ORIGINAL PAPER

Weighted hybrid truncated norm regularization method for low-rank matrix completion

Xiying Wan[1] · Guanghui Cheng[1]

Abstract

Matrix completion is usually formulated as a low-rank matrix approximation problem. Several methods have been proposed to solve this problem, e.g., truncated nuclear norm regularization (TNNR) which performs well in recovery accuracy and convergence speed, and hybrid truncated norm regularization (HTNR) method which has better stability compared to TNNR. In this paper, a modified hybrid truncated norm regularization method, named WHTNR, is proposed to accelerate the convergence of the HTNR method. The proposed WHTNR method can preferentially restore rows with fewer missing elements in the matrix by assigning appropriate weights to the first r singular values. The presented experiments show empirical evidence on significant improvements of the proposed method over the closest four methods, both in convergence speed or in accuracy, it is robust to the parameter truncate singular values r.

Keywords Matrix completion · Low rank · Truncated norm · Weights

1 Introduction

Matrix completion [1] aims to recover an unknown low-rank or approximately low-rank matrix from the observed missing matrix. It is widely used in computer vision [2–4], recommendation system [5–7], machine learning [8], etc. It is well-known that the vast majority of visual data is of low-rank or approximately low-rank structure [9]. Wright et al. [10] transferred the matrix completion problem into a rank minimization model while retaining the low-rank or approximately low-rank structure of

Guanghui Cheng contributed equally to this work.

✉ Guanghui Cheng
  ghcheng@uestc.edu.cn

[1] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, 611731, People's Republic of China

the matrix. Specifically, given the incomplete data matrix $M \in R^{m \times n}$, the general completion model is rewritten as follows

$$
\begin{aligned}
&\min_{X} \quad \text{rank}(X), \\
&\text{s.t.} \ \ X_{ij} = M_{ij}, \ (i, j) \in \Omega,
\end{aligned}
\tag{1}
$$

where rank(X) denotes the rank of the matrix X, and Ω is the set of locations in matrix $X \in R^{m \times n}$, corresponding to the known entries.

Unfortunately, because the rank function is discontinuous and non-convex, solving the rank minimization model (1) is NP-hard [11], which greatly limits the practical application of this model. M. Fazel et al. [12] firstly suggest that the nuclear norm is a good substitute for the matrix rank function. It showed that the nuclear norm is the most compact convex lower bound of the rank function [12]. The rank of matrix X is equal to the number of its non-zero singular values, and the nuclear norm is the sum of singular values. The relationship between rank function and the nuclear norm is similar to the relationship between ℓ_1 and ℓ_0 norm.

Inexact augmented Lagrangian multiplier (IALM) method [13] and accelerated proximal gradient method [14], to name a few, have been developed in recent years to solve this model, however with unsatisfactory performance in practical applications. The main reason is that the nuclear norm is not the best substitute for the rank function when the matrix does not have a strict low-rank structure. In the rank function, all non-zero singular values should have an equal contribution. However, in the nuclear norm, the contributions of singular values are different according to their magnitudes. The singular value threshold (SVT) method [15] is widely used for solving the nuclear norm minimization model. But the SVT method needs to make a large number of iterations to converge. Hence, the nuclear norm may not be the best substitution of the rank function in the practical application.

To obtain a better approximation of the rank function, Zhang et al. [16] firstly proposed the truncated nuclear norm considered as an alternative to the rank function, and designed an effective two-step iteration algorithm. Subsequently, to solve the convex subproblem in the second step, three efficient iterative procedures were introduced by Hu et al. [17], called TNNR-APGL, TNNR-ADMM, and TNNR-ADMMAP. The TNNR method has significantly improved convergence accuracy. To improve the rate of convergence and robustness of the TNNR method, a truncated nuclear norm regularization method based on weighted residual error was proposed by Liu et al. [18], called the TNNR-WRE and ETNNR-WRE methods, respectively. Hereafter, the double weighted truncated nuclear norm regularization was proposed by Xue et al. [19], called DW-TNNR. Then, quaternion truncated nuclear norm for LRQMC was proposed by Yang et al. [20], called QTNN. However, as a variant of the nuclear norm, its stability problem still faces challenges. The stability of the matrix completion model aims to keep the output matrix almost unchanged when the elements of the observation matrix are replaced.

To improve the stability of the model, Cai et al. [15] proposed a model that combines the nuclear norm and the Frobenius norm, with which they control the low rank and stability of the model, respectively. The Frobenius norm is the arithmetic

root of the sum of squares of its singular values and the truncated nuclear norm discards r large singular values. Although the Frobenius norm can improve stability, if it is directly integrated into the TNN model, it may not work well. Therefore, it may be inappropriate to directly combine the Frobenius norm with the truncated nuclear norm improving the model stability. To overcome the above shortcomings, Ye et al. [21] combined the truncated nuclear norm with truncated Frobenius norm, proposed a hybrid truncated norm regularization (HTNR) method, and improved the recovery accuracy and the model stability simultaneously. However the convergence speed of the HTNR method is not satisfying. The HTNR designs a two-step iterative scheme that needs to recover all the missing entries of an incomplete matrix in each step. As the variant method of truncated nuclear norm, the robustness of r is very important for HTN, therefore the choice of r will affect the convergence accuracy.

As a general cognition, the matrix completion task gets easier after recovering some rows with more observed entries. With this idea, we propose a weighted hybrid truncated norm regularization method (WHTNR), aiming to improve the convergence speed. At the same time, the WHTNR algorithm has better convergence accuracy and it is very robust to the parameter r.

In the WHTNR method, different weights are assigned according to the proportion of missing elements to the rows, so that the rows with fewer missing elements will restore firstly hoping to accelerate the matrix recovery process. The experimental results show that the WHTNR not only has a better convergence speed, but also has better accuracy compared with SVT, IALM, TNNR, HTNR, and TNNR-WRE methods (the closest four methods).

The rest of the paper is as follows. In Section 2, some closely related works are introduced. In Section 3, we introduce the proposed WHTNR model and discuss its convergence. In Section 4, experimental results on real images are presented.

2 Related work

Suppose the singular value decomposition (SVD) of the matrix $X \in R^{m \times n}$ is as follows

$$X = U \Sigma V^T, \quad \Sigma = \mathrm{diag}\left(\{\sigma_i(X)\}_{1 \le i \le \min(m,n)}\right),$$

where $U = [u_1, u_2, \ldots, u_m] \in R^{m \times m}$ and $V = [v_1, v_2, \ldots, v_n] \in R^{n \times n}$ are orthogonal matrices, $\sigma_i(X)$ is the ith singular value of X, and is assumed to be sorted in descending order, $\sigma_1(X) \ge \sigma_2(X) \ge \cdots \ge 0$.

As mentioned in the introduction, it is intractable to directly solve the optimization model (1). Conventional approaches use an approximate method via convex relaxation. Specifically, the rank minimization model (1) could be rewritten by approximating the rank function with nuclear norm [12] as follows

$$
\begin{aligned}
&\min_{X} \ \|X\|_*, \\
&\text{s.t.} \ \ P_\Omega(X) = P_\Omega(M),
\end{aligned}
\tag{2}
$$

where $\|X\|_* = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X)$ denotes the nuclear norm of the matrix X, P_Ω denotes the orthogonal projection operator onto the span of matrices vanishing outside of Ω. For any matrix X, we have

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

The singular value thresholding (SVT) method for matrix completion proposed by Cai et al. [15] combines the nuclear norm and the Frobenius norm, with which they control the low rank and stability of the model, respectively. The SVT model [15] is as follows

$$\begin{aligned} \min_X \quad & \|X\|_* + \alpha\|X\|_F^2, \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M), \end{aligned} \tag{4}$$

where $\|X\|_F = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2(X)}$ denotes the Frobenius norm of the matrix X.

