Patch-Based Image Inpainting via Two-Stage Low Rank Approximation

Qiang Guo, Shanshan Gao, Xiaofeng Zhang, Yilong Yin, and Caiming Zhang

Abstract—To recover the corrupted pixels, traditional inpainting methods based on low-rank priors generally need to solve a convex optimization problem by an iterative singular value shrinkage algorithm. In this paper, we propose a simple method for image inpainting using low rank approximation, which avoids the time-consuming iterative shrinkage. Specifically, if similar patches of a corrupted image are identified and reshaped as vectors, then a patch matrix can be constructed by collecting these similar patch-vectors. Due to its columns being highly linearly correlated, this patch matrix is low-rank. Instead of using an iterative singular value shrinkage scheme, the proposed method utilizes low rank approximation with truncated singular values to derive a closed-form estimate for each patch matrix. Depending upon an observation that there exists a distinct gap in the singular spectrum of patch matrix, the rank of each patch matrix is empirically determined by a heuristic procedure. Inspired by the inpainting algorithms with component decomposition, a two-stage low rank approximation (TSLRA) scheme is designed to recover image structures and refine texture details of corrupted images. Experimental results on various inpainting tasks demonstrate that the proposed method is comparable and even superior to some state-of-the-art inpainting algorithms.

Index Terms—Component decomposition, image inpainting, low rank approximation, self-similarity, singular value decomposition

1 Introduction

Image inpainting aims to reconstruct the missing or damaged parts of images from observed incomplete data as accurate as possible, which is a very active topic in the field of image processing. Research on this topic is boosted by numerous applications such as removing undesirable objects from photographs or films, suppressing texts and scratches in ancient drawings, restoring the pixels missed during image transmission [1]. Since image inpainting algorithms usually take advantage of the self-similarity inherent in images to synthesize the missing data, the processing strategies for inpainting are also useful for noise reduction, superresolution, demosaicing, etc.

Image inpainting is a typical ill-posed inverse problem, which means that the solution is not unique and some additional prior information should be introduced to yield a unique solution. As a promising candidate for image inpainting, low-rank priors have attracted an increasing interest in recent years. A basic assumption of low-rank priors is that the image can be represented by a low-rank matrix. In fact, for natural images, it is highly probable that they have low rank or approximately low rank structure. Fig. 1b illustrates the low-rank property of the image *Barbara* and its variants shown in Fig. 1a, in which the structural and textural components of this image are generated by the region covariance based decomposition algorithm [2]. From Fig. 1b, it can be seen that the corresponding matrices of four images are approximately low-rank. Therefore, as validated in some previous works [3], [4], [5], low-rank priors can be exploited to deal with the image inpainting problem.

For an observed incomplete data matrix $\mathbf{Y} \in \mathbb{R}^{m\times n}$, existing low-rank methods generally recover the missing data by solving a nuclear norm minimization subject to data constraints [6], which is formulated as

$$\widehat{\mathbf{X}} = \arg \min_{\mathbf{X}\in\mathbb{R}^{m\times n}} \|\mathbf{X}\|_* \quad s.t. \quad \mathcal{P}_\Omega(\mathbf{X}) = \mathcal{P}_\Omega(\mathbf{Y}), \tag{1}$$

where $\|\mathbf{X}\|_*$ denotes the nuclear norm and is defined by the sum of singular values of $\mathbf{X}$, and $\mathcal{P}_\Omega$ is a sampling operator in the observed region Ω, which is defined as follows: $[\mathcal{P}_\Omega(\mathbf{X})]_{ij} = \mathbf{X}_{ij}$ if $(i,j)\in\Omega$ and 0 otherwise. This convex optimization problem is tractable by the following singular value thresholding (SVT) algorithm [7] with $\mathbf{Z}^0 = \mathbf{0} \in \mathbb{R}^{m\times n}$,

$$\begin{cases} \mathbf{X}^k = \text{shrink}(\mathbf{Z}^{k-1}, \lambda), \\ \mathbf{Z}^k = \mathbf{Z}^{k-1} + \delta_k \mathcal{P}_\Omega(\mathbf{Y} - \mathbf{X}^k) \end{cases}, \tag{2}$$

where $\text{shrink}(\mathbf{Z}, \lambda)$ is a function that shrinks the singular values of $\mathbf{Z}$ towards zero by a threshold λ, δ_k denotes the scalar step, and k is the iteration number. While this iterative algorithm has been widely used in the field of image restoration, it is computationally expensive because it needs

- Q. Guo and S. Gao are with the School of Computer Science and Technology, Shandong Provincial Key Laboratory of Digital Media Technology, Shandong University of Finance and Economics, Jinan 250014, China. E-mail: guoqiang@sdufe.edu.cn, gsszxy@aliyun.com.
- X. Zhang is with the School of Information and Electrical Engineering, Ludong University, Yantai 264025, China. E-mail: iamzxf@126.com.
- Y. Yin is with the School of Computer Science and Technology, Shandong University, Jinan 250100, China. E-mail: ylyin@sdu.edu.cn.
- C. Zhang is with the School of Computer Science and Technology, Shandong University, Jinan 250100, China and the Shandong Provincial Key Laboratory of Digital Media Technology, Jinan 250014, China. E-mail: czhang@sdu.edu.cn.

Manuscript received 25 Aug. 2016; revised 3 May 2017; accepted 5 May 2017. Date of publication 9 May 2017; date of current version 27 Apr. 2018. (Corresponding author: Qiang Guo.) Recommended for acceptance by S.-M. Hu. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TVCG.2017.2702738

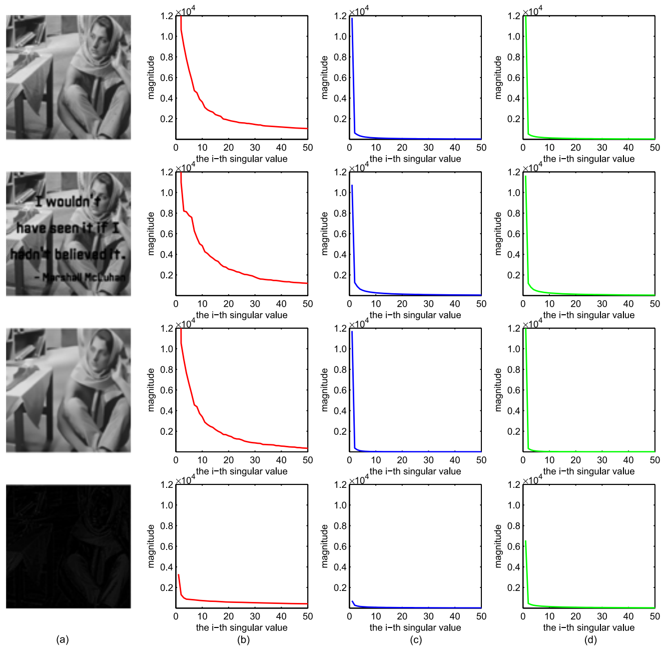

Fig. 1. Illustration of low-rank property of images. (a) From top to bottom: original image *Barbara*, destroyed version by a text mask, structural component, and textural component; (b) empirical distributions of singular values of the image *Barbara* and its variants; (c) empirical distributions of singular values of corresponding patch matrices; (d) averaged empirical distributions of singular values of patch matrices from other test images and their variants.

to compute the singular value decomposition (SVD) of $\mathbf{Z}$ at each iteration.

In essence, the low-rank property of images arises from the nonlocal self-similarity. For each patch, we can find its similar patches and construct a patch matrix by reshaping each patch as a column vector. Due to the high correlation between similar patches, the patch matrix has a low rank structure. Fig. 1c displays the empirical distributions of singular values of patch matrices, in which each point denotes the average ith singular value over all patch matrices. It can be observed that all singular values except the first several ones progressively decrease approaching zero, which means that most energy of patch matrices is compacted into several large singular values. Fig. 1d further demonstrates the energy compaction property of singular values of patch matrices, in which the averaged empirical distributions of singular values of patch matrices from other test images[1] and their variants are shown. Thus we can obtain a good approximation of a patch matrix by only reconstructing the large singular values. Additionally, we also notice that the first several singular values of the structural component have larger magnitude than ones of the textural component. It implies that a universal threshold used in the SVT is not the best choice for recovering different components. All these observations inspire us to develop a simple inpainting algorithm that avoids the iterative calculation of SVT, and is capable of recovering the missing structure and texture.

In this paper, we propose an inpainting method by explicitly using the nonlocal self-similarity prior and the low-rank prior. Thus it inherits the advantages of exemplar-based and low-rank algorithms. For each patch with missing pixels, the proposed method first searches its similar patches and uses them to construct a patch matrix. Due to the fact that the similarity between patches leads the patch matrix to be low-rank, the missing data of this matrix can be recovered by solving an optimization problem with the low-rank constraint. Instead of using the iterative SVT, our method utilizes low rank approximation (LRA) with truncated singular values to derive a closed-form solution for each patch matrix. As a key parameter of LRA, the rank of each patch matrix is determined by calculating the gap between singular values. By successively applying LRA to the structural and textural components of an observed image, we design a two-stage low rank approximation (TSLRA) scheme to recover image structures and refine texture details in the missing region. Extensive experimental results demonstrate the effectiveness of the proposed method.

1. We use all grayscale images in *BM3D_images*, which is available at http://www.cs.tut.fi/%7Efoi/GCF-BM3D/

2 Related Work

Various inpainting methods have been proposed in the literature. They assume that the missing and known pixels share the same geometrical or statistical properties, which are modeled as different priors. Generally, the priors for image inpainting can be roughly divided into four categories: local smooth priors, nonlocal self-similarity priors, sparsity priors and low-rank priors. This section briefly review related work. For more details, we refer the reader to an excellent and thorough review paper [8] on this issue.

2.1 Diffusion-Based Inpainting

The most fundamental inpainting methods based on local smooth priors are diffusion-based methods, which fill the missing region by diffusing local structure from the known region. This diffusion process can be formulated as a partial differential equation (PDE). The pioneered work on diffusion-based inpainting was first introduced by Bertalmio et al. in [9], in which the target region was filled in by propagating the local geometrical information along its isophote direction. In order to diffuse the information in an anisotropic manner, a tensor-driven PDE model [10] was proposed by taking the curvature of specific integral curves into account.

