Neurocomputing 172 (2016) 253–261

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Convex optimization based low-rank matrix decomposition for image restoration

Ning He [a,*], Jin-Bao Wang [a], Lu-Lu Zhang [a], Ke Lu [b]

[a] College of Information Technology, Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China
[b] University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 28 November 2013
Received in revised form
30 October 2014
Accepted 1 November 2014
Available online 9 May 2015

Keywords:
Image restoration
Low-rank matrix
Principal component pursuit
Singular value thresholding

ABSTRACT

This paper addresses the problem of image denoising in the presence of significant corruption. Our method seeks an optimal set of image domain transformations such that the matrix of transformed images can be decomposed as the sum of a sparse matrix of errors and a low-rank matrix of recovered denoised images. We reduce this optimization problem to a sequence of convex programs minimizing the sum of the ℓ_1 – norm and the nuclear norm of the two component matrices, which can be solved efficiently using scalable convex optimization techniques. We verify the efficacy of the proposed image denoising algorithm through extensive experiments on both numerical simulations and different types of images, demonstrating its highly competent objective performance compared with several state-of-the-art methods for matrix decomposition and image denoising. Our subjective quality results compare favorably with those obtained by existing techniques, especially at high noise levels and with a large amount of missing data.

1. Introduction

In recent years, image restoration has been an important, yet challenging problem widely studied in computer vision and image processing. The purpose of image restoration is to "compensate for" or "undo" defects that degrade an image. Degradation comes in many forms, including motion blur, noise, and camera misfocus. In cases like motion blur, it is possible to obtain a very good estimate of the actual blurring function and "undo" the blur to restore the original image. However, in cases where the image is corrupted by noise, the best we can hope to achieve is to compensate for the resulting degradation. Owing to the ill-posed nature of image restoration, an image restoration solution is generally not unique. To find a better solution, prior knowledge of images can be used to regularize the image restoration problem. One of the most commonly used regularization models is the total variation (TV) model [1,2]. Since the TV model favors piecewise constant image structures, it tends to smooth out the finer details of an image. To better preserve the image edges, algorithms have subsequently been developed to improve the TV models [3,4,5].

The success of TV regularization validates the importance of good image prior models in solving image restoration problems. In wavelet based image denoising [6], researchers have found that the sparsity of wavelet coefficients can serve as a good prior. This implies that many types of signals, e.g., natural images, can be sparsely represented using a dictionary of atoms, such as discrete cosine transforms (DCT) or wavelet bases. In addition, recent studies have shown that iteratively reweighting the ℓ_1-norm sparsity regularization term can lead to better image restoration results [7]. Sparse representation has been successfully used in various image processing applications [8,9,10]. However, sparse decomposition over a highly redundant dictionary is potentially unstable and tends to generate visual artifacts [11,12]. Recently, a representative study introduced low-rank matrix recovery (LR) theory [13] into image restoration. Existing LR-based image denoising models share a common assumption that an image can be represented as a highly redundant information part (e.g., background regions) plus a main part (e.g., the foreground object) including several homogeneous regions. The redundant information part usually lies in a low dimensional feature subspace, which can be approximated as a low-rank feature matrix, whereas the main part can be viewed as a sparse sensory matrix. In this paper, we introduce a new method, the matrix rank minimization image restoration algorithm. Our solution builds on recent advances in rank minimization and formulates the image restoration problem as a solution that connects low-rank methods with simultaneous sparse coding. We utilize the low-rank matrix convex optimization scheme to estimate the local sparsity of the image and adjust the

* Corresponding author.
E-mail addresses: xxthening@buu.edu.cn (N. He), luk@ucas.ac.cn (K. Lu).

http://dx.doi.org/10.1016/j.neucom.2014.11.090
0925-2312/

sparsity regularization parameters. Extensive experiments on image denoising show that the proposed approach can effectively reconstruct the image details.

The remainder of this paper is organized as follows: In Section 2, we introduce the related matrix rank optimization technique and an analysis of existing problems. In Section 3, we introduce matrix rank as a measure of image similarity and reformulate the image denoising problem as one of matrix rank minimization. In addition, we propose an efficient algorithm to solve the rank minimization problem by iterative convex optimization. Experimental results showing the efficacy of our method on numerical simulations and different types of images are presented in Section 4.

2. Related work

In recent years, the search for more scalable algorithms for high-dimensional convex optimization problems has prompted a return to first-order methods. Principal component analysis (PCA) is a popular tool for high-dimensional data analysis, with applications ranging across a wide variety of scientific and engineering fields. It relies on the basic assumption that the given high-dimensional data lie in a much lower-dimensional linear subspace. Correctly estimating this subspace is crucial for reducing the dimension of the data and facilitating tasks such as processing [23,29], analyzing [22,28], compressing, or visualizing the data [14,15]. Suppose that the given data are arranged as columns in a large matrix, $D \in R^{m\times n}$. Classical PCA assumes that this data matrix was generated by perturbing a matrix, $A \in R^{m\times n}$, whose columns lie in a subspace of dimension $r << \min(m,n)$. In other words, $D = A+E$, where A is a rank-r matrix and E is a matrix whose entries are independent and identically distributed (i.i.d.) Gaussian random variables. In this setting, PCA seeks an optimal estimate of A, by the following constrained optimization:

$$\min_{A,E} \|E\|_F, \text{s.t. } \operatorname{rank}(A) \le r, D = A+E \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm. It is well-known that this problem can be solved efficiently by simply computing the singular value decomposition (SVD) of D. The optimal estimate of low-rank matrix A is simply the projection of the columns of D onto the subspace spanned by the r principal left singular vectors of D [16].

Although PCA offers an optimal estimate of the subspace for data corrupted by small amounts of Gaussian noise, it breaks down under large corruption, even if that corruption affects only a few of the observations. This undesirable behavior motivated the study of the problem of recovering a low-rank matrix A from a corrupted data matrix $D = A+E$, where some entries of E may be of arbitrarily large magnitude.

Recently, the authors in [17] showed that under surprisingly broad conditions, one can recover low-rank matrix A exactly from $D = A+E$ with gross but sparse errors E, by solving the following convex optimization problem:

$$\min_{A,E} \|A\|_* + \lambda \|E\|_1 \quad \text{s.t.} D = A+E \tag{2}$$

where the nuclear norm $\|\cdot\|_*$ (sum of the singular values of a matrix) is a convex relaxation of the matrix rank function, $\|\cdot\|_1$ denotes l_1-norm, which promotes sparsity, and parameter $\lambda > 0$ is a trade-off between the two items. In [18], this optimization is robust PCA (RPCA), because it enables one to correctly recover the underlying low-rank structure in the data, even in the presence of gross errors or outlying observations. This optimization can easily be reformulated as a semi-definite program and solved by an off-the-shelf interior point solver. However, although interior point methods offer superior convergence rates, the complexity of computing the step direction is $O(n^6)$, and thus they do not scale well with the size of the matrix.

One striking example of this is the current popularity of iterative thresholding algorithms for ℓ_1-norm minimization problems arising in compressed sensing [19,20]. Similar iterative thresholding techniques [21,24] can be applied to the problem of recovering a low-rank matrix from an incomplete subset of its entries [25,26,27]. This optimization is closely related to the RPCA problem, and the convergence proof extends quite naturally to RPCA. However, the iterative thresholding scheme proposed in [30] exhibits extremely slow convergence; solving one instance requires about 10^4 iterations, each of which has the same cost as one SVD. Hence, even for matrix sizes as small as 800×800, the algorithm requires more than 8 h on a typical PC.

In this paper, our goal is to develop faster and more scalable algorithms, by further studying the convex optimization problem in Eq. (2) associated with RPCA and applied to the image restoration problem.

