Signal Processing: *Image Communication* 27 (2012) 1109–1122

Contents lists available at SciVerse ScienceDirect

Signal Processing: *Image Communication*

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

Image reconstruction with locally adaptive sparsity and nonlocal robust regularization

Weisheng Dong [a,*], Guangming Shi [a], Xin Li [b], Lei Zhang [c], Xiaolin Wu [d]

[a] *Key Laboratory of Intelligent Perception and Image Understanding (Chinese Ministry of Education), School of Electronic Engineering, Xidian University, Xi'an, China*

[b] *Lane Dept. of Comp. Sci. and Elec. Engr., West Virginia University, Morgantown, WV 26506, USA*

[c] *Dept. of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

[d] *Dept. of Electrical and Computer Engineering, McMaster University, ON, Canada*

ARTICLE INFO

Article history:
Received 30 September 2011
Accepted 9 September 2012
Available online 16 September 2012

Keywords:
Sparse representation
Local dictionary learning
Nonlocal regularization
Image reconstruction

ABSTRACT

Sparse representation based modeling has been successfully used in many image-related inverse problems such as deblurring, super-resolution and compressive sensing. The heart of sparse representations lies on how to find a space (spanned by a dictionary of atoms) where the local image patch exhibits high sparsity and how to determine the image local sparsity. To identify the locally varying sparsity, it is necessary to locally adapt the dictionary learning process and the sparsity-regularization parameters. However, spatial adaptation alone runs into the risk of over-fitting the data because variation and invariance are two sides of the same coin. In this work, we propose two sets of complementary ideas for regularizing image reconstruction process: (1) the sparsity regularization parameters are locally estimated for each coefficient and updated along with adaptive learning of PCA-based dictionaries; (2) a nonlocal self-similarity constraint is introduced into the overall cost functional to improve the robustness of the model. An efficient alternative minimization algorithm is present to solve the proposed objective function and then an effective image reconstruction algorithm is presented. The experimental results on image deblurring, super-resolution and compressive sensing demonstrate that the proposed image reconstruct method outperforms many existing image reconstruction methods in both PSNR and visual quality assessment.

1. Introduction

Reconstructing a high-quality image from a degraded measurement is a typical inverse problem, which has been extensively studied in the past decades [1–3,5,7,13,22]. The image reconstruction (IR) problem is generally modeled by a linear system [1]: $\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{v}$, where $\boldsymbol{x}\in\Re^N$ is the unknown high-quality image, $\boldsymbol{H}\in\Re^{P\times N}$ is the degradation matrix, $\boldsymbol{v}\in\Re^N$ is the additive noise, and $\boldsymbol{y}\in\Re^P$ is the observed measurement. When $\boldsymbol{H}$ is specified by a blurring kernel ($P=N$), then the IR problem is image deblurring; when $\boldsymbol{H}$ is a composite operator of blurring and regular sampling ($P<N$), then the IR problem is single image super-resolution; when $\boldsymbol{H}$ is a random sampling matrix ($P<N$), then the IR problem becomes compressive sensing reconstruction.

The challenge of recovering $\boldsymbol{x}$ from $\boldsymbol{y}$ is due to its ill-posed nature of the inverse problem. To make the problem solvable, additional prior knowledge of the image $\boldsymbol{x}$ has to be incorporated into the reconstruction process. The prior knowledge is often incorporated by a regularization term $J(\boldsymbol{x})$, leading to the following energy minimization problem:

$$\hat{\boldsymbol{x}}=\arg\min_{x}\left\{\|\mathbf{y}-\mathbf{H}\boldsymbol{x}\|_2^2+\lambda\cdot J(\boldsymbol{x})\right\}, \tag{1}$$

where λ is the Lagrangian multiplier balancing the tradeoff between the "prior term" $J(\boldsymbol{x})$ and "likelihood term" $\|\mathbf{y}-\mathbf{H}\boldsymbol{x}\|_2^2$. The "prior term" $J(\boldsymbol{x})$ plays a key role in the

* Corresponding author.
E-mail address: wsdong@mail.xidian.edu.cn (W. Dong).

0923-5965/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.image.2012.09.003

reconstruction process and has been extensively studied in the past decades. The classic "smoothing" regularization terms, such as the quadratic Tikhonov regularization [2] and the total-variation (TV) regularization [3,4], are widely used. However, these prior terms are known to cause over-smoothed results. Other improved regularization terms, including adaptive TV regularization [5–7], nonlocal TV regularization [8,9], have also been proposed.

An alternative approach to incorporating the prior knowledge about images is via *sparse representation*, which has led to a highly competent class of sparsity-based image reconstruction methods for various applications, such as denoising [10,11], compressive sensing [12–14], deblurring [15,16], and super-resolution [17–19]. Sparsity-based IR methods assume that the natural image is *sparse* in some domain spanned by a set of bases or a dictionary of atoms. Here the sparsity means that the signal $\boldsymbol{x}$ can be well approximated by a small set of selected atoms $\{\boldsymbol{d}_j\}_{j\in\Lambda}$ from the dictionary $\boldsymbol{D}\in\Re^{N\times M}(M\geq N)$, i.e., $x\approx\sum_{j\in\Lambda}\alpha_j\boldsymbol{d}_j$. Thus, the recovery of $\boldsymbol{x}$ can be casted as an estimation problem of sparse representation using the observed $\boldsymbol{y}$ over a specific $\boldsymbol{D}$, which amounts to minimize the following cost functional:

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{y}-\mathbf{H}\mathbf{D}\boldsymbol{\alpha}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_0\right\}, \tag{2}$$

where the l_0-norm denotes the number of nonzero entries in vector $\boldsymbol{\alpha}\in\Re^M$. The problem of l_0-norm optimization is a notoriously difficult (NP-hard) problem in combinatorics, and it is often suboptimally solved by greedy algorithms, e.g., matching pursuit [20]. Alternatively, the l_0-norm can be replaced by the l_1-norm, which leads to a convex minimization problem that can be efficiently solved by recently-developed large-scale solvers, such as the basis pursuit [21], iterative shrinkage algorithms [22,23], and the split Bregman algorithm [8].

At the heart of sparse representations lies the following tantalizing question: How should we choose the dictionary $\boldsymbol{D}$? In other words, which is the best domain for sparsifying a given signal $\boldsymbol{x}$? The construction and comparison of various dictionaries for sparsifying natural images has been extensively studied, ranging from the analytically-designed dictionaries (e.g., discrete-cosine-transform, wavelet, curvelet [24], contourlet [25], bandelet [26]) to the learning-based dictionaries [27–29]. The former class attempts to approximate the patches in natural images using basis functions with good localization properties in the Hilbert space. However, these dictionaries are often insufficient to characterize the formidable complexity of natural images, resulting in suboptimal sparse representations. The later class directly learns the image structures from a training dataset and has shown impressive performance improvement over the former on various image restoration tasks [10,11,15,17,18,29].

It should be noted that the learned dictionary $\boldsymbol{D}\in\Re^{N\times M}$ consists of various prototype signal structures, and is *highly redundant*, i.e., $M\gg N$. For a K-sparse signal $\boldsymbol{x}\in\Re^N$ $(N\gg K)$, the total number of sparse representations for $\boldsymbol{x}$ over $\boldsymbol{D}$ amounts to C_M^K. Hence, the sparse representation over the redundant dictionary is potentially unstable—e.g., greedy algorithms might not find the sparsest solution but slightly inferior ones (due to the nonconvexity of l_0-norm optimization). Indeed, some attempts have been made to improve the stability of the sparse approximation by mixing a collection of competitive sparse representations [30], which is equivalent to letting $M\to\infty$. In addition to the prohibitive complexity, we argue that the fundamental problem with this line of reasoning lies on the fact that linear tools such as subspace projections are inefficient in exploiting the nonlinear image manifold.

A more promising approach toward learning the manifold structure of natural images is to combine redundancy with adaptation. Considering the fact that the image content can vary significantly even within a single image, it is important to have dictionaries locally adapted for each patch (in addition to the redundancy introduced by overlapping patches). In this paper, we propose to adaptively learn the dictionary for each patch using nonlocal similar patches both *within the same scale* and *across different scales*. The effectiveness of exploiting nonlocal similarity has been validated in the applications of image denoising [31,11,29] and resolution enhancement [32,19]. The local dictionary learning using nonlocal similar patches has been recently addressed in our previous work [19], which can be viewed as the extension of local principle component analysis (PCA) [33] and locally linear embedding [34] from dimensionality reduction to regression analysis. Considering that there is often rich multiscale visual information in natural images, it is desirable to extend the single-scale local dictionaries to multiscale versions. Here for each local patch, several sub-dictionaries associated with different scales are learned using the nonlocal information. These sub-dictionaries are then used to form an overall multiscale dictionary that will be used for sparse coding.

It has also been shown that varying the sparsity-regularization parameters according to the actual local sparsity can significantly improve the quality of sparsity-based signal reconstruction [35]. In this paper, we formulate the measuring of actual sparsity as a *Maximum a Posterior* (MAP) estimation problem and locally update the estimate of sparsity parameter for each pixel in an iterative fashion. The balance between spatial adaptation and nonlocal regularization is also discussed. The benefit of introducing nonlocal regularization can be understood from inspecting the tradeoff between the *adaptation* and the *robustness* of an image model. Considering similar patches along a regular edge, pixelwise spatial adaptation might cause inconsistency among local patches (i.e., robustness is sacrificed). Nonlocal regularization addresses this issue by introducing a nonlocal means (NLM) based penalty term into the cost functional of image reconstruction solution. As will be seen in Section 4, this NLM-based regularization term can be easily integrated into the proposed adaptive sparse representation framework. In fact, the estimation accuracy of actual sparsity for local patches can be improved by incorporating the coefficients from nonlocal similar patches. In this work, by using a variable splitting and penalty technique we proposed an efficient alternative minimization algorithm to solve the resulted optimization problem. The experimental results of image deblurring, super-resolution and compressive sensing clearly demonstrate that the proposed adaptive sparse representation based IR method significantly outperforms

many existing IR methods in both PSNR values and visual quality.

The rest of this paper is organized as follows. Section 2 presents the proposed adaptive sparse representation with locally computed dictionaries. The local sparsity adaptation and NLM-based nonlocal regularization are studied in Section 3. Section 4 presents the alternative minimization algorithm for solving the resulting optimization problems. Section 5 presents the experimental results, and Section 6 concludes the paper.