By formulating image inpainting as a variational problem, Shen and Chan [11] proposed a variational inpainting model based on total variation (TV). They further introduced the elastica based variational models into the task of image inpainting [12]. To propagate the gray-levels and the gradient orientations, a set of coupled second order PDEs were also developed [13]. These variational or PDE-based methods are well-suited for handling thin missing regions and can provide interesting results in terms of the global geometry of images. However, they tend to oversmooth the image and produce some blurring artifacts, especially when missing parts of the image are large and textured. The main reason for this is that the local smooth assumption makes the performance of diffusion-based inpainting algorithms depend heavily upon local redundancy, which leads to failing in reconstructing the missing region that is not characterized by local information.

2.2 Exemplar-Based Inpainting

To overcome the disadvantages of local smooth priors, nonlocal self-similarity priors, first stemmed from the field of texture synthesis, are used to fill the missing region. They assume that a target region can be synthesized by estimating the value of each pixel according to a patch centered on it or multiple similar patches [14]. Thus such kind of methods is known as exemplar-based synthesis and widely used in the field of image editing [15], [16], [17], [18], [19]. Criminisi et al. [20] extended the exemplar-based algorithm by considering the patch priority. It is capable of propagating both linear structure and 2D texture into the missing region.

In fact, the structure and texture information can also be propagated by solving a global optimization problem with structure and consistency constraints [21]. To obtain more globally consistent fills, the method proposed by Wexler et al. [22] optimizes a global objective function based on the coherence measure. It can be significantly accelerated by a fast randomized patch search algorithm called PatchMatch [23] or an inverted index called PatchTable [24]. The combination of Wexler et al.'s method and PatchMatch is implemented as the *Content-Aware Fill* in Adobe Photoshop CS5. To further improve this method, different patch constraints, such as translational regularity [25], statistics of patch offsets [26], geometric transform [27], and symmetric transform [28], are included in the objective functions. Besides, Herling and Broll [29] developed a fast algorithm called PixMix for real-time inpainting. From a variational perspective, the exemplar-based inpainting can also be formulated as a variational problem. Many types of variational formulations [30], [31] have been proposed to reconstruct the missing information. Although the exemplar-based methods provide interesting results for inpainting large missing regions, they are sensitive to noises.

2.3 Sparsity-Based Inpainting

With the advent of sparse representation, sparsity priors have also been introduced to image inpainting. The basic assumption of sparsity priors is that the known and missing patches can be sparsely represented by a fixed basis or a learned dictionary. Under this assumption, Elad et al. [32] proposed a sparsity-based inpainting algorithm by minimizing a convex optimization problem with sparsity and TV constraints. To enhance the inpainting quality, He and Wang [33] proposed a weighted sparse model by incorporating the structure information of wavelet frame coefficients into the sparsity constraint. Unlike this method, which uses the fixed wavelet bases for sparse representation, Zhou et al. [34] learned a dictionary by a beta-Bernoulli process factor analysis (BPFA) model and used it to recover the incomplete image by imposing a sparsity constraint on dictionary coefficients. In [35], an iterative thresholding-based algorithm was developed by combining local and nonlocal sparse representation, in which the idea of deterministic annealing (DA) is adopted to optimize the algorithm performance.

Similar to local and nonlocal priors, sparsity priors can also be included into a variational inpainting framework. A sparsity-based variational method, called iterative decoupled inpatinting [36], was proposed. It decouples image inpainting into denoising and linear combination steps. In [37], the intrinsic local sparsity and nonlocal self-similarity of images were simultaneously considered for inpainting using a group-based sparse representation (GSR) model, which achieves better inpainting performance. Furthermore, a patch structure sparsity was adopted to calculate the priority of patch propagation [38]. Then the target patch was represented by a sparse linear combination of multiple candidate patches with high priorities. Compared with diffusion-based and exemplar-based methods, sparsity-based methods are more robust to noises.

2.4 Low-Rank Inpainting

Recently, there has been a growing interest in exploiting low-rank priors of images to deal with various image processing problems [39], [40], [41], [42]. These works assume that image patches, especially texture patches, can be represented by a low-rank matrix. Under this assumption, the intrinsic low-rank textures in an image can be accurately rectified by solving a convex optimization [43]. In fact, image inpainting can be formulated as a low-rank matrix completion problem, which equals to solve a rank minimization. Unfortunately, finding the lowest rank matrix with equality constraints is NP-hard.

In the seminal work of Candès and Recht [6], the nuclear norm was adopted to model the low-rank prior and recover the low-rank matrices from the incomplete set of entries. A theoretical analysis shows that the nuclear norm is the best convex approximation of the rank function of matrices [44]. As a result of the convexity of nuclear norm, the minimization problem can be solved by an iterative scheme. Dong et al. [4] developed a low-rank inpainting algorithm called spatially adaptive iterative singular-value thresholding, in which the missing data are filled in by iteratively shrinking all the singular values of matrices. However, the minimization of nuclear norm, i.e., minimizing all the singular values simultaneously, will lead to an inaccurate estimate of the rank of a matrix in practice. To address this problem, a truncated nuclear norm regularization (TNNR) was proposed for matrix completion [3]. In spite of this, most existing low-rank methods for inpainting rely on iteratively optimizing a cost function with respect to the rank of the patch matrix, which is computationally expensive.

3 TSLRA-Based Inpainting

3.1 Algorithm Overview

In light of the observations mentioned in Section 1, we introduce a patch-based inpainting algorithm by exploiting the nonlocal self-similarity and the low-rank property of natural images, which consists of two stages: *structure completion* and *texture refinement*. The main procedures of the proposed inpainting algorithm are illustrated in Fig. 2.

Given an incomplete image $\mathbf{Y}$ with the missing region $\overline{\Omega}$, the structure of $\overline{\Omega}$ can be recovered by LRA in the first stage called *structure completion*. As a by-product of LRA, the smooth structure $\widehat{\mathbf{Y}}_S$ of the image can also be achieved by applying LRA on the whole image. However, the texture information of the missing region cannot be effectively recovered by directly applying LRA to $\mathbf{Y}$. In the second stage, in order to inpaint the texture of $\overline{\Omega}$, LRA is applied to

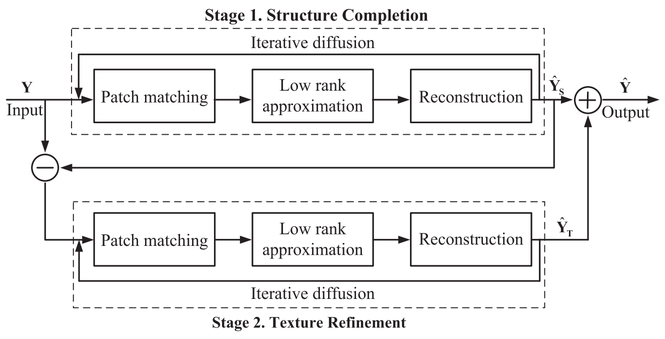

Fig. 2. Flowchart of the proposed inpainting algorithm.

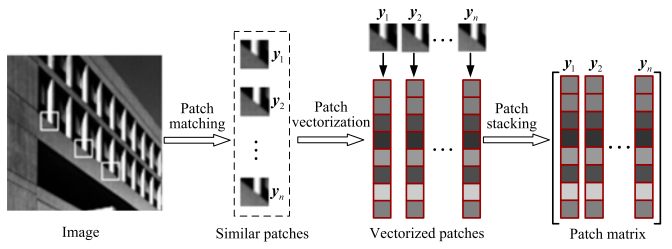

Fig. 3. Illustration of a patch matrix constructed by stacking the similar patches. The white boxes in the left sub-figure denote the similar patches. In the right sub-figure, each red box represents a pixel, and the intensity of each pixel is visualized in a different gray level.

the residual image $\mathcal{P}_\Omega(\mathbf{Y} - \widehat{\mathbf{Y}}_S)$ instead of the image $\mathbf{Y}$. This stage aims to refine the texture information of $\overline{\Omega}$ using the residual image and generate the textural component $\widehat{\mathbf{Y}}_T$. Therefore, it is called *texture refinement*.

More specifically, each stage may be further decomposed into three main steps: *patch matching*, *low rank approximation*, and *reconstruction*. In the step of patch matching, the incomplete image $\mathbf{Y}$ is first divided into N overlapping patches $\{\mathbf{y}_i\}_{i=1}^N$. For each patch $\mathbf{y}_i$, we search its similar patches by a similarity metric. Then we represent each similar patch as a column vector by concatenating all its columns and stack them to obtain a patch matrix $\mathbf{P}_i$. Fig. 3 graphically illustrates the process of constructing a patch matrix. In the step of low rank approximation, a patch matrix with missing data can be efficiently estimated by LRA in SVD domain, i.e., replacing the small nonzero singular values with exact zeros. The effectiveness of LRA arises from the fact that SVD can provide the optimal energy compaction in the least square sense [45], which implies that the missing data can be recovered by taking only the first several singular values and corresponding singular vectors. In the last step, the image is reconstructed by aggregating all processed patches with a weighted averaging scheme.

The major difference between two stages is that different values of parameters are set. For example, in the first stage, a large patch size is used to capture the image structure. On the contrary, a small size is set for capturing the texture patterns in the second stage. More details about parameters setting are given in the next section.

3.2 Structure Completion

3.2.1 Patch Matching Using Pixel Intensity

In our method, the purpose of patch matching is to construct low-rank patch matrices by identifying similar patches in the given image $\mathbf{Y}$. The key issue for patch matching is the choice of features and similarity metrics.

In the literature, a variety of features are adopted to identify similar patches, such as transform coefficients, gradient information, and texture features. In this paper, we utilize pixel intensities as features. The reason is that the pixel intensity is the simplest feature for representing the image structure. On the other hand, we need to specify a metric to quantify the similarity between two patches. A commonly used similarity metric is the squared euclidean distance, i.e., a summation of squared differences. One advantage of this metric is its simplicity of computation. Therefore, we employ the squared euclidean distance to measure the similarity of patches,[2] which is expressed as

$$\mathcal{S}_{i,j} = \|\mathbf{y}_i - \mathbf{y}_j\|_2^2, \tag{3}$$

where $\mathbf{y}_i$ and $\mathbf{y}_j$ denote the target patch and the candidate patch, respectively, and $\|\cdot\|_2$ is the euclidean norm. The smaller $\mathcal{S}_{i,j}$ is, the more similar $\mathbf{y}_i$ and $\mathbf{y}_j$ are.