3. Image restoration by matrix rank minimization

In this section, we formulate image restoration as a search for a set of transformations minimizing the rank of the transformed images, viewed as the columns of a matrix. We discuss why rank is a natural measure of image similarity, and how this conceptual framework can be made robust to gross errors due to corruption or occlusion.

3.1. Low-rank matrix structure and iterative decomposition

Measuring the degree of similarity within a set of images is a fundamental problem in computer vision and image processing. Consider matrix $X \in R^{m\times n}$ constructed by stacking all the vectorized images, denoted by $\operatorname{vec}(I_k)$, as $X = [\operatorname{vec}(I_1)|\cdots|\operatorname{vec}(I_n)]$, where $\operatorname{vec}(I_j) = [I_j(1),\cdots,I_j(m)]^T$ for $j = 1,\cdots,n$. It follows that X can be factorized as $X = NL$, where $N = [\rho_1 n_1|\cdots|\rho_m n_m]^T \in R^{m\times 3}$ and $L = [l_1|\cdots|l_n] \in R^{3\times n}$. Suppose that the number of images is $n \ge 3$. Irrespective of the size of n, the rank of matrix X is clearly at most 3.

Notions: Let $X = (x_1,\cdots,x_n)$ be an $m \times n$ matrix, $\Omega \subset \{1,\cdots,m\} \times \{1,\cdots,n\}$ denote the indices of the observed entries of X, and Ω^c denote the indices of the missing entries. The Frobenius norm of X is defined as $\|X\|_F^2 = \sqrt{\sum_{(i,j)} X_{ij}^2}$. Let P_Ω be the orthogonal projection operator onto the span of matrices vanishing outside of Ω so that the $(i,j)-$th component of $P_\Omega(X)$ is equal to X_{ij} when $(i,j) \in \Omega$, and zero otherwise. Let $X = U\Sigma V^T$ be the SVD of X, where $\Sigma = \operatorname{diag}(\sigma_i)$, $1 < i < \min\{m,n\}$ and σ_i the $i-$th largest singular value of X. The shrinkage operator $D_T(X)$ is defined [17] as $D_T(X) = U\Sigma_T V^T$, where $\Sigma_T = \operatorname{diag}(\max\{\sigma_i - T, 0\})$, $1 \le i \le \min\{m,n\}$. We summarize the main result below.

Lemma 1. For a given vector $y \in R^n$ and the thresholding weight vector $w \in R^n_{++}$, the non-uniform singular value operator $S_w[y]$ satisfies:

$$S_w[y] = \operatorname*{argmin}_x \left(\frac{\mu}{2}\|x-y\|_2^2 + \|w \odot x\|_1\right), \tag{3}$$

where $\odot$ is the vector of corresponding matrix multiplication operators, and $\|\cdot\|_2$ and $\|\cdot\|_1$ are the ℓ_2- and ℓ_1-norms, respectively. $\mu > 0$is the penalty factor. As μ approaches 0, any solution to Eq. (3) approaches the solution set of Eq. (2). In other words, the non-uniform singular value operator satisfies:

$$S_w[y] = S_{w0}[z] = \operatorname*{argmin}_x \left(\frac{\mu}{2}\|x-z\|_2^2 + w_0\|x\|_1\right), \tag{4}$$

where $z = \operatorname{sign}(y) \odot (|y| + w_0\mathbf{1} - w)$, $w_0 = \max(\{w_i\})$, $|y| = \operatorname{sign}(y) \odot y$, and vector $\mathbf{1} \in R^n$ with all elements equal to 1.

Proof. For any element of vector y,

$$S_{w_i}[y_i] = \operatorname*{argmin}_{x_i} \left(\frac{\mu}{2}\|x_i - y_i\|_2^2 + w_i\|x_i\|_1\right), \tag{5}$$

To minimize $\mu/2\|x_i-y_i\|_2^2+w_i\|x_i\|_1$, we get the optimization resolution

$$S_{w_0}[z_i]=\{\mathrm{sign}(z_i)\cdot(|z_i|-w_0)_+\}=\{\mathrm{sign}(y_i)\cdot(|y_i|-w_i)_+\}=S_{w_i}[y_i].$$

The singular value shrinkage operator: Consider the SVD of matrix $X\in R^{n_1\times n_2}$ of rank r

$$Y=U\Sigma V^T,\quad \Sigma=\mathrm{diag}(\{\sigma_i\}_{1\le i\le r}), \tag{6}$$

where U and V are left-multiplying and right-multiplying matrices respectively, $m\times r$ and $n\times r$ matrices with orthonormal columns, and the singular values σ_i are positive. For each $w\in R^r_{++}$, we introduce the soft-thresholding operator as follows:

$$D_w[Y]=US_w[\Sigma]V^T, \tag{7}$$

where $S_w[\Sigma]=\mathrm{diag}(\{(\sigma_i-w_i)_+\})$ is a diagonal matrix.

Lemma 2. For each $w\in R^r_{++}$ and $Y\in R^{m\times n}$, the non-uniform singular value soft-thresholding operator obeys

$$D_w[y]=D_{w_0}[z]=\underset{X}{\mathrm{argmin}}\left(\frac{\mu}{2}\|X-Z\|_F^2+w_0\|X\|_*\right), \tag{8}$$

where $Z=U(\Sigma+w_0I-W)V^T$, $Y=U\Sigma V^T$ denotes the problem defined in Eq. (6), $w_0=\max(\{w_i\})$, $I\in R^{r\times r}$ is an unit matrix, and $W=\mathrm{diag}(w)$ is a diagonal matrix.

Proof. The authors in [17] proved that $D_\tau[Y]=\mathrm{argmin}\left(\frac{\mu}{2}\|X-Y\|_F^2+\tau\|X\|_*\right)$, $\tau>0$. Therefore, minimizing $\mu/2\|X-Z\|_F^2+w_0\|X\|_*$, its optimization solution is given as

$$\begin{aligned}D_{w_0}[Z]&=US_{w_0}[\Sigma+w_0I-W]V^T\\&=US_w[\Sigma]V^T=D_w(Y)\end{aligned} \tag{9}$$

3.2. Nuclear norm minimization

In the singular value thresholding (SVT) method, the shrinkage operator plays a critical role in the entire iteration scheme. We utilize a similar scheme for assigning weights to the singular value of a matrix. We attach a smaller weight to a larger singular value of the matrix, and vice versa. We obtain the following result for the nuclear norm minimization problem.

$$\begin{aligned}&\min_{A,E}\sum_{j=1}^{n}w_A\left(\sigma_j^2(A)+\gamma\right)^{p/2}+\lambda\|W_E\odot E\|_1\\&\text{s.t.}\qquad D=A+E\end{aligned} \tag{10}$$

where $w_A=\{w_{A,j}\}$ and W_E are the weights of the singular value of matrix A and the weight of every element of matrix E, respectively. $\{\sigma_j\}$ is the singular value of matrix A. $\gamma>0$ being regularization parameter added to ensure that w_A is well defined. Note that Eq. (10) is differentiable for $p>0$ and convex for $p\ge1$.

The optimization problem in Eq. (10) can be reformulated as an SVT algorithm using a kind of Lagrange multiplier algorithm known as Uzawa's algorithm [31]. An important consequence is that this allows us to extend the SVT algorithm to other problems involving minimization of the nuclear norm under convex constraints.

The following algorithm provides a method for alternately estimating the matrix and updating the weights. The purpose is to enhance the low-rank of a matrix and sparsity of a noisy matrix, and to expand the scope of matrix decomposition successfully.

This algorithm provides a way of constructing the estimation sequence:

Algorithm 1. Alternating direction method (ADM)

Step 1: Assuming iteration number $k=0$, initialize weight $w_A{}^{(0)}=\mathbf{1}\in R^n$, $W_E{}^{(0)}=\mathbf{1}\cdot\mathbf{1}^T\in R^{m\times n}$(assuming $m\ge n$).

