Group-based Sparse Representation for Image Restoration

Jian Zhang, *Student Member*, *IEEE*, Debin Zhao, *Member*, *IEEE*, and Wen Gao, *Fellow*, *IEEE*

Abstract

Traditional patch-based sparse representation modeling of natural images usually suffer from two problems. First, it has to solve a large-scale optimization problem with high computational complexity in dictionary learning. Second, each patch is considered independently in dictionary learning and sparse coding, which ignores the relationship among patches, resulting in inaccurate sparse coding coefficients. In this paper, instead of using patch as the basic unit of sparse representation, we exploit the concept of group as the basic unit of sparse representation, which is composed of nonlocal patches with similar structures, and establish a novel sparse representation modeling of natural images, called group-based sparse representation (GSR). The proposed GSR is able to sparsely represent natural images in the domain of group, which enforces the intrinsic local sparsity and nonlocal self-similarity of images simultaneously in a unified framework. Moreover, an effective self-adaptive dictionary learning method for each group with low complexity is designed, rather than dictionary learning from natural images. To make GSR tractable and robust, a split Bregman based technique is developed to solve the proposed GSR-driven ℓ_0 minimization problem for image restoration efficiently. Extensive experiments on image inpainting, image deblurring and image compressive sensing recovery manifest that the proposed GSR modeling outperforms many current state-of-the-art schemes in both PSNR and visual perception.

***Index Terms*—Image restoration, sparse representation, nonlocal self-similarity, deblurring, inpainting, compressive sensing**

J. Zhang and D. Zhao are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: jzhangcs@hit.edu.cn; dbzhao@hit.edu.cn).

W. Gao is with the National Engineering Laboratory for Video Technology, and Key Laboratory of Machine Perception (MoE), School of Electrical Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: wgao@pku.edu.cn).

I. Introduction

Image restoration has been extensively studied in the past two decades [1]–[20], whose purpose is to reconstruct the original high quality image $\boldsymbol{x}$ from its degraded observed version $\boldsymbol{y}$. It is a typical ill-posed linear inverse problem and can be generally formulated as:

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are lexicographically stacked representations of the original image and the degraded image, respectively, $\boldsymbol{H}$ is a matrix representing a non-invertible linear degradation operator and $\boldsymbol{n}$ is usually additive Gaussian white noise. When $\boldsymbol{H}$ is a mask, that is, $\boldsymbol{H}$ is a diagonal matrix whose diagonal entries are either 1 or 0, keeping or killing the corresponding pixels, the problem (1) becomes image inpainting [5], [6]; when $\boldsymbol{H}$ is a blur operator, the problem (1) becomes image deblurring [9], [18]; when $\boldsymbol{H}$ is a set of random projections, the problem (1) becomes compressive sensing (CS) [19], [42].

To cope with the ill-posed nature of image restoration, image prior knowledge is usually employed for regularizing the solution to the following minimization problem [8]–[18]:

$$\operatorname{argmin}_{\boldsymbol{x}} \tfrac{1}{2}\|\boldsymbol{Hx} - \boldsymbol{y}\|_2^2 + \lambda \Psi(\boldsymbol{x}), \tag{2}$$

where $\tfrac{1}{2}\|\boldsymbol{Hx} - \boldsymbol{y}\|_2^2$ is the ℓ_2 data-fidelity term, $\Psi(\boldsymbol{x})$ is called the regularization term denoting image prior and λ is the regularization parameter. Many optimization approaches for the above regularization-based image inverse problems have been developed [16]–[18], [43].

Due to that image prior knowledge plays a critical role in the performance of image restoration algorithms, designing effective regularization terms to reflect the image priors is at the core of image restoration. Classical regularization terms, such as half quadrature formulation [21], Mumford-Shah (MS) model [22], and total variation (TV) models [1] [4], utilize local structural patterns and are built on the assumption that images are locally smooth except at edges. These regularization terms demonstrate high effectiveness in preserving edges and recovering smooth regions. However, they usually smear out image details and cannot deal well with fine structures.

In the past several years, sparsity has been emerging as one of the most significant properties of natural images [23], [24] and the sparsity-based regularization has achieved great success in various image processing applications, such as denoising [25], deblurring [11], and super-resolution [26]. The sparse model assumes that each patch of an image can be accurately represented by a few elements from a basis set called a dictionary, which is learned from natural images. Compared with traditional analytically-designed dictionaries, such as wavelets, curvelets, and bandlets, the learned dictionary enjoys the advantage of being better adapted to the images, thereby enhancing the sparsity and showing impressive performance improvement. However, there exist two main problems in the current patch-based sparse representation model. First, dictionary learning is a large-scale and highly non-convex problem, which often requires high computational complexity [24], [27]. Second, patch is the unit of sparse representation, and each patch is usually considered independently in dictionary learning and sparse coding, which ignores the relationships between similar patches in essence, such as self-similarity. Moreover, with the learned dictionary, the actual sparse coding process is always calculated with relatively expensive nonlinear estimations, such as match pursuits [28], [38], which also may be unstable and imprecise due to the coherence of the dictionary [37].

Another alternative significant property exhibited in natural images is the well-known nonlocal self-similarity, which depicts the repetitiveness of higher level patterns (e.g., textures and structures) globally positioned in images. Inspired by the success of nonlocal means (NLM) denoising filter [8], a series of nonlocal regularization terms for inverse problems exploiting nonlocal self-similarity property of natural images are emerging [32]–[36]. Due to the utilization of self-similarity prior by adaptive nonlocal graph, nonlocal regularization terms produce superior results over the local ones, with sharper image edges and more image details [33], [36]. Nonetheless, there are still plenty of image details and structures that cannot be recovered accurately. One of the reasons is that the weighted graphs adopted by the above nonlocal regularization terms inevitably give rise to disturbance and inaccuracy, due to the inaccurate weights [35].

In recent works, the sparsity and the self-similarity of natural images are usually combined to achieve better performance. In [11], sparsity and self-similarity are separately characterized by two regularization terms, which are incorporated together into the final cost functional of image restoration solution to enhance the image quality. In [12], simultaneous sparse coding is utilized to impose that similar patches should share the same dictionary elements in their sparse decomposition, which acquired impressive denoising and demosaicking results. In [15], a nonlocally centralized sparse representation (NCSR) model is proposed, which first obtains good estimates of the sparse coding coefficients of the original image by the principle of NLM in the domain of sparse coding coefficients, and then centralizes the sparse coding coefficients of the observed image to those estimates to improve the performance of sparse representation based image restoration.

Lately, low-rank modeling based approaches have also achieved great success in image or video restoration. To remove the defects in a video, unreliable pixels in the video are first detected and labeled as missing. Similar patches are grouped such that the patches in each group share similar underlying structure and form a low-rank matrix approximately. Finally, the matrix completion is carried out on each patch group to restore the image [50] [51]. In [5], a low-rank approach toward modeling nonlocal similarity denoted by SAIST is proposed, which not only provides a conceptually simple interpretation for simultaneous sparse coding [12] from a bilateral variance estimation perspective, but also achieves highly competent performance to several state-of-the-art methods.

In this paper, instead of using patch as the basic unit of sparse representation, we exploit the concept of group as the basic unit of sparse representation, and establish a novel sparse representation modeling of natural images, called group-based sparse representation (GSR). Compared with traditional patch-based sparse representation, the contributions of our proposed GSR modeling are mainly three folds. First, GSR explicitly and effectively characterizes the intrinsic local sparsity and nonlocal self-similarity of natural images simultaneously in a unified framework, which adaptively sparsifies the natural image in the domain of group. Second, an effective self-adaptive group dictionary learning method with low complexity is designed, rather than dictionary learning from natural images. Third, an efficient split Bregman based iterative algorithm is developed to solve the proposed GSR-driven ℓ_0 minimization problem for image restoration. Experimental results on three applications: image inpainting, deblurring and image CS recovery have shown that the proposed GSR model outperforms many current state-of-the-art schemes. Part of our previous work for image CS recovery via GSR has been presented in [47].

The remainder of this paper is organized as follows. Traditional patch-based sparse representation is introduced in Section II. Section III elaborates the design of group-based sparse representation (GSR) modeling, and discusses the close relationships among the GSR model, the group sparsity model and the low rank model. Section IV proposes a new objective functional formed by our proposed GSR, and gives the implementation details of solving optimization problem. Extensive experimental results are reported in Section V. In Section VI, we summarize this paper.

II. Traditional Patch-based Sparse Representation

In literature, the basic unit of sparse representation for natural images is patch [24]–[26]. Mathematically, denote by $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{x}_k \in \mathbb{R}^{\mathcal{B}_s}$ the vector representations of the original image and an image patch of size $\sqrt{\mathcal{B}_s} \times \sqrt{\mathcal{B}_s}$ at location $k, k = 1, 2, ..., n$, where N and $\mathcal{B}_s$ are the size of the whole image vector and each image patch vector, respectively, and n is the number of image patches. Then we have

$$\boldsymbol{x}_k = \boldsymbol{R}_k(\boldsymbol{x}), \tag{3}$$

where $\boldsymbol{R}_k(\cdot)$ is an operator that extracts the patch $\boldsymbol{x}_k$ from the image $\boldsymbol{x}$, and its transpose, denoted by $\boldsymbol{R}_k^T(\cdot)$, is able to put back a patch into the k-th position in the reconstructed image, padded with zeros elsewhere. Considering that patches are usually overlapped, the recovery of $\boldsymbol{x}$ from $\{\boldsymbol{x}_k\}$ becomes

$$\boldsymbol{x} = \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{x}_k) \,./ \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{1}_{\mathcal{B}_s}), \tag{4}$$

where the notation $./$ stands for the element-wise division of two vectors, and $\boldsymbol{1}_{\mathcal{B}_s}$ is a vector of size $\mathcal{B}_s$ with all its elements being 1. Eq. (4) indicates nothing but an abstraction strategy of averaging all the overlapped patches.

For a given dictionary $\boldsymbol{D} \in \mathbb{R}^{\mathcal{B}_s \times M}$ (M is the number of atoms in $\boldsymbol{D}$), the sparse coding process of each patch $\boldsymbol{x}_k$ over $\boldsymbol{D}$ is to find a sparse vector $\boldsymbol{\alpha}_k \in \mathbb{R}^M$ (i.e., most of the coefficients in $\boldsymbol{\alpha}_k$ are zero or close to zero) such that $\boldsymbol{x}_k \approx \boldsymbol{D}\boldsymbol{\alpha}_k$. Then the entire image can be sparsely represented by the set of sparse codes $\{\boldsymbol{\alpha}_k\}$. In practice, the sparse coding problem of $\boldsymbol{x}_k$ over $\boldsymbol{D}$ is usually cast as

$$\boldsymbol{\alpha}_k = \operatorname{argmin}_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{x}_k - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p, \tag{5}$$

where λ is a constant, and p is 0 or 1. If p = 0, that means the sparsity is strictly measured by the ℓ_0-norm of $\boldsymbol{\alpha}_k$, which counts the nonzero elements in $\boldsymbol{\alpha}_k$. Nonetheless, since the problem of ℓ_0-norm optimization is non-convex and usually NP-hard, it is often sub-optionally solved by greedy algorithms, e.g., orthogonal matching pursuit (OMP) [28]. Alternatively, if p = 1, the ℓ_0-norm minimization is approximated by the convex ℓ_1-norm, which can be efficiently solved by some recent large-scale tools [16]–[18], [38], [43].

Similar to Eq. (4), reconstructing $\boldsymbol{x}$ from its sparse codes $\{\boldsymbol{\alpha}_k\}$ is formulated:

$$\boldsymbol{x} = \boldsymbol{D} \circ \boldsymbol{\alpha} \overset{def}{=} \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{D}\boldsymbol{\alpha}_k) \,./ \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{1}_{\mathcal{B}_s}), \tag{6}$$

where $\boldsymbol{\alpha}$ denotes the concatenation of all $\boldsymbol{\alpha}_k$, that is, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T, ..., \boldsymbol{\alpha}_n^T]^T$. The purpose of introducing the notation $\circ$ is to exploit $\boldsymbol{D} \circ \boldsymbol{\alpha}$ to make the expression of $\sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{D}\boldsymbol{\alpha}_k) \,./ \sum\nolimits_{k=1}^{n} \boldsymbol{R}_k^T(\boldsymbol{1}_{\mathcal{B}_s})$ more convenient.

Now, considering the degraded version in Eq. (1), the regularization-based image restoration scheme utilizing traditional patch-based sparse representation model is formulated as

$$\hat{\boldsymbol{\alpha}} = \operatorname{argmin}_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{H}\boldsymbol{D} \circ \boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p, \tag{7}$$

where λ is the regularization parameter, and p is 0 or 1. With $\hat{\boldsymbol{\alpha}}$, the reconstructed image can be expressed by $\hat{\boldsymbol{x}} = \boldsymbol{D} \circ \hat{\boldsymbol{\alpha}}$.