2. Sparse representation with locally adaptive dictionaries

In this section, we present an adaptive sparse representation with locally computed dictionaries aiming at improving the flexibility of adapting to the diverse image structures within an image. Conceptually it is related to the idea of local PCA for dimensionality reduction [33], but unlike local PCA, our objective is to obtain a sparse representation and we have to deal with degraded observation data.

2.1. Sparse representation over local dictionaries

Let us consider the image patch of size $\sqrt{n}\times\sqrt{n}$. Denote by $\boldsymbol{x}_i\in\Re^n$, $i=1,2,\ldots N$, the local patch vector extracted at position i, i.e., $\boldsymbol{x}_i=\boldsymbol{R}_i x$, where $\boldsymbol{R}_i\in\Re^{n\times N}$ is a matrix operator that extracts patch $\boldsymbol{x}_i$ from $\boldsymbol{x}$. Suppose that $\boldsymbol{x}_i$ can be sparsely represented over a dictionary $\boldsymbol{D}_i\in\Re^{n\times m}(m\geq n)$, i.e., the solution to

$$\hat{\boldsymbol{\alpha}}_i=\arg\min_{\boldsymbol{\alpha}_i}\left\{\|\boldsymbol{x}_i-\mathbf{D}_i\boldsymbol{\alpha}_i\|_2^2+\lambda\|\boldsymbol{\alpha}_i\|_1\right\} \tag{3}$$

is very sparse, where $\boldsymbol{\alpha}_i\in\Re^m$. Since the local content of an image can vary significantly from region to region, the optimal dictionary for each local patch can also vary across the image. Hence, it is desirable to locally determine the best $\boldsymbol{D}_i$ for each $\boldsymbol{x}_i$. Temporarily, we assume that $\boldsymbol{D}_i$ is known for each $\boldsymbol{x}_i$ (details of deriving them will be elaborated in the next subsection). The whole image $\boldsymbol{x}$ can be sparsely represented by a set of coefficient vectors $\{\hat{\boldsymbol{\alpha}}_i\}$ and the associated locally adaptive dictionaries $\{\boldsymbol{D}_i\}$. Mathematically, this can be written as:

$$\hat{\boldsymbol{x}}=\left(\sum_{i=1}^{N}\mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\sum_{i=1}^{N}(\mathbf{R}_i^T\mathbf{D}_i\hat{\boldsymbol{\alpha}}_i), \tag{4}$$

where the matrix to be inverted is a diagonal one and matrix inversion can be carried out in a pixel-by-pixel manner [10]. It follows from Eq. (4) that the whole image $\boldsymbol{x}$ can be approximated by first sparsely coding each patch $\boldsymbol{x}_i$ and then averaging over all reconstructed patches $\hat{\boldsymbol{x}}_i$. For the convenience of expression, let $\boldsymbol{D}$ denote the concatenation of the local dictionaries $\{\boldsymbol{D}_i\}$ and $\hat{\boldsymbol{\alpha}}$ the concatenation of $\{\hat{\boldsymbol{\alpha}}_i\}$. Then the formula of image reconstruction from sparse representation in Eq. (4) can be rewritten as follows:

$$\hat{\boldsymbol{x}}=\mathbf{D}\circ\hat{\boldsymbol{\alpha}}=\left(\sum_{i=1}^{N}\mathbf{R}_i^T\mathbf{R}_i\right)^{-1}\sum_{i=1}^{N}(\mathbf{R}_i^T\mathbf{D}_i\hat{\boldsymbol{\alpha}}_i). \tag{5}$$

Now, consider the degraded version of $\boldsymbol{x}$: $\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\upsilon$. We are interested in recovering the set of sparse codes $\{\hat{\boldsymbol{\alpha}}_i\}$ from the observation $\boldsymbol{y}$. This recovery problem can be formulated as follows:

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{y}-\mathbf{H}\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\lambda\sum_{i=1}^{N}\|\boldsymbol{\alpha}_i\|_1\right\}, \tag{6}$$

where λ is the regularization parameter. Then $\boldsymbol{x}$ can be recovered by $\hat{\boldsymbol{x}}=\mathbf{D}\circ\hat{\boldsymbol{\alpha}}$. When compared with universal dictionary learning [27], we can achieve a sparser representation by adapting the dictionary $\boldsymbol{D}_i$ to each local patch. In next subsection, we will show how to adaptively learn the local dictionaries $\{\boldsymbol{D}_i\}$ by exploiting the nonlocal redundancies among patches of both the same scale and across different scales.

2.2. Adaptive dictionary learning by iterative PCA

In our previous works [36,37], principal component analysis (PCA) was successfully used to de-correlate the local image structure for noise removal. More specifically, a collection of patches similar to the current one are grouped and the PCA transformation matrix is computed over a set of training data. Since the local patch can be well approximated by a small number of significant principal components, the PCA transformation (also-called Karhunen–Loeve Transform) matrix naturally defines a dictionary for the given image structure. Here we propose to extend this idea of PCA-based computation of locally adaptive dictionaries into image deblurring, super-resolution and compressive sensing recovery.

We will first consider the problem of learning a local dictionary for image $\boldsymbol{x}$ (temporarily ignoring the interference from noise, blur, etc). One way to construct such a training set for an image patch $\boldsymbol{x}_i$ is to collect similar patches from many training images. However, in many practical situations training images are simply not available (let alone the selection of training images, which is an open issue on itself). Based on the observation that natural images often contain a rich amount of repetitive structures, we propose to construct the training dataset by searching for similar patches both within the same scale and across different scales. For image $\boldsymbol{x}$, we first generate a set of images $\{\boldsymbol{x}^s\}$ by scaling down $\boldsymbol{x}$ with scale factors 0.8^s, $s=0,1,\ldots,6$. A similar method was also adopted in [32] for image set generation. Then we can search for the similar patches across the image set $\{\boldsymbol{x}^s\}$ for each $\boldsymbol{x}_i$. All image patches (mean value removed) whose Euclid distances to $\boldsymbol{x}_i$ are smaller than a threshold are selected as similar patches:

$$S_i=\{\boldsymbol{x}_j^s\,|\,\|\overline{\boldsymbol{x}}_i-\overline{\boldsymbol{x}}_j^s\|\leq t, s=0,\ldots,6\}, \tag{7}$$

where $\overline{\boldsymbol{x}}_i$ and $\overline{\boldsymbol{x}}_j^s$ are the mean-removed versions of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j^s$, respectively, and $\boldsymbol{x}_j^s$ is the patch extracted from $\boldsymbol{x}^s$, and t is the pre-defined threshold. Alternatively, we can select those image patches $\boldsymbol{x}_j^s$ such that $\overline{\boldsymbol{x}}_j^s$ is within the Jth closest patches to $\boldsymbol{x}_i$. By concatenating similar patches together, we can obtain a 2D data matrix for $\boldsymbol{x}_i$: $\boldsymbol{x}_i=[\boldsymbol{x}_i,\boldsymbol{x}_{i,1},\ldots,\boldsymbol{x}_{i,J-1}]$, where each column is a similar patch vector extracted from $\{\boldsymbol{x}^s\}$. Then standard PCA computation can be applied to the covariance of $\boldsymbol{X}_i$ given by: ($\overline{\boldsymbol{X}}_i$ is

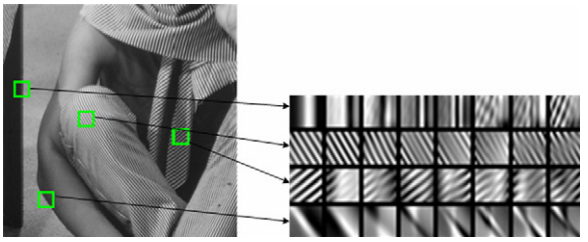

Fig. 1. Examples of the learned local PCA dictionaries. The left image shows the selected local patches of part of the image *Barbara*, and the right image shows the corresponding learned PCA dictionaries. Note that only the first 8 atoms of the learned PCA dictionaries are shown.

the mean-removed version of $\boldsymbol{X}_i$)

$$\boldsymbol{\Omega}_i = \frac{1}{J}\overline{\boldsymbol{X}}_i\overline{\boldsymbol{X}}_i^T. \tag{8}$$

Some learned local PCA dictionaries corresponding to the local patches of image *Barbara* are shown in Fig. 1. As shown in Fig. 1, we can see that the local PCA dictionaries can well represent the local image patterns.

Now the question is how to train the dictionary when we only have access to the degraded version of $\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\upsilon$? The peer relationship between $\boldsymbol{x}$ and $\boldsymbol{D}$ (in fact it is the sparse coefficient $\hat{\boldsymbol{\alpha}}$) suggests that we can treat the dictionary as a latent variable just like the centroids in k-means clustering. Such a problem with chicken-and-egg flavor calls for an iterative solution in which we alternate our estimates of $\boldsymbol{x}$ and $\boldsymbol{D}$. Specifically, we propose to first obtain an initial estimate of $\boldsymbol{x}$ using a fixed dictionary (e.g., DCT or wavelet), denoted by $\hat{\boldsymbol{x}}$, and calculate the locally adaptive dictionaries $\{\boldsymbol{D}_i\}$ using $\hat{\boldsymbol{x}}$. With the updated dictionaries $\{\boldsymbol{D}_i\}$, we can achieve better IR results by Eqs. (5) and (6), which provide an updated estimate for $\boldsymbol{x}$. Then the procedure of updating $\{\boldsymbol{D}_i\}$ and $\boldsymbol{x}$ can be iterated. Such a successive refinement of local dictionaries and reconstructed image shares conceptual similarity to many iterative estimation techniques such as the expectation-maximization (EM) algorithm.

2.3. Multiscale adaptive dictionaries

By adapting the dictionaries to the local image structure, the sparsity-based IR method can lead to better IR results than most existing methods. Nevertheless, the single-scale dictionaries are not able to effectively exploit the multiscale information that is commonly contained in natural images. For instance, dictionaries of small atom size (e.g., 5×5) are suitable for representing fine edge structures but are unable to accurately characterize large-scale image structures or flat image regions, and vice visa. To address this issue, we extend the single-scale adaptive dictionary learning to multi-scale versions using a quad-tree structure model. As a common tool, the quad-tree structure model is widely used in image/video coding as well as in image representation (e.g., wedgelet [38] and multiscale K-SVD [28]).