Step 2: Using Algorithm 2 to solve the reweighted nuclear norm and ℓ_1-norm minimization problem

$$\begin{aligned}\left(A^{(k+1)},E^{(k+1)}\right)=\underset{A,E}{\mathrm{argmin}}\Bigg(&\sum_{j=1}^{n}w_{A,j}^{(k)}\left(\sigma_j^2(A)+\gamma\right)^{p/2}\\&+\lambda\|W_E^{(k)}\odot E\|_1,\ \text{s.t.}D=A+E\Bigg),\end{aligned} \tag{11}$$

where $w_A{}^{(k)}$ and $W_E{}^{(k)}$ are weight coefficients, we can obtain the estimated value $(A^{(k+1)},E^{(k+1)})$. We defined $p=1$.

Step 3: For $i=1,\cdots,m$ and $j=1,\cdots,n$ update the weight coefficients as follows:

$$\begin{cases}w_{A,j}^{(k+1)}=\frac{1}{\sigma_j^{(k)}(A)+\varepsilon_A}\\ w_{E,ij}^{(k+1)}=\frac{1}{\left|E_{ij}^{(k)}\right|+\varepsilon_E}\end{cases}, \tag{12}$$

where ε_A and ε_E are positive preset constants, and the SVT matrix is

$$\Sigma^{(k)}=\mathrm{diag}\left(\left\{\sigma_i^{(k)}\right\}_{i=1}^{r}\right)\in R^{n\times n}, \tag{13}$$

where $U^{(k)},\Sigma^{(k)},V^{(k)}$ are generated by the following SVD:

$$D-A^{(k+1)}+Y^{(k)}/\mu=U^{(k)}\Sigma^{(k)}\left(V^{(k)}\right)^T \tag{14}$$

with $U^{(k)}\in R^{m\times n},V^{(k)}\in R^{n\times n}$.

Step 4: If the termination conditions are satisfied or k reaches the maximum iteration step k_{max}, output solutions $(\tilde{A},\tilde{E})$; otherwise, increase k and return to step 2.

Table 1
Numerical simulation comparison of PCP, R-ℓ_1, and our proposed method on data type 1.

	Algorithm	k_{max}	$\frac{\|\tilde{A}-A^0\|_F}{\|A^0\|_F}$	Rank($\tilde{A}$)	$\|\tilde{E}\|_0$	No. of iterations	Successful restoration?
$m=n=400$ Rank$(A^0)=80$ $\|E^0\|_0=12800$	PCP		0.0153	217	101,089	35	×
	R-ℓ_1	1	4.47×10^{-6}	88	1282	67	✓
	R-ℓ_1	2	4.43×10^{-6}	88	127,979	102	✓
	Ours	1	5.85×10^{-6}	87	12,697	67	✓
	Ours	2	4.87×10^{-6}	82	12,740	101	✓
$m=n=800$ Rank$(A^0)=160$ $\|E^0\|_0=51{,}200$	PCP		0.0063	434	4,029,479	37	×
	R-ℓ_1	1	7.59×10^{-6}	181	51,743	69	✓
	R-ℓ_1	2	7.57×10^{-6}	179	51,386	102	✓
	Ours	1	7.55×10^{-6}	177	51,098	65	✓
	Ours	2	6.79×10^{-6}	160	51,163	105	✓
$m=n=1200$ Rank$(A^0)=240$ $\|E^0\|_0=115{,}200$	PCP		0.0095	635	913,647	37	×
	R-ℓ_1	1	4.05×10^{-6}	283	115,679	71	✓
	R-ℓ_1	2	3.97×10^{-6}	281	115,241	105	✓
	Ours	1	7.89×10^{-6}	307	115,098	66	✓
	Ours	2	5.18×10^{-6}	248	114,993	101	✓

Table 2
Numerical simulation comparison of PCP, R-ℓ_1, and our proposed method on data type 2.

	algorithm	k_{max}	$\frac{\|\tilde{A}-A^0\|_F}{\|A^0\|_F}$	Rank($\tilde{A}$)	$\|\tilde{E}\|_0$	No. of iterations	Successful restoration?
$m=n=400$ Rank$(A^0)=120$ $\|E^0\|_0=32{,}000$	PCP		0.4957	239	102,199	35	×
	R-ℓ_1	1	0.2931	390	31,800	73	✓
	R-ℓ_1	2	0.2812	379	31,908	107	✓
	Ours	1	4.39×10^{-6}	157	31,902	101	✓
	Ours	2	4.83×10^{-7}	125	31,990	175	✓
$m=n=800$ Rank$(A^0)=240$ $\|E^0\|_0=128{,}000$	PCP		0.4567	448	404,936	35	×
	R-ℓ_1	1	0.2739	763	128,000	77	✓
	R-ℓ_1	2	0.2413	763	128,000	107	✓
	Ours	1	8.39×10^{-7}	287	128,000	93	✓
	Ours	2	6.35×10^{-7}	243	128,000	164	✓
$m=n=1200$ Rank$(A^0)=360$ $\|E^0\|_0=288{,}000$	PCP		0.46798	681	921,438	35	×
	R-ℓ_1	1	0.2276	1097	288,000	73	✓
	R-ℓ_1	2	0.2265	1239	288,000	108	✓
	Ours	1	5.71×10^{-7}	418	288,000	94	✓
	Ours	2	7.89×10^{-7}	358	288,000	159	✓

Table 3
Numerical simulation comparison of PCP, R-ℓ_1, and our proposed method on data type 3.

	algorithm	k_{max}	$\frac{\|\tilde{A}-A^0\|_F}{\|A^0\|_F}$	Rank($\tilde{A}$)	$\|\tilde{E}\|_0$	No. of iterations	Successful restoration?
$m=$i$=400$ Rank$(A^0)=120$ $\|E^0\|_0=32{,}000$	PCP		0.1297	219	98,179	37	×
	R-ℓ_1	1	0.0448	369	32,000	71	✓
	R-ℓ_1	2	0.0437	364	32,000	107	✓
	Ours	1	4.88×10^{-7}	122	32,000	105	✓
	Ours	2	4.79×10^{-7}	122	32,000	167	✓
$m=n=800$ Rank$(A^0)=240$ $\|E^0\|_0=128{,}000$	PCP		0.1298	452	387,096	37	×
	R-ℓ_1	1	0.0437	719	128,000	73	✓
	R-ℓ_1	2	0.0412	713	128,000	109	✓
	Ours	1	6.47×10^{-7}	243	128,000	104	✓
	Ours	2	6.30×10^{-7}	243	128,000	177	✓
$m=n=1200$ Rank$(A^0)=360$ $\|E^0\|_0=288{,}000$	PCP		0.1370	661	881,313	38	×
	R-ℓ_1	1	0.0414	1068	288,000	73	✓
	R-ℓ_1	2	0.0299	1073	288,000	109	✓
	Ours	1	8.11×10^{-7}	358	288,000	105	✓
	Ours	2	7.79×10^{-7}	358	288,000	161	✓

The augmented Lagrange function of the optimization problem in Eq. (10) is given as follows:

$$L_\mu(A,E,Y)=\sum_{j=1}^{n} w_{Aj}\left(\sigma_j^2+\gamma\right)^{p/2}+\lambda\|W_E\odot A\|_1 +\langle Y,A+E-D\rangle+\frac{\mu}{2}\|A+E-D\|_F^2 \tag{15}$$

where $Y\in R^{m\times n}$ is the Lagrange multiplier matrix, and $\langle\cdot,\cdot\rangle$ is the inner product of two matrices. That is, $\langle X_1,X_2\rangle=\mathrm{trace}(X_1^TX_2)$, and $\mu>0$ is the penalty factor for the violation of the linear constraint.

