



# A generalized tri-factorization method for accurate matrix completion

Qing Liu<sup>1,2</sup> · Hao Wu<sup>1</sup> · Yu Zong<sup>1</sup> · Zheng-Yu Liu<sup>1</sup>

Received: 3 August 2023 / Accepted: 14 July 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

## Abstract

To improve the speeds of the traditional nuclear norm minimization methods, a fast tri-factorization method (FTF) was recently proposed for matrix completion, and it received widespread attention in the fields of machine learning, image processing and signal processing. However, its low convergence accuracy became increasingly obvious, limiting its further application. To enhance the accuracy of FTF, a generalized tri-factorization method (GTF) is proposed in this paper. In GTF, the nuclear norm minimization model of FTF is improved to a novel  $L_{1,p}$  ( $0 < p < 2$ ) norm minimization model that can be optimized very efficiently by using QR decomposition. Since the  $L_{1,p}$  norm is a tighter relaxation of the rank function than the nuclear norm, the GTF method is much more accurate than the traditional methods. The experimental results demonstrate that GTF is more accurate and faster than the state-of-the-art methods.

**Keywords** Matrix completion · Fast tri-factorization ·  $L_{1,1}$  · QR decomposition

## 1 Introduction

Low-rank matrix completion is a technique of data recovery that completes the missing elements in a matrix and has received a wide range of interest from researchers in the subfields of machine learning, such as image processing [1–3], data mining [4, 5], face recognition [6, 7] and pattern recognition [8–10]. Since the rank function is discrete, the earliest rank minimization problems [11, 12] for matrix completion are NP-hard and difficult to optimize. Hence, the nuclear norm, which is the summation of singular values, is proposed as a convex relaxation of the rank function [13] for improving the speeds of traditional rank minimization methods. The representative methods of nuclear norm

minimization including singular value thresholding (SVT) [14] and approximated proximal gradient search [15]. These methods are much faster and more accurate than the rank minimization-based methods. More recent research findings [16–18] show that different singular values contribute differently to the recovery results. It is not reasonable to assign the same weight to each singular value [16]. Therefore, some weighed nuclear norm minimization methods have been investigated, such as a Schatten-p norm minimization method [17], a Schatten Capped P Norm minimization method [5], a truncated nuclear norm minimization method (TNNR) [18], and an iteratively reweighted nuclear norm minimization method (IRNN) [19]. The weighted nuclear norm minimization methods adopt different weighting strategies to adjust the singular value thresholds and thus improve their convergence accuracy. It is reported that these methods are much more accurate than the nuclear norm minimization-based methods, such as SVT. Since Singular Value Decomposition (SVD) is required in the methods based on nuclear norm minimization or based on weighted nuclear norm minimization, these methods have a common shortcoming, that is, they are all computationally demanding, especially for the applications of moving object detection [28], background subtraction [29], recommendation system [30] and so on.

To reduce the computational cost of methods using SVD, some methods based on matrix factorization have been

✉ Yu Zong  
zongyu\_wxc@126.com

✉ Zheng-Yu Liu  
lzy\_wahu@126.com

Qing Liu  
clyqiq2008@126.com

Hao Wu  
wuhao\_wxc\_@126.com

<sup>1</sup> School of Electronics and Information Engineering, West Anhui University, Lu'an 237012, China

<sup>2</sup> School of Mathematics and Big Data, Anhui University of Science and Technology, Huai'an 232001, China

developed, such as the low rank matrix fitting (LMAFit) [21], matrix bifactorization (MBF) [22] and fast tri-factorization (FTF) methods [23]. Since the LMAFit method directly replaces the SVD decomposition with the less computationally intensive QR decomposition, it is much faster than the traditional matrix completion methods using SVD. The MBF method can be seen as an improvement of the LMAFit method. It optimizes the nuclear norm of a submatrix whose size is much smaller than the size of the original incomplete matrix. Consequently, the computational cost of SVD decomposition in its updating steps is not large. A benefit of using SVD decomposition, is that it is slightly more accurate than the low-rank matrix fitting method.

The FTF method decomposes the underlying low-rank matrix  $X$  as follows:

$$X = ABC, \quad (1)$$

where the size of  $X$  is  $m \times n$  ( $m \leq n$ ) and the rank of  $X$  is  $r$  ( $0 < r \leq m$ ).  $A$  is a column orthonormal matrix of size  $m \times r$ , and  $C$  is a row orthonormal matrix of size  $r \times n$ .  $B$  ( $B \in R^{r \times r}$ ) is a square matrix. FTF uses QR decomposition instead of SVD to calculate the eigenvectors of  $A$  and  $C$ , to reduce its computational cost.

In fact, the FTF method still has some disadvantages. First, FTF becomes significantly slow when dealing with matrices of complex structures. The main reason is that the size of  $B$  increases, and the SVD decomposition computations, at its updating steps, also increases in such cases. Second, FTF may not be accurate on some complex images. Since FTF is still a nuclear norm minimization method, its convergence accuracy is essentially the same as that of SVT, and is much lower than that of a weighted nuclear norm minimization method, such as the truncated nuclear norm minimization method [18].

To improve the speed and accuracy of FTF, Dr. Liu [20] proposed an improved FTF method based on the  $L_{2,1}$  norm (the summation of Frobenius norms of rows/columns of a matrix) and QR decomposition that can be abbreviated as LNMQR. The LNMQR method is much faster than the FTF, SVT and IRNN methods. And moreover, LNMQR is almost as accurate as the IRNN method [19], because it can converge to a weighted nuclear norm optimization method.

In recent years, the  $L_{1,1}$  norm has been widely used in sparse coding [31], and low rank representation methods [24,25] that always have excellent convergence accuracies. Moreover, the  $L_{1,1}$  norm is easier to calculate and optimize than the  $L_{2,1}$  norm used in LNMQR. Given these advantages, there is every reason to believe that FTF could be modified into a  $L_{1,1}$  norm optimization method.

In this paper, a generalized tri-factorization method based on the  $L_{1,p}$  norm is proposed for accurate matrix completion (GTF). In GTF, a  $L_{1,p}$  norm minimization

model is applied to matrix completion to improve the accuracy of FTF, where the  $L_{1,1}$  norm is a special case of the  $L_{1,p}$  norm. The main contributions of GTF are as follows:

- The optimization model of GTF is a generalized weighted nuclear norm minimization model, which have much better solutions than those of FTF and other weighted nuclear norm minimization methods. The main reason is that the  $L_{1,p}$  norm is a tighter relaxation of matrix rank than the nuclear norm when the parameter  $p \in (0, 1)$ .
- A novel thresholding function is designed for the  $L_{1,p}$  norm used in GTF, which makes GTF more accurate than FTF and the other state-of-the-art methods compared in this paper.
- The GTF method is much faster than the traditional methods using SVD. In GTF, QR decomposition, with a speed approximately 7 times [20] that of SVD, is used as a replacement of SVD to calculate the eigenvectors of incomplete matrices.

