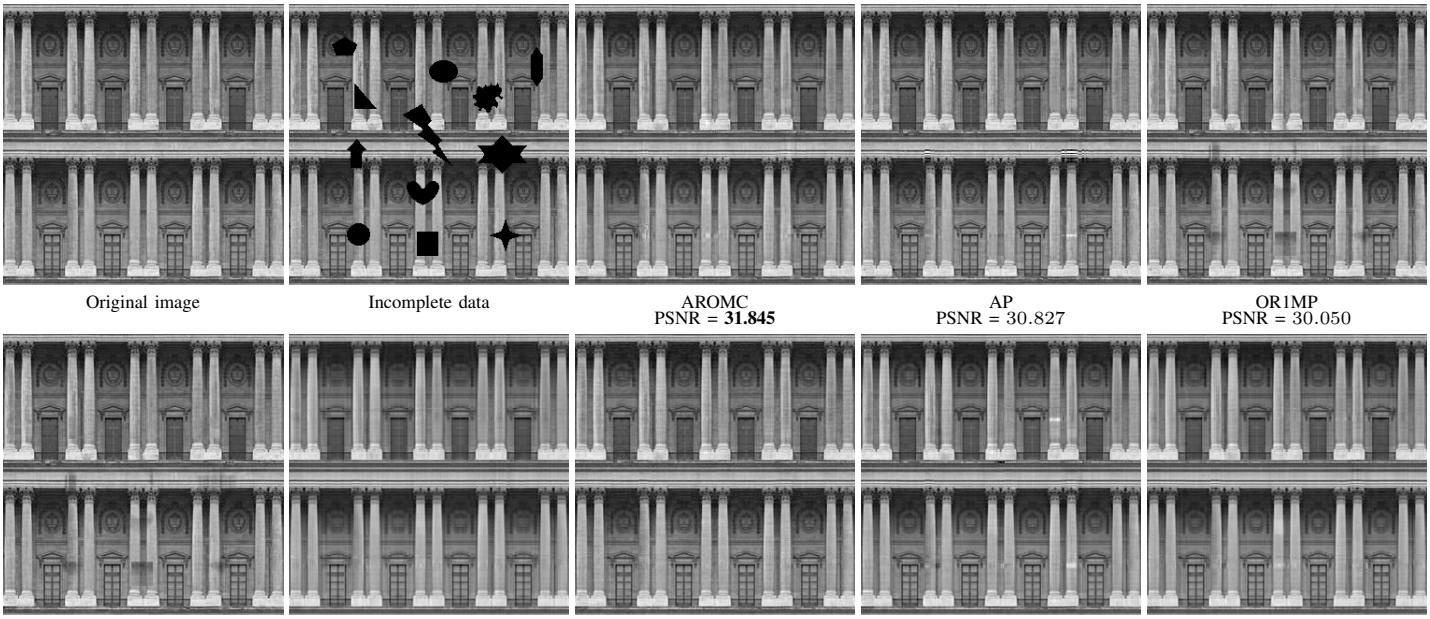
This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3250651




Original image | Incomplete data | AROMC PSNR = **32.269** | AP PSNR = 30.096 | OR1MP PSNR = 30.927

EOR1MP PSNR = 30.850 | SVP PSNR = 28.374 | L1MC-RF PSNR = 31.944 | LMaFit PSNR = 27.936 | PAM PSNR = 28.969

Fig. 9: Image recovery results with text mask by different algorithms



Original image | Incomplete data | AROMC PSNR = **31.845** | AP PSNR = 30.827 | OR1MP PSNR = 30.050