The basic idea of the proposed multi-scale adaptive sparse representation is to partition a large root patch of size $\sqrt{n}\times\sqrt{n}$ into four $\sqrt{n}/2\times\sqrt{n}/2$ sub-patches, each of which can further be partitioned to generate more scales, as shown in Fig. 2. Supposing that L different scales are generated, for scale l there are totally 4^{l-1}, $l=1,2,\ldots,L$ sub-patches of size $\sqrt{n}/2^{l-1}\times\sqrt{n}/2^{l-1}$. To jointly exploit the structural information among the different scales, for each patch $\boldsymbol{x}_i$ we propose to learn a joint dictionary $\tilde{\boldsymbol{D}}_i$ that consists of totally $\sum_{l=1}^{L}4^{l-1}$ sub-dictionaries corresponding to each sub-patch. We use the above described iterative PCA approach to learn the sub-dictionaries to exploit the multi-scale nonlocal redundancies. Since those sub-dictionaries are of different sizes, we use the zero padding to embed the atoms into bigger atoms of size n, resulting in the multiscale dictionary $\tilde{\boldsymbol{D}}_i=[\boldsymbol{D}_{i,0},\boldsymbol{D}_{i,l_1},\ldots,\boldsymbol{D}_{i,L_q}]\in\Re^{n\times nL}$, where $q=1,2,\ldots,4^{l-1}$. Similar consideration was also adopted in [28].

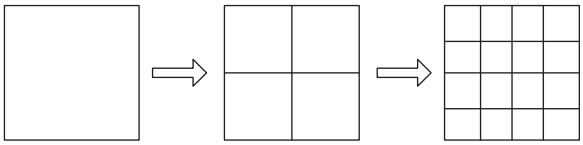

Fig. 2. Patch partition with quad-tree structure model for the multi-scale representation.

For each local patch $\boldsymbol{x}_i$ (called root patch), we solve the problem of Eq. (3) with $\tilde{\boldsymbol{D}}_i$ using the iterative shrinkage algorithm [22]. Since each local patch $\boldsymbol{x}_i$ is coded with multi-scale dictionaries, it will be reconstructed by averaging the results from multi-scales. And the whole image will be reconstructed by averaging all the reconstructed root patches using Eq. (5).

The objective function of Eq. (6) can be solved by the iterative shrinkage (IS) algorithm [22]. However, the major drawback of the IS algorithm is its low convergence speed. In this paper we present a variable-splitting based minimization algorithm [46] to solve Eq. (6), which allows faster convergence speed, and thus can significantly reduce the computational complexity. Specifically, we introduce an auxiliary variable $\boldsymbol{z}$ for the reconstructed image $\boldsymbol{x}$ into Eq. (6) and obtain the following equation

$$(\hat{\boldsymbol{\alpha}},\hat{\boldsymbol{z}})=\arg\min_{\alpha,z}\left\{\|\mathbf{y}-\mathbf{H}\mathbf{z}\|_2^2+\gamma\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\lambda\sum_{i=1}^{N}\|\boldsymbol{\alpha}_i\|_1\right\}, \tag{9}$$

where γ is a penalty parameter. It can be seen that when γ becomes large enough, Eq. (9) is equal to Eq. (6). Eq. (9) can be solved by alternatively solving the following two sub-problems, i.e.,

$$\hat{\mathbf{z}}=\arg\min_{\mathbf{z}}\left\{\|\mathbf{y}-\mathbf{H}\mathbf{z}\|_2^2+\gamma\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2\right\}, \tag{10}$$

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\lambda/\gamma\sum_{i=1}^{N}\|\boldsymbol{\alpha}_i\|_1\right\}. \tag{11}$$

Note that both of the two sub-problems have closed-form solutions and thus can be solved efficiently. Eq. (10) is a quadratic minimization problem and thus can be solved in a closed-form, i.e.,

$$\mathbf{z}=\frac{\gamma\mathbf{D}\circ\boldsymbol{\alpha}+\mathbf{H}^T\mathbf{y}}{\mathbf{H}^T\mathbf{H}+\gamma\mathbf{I}}. \tag{12}$$

where $\boldsymbol{I}$ is an identity matrix. In our implementation, we

use the conjugate gradients (CG) algorithm to compute Eq. (12) for an estimated $\boldsymbol{z}$. For a fixed $\boldsymbol{z}$, the optimization of the sparse code $\boldsymbol{\alpha}$ by solving Eq. (11) is a typical denoising problem that can be solved by soft thresholding. The overall algorithm for solving Eq. (6) is outlined in **Algorithm 1**.

Algorithm 1.

1. Initialization:
   (a). Initializing the local dictionaries $\{\boldsymbol{D}_i\}$ as DCT basis, we can compute an initial estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ by solving Eq. (6) using the IS algorithm [22];
   (b). With the initial estimate $\hat{\boldsymbol{x}}$, we locally compute the multi-scale $\{\tilde{\boldsymbol{D}}_i\}$;
   (c). Set the initial sparse codes $\boldsymbol{\alpha}^{(0)}=0$, $k=0$, and preset a small constant Δ, the maximal iteration number, denoted by *Max_Iter*, and T as an integer;
2. Iterate on k until $\|\hat{\boldsymbol{x}}^{(k)}-\hat{\boldsymbol{x}}^{(k+1)}\|_2^2/N \le \Delta$ or $k \ge Max\_Iter$ is satisfied.
   (a). Solve for $\hat{\boldsymbol{z}}^{(k+1)}$ for a given set of vectors $\boldsymbol{\alpha}^{(k)}$ by solving Eq. (12) using a CG algorithm;
   (b). Compute the PCA transformation coefficients $\boldsymbol{\alpha}^{(k+1/2)}=[\tilde{\boldsymbol{D}}_1^T\boldsymbol{R}_1\hat{\boldsymbol{z}}^{(k+1)},\ldots,\tilde{\boldsymbol{D}}_N^T\boldsymbol{R}_N\hat{\boldsymbol{z}}^{(k+1)}]$;
   (c). Threshold each coefficient $\alpha_{i,j}^{(k+1)}=\mathrm{soft}(\alpha_{i,j}^{(k+1/2)},\ \tau)$, where $\mathrm{soft}(\cdot,\tau)$ is a soft thresholding function with threshold τ;
   (d). Compute $\hat{\boldsymbol{x}}^{(k+1)}=\boldsymbol{D}\circ\boldsymbol{\alpha}^{(k+1)}$ using Eq. (5), which can be calculated by first reconstructing each root patch with $\hat{\boldsymbol{x}}_i^{(k+1)}=\tilde{\boldsymbol{D}}_i\boldsymbol{\alpha}_i^{(k+1)}$ and then averaging all the reconstructed image patches;
   (e). If $mod(k, T)=0$, update the adaptive sparse domain $\{\tilde{\boldsymbol{D}}_i\}$ using the improved $\hat{\boldsymbol{x}}^{(k+1)}$.

In **Algorithm 1**, Δ is a pre-specified constant controlling the convergence of the iterative process, and *Max_Iter* is the allowed maximum number of iterations. We only update the adaptive sparse domain $\{\boldsymbol{D}_i\}$ in every T iterations to save computational complexity. The threshold τ in step 2 (c) is given by $\tau=c\lambda/\gamma$, where c is a pre-selected scalar parameter. As will be shown in the next section, by adapting the parameter λ in Eq. (6) to each coefficient, we can further improve the performance of **Algorithm 1**.

3. Local sparsity adaptation and nonlocal regularization

In this section, we extend **Algorithm 1** by incorporating two complementary ideas: spatial adaptation and nonlocal regularization. Spatial adaptation is to adjust the local sparsity to better match the spatially-varying structures (e.g., from smooth regions to edges or texture regions). Such an objective can be met by locally varying the local dictionaries and the regularization parameter λ in Eq. (6). However, the risk of doing spatial adaptation on a pixel-by-pixel basis is the lack of consistency across patches of similar structures. Therefore, nonlocal regularization is introduced as an effective countermeasure to improve the *robustness* of adaptive sparse representations.

3.1. Locally adaptive estimation of sparsity regularization parameter

Eq. (6) employs a global regularization parameter λ for all patches. It has been shown that better image reconstruction results can be obtained by varying the sparsity-regularization parameters [35]. Assume that the observation $\boldsymbol{y}$ is contaminated with additive Gaussian noise with standard deviation σ_n. Under the Bayesian framework, the estimation of sparsity vector $\hat{\boldsymbol{\alpha}}$ can be modeled as the following MAP problem:

$$\hat{\boldsymbol{\alpha}}=\arg\max_{\boldsymbol{\alpha}}\{\log P(\boldsymbol{\alpha}|\mathbf{y})\}=\arg\min_{\boldsymbol{\alpha}}\{-\log P(\mathbf{y}|\boldsymbol{\alpha})-\log P(\boldsymbol{\alpha})\}, \tag{13}$$

where

$$P(\mathbf{y}|\boldsymbol{\alpha})=\frac{1}{\sigma_n\sqrt{2\pi}}\exp-\left(\frac{1}{2\sigma_n^2}\|\mathbf{y}-\mathbf{H}\mathbf{D}\boldsymbol{\alpha}\|_2^2\right). \tag{14}$$

The prior distribution of $\boldsymbol{\alpha}$ is often characterized by an independent and identically distributed (i.i.d.) zero-mean Laplacian probability model:

$$P(\boldsymbol{\alpha})=\prod_{i=1}^{N}\prod_{j=1}^{n}\frac{1}{\sqrt{2}\sigma}\exp\left(-\frac{1}{\sigma}\sqrt{2}|\alpha_{i,j}|\right), \tag{15}$$

where σ is the standard deviation of the Laplacian model. By plugging $P(\mathbf{y}|\boldsymbol{\alpha})$ and $P(\boldsymbol{\alpha})$ into Eq. (13), we can obtain the l_1-norm sparse representation model of Eq. (6) with $\lambda=2\sqrt{2}\sigma_n^2/\sigma$.