## 2 Related work

### A. The Fast Tri-Factorization method (FTF) [23]

Suppose  $M$  is a matrix of size  $m \times n$  with many missing entries, and  $X$  is the underlying low rank matrix. FTF decomposes  $X$  into the form of a product of three submatrices as in Eq. (1). Then, the FTF method recovers the missing entries of  $M$  by optimizing the following minimization model:

$$\min_{A,B,C,X} \|B\|_*, s.t. ABC = X, P_\Omega(X) = P_\Omega(M), \quad (2)$$

where the parameters  $A$ ,  $B$ ,  $C$  and  $X$  are defined in the same way as the corresponding ones in Eq. (1), respectively.  $\|B\|_*$  stands for the nuclear norm of  $B$ .  $\Omega$  is the set of locations for the known entries of  $M$ . For any matrix  $X$ ,  $P_\Omega(X)$  satisfies

$$[P_\Omega(X)]_{ij} = \begin{cases} X_{ij}, & \text{if } (i,j) \in \Omega \\ 0, & \text{otherwise} \end{cases}. \quad (3)$$

In the FTF method, the variables  $A$  and  $C$  can be optimized by using QR decomposition [23], which reduces its computation complexity. The variable  $B$  whose size is  $r \times r$  can be optimized via a singular value thresholding operator [14] by using SVD. Since the FTF method can obtain the singular values of matrix  $B$  directly, it always converge with a higher accuracy than LMAFit. Yet, It has been reported [20] that the FTF method is not as accurate as a weighted nuclear norm minimization based method, such as the IRNN and LNMQR methods.

**B. The  $L_{2,1}$  Norm Minimization method based on QR decomposition (LNMQR) [20]**

In order to improve the convergence accuracy of FTF, a  $L_{2,1}$  norm minimization model has been proposed in the LNMQR method [20] as follows,

$$\min_{A,B,C,X} \|B\|_{w \cdot (2,1)}, s.t. X = ABC, P_{\Omega}(X) = P_{\Omega}(M), \tag{4}$$

where  $\|B\|_{w \cdot (2,1)} = \sum_{i=1}^r w_i \|B_i\|_F$  is the weighted  $L_{2,1}$  norm of  $B$  and  $B_i$  is the  $i$ th row of  $B$ . In fact, the minimization model in Eq. (4) is actually a weighted Frobenius norm minimization model, which indicates the SVD decomposition is not required for optimizing the variable  $B$  in LNMQR. The main computation cost of LNMQR at each iteration is the two times of QR decomposition for updating  $A$  and  $C$ . Since the computation cost of QR is much smaller than that of SVD, LNMQR is much faster than the methods using SVD, such as FTF, SVT and IRNN. And moreover, the LNMQR method is much more accurate than FTF, because it can converge to a weighted nuclear norm optimization method. In 2023, Dr. Liu has proposed an improved version of LNMQR, i.e., a Fast Matrix Bi-Factorization method [27], in which only one time of QR decomposition is performed. Hence, FMBF slightly faster than LNMQR. Nevertheless, LNMQR still has an important weakness, i.e., it cannot perform successful matrix completion in case of matrices with structured missing blocks. (please see the experimental results in Sect. 4).

**C. The Schatten Capped P Norm minimization method (SCPN) [5]**

In the SCPN method [5], a novel Schatten capped  $p$  norm minimization model is proposed as follows,

$$\min_X \|X\|_{S_{p,\tau}}^p, s.t. P_{\Omega}(X) = P_{\Omega}(M), \tag{5}$$

where  $p > 0, \tau > 0, \|X\|_{S_{p,\tau}}^p$  is the Schatten capped  $p$  norm of  $X$  and  $\|X\|_{S_{p,\tau}}^p = \sum_{i=1}^m \min(\sigma_i, \tau)^p$ .

Since the Schatten capped  $p$  norm is not convex, a complex multipolar function is employed to regulate the weights for singular values. Experimental results shows that the SCPN method performs much better than the traditional weighted nuclear norm minimization methods, such as TNNR and IRNN, especially on images with structured missing blocks. It is reasonable to assume that a matrix norm with  $p$  powers of singular values is a better relaxation of rank function than nuclear norm.

**D. The  $L_{1,1}$  norm minimization used in sparse coding [31]**

In 2013, the  $L_{1,1}$  norm was successfully applied to low rank sparse coding and the optimization model was solved very efficiently without using SVD [31]. The corresponding subproblem of  $L_{1,1}$  norm minimization is rewritten as follows,

$$\min_Z \alpha \|Z\|_{1,1} + \frac{1}{2} \|Z - T\|_F^2, \tag{6}$$

where  $\alpha > 0, Z (Z \in \mathbb{R}^{m \times n})$  is a real matrix,  $\|Z\|_{1,1} = \sum_{i=1}^m \sum_{j=1}^n |Z_{i,j}|$  and  $T (T \in \mathbb{R}^{m \times n})$  is a known temporary matrix. The problem in Eq. (6) can be solved as follows,

$$Z_{i,j} = \text{sign}(T_{i,j}) \cdot \max(T_{i,j} - \alpha, 0). \tag{7}$$

From Eq. (7), it is clear that the computation cost for solving the problem in Eq. (6) is much smaller than calculating a SVD decomposition of  $T$ . In this paper, we will propose a novel matrix norm, i.e., the summation of  $p$  powers of absolute values of entries in each row of a matrix, to improve the convergence accuracy and speed of FTF.

**3 Our proposed method**

**A. A new tri-factorization framework**

To improve the recovery accuracy of the FTF, a generalized tri-factorization method is proposed in this section. Since the FTF method does not consider the noisy information that may be contained in the observations, the matrix  $X$  in Eq. (1) is factorized as follows:

$$X = \frac{1}{\lambda} ABC + E, \tag{8}$$

where the definitions of  $A, B$  and  $C$  are the same as those of the corresponding ones shown in Eq. (1), respectively. The variable  $\lambda (\lambda > 0)$  is a balanced factor that adjusts the weights of the low-rank matrix  $ABC$  and the noise matrix  $E$ , where  $E$  is a noise matrix with a size of  $m \times n$ . Therefore, the entries of  $E$  obey

$$E_{i,j} = \begin{cases} E_{i,j}, & \text{if } E_{i,j} > t \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where  $t$  stands for the threshold for noisy information. More exactly,  $t$  should be predefined according to the tested data. In our experiment of this paper,  $t$  is set to 255, because the proposed method is tested on images whose values of entries are smaller than 255. Hence, it is not difficult to see that the elements with values greater than 255 can be considered noisy information in image recovery.

**B. The proposed generalized tri-factorization model for matrix completion**

In this paper, the FTF method is improved by using a  $L_{1,1}$  norm and a generalized  $L_{1,p}$  norm to further increase its convergence accuracy. Suppose the matrix  $X$  is decomposed into two parts as in Eq. (8) at first. Then, the FTF model can be improved to be

$$\min_{A,B,C,X,E} \|B\|_{1,1} + \frac{\mu}{2} \left\| X - \frac{1}{\lambda} ABC - E \right\|_F^2, \tag{10}$$

*s.t.*  $P_\Omega(X) = P_\Omega(M)$ ,

where  $\mu > 0$ . Inspired by the core idea of a weighted nuclear minimization method [16], i.e., different singular values should be treated differently, the  $L_{1,1}$  norm in Eq. (10) is generalized to

$$\|B\|_{1,p} = \sum_{i=1}^m \|B_i\|_1^p, \tag{11}$$

where  $\|B_i\|_1 = \sum_{j=1}^n |B_{i,j}|$  and  $p(0 < p < 2)$  is the power of the  $L_1$  norms of the rows in  $B$ . By regulating the value of  $p$ , the weight for the  $L_1$  norm of each row in  $B$  can be adjusted appropriately. The reason is that

$$\|B_i\|_1^p = \|B_i\|_1^{p-1} \cdot \|B_i\|_1, \tag{12}$$

From Eq. (12), we see that the weight for  $\|B_i\|_1$  is  $\|B_i\|_1^{p-1}$ . Consequently, the model in Eq. (10) is further improved as follows:

$$\min_{A,B,C,X,E} \|B\|_{1,p} + \frac{\mu}{2} \left\| X - \frac{1}{\lambda} ABC - E \right\|_F^2, \tag{13}$$

*s.t.*  $P_\Omega(X) = P_\Omega(M)$ ,

where the variables in Eq. (13) have been defined in Eqs. (2), (4) and (8). Since the  $L_{1,1}$  norm is a special case of the  $L_{1,p}$  norm when  $p = 1$ , the optimization model in Eq. (13) is a generalized tri-factorization model based on the  $L_{1,p}$  norm, which can be abbreviated as the GTF model.