The heart of the sparse representation modeling lies in the choice of dictionary $\boldsymbol{D}$. In other words, how to seek the best domain to sparsify a given image? Much effort has been devoted to learning a redundant dictionary from a set of training example image patches. To be concrete, given a set of training image patches $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_J]$, where J is the number of training image patches, the goal of dictionary learning is to jointly optimize the dictionary $\boldsymbol{D}$ and the representation coefficients matrix

$\boldsymbol{\Lambda} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_J]$ such that $\boldsymbol{x}_k \approx \boldsymbol{D}\boldsymbol{\alpha}_k$ and $\|\boldsymbol{\alpha}_k\|_p \le L$, where p is 0 or 1. This can be formulated by the following minimization problem:

$$(\hat{\boldsymbol{D}}, \hat{\boldsymbol{\Lambda}}) = \underset{\boldsymbol{D},\boldsymbol{\Lambda}}{\operatorname{argmin}} \sum\nolimits_{k=1}^{J} \|\boldsymbol{x}_k - \boldsymbol{D}\boldsymbol{\alpha}_k\|_2^2 \text{ s.t. } \|\boldsymbol{\alpha}_k\|_p \le L, \forall k. \tag{8}$$

Apparently, the above minimization problem in Eq. (8) is large-scale and highly non-convex even when p is 1. To make it tractable and solvable, some approximation approaches, including MOD [27] and K-SVD [24], have been proposed to optimize $\boldsymbol{D}$ and $\boldsymbol{\Lambda}$ alternatively, leading to many state-of-the-art results in image processing. However, these approximation approaches for dictionary learning inevitably require high computational complexity.

Apart from high computational complexity, from Eq. (5) and Eq. (8), it can be noticed that each patch is actually considered independently in dictionary learning and sparse coding, which ignores the relationships between similar patches in essence, such as self-similarity [4], [11].

III. Group-based Sparse Representation (GSR)

In this paper, to rectify the above problems of traditional patch-based sparse representation, we propose a novel sparse representation modeling in the unit of group instead of patch, aiming to exploit the local sparsity and the nonlocal self-similarity of natural images simultaneously in a unified framework. Each group is represented by the form of matrix, which is composed of nonlocal patches with similar structures. Thus, the proposed sparse representation modeling is named as group-based sparse representation (GSR). Moreover, an effective self-adaptive dictionary learning method for each group with low complexity is designed rather than dictionary learning from natural images, enabling the proposed GSR more efficient and effective. This section will give detailed description of GSR modeling, and elaborate the self-adaptive dictionary learning technique.

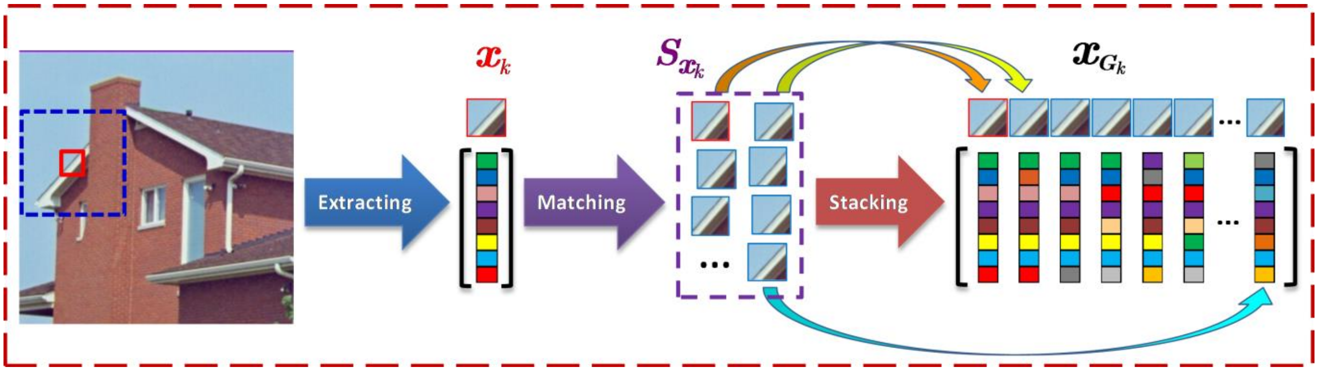


Figure 1: Illustrations for the group construction. Extract each patch vector $\boldsymbol{x}_k$ from image $\boldsymbol{x}$. For each $\boldsymbol{x}_k$, denote $\boldsymbol{S}_{\boldsymbol{x}_k}$ the set composed of its c best matched patches. Stack all the patches in $\boldsymbol{S}_{\boldsymbol{x}_k}$ in the form of matrix to construct the group, denoted by $\boldsymbol{x}_{G_k}$.

A. Group Construction

Since the unit of our proposed sparse representation model is group, this subsection will give details to show how to construct it.

As shown in Fig. 1, first, divide the image $\boldsymbol{x}$ with size N into n overlapped patches of size $\sqrt{\mathcal{B}_s}\times\sqrt{\mathcal{B}_s}$ and each patch is denoted by the vector $\boldsymbol{x}_k \in \mathbb{R}^{\mathcal{B}_s}$, i.e., $k=1,2,...,n$.

Then, for each patch $\boldsymbol{x}_k$, denoted by small red square in Fig. 1, in the $L\times L$ training window (big blue square), search its c best matched patches, which comprise the set $\boldsymbol{S}_{\boldsymbol{x}_k}$. Here, Euclidean distance is selected as the similarity criterion between different patches.

Next, all the patches in $\boldsymbol{S}_{\boldsymbol{x}_k}$ are stacked into a matrix of size $\mathcal{B}_s\times c$, denoted by $\boldsymbol{x}_{G_k}\in\mathbb{R}^{\mathcal{B}_s\times c}$, which includes every patch in $\boldsymbol{S}_{\boldsymbol{x}_k}$ as its columns, i.e., $\boldsymbol{x}_{G_k}=\{\boldsymbol{x}_{G_k\otimes 1},\boldsymbol{x}_{G_k\otimes 2},...,\boldsymbol{x}_{G_k\otimes c}\}$. The matrix $\boldsymbol{x}_{G_k}$ containing all the patches with similar structures is named as a group. Analogous to Eq. (3), we define

$$\boldsymbol{x}_{G_k}=\boldsymbol{R}_{G_k}(\boldsymbol{x}), \tag{9}$$

where $\boldsymbol{R}_{G_k}(\cdot)$ is actually an operator that extracts the group $\boldsymbol{x}_{G_k}$ from $\boldsymbol{x}$, and its transpose, denoted by $\boldsymbol{R}_{G_k}^T(\cdot)$, can put back a group into the k-th position in the reconstructed image, padded with zeros elsewhere.

By averaging all the groups, the recovery of the whole image $\boldsymbol{x}$ from $\{\boldsymbol{x}_{G_k}\}$ becomes

$$\boldsymbol{x}=\sum\nolimits_{k=1}^{n}\boldsymbol{R}_{G_k}^T(\boldsymbol{x}_{G_k})\ ./\sum\nolimits_{k=1}^{n}\boldsymbol{R}_{G_k}^T(\boldsymbol{1}_{\mathcal{B}_s\times c}), \tag{10}$$

where $./$ stands for the element-wise division of two vectors and $\boldsymbol{1}_{\mathcal{B}_s\times c}$ is a matrix of size $\mathcal{B}_s\times c$ with all the elements being 1.

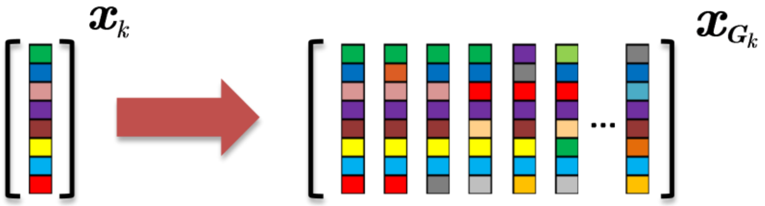


Figure 2: Comparison between patch $\boldsymbol{x}_k$ and group $\boldsymbol{x}_{G_k}$.

Note that, in our paper, each patch $\boldsymbol{x}_k$ is represented as a vector, while each group $\boldsymbol{x}_{G_k}$ is represented as a matrix, as illustrated in Fig. 2. According to above definition, it is obvious to observe that each patch $\boldsymbol{x}_k$ corresponds to a group $\boldsymbol{x}_{G_k}$. One can also see that the construction of $\boldsymbol{x}_{G_k}$ explicitly exploits the self-similarity of natural images.

B. Group-based Sparse Representation Modeling

To enforce the local sparsity and the nonlocal self-similarity of natural images simultaneously in a unified framework, we propose to sparsify natural images in the domain of group. Therefore, our proposed model is named as group-based sparse representation

(GSR). The proposed GSR model assumes that each group $\boldsymbol{x}_{\boldsymbol{G}_k}$ can be accurately represented by a few atoms of a self-adaptive learning dictionary $\boldsymbol{D}_{\boldsymbol{G}_k}$.

In this subsection, $\boldsymbol{D}_{\boldsymbol{G}_k} = [\boldsymbol{d}_{\boldsymbol{G}_k \otimes 1}, \boldsymbol{d}_{\boldsymbol{G}_k \otimes 2}, ..., \boldsymbol{d}_{\boldsymbol{G}_k \otimes m}]$ is supposed to be known. Note that each atom $\boldsymbol{d}_{\boldsymbol{G}_k \otimes i} \in \mathbb{R}^{\mathcal{B}_s \times c}$ is a matrix of the same size as the group $\boldsymbol{x}_{\boldsymbol{G}_k}$, and m is the number of atoms in $\boldsymbol{D}_{\boldsymbol{G}_k}$. Different from the dictionary in patch-based sparse representation, here $\boldsymbol{D}_{\boldsymbol{G}_k}$ is of size $(\mathcal{B}_s \times c) \times m$, that is, $\boldsymbol{D}_{\boldsymbol{G}_k} \in \mathbb{R}^{(\mathcal{B}_s \times c) \times m}$. How to learn $\boldsymbol{D}_{\boldsymbol{G}_k}$ with low complexity will be given in the next subsection.

Thus, some notations about GSR can be readily extended from patch-based sparse representation. Specifically, the sparse coding process of each group $\boldsymbol{x}_{\boldsymbol{G}_k}$ over $\boldsymbol{D}_{\boldsymbol{G}_k}$ is to seek a sparse vector $\boldsymbol{\alpha}_{\boldsymbol{G}_k} = [\boldsymbol{\alpha}_{\boldsymbol{G}_k \otimes 1}, \boldsymbol{\alpha}_{\boldsymbol{G}_k \otimes 2}, ..., \boldsymbol{\alpha}_{\boldsymbol{G}_k \otimes m}]$ such that $\boldsymbol{x}_{\boldsymbol{G}_k} \approx \sum_{i=1}^{m} \boldsymbol{\alpha}_{\boldsymbol{G}_k \otimes i} \boldsymbol{d}_{\boldsymbol{G}_k \otimes i}$. For simplicity, we utilize the expression $\boldsymbol{D}_{\boldsymbol{G}_k} \boldsymbol{\alpha}_{\boldsymbol{G}_k}$ to represent $\sum_{i=1}^{m} \boldsymbol{\alpha}_{\boldsymbol{G}_k \otimes i} \boldsymbol{d}_{\boldsymbol{G}_k \otimes i}$ without confusion. Note that $\boldsymbol{D}_{\boldsymbol{G}_k} \boldsymbol{\alpha}_{\boldsymbol{G}_k}$ is not a strict matrix-vector multiplication. It is also worth emphasizing that the sparse coding process of each group under our proposed $\boldsymbol{D}_{\boldsymbol{G}_k}$ is quite efficient without iteration, which will be seen in the following section. Then the entire image can be sparsely represented by the set of sparse codes $\{\boldsymbol{\alpha}_{\boldsymbol{G}_k}\}$ in the group domain. Reconstructing $\boldsymbol{x}$ from the sparse codes $\{\boldsymbol{\alpha}_{\boldsymbol{G}_k}\}$ is expressed as:

$$\boldsymbol{x} = \boldsymbol{D}_{\boldsymbol{G}} \circ \boldsymbol{\alpha}_{\boldsymbol{G}} \overset{def}{=} \sum_{k=1}^{n} \boldsymbol{R}_{\boldsymbol{G}_k}^T (\boldsymbol{D}_{\boldsymbol{G}_k} \boldsymbol{\alpha}_{\boldsymbol{G}_k}) \,./ \sum_{k=1}^{n} \boldsymbol{R}_{\boldsymbol{G}_k}^T (\boldsymbol{1}_{\mathcal{B}_s \times c}), \tag{11}$$

where $\boldsymbol{D}_{\boldsymbol{G}}$ denotes the concatenation of all $\boldsymbol{D}_{\boldsymbol{G}_k}$, and $\boldsymbol{\alpha}_{\boldsymbol{G}}$ denotes the concatenation of all $\boldsymbol{\alpha}_{\boldsymbol{G}_k}$.

Accordingly, by considering the degraded version in Eq. (1), the proposed regularization-based image restoration scheme via GSR is formulated as

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}} = \operatorname{argmin}_{\boldsymbol{\alpha}_{\boldsymbol{G}}} \frac{1}{2} \left\| \boldsymbol{H} \boldsymbol{D}_{\boldsymbol{G}} \circ \boldsymbol{\alpha}_{\boldsymbol{G}} - \boldsymbol{y} \right\|_2^2 + \lambda \left\| \boldsymbol{\alpha}_{\boldsymbol{G}} \right\|_0 . \tag{12}$$

With $\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}}$, the reconstructed image can be expressed by $\hat{\boldsymbol{x}} = \boldsymbol{D}_{\boldsymbol{G}} \circ \hat{\boldsymbol{\alpha}}_{\boldsymbol{G}}$. Note that, in this paper, ℓ_0-norm is exploited to measure the real sparsity of $\boldsymbol{\alpha}_{\boldsymbol{G}}$ in the group domain in order to enhance the image restoration quality. Nonetheless, Eq. (12) is usually hard to solve owing that ℓ_0-norm optimization is non-convex. Thus, solving Eq. (12) efficiently and effectively is one of our main contributions, which will be seen in the following.