By varying the parameter σ according to the local statistics, the Laplacian model becomes more accurate for characterizing the distribution of $\boldsymbol{\alpha}$. Therefore, we could improve the sparsity-based image model by tuning λ for each element of $\boldsymbol{\alpha}$, i.e.,

$$\hat{\boldsymbol{\alpha}}=\arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{y}-\mathbf{H}\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\sum_{i=1}^{N}\sum_{j=1}^{n}\lambda_{i,j}|\alpha_{i,j}|\right\}, \tag{16}$$

where $\alpha_{i,j}$ is the jth coefficient associated with the jth atom of $\tilde{\boldsymbol{D}}_i$, $\lambda_{i,j}=2\sqrt{2}\sigma_n^2/\sigma_{i,j}$ and $\sigma_{i,j}$ is the standard deviation of $\alpha_{i,j}$. For numerical stability, we let:

$$\lambda_{i,j}=\frac{2\sqrt{2}\sigma_n^2}{\hat{\sigma}_{i,j}+\varepsilon}, \tag{17}$$

where ε is a constant ($\varepsilon=0.1$ in our implementation) and $\hat{\sigma}_{i,j}$ is the estimate of $\sigma_{i,j}$. To estimate $\sigma_{i,j}$, we can first obtain an initial estimate of the image $\boldsymbol{x}$ by solving Eq. (6), denoted by $\hat{\boldsymbol{x}}$, and then estimate $\sigma_{i,j}$ using the PCA transformed dataset $\hat{\mathbf{Y}}_i=\tilde{\mathbf{D}}_i^T\hat{\boldsymbol{x}}_i=[\hat{\boldsymbol{\alpha}}_i,\hat{\boldsymbol{\alpha}}_{i,1},\ldots,\hat{\boldsymbol{\alpha}}_{i,J}]$, where $\hat{\boldsymbol{x}}_i$ is the training data set extracted from $\hat{\boldsymbol{x}}$ using the method described in Section 2. In the scenario of image restoration, σ_n^2 is often assumed to be known, or it can be estimated.

Such a local regularization parameter adjustment strategy has been used in [35] by letting $\lambda_{i,j}=1/(|\hat{\alpha}_{i,j}|+\varepsilon)$. Here, by using the coefficients $\hat{\alpha}_{i,j}$ from nonlocal similar patches, $\lambda_{i,j}$ can be more accurately estimated. On the other hand, the nonlocal redundancy can be used to improve the robustness of adaptive sparse representation in Eq. (16) by using the strategy of nonlocal regularization, which is introduced next.

3.2. Enforcing self-similarity constraint via nonlocal regularization

Local adaptation of dictionaries and local estimation of regularization parameter $\lambda_{i,j}$ both contribute to improving the characterization of spatially varying structures in natural images. However, spatial adaption alone runs into the risk of overfitting the data. Considering similar patches along an edge, the pixel-wise spatial adaption may cause non-consistence among local patches. To address this issue, we further introduce a nonlocal self-similarity constraint into the proposed sparse model. The nonlocal self-similarity constraint enforces that the pixels of the reconstructed images should be well predicted with the nonlocal similar neighbors. Such nonlocal self-similarity constraint has been proved to be effective on preserving edge and texture structures for various tasks, such as denoising [31,11] and super-resolution [32]. In the above PCA-based local dictionary learning process, nonlocal redundancy information has been partially exploited by the training set (i.e., through search of similar patches) but its role has been limited to promoting local adaptation in the previous section. In this subsection, we show how to more effectively exploit the nonlocal redundancy information to improve the robustness of our sparsity-based model.

In PCA-based dictionary learning, we have grouped a set of similar patches for each patch $\boldsymbol{x}_i$. Suppose that there is a set of L patches, $\{\hat{\boldsymbol{x}}_l\}$, $l=1,2,\ldots,L$, from the same scale that are within the Lth ($L=10$ in our experiments) closest patches to $\hat{\boldsymbol{x}}_i$, where $\hat{\boldsymbol{x}}_i$ and $\hat{\boldsymbol{x}}_l$ are the current estimates of $\boldsymbol{x}_i$ and $\boldsymbol{x}_l$, respectively. Let $\boldsymbol{x}(i)$ be the central pixel of patch $\boldsymbol{x}_i$ (i.e., the pixel located at position i) and $\boldsymbol{x}(l)$ be the central pixel of patch $\boldsymbol{x}_l$. Then a prediction of $\boldsymbol{x}(i)$ can be formed as $\sum_{l=1}^{L} b_{i,l}\boldsymbol{x}(l)$. The weights assigned to $\boldsymbol{x}(l)$ are set as

$$b_{i,l} = \exp\left(-\left\|\hat{\boldsymbol{x}}_i-\hat{\boldsymbol{x}}_l\right\|/h\right)/c_i, \tag{18}$$

where h is a controlling factor of the weight and c_i is the normalization factor. Considering the nonlocal self-similarity property, we expect that the prediction error $\left\|\boldsymbol{x}(i)-\sum_{l=1}^{L} b_{i,l}\boldsymbol{x}(l)\right\|_2^2$ should be small. Therefore, it is natural to introduce the following energy function:

$$f_{NLM}(x) = \sum_{i=1}^{N}\left\|\boldsymbol{x}(i)-\sum_{l=1}^{L} b_{i,l}\boldsymbol{x}(l)\right\|_2^2. \tag{19}$$

The above formula can be more compactly written in a matrix form:

$$f_{NLM}(\boldsymbol{x}) = \|(\mathbf{I}-\mathbf{B})\boldsymbol{x}\|_2^2,, \tag{20}$$

where $\boldsymbol{B}\in\Re^{N\times N}$ and $\boldsymbol{I}$ is the identity matrix and

$$\boldsymbol{B}(i,j) = \begin{cases} b_{i,j}, & \text{if } x(j) \text{ is the selected similar pixel to } x(i) \\ 0, & \text{otherwise} \end{cases}, \tag{21}$$

where $b_{i,j}$ is computed using Eq. (18). By incorporating this nonlocal self-similarity regularization into Eq. (16), we obtain a balanced objective functional as follows:

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{y}-\mathbf{HD}\circ\boldsymbol{\alpha}\|_2^2+\eta\|(\mathbf{I}-\mathbf{B})\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\sum_{i=1}^{N}\sum_{j=1}^{n}\lambda_{i,j}|\alpha_{i,j}|\right\}, \tag{22}$$

where the first l_2-norm term is the likelihood term, guaranteeing that the reconstructed image $\hat{\boldsymbol{x}}=\mathbf{D}\circ\hat{\boldsymbol{\alpha}}$ should fit the observation model $\boldsymbol{y}=\boldsymbol{H}\boldsymbol{x}+\boldsymbol{v}$, the second l_2-norm term is the nonlocal self-similarity regularization term, enforcing that the reconstructed image pixels should be well predicted by the nonlocal similar neighbors, and the third weighted l_1-norm term is the sparsity penalty term, requiring that the reconstructed image should exhibit sparsity in the local sparse domain.

4. Numerical algorithm and implementation

After incorporating the nonlocal similarity regularization term, the minimization of Eq. (22) is still convex and can be solved efficiently using existing l_1-norm minimization algorithms. For high speed convergence, we also present an alternative minimization algorithm for solving Eq. (22) using the variable-splitting and penalty techniques [46]. By introducing an auxiliary variable $\boldsymbol{z}$ for the reconstructed image $\boldsymbol{x}$ into Eq. (22), we obtain the following equation

$$(\hat{\boldsymbol{\alpha}},\hat{\mathbf{z}}) = \arg\min_{\boldsymbol{\alpha},z}\left\{\|\mathbf{y}-\mathbf{Hz}\|_2^2+\eta\|(\mathbf{I}-\mathbf{B})\mathbf{z}\|_2^2+\gamma\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2 + \sum_{i=1}^{N}\sum_{j=1}^{n}\lambda_{i,j}|\alpha_{i,j}|\right\}, \tag{23}$$

where γ is a penalty parameter. Eq. (23) can be solved by alternatively solving the following two sub-problems, i.e.,

$$\hat{\mathbf{z}} = \arg\min_{z}\{\|\mathbf{y}-\mathbf{Hz}\|_2^2+\eta\|(\mathbf{I}-\mathbf{B})\mathbf{z}\|_2^2+\gamma\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2\}, \tag{24}$$

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}}\left\{\|\mathbf{z}-\mathbf{D}\circ\boldsymbol{\alpha}\|_2^2+\frac{1}{\gamma}\sum_{i=1}^{N}\sum_{j=1}^{n}\lambda_{i,j}|\alpha_{i,j}|\right\}. \tag{25}$$

Both of the two sub-problems admit closed-form solutions and thus can be solved with low computational complexity. For a fixed sparse code $\boldsymbol{\alpha}$ and dictionary $\boldsymbol{D}$, Eq. (24) is a quadratic minimization problem and can be solved in a closed-form, i.e.,

$$\mathbf{z} = \frac{\gamma\mathbf{D}\circ\boldsymbol{\alpha}+\mathbf{H}^T\mathbf{y}}{\mathbf{H}^T\mathbf{H}+\eta(\mathbf{I}-\mathbf{B})^T(\mathbf{I}-\mathbf{B})+\gamma\mathbf{I}}. \tag{26}$$

In our implementation, we use the conjugate gradients (CG) algorithm to compute Eq. (26) for an estimated $\boldsymbol{z}$. For a fixed $\boldsymbol{z}$, the optimization of the sparse code $\boldsymbol{\alpha}$ by solving Eq. (25) is a typical denoising problem, i.e., first the image patches were extracted and then sparsely coded with respect to the local dictionary $\boldsymbol{D}_i$ and the sparsity regularization parameter $\lambda_{i,j}/\gamma$. In general, the alternative optimization of $\boldsymbol{z}$ and $\boldsymbol{\alpha}$ will converge to a local minima after several iterations (typically 40–50 iterations in our deblurring experiments). Compared to the IS algorithm that typically converges after 1000 iterations in our deblurring experiments, the alternative minimization

algorithm has a much faster convergence speed. We outline the details of the algorithm for solving Eq. (23) in **Algorithm 2**.

In **Algorithm 2**, the threshold $\tau_{i,j}$ is locally computed as $\tau_{i,j}=c\lambda_{i,j}/\gamma$, where c is introduced to control the denoising effects in each iteration. After we obtain a reconstructed image, $\{\tilde{\mathbf{D}}_i\}$, $\boldsymbol{B}$ and $\lambda_{i,j}$ can be recomputed and then **Algorithm 2** can be iterated to improve the quality of the reconstructed image. Our experiments show that two iterations of **Algorithm 2** are enough for good results.

Algorithm 2.