**C. Optimization of the GTF model**

The optimization model in Eq. (13) can be solved by an alternating direction method, i.e., the variables can be updated, one by one, with the rest fixed. Suppose that  $X_j$ ,  $A_j$ ,  $B_j$ ,  $C_j$  and  $E_j$  are the results of the alternating method at its  $j$ th iteration.

First,  $A_{j+1}$  can be updated by solving the following subproblem:

$$\begin{aligned} A_{j+1} &= \arg \min_A \left\| X_j - \frac{1}{\lambda} AB_j C_j - E_j \right\|_F^2 \\ &= \arg \min_A \left\| \lambda (X_j - E_j) (B_j C_j)^+ - A \right\|_F^2 \\ &= qr \left( (X_j - E_j) C_j^T \right), \end{aligned} \tag{14}$$

where  $(B_j C_j)^+$  is the pseudo inverse matrix of  $B_j C_j$ .

Similarly, it is easy to obtain the updating step for  $C_{j+1}$ , i.e.,

$$\begin{aligned} C_{j+1} &= \arg \min_C \left\| X_j - \frac{1}{\lambda} A_{j+1} B_j C - E_j \right\|_F^2 \\ &= \arg \min_A \left\| \lambda (A_{j+1} B_j)^+ (X_j - E_j) - C \right\|_F^2 \\ &= qr \left( A_{j+1}^T (X_j - E_j) \right). \end{aligned} \tag{15}$$

Second,  $B_{j+1}$  is optimized by solving the following problem:

$$\min_B \frac{1}{\mu} \|B\|_{1,p} + \frac{1}{2} \left\| \lambda (X_j - E_j) - A_{j+1} B C_{j+1} \right\|_F^2. \tag{16}$$

It is easy to see that the problem in Eq. (16) can be rewritten as follows:

$$\min_B \frac{1}{\mu} \|B\|_{1,p} + \frac{1}{2} \|B - T\|_F^2. \tag{17}$$

where  $T = \lambda A_{j+1}^T (X_j - E_j) C_{j+1}^T$  is a temporary variable. Obviously, the problem in Eq. (17) can be rewritten as follows:

$$\min_{B_i} G(B_i), \tag{18}$$

where  $G(B_i) = \sum_{i=1}^r [\frac{1}{\mu} \|B_i\|_1^p + \frac{1}{2} \|B_i - T_i\|_F^2]$ . By letting  $\frac{\partial G}{\partial B_i} = 0$ , we have

$$\frac{p}{\mu} \|B_i\|_1^{p-1} \frac{B_i}{\|B_i\|_1} + B_i - T_i = 0. \tag{19}$$

From Eq. (19), it is suitable to suppose that  $B_i = xT_i$ , where  $x > 0$ . Then, we have

$$\frac{p}{\mu} \cdot x^{p-2} \cdot \|T_i\|_1^{p-2} \cdot xT_i + xT_i - T_i = 0, \tag{20}$$

and Eq. (20) can be reformulated into the following form:

$$x + b \cdot x^{p-1} - 1 = 0, \tag{21}$$

where  $b$  is equal to

$$b = \frac{p}{\mu} \cdot \|T_i\|_1^{p-2}. \tag{22}$$

Since it is difficult to obtain the analytical solution of the equation in (21), a particular solution at  $p = \frac{3}{2}$  is shown as follows:

$$x^{\frac{1}{2}} = \frac{-b + \sqrt{b^2 + 4}}{2}. \tag{23}$$

By varying the value of  $p$ , we can obtain more solutions to the equation in (21). As a consequence, it is suitable to let

$$x = \left[ \frac{-b + \sqrt{b^2 + 4}}{2} \right]^q. \tag{24}$$

where  $q > 0$ . Therefore, the optimal solution to the problem in Eq. (18) is

$$B_{j+1(i)} = \left[ \frac{-b + \sqrt{b^2 + 4}}{2} \right]^q \cdot T_i, \tag{25}$$

where  $i = 1, 2, \dots, r$ ,  $T_i$  has been defined in Eq. (17). To accelerate the convergence of our proposed method,  $\mu_{j+1}$  in Eq. (22) is updated as follows:

$$\mu_{j+1} = \frac{\mu_j}{\rho}, \tag{26}$$

where  $0 < \rho < 1$ . The strategy of accelerating the convergence by updating the value of  $\rho$  in real time is also commonly used in other methods, such as the TNNR and IRNN methods.

Finally,  $E_{j+1}$  and  $X_{j+1}$  can be updated as follows:

$$E_{j+1} = \hat{X} - C_j, \tag{27}$$

$$X_{j+1} = \hat{X} - P_{\Omega}(\hat{X}) + M, \tag{28}$$

where  $\hat{X} = \frac{1}{\lambda} A_{j+1} B_{j+1} C_{j+1}$ . Notably, the variable  $X_{j+1}$  in Eq. (28) contains noisy information, so the output of the proposed method is that  $X_{out} = X_{j+1} - E_{j+1}$ .

**Algorithm 1** The updating steps of GTF method

---

**Input:**  $M, \Omega$ ;  
 1: **Initialization:**  
 2:  $A_0, B_0, C_0$  are randomized matrices;  
 3:  $j = 0, X_j = M, E_j = \text{zeros}(m, n)$ ;  
 4:  $\varepsilon$  is a positive tolerance.  
 5: **Repeat:**  
 6:  $A_{j+1}$  : Eq. (14);  $C_{j+1}$ : Eq. (15);  
 7:  $B_{j+1}$  : Eq. (25);  $\mu_{j+1}$  : Eq. (26);  
 8:  $E_{j+1}$  : Eq. (27);  $X_{j+1}$ : Eq. (28);  
 9:  $j = j + 1$ .  
 10: **Until:**  $\|X_j - X_{j-1}\|_F^2 < \varepsilon$  or  $j > It_{max}$   
**Output:**  $X_{out} = X_j - E_j$ .

---

The proposed generalized tri-factorization matrix completion method is called GTF for convenience, whose updating steps have been summarized in Algorithm 1.

#### D. Convergence analysis

According to the updating steps in Algorithm 1, it is not difficult to see that the proposed GTF model can converge to a weighted nuclear norm minimization model. The main reason is that the updating steps of GTF for  $A$  and  $C$  are consistent with the updating formulas of  $L$  and  $R$  in the LNMQR method [20], respectively. Consequently, the key variable  $B$  in the GTF model will also inevitably converge to a diagonal matrix. In such cases, the  $L_{1,p}$  norms of the rows in  $B$  satisfy

$$\|B_i\|_1^p = \sigma_i(B)^p \tag{29}$$

For any  $p$ , there exists a  $w$  such that

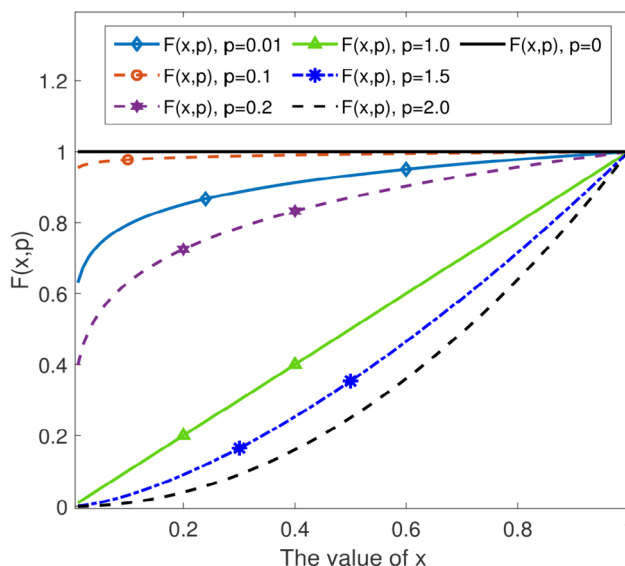
$$\sigma_i(B)^p = w \cdot \sigma_i(B), \tag{30}$$

where  $w(w > 0)$ . Hence, the model in Eq. (30) can be seen as a weighted nuclear norm minimization model.