To understand GSR model more clearly, here, we make a comparison between Eq. (12) and previous patch-based sparse representation for image restoration in Eq. (7). We can see the differences between Eq. (12) and Eq. (7) lie in the dictionary and the unit of sparse representation. The advantages of Eq. (12) are mainly three-folds. First, GSR adopts group as the unit of sparse representation and sparsely represents the entire image in the group domain. Since the group is composed of patches with similar structures, GSR exploits self-similarity explicitly in dictionary learning and sparse coding, which is more robust and effectual.

Second, rather than learning a general dictionary $\boldsymbol{D}$ for all patches in Eq. (7), a self-adaptive dictionary $\boldsymbol{D}_{G_k}$ is designed for each $\boldsymbol{x}_{G_k}$, which is more effective. Third, as will be seen below, the proposed self-adaptive dictionary learning of $\boldsymbol{D}_{G_k}$ is with low complexity, which doesn't require high computational complexity to solve large-scale optimizations.

C. Self-Adaptive Group Dictionary Learning

In this subsection, we will show how to learn the adaptive dictionary $\boldsymbol{D}_{G_k}$ for each group $\boldsymbol{x}_{G_k}$. Note that, on one hand, we hope that each $\boldsymbol{x}_{G_k}$ can be faithfully represented by $\boldsymbol{D}_{G_k}$. On the other hand, it is expected that the representation coefficient vector of $\boldsymbol{x}_{G_k}$ over $\boldsymbol{D}_{G_k}$ is as sparse as possible. Like traditional patch-based dictionary learning algorithm in Eq. (8), the adaptive dictionary learning of group can be intuitively formulated as:

$$\underset{\boldsymbol{D}_{\boldsymbol{x}},\{\boldsymbol{\alpha}_{G_k}\}}{\operatorname{argmin}} \sum\nolimits_{k=1}^{n} \left\|\boldsymbol{x}_{G_k} - \boldsymbol{D}_{\boldsymbol{x}}\boldsymbol{\alpha}_{G_k}\right\|_F^2 + \lambda \sum\nolimits_{k=1}^{n} \left\|\boldsymbol{\alpha}_{G_k}\right\|_p, \tag{13}$$

where p is 0 or 1. Eq. (13) is a joint optimization problem of $\boldsymbol{D}_{\boldsymbol{x}}$ and $\{\boldsymbol{\alpha}_{G_k}\}$, which can be solved by alternatively optimizing $\boldsymbol{D}_{\boldsymbol{x}}$ and $\{\boldsymbol{\alpha}_{G_k}\}$.

Nevertheless, we do not directly utilize Eq. (13) to learn the dictionary for each group $\boldsymbol{x}_{G_k}$ based on the following three considerations. First, solving the joint optimization in Eq. (13) requires much computational cost, especially in the unit of group. Second, the learnt dictionary from Eq. (13) is actually adaptive for a given image $\boldsymbol{x}$, not adaptive for a group $\boldsymbol{x}_{G_k}$, which means that all the groups $\{\boldsymbol{x}_{G_k}\}$ are represented by the same dictionary $\boldsymbol{D}_{\boldsymbol{x}}$. That's why the dictionary learnt by Eq. (13) here is denoted by $\boldsymbol{D}_{\boldsymbol{x}}$, rather than $\{\boldsymbol{D}_{G_k}\}$. Finally, the dictionary learning process in Eq. (13) neglects the characteristics of each group $\boldsymbol{x}_{G_k}$, which contains patches with similar patterns. That is to say it is not necessary to learn an over-complete dictionary, and it is even possible to learn a dictionary by a more efficient and effective manner.

Similar to the idea of dictionary learning strategy using similar patches in [3], we propose to learn the adaptive dictionary $\boldsymbol{D}_{G_k}$ for each group $\boldsymbol{x}_{G_k}$ directly from its estimate $\boldsymbol{r}_{G_k}$, because in practice the original image $\boldsymbol{x}$ is not available for learning all the groups' dictionaries $\{\boldsymbol{D}_{G_k}\}$. The estimate $\boldsymbol{r}_{G_k}$ will be naturally selected in the process of optimization, which will be explained in the following sections.

After obtaining $\boldsymbol{r}_{G_k}$, we then apply SVD to it, that is,

$$\boldsymbol{r}_{G_k} = \boldsymbol{U}_{G_k}\boldsymbol{\Sigma}_{G_k}\boldsymbol{V}_{G_k}^T = \sum\nolimits_{i=1}^{m} \gamma_{\boldsymbol{r}_{G_k \otimes i}} (\boldsymbol{u}_{G_k \otimes i}\boldsymbol{v}_{G_k \otimes i}^T), \tag{14}$$

where $\gamma_{\boldsymbol{r}_{G_k}} = [\gamma_{\boldsymbol{r}_{G_k \otimes 1}}; \gamma_{\boldsymbol{r}_{G_k \otimes 2}}; \ldots; \gamma_{\boldsymbol{r}_{G_k \otimes m}}]$, $\boldsymbol{\Sigma}_{G_k} = \operatorname{diag}(\gamma_{\boldsymbol{r}_{G_k}})$ is a diagonal matrix with the elements of $\gamma_{\boldsymbol{r}_{G_k}}$ on its main diagonal, and $\boldsymbol{u}_{G_k \otimes i}, \boldsymbol{v}_{G_k \otimes i}$ are the columns of $\boldsymbol{U}_{G_k}$ and $\boldsymbol{V}_{G_k}$, separately. Each atom in $\boldsymbol{D}_{G_k}$ for group $\boldsymbol{x}_{G_k}$, is defined as

$$\boldsymbol{d}_{\boldsymbol{G}_k \otimes i} = \boldsymbol{u}_{\boldsymbol{G}_k \otimes i} \boldsymbol{v}_{\boldsymbol{G}_k \otimes i}^{T}, \;\; i = 1, 2, ..., m \,, \tag{15}$$

where $\boldsymbol{d}_{\boldsymbol{G}_k \otimes i} \in \mathbb{R}^{\mathcal{B}_s \times c}$. Therefore, the ultimate adaptively learned dictionary for $\boldsymbol{x}_{\boldsymbol{G}_k}$ is defined as

$$\boldsymbol{D}_{\boldsymbol{G}_k} = [\boldsymbol{d}_{\boldsymbol{G}_k \otimes 1}, \boldsymbol{d}_{\boldsymbol{G}_k \otimes 2}, ..., \boldsymbol{d}_{\boldsymbol{G}_k \otimes m}]. \tag{16}$$

According to the above definitions, the main difference between [3] and this paper for dictionary learning is that we utilize SVD to learn an adaptive dictionary for each group, while [3] utilizes PCA to learn an adaptive dictionary for each patch. The advantage of our proposed dictionary learning for each group is that it can guarantee all the patches in each group use the same dictionary and share the same dictionary atoms, which is more effective and robust, while [3] just trained the dictionary for each patch independently using its similar patches. It is clear to see that the proposed group dictionary learning is self-adaptive to each group $\boldsymbol{x}_{\boldsymbol{G}_k}$ and is quite efficient, requiring only one SVD for each group. In addition, owing to unitary property of $\boldsymbol{D}_{\boldsymbol{G}_k}$, the sparse coding process is not only efficient, but also stable and precise, which will be seen in the next section.

D. Discussions

This subsection will provide the detailed discussions about the close relationships among the proposed GSR model, the group sparsity model, and the low rank model. In fact, all the three models are involved with a set of similar patches to exploit the self-similarity of natural images.

As illustrated in Fig. 1, the proposed GSR model aims to adaptively seek the sparse representation of natural images in the unit of the group $\boldsymbol{x}_{\boldsymbol{G}_k}$. The group sparsity model imposes that similar patches in $\boldsymbol{x}_{\boldsymbol{G}_k}$ should share the same dictionary elements in their sparse decomposition. The low rank model hopes to find a low rank approximation of $\boldsymbol{x}_{\boldsymbol{G}_k}$ in order to find the robust estimate. These three models seem different at first glance, since they start from different views. However, interestingly, with the aid of the particular dictionary learning method by SVD, one type of the group sparsity model, and one type of the low rank model can be derived from our proposed GSR model, respectively. That means these three models are equivalent to some extent, which is of great help to understand these three models integrally. The details are given below.

As shown in Fig. 1, for each group $\boldsymbol{x}_{\boldsymbol{G}_k}$, given its noisy estimate $\boldsymbol{r}_{\boldsymbol{G}_k}$, the proposed GSR model to estimate $\boldsymbol{x}_{\boldsymbol{G}_k}$ such that $\boldsymbol{x}_{\boldsymbol{G}_k} = \boldsymbol{D}_{\boldsymbol{G}_k} \boldsymbol{\alpha}_{\boldsymbol{G}_k}$ is formulated as

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}_k} = \operatorname{argmin}_{\boldsymbol{\alpha}_{\boldsymbol{G}_k}} \frac{1}{2} \left\| \boldsymbol{r}_{\boldsymbol{G}_k} - \boldsymbol{D}_{\boldsymbol{G}_k} \boldsymbol{\alpha}_{\boldsymbol{G}_k} \right\|_F^2 + \lambda \left\| \boldsymbol{\alpha}_{\boldsymbol{G}_k} \right\|_0. \tag{17}$$

With $\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}_k}$, the reconstructed group is then expressed by $\hat{\boldsymbol{x}}_{\boldsymbol{G}_k} = \boldsymbol{D}_{\boldsymbol{G}_k} \hat{\boldsymbol{\alpha}}_{\boldsymbol{G}_k}$.

Assume $\boldsymbol{x}_{G_k} = \boldsymbol{DA}$, where $\boldsymbol{D}\in\mathbb{R}^{\mathcal{B}_s\times m}$ is the dictionary to sparsely represent all the patches in $\boldsymbol{x}_{G_k}$, and $\boldsymbol{A}\in\mathbb{R}^{m\times c}$ denotes the coefficient matrix. Here, set $\boldsymbol{D}$ to be $\boldsymbol{U}_{G_k}$ in Eq. (14), and in the light of all the definitions above, Eq. (17) is equivalent to the following form:

$$\hat{\boldsymbol{A}} = \operatorname{argmin}_{\boldsymbol{A}} \frac{1}{2}\left\|\boldsymbol{r}_{G_k} - \boldsymbol{DA}\right\|_F^2 + \lambda\left\|\boldsymbol{A}\right\|_{0,\infty}, \tag{18}$$

where $\|\bullet\|_{0,\infty}$ denotes the number of the nonzero rows of a matrix and is a pseudo norm [12] [29]. With $\hat{\boldsymbol{A}}$, we get $\hat{\boldsymbol{x}}_{G_k} = \boldsymbol{D}\hat{\boldsymbol{A}}$.

Due to the definition of the group sparsity model [5], [12], [29], one can see that Eq. (18) is just the special case of the group sparsity model with the constraint of the $\ell_{0,\infty}$ matrix norm, which differs from the previous group sparsity models with the constraint of the $\ell_{1,2}$ matrix norm [12] [5] .

Similarly, define $\boldsymbol{\gamma}_{\boldsymbol{x}_{G_k}}$ the vector composed of all the singular values of $\boldsymbol{x}_{G_k}$, i.e., $\gamma_{\boldsymbol{x}_{G_k}} = [\gamma_{\boldsymbol{x}_{G_k}\otimes 1}; \gamma_{\boldsymbol{x}_{G_k}\otimes 2}; \ldots; \gamma_{\boldsymbol{x}_{G_k}\otimes m}]$, . Due to $\boldsymbol{x}_{G_k} = \boldsymbol{D}_{G_k}\boldsymbol{\alpha}_{G_k}$ and the definitions of $\boldsymbol{D}_{G_k}$, we obtain

$$\|\boldsymbol{\gamma}_{\boldsymbol{x}_{G_k}}\|_0 = \operatorname{rank}(\boldsymbol{x}_{G_k}) = \|\boldsymbol{\alpha}_{G_k}\|_0, \tag{19}$$

where $\operatorname{rank}(\cdot)$ represents the rank of a matrix. Therefore, the following equation can be derived from Eq. (17):

$$\hat{\boldsymbol{x}}_{G_k} = \operatorname{argmin}_{\boldsymbol{x}_{G_k}} \frac{1}{2}\left\|\boldsymbol{x}_{G_k} - \boldsymbol{r}_{G_k}\right\|_F^2 + \lambda\|\boldsymbol{\gamma}_{\boldsymbol{x}_{G_k}}\|_0, \tag{20}$$

which is just the low rank model with the ℓ_0 norm of the vector composed of all the singular values of $\boldsymbol{x}_{G_k}$ and differs from previous low rank models with the ℓ_1 norm of the singular values vector [50] [51].