For each target patch $\mathbf{y}_i$ with size $\sqrt{m} \times \sqrt{m}$, we can find its $n-1$ similar patches (denoted as $\{\mathbf{y}_{i,j}\}_{j=1}^{n-1}$) in the image using the similarity metric above. As illustrated in Fig. 3, then these similar patches are stacked to form a patch matrix $\mathbf{P}_i$, i.e.,

$$\mathbf{P}_i = [\mathbf{y}_i, \mathbf{y}_{i,1}, \dots, \mathbf{y}_{i,n-1}], \tag{4}$$

where each column denotes a patch-vector. Note that some patches forming the patch matrix $\mathbf{P}_i \in \mathbb{R}^{m\times n}$ may be with missing pixels. Without loss of generality, we assume that $m \leq n$. Thus $\mathbf{P}_i$ can be formulated as

$$\mathbf{P}_i = \mathcal{P}_{\Omega_i}(\mathbf{Q}_i), \tag{5}$$

where $\mathbf{Q}_i$ denotes the ideal patch matrix without missing pixels, and $\mathcal{P}_{\Omega_i}$ is the sampling operator used in (1). From the construction process of $\mathbf{P}_i$, it can be seen that our algorithm explicitly exploits the self-similarity of images, which leads $\mathbf{P}_i$ to be low-rank. The task of the next step is to recover the missing pixels in $\mathbf{P}_i$ by LRA. For ease of presentation, we drop the subscript i and write $\mathbf{P}$ and $\mathbf{Q}$ instead of $\mathbf{P}_i$ and $\mathbf{Q}_i$ with a slight abuse of notation.

3.2.2 Recovering Missing Pixels Using Low Rank Approximation

Unlike existing inpainting algorithms based on the nuclear norm minimization, we estimate $\mathbf{Q}$ from its incomplete version $\mathbf{P}$ by solving the following LRA problem

$$\widehat{\mathbf{Q}} = \arg\min_{\mathbf{Q}} \|\mathbf{P} - \mathbf{Q}\|_F^2 \quad s.t. \quad \mathrm{rank}(\mathbf{Q}) = r, \tag{6}$$

where $\mathrm{rank}(\cdot)$ denotes the rank of $\mathbf{Q}$.

Let

$$\mathbf{P} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^T, \tag{7}$$

2. In this similarity metric, the missing pixels are treated as zero.

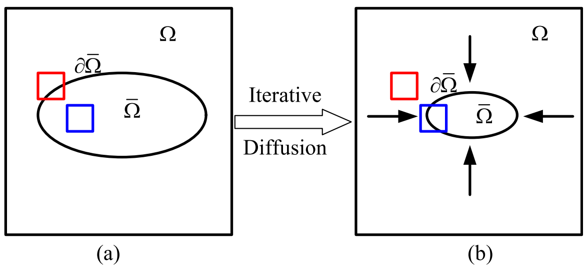

Fig. 4. Illustration of iterative diffusion. Ω is the known region, $\bar{\Omega}$ is the missing region, and the boundary $\partial\bar{\Omega}$ of the missing region is the fill-front of iterative diffusion. The patch (in red) located in the boundary $\partial\bar{\Omega}$ is first filled in, and then the patch (in blue) located in the inner part of $\bar{\Omega}$ is gradually recovered with the increasing iteration.

be the singular value decomposition of $\mathbf{P}$, where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ are orthogonal matrices, and the diagonal matrix $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_n)$ is with the ordered singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Then for $r < n$, the solution of LRA problem (6) is

$$\mathbf{P}_r = \mathbf{U}_r\Sigma_r\mathbf{V}_r^T = \sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T. \tag{8}$$

The following Schmidt-Mirsky theorem, also known as Eckart-Young-Mirsky theorem, shows that $\mathbf{P}_r$ is optimal in both euclidean and Frobenius norm sense.

Theorem 1 (Schmidt-Mirsky). *For any real matrix* $\mathbf{P}$, *if the matrix* $\mathbf{Q}$ *is of rank* r, *then*

$$\|\mathbf{P} - \mathbf{Q}\|_2 \geq \sigma_{r+1},$$

and

$$\|\mathbf{P} - \mathbf{Q}\|_F^2 \geq \sum_{i=r+1}^{n}\sigma_i^2,$$

where $\sigma_i (i = 1, \ldots, n)$ *are the singular values of* $\mathbf{P}$, *and equality is attained for both norms when* $\mathbf{Q} = \mathbf{P}_r$.

See Ref. [46] for a proof.

Owing to the energy compaction property of SVD, the closed-form solution $\mathbf{P}_r$ offers an optimal LRA, i.e., $\widehat{\mathbf{Q}} = \mathbf{P}_r$, and avoids the iterative calculation of SVT as described in (2). The most crucial problem for computing $\mathbf{P}_r$ is to determine the rank r of $\mathbf{P}$, which is a classical problem in the field of signal processing. In [47], Yadav et al. presented a rank estimation method based on Stein's unbiased risk estimator (SURE). Guo et al. [48] used the noise variance to determine the rank of a matrix for denoising. Both methods assume that the noise or error can be described by the Gaussian distribution $\mathcal{N}(0, \tau^2)$. However, this assumption is not valid for inpainting. This is mainly because that the missing pixels depend on the region Ω, and are not independent and identically distributed random variables. Although there exist other methods available for rank estimation, they are generally computationally expensive.

In fact, it is very difficult to determine the accurate value of the rank r. Fortunately, as illustrated in Figs. 1c and 1d, there exists a distinct gap in the singular spectrum of the patch matrix $\mathbf{P}$, and most energy of $\mathbf{P}$ is compacted into several large singular values. Depending upon this observation, we propose an empirical approach to estimate the rank r, which is based on the magnitude ratio of singular values. The value of r is defined as

$$r = \begin{cases} \inf\left\{i \middle| \frac{\sigma_i}{\sigma_{i+1}} < \lambda\right\}, & \text{if } i < m, \\ m, & \text{otherwise}, \end{cases} \tag{9}$$

where λ is a given empirical threshold to determine the gap between σ_i and σ_{i+1}. Unlike existing complicated methods for rank determination, this method is computationally simple and efficient for the low-rank matrix. The threshold λ has an impact on the performance of our algorithm. We will discuss its influence and selection in Section 4.1.

3.2.3 Reconstruction Using Rank-Based Weights

After applying LRA to the patch matrix $\mathbf{P}$, the patches with missing pixels are estimated. Since the patches are chosen to overlap each other to avoid visual artifacts, multiple estimates of each pixel in the image are obtained. Thus, these estimates need to be aggregated to generate a final inpainting image. A simple scheme of aggregating such multiple estimates is to perform a weighted averaging, i.e.,

$$\hat{x}_i = \frac{1}{\mathcal{W}}\sum_{j\in I_i} w_j\hat{x}_{i,j}, \tag{10}$$

where $\hat{x}_{i,j}$, I_i and $\mathcal{W} = \sum_{j\in I_i} w_j$ denote the jth estimate of the pixel x_i, the index set of the patches containing x_i, and a normalizing factor, respectively.

The key issue of aggregation is to determine the weights used in (10). Different from some existing aggregation methods that uses adaptive weights such as exponential weights, variance-based weights, and SURE-based weights [49], the proposed algorithm defines the weights based on the rank of the patch matrix, which can be formulated as

$$w_j = 1 - \frac{r_j - 1}{m}, \tag{11}$$

where r_j is the rank of the jth patch matrix containing x_i.

The reasons for using the rank-based weights are twofold. First, these weights are computationally simple. Due to the fact that the rank estimated in (9) has been calculated in LRA, the weight w_j avoids a complicated calculation and only needs to perform one division and two subtractions. Second, the weights are intuitively reasonable. If a patch matrix is low-rank ($r_j < m$), the patches forming this matrix are linearly correlated. The smaller the rank r_j is, the more reliable the estimates of pixels in this matrix are. On the contrary, if a patch matrix is full-rank ($r_j = m$), there is no correlation among patches and the simple uniform weight $\frac{1}{m}$ is assigned.

3.2.4 Iterative Diffusion

For each patch $\mathbf{y}$ located in the boundary $\partial\bar{\Omega}$ of the missing region $\bar{\Omega}$, we can obtain its estimate by the processing steps described above. However, if a patch is located in the interior of $\bar{\Omega}$, e.g., the blue patch illustrated in Fig. 4a, the estimate of this patch is inaccurate due to lacking reliable information for patch matching. To address this problem, the proposed algorithm propagates the structure information of images

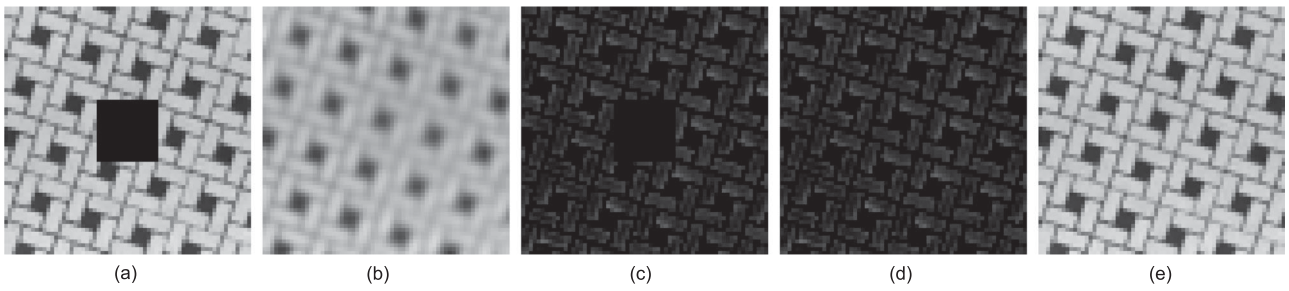

Fig. 5. Visual results of different components of the image *Tile*. (a) Image with block mask $\mathbf{Y}$. (b) Inpainted structural component $\widehat{\mathbf{Y}}_S$. (c) Residual texture $\mathbf{Y} - \mathcal{P}_\Omega(\widehat{\mathbf{Y}}_S)$. (d) Inpainted textural component $\widehat{\mathbf{Y}}_T$. (e) Final output $\widehat{\mathbf{Y}}$.

into the target region via iterative diffusion, which is also used by other existing inpainting algorithms.