To obtain a deeper understanding of the optimization model of GTF shown in Eq. (13), the curves of the  $L_{1,p}$  norm with different values of  $p$  are plotted in Fig. 1.

From Fig. 1, we see that the rank function and the nuclear norm are special cases of the  $L_{1,p}$  norm with  $p = 0$  and  $p = 1$ , respectively. Consequently, the proposed  $L_{1,p}$  norm minimization under the framework of matrix tri-factorization can be called a generalized tri-factorization method.

Moreover, the curves in Fig. 1 also show that the  $L_{1,p}$  norm with  $0 < p < 1$  is a tighter relaxation of the rank function than the nuclear norm. Hence, a smaller value for the parameter  $p$  should be set to obtain a higher convergence accuracy because the smaller the value of  $p$  is, the closer the curve of the  $L_{1,p}$  norm is to the rank function.



**Fig. 1** The curves of the  $F(x, p)$  function with different  $p$ , where the  $F(x, p)$  function stands for the  $L_{1,p}$  norm of  $x$

In the next section, a sufficient number of experiments are conducted to verify the effectiveness and efficiency of our proposed GTF method.

## 4 Experimental results and discussion

In this section, our proposed GTF method for matrix completion is compared with the following methods, i.e., the FTF method [23], the Schatten Capped P Norm minimization method (SCPN) [5], the Feature and Nuclear Norm Minimization method (FNNM) [4], the LNMQR method [20] and the  $L_p$  norm minimization inpainting method (LPMP) [26], respectively.

SCPN and FNNM are representative methods of improved weighted nuclear norm minimization and improved nuclear norm minimization, respectively. The LNMQR method, which is a weighted  $L_{2,1}$  norm minimization method that has advantages both on convergence accuracy and CPU time, is highly influential on matrix completion. LPMP is a matrix completion method based on  $L_p$  norm minimization that is closely related to our proposed  $L_{1,p}$  norm minimization based GTF.

The major parameters for GTF that need to be regulated by hand include  $r$ ,  $\mu$ ,  $\rho_0$ ,  $\lambda$ ,  $p$  and  $q$ . The parameter  $r$  is set to one-fourth of the row number of  $X$ ,  $\mu = \frac{1}{8000 \cdot \|M\|_F}$ ,  $\rho_0$  is equal to 0.97 and  $\lambda$  is equal to 0.8. The values of  $p$  and  $q$  need to be adjusted according to the different situations of the test sets. The key parameter  $r$  for FTF is set to one-half of the row number of  $X$  according to the suggestion given by Liu [23]. The maximum iteration numbers for the six methods

are all set to 300 and 500 in Section IV. C and IV. D, respectively.

The GTF method and the compared methods are tested on some commonly used images that are plotted in Fig. 2. Each image in Fig. 2 will be masked by a randomized mask, i.e., 50% of their entries are randomly selected missing entries.

### A. The Convergence of $B_j$

The convergence of  $B_j$  generated by GTF is tested first. Since the analyses in Sect. 3.4 indicate that the submatrix  $B$  should converge to a diagonal matrix, the  $L_1$  norms of the rows in  $B$  should be equal to the singular values. To verify this conjecture, the intermediate iteration results of GTF for  $B_j$  ( $j = 1, 2, \dots, Itmax$ ) are plotted in Fig. 3.

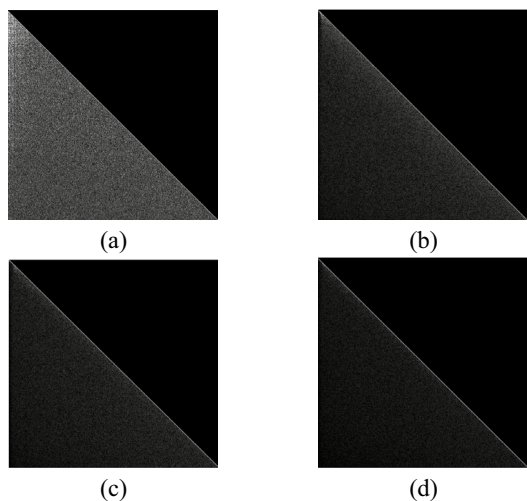
Figure 3 shows that the submatrix  $B$  is a lower triangular matrix when  $j = 1$ , and it can converge to a diagonal matrix at the 30th iteration. This result is highly consistent with the theoretical analysis in Sect. 3.4, which in turn indicates that the GTF method should be much more accurate than FTF and other nuclear norm minimization-based matrix completion methods.

### B. The effects of $p$ and $q$ on GTF

The variables  $p$  and  $q$  are the two most important parameters for GTF, and play a very important role in the convergence performance. To facilitate the setting of the parameters for GTF, their effects on the convergence accuracy are tested, and the PSNR (peak signal-to-noise ratio) [18] is used as the evaluation criterion, which is defined as follows:

**Fig. 2** The selected original images, i.e.,  $I_a, \dots, I_h$ , for our experiment in this section. These images have been widely used in the experiments of matrix completion methods, such as LMaFit, and TNNR





**Fig. 3** The partial intermediate results of the matrices  $B_j$  in GTF. **a**  $B_j, j = 1$ . **b**  $B_j, j = 10$ . **c**  $B_j, j = 20$ . **d**  $B_j, j = 30$ . The tested incomplete image for this experiment is generated by randomly masking one half of the entries from  $I_a$  shown in Fig. 2. The size of  $B$  is  $128 \times 128$ .

$$PSNR = 10 \cdot \log_{10} \left( \frac{N \cdot 255^2}{P_{\bar{\Omega}}(I - X_{out})_F^2} \right) \tag{31}$$

where  $I$  is the original matrix of  $M$ .  $N$  is the total number of missing entries in the matrix  $M$ .  $\bar{\Omega}$  is the set of locations

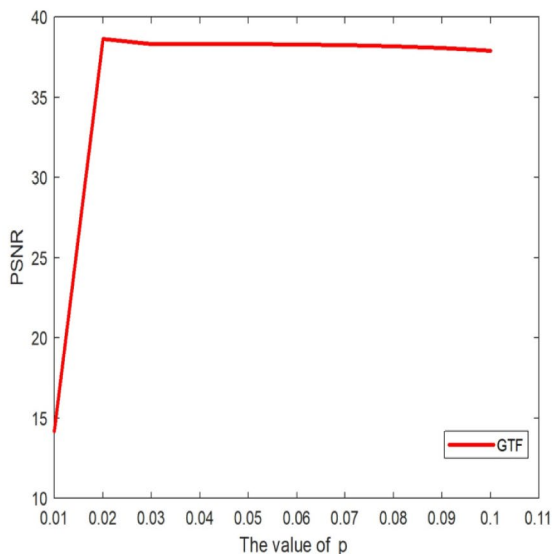
for the missing entries of the matrix  $M$ .  $X_{out}$  stands for the output of a matrix completion method.

By fixing  $q = 0.1$  and letting  $p$  increase from 0.01 to 0.1, the effect of parameter  $p$  on GTF is tested. The incomplete image tested in this experiment is generated by randomly masking one half of the entries from  $I_a$ , as shown in Fig. 2. The PSNR values of GTF with different  $p$  are plotted in Fig. 4a, b.