IV. Optimization for GSR-driven ℓ_0 minimization

In this section, an efficient approach is developed to solve the proposed GSR-driven ℓ_0 minimization for image restoration in Eq. (12), which is one of our main contributions.

$$\hat{\boldsymbol{\alpha}}_G = \operatorname{argmin}_{\boldsymbol{\alpha}_G} \frac{1}{2}\left\|\boldsymbol{HD}_G \circ \boldsymbol{\alpha}_G - \boldsymbol{y}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}_G\right\|_0. \tag{12}$$

Since ℓ_0 minimization is non-convex and NP-hard, the usual routine is to solve its optimal convex approximation, i.e., ℓ_1 minimization, which has been proved that, under some conditions, ℓ_1 minimization is equivalent to ℓ_0 minimization in a technical sense. The ℓ_1 minimization can be solved efficiently by some recent convex optimization algorithms, such as iterative shrinkage/thresholding [16], [17], split Bregman algorithms [43]. Therefore, the straightforward method to solve Eq. (12) is translated into solving its ℓ_1 convex form, that is

$$\hat{\boldsymbol{\alpha}}_G = \operatorname{argmin}_{\boldsymbol{\alpha}_G} \frac{1}{2}\left\|\boldsymbol{H}\boldsymbol{D}_G \circ \boldsymbol{\alpha}_G - \boldsymbol{y}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}_G\right\|_1. \quad (21)$$

However, a fact that is often neglected is, for some practical problems including image inverse problems, the conditions describing the equivalence of ℓ_0 minimization and ℓ_1 minimization are not necessarily satisfied. Therefore, this paper attempts to exploit the framework of convex optimization algorithms to solve the ℓ_0 minimization. Experimental results demonstrate the effectiveness and the convergence of our proposed approach. The superiority of solving Eq. (12) over solving Eq. (21) is further discussed in the experimental section.

In this paper, we adopt the framework of split Bregman iteration (SBI) [43] to solve Eq. (12), which is verified to be more effective than iterative shrinkage/thresholding (IST) [16] in our experiments (See Section V for more details).

First of all, let's make a brief review of SBI. The SBI algorithm was first proposed in [43] and was shown to be powerful in for solving various variational models [43], [44], [49]. Consider a constrained optimization problem

$$\min_{\boldsymbol{u}\in\mathbb{R}^N, \boldsymbol{v}\in\mathbb{R}^M} f(\boldsymbol{u}) + g(\boldsymbol{v}), \text{ s.t. } \boldsymbol{u} = \boldsymbol{G}\boldsymbol{v}, \quad (22)$$

where $\boldsymbol{G}\in\mathbb{R}^{M\times N}$ and $f:\mathbb{R}^N \to \mathbb{R}$, $g:\mathbb{R}^M \to \mathbb{R}$ are convex functions. The SBI to address problem (22) works as follows:

Algorithm 1 *Split Bregman Iteration (SBI)*

```
1.    Set $t = 0$, choose $\mu > 0$, $\boldsymbol{b}_0 = \boldsymbol{0}, \boldsymbol{u}_0 = \boldsymbol{0}, \boldsymbol{v}_0 = \boldsymbol{0}$.
2.    Repeat
3.        $\boldsymbol{u}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{u}} f(\boldsymbol{u}) + \frac{\mu}{2}\|\boldsymbol{u} - \boldsymbol{G}\boldsymbol{v}^{(t)} - \boldsymbol{b}^{(t)}\|_2^2$;
4.        $\boldsymbol{v}^{(t+1)} = \operatorname{argmin}_{\boldsymbol{v}} g(\boldsymbol{v}) + \frac{\mu}{2}\|\boldsymbol{u}^{(t+1)} - \boldsymbol{G}\boldsymbol{v} - \boldsymbol{b}^{(t)}\|_2^2$;
5.        $\boldsymbol{b}^{(t+1)} = \boldsymbol{b}^{(t)} - (\boldsymbol{u}^{(t+1)} - \boldsymbol{G}\boldsymbol{v}^{(t+1)})$;
6.        $t \leftarrow t + 1$;
7.    Until stopping criterion is satisfied
```

In SBI, the parameter μ is fixed to avoid the problem of numerical instabilities instead of choosing a predefined sequence that tends to infinity as done in [30]. According to SBI, the original minimization (22) is split into two sub-problems. The rationale behind is that each sub-problem minimization may be much easier than the original problem (22).

Now, let us go back to Eq. (12) and point out how to apply the framework of SBI to solve it. By introducing a variable $\boldsymbol{u}$, we first transform Eq. (12) into an equivalent constrained form,

$$\min_{\boldsymbol{\alpha}_G, \boldsymbol{u}} \frac{1}{2}\left\|\boldsymbol{H}\boldsymbol{u} - \boldsymbol{y}\right\|_2^2 + \lambda\left\|\boldsymbol{\alpha}_G\right\|_0, \text{ s.t. } \boldsymbol{u} = \boldsymbol{D}_G \circ \boldsymbol{\alpha}_G. \quad (23)$$

Define $f(\boldsymbol{u})=\frac{1}{2}\|\boldsymbol{H}\boldsymbol{u}-\boldsymbol{y}\|_2^2$, $g(\boldsymbol{\alpha}_G)=\lambda\|\boldsymbol{\alpha}_G\|_0$.

Then, invoking SBI, Line 3 in **Algorithm 1** becomes:

$$\boldsymbol{u}^{(t+1)}=\underset{\boldsymbol{u}}{\operatorname{argmin}}\frac{1}{2}\|\boldsymbol{H}\boldsymbol{u}-\boldsymbol{y}\|_2^2+\frac{\mu}{2}\|\boldsymbol{u}-\boldsymbol{D}_G\circ\boldsymbol{\alpha}_G^{(t)}-\boldsymbol{b}^{(t)}\|_2^2. \tag{24}$$

Next, Line 4 in **Algorithm 1** becomes:

$$\boldsymbol{\alpha}_G^{(t+1)}=\operatorname{argmin}_{\boldsymbol{\alpha}_G}\lambda\|\boldsymbol{\alpha}_G\|_0+\frac{\mu}{2}\|\boldsymbol{u}^{(t+1)}-\boldsymbol{D}_G\circ\boldsymbol{\alpha}_G-\boldsymbol{b}^{(t)}\|_2^2. \tag{25}$$

According to Line 5 in **Algorithm 1**, the update of $\boldsymbol{b}^{(t)}$ is

$$\boldsymbol{b}^{(t+1)}=\boldsymbol{b}^{(t)}-(\boldsymbol{u}^{(t+1)}-\boldsymbol{D}_G\circ\boldsymbol{\alpha}_G^{(t+1)}). \tag{26}$$

Thus, by SBI, the minimization for Eq. (12) is transformed into solving two sub-problems, namely, $\boldsymbol{u}$, $\boldsymbol{\alpha}_G$ sub-problems. In the following, we will provide the implementation details to obtain the efficient solutions to each separated sub-problem. For simplicity, the subscript t is omitted without confusion.

A. $\boldsymbol{u}$ Sub-problem

Given $\boldsymbol{\alpha}_G$, the $\boldsymbol{u}$ sub-problem denoted by Eq. (24) is essentially a minimization problem of strictly convex quadratic function, that is

$$\min_{\boldsymbol{u}}Q_1(\boldsymbol{u})=\min_{\boldsymbol{u}}\frac{1}{2}\|\boldsymbol{H}\boldsymbol{u}-\boldsymbol{y}\|_2^2+\frac{\mu}{2}\|\boldsymbol{u}-\boldsymbol{D}_G\circ\boldsymbol{\alpha}_G-\boldsymbol{b}\|_2^2. \tag{27}$$

Setting the gradient of $Q_1(\boldsymbol{u})$ to be zero gives a closed solution for Eq. (27), which can be expressed as

$$\hat{\boldsymbol{u}}=(\boldsymbol{H}^T\boldsymbol{H}+\mu\boldsymbol{I})^{-1}\boldsymbol{q}, \tag{28}$$

where $\boldsymbol{q}=\boldsymbol{H}^T\boldsymbol{y}+\mu(\boldsymbol{D}_G\circ\boldsymbol{\alpha}_G+\boldsymbol{b})$, $\boldsymbol{I}$ is identity matrix.

As for image inpainting and image deblurring, due to the special structure of $\boldsymbol{H}$, Eq. (28) can be computed efficiently without computing the matrix inverse (more details can be found in [18]).

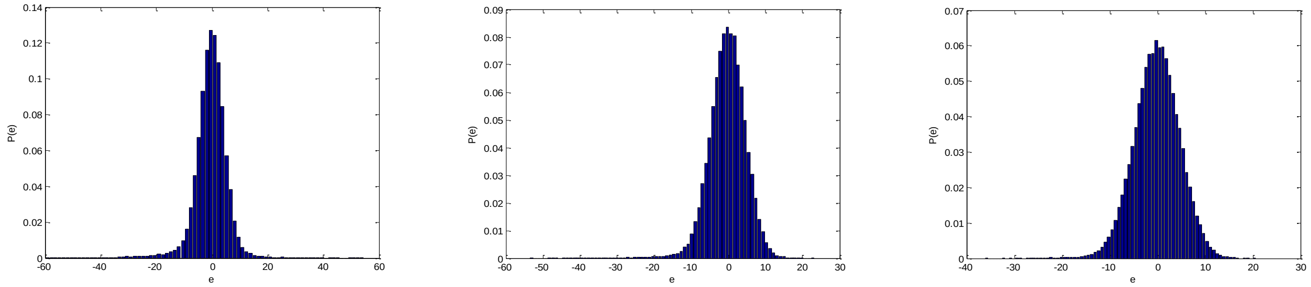
As for image compressive sensing (CS) recovery, $\boldsymbol{H}$ is a random projection matrix without special structure. Thus, it is too costly to solve Eq. (27) directly by Eq. (28). Here, to avoid computing the matrix inverse, the gradient descent method is utilized to solve Eq. (27) by applying

$$\hat{\boldsymbol{u}}=\boldsymbol{u}-\eta\boldsymbol{d}, \tag{29}$$

where $\boldsymbol{d}$ is the gradient direction of the objective function $Q_1(\boldsymbol{u})$ and η represents the optimal step. Therefore, solving $\boldsymbol{u}$ sub-problem for image CS recovery only requires computing the following equation iteratively

$$\hat{\boldsymbol{u}} = \boldsymbol{u} - \eta\left(\boldsymbol{H}^T\boldsymbol{H}\boldsymbol{u} - \boldsymbol{H}^T\boldsymbol{y} + \mu\left(\boldsymbol{u} - \boldsymbol{D}_G \circ \boldsymbol{\alpha}_G - \boldsymbol{b}\right)\right), \tag{30}$$

where $\boldsymbol{H}^T\boldsymbol{H}$ and $\boldsymbol{H}^T\boldsymbol{y}$ can be calculated before, making above computation more efficient.



(a) (b) (c)

Figure 3: The distribution of $\boldsymbol{e}^{(t)}$ and its corresponding variance $\mathrm{Var}(\boldsymbol{e}^{(t)})$ for image *Parrots* in the case of image deblurring at different iterations. (a) t = 3 and $\mathrm{Var}(\boldsymbol{e}^{(3)}) = 25.70$; (b) t = 5 and $\mathrm{Var}(\boldsymbol{e}^{(5)}) = 23.40$; (c) t = 7 and $\mathrm{Var}(\boldsymbol{e}^{(7)}) = 23.16$.

B. $\boldsymbol{\alpha}_G$ Sub-problem

Given $\boldsymbol{u}$, according to Eq. (25), the $\boldsymbol{\alpha}_G$ sub-problem can be formulated as

$$\min_{\boldsymbol{\alpha}_G} Q_2(\boldsymbol{\alpha}_G) = \min_{\boldsymbol{\alpha}_G} \frac{1}{2}\left\|\boldsymbol{D}_G \circ \boldsymbol{\alpha}_G - \boldsymbol{r}\right\|_2^2 + \frac{\lambda}{\mu}\left\|\boldsymbol{\alpha}_G\right\|_0, \tag{31}$$

where $\boldsymbol{r} = \boldsymbol{u} - \boldsymbol{b}$.

Note that it is difficult to solve Eq. (31) directly due to the complicated definition of $\boldsymbol{\alpha}_G$. Instead, we make some transformation. Let $\boldsymbol{x} = \boldsymbol{D}_G \circ \boldsymbol{\alpha}_G$, then Eq. (31) equally becomes

$$\min_{\boldsymbol{\alpha}_G} \frac{1}{2}\left\|\boldsymbol{x} - \boldsymbol{r}\right\|_2^2 + \frac{\lambda}{\mu}\left\|\boldsymbol{\alpha}_G\right\|_0. \tag{32}$$

By viewing $\boldsymbol{r}$ as some type of noisy observation of $\boldsymbol{x}$, we perform some experiments to investigate the statistics of $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{r}$. Here, we use image *Parrots* as an example in the case of image deblurring, where the original image is first blurred by uniform blur kernel and then is added by Gaussian white noise of standard deviation 0.5. At each iteration t, we can obtain $\boldsymbol{r}^{(t)}$ by $\boldsymbol{r}^{(t)} = \boldsymbol{u}^{(t)} - \boldsymbol{b}^{(t\text{-}1)}$. Since the exact minimizer of Eq. (32) is not available, we then approximate $\boldsymbol{x}^{(t)}$ by the original image without generality. Therefore, we are able to acquire the histogram of $\boldsymbol{e}^{(t)} = \boldsymbol{x}^{(t)} - \boldsymbol{r}^{(t)}$ at each iteration t. Fig. 3 shows the distributions of $\boldsymbol{e}^{(t)}$ when t equals to 3, 5 and 7, respectively.