ADM utilizes separability of variables A and E, alternating the estimation in the following steps:

$$\text{fixed } E,\ A^{(k+1)}\leftarrow\min_A L_\mu(A,E^{(k)},Y^{(k)}) \tag{16}$$

$$\text{fixed } A,\ E^{(k+1)}\leftarrow\min_E L_\mu(A^{(k+1)},E,Y^{(k)}) \tag{17}$$

$$\text{update } ADM \text{ by } Y^{(k+1)}\leftarrow Y^{(k)}+\mu(A+E-D) \tag{18}$$

where updating matrix A using the uniform soft-thresholding operator is defined as

$$A\leftarrow D_{\mu^{-1}w_A}[D-E+\mu^{-1}Y], \tag{19}$$

with the thresholding vector $\mu^{-1}w_A$, and updating matrix E using the uniform soft-thresholding operator is defined as

$$E\leftarrow S_{\lambda\mu^{-1}W_E}[D-A+\mu^{-1}Y], \tag{20}$$

with the thresholding matrix $\lambda\mu^{-1}W_E$.

From the above, the ADM for resolving the minimization of the nuclear and ℓ_1-norms is given below.

Algorithm 2. ADM for nuclear-norm and ℓ_1-norm minimization problem (Eq. 10)

INPUT: initial values $(A_0,E_0)\in R^{m\times n}\times R^{m\times n}$, $Y_0\in R^{m\times n}$, $\mu_0>0$, $\rho>1$, $t=0$

WHILE not converged DO

$A_{t+1}=D_{\mu_t^{-1}w_A}[D-E_t+\mu_t^{-1}Y_t]$

$E_{t+1}=S_{\lambda\mu_t^{-1}w_E}[D-A_{t+1}+\mu_t^{-1}Y_t]$

$Y_{t+1}=Y_t+\mu_t(D-A_{t+1}-E_{t+1})$

$\mu_{t+1}=\rho\mu_t$

$t\leftarrow t+1$

END WHILE

OUTPUT: solution $(\tilde{A},\tilde{E})$ to Eq. (10)

4. Experiments and applications

In this section, we report on testing the proposed model on different types of data to enhance the performance of low-rank data and sparse error matrices. At the same time, we also apply the model to a single image recovery and hyperspectral image restoration. In the experiments, weight coefficient λ is just like the parameters of a principal component tracing problem, and is set to $1/\sqrt{\max(m,n)}$. In Algorithm 1, we defined $\varepsilon_A = 0.01$, and $\varepsilon_E = 0.01$. For Algorithm 2, the original value $(A_0, E_0) = (0, 0)$, $Y_0 = D/\max(\|D\|, \lambda^{-1}\|D\|_\infty)$, where $\|D\|$ is the spectral norm, and $\|D\|_\infty$ is the ℓ_∞-norm of matrix D, which is the largest absolute value of the matrix elements. Two parameters, $\mu_0 = 1.25/\|D\|$ and $\rho = 1.5$ are defined if $\|D - A_t - E_t\|_F/\|D\|_F < 10^{-7}$ is satisfied, after which iteration stops.

We carried out a detailed experiment, comparing our proposed algorithm with the principal component pursuit (PCP) convex optimization method of the original PCP and reweighting the ℓ_1-norm only minimization method (referred to as the Reweighted ℓ_1). The optimization problem is given by the following formula, which is also the research experiment in [17].

$$\min_{A,E} \|A\|_* + \lambda \|W_E \odot E\|_1 \quad \text{s.t.} \quad D = A + E \tag{21}$$

Similarly, only if the recovery result of solution $\tilde{A}$ satisfies the following equation, is the matrix considered a successful recovery.

$$\frac{\|\tilde{A} - A_0\|_F}{\|A^0\|_F} \le \varepsilon,$$

where $\varepsilon = 10^{-7}$ is the tolerance.

The ADM is adopted for the PCP, Reweighted ℓ_1 (Eq. (21)), and proposed problem in Eq. (10).

4.1. Numerical simulations

We carried out numerical simulations on different types of observation data. We also studied the capacity of the proposed algorithms to recover data matrices with different rank sizes and error matrices with different coefficients.

The real value of the random matrix in the experiment on $\left(A^0, E^0\right) \in R^{m\times m} \times R^{m\times m}$ satisfies $\text{rank}\left(A^0\right) = \rho_r n$, $\|E^0\|_0 = \rho_s n^2$, where $0 < \rho_r < 1$, $0 < \rho_s < 1$. If output $\tilde{A}$ of Algorithm 1 satisfies the formula below, the matrix is considered to be a successful recovery.

$$\frac{\|\tilde{A} - A_0\|_F}{\|A^0\|_F} \le 10^{-4} \tag{22}$$

Our experiment was independently repeated 15 times using different types of observation data. We also tested three different types of data matrices with random observations. These have frequently been used in existing literature and all were generated by $D = A^0 + E^0$, where A^0 is a low rank data matrix satisfying$A^0 = R_1 R_2^T$. Each element of $R_1, R_2 \in R^{n\times r}$is a random number that satisfies the independent Gaussian distribution, $N(0, 1/n)$. Sparse error matrices were generated in the following three ways:

- Data type 1: Nonzero elements of the matrix have a random uniform distribution and a uniform independent value within the scope [500 500];