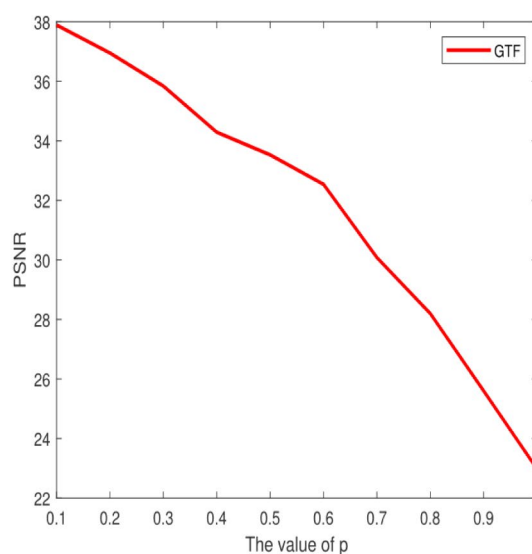
From Fig. 4a, we can see that the PSNR of GTF increases obviously when  $p \in (0.01, 0.02)$ , reaches its optimum at  $p = 0.02$  and decreases smoothly and slowly when  $p \in [0.02, 0.1)$ . To analyze the effect of  $p$  more comprehensively, the PSNR values of GTF when  $p$  is greater than 0.1 and less than 1 are plotted in Fig. 4b. It is easy to see that the accuracy of the GTF decreases rapidly in such cases from Fig. 4b.

Then, the effect of  $q$  on GTF is tested. Let  $p = 0.02$  and  $q$  increase from 0.1 to 1 with a step size equal to 0.1. The PSNR values of GTF with these values are tested and plotted in Fig. 5.

The PSNR curve of GTF in Fig. 5 shows that GTF reaches its optimal PSNR at  $q = 0.7$ . By comparing the PSNR curves in Figs. 4 and 5, it is easy to see that the GTF method is more sensitive to the parameter  $p$  than to the parameter  $q$ . In addition, the proposed GTF method has a good convergence accuracy performance when the parameter  $p \in [0.02, 0.1)$  and  $q \in [0.4, 0.8]$ .

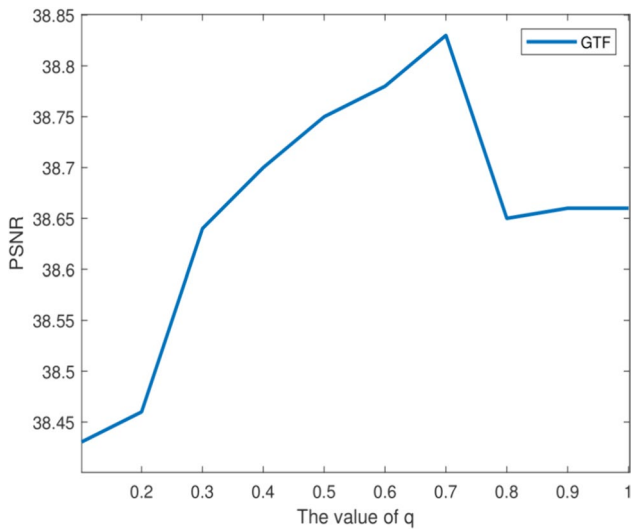


**(a)**  $p = 0.01 \sim 0.1$



**(b)**  $p = 0.1 \sim 1.0$

**Fig. 4** The PSNR values of GTF with different values of  $p$ . **a** The PSNR values of GTF with  $p$  increases from 0.01 to 0.1. **b** The PSNR values of GTF with  $p$  is greater than 0.1 and less than 1



**Fig. 5** The effect of parameter  $q$  on the accuracy of GTF. The parameter  $q$  increases from 0.1 to 1

**C. Matrix completion results on images with random missing entries**

The incomplete images tested in this section are constructed by randomly selecting 50% of the elements from the original images (in Fig. 2.), i.e.,  $I_a, I_b, \dots, I_h$ , are missing elements that are set to 0. The incomplete images, whose labels are  $M_a^{0.5}, M_b^{0.5}, \dots, M_h^{0.5}$ , are plotted in Fig. 6.

First, the convergence accuracy of GTF is tested. Let  $p = 0.02$  and  $q = 0.7$  for the GTF method. Using these values, the GTF method is tested and compared with the FTF method and other conventional methods, i.e., FNNM, LPMP, LNMQR, and SCPN. The convergence accuracies, i.e., PSNR, of the tested methods are shown in Table 1.

The PSNR values in Table 1 show that GTF is the most accurate among the six matrix completion methods. Since FTF is a nuclear norm minimization-based matrix completion method, its convergence accuracy is smaller than that

**Fig. 6** The tested incomplete images, whose 50% entries have randomly been initialized to 0, for our experiment in this section



**Table 1** The convergence accuracies (PSNRs) of the six methods

Images	FNNM	LPMP	SCPN	LNMQR	FTF	GTF
$M_a^{0.5}$	36.15	35.53	36.29	37.65	34.87	<b>38.29</b>
$M_b^{0.5}$	34.32	33.74	34.56	35.26	33.28	<b>35.88</b>
$M_c^{0.5}$	36.47	35.46	36.73	39.58	34.96	<b>40.06</b>
$M_d^{0.5}$	30.29	29.83	30.51	31.42	29.28	<b>31.67</b>
$M_e^{0.5}$	28.58	27.84	28.76	28.79	27.69	<b>29.78</b>
$M_f^{0.5}$	30.67	30.32	31.11	31.73	29.78	<b>33.03</b>
$M_g^{0.5}$	32.08	30.96	31.88	33.79	30.27	<b>34.28</b>
$M_h^{0.5}$	29.31	28.98	29.26	29.36	28.19	<b>29.83</b>

The bold values stand for the best PSNR values of the tested images

The incomplete images used in this experiment, i.e.,  $M_a^{0.5}, \dots, M_h^{0.5}$ , are generated by randomly masking 50% of the entries from the original images, i.e.,  $I_a, \dots, I_h$ , respectively



of a weighted nuclear norm minimization method, such as the LPMP method. The main reason is that the LPMP method can dynamically adjust the weights of the singular values by a nonconvex thresholding function. Similarly, the SCPN method also converges much more accurate than the FTF method does. Although the FNNM is also a nuclear-norm-based method, it is much more accurate than the FTF method. It might be because that the side matrices designed in FNNM make the incomplete matrices to be recovered easily.

The LNMQR method is much more accurate than the FNNM, LPMP and SCPN methods, because it has been proven to converge to an iteratively re-weighted nuclear norm minimization method. The PSNR of LNMQR in Table 1 also shows that LNMQR is much more accurate than FTF, which indicates that LNMQR is a very successful improvement of FTF.

Compared with the LNMQR method, the GTF method proposed in this paper performs much better in convergence accuracy, i.e., the PSNR of the latter is obviously larger than that of the former. The main reason is that the optimization model of GTF is easier to optimize by its updating step than that of LNMQR.

Second, the speed of GTF is compared with those of the other five conventional methods. To study the convergence process of the GTF method in more depth, the curves of PSNR versus the iteration number of the six methods are plotted in Fig. 7.