From Fig. 3, it is obvious to observe that the distribution of $\boldsymbol{e}^{(t)}$ at each iteration is quite suitable to be characterized by generalized Gaussian distribution (GGD) [39] with zero-mean and variance $\mathrm{Var}(\boldsymbol{e}^{(t)})$. The variance $\mathrm{Var}(\boldsymbol{e}^{(t)})$ can be estimated by

$$\mathrm{Var}(\boldsymbol{e}^{(t)})=\frac{1}{N}\left\|\boldsymbol{x}^{(t)}-\boldsymbol{r}^{(t)}\right\|_2^2. \tag{33}$$

Fig. 3 also gives the corresponding estimated variances at different iterations. Furthermore, owing that the residual of images is usually de-correlated, each element of $\boldsymbol{e}^{(t)}$ can be modeled independently.

Accordingly, to enable solving Eq. (32) tractable, in this paper, a reasonable assumption is made, with which even a closed-form solution of Eq. (32) can be obtained. We suppose that each element of $\boldsymbol{e}^{(t)}$ follows an independent zero-mean distribution with variance $\mathrm{Var}(\boldsymbol{e}^{(t)})$. It is worth emphasizing that the above assumption does not need to be Gaussian, or Laplacian, or GGD process, which is more general. By this assumption, we can prove the following conclusion.

THEOREM 1. *Let $\boldsymbol{x},\boldsymbol{r}\in\mathbb{R}^N$, $\boldsymbol{x}_{G_k},\boldsymbol{r}_{G_k}\in\mathbb{R}^{B_S\times c}$, and denote the error vector by $\boldsymbol{e}=\boldsymbol{x}-\boldsymbol{r}$ and each element of $\boldsymbol{e}$ by $\boldsymbol{e}(j)$, $j=1,...,N$. Assume that $\boldsymbol{e}(j)$ is independent and comes from a distribution with zero mean and variance σ^2. Then, for any $\varepsilon>0$, we have the following property to describe the relationship between $\|\boldsymbol{x}-\boldsymbol{r}\|_2^2$ and $\sum_{k=1}^n\left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2$, that is,*

$$\lim_{\substack{N\to\infty\\K\to\infty}}P\left\{\left|\frac{1}{N}\|\boldsymbol{x}-\boldsymbol{r}\|_2^2-\frac{1}{K}\sum_{k=1}^n\left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2\right|<\varepsilon\right\}=1, \tag{34}$$

here $P(\cdot)$ represents the probability and $K=B_s\times c\times n$ (See Appendix A for detailed proof).

According to **Theorem 1**, there exists the following equation with very large probability (limited to 1) at each iteration t:

$$\frac{1}{N}\left\|\boldsymbol{x}^{(t)}-\boldsymbol{r}^{(t)}\right\|_2^2=\frac{1}{K}\sum_{k=1}^n\left\|\boldsymbol{x}_{G_k}^{(t)}-\boldsymbol{r}_{G_k}^{(t)}\right\|_F^2. \tag{35}$$

Now let's verify Eq. (35) by the above case of image deblurring. We can clearly see that the left hand of Eq. (35) is just $\mathrm{Var}(\boldsymbol{e}^{(t)})$ defined in Eq. (33), with $\mathrm{Var}(\boldsymbol{e}^{(3)})=25.70$, $\mathrm{Var}(\boldsymbol{e}^{(5)})=23.40$, and $\mathrm{Var}(\boldsymbol{e}^{(7)})=23.16$, which is shown in Fig. 3. At the same time, we can calculate the corresponding right hand of Eq. (35), denoted by $\mathrm{Var}(\boldsymbol{e}_G^{(t)})$, with the same values of t, leading to $\mathrm{Var}(\boldsymbol{e}_G^{(3)})=25.21$, $\mathrm{Var}(\boldsymbol{e}_G^{(5)})=23.15$, and $\mathrm{Var}(\boldsymbol{e}_G^{(7)})=23.07$. Apparently, at each iteration, $\mathrm{Var}(\boldsymbol{e}^{(t)})$ is very close to $\mathrm{Var}(\boldsymbol{e}_G^{(t)})$, especially when t is larger, which sufficiently illustrates the validity of our assumption.

Next, by incorporating Eq. (35) into Eq. (32), it yields

$$\begin{aligned}&\min_{\boldsymbol{\alpha}_G}\frac{1}{2}\sum_{k=1}^n\left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2+\frac{\lambda K}{\mu N}\left\|\boldsymbol{\alpha}_G\right\|_0\\&=\min_{\boldsymbol{\alpha}_G}\frac{1}{2}\sum_{k=1}^n\left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2+\frac{\lambda K}{\mu N}\sum_{k=1}^n\left\|\boldsymbol{\alpha}_{G_k}\right\|_0\\&=\min_{\boldsymbol{\alpha}_G}\sum_{k=1}^n\frac{1}{2}\left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2+\tau\left\|\boldsymbol{\alpha}_{G_k}\right\|_0,\end{aligned} \tag{36}$$

where $\tau=(\lambda K)/(\mu N)$.

It is obvious to see that Eq. (36) can be efficiently minimized by solving n sub-problems for all the groups $\boldsymbol{x}_{G_k}$. Each group based sub-problem is formulated as:

$$\begin{aligned} &\operatorname{argmin}_{\boldsymbol{\alpha}_{G_k}} \frac{1}{2}\left\|\boldsymbol{x}_{G_k} - \boldsymbol{r}_{G_k}\right\|_F^2 + \tau\left\|\boldsymbol{\alpha}_{G_k}\right\|_0 \\ &= \operatorname{argmin}_{\boldsymbol{\alpha}_{G_k}} \frac{1}{2}\left\|\boldsymbol{D}_{G_k}\boldsymbol{\alpha}_{G_k} - \boldsymbol{r}_{G_k}\right\|_F^2 + \tau\left\|\boldsymbol{\alpha}_{G_k}\right\|_0 \end{aligned}. \tag{37}$$

where $\boldsymbol{D}_{G_k}$ is the self-adaptive learned dictionary from $\boldsymbol{r}_{G_k}$ using our proposed scheme described in subsection III.C. Obviously, Eq. (37) can also be considered as the sparse coding problem. Now we will show that the accurate solution of Eq. (37) can be achieved efficiently. With the definitions of $\{\boldsymbol{\alpha}_{G_k}\}$ and $\{\boldsymbol{\gamma}_{G r_k}\}$, we get $\boldsymbol{x}_{G_k} = \boldsymbol{D}_{G_k}\boldsymbol{\alpha}_{G_k}$, $\boldsymbol{r}_{G_k} = \boldsymbol{D}_{G_k}\boldsymbol{\gamma}_{G r_k}$. Due to the unitary property of $\boldsymbol{D}_{G_k}$, we have

$$\left\|\boldsymbol{D}_{G_k}\boldsymbol{\alpha}_{G_k} - \boldsymbol{D}_{G_k}\boldsymbol{\gamma}_{r_{G_k}}\right\|_F^2 = \left\|\boldsymbol{\alpha}_{G_k} - \boldsymbol{\gamma}_{r_{G_k}}\right\|_2^2. \tag{38}$$

With Eq. (38), the sub-problem (37) is equivalent to

$$\operatorname{argmin}_{\boldsymbol{\alpha}_{G_k}} \frac{1}{2}\left\|\boldsymbol{\alpha}_{G_k} - \boldsymbol{\gamma}_{r_{G_k}}\right\|_2^2 + \tau\left\|\boldsymbol{\alpha}_{G_k}\right\|_0. \tag{39}$$

Therefore, the closed-form solution of (39) is expressed as

$$\hat{\boldsymbol{\alpha}}_{G_k} = \operatorname{hard}(\boldsymbol{\gamma}_{r_{G_k}}, \sqrt{2\tau}) = \boldsymbol{\gamma}_{r_{G_k}} \odot 1(\operatorname{abs}(\boldsymbol{\gamma}_{r_{G_k}}) - \sqrt{2\tau}), \tag{40}$$

where $\operatorname{hard}(\cdot)$ denotes the operator of hard thresholding and $\odot$ stands for the element-wise product of two vectors. This process is applied for all n groups to achieve $\hat{\boldsymbol{\alpha}}_G$, which is the final solution for $\boldsymbol{\alpha}_G$ sub-problem in Eq. (31).

C. Summary of Proposed Algorithm

So far, all issues in the process of handing the above two sub-problems have been solved. In fact, we acquire the efficient solution for each separated sub-problem, which enables the whole algorithm more efficient and effective. In light of all derivations above, a detailed description of the proposed algorithm for image restoration using GSR is provided in Table 1.

Table 1: A Complete Description of Proposed GSR Modeling for Image Restoration

```
Input: the observed image or measurement $\boldsymbol{y}$ and the degraded operator $\boldsymbol{H}$
Initialization: $t=0, \boldsymbol{b}^{(0)}=\boldsymbol{0}, \boldsymbol{\alpha}_{\boldsymbol{G}}^{(0)}=\boldsymbol{0}, \boldsymbol{u}^{(0)}, \mathcal{B}_S, c, \lambda, \mu, ;$
Repeat
    if $\boldsymbol{H}$ is mask operator
        Update $\boldsymbol{u}^{(t+1)}$ by Eq. (28);
    else if $\boldsymbol{H}$ is blur operator
        Update $\boldsymbol{u}^{(t+1)}$ by Eq. (28);
    else if $\boldsymbol{H}$ is random projection operator
        Update $\boldsymbol{u}^{(t+1)}$ by Eq. (30);
    end if
    $\boldsymbol{r}^{(t+1)}=\boldsymbol{u}^{(t+1)}-\boldsymbol{b}^{(t)}$; $\tau=(\lambda K)/(\mu N)$;
    for Each group $\boldsymbol{x}_{\boldsymbol{G}_k}$
        Construct dictionary $\boldsymbol{D}_{\boldsymbol{G}_k}$ by computing Eq. (16);
        Reconstruct $\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}_k}$ by computing Eq. (40);
    end for
    Update $\boldsymbol{D}_{\boldsymbol{G}}^{(t+1)}$ by concatenating all $\boldsymbol{D}_{\boldsymbol{G}_k}$;
    Update $\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}}^{(t+1)}$ by concatenating all $\hat{\boldsymbol{\alpha}}_{\boldsymbol{G}_k}$;
    Update $\boldsymbol{b}^{(t+1)}$ by computing Eq. (26);
    $t \leftarrow t+1$;
Until maximum iteration number is reached
Output: Final restored image $\hat{\boldsymbol{x}}=\boldsymbol{D}_G \circ \hat{\boldsymbol{\alpha}}_G$ .
```

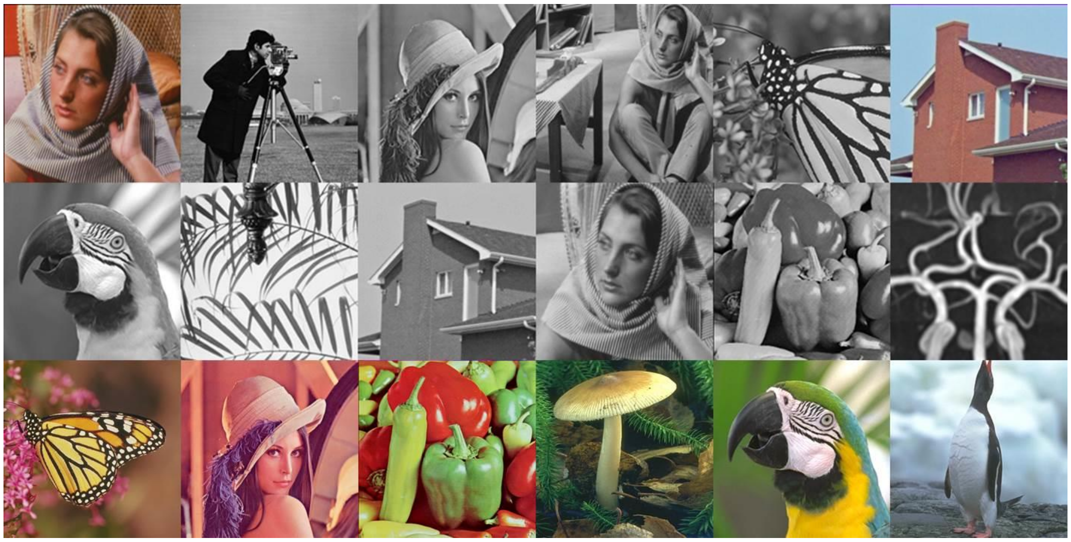


Figure 4: All experimental test images.

V. Experimental Results

In this section, extensive experimental results are conducted to verify the performance of the proposed GSR for image restoration applications, which include image inpainting, image deblurring and image compressive sensing recovery. The parameter setting of GSR is as follows: the size of a group is set to be 64×60, with $\mathcal{B}_S$ being 64 and c being 60. The width of overlapping between adjacent patches is 4 pixels, leading to the relationship $K=240N$. The range of training window for constructing group, i.e., $L\times L$ is set to be 40×40. The parameters μ and λ are set accordingly for different image restoration applications, which will be given below. All the experiments are performed in Matlab 7.12.0 on a Dell OPTIPLEX computer with Intel(R) Core(TM) 2 Duo CPU E8400 processor (3.00GHz), 3.25G memory, and Windows XP operating system.