EOR1MP PSNR = 30.007 | SVP PSNR = 27.849 | L1MC-RF PSNR = 30.329 | LMaFit PSNR = 29.691 | PAM PSNR = 28.458

Fig. 10: Image recovery results with block mask by different algorithms

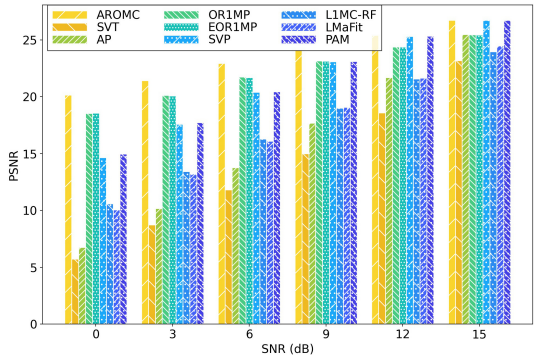



Fig. 11: PSNR versus SNR for text mask

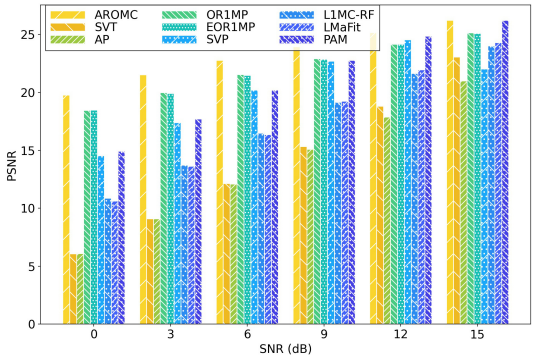



Fig. 12: PSNR versus SNR for block mask

TABLE III: Image inpainting performance comparison using ZJU dataset.

| | | AROMC | SVT [22] | AP [27] | OR1MP [40] | EOR1MP [41] | SVP [25] | L1MC-RF [38] | LMaFit [32] | PAM [36] |
|---|---|---|---|---|---|---|---|---|---|---|
| | Image-1 | **21.44** | 17.91 | 14.23 | 20.54 | 20.49 | 16.61 | 15.95 | 13.81 | 16.99 |
| | Image-2 | **20.00** | 18.49 | 17.48 | 19.77 | 19.76 | 18.07 | 18.63 | 16.41 | 18.35 |
| | Image-3 | **20.84** | 19.16 | 15.28 | 20.31 | 20.30 | 16.32 | 17.33 | 14.60 | 17.23 |
| Random | Image-4 | **21.81** | 19.83 | 16.99 | 21.37 | 21.39 | 18.17 | 18.61 | 16.33 | 18.85 |
| Mask | Image-5 | **28.55** | 20.26 | 15.54 | 26.37 | 26.17 | 17.99 | 17.79 | 15.38 | 19.00 |
| | Image-6 | **19.50** | 18.42 | 15.01 | 19.07 | 19.03 | 15.79 | 17.19 | 14.41 | 16.49 |
| | Image-7 | **26.28** | 20.41 | 19.81 | 25.53 | 25.53 | 21.99 | 22.88 | 19.01 | 22.80 |
| | Image-8 | **22.24** | 20.09 | 17.18 | 21.68 | 21.64 | 19.91 | 19.07 | 16.75 | 20.13 |
| | Image-1 | **22.35** | 13.70 | 10.83 | 19.94 | 19.85 | 18.96 | 13.88 | 16.30 | 19.41 |
| | Image-2 | **20.56** | 18.91 | 16.54 | 20.25 | 20.22 | 19.05 | 18.44 | 17.38 | 20.14 |
| | Image-3 | **21.44** | 16.10 | 13.46 | 20.16 | 20.10 | 20.15 | 16.16 | 18.51 | 20.23 |
| Text | Image-4 | **22.28** | 17.79 | 15.94 | 21.57 | 21.55 | 20.29 | 18.98 | 18.17 | 20.84 |
| Mask | Image-5 | **30.25** | 15.26 | 11.92 | 23.10 | 22.83 | 18.93 | 15.27 | 16.23 | 20.80 |
| | Image-6 | **20.86** | 16.28 | 13.97 | 19.75 | 19.70 | 20.24 | 16.54 | 17.93 | 20.44 |
| | Image-7 | **25.81** | 20.01 | 16.35 | 21.50 | 21.61 | 19.76 | 19.46 | 18.51 | 22.78 |
| | Image-8 | **22.38** | 16.88 | 14.15 | 19.92 | 20.00 | 20.60 | 17.16 | 17.68 | 21.06 |
| | Image-1 | **22.55** | 13.38 | 13.17 | 21.71 | 21.76 | 19.70 | 17.56 | 17.50 | 19.96 |
| | Image-2 | **20.17** | 18.98 | 15.55 | 21.09 | 21.08 | 16.04 | 20.05 | 16.09 | 20.37 |
| | Image-3 | **21.74** | 15.89 | 15.43 | 21.38 | 21.41 | 20.00 | 19.05 | 18.33 | 20.26 |
| Block | Image-4 | **22.48** | 17.53 | 16.95 | 21.68 | 21.67 | 21.32 | 20.24 | 19.68 | 21.40 |
| Mask | Image-5 | **30.72** | 14.93 | 14.58 | 27.69 | 27.67 | 20.22 | 19.55 | 17.83 | 21.29 |
| | Image-6 | **21.08** | 16.10 | 15.33 | 20.74 | 20.73 | 19.05 | 18.92 | 16.94 | 20.10 |
| | Image-7 | **27.03** | 20.06 | 16.26 | 25.95 | 25.96 | 24.76 | 20.97 | 22.84 | 25.27 |
| | Image-8 | **23.20** | 16.61 | 12.35 | 22.26 | 22.32 | 22.43 | 18.49 | 20.56 | 22.48 |