From Fig. 7, we see that the LNMQR and FTF methods can converge to their optimal solutions in the first 80 iterations. The SCPN, GTF and LPMP methods take 100

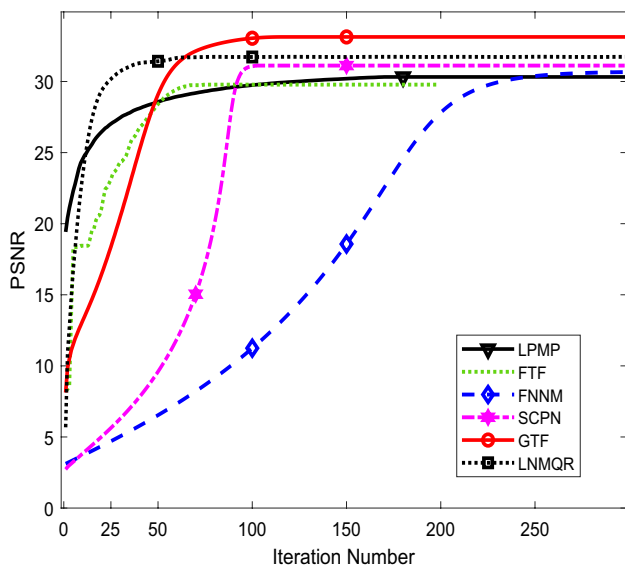


Fig. 7 The curves of PSNR versus the iteration number of the six tested methods

iterations, 110 iterations and 200 iterations to converge, respectively. The FNNM method, which takes 250 iterations to converge, is the slowest one among the six tested methods. Although the proposed GTF method takes more iterations to converge than the FTF method takes, the CPU time of GTF is smaller than that of FTF.

The CPU times of GTF, SCPN, LNMQR and FTF are plotted in Fig. 8. The CPU times of FNNM and LPMP are not reported in this section, because FNNM and LPMP take much more iterations to converge than the SCPN method takes, their CPU times should be as large as two times of that of SCPN or more. From Fig. 8, it is easy to see that the SCPN method is the slowest one among the four methods compared because of its multiple SVD iterations.

The curves in Fig. 8 shows that the proposed GTF method is almost as fast as two times of FTF. The main reason is that the computational cost of FTF is larger than that of GTF, i.e., the former takes more time to perform the SVD decomposition at each iteration than the latter does. In view of the PSNR values in Table 1 and the curves in Fig. 8, the proposed GTF method is much faster than the SCPN and FTF methods and is more accurate than the FTF, SCPN, LPMP, LNMQR and FNNM methods. The curves in Fig. 8 still shows that GTF takes 1–2 s more CPU time to converge than LNMQR takes, because the former performs more iterations to converge than the latter. Hence, we will further improve GTF to reduce its number of iterations for increasing its speed.

Third, the convergence performance of GTF on images with a low observation rate is tested. From the previous experimental results, it is clear that the proposed GTF

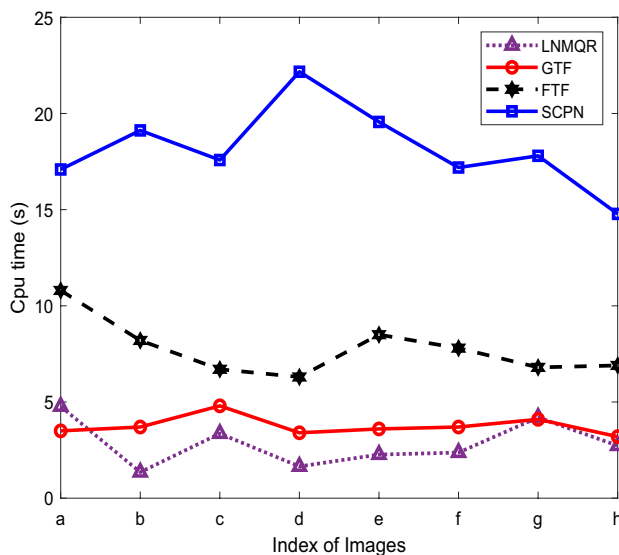


Fig. 8 The CPU times of GTF, SCPN, FTF and LNMQR. The CPU times of LPMP and SVT are not reported because their convergence accuracies are obviously smaller than the accuracy of GTF

method is more accurate and faster than the FNNM, LPMP, FTF, LNMQR and SCPN methods. In fact, the GTF method still has advantages over the traditional methods in the case of images with low observation rates. We randomly select 70% of the elements from the eight original images in Fig. 1 as missing elements for generating incomplete images, which are labeled  $M_a^{0.7}, \dots, M_h^{0.7}$ . The PSNR values of GTF, FTF, SCPN, FNNM, LPMP and LNMQR on these incomplete images are reported in Table 2.

From Table 2, it is clear that the GTF method is the most accurate among the tested matrix completion methods. The SCPN method is a bit more accurate than the FNNM

method. The PSNR value of FTF is obviously smaller than those of LNMQR, FNNM and SCPN. Therefore, the GTF method is ideally suitable for application cases with low observation rates.

To visually demonstrate the differences between the six matrix completion methods, some recovered images are displayed in Fig. 9.

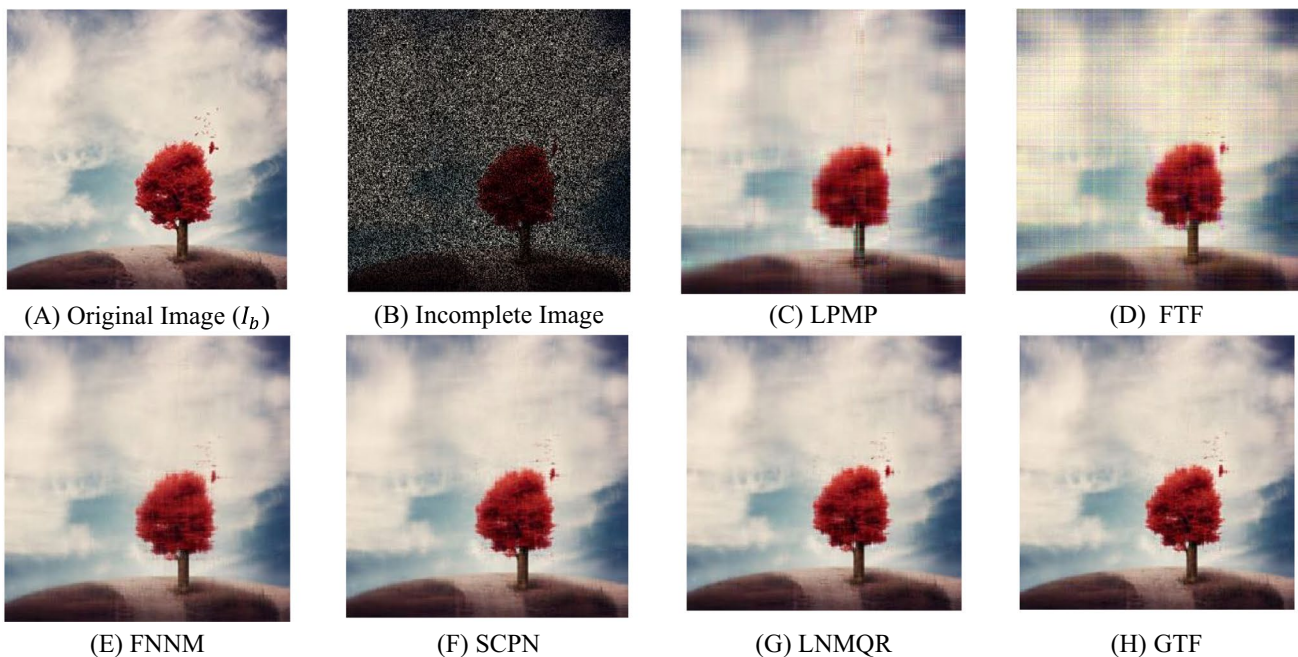
The recovered images given by the six tested methods in Fig. 9 show that the output of the FTF is not clear because there is considerable visible noise randomly distributed in the image shown in Fig. 9D. The output of LPMP shown in (C) is much better than that of FTF. However, it is clearly