To evaluate the quality of the reconstructed image, in addition to PSNR (Peak Signal to Noise Ratio, unit: dB), which is used to evaluate the objective image quality, a recently proposed powerful perceptual quality metric FSIM [45] is calculated to evaluate the visual quality. The higher FSIM value means the better visual quality. For color images, image restoration operations are only applied to the luminance component. All the experimental test images are given in Fig. 4. **Due to the limit of space, only parts of the experimental results are shown in this paper. Please enlarge and view the figures on the screen for better comparison.** Our Matlab code and all the experimental results can be downloaded at the website: http://idm.pku.edu.cn/staff/zhangjian/GSR/.

A. Image Inpainting

In this subsection, two interesting cases of image inpainting with different masks are considered, i.e., image restoration from partial random samples and text removal. For image inpainting application, $\mu=0.0025$ and $\lambda=0.082$.

The proposed GSR is compared with five recent representative methods for image inpainting: SKR (steering kernel regression) [6], NLTV [35], BPFA [48], HSR [10] and SAIST [5]. SKR [6] is a classic method that utilizes a steering kernel regression framework to characterize local structures for image restoration. NLTV [35] is an extension of traditional total variation (TV) with a nonlocal weight function. BPFA [48] employs a truncated beta-Bernoulli process to infer an appropriate dictionary for image recovery and exploits the spatial inter-relationships within imagery through the use of the Dirichlet and probit stick-breaking processes. HSR [10] combines the strength of local and nonlocal sparse representations under a systematic framework called Bayesian model averaging, characterizing local smoothness and nonlocal self-similarity simultaneously.

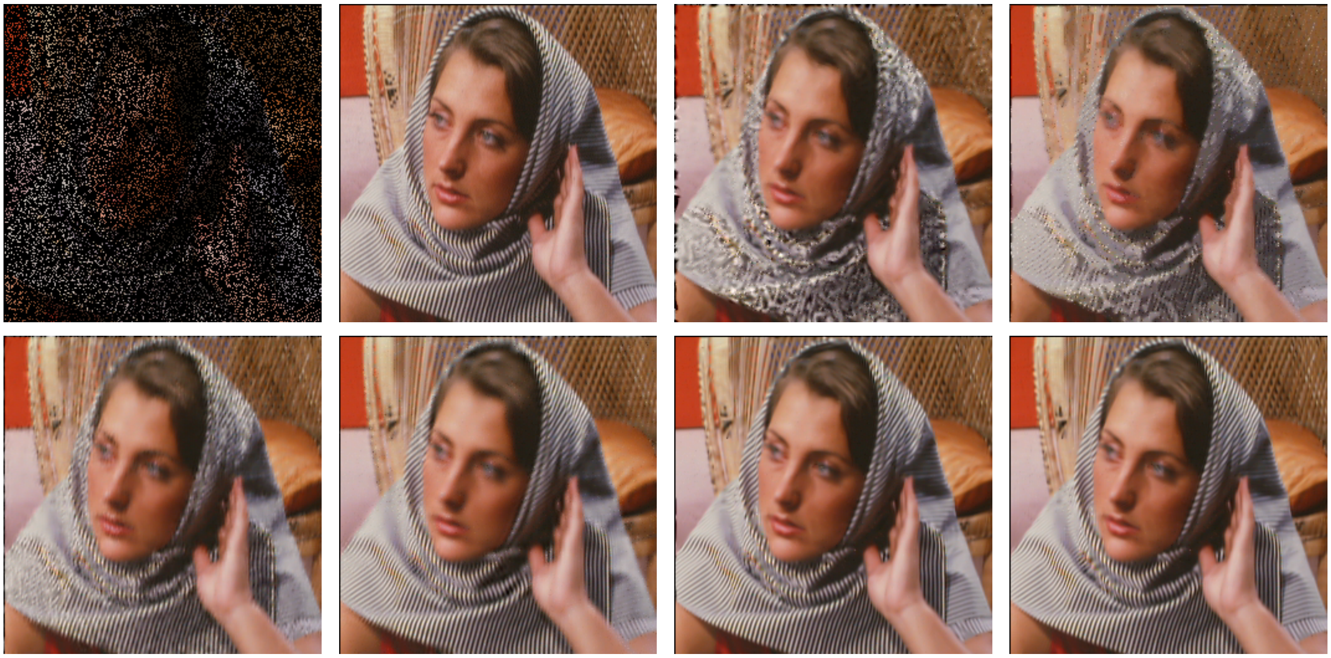


Figure 5: Visual quality comparison of image restoration from partial random samples for color image *Barbara*. From left to right and top to bottom: the degraded image with only 20% random samples available, original image, the recovered images by SKR [6] (PSNR=21.92dB; FSIM=0.8607), NLTV [35] (PSNR= 23.46dB; FSIM=0.8372), BPFA [48] (PSNR=25.70dB; FSIM=0.8926), HSR [10] (PSNR=28.83dB; FSIM=0.9273), SAIST [5] (PSNR=29.68dB; FSIM =0.9485) and the proposed GSR (PSNR=**31.32**dB; FSIM=**0.9598**).

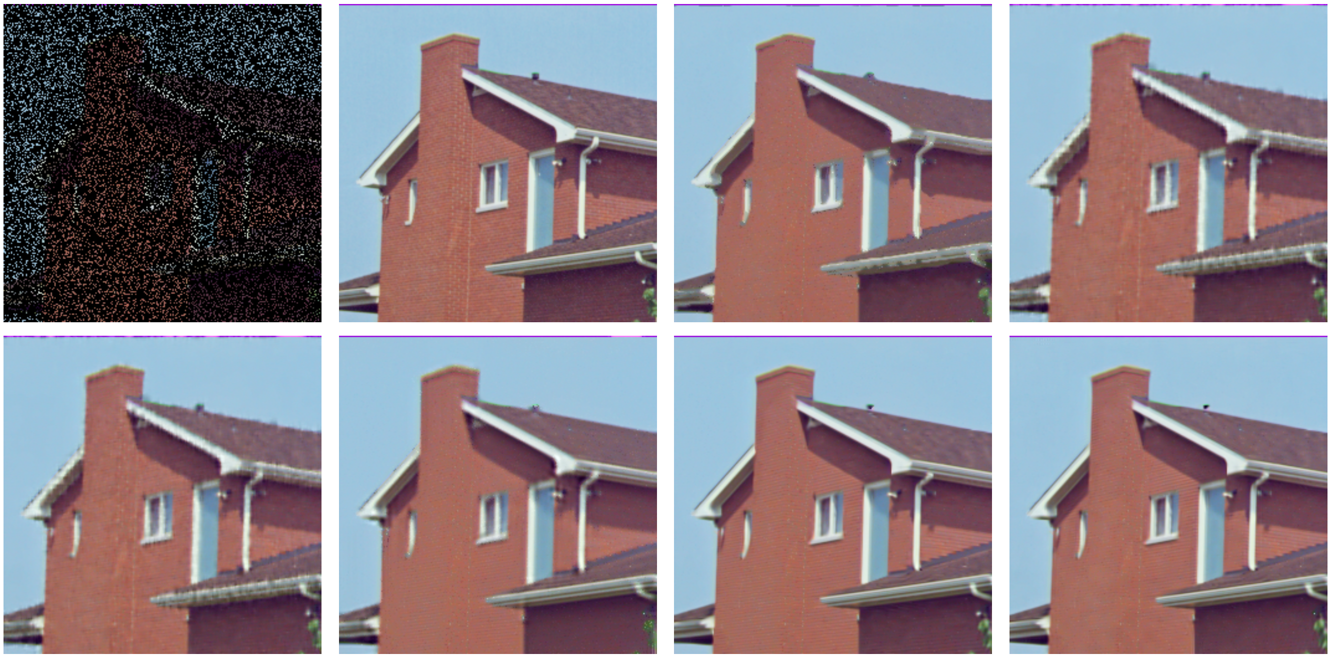


Figure 6: Visual quality comparison of image restoration from partial random samples for color image *House*. From left to right and top to bottom: the degraded image with only 20% random samples available, original image, the recovered images by SKR [6] (PSNR=30.40dB; FSIM=0.9198), NLTV [35] (PSNR=31.19dB; FSIM=0.9093), BPFA [48] (PSNR=30.89dB; FSIM=0.9111), HSR [10] (PSNR=32.35dB; FSIM=0.9255), SAIST [5] (PSNR=**35.73**dB; FSIM=**0.9615**) and the proposed GSR (PSNR=35.61dB; FSIM=0.9594).

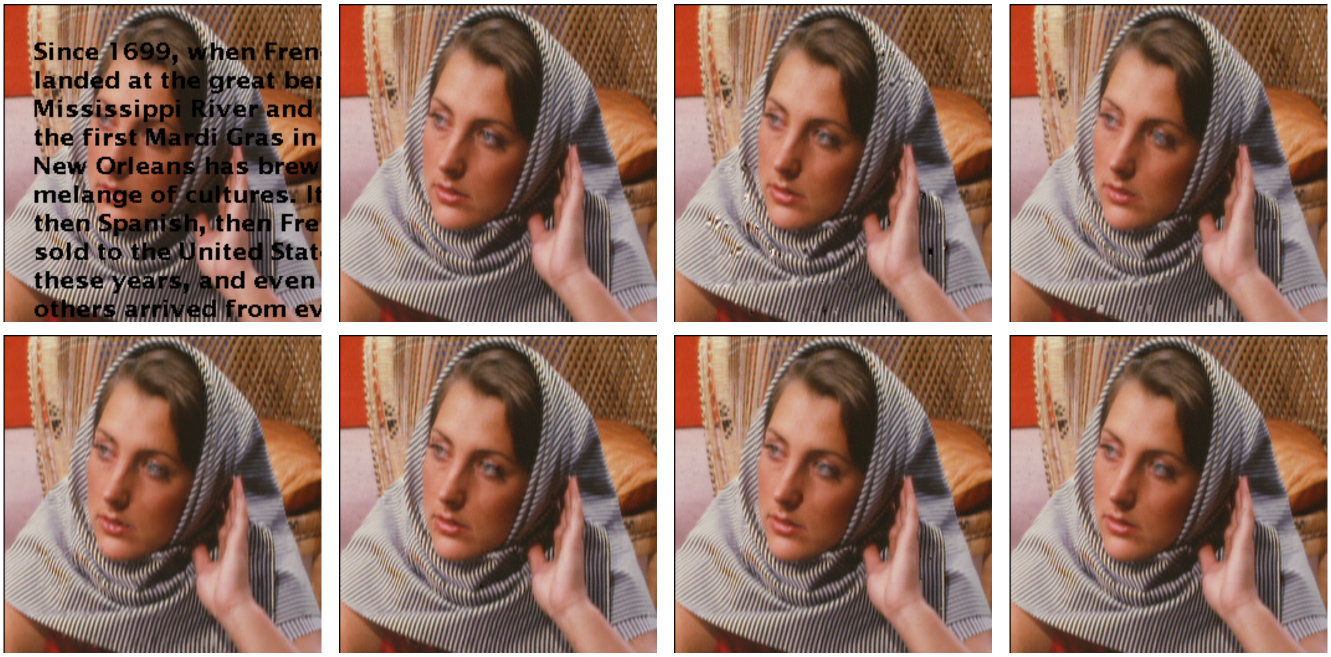


Figure 7: Visual quality comparison in the case of text removal for color image *Barbara*. From left to right and top to bottom: the masked image, original image, the recovered images by SKR [6] (PSNR=30.81dB; FSIM=0.9747), NLTV [35] (PSNR=32.60dB; FSIM=0.9749), BPFA [48] (PSNR=34.28dB; FSIM=0.9790), HSR [10] (PSNR=38.86dB; FSIM=0.9901), SAIST [5] (PSNR=39.00dB; FSIM=0.9915) and the proposed GSR (PSNR=**40.86**dB; FSIM=**0.9936**).