Table III tabulates the results tested on the ZJU dataset and it is seen that our method has the best restoration results, compared with the remaining methods.

## VI. Conclusion

We have developed a novel low-rank matrix completion algorithm, referred to as adaptive rank-one matrix completion (AROMC). By utilizing the sum of outer product matrix representation, it aims to alternately find a low-dimensional vector subspace and learn the weights of those basis vectors by performing an orthogonal projection of the observed matrix onto the subspace. Compared with employing fixed rank-one basis matrices in OR1MP, the rank-one basis matrices, constructed by the basis vectors and their corresponding weight vectors, are adjustable. Another advantage of the AROMC is that it only needs termination conditions without knowing the rank information, which is very easy to perform. We also prove that the proposed algorithm has a linear convergence rate, and a lower recovery error than the OR1MP. Based on extensive numerical examples using synthetic data and real images, the superiority of the AROMC over other competing algorithms in terms of RMSE, and/or PSNR, is demonstrated.

## References


[1] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.

[2] Y. Chen and Y. Chi, "Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 14–31, Jul. 2018.

[3] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Aug. 2019.

[4] A. Sobral, T. Bouwmans, and E.-H. Zahzah, "Comparison of matrix completion algorithms for background initialization in videos," in *Proc. Int. Conf. Image Anal. Process.*, Aug. 2015, pp. 510–518.

[5] Z. Li, Z. Hu, F. Nie, R. Wang, and X. Li, "Matrix completion with column outliers and sparse noise," *Inf. Sci.*, vol. 573, pp. 125–140, Sep. 2021.

[6] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining Anal.*, vol. 1, no. 4, pp. 308–323, Jul. 2018.

[7] S. Mao, L. Xiong, L. Jiao, T. Feng, and S.-K. Yeung, "A novel Riemannian metric based on Riemannian structure and scaling information for fixed low-rank matrix completion," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1299–1312, May 2017.

[8] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3050–3061, Jun. 2015.

[9] Z.-Y. Wang, H. C. So, and Z. Liu. "Fast and robust rank-one matrix completion via maximum correntropy criterion and half-quadratic optimization." *Signal Process.*, vol. 198, pp. 108580, Sep. 2022.

[10] F. Cao, M. Cai, and Y. Tan, "Image interpolation via low-rank matrix completion and recovery," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1261–1270, Aug. 2015.

[11] A. Argyriou, T. Evgeniou and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.,* vol. 73, no. 3, pp. 243–272, Jan. 2008.

[12] J. Fan and T. Chow, "Deep learning based matrix completion," *Neurocomputing,* vol. 266, pp. 540–549, Nov. 2017.

[13] F. Taherkhani, H. Kazemi, and N. M. Nasrabadi, "Matrix completion for graph-based deep semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, Hawaii, USA, Jan. 2019, pp. 5058–5065.

[14] S. Ahmadi and M. Rezghi, "Generalized low-rank approximation of matrices based on multiple transformation pairs," *Pattern Recognit.*, vol. 108, p. 107545, 2020.

[15] T. Xie, S. Li, L. Fang, and L. Liu, "Tensor completion via nonlocal low-rank regularization," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2344–2354, Jun. 2019.

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

[17] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Dept. Elect. Eng, Stanford Univ, California, USA, 2002.

[18] E. J. Candès and Y. Plan, "Matrix completion with noise," *Proc. IEEE*, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[19] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, Aug. 2010.

[20] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optim. Methods. Softw.*, vol. 11, no. 1-4, pp. 625–653, Jan. 2008.

[21] Z. Liu, and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1235–1256, Aug. 2009.

[22] 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.

[23] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Program. Ser. A*, vol. 128, no. 1, pp. 321–353, Jun. 2011.