**Table 2** The PSNR values of GTF, FTF, SCPN, FNNM, LPMP and LNMQR

Images	FNNM	LPMP	SCPN	LNMQR	FTF	GTF
$M_a^{0.7}$	31.42	28.85	32.31	32.55	25.72	<b>33.61</b>
$M_b^{0.7}$	30.11	27.37	31.20	31.52	22.41	<b>31.85</b>
$M_c^{0.7}$	29.25	24.83	31.42	32.17	24.51	<b>33.06</b>
$M_d^{0.7}$	25.42	23.27	27.22	27.29	24.18	<b>27.83</b>
$M_e^{0.7}$	25.11	24.07	25.60	25.53	22.49	<b>25.84</b>
$M_f^{0.7}$	25.26	22.20	26.74	25.75	23.89	<b>27.22</b>
$M_g^{0.7}$	26.29	24.97	28.12	27.87	24.83	<b>28.58</b>
$M_h^{0.7}$	25.45	24.83	25.91	25.73	24.04	<b>26.16</b>

The bold values stand for the best PSNR values of the tested images

The incomplete images used in this experiment are generated by randomly masking 70% of the entries from the original images



**Fig. 9** The recovered images given by the GTF, FTF, FNNM, LNMQR, LPMP and SCPN methods. **A** An original image. **B** The tested incomplete image whose 70% entries are randomly missing. **C** The output of LPMP, PSNR=27.37. **D** The output of FTF,

PSNR=22.41. **E** The output of FNNM, PSNR=30.11. **F** The output of SCPN, PSNR=31.20. **G** The output of LNMQR, PSNR=31.52. **H** The output of GTF, PSNR=31.85

and visually still very different from the original image shown in (A). The result given by FNNM shown in (E) is clearer than the recovered image shown in (C). However, there is considerable visible noise randomly distributed in (E). The SCPN, LNMQR and GTF methods can recover the incomplete image accurately because the images in (F), (G) and (H) are almost as clear as the original image.

**D. Matrix completion results on images with structured missing blocks**

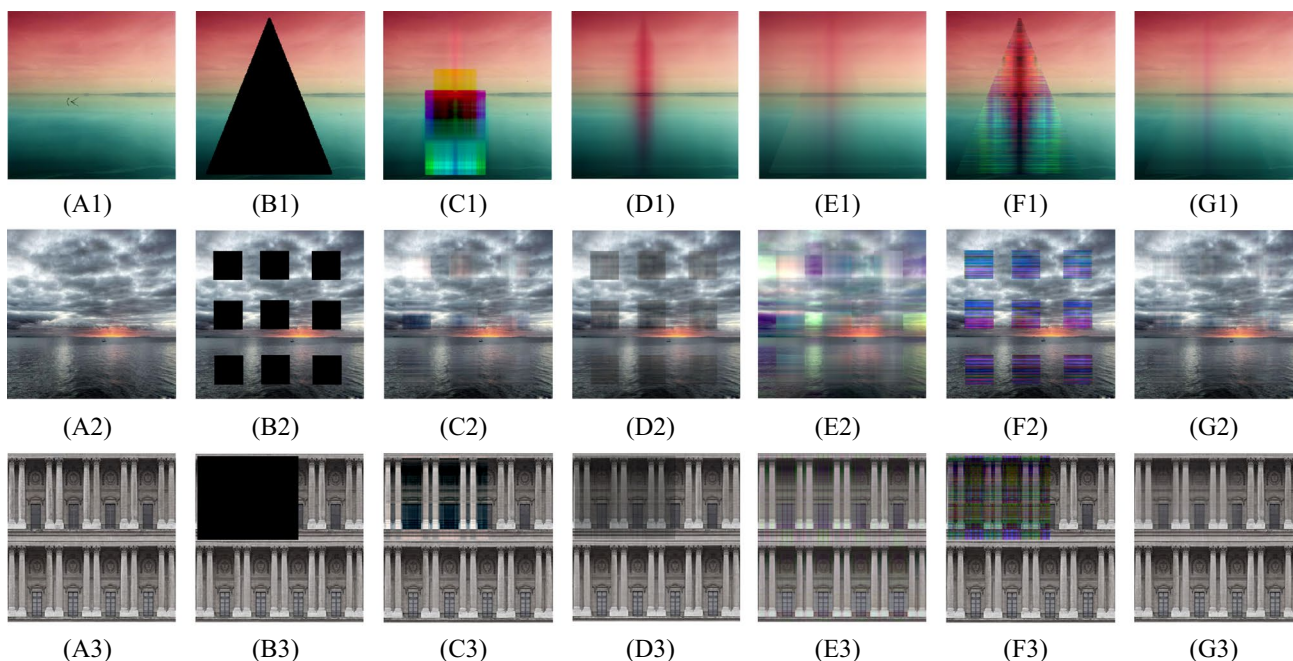
In this section, three types of structured missing blocks are added to the tested images, i.e., the triangular missing block, the square missing blocks and the large square missing block. The original images and the incomplete images have been shown in Fig. 10. By using these three incomplete images, the GTF method is tested and its convergence speed, CPU time and PSNR value are compared with those of FTF, SCPN, LNMQR, and FNNM.

From Fig. 10, it is not easy to see that the FTF method cannot recover the three incomplete images with structured missing blocks accurately, because the recovery results shown in (F1)–(F3) are significantly different from the original images plotted in (A1)–(A3), respectively. Although the LNMQR method has a good convergence accuracy on images with random missing values, it fails to recover the tested images arranged with structured missing blocks in this section. It’s clear that the images given by LNMQR shown in (C1)–(C3) have many distorted pixels. The recovered images

given by FNNM are much better than those of LNMQR and FTF. However, the recovered images of FNNM shown in (D1)–(D3) are still visibly different from the original images. The recovered images given by SCPN are clearer than those of FNNM, FTF and LNMQR. However, the colors of some pixels in the recovered images in (E1)~(E3) are quite different from those of the original images. It might be because that the SCPN method is very sensitive to its parameters and it cannot recover the three channels of incomplete images with the same fixed set of parameters. The recovered images of our proposed GTF method are much better than the corresponding ones of FTF, SCPN, FNNM and LNMQR, which indicates that the convergence accuracy of GTF should be much larger than those of other four methods.

In order to compare the convergence performances of GTF, FTF, SCPN, FNNM and LNMQR in more detail, the PSNR values, iteration numbers and CPU times of these methods on the three incomplete images in Fig. 10, i.e., (B1), (B2) and (B3), are reported in Table 3.

Table 3 shows that FTF is the fastest among the five matrix completion methods and it also has the smallest convergence accuracy. Although the LNMQR method takes more iterations than the FNNM and SCPN methods do, its CPU time is smaller than those of the latter. The main reason is that the LNMQR method does not require the usage of SVD decomposition at each iteration. The FNNM and SCPN methods are much slower than the GTF



**Fig. 10** The tested incomplete images and the results given by GTF, FTF, SCPN, FNNM and LNMQR. **A1–A3** Three original images. **B1–B3** The incomplete images. **C1–C3** The results given by LNMQR. **(D1)–(D3)**: the results given by FNNM. **E1–E3** The

results given by SCPN. **F1–F3** The results given by FTF. **G1–G3** The results given by GTF. The maximum iteration numbers for these five methods are all equal to 500

**Table 3** The PSNR values, iteration numbers and CPU times of GTF, FTF, SCPN, FNNM and LNMQR

Tested Image		LNMQR	FNNM	SCPN	FTF	GTF
(B1)	PSNR	12.41	20.23	24.55	15.34	28.16
	Iteration Number	500	379	371	150	369
	CPU Time (s)	21.24	62.84	57.20	8.43	10.02
(B2)	PSNR	19.11	15.48	19.34	10.44	21.49
	Iteration Number	497	419	368	109	275
	CPU Time (s)	21.01	64.72	54.01	6.79	9.79
(B3)	PSNR	16.35	18.76	22.02	13.54	23.53
	Iteration Number	500	305	381	85	270
	CPU Time (s)	22.39	50.42	57.75	5.75	9.18

method, because their CPU times are about 5~6 times of that of GTF. According to the iteration numbers and the CPU times of the five methods, it is not difficult to see that the GTF method is about as fast as 2.1 times, 6.2 times and 5.5 times of LNMQR, FNNM and SCPN, respectively. Although the GTF method takes 1~3 s longer to converge than FTF takes, its convergence accuracy is much better than that of FTF.