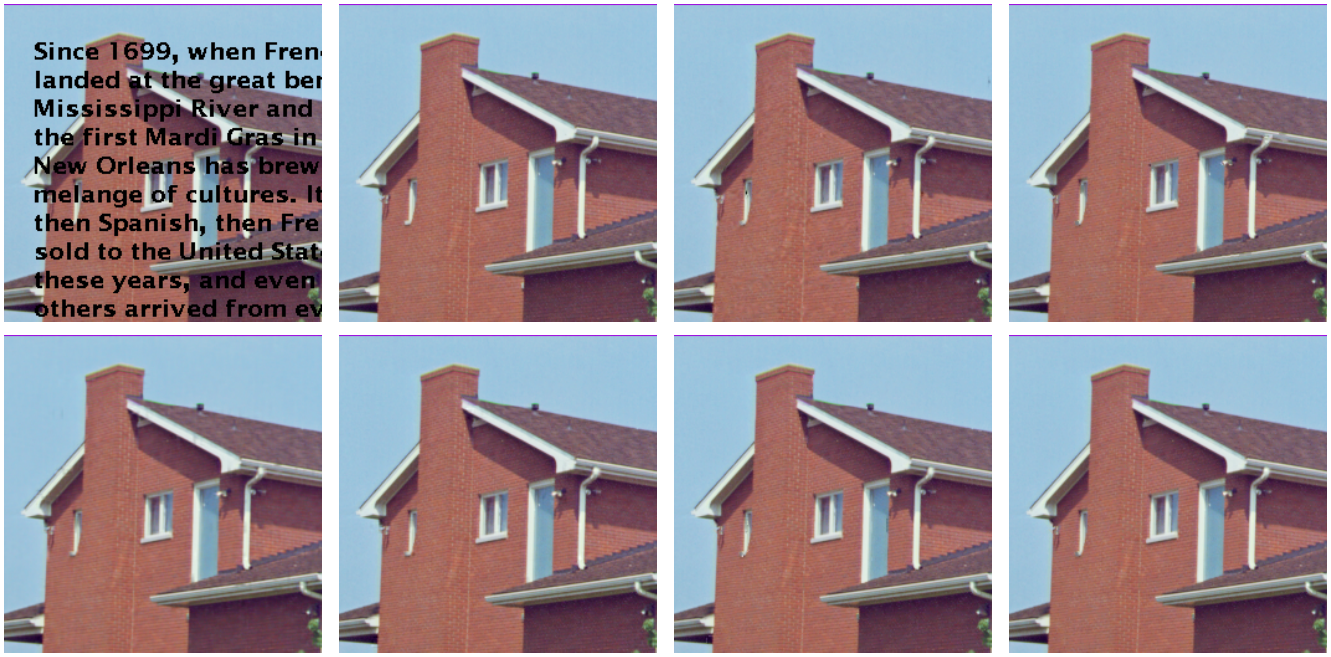


Figure 8: Visual quality comparison in the case of text removal for color image *House*. From left to right and top to bottom: the masked image, original image, the recovered images by SKR [6] (PSNR=38.65dB; FSIM=0.9850), NLTV [35] (PSNR=38.44dB; FSIM=0.9820), BPFA [48] (PSNR=39.01dB; FSIM=0.9818), HSR [10] (PSNR=42.06dB; FSIM=0.9913), SAIST [5] (PSNR=41.20dB; FSIM=0.9893) and the proposed GSR (PSNR=**42.51**dB; FSIM=**0.9916**).

The visual quality comparisons in the case of image restoration from only 20% random samples and in the case of text removal for two standard color test images are provided in Figs. 5~8. It is obvious to see that SKR and NLTV are good at capturing contour structures, but fail in recovering textures and produce blurred effects. BPFA is able to recover some textures, while generating some incorrect textures and some blurred effects due to less robustness with so small percentage of retaining samples for dictionary learning. HSR usually restores better textures than SKR, NLTV and BPFA. However, it often produces noticeable striped artifacts. We can observe that SAIST and the proposed GSR modeling can provide better restoration on both edges and textures than other competing methods. Concretely, for image *Barbara* which is full of textures, GSR achieves much better PSNR and FSIM than SAIST, with more image details and textures in both cases as shown in Fig. 5 and Fig. 7. For image *House* which is rich of edges, GSR achieves almost the same performance with SAIST in the case of image restoration from only 20% random sample (see Fig. 6), and achieves better result than SAIST in the case of text removal (see Fig. 8). Additional qualitative PSNR and FSIM results by our proposed GSR for image restoration from partial random samples on eight standard color images at different percentages of random samples are shown in Table 2.

Table 2: PSNR and FSIM results by GSR for image restoration from partial random samples at different percentages of random samples

Data Percentage	***Image***			
	Barbara	***House***	***Parrots***	***Lena***
20%	31.32/0.9598	35.61/0.9594	29.84/0.9530	34.12/0.9815
30%	34.42/0.9768	37.65/0.9745	33.31/0.9695	36.38/0.9893
50%	39.12/0.9906	41.61/0.9886	37.78/0.9853	39.83/0.9956
Data Percentage	***Image***			
	Butterfly	***Mushroom***	***Penguin***	***Peppers***
20%	30.31/0.9792	28.85/0.9020	33.23/0.9594	33.82/0.9811
30%	33.02/0.9888	30.87/0.9346	35.62/0.9732	35.47/0.9874
50%	37.26/0.9956	34.60/0.9702	39.28/0.9872	38.30/0.9941

B. Image Deblurring

In this subsection, two sets of experiments are conducted to verify the performance of the proposed GSR method for image deblurring. In the first set, two types of blur kernels, including a 9×9 uniform kernel and a Gaussian blur kernel, are exploited for simulation, with standard deviation of additive Gaussian noise $\sigma = \sqrt{2}$ (see Table 4). In the second set, six typical deblurring experiments (as shown in Table 3) with respect to four standard gray images, which have been presented in [14] [15] are provided.

The proposed GSR deblurring method is compared with four recently developed deblurring approaches, i.e., TVMM [9], L0_ABS [46], NCSR [15], and IDDBM3D [14]. TVMM [8] is a TV-based deblurring approach that can well reconstruct the piecewise smooth regions but often fails to recover fine image details. The L0_ABS [9] is a sparsity-based deblurring method exploiting a fixed sparse domain. IDDBM3D [10] method is an improved version of BM3D deblurring method [31]. NCSR proposed a centralized sparse constraint, which exploits the image nonlocal redundancy to reduce the sparse coding noise [15]. As far as we know, NCSR and IDDBM3D provide the current best image deblurring results in the literature.

Table 3: Six typical deblurring experiments with various blur PSFs and noise variances in the second set

Scenario	PSF	σ^2
1	$1/(1+z_1^2+z_2^2), z_1, z_2 = -7, \ldots, 7$	2
2	$1/(1+z_1^2+z_2^2), z_1, z_2 = -7, \ldots, 7$	8
3	9×9 uniform	≈ 0.3
4	$[1\ 4\ 6\ 4\ 1]^T[1\ 4\ 6\ 4\ 1]/256$	49
5	Gaussian with *std* = 1.6	4
6	Gaussian with *std* = 0.4	64

The PSNR and FSIM results on six gray test images in the first set of experiments are reported in Table 4. For the case of 9×9 uniform kernel with noise $\sigma = \sqrt{2}$, $\mu = 0.0075$ and $\lambda = 0.554$, and for the case of Gaussian kernel with noise $\sigma = \sqrt{2}$, $\mu = 0.0125$ and $\lambda = 0.41$.

Table 4: PSNR and FSIM Comparisons for Image Deblurring in the First Set

Image	***Barbara***	***Boats***	***House***	***C. Man***	***Peppers***	***Lena***	**Avg.**
Uniform Kernel: 9×9, $\sigma = \sqrt{2}$							
TVMM [9]	26.00/0.8538	29.39/0.8978	32.47/0.9134	26.83/0.8674	28.77/0.8983	28.68/0.8937	28.69/0.8874
L0_ABS [46]	26.41/0.8692	29.77/0.9071	33.01/0.9241	27.12/0.8729	28.68/0.9078	28.76/0.9056	28.96/0.8978
IDDBM3D [14]	27.98/0.9014	31.20/0.9304	34.44/0.9369	28.56/0.9007	29.62/0.9200	29.70/0.9197	30.25/0.9182
NCSR [15]	28.10/0.9117	31.08/0.9294	34.31/**0.9415**	**28.62**/**0.9026**	**29.66**/0.9220	29.96/0.9254	30.29/0.9221
GSR	**28.95**/**0.9227**	**31.34**/**0.9326**	**34.48**/0.9403	28.28/0.8937	**29.66**/**0.9231**	**30.10**/**0.9281**	**30.47**/**0.9234**
Gaussian Kernel: fspecial (Gaussian, 25, 1.6), $\sigma = \sqrt{2}$							
TVMM [9]	24.81/0.8435	30.44/0.9219	33.01/0.9139	27.04/0.8911	29.20/0.9237	30.72/0.9259	29.21/0.9034
L0_ABS [46]	24.78/0.8543	30.54/0.9297	33.07/0.9212	27.34/0.8955	28.88/0.9303	30.63/0.9361	29.21/0.9112
IDDBM3D [14]	27.19/0.8986	31.68/**0.9426**	34.08/0.9359	28.17/**0.9136**	29.99/**0.9373**	31.45/0.9430	30.43/0.9285
NCSR [15]	27.91/0.9088	31.49/0.9371	33.63/0.9333	**28.34**/0.9078	30.16/0.9331	31.26/0.9389	30.47/0.9265
GSR	**28.26**/**0.9155**	**31.69**/0.9411	**34.45**/**0.9420**	27.78/0.9006	**30.19**/0.9349	**31.47**/**0.9463**	**30.64**/**0.9301**

From Table 4, we can see that the proposed GSR achieves highly competitive performance compared with other leading deblurring methods. L0_ABS produces slightly higher average PSNR and FSIM than TVMM, while GSR outperforms L0_ABS by 1.5 dB and 1.4 dB for the uniform blur and Gaussian blur, respectively. One can observe that IDDBM3D, NCSR and GSR produce very similar results, and obtain significant PSNR/FSIM improvements over other competing methods. In average, GSR outperforms IDDBM3D and NCSR by (0.22 dB, 0.21 dB) and (0.18 dB, 0.17 dB) for the two blur kernels, respectively. The visual comparisons of the deblurring methods are shown in Figs. 9~10, from which one can observe that the GSR model produces cleaner and sharper image edges and textures than other competing methods.

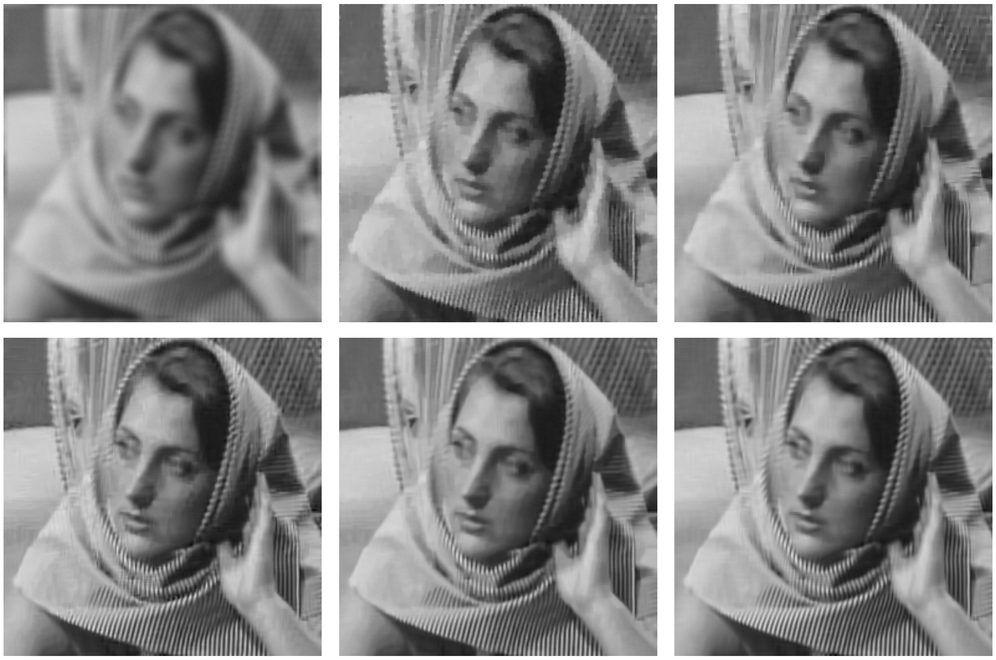


Figure 9: Visual quality comparison of image deblurring on gray image *Barbara* (256×256). From left to right and top to bottom: noisy and blurred image (uniform kernel: 9×9, $\sigma=\sqrt{2}$), the deblurred images by TVMM [9] (PSNR=26.00dB; FSIM=0.8538), L0_ABS [46] (PSNR=26.41dB; FSIM=0.8692), NCSR [15] (PSNR=28.10dB; FSIM=0.9117), IDDBM3D [14] (PSNR=27.98dB; FSIM= 0.9014) and the proposed GSR (PSNR=**28.95**dB; FSIM=**0.9227**).

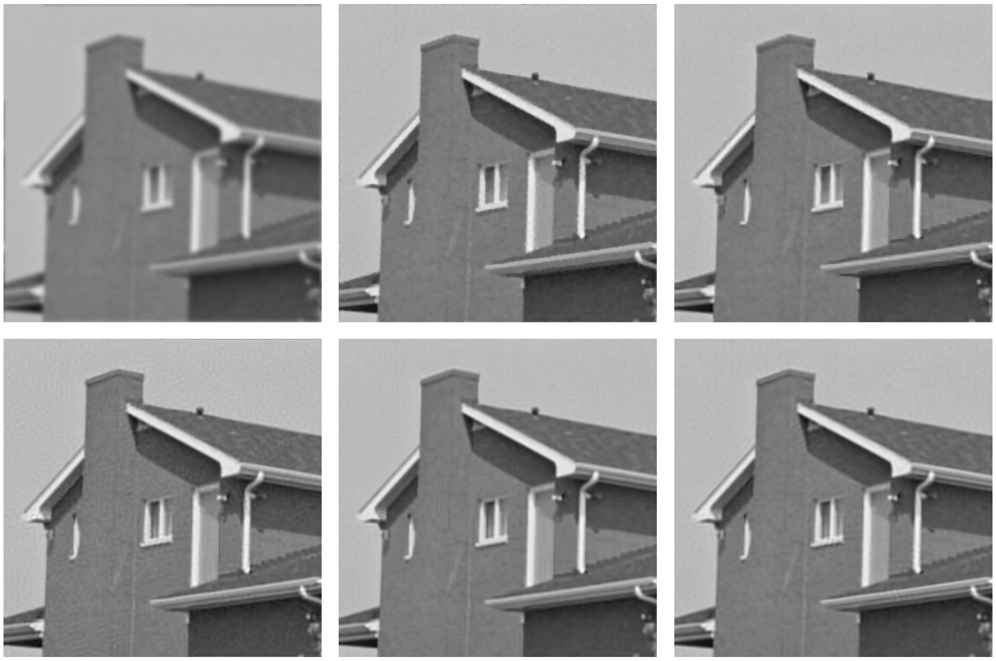


Figure 10: Visual quality comparison of image deblurring on gray image *House* (256×256). From left to right and top to bottom: noisy and blurred image (Gaussian kernel: $\sigma=\sqrt{2}$), the deblurred images by TVMM [9] (PSNR=33.01dB; FSIM=0.9139), L0_ABS [46] (PSNR=33.07dB; FSIM=0.9212), NCSR [15] (PSNR=33.63dB; FSIM=0.9333), IDDBM3D [14] (PSNR=34.08dB; FSIM= 0.9359) and the proposed GSR (PSNR=**34.45**dB; FSIM=**0.9420**).