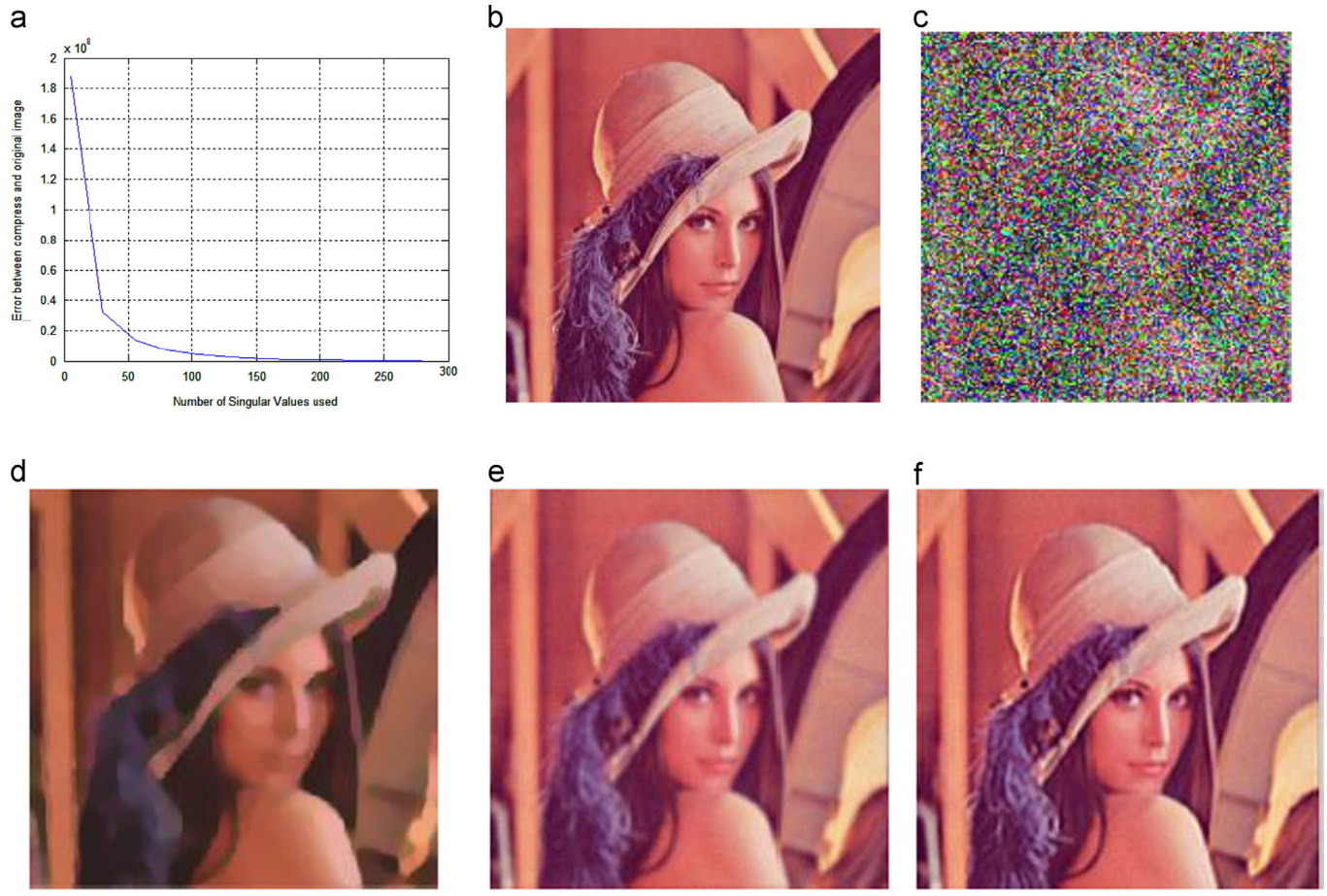


Fig. 1. Single image restored (example 1). (a) Singular value distribution, (b) original image, (c) input image, (d) PCP, (e) Reweighted ℓ_1 and (f) Ours.

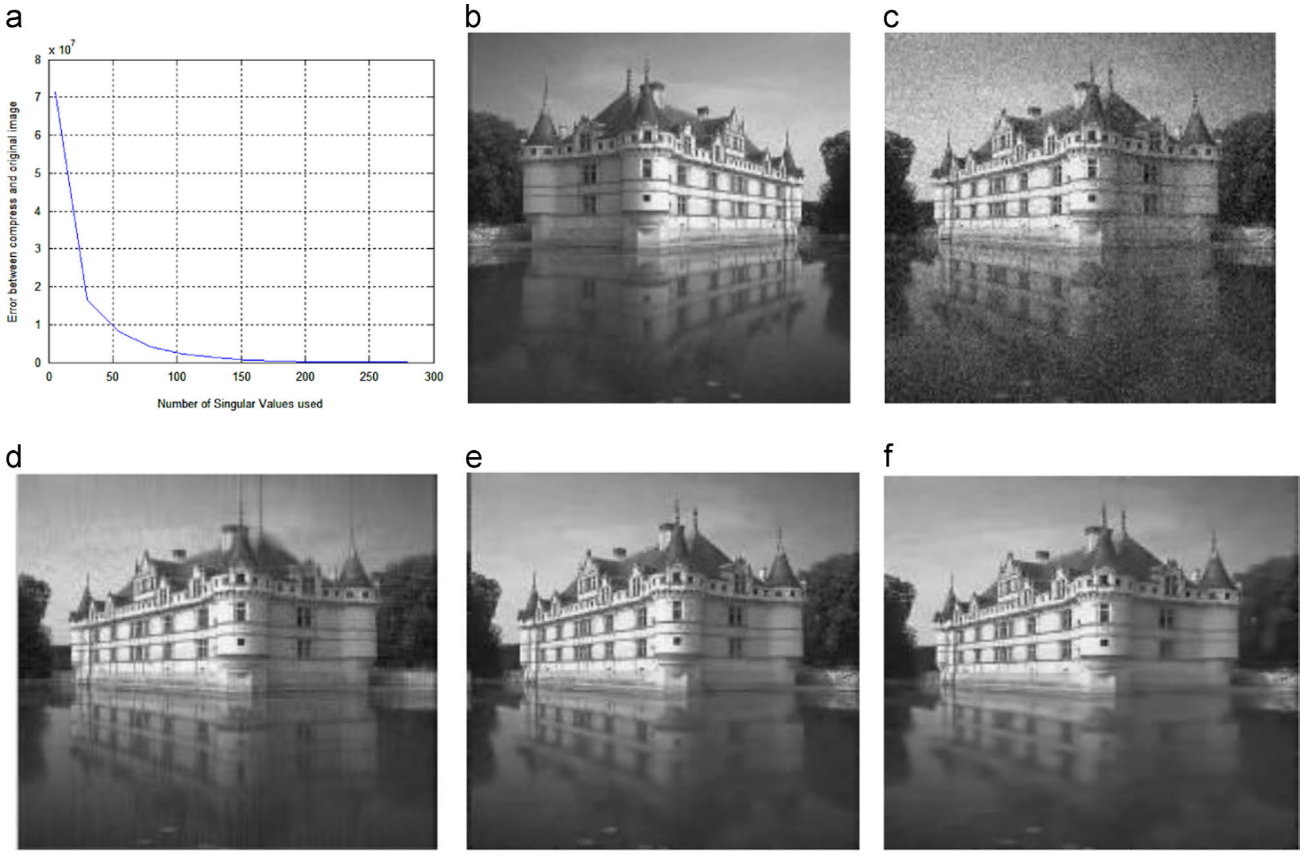


Fig. 2. Single image restored (example 2). (a) Singular value distribution, (b) original image (c,) input image, (d) PCP, (e) Reweighted ℓ_1 and (f) Ours.