1. Initialization:
   a. Initializing the local dictionaries $\{\boldsymbol{D}_i\}$ as DCT basis, we can compute an initial estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$ by solving Eq. (6) using the IS algorithm [22];
   b. Using the initial estimate$\hat{\boldsymbol{x}}$, we locally compute $\{\tilde{\boldsymbol{D}}_i\}$, the local sparsity regularization parameters $\lambda_{i,j}$ for each coefficient using Eq. (17), and the non-local weight matrix $\boldsymbol{B}$;
   c. Set the initial sparse codes $\boldsymbol{\alpha}^{(0)}=0$, $k=0$, and preset a small constant Δ, the maximal iteration number, denoted by *Max_Iter*, and T as an integer;

2. Iterate on k until $\|\hat{\boldsymbol{x}}^{(k)}-\hat{\boldsymbol{x}}^{(k+1)}\|_2^2/N\le\Delta$ or $k\ge Max\_Iter$ is satisfied.
   a. Solve for $\hat{\boldsymbol{z}}^{(k+1)}$ for a given set of sparse vector $\boldsymbol{\alpha}^{(k)}$ by solving Eq. (26) using a CG algorithm;
   b. Compute the PCA transformation coefficients $\boldsymbol{\alpha}^{(k+1/2)}=[\tilde{\boldsymbol{D}}_1^T\boldsymbol{R}_1\hat{\boldsymbol{z}}^{(k+1)},\ldots,\tilde{\boldsymbol{D}}_N^T\boldsymbol{R}_N\hat{\boldsymbol{z}}^{(k+1)}]$;
   c. Threshold each coefficient $\alpha_{i,j}^{(k+1)}=\mathrm{soft}(\alpha_{i,j}^{(k+1/2)},\tau_{i,j})$, where $\mathrm{soft}(\cdot,\tau_{i,j})$ is a soft thresholding function with threshold $\tau_{i,j}$;
   d. Compute $\hat{\boldsymbol{x}}^{(k+1)}=\boldsymbol{D}\circ\boldsymbol{\alpha}^{(k+1)}$ using Eq. (5), which can be calculated by first reconstructing each image patch with $\hat{\boldsymbol{x}}_i^{(k+1)}=\tilde{\boldsymbol{D}}_i\boldsymbol{\alpha}_i^{(k+1)}$ and then averaging all the reconstructed image patches;
   e. If $mod(k,T)=0$, update the adaptive sparse domain $\{\tilde{\boldsymbol{D}}_i\}$ and $\boldsymbol{B}$ using the improved $\hat{\boldsymbol{x}}^{(k+1)}$.

It has been widely observed that PCA suffers from high computational complexity especially when it is applied for each extracted patch such as in **Algorithm 2**. To reduce the computational complexity, we propose to only apply PCA to the class of edge and texture patches. A local patch is declared to be an edge or texture patch if its pixel intensity variance is above a threshold T_h ($T_h=16$ in our implementation). For all smooth image patches, we use a DCT dictionary. Since the class of edge and texture patches often consists of only a small fraction of the whole, the computational complexity can be significantly reduced.

5. Experimental results

In this section, we conduct extensive experiments to verify the performance of the proposed adaptive sparse representation (ASR) method for image deblurring, super-resolution and compressive sensing with comparisons to state-of-the-art image restoration methods. In the proposed method, for simplicity the scale number is set to be two and the root patches of size 10×10 (with 3 pixels overlapping between adjacent patches) are extracted and sparsely coded with locally computed multiscale dictionaries. The parameters of **Algorithms 1** and **2** will be discussed separately in each experiment. We empirically found that the performance of the proposed **Algorithms 1** and **2** are non-sensitive to the selection of the λ and γ, though different pairs of λ and γ will lead to slightly different convergence rate of the proposed algorithms. Due to the limitation of space, only part of the experimental results can be shown in the paper.

5.1. Image deblurring

The deblurring performance of the proposed ASR approach is verified on both simulated blurred images and real motion blurred images. The simulated blurred and noisy images are generated by first applying a blur kernel and adding the additive Gaussian noises with different noise levels $\sigma_n=\sqrt{2}$ and 2. Two blur kernels, i.e., 9×9 uniform blur and 2D Gaussian function with standard deviation of 3, are applied for simulations. For real motion blurred images, we borrowed the kernel estimation method from [39] to estimate the blur kernel. The estimated blur kernel is then fed into the ASR approach to deblur the real motion blurred images. For simplicity, we only apply deblurring to the luminance component for color images.

In the experiments of image deblurring, we empirically found that the proposed **Algorithms 1** and **2** usually converged after 50 iterations. Thus we simply set *Max_Iter*$=50$ in the proposed **Algorithms 1** and **2**. In **Algorithm 1**, we empirically set $\lambda=0.022\sigma_n^2$, and $\lambda=0.011\sigma_n^2$ for uniform blur kernel and Gaussian blur kernel, respectively, where σ_n denotes the standard deviation of Gaussian noise. While in **Algorithm 2**, we set the threshold $\tau_{i,j}=0.15\lambda_{i,j}/\gamma$, $\eta=0.012$, and $\tau_{i,j}=0.15\lambda_{i,j}/\gamma$, $\eta=0.012$ for uniform blur and Gaussian blur, respectively, wherein $\lambda_{i,j}$ is computed using Eq. (17). In both **Algorithms 1** and **2**, we empirically set $\gamma=0.02$ and $\gamma=0.007$ for uniform blur kernel and Gaussian blur kernel, respectively.

Fig. 3 shows the deblurred *Parrots* images by the proposed methods. We can see that **Algorithm 1** is very effective in removing the blurring effect and reconstructing the edges and textures. However, artifacts can still be observed around the edges. Through spatial adaption of the sparsity parameters and nonlocal regularization, **Algorithm 2** can much better suppress the artifacts and produce much cleaner and sharper edges than **Algorithm 1**.

We also compared the proposed **Algorithm 2** with several recently developed deblurring methods: the constrained TV deblurring (denoted by FISTA) method [40], and the BM3D method [41] [1], and our recently developed adaptive sparse domain selection method (denoted by ASDS-Reg) [15]. Note that the BM3D is a very effective image restoration method and the ASDS-Reg is the current state-of-the-art image restoration methods.

The PSNR comparison results are listed in Table 1, from which we can see that **Algorithm 2** produces the highest

[1] We thank the authors of [40,41,39,17,44,45] for providing their source codes or executable programs.

Fig. 3. The deblurred *Parrot* image by the proposed methods (9×9 uniform blur with $\sigma_n=\sqrt{2}$). From left to right: original image; blurred and noisy; deblurred by **Algorithm 1**; deblurred by **Algorithm 2**.

Table 1
The PSNR (dB) results of the deblurred images.

Images	9×9 uniform blur, $\sigma_n=\sqrt{2}$							
	Butterfly	*Tower*	*C. Man*	*Starfish*	*Parrots*	*Barbara*	*Leaves*	*Average*
FISTA [40]	28.37	29.83	26.82	27.75	29.11	25.75	26.49	27.73
BM3D [41]	27.21	29.38	27.30	28.61	30.50	27.99	27.45	28.35
ASDS-Reg [15]	28.70	30.48	28.02	29.69	31.22	27.79	28.55	29.21
Algorithm 1	28.63	30.47	27.07	29.66	31.22	27.34	28.51	28.98
Algorithm 2	**29.09**	**30.47**	**27.92**	**30.10**	**31.46**	**28.10**	**29.18**	**29.47**
	9×9 uniform blur, $\sigma_n=2$							
FISTA [40]	27.73	29.55	26.13	27.50	28.88	25.24	26.03	27.29
BM3D [41]	26.56	28.83	26.60	27.96	29.74	**27.26**	26.60	27.65
ASDS-Reg [15]	27.91	29.61	**27.35**	28.77	30.37	27.03	27.55	28.37
Algorithm 1	27.79	29.53	26.48	28.75	30.31	26.74	27.58	28.17
Algorithm 2	**28.37**	**29.67**	27.07	**29.25**	**30.70**	27.17	**28.33**	**28.65**
Images	Gaussian blur, $\sigma_n=\sqrt{2}$							
FISTA [40]	28.31	28.91	25.99	27.84	29.70	24.27	26.82	27.40
BM3D [41]	29.01	29.90	27.46	30.77	32.23	**28.19**	29.69	29.61
ASDS-Reg [15]	29.82	**30.44**	27.28	31.91	32.93	26.91	30.78	29.93
Algorithm 1	29.72	29.72	**27.78**	31.51	32.33	27.37	30.41	29.69
Algorithm 2	**30.13**	30.13	27.42	**31.93**	**32.82**	28.10	**30.93**	**30.21**
	Gaussian blur, $\sigma_n=2$							
FISTA [40]	28.20	28.88	25.92	27.79	29.61	24.20	26.71	27.33
BM3D [41]	28.56	29.52	27.08	30.27	31.73	27.02	29.12	29.04
ASDS-Reg [15]	29.56	**30.20**	26.71	**31.39**	**32.67**	26.09	**30.22**	**29.55**
Algorithm 1	29.27	29.18	26.48	30.99	31.58	26.48	29.75	29.11
Algorithm 2	**29.62**	29.26	**27.16**	31.29	32.09	**26.97**	30.24	29.52

Fig. 4. Image deblurring performance comparison on *Starfish* image (9×9 uniform blur, $\sigma_n=\sqrt{2}$). From left to right: blurred and noisy; FISTA [40] (PSNR=27.75 dB); BM3D [41] (PSNR=28.61 dB); ASDS-Reg [15] (PSNR=29.59 dB); **Algorithm 2** (PSNR=**30.10** dB).

PSNR results in average. The visual comparisons of the deblurred images by the test methods are shown through Figs. 4 and 5, from which we can see that proposed method produces the best visual perception results with clearer and sharper edges and fewer artifacts.

We also test the deblurring approaches on real motion blurred images. Since the blur kernel estimation is a non-trivial task and out of the scope of this paper, we borrowed the kernel estimation method of [39] to estimate the kernel. The estimated blur kernel is then fed into the proposed **Algorithm 2** to deblur the real motion blurred images. As shown in Fig. 6, we can see that the proposed **Algorithm 2** produced much clearer and sharper image than [39].