[24] 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.

[25] P. Jain, R. Meka, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 937–945.

[26] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. S104–S125, Oct. 2013.

[27] X. Jiang, Z. Zhong, X. Liu, and H. C. So, "Robust matrix completion via alternating projection," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 579–583, May 2017.

[28] X. Su, Y. Wang, X. Kang, and R. Tao, "Nonconvex truncated nuclear norm minimization based on adaptive bisection method," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3159–3172, Nov. 2019.

[29] Z. -Y. Wang, X. P. Li and H. C. So, "Robust matrix completion based on factorization and truncated-quadratic loss function," in *IEEE Trans. Circuits Syst. Video Technol.*, 2022, doi: 10.1109/TCSVT.2022.3214583.

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

[31] 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.

[32] 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.

[33] Z. Wang, M. Liu, M. Dong, and L. Wu, "Riemannian alternative matrix completion for image-based flame recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2490–2503, Nov. 2017.

[34] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[35] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, Palo Alto, USA, Jun. 2013, pp. 665–674.

[36] Q. Li, Z. Zhu, and G. Tang, "Alternating minimizations converge to second-order optimal solutions," in *Proc. Int. Conf. Mach. Learn.*, California, USA, Jun. 2019, pp. 3935–3943.

[37] S. Li, Q. Li, Z. Zhu, G. Tang, and M. B. Wakin, "The global geometry of centralized and distributed low-rank matrix recovery without regularization," *IEEE Signal Process. Lett.*, vol. 27, pp. 1400–1404, Jul. 2020.

[38] Q. Shi, H. Lu, and Y.-M. Cheung, "Rank-one matrix completion with automatic rank estimation via L1-norm regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 4744–4757, Oct. 2018.

[39] K. Zhong, P. Jain, and I. S. Dhillon, "Efficient matrix sensing using rank-1 Gaussian measurements," in *Proc. Int. Conf. Algorithmic Learn. Theory*, Banff, AB, Canada, Oct. 2015, pp. 3–18.

[40] Z. Wang, M.-J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye, "Rank-one matrix pursuit for matrix completion," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 91–99.

[41] Z. Wang, M.-J. Lai, Z. Lu, W. Fan, H. Davulcu, and J. Ye, "Orthogonal rank-one matrix pursuit for low-rank matrix completion," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. A488–A514, 2015.

[42] Y. Hu, C. Zhao, D. Cai, X. He, and X. Li, "Atom decomposition with adaptive basis selection strategy for matrix completion," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 3, Jun. 2016, Art. no. 43.

[43] X. P. Li, Q. Liu, and H. C. So, "Rank-one matrix approximation with $\ell_p$-norm for image inpainting," *IEEE Signal Process. Lett.*, vol. 27, pp. 680–684, Apr. 2020.

[44] X. Jiang, A. M. Zoubir, and X. Liu, "Robust matrix completion via $\ell_p$-greedy pursuits," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 5984–5988.

[45] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol.53, no. 12, pp. 4655–4666, Dec. 2007.

[46] E. Liu and V. N. Temlyakov, "The orthogonal super greedy algorithm and applications in compressed sensing," *IEEE Trans. Inf. Theory*, vol.58, no. 4, pp. 2040–2047, Apr. 2012.

[47] J. Fan and T. W. Chow, "Matrix completion by least-square, low-rank, and sparse self-representations," *Pattern Recognit.*, vol. 71, pp. 290–305, Nov. 2017.

[48] Z. Allen-Zhu and Y. Li, "LazySVD: Even faster SVD decomposition yet without agonizing pain," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 974–982.

[49] 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.

[50] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5040–5053, Aug. 2014.

[51] G. H. Golub, A. Hoffman, and G. W. Stewart, "A generalization of the Eckart-Young-Mirsky matrix approximation theorem," *Linear Algebra Appl.*, vols. 88–89, pp. 317–327, Apr. 1987.

[52] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936.

[53] Y.-L. Yu and D. Schuurmans, "Rank/norm regularization with closed-form solutions: Application to subspace clustering." [Online]. Available: https://arxiv.org/abs/1202.3772

[54] T. G. Kolda, "Orthogonal tensor decompositions," *SIAM J. Matrix Anal. Appl.*, vol. 23, no. 1, pp. 243–255, 2001.

[55] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.