Since the nuclear norm is the sum of singular values, a few largest singular values have an important contribution to the nuclear norm. To obtain a better approximation to the rank function, Zhang et al. proposed a truncated nuclear norm [16]. For any matrix $X \in R^{m \times n}$, the truncated nuclear norm minimization model (TNN) is defined as follows

$$\begin{aligned} \min_X \quad & \|X\|_{r,*}, \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M). \end{aligned} \tag{5}$$

And the inequality holds [16] as follows

$$\max_{CC^T=I, DD^T=I} \text{Tr}\left(CXD^T\right) = \sum_{i=1}^{r} \sigma_i(X). \tag{6}$$

Thus, the optimization model (5) can be rewritten as follows:

$$\begin{aligned} \min_X \quad & \|X\|_* - \max_{CC^T=DD^T=I} \text{Tr}\left(CXD^T\right), \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M). \end{aligned} \tag{7}$$

However, it is inappropriate to directly combine the Frobenius norm with the TNN term to improve the stability of the model. Later, Ye et al. [21] proposed a hybrid truncated norm model instead of integrating the Frobenius norm into the TNN term. This method improves recovery accuracy and enhances stability. Experimental results demonstrate the effectiveness of the HTNR model. For any matrix $X \in R^{m \times n}$, the truncated Frobenius norm (TFN) is defined as follows

$$\|X\|_{r,F} = \sqrt{\sum_{i=r+1}^{\min(m,n)} \sigma_i^2(X)}. \tag{8}$$

So, the hybrid truncated norm minimization model (HTNR) is as follows

$$\min_{X} \quad \|X\|_* - \max_{CC^T=DD^T=I} \text{Tr}\left(CXD^T\right) + \lambda \left(\|X\|_F^2 - \max_{CC^T=I} \|CX\|_F^2\right), \quad (9)$$
$$\text{s.t.} \quad P_\Omega(X) = P_\Omega(M).$$

The main iterative steps of the HTNR algorithm are presented in Algorithm 1.

Input: An original matrix with missing elements, M; The position coordinate collection of the missing element, Ω;

Output: returns the recovery matrix;

  1: **Initialization:** $X_1 = P_\Omega(M) = M_\Omega$

  2: **repeat**

  3:     **Step1:** Given X_k, calculate the singular value decomposition of X_k:

$$[U_k, \Sigma_k, V_k] = \text{SVD}\,(X_k)$$

  4:     where

$$U_k = [u_1, u_2, \ldots, u_m] \in R^{m \times m}, \; V_k = [v_1, v_2, \ldots, v_n] \in R^{n \times n}$$

  5:     Let C_k and D_k as:

$$C_k = [u_1, u_2, \ldots, u_r]^T, \; D_k = [v_1, v_2, \ldots, v_r]^T$$

  6:     **Step2:** Solve the following convex optimization model:

$$\min_{X} \quad \|X\|_* - \text{Tr}\left(C_k X D_k^T\right) + \lambda \left(\|X\|_F^2 - \|C_k X\|_F^2\right)$$
$$\text{s.t.} \quad P_\Omega(X) = P_\Omega(M)$$

  7:   **until**

$$\|X_{k+1} - X_k\|_F \le \varepsilon_0$$

Algorithm 1 HTNR algorithm [21].

3 WHTNR method for matrix completion

In this section, we propose a weighted hybrid truncated norm regularization method (WHTNR) which is dedicated to improving the iteration speed of the HTNR model while ensuring recovery accuracy. Aiming to improve the matrix restoration speed, the rows with fewer missing elements will be recovered first.

3.1 The WHTNR method

In line with the trace inequality [22], if $r = \min\{m, n\}$, then we have

$$\text{Tr}\left(AXB^T\right) \le \|X\|_*, \qquad (10)$$

where $AA^T = BB^T = I$. Consequently, the truncated nuclear norm is rewritten as follows

$$\|X\|_* = \max_{AA^T=BB^T=I} \mathrm{Tr}\left(AXB^T\right). \tag{11}$$

On the basis of (11), the model (9) is presented as follows

$$\begin{aligned}
\min_X & \max_{AA^T=BB^T=I} \mathrm{Tr}\left(AXB^T\right) - \max_{CC^T=DD^T=I} \mathrm{Tr}\left(CXD^T\right) \\
& + \lambda \left(\|X\|_F^2 - \max_{CC^T=I} \|CX\|_F^2 \right), \\
\text{s.t.} \quad & P_\Omega(X) = P_\Omega(M).
\end{aligned} \tag{12}$$

In order to restore the rows with fewer missing elements in the matrix preferentially, Liu et al. [18] assigned different weights to each row of the residual matrix $X - H$. Inspired by this, the X terms of $\mathrm{Tr}\left(AXB^T\right)$ in the model (12) are weighted to restore the rows with few missing elements first. The optimization model (12) is changed as follows

$$\begin{aligned}
\min_X & \quad \mathrm{Tr}\left(AWXB^T\right) - \mathrm{Tr}\left(CXD^T\right) + \lambda \left(\|X\|_F^2 - \|CX\|_F^2 \right), \\
\text{s. t.} & \quad P_\Omega(X) = P_\Omega(M),
\end{aligned} \tag{13}$$

where $A = U^T$, $B = V^T$, $C = (u_1, \ldots, u_r)^T$, $D = (v_1, \ldots, v_r)^T$. U and V are the left and right orthogonal matrices in the SVD decomposition of X, respectively, and r is the number of subtracted singular values.

In addition, the weight matrix W is defined as follows

$$W = \mathrm{diag}\left(w_1, \ldots, w_m\right). \tag{14}$$

where $\lambda_i > 0$, $1 \le w_i \le w_j$, if $N_i > N_j$, $(i, j = 1, \ldots, m)$, and N_i is the number of observed entries in the ith row of X. Any matrix with rank r can be expressed as the sum of r matrices with rank 1, and singular values measure the weight of these matrices with rank 1 to the original matrix. In image processing, small singular values can be considered to be caused by missing elements. For the nuclear norm minimization model of the matrix completion algorithm, missing elements can have a large impact on the larger first few singular values of the matrix. The WHTNR algorithm, which is one of the variants of truncation of the nuclear norm, completely truncates the first r singular values, which may lead to poor robustness to the parameter r.

The optimization function of the model (13) is

$$\begin{aligned}
& \mathrm{Tr}(AWXB^T) - \mathrm{Tr}\left(CXD^T\right) + \lambda \left(\|X\|_F^2 - \|CX\|_F^2 \right) \\
&= \sum_{i=1}^m \sigma_i(X)\mathrm{Tr}\left(u_i^T W u_i\right) - \sum_{i=1}^r \sigma_i(X)\mathrm{Tr}\left(u_i^T u_i\right) + \lambda \sqrt{\sum_{i=r+1}^m \sigma_i^2(X)} \\
&= \sum_{i=1}^r \sigma_i(X)\mathrm{Tr}\left(u_i^T (W - I) u_i\right) + \sum_{i=r+1}^m \sigma_i(X)\mathrm{Tr}\left(u_i^T W u_i\right) + \lambda \sqrt{\sum_{i=r+1}^m \sigma_i^2(X)}.
\end{aligned} \tag{15}$$

The weight of each singular value $\sigma_i(X)$ should be greater than 0. Based on the above analysis, for a given W, we have $\mathrm{Tr}\left(u_i^T\left(W-I\right)u_i\right) > 0$, let w_i satisfy

$$1 \le w_i \le w_j, \text{ if } N_i > N_j. \tag{16}$$

That is, the rows with more missing elements are assigned smaller weights.