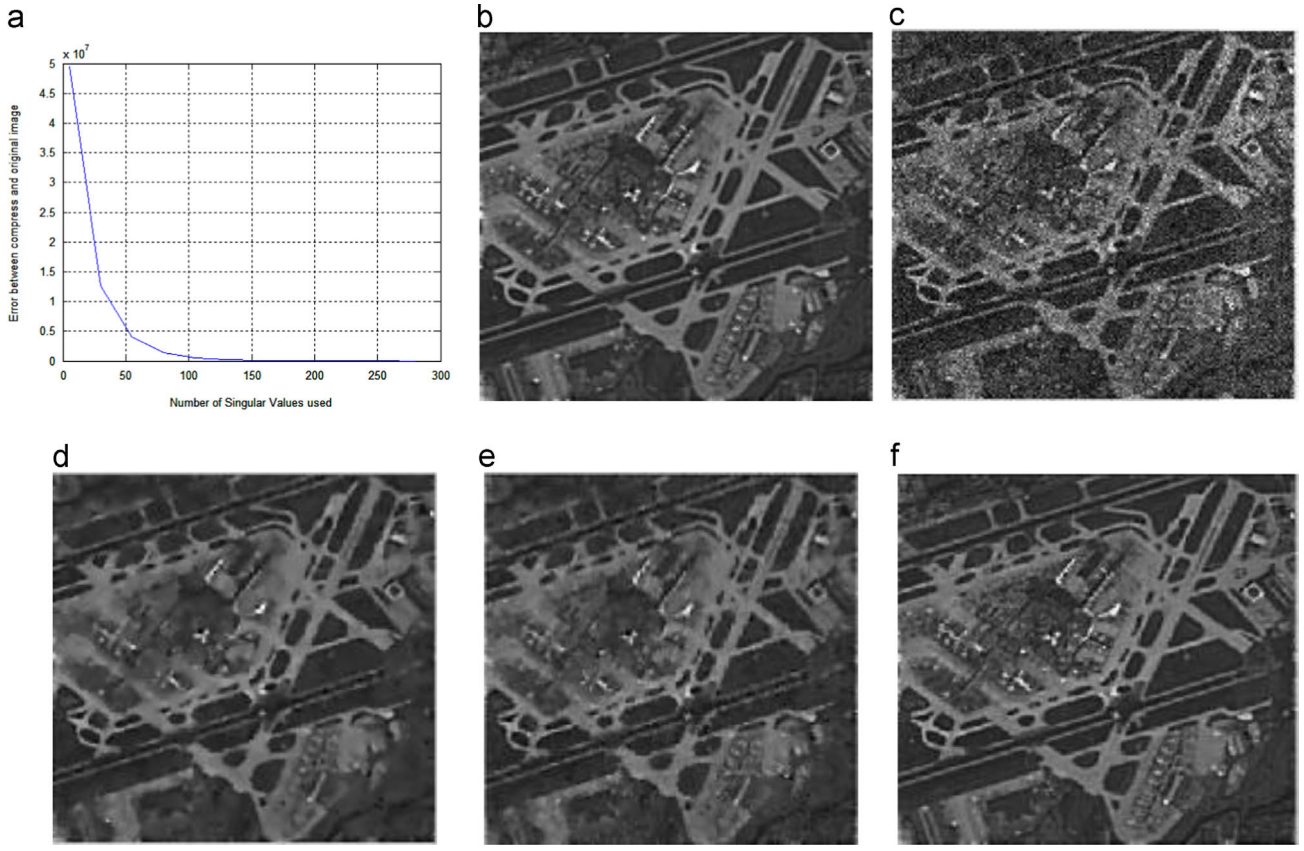


Fig. 3. Single image restored (example 3). (a) Singular value distribution, (b) original image, (c) input image, (d) PCP, (e) Reweighted ℓ_1 and (f) Ours.

- Data type 2: Nonzero elements of the matrix have a random uniform distribution and a random value, either $+1$ or -1 with $\rho_s/2$ probability;
- Data type 3: Nonzero elements of the matrix have a random uniform distribution, where Ω represents nonzero elements with values $P_\Omega(E^0) = P_\Omega(\mathrm{sign}(A^0))$, while the rest of the elements have values of 0, where $P_\Omega(\cdot)$ denotes obtaining values for the matrix in its location [24].

Some digital simulation results are shown in Tables 1–3. In Table 1, $\rho_r = 0.2$ and $\rho_s = 0.08$, while in Tables 2 and 3, $\rho_r = 0.3$ and $\rho_s = 0.2$. These tables list the recovery results including the matrix dimensions, rank, error ratio of the error matrix, result of matrix recovery, and total number of iterations (i.e., total number of SVDs in the operation process). According to these tables, even though PCP and the Reweighted ℓ_1 algorithm resulted in failure, our proposed method was able to recover the original data matrix successfully. The performance of LR was greatly improved in that the method only required one or two rounds of iterations. With regard to the iteration steps in a round of the reweighting iterative process (step 2 in Algorithm 1), we found that the Reweighted ℓ_1 and PCP algorithms used almost the same number of iteration steps (about 35 iterations) in each round on the three different types of data. We also found that the number of iterative steps in our proposed algorithm was not the same for the different types of data. The proposed algorithm required approximately the same number of iterative steps as the PCP and Reweighted ℓ_1 algorithms for data type 1, while for data type 2, the proposed algorithm required about twice as many iterative steps as the PCP and Reweighted ℓ_1 algorithms.

4.2. Single image restoration

In this section we apply the proposed model and algorithm to single image restoration. Most natural images often show approximate low rank properties. Using an image as a matrix and performing SVD, we find that the size of the singular value presents a fast attenuation trend as shown in Fig. 1. This corresponds to the singular value distribution of Fig. 1(b). Having input the damaged image "Lena" shown in Fig. 1(c) with 512×512 pixels and 40% of the pixels damaged by large random errors, we wish to use the (approximate) low rank properties of the original image to recover the damaged image. Since we do not know which pixels have been damaged in this image restoration problem, we model the image restoration problem as the restoration of low rank images without error from observation errors of images with large sparse errors. (The original image is an (approximate) low rank image without errors.) We used PCP, Reweighted ℓ_1 and our algorithm for image restoration and set the largest iteration step $k_{max} = 1$ for both the Reweighted ℓ_1 and our algorithm in the experiments. The results of the image restoration using the three algorithms are shown in Fig. 1(d)–(f). The figure shows that our

Table 4
PSNR of input image and restored images.

Restoration example	PSNR (dB)			
	Image (c)	Image (d)	Image (e)	Image (f)
Fig. 1	10.13	28.49	25.86	34.37
Fig. 2	11.55	28.67	27.96	33.01
Fig. 3	10.89	29.32	27.59	34.15
Fig. 4	12.91	24.96	23.82	26.51

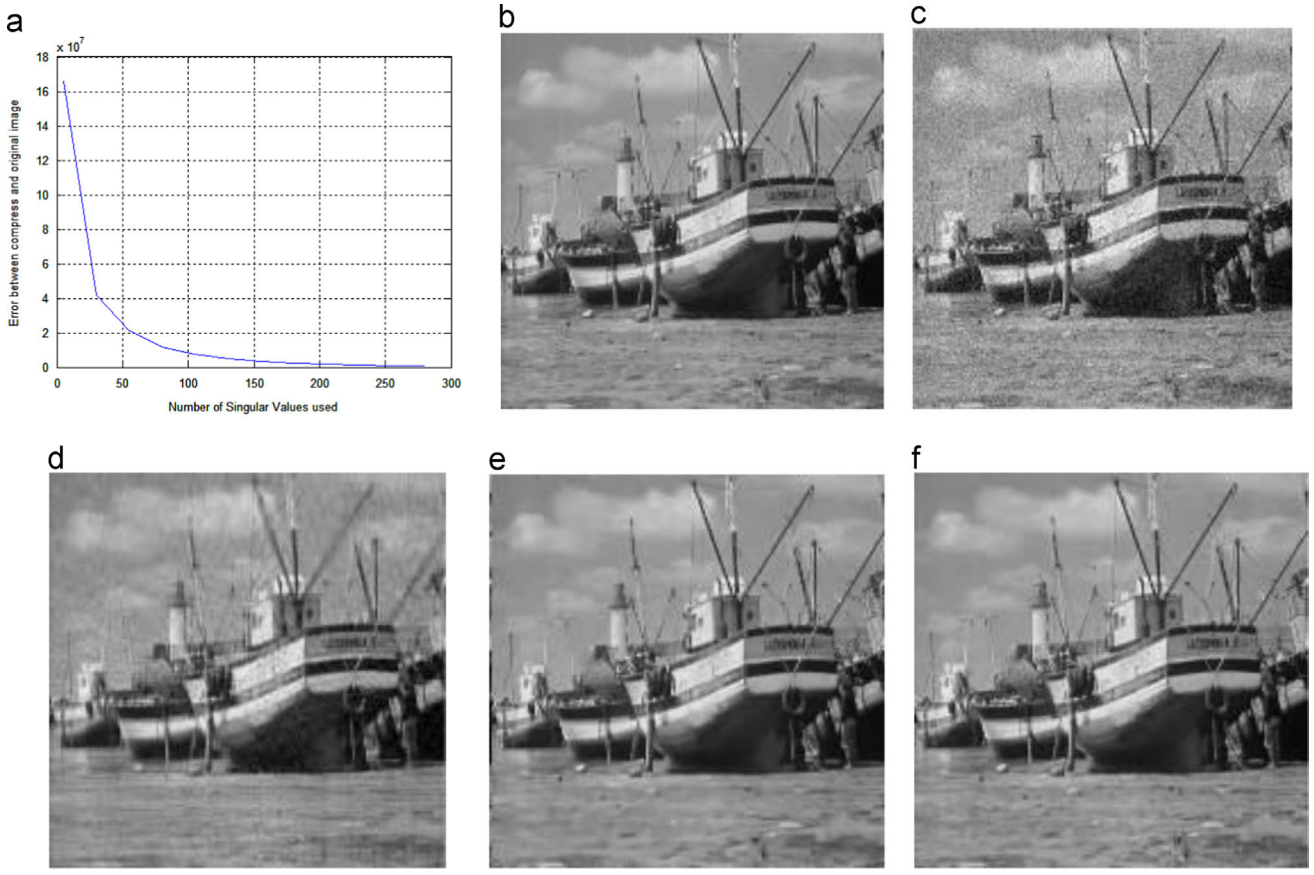


Fig. 4. Single image restored (example 4). (a) Singular value distribution, (b) original image, (c) input image, (d) PCP, (e) Reweighted ℓ_1 and (f) Ours.

algorithm achieves the best image recovery results, while also dealing with the low rank data items and sparse error terms, thus having the best effect on image restoration. Additional image restoration results are shown in Figs. 2–4. These images contain a variety of different scenes, such as portraits, houses, bridges, ships, and so on. It can be seen that the proposed algorithm is significantly better than the other two algorithms (PCP and Reweighted ℓ_1) for all different types of image restoration.