The idea of iterative diffusion is to use the output image at the previous iteration as initialization for the next iteration. Specifically, the iterative diffusion process can be expressed by the following steps.

1) For each patch with missing pixels $\mathbf{y}_i$, identify its $n-1$ similar patches by the similarity metric (3), and then construct its patch matrix $\mathbf{P}_i$ by (4).
2) Fill in the missing pixels of $\mathbf{y}_i$ by solving the LRA problem (6).
3) Update the patch $\mathbf{y}_i$ by performing a weighted averaging formulated in (10).
4) Repeat steps 1 to 3 until all missing pixels are recovered.

Fig. 4 illustrates an iterative diffusion process, in which the missing region $\overline{\Omega}$ is gradually filled in by an iterative diffusion from the boundary $\partial\overline{\Omega}$ inwards. By applying a few iterations of structure diffusion, a number of pixels in the blue patch are recovered, as illustrated in Fig. 4b, which makes patch matching for the blue patch possible. It is important to point out that, in the real implementation, we directly apply LRA for all pixels located in the missing region $\overline{\Omega}$ to avoid detecting the boundary $\partial\overline{\Omega}$. In this process, the patches located in the boundary $\partial\overline{\Omega}$ can be first recovered, and the patches located in the interior of $\overline{\Omega}$ cannot be accurately estimated by LRA due to lacking reliable information for finding similar patches. But with the number of iteration increasing, the patches located in the interior can be incrementally filled in.

3.3 Texture Refinement

Although the missing regions of an incomplete image can be filled in by the structure completion described above, the fine texture of the inpainting region is unsatisfactory. The major reason is that the fine texture is compacted into a few small singular values in SVD domain, and these details are lost in LRA process by replacing the small singular values with exact zeros. Therefore, the output of LRA can be viewed as a smooth approximation of the desired result.

Inspired by the simultaneous filling-in of structures and textures [50], we refine texture details of the missing regions by propagating the texture information of an observed image. Let $\mathbf{Y}_S$ and $\mathbf{Y}_T$ denote the structural component and the textural component of an observed image $\mathbf{Y}$, respectively. Then a simple relation between $\mathbf{Y}_S$ and $\mathbf{Y}_T$ can be represented by a linear model $\mathbf{Y} = \mathbf{Y}_S + \mathbf{Y}_T$. A key problem for texture refinement is how to separate the textural component $\mathbf{Y}_T$ from the image $\mathbf{Y}$. Different from the variational methods that decompose an image into a structural part and a textural part by solving a minimization problem [32], [50], we first generate a structural image $\widehat{\mathbf{Y}}_S$ by directly applying LRA to all patch matrices of the output of structure competion and reconstructing the processed patches. Due to the smooth property of LRA (See [51] for more details), $\widehat{\mathbf{Y}}_S$ consists of the local structural component with little texture details. Once the structural component $\widehat{\mathbf{Y}}_S$ is obtained, the textural part can be produced by

$$\mathbf{Y}_T = \mathbf{Y} - \mathcal{P}_\Omega(\widehat{\mathbf{Y}}_S). \tag{12}$$

For the textural component $\mathbf{Y}_T$, texture refinement is implemented by performing the same procedures in the stage of structure completion, i.e., *patch matching*, *low rank approximation*, *reconstruction*, and *iterative diffusion*. After completing the stage of texture refinement, the final result $\widehat{\mathbf{Y}}$ is then reconstructed by adding back the outputs of two process stages, i.e.,

$$\widehat{\mathbf{Y}} = \widehat{\mathbf{Y}}_S + \widehat{\mathbf{Y}}_T, \tag{13}$$

where $\widehat{\mathbf{Y}}_T$ is the result of texture refinement.

Fig. 5 gives an example of the application of TSLRA to the image *Tile*. We can find that the structural component reconstructed by LRA in the first stage contains little texture. This is because LRA is approximately equivalent to a smooth regularization operator [51]. On the other hand, the textural component can be modeled by a low-rank prior [52]. This means that the texture details can be recovered by applying LRA to the residual texture image. Fig. 5d clearly shows that the textural component of the image *Tile* is well recovered by texture refinement.

Note that different patch sizes are used for structure completion and texture refinement in this paper. It arises from the multiscale nature of images. The structural and textural components of natural images appear significantly different in scales. In general, a structural component corresponds to large objects in the image, and a textural part contains fine-scale details. Therefore, a large patch size (large-scale) is helpful for capturing the varying local structure, and a small patch size (fine-scale) can capture the textural patterns well. Under the low rank assumption of similar patches, both the structural component and the textural component can be recovered by LRA at different scales. Finally, the complete TSLRA algorithm is summarized in Algorithm 1. It is necessary to point out that, in Algorithm 1, we directly perform component decomposition on all image patches to avoid detecting the boundary $\partial\overline{\Omega}$.

4 Experiments

To validate the performance of the proposed inpainting algorithm, we apply it on a variety of natural images with different inpainting tasks: block completion, text removal, and randomly missing pixels filling. Moreover, we compare our algorithm with five state-of-the-art inpainting algorithms: NLM[3] [31], BPFA[4] [34], DA[5] [35], GSR[6] [37], and TNNR[7] [3]. These algorithms use different priors to fill in the missing data. It is necessary to point out that all experiments are performed under Matlab R2010b with Intel Core i7-4790 CPU 3.60 GHz and 16 GB RAM, except that the experimental results of NLM are obtained with an online demo implemented by the authors. To make a fair comparison between algorithms, the tunable parameters in these five algorithms are manually selected to achieve good comprehensive performance in terms of quantitative criteria, visual assessment, and computational cost.

To evaluate the quality of recovered images, peak signal-to-noise (PSNR) and feature-similarity (FSIM) index are used as quantitative metrics. Although PSNR is often inconsistent with human eye perception, it is a widely-used criterion for quantitatively evaluating image quality, and is at least partially related to the perceptual quality. Based on the fact that human visual system understands an image quality according to its low-level features, FSIM measures the similarity between two images by combining the phase congruency feature and the gradient magnitude feature. The higher PSNR and FSIM values mean the better quality.

4.1 Parameter Analysis

Our method contains four main algorithmic parameters: the size of patches m, the number of columns in each patch matrix n, the number of iteration K, and the threshold λ used in (9). An important issue is how to determine appropriate values for these parameters. To evaluate the influence of m, we fix the other three parameters and perform TSLRA on the test images with different values of m, where the test images include 8 grayscale images for block completion, 10 for text removal, and 5 for random missing pixels filling. These images are also used in the following inpainting experiments and computational time evaluation. Fig. 6 plots the average PSNR curves over the test images for different stages of inpainting tasks. It can be seen that the optimal value for the patch size m is obviously different in each stage of TSLRA. As mentioned before, a large patch size is helpful for capturing the local structure, and a small patch size is for texture details. Therefore, the patch size m needs to be differently set for stages 1 and 2. The parameters n and K also play important roles in our algorithm. By evaluating the influences of n and K, we find that they should be set according to different inpainting applications. The values of parameters m, n and K used in our experiments are listed in Table 1. Note that the parameter n varies over a wide range for texture images and non-texture cases. It is because the similarity metric based on the euclidean distance (3) is incapable of efficiently capturing the intrinsic similarity between texture patterns. For texture images, increasing the number of patches n will lead to some dissimilar patches being included in the patch matrices, which will decrease the accuracy of LRA. So we choose a small value of n for texture cases. Empirically, we find that using the parameters with $m = 18$, $n = 60$, $K = 200$ for stage 1 and $m = 10$, $n = 60$, $K = 14$ for stage 2 can make our algorithm more practical at the cost of a slight performance degradation. In addition, Fig. 7 gives the evolution of PSNR as a function of the threshold λ. It can be observed that the proposed TSLRA is insensitive to λ in the range from 1.05 to 1.1. Thus, it is empirically set to be 1.06, which is sufficient in most cases.

3. Demo available at http://demo.ipol.im/demo/136
4. Codes available at http://mingyuanzhou.github.io/Code.html
5. Codes available at http://www.csee.wvu.edu/%7Exinl/demo/inpainting.html
6. Codes available at http://idm.pku.edu.cn/staff/zhangjian/GSR
7. Codes available at https://sites.google.com/site/zjuyaohu

Algorithm 1. Inpaiting Algorithm with TSLRA

```
Input: Incomplete image $\mathbf{Y}$, maximum iteration number $K$
Output: Inpainting image $\widehat{\mathbf{Y}}$
1:  // Stage 1: Structure Completion
2:  Initialization $\widehat{\mathbf{Y}} = \mathbf{Y}$, $k = 1$;
3:  while $k \leq K$ do
4:      for each $i \in \overline{\Omega}$ do
5:          $\mathbf{P}_i \leftarrow$ Identify the similar patches of $i$th patch from the
            image $\widehat{\mathbf{Y}}$ to construct a patch matrix, as described in (3)
            and (4);
6:          $\widehat{\mathbf{Q}}_i \leftarrow$ Calculate the LRA of $\mathbf{P}_i$ by (8);
7:          $w_i \leftarrow$ Calculate the weights by (11);
8:      end for
9:      $\widehat{\mathbf{Y}}(\overline{\Omega}) \leftarrow$ Reconstruct the missing pixels by
        aggregating the estimates by (10);
10: end while
11: // Component Decomposition
12: for each $j \in (\Omega \cup \overline{\Omega})$ do
13:     $\mathbf{P}_j \leftarrow$ Identify the similar patches of $j$th patch from the
        image $\widehat{\mathbf{Y}}$ to construct a patch matrix;
14:     $\widehat{\mathbf{Q}}_j \leftarrow$ Calculate the LRA of $\mathbf{P}_j$;
15:     $w_j \leftarrow$ Calculate the weights;
16: end for
17: $\widehat{\mathbf{Y}}_S \leftarrow$ Reconstruct the image by aggregating patches;
18: $\mathbf{Y}_T \leftarrow$ Calculate the incomplete texture image by (12);
19: // Stage 2: Texture Refinement
20: Initialization $\widehat{\mathbf{Y}}_T = \mathbf{Y}_T$, $k = 1$;
21: while $k \leq K$ do
22:     for each $i \in \overline{\Omega}$ do
23:         $\mathbf{P}_i \leftarrow$ Identify the similar patches of $i$th patch from the
            image $\widehat{\mathbf{Y}}_T$ to construct a patch matrix;
24:         $\widehat{\mathbf{Q}}_i \leftarrow$ Calculate the LRA of $\mathbf{P}_i$;
25:         $w_i \leftarrow$ Calculate the weights;
26:     end for
27:     $\widehat{\mathbf{Y}}_T \leftarrow$ Reconstruct the texture image by
        aggregating patches;
28: end while
29: $\widehat{\mathbf{Y}} \leftarrow$ Obtain the final inpainting image by (13).
```