[56] Q. Li, Z. Zhu, and G. Tang, "The non-convex geometry of low-rank matrix optimization," *Inf. Inference*, vol 8, no 1, pp. 51–96, Mar. 2019.

[57] S. Li, G. Tang, and M. B. Wakin, "The landscape of non-convex empirical risk with degenerate population risk," in *Proc. Advances Neural Inform. Process. Syst.*, Vancouver, Canada, Jul. 2019, pp. 3502–3512.

[58] Q. Li, Z. Zhu, and G. Tang, "Geometry of factored nuclear norm regularization," arXiv preprint arXiv:1704.01265, 2017.

[59] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "The global optimization geometry of low-rank matrix optimization," arXiv preprint arXiv:1703.01256, 2017.

[60] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low-rank problems: A unified geometric analysis," in *Proc. Int. Conf. Mach. Learn.*, Sydney, Australia, Aug. 2017, pp. 1233–1242.

[61] Z. Zhu, D. Soudry, Y. C. Eldar, and M. B. Wakin, "The global optimization geometry of shallow linear neural networks," *J. Math. Imag. Vis.*, vol. 62, no. 3, pp. 1–14, May 2019.

[62] D. Park, A. Kyrillidis, C. Carmanis, and S. Sanghavi, "Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach," in *Proc. Artif. Intell. Statist.*, Fort Lauderdale, USA, Apr. 2017, pp. 65–74.

[63] F. Ruppel, M. Muma, and A. M. Zoubir, "Globally optimal robust matrix completion based on M-estimation," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, 2020, pp. 1–6.
[64] L. Zhao, P. Babu, and D. Palomar, "Efficient algorithms on robust low-rank matrix completion against outliers," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4767–4780, Sep. 2016.
[65] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, Sep. 2020.

**Zhi-Yong Wang** received the B.S. degree from the Zhengzhou University, Zhengzhou, China in Mechanical Engineering and the from the Xi'an Jiaotong University, Xi'an, China in Mechanical Engineering, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in 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.

**Xiao Peng Li** received the B.Eng. degree in electronic science & technology from Yanshan University, Qinhuangdao, China, in 2015, M.Sc. degree with distinction in electronic information engineering, and Ph.D. degree in electrical engineering from City University of Hong Kong, Hong Kong SAR, China, in 2018 and 2022, respectively. From 2018 to 2019, he was a Research Assistant with the Department of Information Engineering, Shenzhen University, Shenzhen, China. He is currently a Postdoctoral Fellow in the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China.

His research interests include optimization methods, machine learning, sparse recovery, matrix processing, tensor processing, and their applications on target estimation, image recovery, video restoration, hyperspectral unmixing as well as stock market analysis.

**Hing Cheung So** (S'90–M'95–SM'07-F'15) was born in Hong Kong. He received the B.Eng. degree from the City University of Hong Kong and the Ph.D. degree from The Chinese University of Hong Kong, both in electronic engineering, in 1990 and 1995, respectively. 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. 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 (2014–2017), IEEE Transactions on Signal Processing (2010–2014), Signal Processing (2010–), and Digital Signal Processing (2011–). He was also Lead Guest Editor for IEEE Journal of Selected Topics in Signal Processing, special issue on "Advances in Time/Frequency Modulated Array Signal Processing" in 2017. In addition, he was an elected member in Signal Processing Theory and Methods Technical Committee (2011–2016) of the IEEE Signal Processing Society where he was chair in the awards subcommittee (2015–2016).

**Abdelhak M. Zoubir** has been a Professor of signal processing and Head of the Signal Processing Group with Technische Universität Darmstadt, Darmstadt, Germany, since 2003. 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 has authored or coauthored more than 500 journal and conference papers on the above areas. He was the recipient of several Best Paper awards, and the 2018 IEEE Leo L. Beranek Meritorious Service Award. He was a 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 Guest Editor of various journals, and the Editor-in-Chief of the IEEE Signal Processing Magazine (2012–2014). Dr. Zoubir was the Chair (2010–2011) of the IEEE Signal Processing Society (SPS) Technical Committee Signal Processing Theory and Methods (SPTM) and was on the Board of Governors of the IEEE SPS as a Member-at-Large (2015–2017). He was the President of the European Association of Signal Processing from 2017 to 2018. He is a Fellow of the IEEE and an IEEE Distinguished Lecturer (Class 2010–2011).