To measure the quality of image restoration objectively, we calculated the peak signal-to-noise ratio (PSNR) of the input and restored images:

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{23}$$

where mean squared error (MSE) is defined as

$$MSE = \frac{1}{mn}\|\tilde{A} - A^0\|_F^2. \tag{24}$$

The higher the PSNR, the greater is the objective measurement of image restoration and the better is the image recovery result. Various PSNRs are listed in Table 4, where columns 2–5 give the PSNR relative to the original image of the damaged input image, and the restored images using the PCP, Reweighted ℓ_1, and proposed methods. The PSNR of the restored image using our method to the original image is 2–6 dB greater than that using the PCP method.

4.3. Hyperspectral image restoration

Hyperspectral imaging can depict scene information in different spectral bands. If the image is arranged as a column vector on each spectral band, then by arranging these column vectors as a matrix, the scene will have n spectral bands and each spectral band image will have $m$pixels of the hyperspectral image, thus, constituting matrix$H \in R^{m\times n}$. Hyperspectral images tend to be low rank (approximations), because matrix H of the hyperspectral image can often be expressed as a nonnegative linear combination of the basis matrix $B \in R_{++}^{m\times k}$; that is, $H = BC$, where $C \in R_{++}^{k\times n}$, $k < \min(m,n)$. Fig. 5(a) shows the singular value distribution of a scene in a hyperspectral image, while the scene image on a band of the hyperspectral image is shown in Fig. 5(b). The scenes in the hyperspectral image have a total of 31 different spectral bands, where each band has an image with 256×256 pixels and matrix $H \in R^{65536\times 31}$. From the hyperspectral singular value distribution, we find that only the first few singular values are very large, with the remainder falling off rapidly. The low rank characteristics of the hyperspectral images are also clearly visible. As shown in Fig. 5 (c), 40% of the image pixels were randomly destroyed in the damaged input image. In fact, it is difficult to even identify the scene corresponding to the damaged image. We therefore use low rank prior information of the hyperspectral images in the recovery process. Fig. 5(d), (e), and (f) shows the restored images using the PCP, Reweighted ℓ_1, and proposed methods, respectively. We set $k_{\max} = 1$, but the images obtained by the PCP method still contain a number of errors according to the experimental results. Although the Reweighted ℓ_1 algorithm reduces these errors, it also introduces additional errors. In summary, our proposed algorithm achieves the best restoration results.

5. Conclusion

In this paper, we proposed a matrix model for low-rank sparse decomposition under linear constraints, which was used to improve the successful range of low-rank sparse matrix decomposition. From the numerical simulations and single and

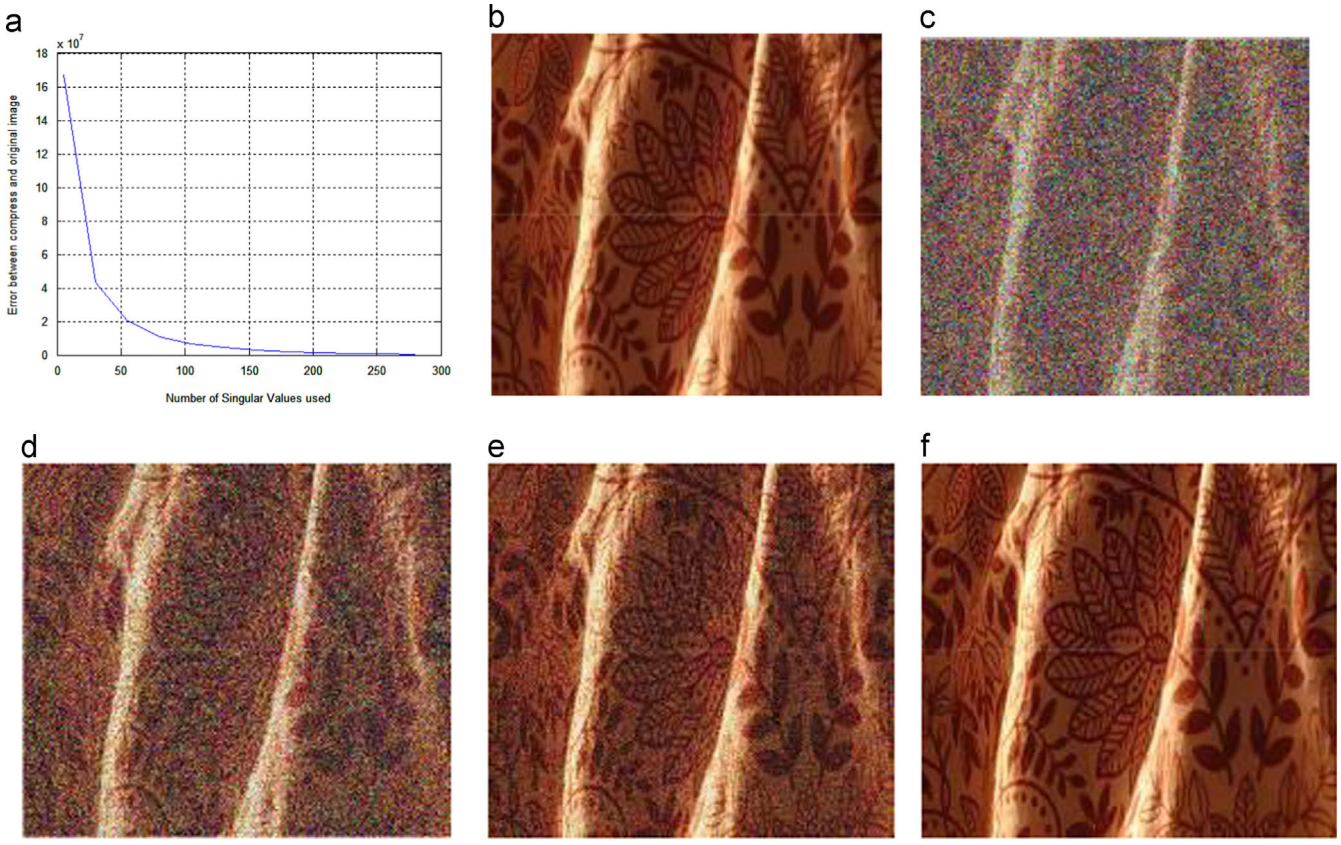


Fig. 5. Hyperspectral image restoration. (a) Singular value distribution, (b) original image, (c) input image, (d) PCP, (e) Reweighted ℓ_1 and (f) Ours.

hyperspectral image restoration experiments on a large number of different types of data, it is clear that the proposed algorithm achieves improved results compared with existing algorithms, that is, PCP and the Reweighted ℓ_1 algorithm.