4.2 Block Completion

In this section, the missing block cases of image inpainting are first considered. We use eight 64×64 test images with an isolated 16×16 block to validate the effectiveness of our algorithm, which are also used in [35]. In Table 2, we quantify the average performances of competing algorithms for

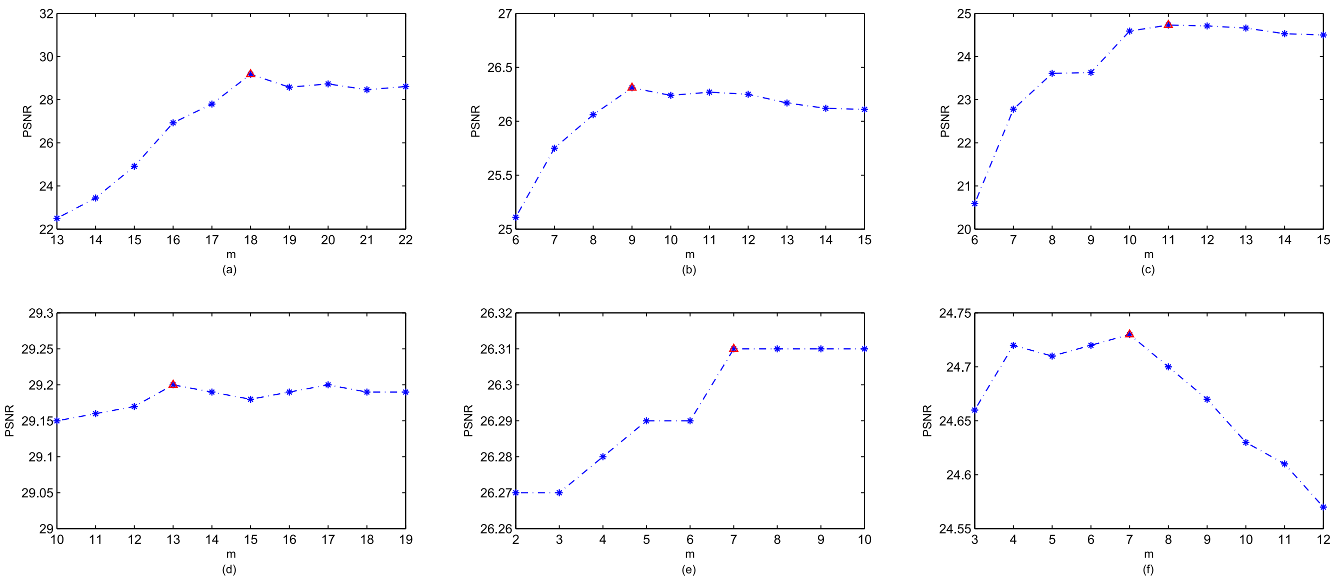

Fig. 6. Influence of the patch size m on the PSNR for different stages of inpainting tasks. We set $\lambda = 1.06$ and fix the parameters n and K for different tasks as listed in Table 1. (a) Stage 1 of block completion. (b) Stage 1 of random missing pixels filling. (c) Stage 1 of text removal. (d) Stage 2 of block completion. (e) Stage 2 of random missing pixels filling. (f) Stage 2 of text removal. The red triangle denotes the optimal value.

test images in terms of PSNR and FSIM. It is obvious to observe that TSLRA achieves the best PSNR and FSIM values on the average. Our algorithm outperforms the second best one, i.e., DA, by 0.33 dB in PSNR and by 0.0241 in FSIM.

The visual comparisons of different inpainting algorithms are shown in Fig. 8. As can be seen from it, the results by the proposed TSLRA is quite comparable to DA, and clearly superior to other competing methods, especially for the images with high repeating patterns such as *Barb* and *Tile*. These images contain a number of redundant information, namely self-similarity, which are explicitly used in our algorithm by constructing the low-rank patch matrix. Although GSR achieves the best visual quality for *Lena* image, as shown in Fig. 8f, it oversmoothes the texture on *Tile* and the edge on *Boat*. For the images *Lena* and *Roof*, NLM produces block artifacts. Besides, BPFA is not suitable to recover a large missing region, which has a worst performance in terms of both quantitative criteria and visual quality. From Fig. 8d, we can find that TNNR fails in recovering the missing block. The reason is that TNNR applies the low-rank constraint to a whole image. In fact, the performance of TNNR significantly depends on the rank of the image, which needs to be predefined before performing it. However, it is hard to accurately determine the rank of a whole image. Different from TNNR, the proposed TSLRA applies the low-rank constraint to each patch matrix by empirically estimating the rank, which leads to a significant improvement in the visual quality of inpainting results.

TABLE 1
Parameters Used in the Experiments

Task	Image type	Parameters					
		Stage 1			Stage 2		
		m	n	K	m	n	K
Block	Texture	18	14	280	13	60	14
	Non-texture	18	80	300	13	60	14
Text	All	11	70	100	7	60	14
Random	All	9	70	70	7	60	14

4.3 Text Removal

We now present the experimental results on ten grayscale images in *BM3D_images* for text removal. In this experiment, the text may cover not only structure information but also texture. Figs. 9 and 10 show the visual results generated by different algorithms on *Lena* and *Pirate*. These two images are good test images because they contain a nice mixture of smooth region, texture, and detail. Table 2 lists the quantitative results on test images with two text masks used in Figs. 9 and 10. Overall, TSLRA is comparable to GSR and DA, and outperforms other competing methods in terms of both visual quality and quantitative metrics.

Unlike the cases in block completion, BPFA is capable of removing the texts, which is superior to TNNR and NLM, and comparable to GSR, DA, and TSLRA. For the image

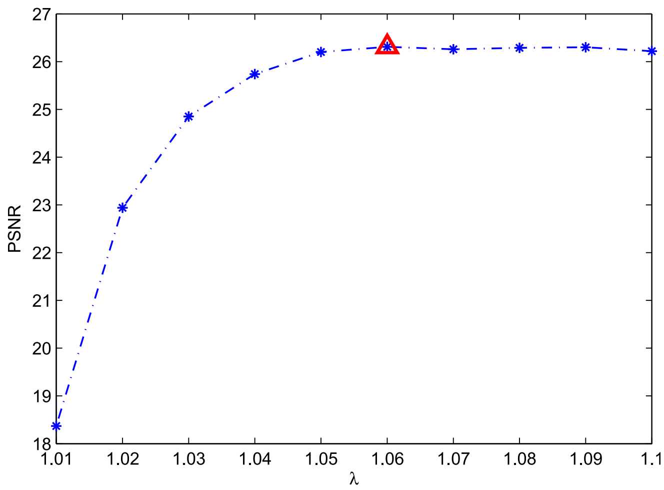

Fig. 7. Influence of the threshold λ on the PSNR. The red triangle denotes the optimal value.

TABLE 2
Average PSNR and FSIM Values by Inpainting Methods on Test Images for Different Tasks

Task		BPFA [34]	TNNR [3]	NLM [31]	GSR [37]	DA [35]	TSLRA
Block		12.78/0.6903	17.27/0.7693	25.95/0.9203	26.46/0.8428	28.85/0.9247	**29.18/0.9488**
Text	Mask 1	25.28/0.8968	20.76/0.7940	23.86/0.8627	**26.22/0.9141**	25.34/0.9024	25.67/0.9053
	Mask 2	23.24/0.8561	18.95/0.7391	22.68/0.8321	23.48/0.8701	23.45/0.8629	**23.79/0.8712**
Random	10%	21.74/0.8192	15.99/0.5959	N/A	**22.91/0.8626**	22.06/0.8230	22.02/0.8320
	20%	24.25/0.8786	18.75/0.6897	N/A	**25.71/0.9189**	25.35/0.8850	**25.71**/0.9004
	30%	26.14/0.9116	20.97/0.7600	N/A	**28.71/0.9483**	27.27/0.9156	27.82/0.9309
	40%	28.22/0.9365	22.92/0.9165	N/A	**30.54/0.9590**	28.95/0.9389	29.39/0.9499

The best results are highlighted in bold.

Lena shown in Fig. 9, the PSNR values for BPFA, TNNR, NLM, GSR, DA, and TSLRA are 26.34, 21.17, 24.86, 27.53, 26.91, and 27.02 dB, respectively. From Fig. 9, it can be seen that the proposed TSLRA achieves highly competitive performance in visual quality. For the image *Pirate* shown in Fig. 10, the PSNR values for BPFA, TNNR, NLM, GSR, DA, and TSLRA are 19.54, 15.89, 18.91, 19.18, 19.26, and 19.71 dB, respectively. Our TSLRA has the highest PSNR value, and the gain over BPFA is 0.17 dB. Besides, as shown in Figs. 9d and 10d, TNNR still does not work well, which produces visual artifacts.

4.4 Random Missing Pixels Filling

We test the performance of our inpainting algorithm on recovering random missing pixels, and report the experimental results for five commonly used grayscale images (size 256×256) with different observed percentage. Our algorithm is also compared with BPFA, TNNR, GSR and DA to demonstrate its effectiveness.[8]

The PSNR and FSIM values of the results obtained by different algorithms are listed in Table 2. It can be observed that GSR achieves the highest PSNR and FSIM values on the average, and TSLRA takes the second place. Overall, TSLRA is robust to random pixels missing with low observed percentage. Additionally, BPFA also works in this case, while the corresponding PSNR and FSIM values obtained by BPFA are inferior to GSR, DA, and TSLRA.