5.2. Image super-resolution

In image super-resolution, the observed low-resolution (LR) image is generated by first applying a blurring kernel

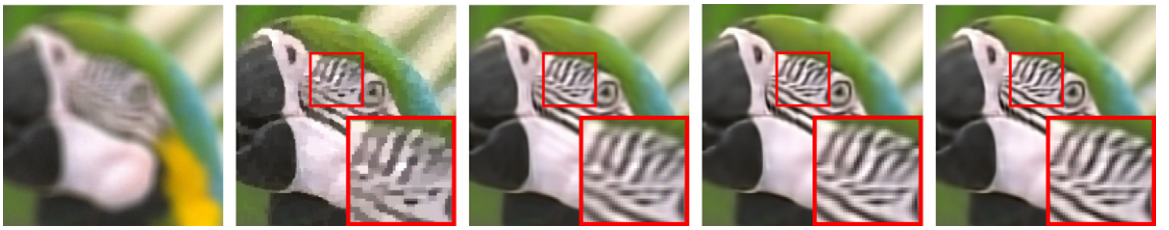

Fig. 5. Image deblurring performance comparison on *Parrots* image (9×9 uniform blur, $\sigma_n=2$). From left to right: blurred and noisy; FISTA [40] (PSNR=28.88 dB); BM3D [41] (PSNR=29.74 dB); ASDS-Reg [15] (PSNR=30.17 dB); **Algorithm 2** (PSNR=**30.70** dB).

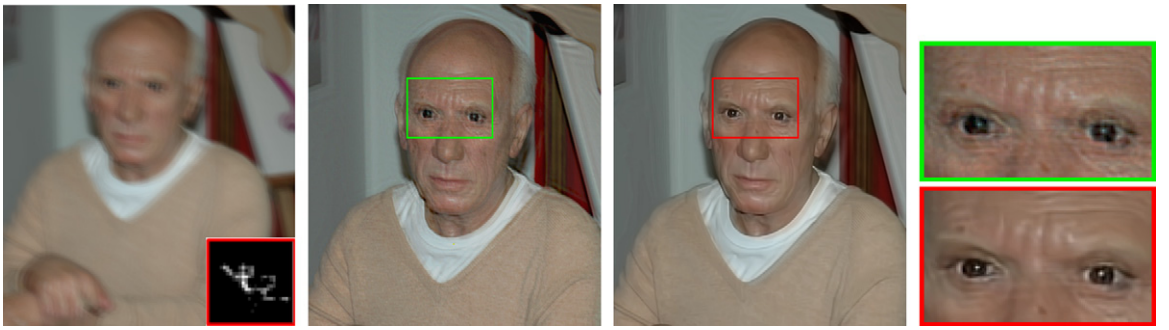

Fig. 6. Deblurring performance comparison for a real blurred image. From left to right: input blurred image and the estimated blur kernel by method in [39]; deblurred image by [39]; the deblurred image by **Algorithm 2** with blurring kernel estimated using [39]; close-up views.

Fig. 7. The reconstructed HR *Butterfly* images by the proposed methods. From left to right: original image; the LR image; reconstructed by **Algorithm 1**; reconstructed by **Algorithm 2**.

(i.e., the 7×7 Gaussian blurring kernel of standard deviation 1.6) and then downsampling by scale factor 3 in both horizontal and vertical directions. The Gaussian noises with standard deviation of 5 are also added to the simulated noisy LR images. Since the human visual system is more sensitive to luminance changes, we only apply the reconstruction methods to the luminance channel and use the bicubic interpolator for other chromatic channels.

In the experiments of image super-resolution, we empirically found that the proposed **Algorithms 1** and **2** usually converged after 80 iterations. Thus we simply set *Max_Iter*=80 in the proposed **Algorithms 1** and **2**. In the proposed **Algorithm 1**, we empirically set $\lambda=0.035$, and $\lambda=0.008\ \sigma_n^2$ for noiseless and noisy low-resolution images, respectively. While in the proposed **Algorithm 2**, we set the threshold $\tau_{i,j}=0.11/((\hat{\sigma}_{i,j}+0.1)\gamma)$, $\eta=0.015$, and $\tau_{i,j}=0.15\lambda_{i,j}/\gamma$, $\eta=0.015$ for noiseless and noisy low-resolution images, respectively, where $\hat{\sigma}_{i,j}$ denotes the standard deviation of the sparse coefficients $\alpha_{i,j}$, and $\lambda_{i,j}$ is computed with Eq. (17). In both **Algorithms 1** and **2**, we set $\gamma=0.015$.

Fig. 7 shows the reconstructed *Butterfly* images by the proposed **Algorithms 1** and **2** methods. It can be seen that **Algorithm 1** can well reconstruct the fine local image structures, while there are still some small artifacts around the edges. By locally adjusting the sparsity parameters and incorporating the nonlocal regularization, **Algorithm 2** can better suppress the artifacts leading to more visually pleasant results.

We then compare the proposed IR methods to several latest developed image super-resolution methods, including the TV-based super-resolution method [42], and the sparse-based method [17], and the recently developed ASDS-Reg method [15]. As suggested in [17], we used the iterative back-projection to remove the blur of the reconstructed HR images. The PSNR results of the reconstructed images are listed in Table 2. From the table, we can see that the proposed **Algorithm 2** significantly outperforms TV [42] method and sparsity-based method [17] and it slightly outperforms the ASDS-Reg method that incorporates the prior statistics learned from a training set.

Table 2
PSNR (dB) results (luminance components) of the reconstructed HR images.

Images	*Butterfly*	*flower*	*Girl*	*Pathenon*	*Parrot*	*Leaves*	*Plants*	Average
	Noiseless, $\sigma_n=0$							
TV [42]	26.56	27.51	31.24	26.00	27.85	24.51	31.34	27.86
Sparsity [17]	24.70	27.87	32.87	26.27	28.70	24.14	31.55	28.01
ASDS-Reg [15]	27.14	**29.32**	**33.55**	26.95	30.37	**26.65**	33.42	**29.63**
Algorithm 1	26.85	28.87	33.32	26.88	30.14	26.20	32.93	29.31
Algorithm 2	**27.13**	29.23	33.37	**26.99**	**30.44**	26.64	**33.44**	29.61
	Noisy, $\sigma_n=5$							
TV [42]	25.49	26.57	29.86	25.35	27.01	23.75	29.70	26.82
Sparsity [17]	23.61	26.60	30.71	25.40	27.15	22.94	29.57	26.57
ASDS-Reg [15]	25.91	27.66	31.70	26.07	28.86	25.23	30.91	28.05
Algorithm 1	25.86	27.66	31.70	26.25	29.06	25.20	30.99	28.10
Algorithm 2	**26.06**	**27.76**	**31.73**	**26.29**	**29.28**	**25.48**	**31.26**	**28.27**

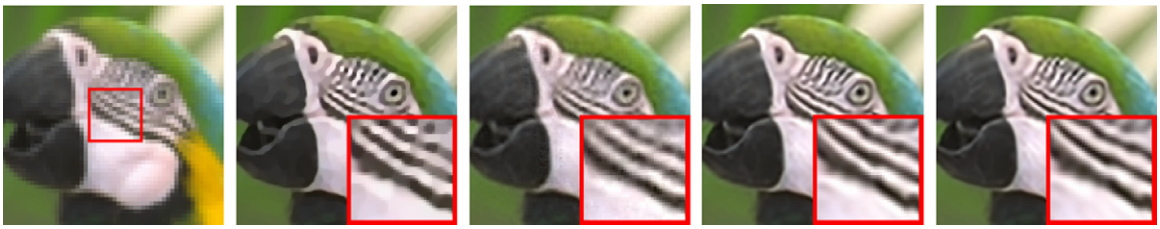

Fig. 8. Reconstructed HR *Parrots* images ($\sigma_n=0$). From left to right: input LR image; TV [42] (PSNR=27.85 dB); Sparsity [17] (PSNR=28.70 dB); ASDS-Reg [15] (PSNR=30.37 dB); **Algorithm 2** (PSNR=**30.44** dB).

Fig. 9. Reconstructed HR *Plants* images ($\sigma_n=0$). From left to right: input LR image; TV [42] (PSNR=31.34 dB); Sparsity [17] (PSNR=31.55 dB); ASDS-Reg [15] (PSNR=33.42 dB); **Algorithm 2** (PSNR=**33.44** dB).

Fig. 10. Reconstructed HR *Girl* images ($\sigma_n=5$). From left to right: input LR image; TV [42] (PSNR=29.86 dB); Sparsity [17] (PSNR=30.71 dB); ASDS-Reg [15] (PSNR=31.70 dB); **Algorithm 2** (PSNR=**31.73** dB).

The visual comparison of the reconstructed HR images by the test methods are shown in Figs. 8–10. From these figures, we can see that the proposed **Algorithm 2** enjoys significant advantages over competing methods in reconstructing sharp and clear image edges and textures.

5.3. Compressive sensing

In compressive sensing (CS), the observed measurement data $\boldsymbol{y}$ is generated by random projection of the original images. Specifically, we generated the random

measurements by randomly sampling the Fourier transform coefficients of the input image $\boldsymbol{x}$ [43]. We then use the proposed IR based methods for CS recovery and compared them to some recently developed CS recovery methods, i.e., the well-known l_1-magic method [44] that assumes the sparsity in gradient space; the iterative reweighted l_1-minimization method [35]; and the auto-regressive model based method [45]. Four photographic images and two biomedical images (i.e., Mic. image1 and Mic. image 2) are used in our comparison study. The number of measurements is presented as the percentage of the total number of the pixels N.

In the experiments of image compressive sensing, the proposed **Algorithm 1** and **2** usually converged after 160 iterations. Thus we empirically set $Max\_Iter=160$ in the proposed **Algorithms 1** and **2**. In the proposed **Algorithm 1**, the λ is set to 0.018, 0.014, 0.009, and 0.006 for sensing rates 10–40%, respectively. While in the proposed **Algorithm 2**, the threshold $\tau_{i,j}$ is set to $0.2/((\hat{\sigma}_{i,j}+0.1)\gamma)$, $0.14/((\hat{\sigma}_{i,j}+0.1)\gamma)$, $0.08/((\hat{\sigma}_{i,j}+0.1)\gamma)$, and $0.03/((\hat{\sigma}_{i,j}+0.1)\gamma)$ for sensing rates 10–40%, respectively, where $\hat{\sigma}_{i,j}$ denotes the standard deviation of the sparse coefficients $\alpha_{i,j}$, and η is set to 0.005. In both **Algorithms 1** and **2**, γ is set to 0.01.

Again, we first presented the results obtained with the proposed methods. As shown in Fig. 11, **Algorithm 1** can recover the large image structures, while there are still many artifacts around the edges. **Algorithm 2** can better recover fine image details and greatly suppress the artifacts. The PSNR results of the competing CS recovery approaches are listed in Table 3. From this table, we can see that the proposed **Algorithm 2** significantly outperforms other competing methods. The average PSNR gain of **Algorithm 2** over the second best method, i.e., MARX, can be up to 3.04 dB (at $M=0.3N$). In many cases, the proposed **Algorithm 2** can achieve comparable or even higher PSNR than other methods using only $0.1N$ measurements.