3.2 Solving WHTNR model

According to model (13), we can update X_{k+1} by solving the next convex optimization model:

$$\begin{aligned}\min_X \quad & \mathrm{Tr}\left(A_k W X B_k^T\right) - \mathrm{Tr}\left(C_k X D_k^{\,T}\right) + \lambda\left(\|X\|_F^2 - \|C_k X\|_F^2\right), \\ \text{s. t.} \quad & P_\Omega(X) = P_\Omega(M).\end{aligned} \tag{17}$$

For solving the model (13), an auxiliary variable $H \in R^{m \times n}$ is added to relax the constraint, and the above model is equivalent to

$$\begin{aligned}\min_X \quad & \mathrm{Tr}\left(A_k W X B_k^T\right) - \mathrm{Tr}\left(C_k H D_k^{\,T}\right) + \lambda\left(\|X\|_F^2 - \|C_k X\|_F^2\right), \\ \text{s. t.} \quad & X = H,\ P_\Omega(H) = P_\Omega(M).\end{aligned} \tag{18}$$

To preferentially recover the rows wit h fewer missing elements in the matrix, the Lagrange function as follows

$$\begin{aligned}L(X, H, Y) = \ & \mathrm{Tr}\left(A_k W X B_k^T\right) - \mathrm{Tr}\left(C_k H D_k^T\right) + \lambda\left(\|X\|_F^2 - \|C_k X\|_F^2\right) \\ & + \frac{\mu}{2}\|(X - H)\|_F^2 + \mathrm{Tr}\left(Y^T(X - H)\right).\end{aligned} \tag{19}$$

We solve the Lagrange function (19) by alternating direction iterative method. Firstly, fix X_k and Y_k to update H_{k+1},

$$\begin{aligned}H_{k+1} &= \arg\min_H L\left(X_k, H, Y_k\right) \\ &= \arg\min_H -\mathrm{Tr}\left(C_k H D_k^T\right) + \frac{\mu_k}{2}\left\|(X_k - H) + \frac{1}{\mu_k}Y_k\right\|_F^2 \\ &= X_k + \frac{1}{\mu_k}C_k^T D_k + \frac{1}{\mu_k}Y_k.\end{aligned} \tag{20}$$

Based on the constraints in model (18), the value of the recovery matrix is consistent with the observed value. We have

$$H_{k+1} = P_{\Omega^c}\left(H_{k+1}\right) + P_\Omega(M). \tag{21}$$

where

$$\left(P_\Omega^c(H)\right)_{ij} = \begin{cases} 0, & \text{if } (i, j) \in \Omega, \\ H_{ij}, & \text{otherwise}. \end{cases}$$

Secondly, fix H_{k+1} and Y_k to update X_{k+1},

$$
\begin{aligned}
X_{k+1} &= \arg\min_X L\left(X, H_{k+1}, Y_k\right) \\
&= \arg\min_X \mathrm{Tr}\left(A_k W X B_k^T\right) + \lambda\left(\|X\|_F^2 - \|C_k X\|_F^2\right) \\
&\quad + \frac{\mu}{2}\left\|(X - H_{k+1}) + \frac{1}{\mu_k}Y_k\right\|_F^2 .
\end{aligned}
\tag{22}
$$

Then, combining (20) and (22), we obtain the following iterative scheme

$$
X_{k+1} = T_k(X_k - Z_k),
\tag{23}
$$

where $T_k = \left[\frac{2\lambda}{\mu}\left(I - C^\top C\right) + I\right]^{-1}$, $Z_k = \frac{1}{\mu}\left(W A^\top B - C^\top D\right)$. Like (20),

$$
X_{k+1} = P_{\Omega^c}\left(X_{k+1}\right) + P_\Omega(M),
\tag{24}
$$

Updating Y_{k+1} with fixed H_{k+1} and X_{k+1} is performed as follows

$$
Y_{k+1} = Y_k + \mu_k\left(X_{k+1} - H_{k+1}\right).
\tag{25}
$$

By (23), the update expression about X_{k+1} does not contain Y_k or Y_{k+1}, so there is no need to calculate Y_{k+1} in the algorithm. The parameter μ_k is updated by

$$
\mu_k = \rho\mu_{k-1},
$$

where $\mu_0 > 0$, $\rho > 1$ and $k = 1, 2, \ldots$.

In conclusion, the main steps of the WHTNR algorithm are described in Algorithm 2.

Input: An original matrix with missing elements, M and the position coordinate collection of the missing element, Ω;
Output: Returns the recovery matrix;
1: **Initialization:** $X_1 = P_\Omega(M) = M_\Omega$;
2:    **repeat**
3:    Given X_k, calculate the singular value decomposition of X_k:

$$
[U_k, \Sigma_k, V_k] = \mathrm{SVD}\left(X_k\right).
$$

4:    Update A_k, B_k, C_k and D_k by U_k and V_k

$$
A = U^T, B = V^T, C = (u_1, \ldots, u_r)^T, D = (v_1, \ldots, v_r)^T.
$$

5:    Update X_{k+1} through

$$
X_{k+1} = T_k(X_k - Z_k),
$$

$$
X_{k+1} = P_{\Omega^c}\left(X_{k+1}\right) + P_\Omega(M).
$$

6:    **until**

$$
\|X_{k+1} - X_k\|_F \leq \varepsilon_0.
$$

Algorithm 2 WHTNR algorithm.

3.3 Convergence analysis

In this subsection, the convergence of the proposed method is discussed. For convenience, the optimal value of (18) is depicted as follows

$$
\begin{aligned}
p_* = \ & \inf\{ \text{Tr}\left(AWX_*B^T \right) - \text{Tr}\left(CH_*D^T \right) + \lambda(\|X_*\|_F^2 - \|CX_*\|_F^2) \\
& : \ X = H, P_\Omega(H) = P_\Omega(M) \},
\end{aligned}
\tag{26}
$$

the unaugmented lagrangian function (19) is also written by

$$
\begin{aligned}
L(X, H, Y) = \ & \text{Tr}\left(A_k WXB_k^T \right) - \text{Tr}\left(C_k HD_k^T \right) + \lambda(\|X\|_F^2 - \|C_k X\|_F^2) \\
& + \langle \frac{\mu}{2}(X - H) + Y, X - H \rangle.
\end{aligned}
\tag{27}
$$

In order to further illustrate the convergence of WHTNR algorithm, the following three lemmas are established, and the proofs are shown in the Appendices A, B and C.

Lemma 1 *If (X_*, H_*, Y_*) is the optimal solution of the unaugmented Lagrangian function (27), then the inequality holds as follows*

$$
(1) \ \ p_* - p_{k+1} \leq \langle \frac{\mu}{2} R_{k+1} + Y_*, R_{k+1} \rangle,
\tag{28}
$$

$$
(2) \ \ p_{k+1} - p_* \leq \langle -Y_{k+1}, R_{k+1} \rangle - \langle \mu(H_{k+1} - H_k), H_{k+1} - H_* + R_{k+1} \rangle.
\tag{29}
$$

where $R_{k+1} = X_{k+1} - H_{k+1}$. In fact, $X_{k+1} = H_{k+1}$, which implies that (X_*, H_*) is also the optimal solution to the model (18).

Lemma 2 *If (X_*, H_*, Y_*) is the optimal solution of the unaugmented Lagrangian function (27), then V_k decreases in each iteration and satisfies the relationship as follows*

$$
V_k - V_{k+1} \geq \mu \left(\|R_{k+1}\|_F^2 + \|H_{k+1} - H_k\|_F^2 \right).
\tag{30}
$$

where $V_k = \frac{1}{\mu}\|Y_k - Y_*\|_F^2 + \mu\|H_{k+1} - H_k\|_F^2$.

In terms of the above three lemmas, we have the following convergence theorem.

Theorem 1 *If (X_*, H_*, Y_*) is the optimal solution of the unaugmented Lagrangian function (27), the iterative solution (X_k, H_k) converges to the optimal solution (X_*, H_*) if $k \to \infty$. In other words, $p_k \to p_*$ as $k \to \infty$*

On the basis of Theorem 1, the convergence of the WHTNR method is explained by the approach of the objective function to the optimal value.

4 Experiments

In this section, extensive experiments are performed on real data to verify the proposed convergence accuracy, speed, and stability of the WHTNR algorithm. All

experiments on Intel(R), Core(TM) i7- MATLAB R2016b running on CPU@2.90 GHz, 32GB main memory. The proposed algorithm WHTNR will be compared with five related algorithms, including (1) SVT [15]; (2) IALM [13]; (3) TNNR-admmap [17]; (4) TNNR-WRE [18]; (5) HTNR [21]; (6) WHTNR (The proposed method).

There are several parameters (weight matrix W, coefficients of the Frobenius norm λ, coefficients of lagrangian functions μ, and the number of truncated singular values r in WHTNR algorithm. W is given by the formula (33). In experiments, an optimal value of r was chosen between 1 and 20.

The maximum number of iterations for WHTNR is 100, and the maximum number of iterations for other comparison algorithms is 100. The stopping condition of the algorithm is

$$\frac{\|X_{k+1} - X_k\|_F}{\|X_{k+1}\|_F} \leq \varepsilon_0. \tag{31}$$

The stop error is set ε to 0.0001.