For the visual quality comparisons, Figs. 11 and 12 provide the results by the competing methods from only 20 percent random samples. For the image *Barbara* with rich textures, as can be seen from Fig. 11, DA and TSLRA can well recover the fine textures and provide better visual quality than other methods. As shown in Figs. 11c and 11e, BPFA and GSR introduce some visual artifacts. We also observe that TNNR fails in recovering the random missing pixels, which generates incorrect textures. For the image *Peppers* with smooth structures, our TSLRA is competitive in the visual quality with GSR, and clearly superior to BPFA, TNNR, and DA. The visual result provided by BPFA is with some artifacts and blurred edges. Meanwhile, TNNR again fails in this case, which further verifies that applying the low-rank constraint to a whole image is not effective for image inpainting.

8. The original codes of NLM is not applicable to fill in random missing pixels. Here we do not provide the results of NLM.

4.5 Computational Cost

We next evaluate the computational efficiency of TSLRA. Table 3 compares the average running time of our algorithm to that of three state-of-the-art algorithms on the aforementioned tasks. As depicted in this table, the running time of TSLRA is lower than other algorithms. Note that the computational cost reported here does not include that of TNNR and NLM, because TNNR does not obtain satisfactory results as discussed in previous sections and the results of NLM are generated by an online demo implemented by the authors. For random missing pixels filling, smaller observed percentage of pixels is generally associated with longer running time. However, BPFA achieves a counter-intuitive result. This is because, in its implementation, BPFA uses sparse multiplication to eliminate unnecessary computation. A smaller observed percentage means that the known data is more sparse. Consequently, the running time of BPFA versus observed percentage is monotonically increasing. Compared with BPFA, DA and GSR (competitive with our method), a nonoptimized implementation of TSLRA significantly reduces the computational complexity.

The main computational cost of TSLRA is the calculation of SVD for each patch matrix, which takes about 73 percent of the running time. Roughly 25 percent of the time is spent in performing the patch matching to search similar patches, and the running time of the reconstruction step can be negligible. Due to the fact that SVD of each patch matrix is calculated independently, a direct method for accelerating our algorithm can be achieved by a parallel GPU implementation. Besides, for accelerating the LRA of patch matrix with $r \ll m$, a less expensive alternative is to use Lanczos bidiagonalization process [53] for only computing several dominant singular values and the corresponding singular vectors, which is fast and efficient as there is no need of the computation of the complete SVD. For reducing the computational complexity of searching similar patches, the most commonly used strategy is to find the approximate nearest neighbors (ANN) instead of the exact nearest ones. Thus some popular techniques for ANN, such as generalized PatchMatch [54], PatchTable [24], and kd-tree [55], can be employed to further speed up TSLRA.

5 More Analysis

5.1 Extension to Color Image Inpaingting

While mainly designed for grayscale images, with a trivial modification, the proposed TSLRA also offers good

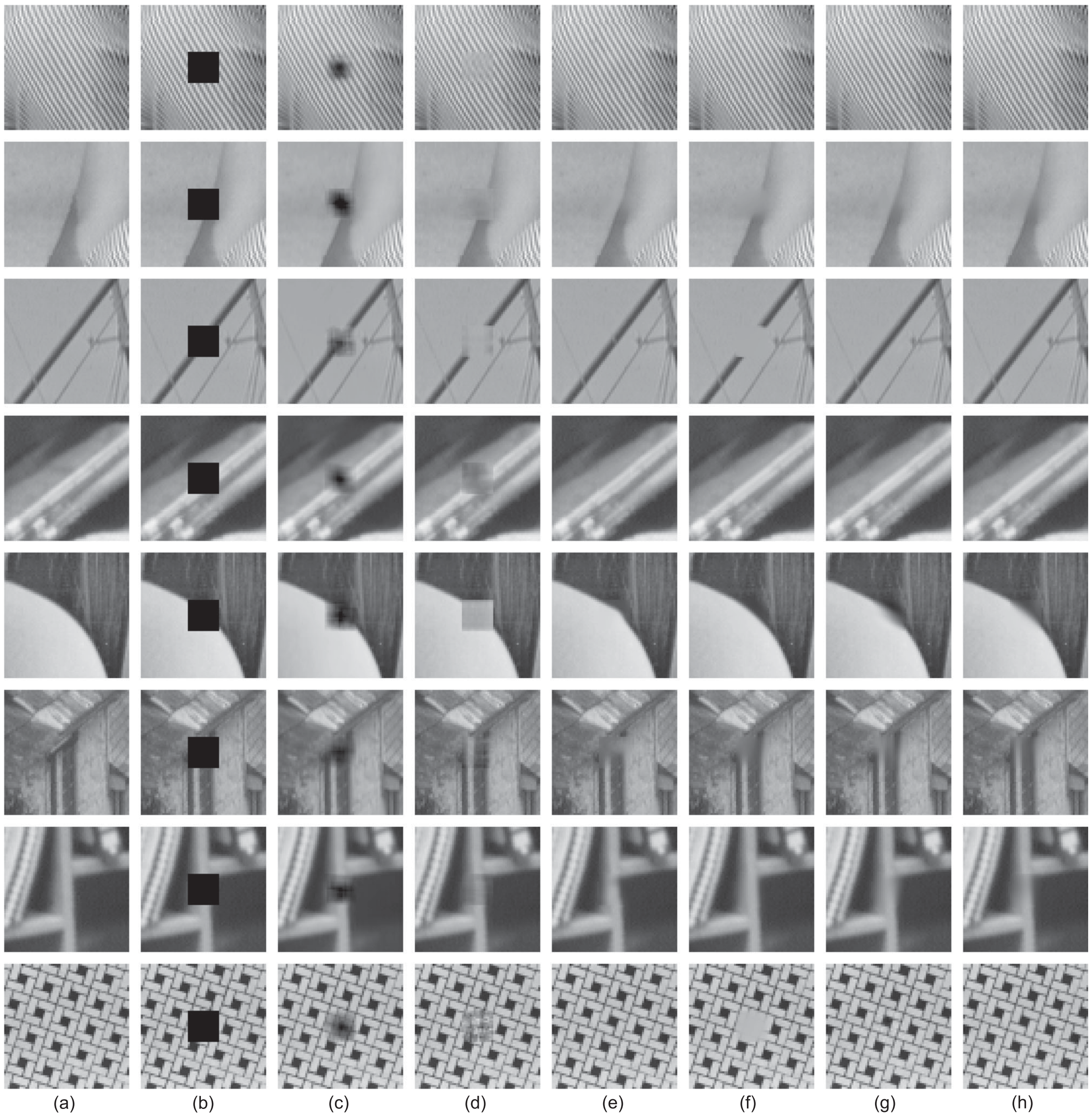

Fig. 8. Visual comparisons for missing block completion. (a) Original images, from top to bottom: *Barb*, *Barb3*, *Boat*, *Book*, *Lena*, *Roof*, *Shelf*, *Tile*. (b) Images with block mask. (c) BPFA. (d) TNNR. (e) NLM. (f) GSR. (g) DA. (h) TSLRA.

performance in inpainting color images. A naive extension of our method to color images would be to apply it on the R, G, B channels independently. We here perform inpainting in YCbCr color space. Due to the fact that the human visual system is more sensitive to luminance information, we compute the patch similarity using only the Y channel and apply it separately on each channel to construct the low-rank patch matrices. Then the missing pixels are recovered by perfroming TSLRA on these patch matrices. Fig. 13 shows two cases of color image inpainting, which further validates the effectiveness of TSLRA. Note that, in the experiments, we assume that the target region is known. This assumption is also used in NLM, BPFA, DA, GSR, and TNNR. In real applications, the target region can be automatically localized by a detection algorithm or manually specified by the user.

5.2 Differences Between TSLRA and Other Methods

The main differences between the proposed TSLRA and other methods are threefold. 1) The existing low-rank methods recover the missing data by solving a nuclear norm minimization subject to data constraints, in which a convergence result is derived by an iterative SVT algorithm. To avoid the time-consuming iterative SVT, TSLRA uses LRA to derive a closed-form solution by directly truncating the small singular values, and an empirical approach based on the magnitude ratio of singular values is given to estimate the rank. Owing to the energy compaction property of SVD,

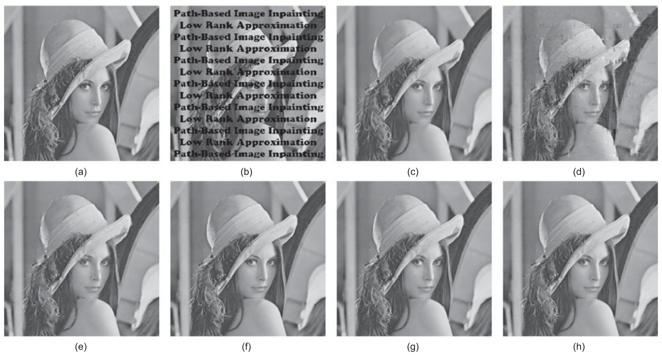

Fig. 9. Visual comparison for text removal on *Lena256*. (a) Original image. (b) Image with text mask 1. (c) BPFA (PSNR=26.34dB). (d) TNNR (PSNR=21.17dB). (e) NLM (PSNR=24.86dB). (f) GSR (PSNR=**27.53**dB). (g) DA (PSNR=26.91dB). (h) TSLRA (PSNR=27.02dB).

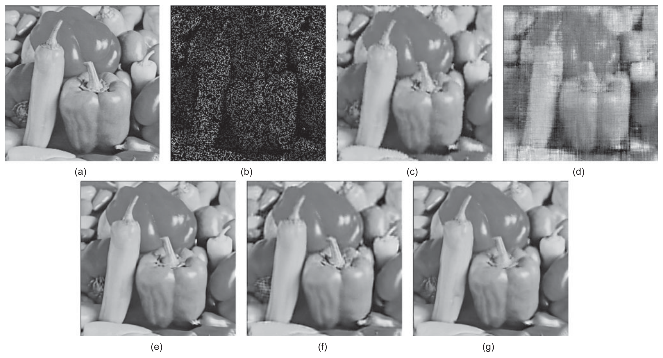

Fig. 12. Visual comparison for random missing pixels filling on *Peppers256*. (a) Original image. (b) Image with 80 percent random missing pixels. (c) BPFA (PSNR=26.08dB). (d) TNNR (PSNR=20.64dB). (e) GSR (PSNR=**28.43**dB). (f) DA (PSNR=27.73dB). (g) TSLRA (PSNR=27.27dB).