In view of the results in Section IV. C and IV. D, it is clear that the FTF, SCPN and FNNM methods are less accurate than the LNMQR and GTF. The GTF method is suitable for matrix completion tasks with high requirements for convergence accuracy and CPU time both on matrices with missing random missing entries and with structured missing blocks. The LNMQR method, which might not be suitable for the applications with missing blocks, is suitable for the cases with random missing entries of high requirements for CPU time.

## 5 Conclusion

A generalized tri-factorization method for matrix completion is proposed in this paper (GTF). In this method, a generalized  $L_{1,p}$  norm optimization model, which can be optimized by an alternating direction method very efficiently, is investigated to improve the convergence accuracy of the traditional FTF method. On the one hand, since the  $L_{1,p}$  norm is a tighter relaxation of the matrix rank than the nuclear norm and can converge to a weighted nuclear norm, the proposed GTF method is much more accurate than the FTF method and the other state-of-the-art methods. On the other hand, the GTF method is much faster than the traditional matrix completion methods that use SVD decomposition because it uses QR decomposition as a replacement for SVD. Numerous experimental results show that GTF is more accurate and much faster than the traditional FTF method and the other compared methods.

**Funding** This work was supported in part by the National Natural Science Foundation of China under Grant 62102002, in part by the transverse project of underwater high-speed navigation test site and technical services (0045022007), in part by the transverse project of designing and processing of gas gun driven by high pressure air mixed with Gas (0045021079), in part by the Natural Science Foundation of Anhui Province of China under Grant 2008085QF291, by the Research Start-up Fund of West Anhui University, No. WGKQ2021053.

**Data availability** The data for supporting the findings of this work will be made available on reasonable request.

## References

- Jia Z, Jin Q, Ng MK, Zhao XL (2022) Non-local robust quaternion matrix completion for large-scale color image and video inpainting. *IEEE Trans Image Process* 31:3868–3883
- Miao J, Kou KI (2022) Color image recovery using low-rank quaternion matrix completion algorithm. *IEEE Trans Image Process* 31:190–201
- Chen L, Jiang X, Liu X, Zhou Z (2021) Logarithmic norm regularized low-rank factorization for matrix and tensor completion. *IEEE Trans Image Process* 30:3434–3449
- Yang M, Li Y, Wang J (2022) Feature and nuclear norm minimization for matrix completion. *IEEE Trans Knowl Data Eng* 34(5):2190–2199
- Li G, Guo G, Peng S, Wang C, Yu S, Niu J, Mo J (2022) Matrix completion via SCHATTEN capped p norm. *IEEE Trans Knowl Data Eng* 34(1):394–404
- Li Q (2022) Face recognition with robust matrix factorization. In: *Proceedings of 2022 15th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI)*, Beijing, China, pp 1–5
- Sridhar KV, Madugula P (2022) Performance analysis of weighted low-rank approximation models for robust face recognition. In: *2022 IEEE World Conference on applied intelligence and computing (AIC)*, Sonbhadra, India, pp 242–246
- Li X, Zhang H, Zhang R (2023) Matrix Completion via Non-Convex Relaxation and Adaptive Correlation Learning. *IEEE Trans Pattern Anal Mach Intell* 45(2):1981–1991
- Chen Y, Cheng L, Wu Y (2023) Bayesian low-rank matrix completion with dual-graph embedding: prior analysis and tuning-free inference. *Signal Process* 204:108826
- Wang Z, So H, Liu Z (2022) Fast and robust rank-one matrix completion via maximum correntropy. *Signal Process* 198:108580

11. Tutuncu RH, Toh KC, Todd MJ (2001). Sdpt3—a Matlab Software Package for Semidefinite Quadratic Linear Programming, Version 3.0
12. Fazel M (2002) Matrix rank minimization with applications [M]. PhD thesis, Stanford Univ.
13. Meka R, Jain P, Dhillon IS (2010) Guaranteed rank minimization via singular value projection. In: Proceedings of advances in neural information processing systems
14. Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. *SIAM J Optim* 20:1956–1982
15. Toh KC, Yun S (2010) An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pac J Optim* 6(3):615–640
16. Gu S, Zhang L, Zuo W, Feng X (2014) Weighted nuclear norm minimization with application to image denoising. In: Proceedings of 2014 IEEE Conference on computer vision and pattern recognition, Columbus, OH, USA, pp 2862–2869
17. Liu L, Huang W, Chen D (2014) Exact minimum rank approximation via Schatten p norm minimization. *J Comput Appl Math* 267:218–227
18. Hu Y, Zhang D, Ye J, Li X, He X (2013) Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE Trans Pattern Anal Mach Intell* 35(9):2117–2130
19. Lu C, Tang J, Yan S, Lin Z (2016) Nonconvex non-smooth low rank minimization via iteratively reweighted nuclear norm. *IEEE Trans Image Process* 25(2):829–839
20. Liu Q, Davoine F, Yang J, Cui Y, Jin Z, Han F (2019) A Fast and Accurate Matrix Completion Method Based on QR Decomposition and L<sub>2,1</sub>-Norm Minimization. *IEEE Trans Neural Netw Learn Syst* 30(3):803–817
21. Wen Z, Yin W, Zhang Y (2012) Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation. *Math Prog Comp* 4:333–361
22. Liu Y, Jiao L, Shang F, Yin F, Liu F (2013) An efficient matrix bi-factorization alternative optimization method for low-rank matrix recovery and completion. *Neural Netw* 48:8–18
23. Liu Y, Jiao L, Shang F (2013) A fast tri-factorization method for low-rank matrix recovery and completion. *Pattern Recogn* 46(1):163–173
24. Zhao J, Liang Y, Yi S, Shen Q, Cao X (2023) Improving generalization of double low-rank representation using Schatten-p norm. *Pattern Recogn* 138:109352
25. Cai B, Lu G (2022) Tensor subspace clustering using consensus tensor low-rank representation. *Inf Sci* 609:46–59
26. Li X, Liu Q, So HC (2020) Rank one matrix approximation with L<sub>p</sub> norm for image inpainting. *IEEE Signal Process Lett* 27:680–684
27. Liu Q, Peng C, Yang P, Zhou X, Liu Z (2023) A fast matrix completion method based on matrix Bi-factorization and QR decomposition. *Wirel Commun Mob Comput* 2023:12 (**Article ID 2117876**)
28. Rezaei B, Ostadabbas S (2018) Moving object detection through robust matrix completion augmented with objectness. *IEEE J Sel Top Signal Process* 12(6):1313–1323
29. Rezaei B, Ostadabbas S (2017) Background subtraction via fast robust matrix completion. In: Proceedings of 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, pp 1871–1879
30. Liu XX, Li C, Xiang YK, Liu K, Hu ZP, Guo XZ (2021) Graph matrix completion for power product recommendation. In: Proceedings of 2021 IEEE 16th Conference on industrial electronics and applications (ICIEA), Chengdu, China, pp 1267–1271
31. Zhang T, Ghanem B, Liu S, Xu C, Ahuja N (2013) Low rank sparse coding for image classification. In: the Proceedings of IEEE International Conference on computer vision, 2013

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.