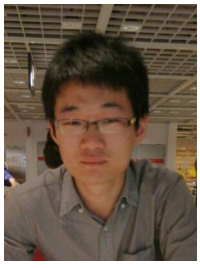
Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61370138, 61271435, 61202245, 61103130, U1301251); National Program on Key Basic Research Projects (973 programs) (Grant nos. 2010CB731804-1, 2011CB706901-4); The Project of Construction of Innovative Teams and Teacher Career Development for Universities and Colleges Under Beijing Municipality (Nos. IDHT20130513, CIT&TCD20130513), Beijing Municipal Natural Science Fundation (Nos. 4141003, 4152017).

References

[1] T. Chan, S. Esedoglu, F. Park, A. Yip, Recent Developments in Total Variation Image Restoration, Mathematical Models of Computer Vision, Springer, New York (2005) 1–18.
[2] R. Molina, J. Mateos, A.K. Katsaggelos, Blind deconvolution using a variational approach to parameter, image, and blur estimation, IEEE Trans. Image Process. 15 (12) (2006) 3715–3727.
[3] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Trans. Image Process. 18 (11) (2009) 419–2434.
[4] G. Chantas, N.P. Galatsanos, R. Molina, A.K. Katsaggelos, Variational Bayesian image restoration with a product of spatially weighted total variation image priors, IEEE Trans. Image Process. 9 (2) (2010) 351–362.
[5] A. Marquina, S.J. Osher, Image super-resolution by TV-regularization and Bregman iteration, J. Sci. Comput. 37 (2008) 367–382.
[6] L. Kaur, S. Gupla, R.C. Chauhan, Image denoising using wavelet thresholding, ICVGIP 2 (2002) 16–18.
[7] E. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted L_1 minimization, J. Fourier Anal. Appl. 14 (2008) 877–905.
[8] J. Yang, J. Wright, Y. Ma, T. Huang, Image super-resolution as sparse representation of raw image patches, IEEE Comput. Vis. Pattern Recognit. (CVPR), 2008.
[9] R. Rubinstein, M. Zibulevsky, M. Elad, Double sparsity: learning sparse dictionaries for sparse signal approximation, IEEE Trans. Signal Process. 58 (3) (2010) 1553–1564.
[10] R. Rubinstein, A.M. Bruckstein, M. Elad, Dictionaries for sparse representation modeling, Proc. IEEE 98 (6) (2010) 1045–1057, Special Issue on Applications of Compressive Sensing & Sparse Representation.
[11] Weisheng Dong, L.e.i. Zhang, Guangming Shi, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Trans. Image Process. 20 (7) (2011) 1838–1857.
[12] M. Protter, I. Yavneh, M. Elad, Closed-form MMSE estimation for signal denoising under sparse representation modeling over a unitary dictionary, IEEE Trans. Signal Process. 58 (7) (2010) 3471–3484.
[13] D. Gross, Recovering low-rank matrices from few coefficients in any basis, IEEE Trans. Inf. Theory 57 (3) (2011) 1548–1566.
[14] M. Wang, R. Hong, X. Yuan, S. Yan, Tat-Seng Chua, Movie2Comics: towards a lively video content presentation, IEEE Trans. Multimed. 14 (3) (2012) 858–870.
[15] M. Wang, R. Hong, G. Li, Z.-J. Zha, S. Yan, Tat-Seng Chua, Event driven web video summarization by tag localization and key-shot identification, IEEE Trans. Multimed. 14 (4) (2012) 975–985.
[16] Zhouchen Lin, Arvind Ganesh, John Wright, Leqin Wu, Minming Chen, Y.i. Ma, Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix, in: Proceedings of the International Workshop on Computational Advances in Multi-Sensor Adapt. Processing, Aruba, Dutch Antilles, 2009.
[17] J. Wright, A. Ganesh, S. Rao, Y. Ma, Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization, J. ACM (2009).
[18] E. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, J. ACM, 58, 20111–37.
[19] M.A.T. Figueiredo, R.D. Nowak, S.J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. 1 (4) (2007) 586–597.
[20] A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, Y. Ma, Fast algorithms for recovering a corrupted low-rank matrix, in: Proceeding of International Workshop on Computational Advances in Kulti-Sensor Adaptive Processing, 2009.
[21] L. Nie, M. Wang, Y. Gao, Z.-J. Zha, Tat-Seng Chua, Beyond text QA: multimedia answer generation by harvesting web information, IEEE Trans. Multimed. 15 (2) (2013) 426–441.
[22] Z.-J. Zha, H. Zhang, M. Wang, H. Luan, T.-S. Chua, Detecting group activities with multi-camera context, IEEE Trans. Circuits Syst. Video Technol. 23 (5) (2013) 856–869.
[23] Z.-J. Zha, M. Wang, Y.-T. Zheng, Y. Yang, R. Hong, T.-S. Chua, Interactive video indexing with statistical active learning, IEEE Trans. Multimed. 14 (1) (2012) 17–27.
[24] Y. Peng, J. Suo, Q. Dai, W. Xu, S. Lu, Robust image restoration via reweighed low-rank matrix recovery, Multimed. Model. (2014) 315–326.
[25] Y. Peng, A. Ganesh, J. Wright, W. Xu, RASL: row-rank decomposition for linearly correlated images, IEEE. Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2233–2246.
[26] K. Mohan, M. Fazel, Iterative reweighted algorithms for matrix rank minimization, J Mach. Learn. Res. 13 (2012) 3441–3473.
[27] N.B. Bras, J. Bioucas-Dias, R.C. Matins, A.C. Serra, An alternating direction algorithm for total variation reconstruction of distributed parameters, IEEE Trans. Image Process. 21 (6) (2012) 3004–3016.
[28] Z.-J. Zha, Y. Yang, J. Tang, M. Wang, T.-S. Chua, Robust multi-view feature learning for RGB-D image understanding, ACM Trans. Intell. Syst. Technol. 6 (2) (2014), article no. 15. http://dx.doi.org/10.1145/2735521.
[29] Z.-J. Zha, L. Yang, T. Mei, M. Wang, Z. Wang, T.-S. Chua, Visual query suggestion: towards capturing user intent in internet image search, ACM Trans. Multimed. Comput., Commun. Appl. 6 (3) (2010), Article No. 13.
[30] M. Fornasier, H. Rauhut, R. Ward, Low-rank matrix recovery via iteratively reweighted least squares minimization, SIAM J. Optim. 21 (4) (2011) 1614–1640.
[31] J.L. Li, T.Z. Huang, L. Li, A modified nonlinear Uzawa algorithm for solving non-symmetric saddle point problem, Acta Math. Sci. 31 (1) (2011) 250–262.

Ning He was born in Panjin, Liaoning Province on July 29, 1970. She graduated from Department of Mathematics at Ningxia University in July 1993. She received M.S. degree and Ph.D. degree in applied mathematics from Northwest University and Capital Normal University in July 2003 and July 2009, respectively. Currently she is a vice professor of Beijing Union University. Her research interests include digital image processing, computer graphics.



Jin-Bao Wang received the BS degree from Electronic Information Science and Technology, Physics Department, Hebei University, BaoDing, in 2013. He is currently working toward the MS degree in the Beijing Key Laboratory of Information Service Engineering, College of Information Technology, Beijing Union University. He is interested in digital image processing.

Lulu Zhang, a master student in software engineering from Beijing Union University, Beijing Key Laboratory of Information Service Engineering. Presently her fields of interest are digital image processing and computer vision.

Ke Lu was born in Ningxia on March 13, 1971. He received master degree and Ph.D. degree from the Department of Mathematics and Department of Computer Science at Northwest University in July 1998 and July 2003, respectively. He worked as a postdoctoral fellow in the Institute of Automation Chinese Academy of Sciences from July 2003 to April 2005. Currently he is a professor of the University of the Chinese Academy of Sciences. His current research areas focus on computer vision, 3D image reconstruction and computer graphics.