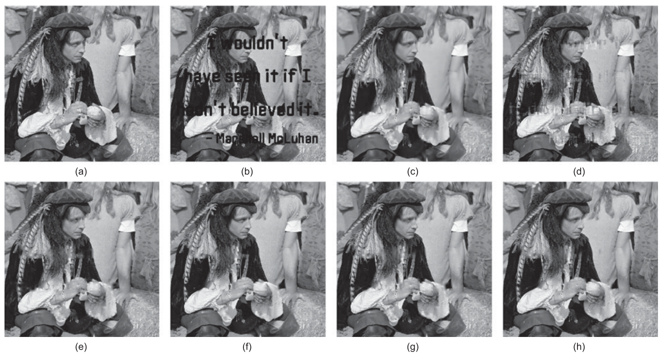

Fig. 10. Visual comparison for text removal on *Pirate256*. (a) Original image. (b) Image with text mask 2. (c) BPFA (PSNR=19.54dB). (d) TNNR (PSNR=15.89dB). (e) NLM (PSNR=18.91dB). (f) GSR (PSNR=19.18dB). (g) DA (PSNR=19.26dB). (h) TSLRA (PSNR=**19.71**dB).

TABLE 3
Average Running Time (in Seconds) of Different Algorithms

Method	Block	Text	Random (Observed percentage)			
			10%	20%	30%	40%
BPFA [34]	81	2,341	1,063	1,451	2,147	3,007
DA [35]	78	1,145	13,452	11,959	9,973	8,861
GSR [37]	73	1,416	3,086	2,035	1,568	1,313
TSLRA	67	952	1,036	940	824	689

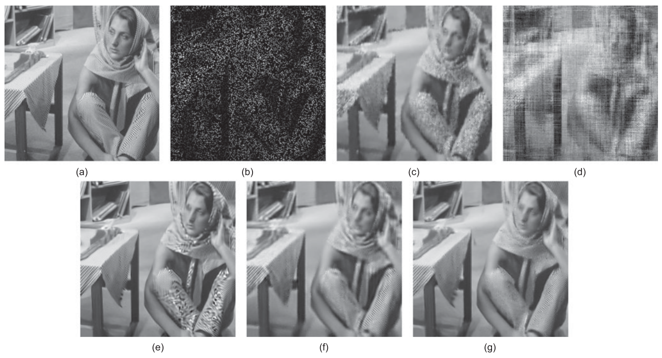

Fig. 11. Visual comparison for random missing pixels filling on *Barbara256*. (a) Original image. (b) Image with 80 percent random missing pixels. (c) BPFA (PSNR=22.52dB). (d) TNNR (PSNR=18.42dB). (e) GSR (PSNR=23.02dB). (f) DA (PSNR=24.68dB). (g) TSLRA (PSNR=**25.35**dB).

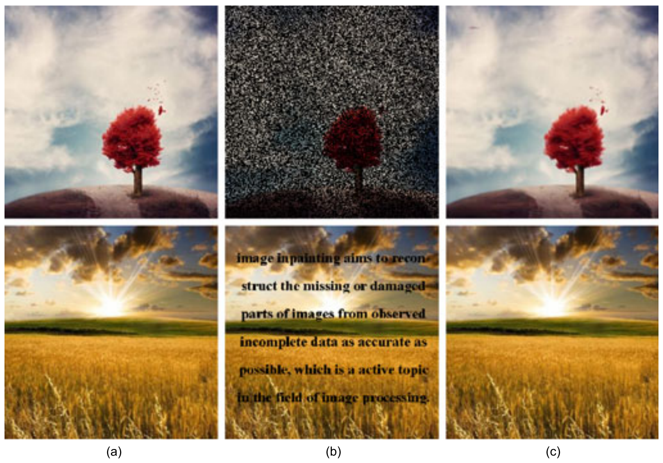
image inpainting aims to reconstruct the missing or damaged parts of images from observed incomplete data as accurate as possible, which is a active topic in the field of image processing.
(a)
(b)
(c)

Fig. 13. Color image inpainting. (a) Original. (b) Masked. (c) Our results.

the solution is optimal in both euclidean and Frobenius norm sense. 2) Different from traditional image decomposition methods on the basis of variational principles, our method applies LRA to separate the structure and texture parts of images. Based on the observation that the singular values of structure and texture parts exhibit different empirical distributions, a two-stage LRA scheme is designed to successively recover the structure and texture in the missing regions. 3) Using the rank estimated in the process of calculating LRA of patch matrices, we define the rank-based weights for the aggregation, which are more robust to noises and computationally simpler than similarity-based weights.

5.3 Limitations

While the proposed inpainting method achieves an impressive performance to various inpainting applications, there are some cases where it fails to recover the missing information. As described in Section 1, our method relies on the low-rank property of image patches, which arises from the nonlocal self-similarity of images and assumes that there exist a number of similar patches in the image. When this underlying assumption is violated, TSLRA might produce unreasonable structures and textures. Fig. 14 shows a failure case, in which the missing structure is irregular and there is few patch redundancy within the image. More precisely, this image exhibits redundancy over a rotation transformation. However, the similarity metric used in TSLRA does not take the rotation transformation into account.

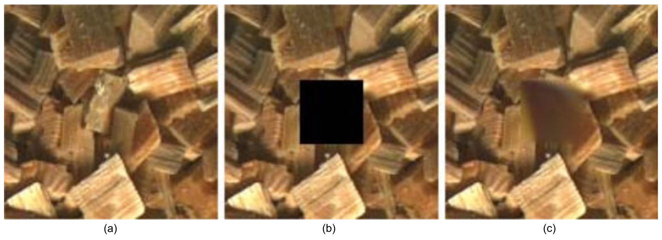

Fig. 14. Failure case. (a) Original. (b) Masked. (c) Our result.

Besides, Figs. 15c, 15d, and 15e show some inpainting results of different automatic methods for object removal. It can be observed that although the visual quality of TSLRA is slightly better than GSR, both of them oversmooth the near-random texture (e.g., meadow), and the results generated by *Content-Aware Fill* in Adobe Photoshop CS5 are clearly superior to GSR and TSLRA. In general, TSLRA performs well on regular or near-regular textures. In such cases, the texture patches can be interpreted to be low-rank, and so we can recover the missing texture by applying LRA to the residual texture image which contains most of high-frequency details of the original image. But it has difficulty in reconstructing irregular or near-random texture patterns due to the low-rank assumption being violated. It is because the euclidean metric (3) is inappropriate to measure the similarity between irregular texture patterns so that the patch matrices do not meet the low-rank property. In fact, measuring the similarity between image patches still remains a challenging task and an active research field.

5.4 Possible Improvements

In TSLRA, we convert the problem of image inpainting into low-rank matrix completion by identifying the similar patches and stacking them together in a patch matrix, and then estimate the missing pixels in the patch matrix by solving a minimization problem subject to a low-rank constraint. It only takes the nonlocal self-similarity prior and the low-rank prior into account. However, for the images with irregular texture patterns, statistical properties of images (e.g., patch offset statistics [26]) could be more appropriate. Therefore, one can incorporate such statistics into the low-rank method to improve inpainting performance.

Additional user interventions can greatly improve the inpainting quality, especially for large missing regions [23]. One way of user intervention is that the user draws a few curves across the target region and guides the missing information along the curves inside the target region [21]. Another alternative is to specify particular content for some portions of the target region and utilize the known region and the user-specified content to fill the remaining missing region. Here we select the latter due to its simple implementation. Fig. 15f shows some interactive inpainting results of TSLRA, in which we manually specify a part of the left-side available content as a part of the target region and then apply TSLRA to inpaint the gap between the known region and the specified part. It can be seen that TSLRA produces visually plausible results. Although this paper mainly focuses on the automatic inpainting, in real applications, interactive guidance used in [21] can be also integrated into our approach by specifying a few curves to guide the similar patch search. Besides, for inpainting an image that exhibits little self-similarity, one can also boost TSLRA with the help of a large external patch dataset or internet images (as used in [24], [56]).

6 Conclusion

Low-rank priors have achieved great success in the field of image restoration. In this paper, we have proposed an image inpainting algorithm that takes advantage of the low-rank representation. By grouping similar patches of a damaged image, we can construct many low-rank patch matrices. Then the missing pixels in each patch matrix are recovered by applying LRA, which makes use of the optimal energy compaction property of SVD. To obtain better visual quality, TSLRA is exploited to recover image structures and refine texture details, respectively. Different from traditional low-rank methods that rely on an iterative thresholding algorithm, the proposed inpainting method is intrinsically simple. Due to avoiding the iterative singular value shrinkage process, it can bring a significant reduction of computational cost for recovering each corrupted patch matrix. Meanwhile, experimental results show that the proposed method is competitive with some state-of-the-art inpainting algorithms.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments. This work was supported by Natural Science Foundation of China (61472220, 61402261, and 61332015), NSFC Joint Fund with Guangdong under Key Project (U1201258) and with Zhejiang under Key Project (U1609218), Key Technology Research and Development

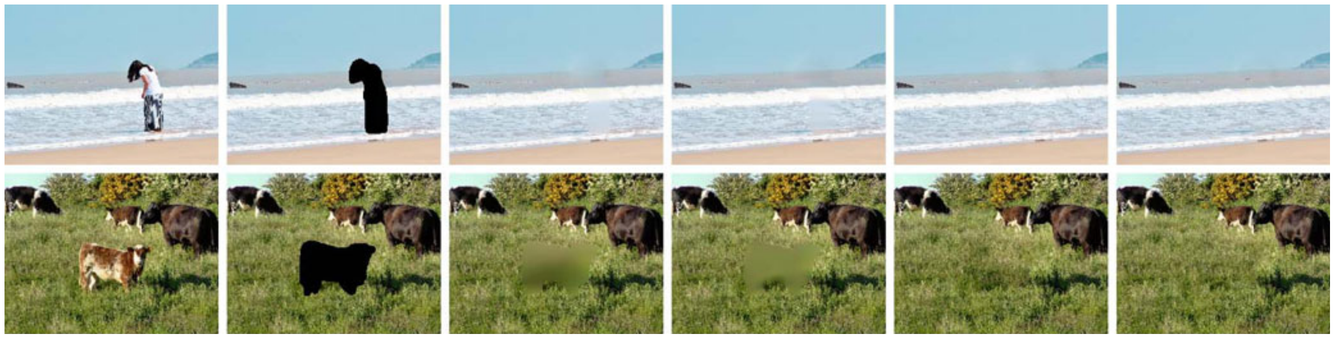

Fig. 15. Object removal. (a) Original. (b) Masked. (c) GSR. (d) TSLRA. (e) Content-Aware Fill. (f) TSLRA with user intervention.