In these experiments, the peak signal to noise ratio (PSNR) is used as evaluation metrics, for which is defined as follows

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\text{MSE}}\right), \tag{32}$$

where $\text{Erec} = \|(X_{\text{rec}} - M)_{\Omega^c}\|_F$, $\text{SE} = \text{Erec}_R^2 + \text{Erec}_G^2 + \text{Erec}_B^2$, $\text{MSE} = \frac{\text{SE}}{T} = \frac{\text{SE}}{T_R + T_G + T_B}$, T is the number of missing elements, and $\{R, G, B\}$ are the different channels of the real image.

In addition, the stability of the proposed method is characterized by the coefficient of variation of PSNR (CV) and the smaller the CV value, the more stable the result. Except that, we use the running time (CPU time(s)) value under the same conditions to evaluate the speed.

In this section, real images are used to verify the performance of the WHTNR algorithm on matrix computation. In practical applications, real-world images sometimes suffer from noise which causes missing of image information to a certain extent. Approximately, the WHTNR algorithm proposed above can effectively restore the damaged images when the image satisfies the low rank or near low rank. Therefore, in the experiments, the WHTNR method is evaluated below, and three types of pixel occlusion are considered, including (1) random mask; (2) text mask; and (3) block mask. The images used in this section are color images in JPEG format with a dimension of 300×300 as shown in Fig. 1.

4.1 The effect of *W* on WHTNR

If $W = \text{diag}(w_1, \ldots, w_m)$, then w_i is defined as follows

$$w_i = 2 - \theta * \frac{N_i}{m}, \quad i = 1, 2, \ldots, m, \tag{33}$$

where $\theta > 1$, m is the number of rows of the matrix to be restored and N_i is the number of observed entries in the ith row. According to the (33), we have $w_i \leq w_j$
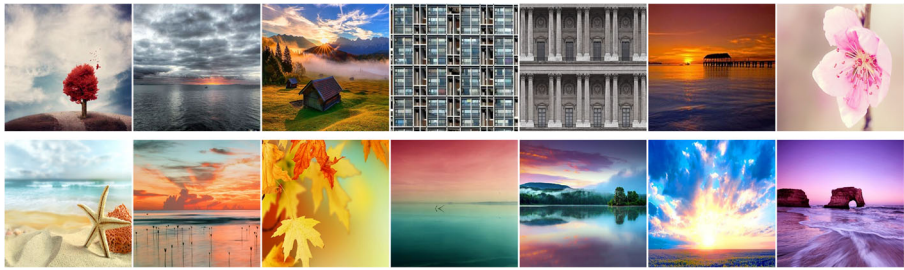
**Fig. 1** The 14 real images used in image denoising experiments

if $N_i \geq N_j$, that is to say, the row with more observed elements is assigned with a smaller value of weight, according to (33). Clearly, $w_i = 0$ denotes that the i-th row has no element lost.

In Fig. 2(b)–(e) illustrate that rows with few missing elements are recovered first. For clarification, Fig. 3(a) depicts the weights applied in Fig. 2. Rows with fewer missing elements have lower weighted values, since they are commonly easier to be restored than the others. Likewise, the bigger blocks require larger values of weights to ensure accuracy. Therefore, the entire process is accelerated by the sequential recovery fashion.

To test the robustness of the WHTNR algorithm when parameter θ varying, we set θ increasing from 0.1 to 3. For different miss rates, if θ is too large, the PSNR will become worse. This is because θ controls the weight range in the row dimension, which affect the recovery of WHTNR. So we choose that $\theta = 1.2$ for our method and apply to subsequent experiments. It shows PSNR is relatively stable when $\theta < 2$, which indicates that WHTNR performs well in robustness when $\theta < 2$ (Fig. 4).

4.2 Convergence of WHTNR

Figure 5 shows the PSNR curve of the first 70 iterations under random occlusion with different missing rates for Fig. 1(a). It can be seen from Fig. 5 that the PSNR value gradually increases in the first 40 iterations, and becomes stable after 40 iterations, which verifies the convergence of the WHTNR algorithm experimentally.

(a)　　　　(b)　　　　(c)　　　　(d)　　　　(e)
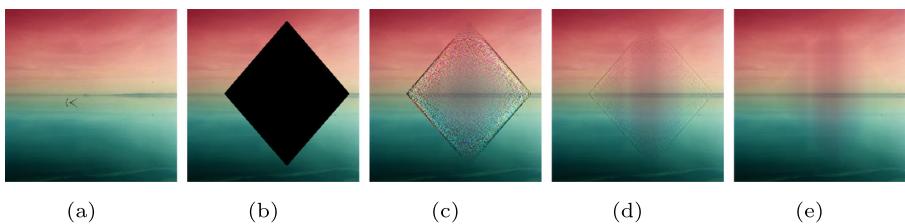
Fig. 2 The optimization process of the WHTNR performed on the image in Fig. 1(k) with the missing diamond block. (a) original image; (b) the incomplete image; (c) the 20th step; (d) the 30th step; (e) the recovered image
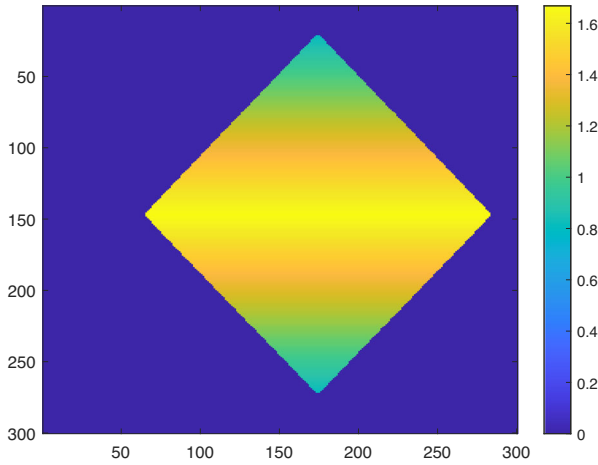
Fig. 3 Weights corresponding to four missing blocks on the image in Fig. 1(a)

4.3 Robustness to *r* of WHTNR

We design an experiment on Fig. 1(a) to verify that the number of truncated singular values r is robust of WHTNR. In Fig. 6, a line graph of PSNR values under different r values is shown and compared with TNNR-admmap, TNNR-WRE, and HTNR methods. As can be seen from Fig. 6, when parameter r varying, WHTNR outperforms other methods in terms of robustness, because WHTNR does not completely truncate the first r singular values, but assigns weights (similar to ETNNR-WRE algorithm [18]), which can optimize the convergence accuracy based on the nuclear norm.
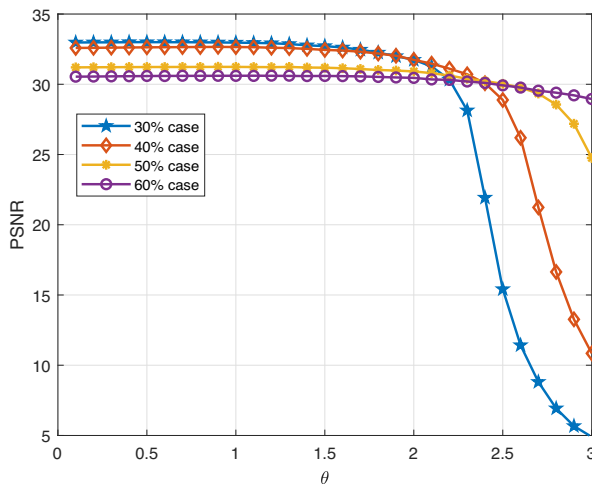

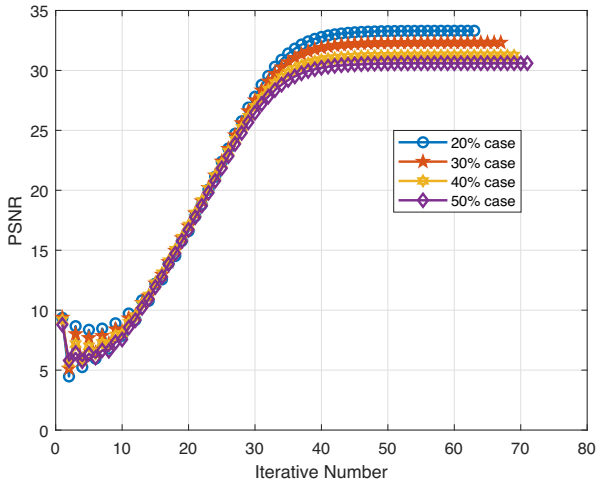**Fig. 4** The effect of θ for W on WHTNR on the image in Fig. 1(a)