For visual comparison, Figs. 12–14 show some cropped images reconstructed by the competing methods for various measurement rates. From these figures, we can see that **Algorithm 2** can reproduce much sharper and clearer images than other competing methods. Such a large difference in visual quality demonstrates the effectiveness of the locally adaptive sparse model and nonlocal self-similarity constraint.

An important issue of the proposed algorithms is the number of scales of the learned multi-scale dictionaries.

Fig. 11. The reconstructed *Barbara* images by the proposed methods (using $M=0.2N$ measurements). From left to right: Original image; reconstructed image by **Algorithm 1**; reconstructed image by **Algorithm 2**.

Table 3
The PSNR (dB) values of the reconstructed images by different methods.

Sensing Rate	Recovery methods	*Barbara*	*House*	*Monarch*	*Boats*	*Mic. Image 1*	*Mic. Image 2*	Average
$M=0.1N$	TV [44]	22.82	30.58	24.27	25.64	20.08	19.03	23.74
	Reweighted l_1 [35]	22.59	30.91	24.39	25.57	19.83	18.80	23.68
	MARX [45]	22.64	29.92	24.72	25.79	21.16	19.59	23.97
	Algorithm 1	24.16	30.50	22.95	25.65	19.71	19.01	23.66
	Algorithm 2	**26.06**	**33.01**	**26.26**	**28.18**	**21.98**	**20.69**	**26.03**
$M=0.2N$	TV [44]	24.80	33.72	28.88	29.08	22.72	21.20	26.73
	Reweighted l_1 [35]	24.74	33.78	29.17	28.95	22.42	20.96	26.67
	MARX [45]	26.98	33.39	29.20	29.60	24.65	22.72	27.76
	Algorithm 1	27.82	34.01	27.72	29.56	22.63	21.24	27.16
	Algorithm 2	**32.11**	**35.61**	**31.45**	**32.70**	**26.03**	**24.02**	**30.32**
$M=0.3N$	TV [44]	26.79	35.63	32.01	31.59	24.79	23.14	28.99
	Reweighted l_1 [35]	26.87	35.75	32.51	31.79	24.56	22.99	29.08
	MARX [45]	30.77	35.10	32.20	32.24	27.39	24.86	30.43
	Algorithm 1	31.15	36.47	31.29	32.74	25.18	23.39	30.04
	Algorithm 2	**35.84**	**37.99**	**35.09**	**36.28**	**28.87**	**26.76**	**33.47**
$M=0.4N$	TV [44]	28.93	37.29	34.91	33.84	26.66	25.05	31.11
	Reweighted l_1 [35]	29.32	37.27	35.36	34.29	26.42	25.11	31.29
	MARX [45]	33.61	36.79	34.94	34.62	29.58	26.88	32.73
	Algorithm 1	34.08	38.66	34.38	35.41	27.36	25.24	32.52
	Algorithm 2	**38.72**	**40.98**	**38.39**	**39.25**	**31.00**	**28.78**	**36.19**

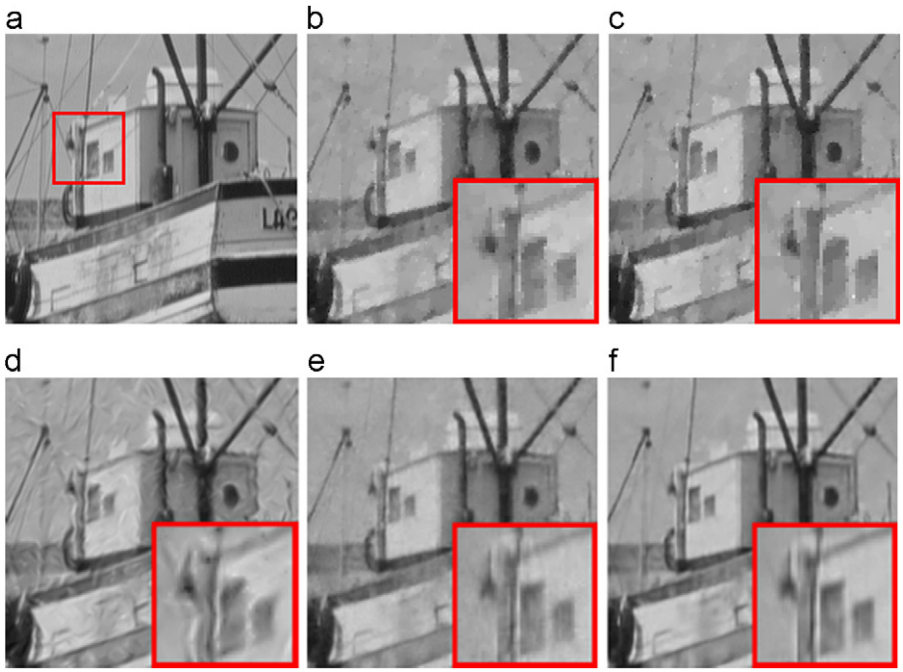

Fig. 12. The reconstructed *Boats* images (using $M=0.2N$ measurements). (a) Original image; (b) TV [44] (PSNR=29.08 dB); (c) Reweighted l_1 [35] (PSNR=28.95 dB); (d) MARX [45] (PSNR=29.60 dB); (e) the proposed **Algorithm 1** (PSNR=29.56 dB); (f) the proposed **Algorithm 2** (PSNR=**32.70** dB).

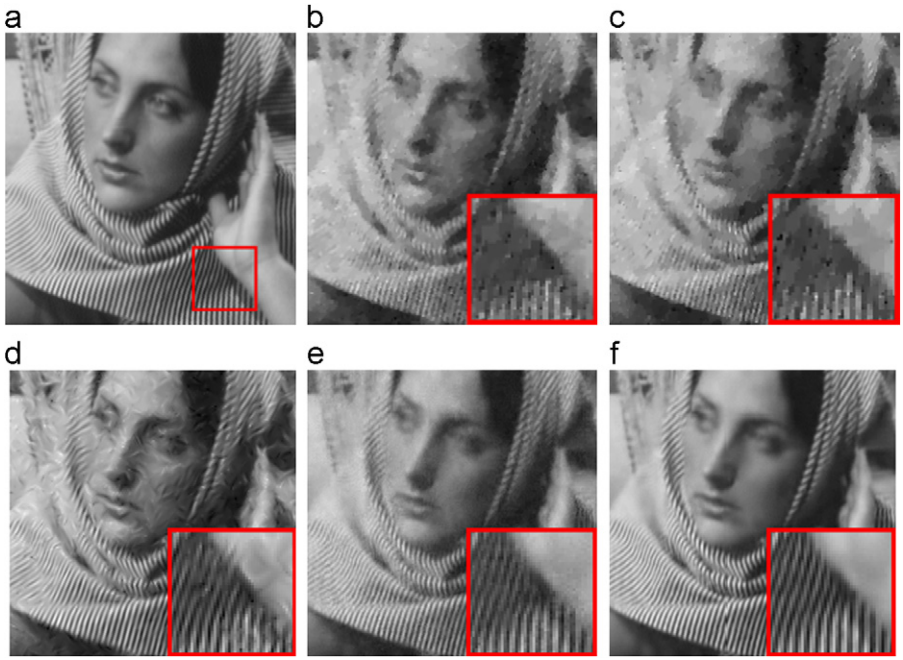

Fig. 13. The reconstructed *Barbara* images (using $M=0.2N$ measurements). (a) Original image; (b) TV [44] (PSNR=24.80 dB); (c) Reweighted l_1 [35] (PSNR=24.74 dB); (d) MARX [45] (PSNR=26.98 dB); (e) the proposed **Algorithm 1** (PSNR=27.82 dB); (f) the proposed **Algorithm 2** (PSNR=**32.11** dB).

Clearly, the number of scales cannot be too big; otherwise the size of the root patch will be too large such that the learned dictionary cannot well represent the micro-structure of natural images. Also, more scales will much increase the computational complexity of the propose algorithms. To evaluate the effects of the number of scales of **Algorithm 2**, we set the number of scales to 3 and the root patch size to 12×12, and apply **Algorithm 2** for image deblurring, super-resolution (SR), and compressive sensing (CS). The average PSNR results are shown in Table 4, from which we can see that 2 and 3 scales lead to similar PSNR results.

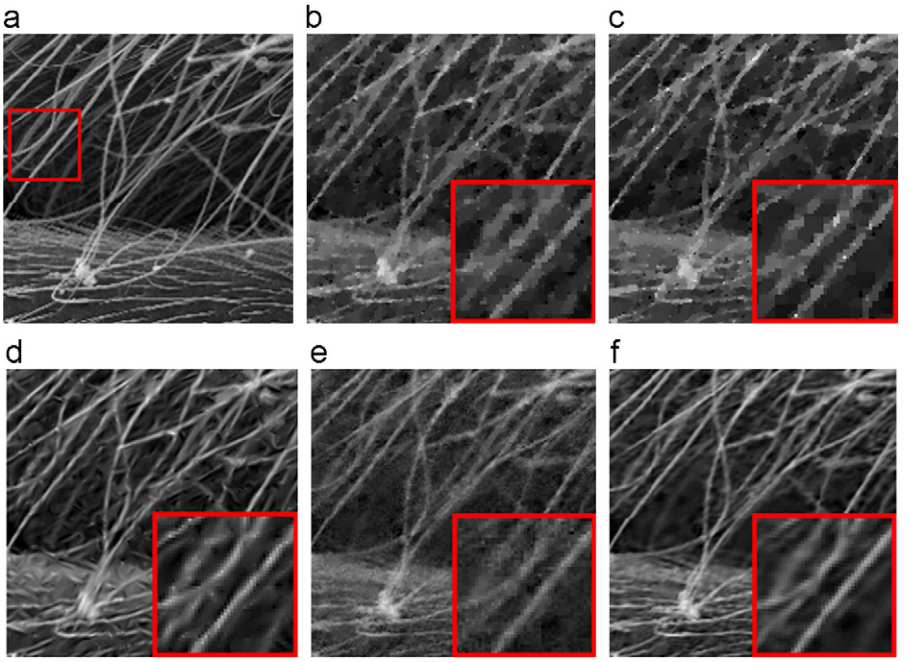

Fig. 14. The reconstructed *Mic Image 2* images (using $M=0.3N$ measurements). (a) Original image; (b) TV [44] (PSNR=23.14 dB); (c) Reweighted l_1 [35] (PSNR=22.99 dB); (d) MARX [45] (PSNR=24.86 dB); (e) the proposed **Algorithm 1** (PSNR=23.39 dB); (f) the proposed **Algorithm 2** (PSNR=**26.76** dB).