Program of Shandong Province (2015GGX101004), Natural Science Foundation of Shandong Province (ZR2016FM21), Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

[1] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray, "Exemplar-based inpainting: Technical review and new heuristics for better geometric reconstructions," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1809–1824, Jun. 2015.

[2] L. Karacan, E. Erdem and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 176:1–176:11, Nov. 2013.

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

[4] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, Feb. 2013.

[5] W. Li, L. Zhao, Z. Lin, D. Xu, and D. Lu, "Non-local image inpainting using low-rank matrix completion," *Comput. Graph. Forum*, vol. 34, no. 6, pp. 111–122, Sep. 2015.

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

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

[8] C. Guillemot and O. L. Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, Jan. 2014.

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Techn.*, Jul. 2000, pp. 417–424.

[10] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's," *Int. J. Comput. Vis.*, vol. 68, no. 1, pp. 65–82, Jun. 2006.

[11] J. Shen and T. Chan, "Mathematical models for local nontexture inpaintings," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, Feb. 2002.

[12] J. Shen, S. H. Kang, and T. Chan, "Euler's elastica and curvature-based inpainting," *SIAM J. Appl. Math.*, vol. 63, no. 2, pp. 564–592, Jan. 2003.

[13] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001.

[14] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, vol. 2, pp. 1033–1038.

[15] S. M. Hu, F. L. Zhang, M. Wang, R. R. Martin, and J. Wang, "PatchNet: A patch-based image representation for interactive library-driven image editing," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 196:1–196:12, Nov. 2013.

[16] S. J. Luo, Y. T. Sun, I. C. Shen, B. Y. Chen, and Y. Y. Chuang, "Geometrically consistent stereoscopic image editing using patch-based synthesis," *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 1, pp. 56–67, Jan. 2015.

[17] W. Dong, N. Zhou, T. Y. Lee, F. Wu, Y. Kong, and X. Zhang, "Summarization-based image resizing by intelligent object carving," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 1, pp. 111–124, Jan. 2014.

[18] C. Barnes and F. L. Zhang, "A survey of the state-of-the-art in patch-based synthesis," *Comput. Visual Media*, vol. 3, no. 1, pp. 3–20, Mar. 2017.

[19] F. L. Zhang, J. Wang, E. Shechtman, Z. Y. Zhou, J. X. Shi, and S. M. Hu, "PlenoPatch: Patch-based plenoptic image manipulation," *IEEE Trans. Vis. Comput. Graph.*, vol. 23, no. 5, pp. 1561–1573, May 2017.

[20] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[21] J. Sun, L. Yuan, J. Jia, and H. Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, Jul. 2005.

[22] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[23] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24:1–24:11, Aug. 2009.

[24] C. Barnes, F. L. Zhang, L. Lou, X. Wu, and S. M Hu, "PatchTable: Efficient patch queries for large datasets and applications," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 97:1–97:10, Aug. 2015.

[25] J. B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image completion using planar structure guidance," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 129:1–129:10, Jul. 2014.

[26] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2423–2435, Dec. 2014.

[27] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, "Image melding: Combining inconsistent images using patch-based synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 82:1–82:10, Jul. 2012.

[28] N. Kawai, T. Sato, and N. Yokoya, "Diminished reality based on image inpainting considering backgroud geometry," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 3, pp. 1236–1247, Mar. 2016.

[29] J. Herling and W. Broll, "High-quality real-time video inpainting with PixMix," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 6, pp. 866–879, Jun. 2014.

[30] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, "A variaitonal framework for exemplar-based image inpainting," *Int. J. Comput. Vis.*, vol. 93, no. 3, pp. 319–347, Jul. 2011.

[31] V. Fedorov, G. Facciolo, and P. Arias, "Variational framework for non-local inpainting," *Image Process. Line*, vol. 5, pp. 362–386, Dec. 2015.

[32] M. Elad, J. L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmonic Anal.*, vol. 19, no. 3, pp. 340–358, Nov. 2005.

[33] L. He and Y. Wang, "Iterative support detection-based split Bregman method for wavelet frame-based image inpainting," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5470–5485, Dec. 2014.

[34] M. Zhou, et al., "Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[35] X. Li, "Image recovery via hybrid sparse representation: A deterministic annealing approach," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 953–962, Sep. 2011.

[36] F. Li and T. Zeng, "A universal variational framework for sparsity-based image inpainting," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4242–4254, Oct. 2014.

[37] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[38] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1153–1165, May 2010.

[39] X. Liang, X. Ren, Z. Zhang, and Y. Ma, "Texture repairing by unified low rank optimization," *J. Comput. Sci. Technol.*, vol. 31, no. 3, pp. 525–546, May 2016.

[40] X. Liang, "Removing mixed noise in low rank textures by convex optimization," *Comput. Visual Media*, vol. 2, no. 3, pp. 267–276, Sep. 2016.

[41] Y. Zhang, Y. Liu, X. Li, and C. Zhang, "Salt and pepper noise removal in surveillance video based on low-rank matrix recovery," *Comput. Visual Media*, vol. 1, no. 1, pp. 59–68, Mar. 2015.

[42] L. Zhu, C. W. Fu, Y. Jin, M. Wei, J. Qin, and P. A. Heng, "Non-local sparse and low-rank regularization for structure-preserving image smoothing," *Comput. Graph. Forum*, vol. 35, no. 7, pp. 217–226, Oct. 2016.

[43] Z. Zhang, X. Liang, A. Ganesh, and Y. Ma, "TILT: Transform invariant low-rank textures," in *Proc. 10th Asian Conf. Comput. Vis.*, 2011, pp. 314–328.

[44] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[45] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Philadelphia, PA, USA: SIAM, 1998.

[46] G. W. Stewart, *Matrix Algorithm: Basic Decompositions*, vol. 1. Philadelphia, PA, USA: SIAM, 1998.

[47] S. K. Yadav, R. Sinha, and P. K. Bora, "An efficient SVD shrinkage for rank estimation," *IEEE Signal Process. Lett.*, vol. 22, no. 12, pp. 2406–2410, Dec. 2015.
[48] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, "An efficient SVD-based method for image denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 868–880, May 2016.
[49] C. A. Deledalle, V. Duval, and J. Salmon, "Non-local methods with shape-adaptive patches (NLM-SAP)," *J. Math. Imag. Vis.*, vol. 43, no. 2, pp. 103–120, Jun. 2012.
[50] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.
[51] P. C. Hansen, "The truncated SVD as a method for regularization," *BIT Numerical Math.*, vol. 27, no. 4, pp. 534–553, Dec. 1987.
[52] H. Schaeffer and S. Osher, "A low patch-rank interpretation of texture," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 226–262, Feb. 2013.
[53] H. D. Simon and H. Zha, "Low-rank matrix approximation using the Lanczos bidiagonalization process with applications," *SIAM J. Sci. Comput.*, vol. 21, no. 6, pp. 2257–2274, Jun. 2006.
[54] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PathMatch correspondence algorithm," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 29–43.
[55] K. He and J. Sun, "Computing nearest-neighbor fields via propagation-assisted KD-trees," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 111–118.
[56] Z. Zhu, H. Z. Huang, Z. P. Tan, K. Xu, and S. M. Hu, "Faithful completion of images of Scenic Landmarks using internet images," *IEEE Trans. Vis. Comput. Graph.*, vol. 22, no. 8, pp. 1945–1958, Aug. 2016.

Qiang Guo received the BS degree from Shandong University of Technology, Zibo, China, in 2002, the MS and PhD degrees from Shanghai University, Shanghai, China, in 2005 and 2010, respectively. From 2012 to 2015, he was a post-doctoral fellow with Shandong University, Jinan, China. He is currently an associate professor in the School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China. His research interests include image restoration, sparse representation, and object detection. He is a member of the Asia Graphics.

Shanshan Gao received the BS degree from Shandong University of Technology, Zibo, China, in 2001, the MS and PhD degrees from Shandong University, Jinan, China, in 2005 and 2011, respectively. She is currently an associate professor in the School of Computer Science and Technology, Shandong University of Financeand Economics, Jinan, China. Her research interests include computer graphics, image saliency detection, and image segmentation.

Xiaofeng Zhang received the BS and MS degrees from Lanzhou University of Technology, Lanzhou, China, in 2000 and 2005, respectively, and the PhD degree in computer science from Shandong University, Jinan, China, in 2014. He is currently an associate professor in the School of Information and Electrical Engineering, Ludong University, Yantai, China. His research interests include image segmentation and pattern recognition.

Yilong Yin received the PhD degree from Jilin University, Changchun, China, in 2000. From 2000 to 2002, he was a post-doctoral fellow in the Department of Electronic Science and Engineering, Nanjing University, Nanjing, China. He is currently the director of the Machine Learning and Applications Group and a professor with Shandong University, Jinan, China. His research interests include machine learning, data mining, and computational medicine.

Caiming Zhang received the BS and MS degrees from Shandong University, Jinan, China, in 1982 and 1984, respectively, and the PhD degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1994. From 1998 to 1999, he was a post-doctoral fellow with the University of Kentucky, Lexington. He is currently a professor with Shandong University, and a distinguished professor with Shandong University of Finance and Economics. His research interests include computer aided geometric design, computer graphics, information visualization, and medical image processing.