Fig. 5 The convergence analysis on WHTNR on the image in Fig. 1(a)

4.3.1 Random mask

Random noise refers to missing information randomly distributed in the image, which is a relatively simple matrix computation problem. In the experiment, we mask 50%, 60%, 70% elements randomly in the 14 test images in Fig. 1 and then use SVT, IALM, TNNR-admmap, TNNR-WRE, HTNR and the WHTNR algorithm proposed in this paper for image restoration and compare their performance. By experience, we set the following settings for the parameters respectively, $\theta = 1.2$, $\rho = 1.15$, $\lambda = 0.001$, $\beta = 1.05$ and $\mu = 0.0001$. The WHTNR algorithm and the
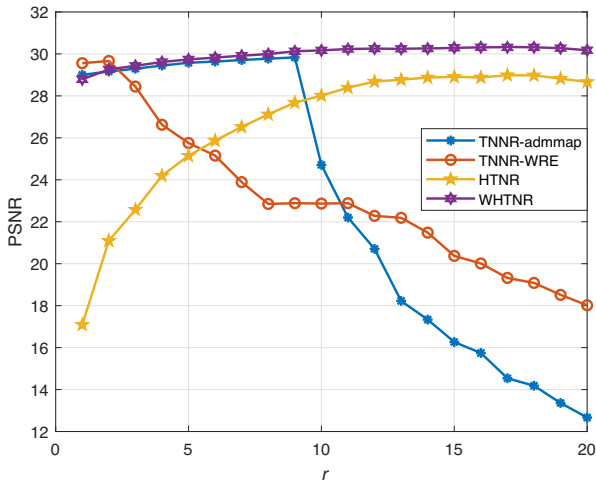

Fig. 6 The comparison of robustness to r by SVT, IALM, TNNR-admmap, TNNR-WRE, HTNR and WHTNR on the image in Fig. 1(a)

Table 1 Observed ratio = 0.5, PSNR of recovery images of the images with random distributed missing entries, bold represents the best performance

Images	SVT	IALM	TNNR-admmap	TNNR-WRE	HTNR	WHTNR
1	28.1121	29.8072	30.1705	29.9151	29.0670	**30.6468**
2	28.1099	31.2981	31.1682	31.4656	30.4860	**32.1641**
3	24.1461	24.4312	24.6214	24.6217	24.0376	**24.9736**
4	24.3065	25.2049	24.9564	25.3303	23.3163	**26.0943**
5	27.2948	28.5808	28.8148	28.6208	27.9988	**28.9238**
6	30.7216	35.2434	34.7264	35.2932	33.8347	**36.4625**
7	27.8719	29.6775	29.3014	29.9007	28.1114	**30.2254**
8	25.7343	26.5771	26.7358	26.9210	25.7986	**27.0819**
9	27.9036	30.9808	30.6419	31.0850	28.9421	**31.5581**
10	26.9251	28.7535	28.7090	28.9679	27.9070	**29.4754**
11	36.9962	42.8085	42.7938	42.8338	40.3267	**42.9238**
12	28.5392	30.7617	30.7667	30.8090	30.1978	**31.6983**
13	26.5124	28.3927	28.0051	28.3326	26.9470	**28.9840**
14	25.8188	26.8950	26.9456	27.1963	26.3043	**27.6408**

quantization results of the convergence accuracy of the other 5 algorithms are shown in Table 1. The original image, the noise image and the restored image of different methods are shown in Fig. 7. The CPU time for different methods is shown in Fig. 8. As can be seen from Table 1, WHTNR has the best quality of the restored image,
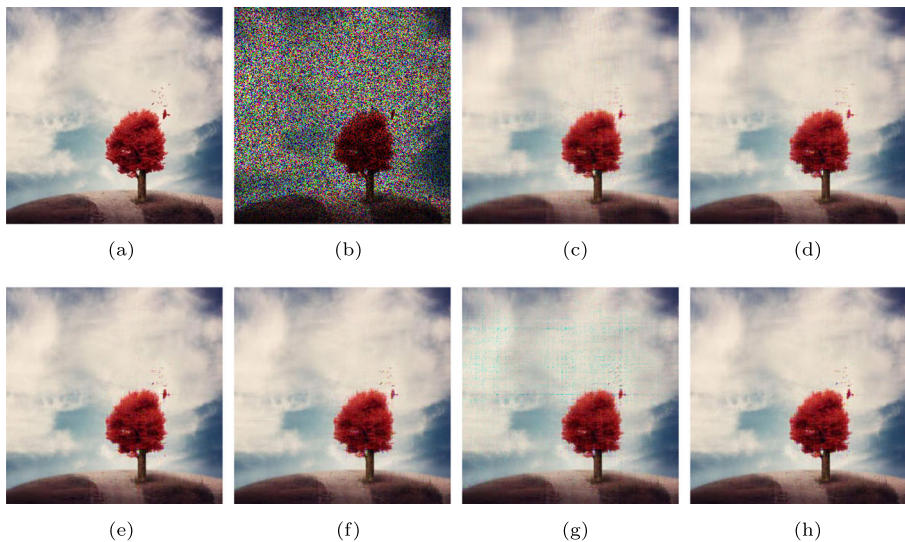
Fig. 7 Recovered results of the image in Fig. 1(a) with random mask (missing ratio = 0.5) by six methods: (a) the original image, (b) the observed image, (c) SVT, (d) IALM, (e) TNNR-admmap, (f)TNNR-WRE, (g) HTNR, (h) WHTNR

Fig. 8 Observed ratio = 0.5, CPU time(s) of recovery images of the images with random distributed missing entries
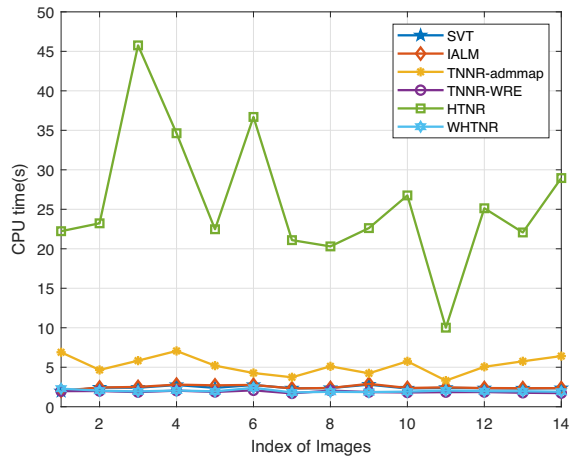
which is due to that WHTNR does not completely truncate the first r singular values. As can be seen from Fig. 6, WHTNR is robust to parameter r. As can be seen from Table 4, WHTNR retains the advantages of the good stability of the HTNR algorithm without affecting the convergence accuracy. As can be seen from Fig. 7, WHTNR can basically restore the original appearance of the image compared with other algorithms, and its image details are clearer. It can be seen from Fig. 8 that the WHTNR algorithm greatly improves the convergence speed based on the HTNR algorithm. Therefore, WHTNR has certain advantages in terms of convergence accuracy, convergence speed, stability, and robustness of parameter r (Tables 2, 3 and 4).