Table 4
Average PSNR (dB) results of **Algorithm 2** with different number of scales.

Number of scales	Deblurring (9×9 uniform blur, $\sigma_n=\sqrt{2}$)	SR ($\sigma_n=0$)	CS, rate=0.1N	CS, rate=0.2N	CS, rate=0.3N	CS, rate=0.4N
2 scales	29.47	29.61	26.03	30.32	33.47	36.19
3 scales	29.61	29.68	26.05	30.36	33.51	36.24

5.4. Computational complexity

In **Algorithms 1** and **2**, the computation of local dictionaries requires $N\times L$ operations of K-nearest neighbor search and PCA, where N and L denotes the number of extracted patches and the number of scales, respectively. The computational complexity of PCA dictionaries is then $O(N\times L\times m\times n^2/2)$, where m and n denote the number of samples collected for PCA and the size of the patch vector, respectively. In Step 2(a), the dominating operation in an iteration of the conjugate gradient (CG) solver is the matrix-vector multiplication, i.e., multiplying $\boldsymbol{A}$ with vector $\boldsymbol{z}$, where $\boldsymbol{A}$ denotes $\boldsymbol{H}^T\boldsymbol{H}+\gamma\boldsymbol{I}$ and $\boldsymbol{H}^T\boldsymbol{H}+\eta(\boldsymbol{I}-\boldsymbol{B})^T(\boldsymbol{I}-\boldsymbol{B})+\gamma\boldsymbol{I}$ in **Algorithms 1** and **2**, respectively. Since both $\boldsymbol{H}$ and $\boldsymbol{B}$ are sparse, the matrix-vector multiplication can be executed fast. Moreover, for image deblurring, the multiplication with $\boldsymbol{H}^T\boldsymbol{H}$ can be implemented by fast FFT, which is much faster than direct matrix calculation. In general, the CG solver converges in κ iterations, where κ denotes the condition number of matrix $\boldsymbol{A}$. Hence, the computation complexity of the CG algorithm is $O(4M\kappa)$, where M denotes the number of total pixels of the image. The Steps 2(b) and (d) require $L\times N\times n$ multiplications. Since each patch is coded individually, Steps 2(b) and (d) can be executed in parallel to speed up the algorithm. To save the computational complexity, we update the dictionaries in every T ($T=40$ in our implementation) iterations. For a 256×256 image, **Algorithm 1** requires about 2–3 min for image deblurring and SR and 3–4 min for CS on an Intel Core2 Duo 2.79 G PC under Matlab R2010a environment. While **Algorithm 2** requires 5–6 min for image deblurring and SR and 9–10 min for CS.

6. Conclusion

A novel image reconstruction (IR) method using local adaptive sparsity and nonlocal regularization was presented. Considering the fact that the image content can vary significantly across the image, it is necessary to adapt the dictionary to each local patch. In this paper, we proposed to learn the dictionaries using similar patches both within the same scale and across different scales. Along with the local learning of the dictionary, we also adaptively estimate the sparsity-regularization parameters using those coefficients from similar patches. To balance the tradeoff between adaptation and robustness, we further introduced a nonlocal self-similarity constraint to suppress the inconsistency that may be caused by the pixel-wise adaptation. A new iterative sparsity-based IR algorithm was then presented.

Experimental results demonstrated that the proposed method outperforms many state-of-the-arts IR methods in terms of PSNR and subject evaluation.

Acknowledgement

This work was supported in part by the Natural Science Foundation of China under (Grants 61100154, 61033004, 61070138, and Grant 61072104), in part by Major State Basic Research Development Program of China (973 Program) (No. 2013CB329402), in part by the Fundamental Research Funds of the Central Universities of China under Grant K50510020003, in part by NSF-CCF-0914353 and NSF-ECCS-0968730, and in part by Hong Kong RGC General Research Fund (PolyU 5375/09E).

References

[1] M. Bertero, P. Boccacci, Introduction to Inverse Problems in Imaging, IOP, Bristol, U.K., 1998.
[2] A.N. Tikhonov, Solution of incorrectly formulated problems and regularization method, Soviet Mathematics. Doklady 4 (1963) 1035–1038.
[3] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.
[4] J.H. Money, S.H. Kang, Total variation minimizing blind deconvolution with shock filter reference, Image and Vision Computing 27 (2008) 302–314.
[5] 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 Transactions on Image Processing 19 (2) (Feb. 2010) 351–362.
[6] Q. Chen, P. Montesinos, Q.S. Sun, P.A. Heng, D.S. Xia, Adaptive total variation denoising based on difference curvature, Image and Vision Computing 28 (2010) 298–306.
[7] M. Lysaker, X. Tai, Iterative image restoration combining total variation minimization and a second-order functional, International Journal of Computer Vision 66 (1) (2006) 5–18.
[8] X. Zhang, M. Burger, X. Bresson, S. Osher, Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM Journal on Imaging Sciences 3 (3) (2010) 253–276.
[9] L.L. Sulem, J. Darbon, E.H.B. Smith, Enhancement of historical printed document images by combining total variation regularization and non-local means filtering, Image and Vision Computing 29 (2011) 351–363.
[10] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Transactions on Image Processing 15 (12) (2006) 3736–3745.
[11] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, IEEE Transactions on Image Processing 16 (2007) 2080–2095.
[12] D. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.
[13] E.J. Candès, T. Tao, Near optimal signal recovery from random projections: universal encoding strategies? IEEE Transactions on Information Theory 52 (12) (2006) 5406–5425.
[14] E.J. Candes, J. Romberg, Sparsity and incoherence in compressive sampling, Inverse Problems 23 (2007) 969–985.
[15] W. Dong, L. Zhang, G. Shi, X. Wu, Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization, IEEE Transactions on Image Processing 20 (2011) 1838–1857.
[16] M.J. Fadili, J.L. Starck, Sparse Representation-based Image Deconvolution by Iterative Thresholding, Astronomical Data Analysis, Marseille, France, 2006.
[17] J. Yang, J. Wright, T. Huang, Y. Ma, Image super-resolution via sparse representation, IEEE Transactions on Image Processing 19 (2010) 2861–2873.
[18] S. Mallat, G. Yu, Super-resolution with sparse mixing estimators, IEEE Transactions on Image Processing 19 (2010) 2889–2900.
[19] W. Dong, G. Shi, L. Zhang, X. Wu, Super-resolution with nonlocal regularization sparse representation, Proceedings of SPIE Visual Communication and Image Processing 7744 (2010).
[20] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (12) (1993) 3397–3415.
[21] S. Chen, D. Donoho, M. Saunders, Atomic decompositions by basis pursuit, SIAM Review 43 (2001) 129–159.
[22] I. Daubechies, M. Defriese, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Communications on Pure and Applied Mathematics 57 (2004) 1413–1457.
[23] P. Combettes, V. Wajs, Signal recovery by proximal forward-backward splitting, SIAM Journal on Multiscale Modeling and Simulation 4 (2005) 1168–1200.
[24] E.J. Candes, D. Donoho, New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities, Communications on Pure and Applied Mathematics 57 (2004) 219–266.
[25] M.N. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, IEEE Transactions on Image Processing 14 (12) (2005) 2091–2106.
[26] S. Mallat, G. Peyre, Orthogonal bandlet bases for geometric images approximation, Communications on Pure and Applied Mathematics 61 (2008) 1173–1212.
[27] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (11) (2006) 4311–4322.
[28] J. Mairal, G. Sapiro, M. Elad, Learning multiscale sparse representations for image and video restoration, SIAM Journal on Multiscale Modeling and Simulation 7 (1) (2008) 214–241.
[29] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, Non-local sparse models for image restoration, IEEE International Conference on Computer Vision, Tokyo, Japan, 2009.
[30] M. Elad, I. Yavneh, A plurality of sparse representation is better than the sparsest one alone, IEEE Transactions on Information Theory 55 (10) (Oct. 2009) 4701–4714.
[31] A. Buades, B. Coll, J.M. Morel, A review of image denoising algorithms, with a new one, SIAM Journal on Multiscale Modeling and Simulation 4 (2) (2005) 490–530.
[32] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," IEEE International Conference on Computer Vision, Tokyo, Japan, 2009.
[33] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, Neural Computation 9 (7) (1997) 1493–1516.
[34] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.
[35] E. Candes, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, Journal of Fourier Analysis and Applications 14 (2008) 877–905.
[36] L. Zhang, R. Lukac, X. Wu, D. Zhang, PCA-based spatially adaptive denoising of CFA images for single-sensor digital cameras, IEEE Transactions on Image Processing 18 (4) (2009) 797–812.
[37] L. Zhang, W. Dong, D. Zhang, G. Shi, Two-stage image denoising by principal component analysis with local pixel grouping, Pattern Recognition 43 (2010) 1531–1549.
[38] D. Donoho, Wedgelets: nearly minimax estimation of edges, Annals of Statistics 27 (1998) 859–897.
[39] R. Fergus, B. Singh, A. Hertzmann, S.T. Roweis, W.T. Freeman, Removing camera shake from a single image, ACM Transactions on Graphics (SIGGRAPH) 25 (2006) 787–794.
[40] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Transactions on Image Processing 18 (11) (2009) 2419–2434.
[41] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, Image Restoration by Sparse 3D Transform-Domain Collaborative Filtering in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6812, 2008.
[42] A. Marquina, S.J. Osher, Image super-resolution by TV-regularization and Bregman iteration, Journal of Scientific Computing 37 (2008) 367–382.
[43] E.J. Candes, J. Romberg, Practical signal recovery from random projections, Proceedings of SPIE Computational Imaging 5674 (2005) 76–86.
[44] 〈http://www.acm.caltech.edu./l1magic/〉.
[45] X. Wu, X. Zhang and J. Wang, Model-guided adaptive recovery of compressive sensing, in: Proc. Data Compression Conference, pp. 123–132, Mar. 2009.
[46] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM Journal on Imaging Sciences 1 (3) (2008) 248–272.