Table 2 Observed ratio = 0.6, PSNR of recovery images of the images with random distributed missing entries, bold represents the best performance

Images	SVT	IALM	TNNR-admmap	TNNR-WRE	HTNR	WHTNR
1	27.0935	28.6233	28.4487	28.5687	28.1066	**29.4741**
2	27.2189	30.0333	29.3451	29.8436	29.6091	**30.8692**
3	23.2543	23.4174	23.2163	23.5796	23.2391	**24.1128**
4	22.9236	23.6558	22.8710	23.4552	22.4605	**24.7131**
5	26.6926	27.6678	27.7733	27.7670	27.2919	**28.1621**
6	29.7115	33.5088	32.8846	33.3185	32.1052	**34.4100**
7	26.8535	28.4994	28.0022	28.5661	27.0452	**29.0369**
8	24.8885	25.7333	25.4477	25.7641	24.9006	**26.2801**
9	26.9378	29.6417	28.9745	29.5486	27.9510	**30.1570**
10	25.8532	27.5152	26.9737	27.4630	26.5333	**28.3397**
11	36.7861	41.5232	41.5248	41.6279	40.0552	**41.7842**
12	27.5205	29.5540	29.3376	29.2040	28.9932	**30.4289**
13	25.3527	26.9629	25.8602	26.5336	25.8116	**27.6976**
14	24.8841	25.6161	25.2616	25.8585	25.2571	**26.4201**

Table 3 Observed ratio = 0.7, PSNR of recovery images of the images with random distributed missing entries, bold represents the best performance

Images	SVT	IALM	TNNR-admmap	TNNR-WRE	HTNR	WHTNR
1	25.6865	27.3416	26.2852	26.7112	26.6568	**28.0530**
2	25.7377	28.0747	26.6681	27.3292	27.2364	**28.8589**
3	22.1677	22.2307	21.6337	22.1372	21.8867	**22.9974**
4	21.2437	21.8698	19.9338	21.3424	20.1566	**22.8227**
5	25.8754	26.6403	26.4699	26.6254	26.5676	**27.2489**
6	28.5542	31.809	30.2775	31.2284	30.1341	**32.4142**
7	25.7401	27.2122	26.3433	26.8773	25.9903	**27.5904**
8	23.7845	24.6063	23.8092	24.3867	23.7739	**25.1674**
9	25.6807	28.0137	26.6184	27.4422	26.7832	**28.5099**
10	24.5153	25.9682	24.3972	25.3951	25.3495	**26.7235**
11	36.4801	40.2858	40.4219	40.7171	39.1148	**40.9232**
12	26.0421	28.0813	26.3698	27.3454	26.9642	**28.7660**
13	24.0608	25.2519	23.2416	24.5447	24.4054	**26.0243**
14	23.7061	24.2434	23.3845	23.9151	23.9917	**25.0542**

4.3.2 Text mask

Unlike random noise, text noise continuously distributed in part of the image. In this paper, a matrix computation algorithm is used to restore low-rank images contaminated by text noise, and text is regarded as the missing element in the matrix computation problem. In this section, Fig. 1(d) and (k) are arbitrarily selected as examples, the text noise is printed, and use SVT, IALM, TNNR-admmap, TNNR-WRE, HTNR and WHTNR to recover the image. The quantized results of its convergence accuracy are shown in Table 5. In Fig. 9 original image, noise image and restored image of different methods are shown. It can be seen from Table 5 that in terms of quantization results, the WHTNR algorithm significantly outperforms the SVT and IALM methods, and slightly better than the TNNR-admmap, TNNR-WRE, and HTNR algorithms. As can be seen from Fig. 9, the upper part of the SVT and IALM results still contains noise, while the edge of the restored image by WHTNR algorithm is clearer.

Table 4 Observed ratio = 0.5, PSNR and CV of recovery images of the images with random distributed missing entries, bold represents the best performance and the italic stands for suboptimal.

| Images	Index	SVT	IALM	TNNR-admmap	TNNR-WRE	HTNR	WHTNR
13	PSNR	26.5124	28.3927	28.0051	28.3326	26.9470	**28.9840**
	CV	**0.1546%**	0.2127%	0.4160%	0.4939%	0.3294%	*0.1943%*
14	PSNR	25.8188	26.8950	26.9456	27.1963	26.3043	**27.6408**
	CV	**0.2213%**	0.2431%	0.3216%	0.6244%	0.3558%	*0.2266%*

**Table 5** PSNR of recovery images of the images with text or block missing entries, bold represents the best performance

Images	Mask	SVT	IALM	TNNR-admmap	TNNR-WRE	HTNR	WHTNR
2	text	29.1588	28.2572	34.3488	35.2722	33.0369	**36.3247**
3	text	23.4513	22.9871	25.8909	26.0748	26.0748	**26.4935**
4	block	21.7347	22.3297	22.3558	22.5553	21.5871	**22.6689**
5	block	23.7514	25.1105	25.6866	26.2994	23.3491	**28.5427**

4.3.3 Block mask

As mentioned earlier, text noise is continuously distributed in the image area, while block noise is continuously distributed in the larger area, which makes it more difficult to restore contaminated images with matrix padding. In this paper, a matrix computation algorithm is used to restore low-rank images contaminated by text noise, and text elements are regarded as missing elements in the matrix computation problem. Figure 1(e) and (f) are arbitrarily chosen in this section as examples, print block noise, and use SVT, IALM, TNNR-admmap, TNNR-WRE, HTNR and HTNR-wre to recover the image. The quantized results of its convergence accuracy are shown in Table 5. In Fig. 10 original image, noise image and restored image of different methods are shown. From Table 5, WHTNR algorithm has better recovery effect than other algorithms. As can be seen from Fig. 10, the WHTNR algorithm can basically restore the visual effect of the image even under the more difficult block noise.
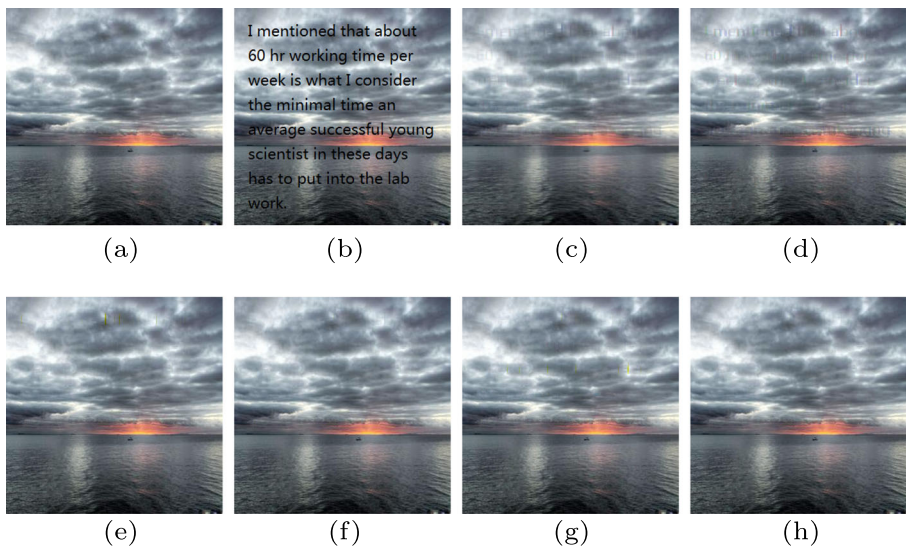
Fig. 9 Recovered results of the image in Fig. 1(b) with text mask by six methods: (a) the original image, (b) the observed image, (c) SVT, (d) IALM, (e) TNNR-admmap, (f) TNNR-WRE, (g) HTNR, (h) WHTNR
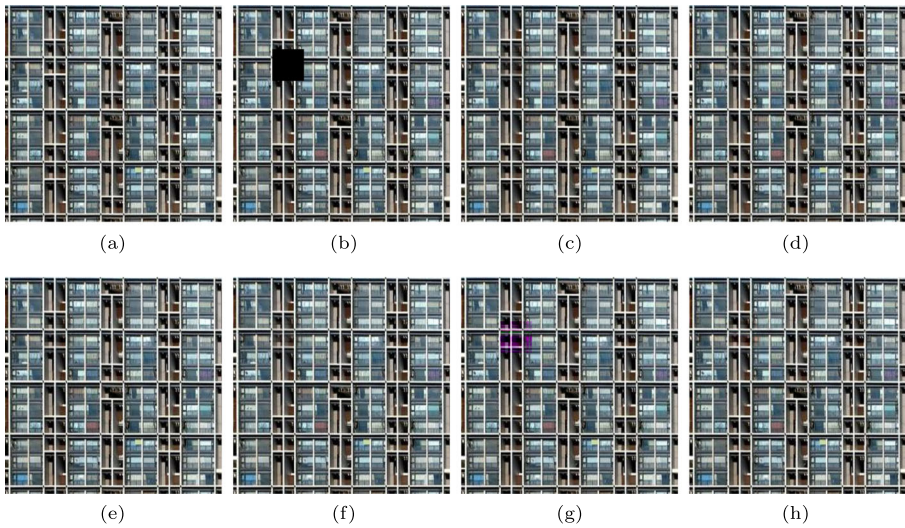
Fig. 10 Recovered results of the image in Fig. 1(d) with block mask by six methods: (a) the original image, (b) the observed image, (c) SVT, (d) IALM, (e) TNNR-admmap, (f) TNNR-WRE, (g) HTNR, (h) WHTNR

5 Conclusions

In this paper, we introduce a new low-rank matrix completion method called the weighted residual-based hybrid truncated norm method (WHTNR). The WHTNR model is used for matrix completion. By assigning appropriate weights to the first r singular values, the rows with fewer missing elements in the matrix can be restored preferentially. At the same time, we verify the convergence of the WHTNR algorithm both theoretically and experimentally. Quantization results and visual effects on real images, illustrating the advantages of WHTNR compared to other methods. Therefore, the WHTNR algorithm is based on the HTNR algorithm, and on the basis of ensuring its stability, compared with the closest 5 methods, it has higher convergence accuracy and convergence speed, and is more robust to the parameter truncate singular values r.

Appendix A: Proof of Lemma 1

First, Lemma 1 (1) is discussed. On the basis of that (X_*, H_*, Y_*) is the optimal solution to the Lagrange function (19). Hence, we have

$$L(X_*, H_*, Y_*) \leq L(X_{k+1}, H_{k+1}, Y_*). \tag{34}$$

Clearly, $X_* = H_*$, then the left-hand side is p_*. And $p_{k+1} = \text{Tr}\left(AWX_{k+1}B^T\right) - \text{Tr}\left(CH_{k+1}D^T\right) + \lambda\left(\|X_{k+1}\|_F^2 - \|CX_{k+1}\|_F^2\right)$, (34) is rewritten to $p_* \leq p_{k+1} + \langle\frac{\mu}{2}R_{k+1} + Y_*, R_{k+1}\rangle$, the inequality (34) holds.

Thus, Lemma 1 (1) is proved. Then Lemma1(2) is discussed.

Clearly, X_{k+1} minimizes $L(X, H_k, Y_k)$ by definition. Since $L(X, H_k, Y_k)$ is closed, convex and subdifferentiable on X. On the basis of the property of subdifferential [23], the optimality condition is obtained as follows

$$
\begin{aligned}
0 &\in \partial L(X_{k+1}, H_k, Y_k) \\
&= \partial \left[\text{Tr}\left(AWX_{k+1}B^T\right) - \text{Tr}\left(CH_kD^T\right) + \lambda \left(\|X_{k+1}\|_F^2 - \|CX_{k+1}\|_F^2 \right) \right] \quad (35) \\
&\quad + \mu(X_{k+1} - H_k) + Y_k.
\end{aligned}
$$

Since $Y_{k+1} = Y_k + \mu R_{k+1}$, we have $Y_k = Y_{k+1} - \mu R_{k+1}$, (35) is rewritten as follows

$$
0 \in \partial \left[\text{Tr}\left(AWX_{k+1}B^T\right) + \lambda \left(\|X_{k+1}\|_F^2 - \|CX_{k+1}\|_F^2 \right) \right] \quad (36)
$$
$$
+ \mu(H_{k+1} - H_k) + Y_{k+1}. \quad (37)
$$

note that X_{k+1} minimizes

$$
\text{Tr}\left(AWXB^T\right) + \lambda \left(\|X\|_F^2 - \|CX\|_F^2 \right) + \langle \mu(H_{k+1} - H_k) + Y_{k+1}, X \rangle. \quad (38)
$$

In the same way, H_{k+1} minimizes $L(X_{k+1}, H, Y_k)$ by definition. Since $L(X_{k+1}, H, Y_k)$ is closed, convex and subdifferentiable on H. On the basis of the property of subdifferential [23], the optimality condition is obtained as follows,

$$
\begin{aligned}
0 &\in \partial L(X_{k+1}, H_{k+1}, Y_k)) \\
&= \partial[-\text{Tr}\left(CH_{k+1}D^T\right)] - \mu(X_{k+1} - H_{k+1}) - Y_k.
\end{aligned} \quad (39)
$$

Since $Y_{k+1} = Y_k + \mu R_{k+1}$, we have $Y_k = Y_{k+1} - \mu R_{k+1}$, (39) is rewritten as follows

$$
0 \in \partial[-\text{Tr}\left(CH_{k+1}D^T\right)] - Y_k. \quad (40)
$$

note that H_{k+1} minimizes

$$
-\text{Tr}\left(CHD^T\right) - \langle Y_{k+1}, H \rangle. \quad (41)
$$

Hence, for the optimal solution (X_*, H_*), we have

$$
\begin{aligned}
&\text{Tr}\left(AWX_{k+1}B^T\right) + \lambda \left(\|X_{k+1}\|_F^2 - \|CX_{k+1}\|_F^2 \right) + \langle \mu(H_{k+1} - H_k) + Y_{k+1}, X_{k+1} \rangle \\
&\leq \text{Tr}\left(AWX_*B^T\right) + \lambda \left(\|X_*\|_F^2 - \|CX_*\|_F^2 \right) + \langle \mu(H_{k+1} - H_k) + Y_{k+1}, X_* \rangle.
\end{aligned} \quad (42)
$$

and

$$
-\text{Tr}\left(CH_{k+1}D^T\right) - \langle Y_{k+1}, H_{k+1} \rangle \leq -\text{Tr}\left(CH_*D^T\right) - \langle Y_{k+1}, H_* \rangle. \quad (43)
$$

With (42), (43), and $X_* = H_*$, we have

$$
\begin{aligned}
p_{k+1} - p_* &\le -\langle \mu(H_{k+1} - H_k) + Y_{k+1}, X_{k+1} - X_* \rangle + \langle Y_{k+1}, H_{k+1} - H_* \rangle \\
&= -\langle \mu(H_{k+1} - H_k) + Y_{k+1}, H_{k+1} - H_* + R_{k+1} \rangle \\
&\quad + \langle Y_{k+1}, H_{k+1} - H_* \rangle \\
&= \langle -Y_{k+1}, R_{k+1} \rangle - \langle \mu(H_{k+1} - H_k), H_{k+1} - H_* + R_{k+1} \rangle.
\end{aligned}
\tag{44}
$$

Thus, Lemma 1(2) is also proved.

Appendix B: Proof of Lemma 2

Add the two inequalities in Lemma 1, multiply both sides by 2, we have

$$
\begin{aligned}
&\langle \frac{\mu}{2} R_{k+1} + Y_*, R_{k+1} \rangle + \langle -Y_{k+1}, R_{k+1} \rangle \\
&- \langle \mu(H_{k+1} - H_k), H_{k+1} - H_* + R_{k+1} \rangle \ge 0.
\end{aligned}
\tag{45}
$$

the inequality (45) is rewritten as follows

$$
\begin{aligned}
&2\langle Y_{k+1} - Y_*, R_{k+1} \rangle - \mu \|R_{k+1}\|_F^2 - 2\mu\langle H_{k+1} - H_k, R_{k+1} \rangle \\
&+ 2\mu\langle H_{k+1} - H_k, H_{k+1} - H_* \rangle \le 0.
\end{aligned}
\tag{46}
$$

Since $Y_{k+1} = Y_k + \mu R_{k+1}$, we have

$$
\begin{aligned}
&2\langle Y_{k+1} - Y_*, R_{k+1} \rangle \\
&= 2\langle Y_k + \mu R_{k+1} - Y_*, R_{k+1} \rangle \\
&= 2\langle Y_k - Y_*, R_{k+1} \rangle + \mu \|R_{k+1}\|_F^2 + \mu \|R_{k+1}\|_F^2 \\
&= \frac{2}{\mu}\langle Y_k - Y_*, Y_{k+1} - Y_k \rangle + \frac{1}{\mu}\|Y_{k+1} - Y_k\|_F^2 + \mu \|R_{k+1}\|_F^2.
\end{aligned}
\tag{47}
$$

Since $Y_{k+1} - Y_k = (Y_{k+1} - Y_*) - (Y_k - Y_*)$, we have

$$
2\langle Y_{k+1} - Y_k, Y_k - Y_* \rangle = \|Y_{k+1} - Y_*\|_F^2 - \|Y_{k+1} - Y_k\|_F^2 - \|Y_k - Y_*\|_F^2.
\tag{48}
$$

The inequality (47) is rewritten as follows

$$
2\langle Y_{k+1} - Y_*, R_{k+1} \rangle = \frac{1}{\mu}\left(\|Y_{k+1} - Y_*\|_F^2 - \|Y_k - Y_*\|_F^2 \right) + \mu \|R_{k+1}\|_F^2.
\tag{49}
$$

Since $H_k - H_{k+1} = (H_k - H_*) - (H_{k+1} - H_*)$, we have

$$
2\langle H_k - H_{k+1}, H_{k+1} - H_* \rangle = \|H_k - H_*\|_F^2 - \|H_k - H_{k+1}\|_F^2 - \|H_{k+1} - H_*\|_F^2.
\tag{50}
$$

The inequality (47) is rewritten as follows

$$
\frac{1}{\mu}\left(\|Y_{k+1} - Y_*\|_F^2 - \|Y_k - Y_*\|_F^2\right) + \mu\left(\|H_{k+1} - H_*\|_F^2 - \|H_k - H_*\|_F^2\right)
$$
$$
+ \mu\|H_k - H_{k+1}\|_F^2 + 2\mu\langle H_k - H_{k+1}, R_{k+1}\rangle \tag{51}
$$
$$
= V_{k+1} - V_k + \mu\|H_k - H_{k+1} + R_{k+1}\|_F^2
$$
$$
\leq 0.
$$

In other words,

$$
V_k - V_{k+1} \geq \mu\|H_k - H_{k+1} + R_{k+1}\|_F^2 \geq 0. \tag{52}
$$

Hence, V_k decreases. Clearly, in order to prove (30), it only need to verify the $-2\langle R_{k+1}, H_k - H_{k+1}\rangle \geq 0$.

In fact, recalling that H_{k+1} minimizes $-\mathrm{Tr}\left(CHD^T\right) - \langle Y_{k+1}, H\rangle$ and H_k minimizes $-\mathrm{Tr}\left(CHD^T\right) - \langle Y_k, H\rangle$, we have

$$
-\mathrm{Tr}\left(CH_{k+1}D^T\right) - \langle Y_{k+1}, H_{k+1}\rangle \leq -\mathrm{Tr}\left(CH_kD^T\right) - \langle Y_{k+1}, H_k\rangle. \tag{53}
$$

and

$$
-\mathrm{Tr}\left(CH_kD^T\right) - \langle Y_k, H_k\rangle \leq -\mathrm{Tr}\left(CH_{k+1}D^T\right) - \langle Y_k, H_{k+1}\rangle. \tag{54}
$$

Add the two inequalities (53) and (53), we have

$$
\langle Y_{k+1} - Y_k, H_{k+1} - H_k\rangle \leq 0. \tag{55}
$$

With $Y_{k+1} - Y_k = \mu R_{k+1}$ and $\mu > 0$, then the inequality (55) is rewritten to $-2\mu\langle R_{k+1}, H_k - H_{k+1}\rangle \geq 0$

Thus, Lemma 2 is proved.

Appendix C: Proof of Theorem 1

In terms of Lemma 2, V_k decreases in each iteration and $V_k - V_{k+1} \geq \mu\left(\|R_{k+1}\|_F^2 + \|H_{k+1} - H_k\|_F^2\right)$, add all the terms on the both sides and rearrange it, we have

$$
\sum_{k=1}^{\infty}\left(\mu\left(\|R_{k+1}\|_F^2 + \|H_{k+1} - H_k\|_F^2\right)\right) \leq \sum_{k=1}^{\infty}(V_k - V_{k+1}) \leq V_1 < \infty.
$$

Notes that $\mu\left(\|R_{k+1}\|_F^2 + \|H_{k+1} - H_k\|_F^2\right) \to 0(k \to \infty)$. Hence, if $k \to \infty$, then $R_{k+1} \to 0$ and $H_{k+1} - H_k \to 0$. In view of Theorem 1, we have

$$
p_* - p_{k+1} \leq \langle\frac{\mu}{2}R_{k+1} + Y_*, R_{k+1}\rangle \to 0(k \to \infty).
$$

and

$$p_{k+1} - p_* \leq \langle -Y_{k+1}, R_{k+1} \rangle - \langle \mu(H_{k+1} - H_k), H_{k+1} - H_* + R_{k+1} \rangle$$
$$\rightarrow 0(k \rightarrow \infty).$$

That is to say, $p_k \rightarrow p_*$ as $k \rightarrow \infty$.

Thus, Theorem 1 is proved.

Conflict of interest The authors declare no competing interests.

References

1. Ma, T.H., Lou, Y., Huang, T.Z.: Truncated l_{1-2} models for sparse recovery and rank minimization. SIAM J. Imaging Sci. **10**(3), 1346–1380 (2017)
2. Zhao, X.L., Wang, F., Huang, T.Z., et al.: Deblurring and sparse unmixing for hyperspectral images. IEEE Trans. Geosci. Remote Sens. **51**(7), 4045–4058 (2013)
3. Zhao, X.L., Xu, W.H., Jiang, T.X., et al.: Deep plug-and-play prior for low-rank tensor completion. Neurocomputing **400**, 137–149 (2020)
4. Zhao, X.L., Zhang, H., Jiang, T.X., et al.: Fast algorithm with theoretical guarantees for constrained low-tubal-rank tensor recovery in hyperspectral images denoising. Neurocomputing **413**, 397–409 (2020)
5. Jannach, D., Resnick, P., Tuzhilin, A., et al.: Recommender systems—beyond matrix completion. Commun. ACM **59**(11), 94–102 (2016)
6. Ramlatchan, A., Yang, M., Liu, Q., et al.: A survey of matrix completion methods for recommendation systems. Big Data Mining and Analytics **1**(4), 308–323 (2018)
7. Wang, W., Chen, J., Wang, J., et al.: Geography-aware inductive matrix completion for personalized Point-of-Interest recommendation in smart cities. IEEE Internet Things J. **7**(5), 4361–4370 (2019)
8. Candes, E.J., Plan, Y.: Matrix completion with noise. Proc. IEEE **98**(6), 925–936 (2010)
9. Zou, C., Hu, Y., Cai, D., et al.: Salient object detection via fast iterative truncated nuclear norm recovery. In: International Conference on Intelligent Science and Big Data Engineering, pp. 238–245. Springer, Berlin (2013)
10. Wright, J., Ganesh, A., Rao, S., et al.: Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. Advances in Neural Information Processing Systems 22 (2009)
11. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found. Comput. Math. **9**(6), 717–772 (2009)
12. Fazel, M.: Matrix Rank Minimization with Applications. PhD thesis, Stanford University (2002)
13. Lin, Z., Chen, M., Ma, Y.: The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv:1009.5055 (2010)
14. Toh, K.C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pacific Journal of Optimization **6**(3), 615–640 (2010)
15. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
16. Zhang, D., Hu, Y., Ye, J., et al.: Matrix completion by truncated nuclear norm regularization. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2192–2199. IEEE (2012)
17. Hu, Y., Zhang, D., Ye, J., et al.: Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. **35**(9), 2117–2130 (2013)
18. Liu, Q., Lai, Z., Zhou, Z., et al.: A truncated nuclear norm regularization method based on weighted residual error for matrix completion. IEEE Trans. Image Process. **25**(1), 316–330 (2015)

19. Xue, S., Qiu, W., Liu, F., et al.: Double weighted truncated nuclear norm regularization for low-rank matrix completion. arXiv:1901.01711 (2019)
20. Yang, L., Kou, K.I., Miao, J.: Weighted truncated nuclear norm regularization for low-rank quaternion matrix completion. J. Vis. Commun. Image Represent. **81**, 103335 (2021)
21. Ye, H., Li, H., Cao, F., et al.: A hybrid truncated norm regularization method for matrix completion. IEEE Trans. Image Process. **28**(10), 5171–5186 (2019)
22. Mirsky, L.: A trace inequality of John von Neumann. Monatshefte für mathematik **79**(4), 303–306 (1975)
23. Tyrrell, R., Fellar, R.: Convex analysis. Princeton University Press (1996)