Table 5: Comparison of the ISNR (dB) Deblurring Results in the Second Set

	Scenario						Scenario					
	1	2	3	4	5	6	1	2	3	4	5	6
Method	*Cameraman* (256×256)						*House* (256×256)					
BSNR	31.87	25.85	40.00	18.53	29.19	17.76	29.16	23.14	40.00	15.99	26.61	15.15
Input PSNR	22.23	22.16	20.76	24.62	23.36	29.82	25.61	25.46	24.11	28.06	27.81	29.98
TVMM [9]	7.41	5.17	8.54	2.57	3.36	1.30	7.98	6.57	10.36	4.12	4.54	2.44
L0_ABS [46]	7.70	5.55	9.10	2.93	3.49	1.77	8.40	7.12	11.06	4.55	4.80	2.15
IDDBM3D [14]	**8.85**	**7.12**	**10.45**	**3.98**	4.31	**4.89**	9.95	8.55	12.89	5.79	5.74	7.13
NCSR [15]	8.78	6.69	10.33	3.78	**4.60**	4.50	9.96	8.48	13.12	5.81	5.67	6.94
GSR	8.39	6.39	10.08	3.33	3.94	4.76	**10.02**	**8.56**	**13.44**	**6.00**	**5.95**	**7.18**
Method	*Lena* (512×512)						*Barbara* (512×512)					
BSNR	29.89	23.87	40.00	16.47	27.18	15.52	30.81	24.79	40.00	17.35	28.07	16.59
Input PSNR	27.25	27.04	25.84	28.81	29.16	30.03	23.34	23.25	22.49	24.22	23.77	29.78
TVMM [9]	6.36	4.98	7.47	3.52	3.61	2.79	3.10	1.33	3.49	0.41	0.75	0.59
L0_ABS [46]	6.66	5.71	7.79	4.09	4.22	1.93	3.51	1.53	3.98	0.73	0.81	1.17
IDDBM3D [14]	7.97	6.61	8.91	4.97	4.85	6.34	7.64	3.96	6.05	1.88	1.16	5.45
NCSR [15]	8.03	6.54	9.25	4.93	4.86	6.19	7.76	3.64	5.92	2.06	1.43	5.50
GSR	**8.24**	**6.76**	**9.43**	**5.17**	**4.96**	**6.57**	**8.98**	**4.80**	**7.15**	**2.19**	**1.58**	**6.20**

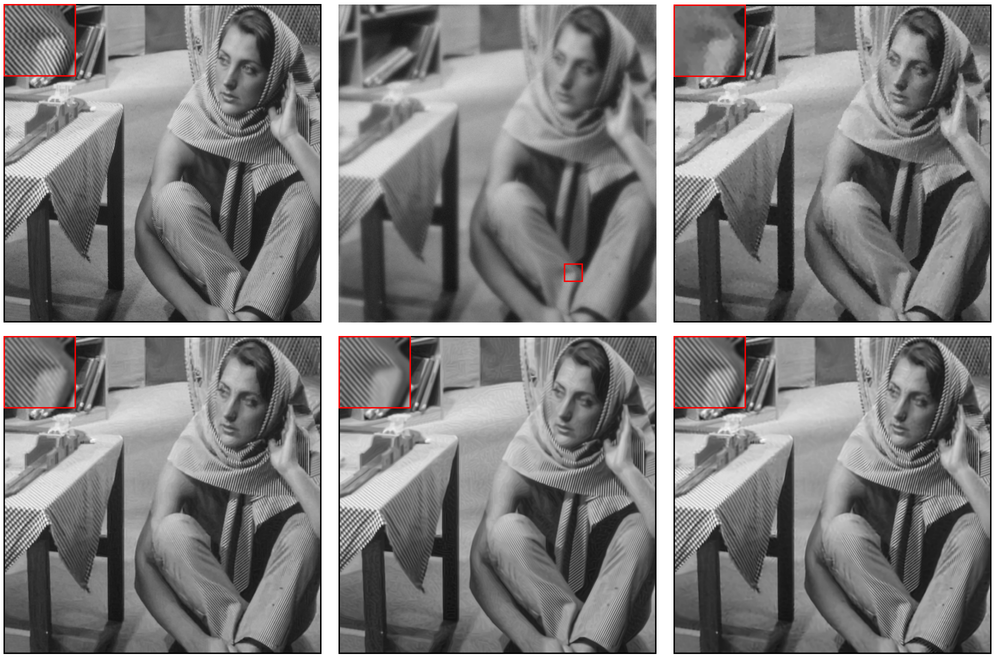


Figure 11: Visual quality comparison of image deblurring on image *Barbara* (512×512). From left to right and top to bottom: original image, noisy and blurred image (scenario 2), the deblurred images by TVMM [8] (PSNR=24.58dB; FSIM=0.9576), IDDBM3D [10] (PSNR=27.21dB; FSIM=0.9699), NCSR [11] (PSNR=26.89dB; FSIM=0.9669) and the proposed GSR (PSNR=**28.05**dB; FSIM=**0.9738**).

Table 5 lists the comparison of ISNR results for six typical deblurring experiments in the second set. It is clear to observe that GSR achieves the highest ISNR results in the most cases, as labeled in bold. In particular, for image *Barbara* (512×512) with rich textures, GSR outperforms current state-of-the-art methods NCSR and IDDBM3D more than 1 dB in the scenarios 1, 2, 3, with more textures and clearer edges than other competing methods, as shown in Fig. 11. More visual results can be found in the website of this paper.

C. Image Compressive Sensing Recovery

From many fewer acquired measurements than suggested by the Nyquist sampling theory, CS theory demonstrates that a signal can be reconstructed with high probability when it exhibits sparsity in some domain, which has greatly changed the way engineers think of data acquisition. More specifically, suppose that we have an image $\boldsymbol{x} \in \mathbb{R}^N$ and its measurement $\boldsymbol{y} \in \mathbb{R}^M$, namely, $\boldsymbol{y} = \boldsymbol{Hx}$. Here, $\boldsymbol{H}$ is an *M*×*N* measurement matrix such that *M* is much smaller than *N*. The purpose of image CS recovery is to recover $\boldsymbol{x}$ from $\boldsymbol{y}$ with measurement rate, denoted by ratio, equal to *M*/*N*. For image CS recovery application, $\mu = 0.0025$ and $\lambda = 0.082$.

In our simulations, the CS measurements are obtained by applying a Gaussian random projection matrix to the original image signal at block level, i.e., block-based CS with block size of 32×32 [47]. GSR is compared with four representative CS recovery methods in literature, i.e., wavelet method (DWT), total variation (TV) method [41], multi-hypothesis (MH) method [40], collaborative sparsity (CoS) method [42], which deal with image signals in the wavelet domain, the gradient domain, the random projection residual domain, and the hybrid space-transform domain, respectively. It is worth emphasizing that MH and CoS are known as the current state-of-the-art algorithms for image CS recovery.

Table 6: PSNR and FSIM Comparisons with Various CS Recovery Methods (dB)

Ratio	*Algorithms*	*House*	*Barbara*	*Leaves*	*Monarch*	*Parrot*	*Vessels*	**Avg.**
20%	**DWT**	30.70/0.9029	23.96/0.8547	22.05/0.7840	24.69/0.8155	25.64/0.9161	21.14/0.8230	24.70/0.8494
	TV [41]	31.44/0.9051	23.79/0.8199	22.66/0.8553	26.96/0.8870	26.6/0.9018	22.04/0.8356	25.59/0.8675
	MH [40]	33.60/0.9370	31.09/0.9419	24.54/0.8474	27.03/0.8707	28.06/0.9332	24.95/0.8756	28.21/0.9010
	CoS [42]	34.34/0.9326	26.60/0.8742	27.38/0.9304	28.65/0.9171	28.44/0.9282	26.71/0.9214	28.69/0.9259
	GSR	**36.78/0.9618**	**34.59/0.9703**	**29.90/0.9499**	**29.55/0.9236**	**30.60/0.9512**	**31.58/0.9599**	**32.17/0.9528**
30%	**DWT**	33.60/0.9391	26.26/0.8980	24.47/0.8314	27.23/0.8628	28.03/0.9445	24.82/0.8924	27.40/0.8947
	TV [41]	33.75/0.9384	25.03/0.8689	25.85/0.9092	30.01/0.9279	28.71/0.9329	25.13/0.9011	28.08/0.9131
	MH [40]	35.54/0.9546	33.47/0.9614	27.65/0.8993	29.18/0.9003	31.20/0.9529	29.36/0.9360	31.07/0.9341
	CoS [42]	36.69/0.9592	29.49/0.9267	31.02/0.9606	31.38/0.9449	30.39/0.9490	31.35/0.9664	31.72/0.9511
	GSR	**38.93/0.9761**	**36.92/0.9811**	**33.82/0.9731**	**33.17/0.9558**	**33.95/0.9671**	**36.33/0.9841**	**35.53/0.9729**
40%	**DWT**	35.69/0.9576	28.53/0.9327	26.82/0.8741	29.58/0.9011	30.06/0.9588	29.53/0.9467	30.03/0.9285
	TV [41]	35.56/0.9574	26.56/0.9088	28.79/0.9442	32.92/0.9538	30.54/0.9530	28.14/0.9441	30.42/0.9436
	MH [40]	37.04/0.9676	35.20/0.9727	29.93/0.9276	31.07/0.9217	33.21/0.9651	33.49/0.9677	33.32/0.9537
	CoS [42]	38.46/0.9724	32.76/0.9618	33.87/0.9744	33.98/0.9637	32.55/0.9627	33.95/0.9784	34.26/0.9689
	GSR	**40.60/0.9836**	**38.99/0.9877**	**37.02/0.9846**	**36.07/0.9714**	**36.44/0.9780**	**40.24/0.9922**	**38.23/0.9829**

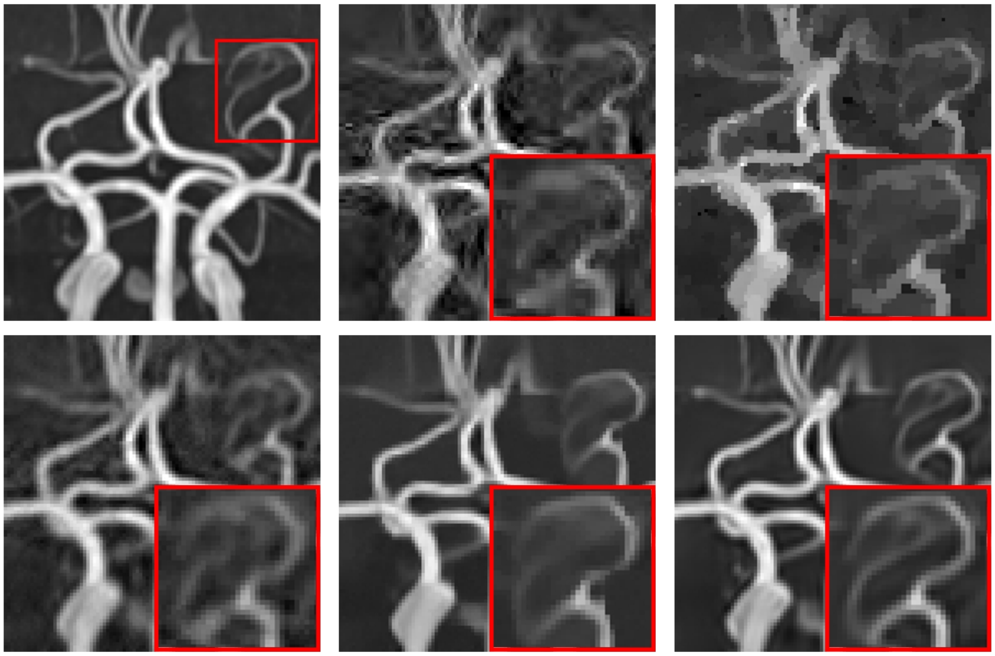


Figure 12: Visual quality comparison of image CS recovery on gray image *Vessels* in the case of ratio = 20%. From left to right: original image, the CS recovered images by DWT (PSNR=21.14dB; FSIM=0.8230), TV [41] (PSNR=22.04dB; FSIM=0.8356), MH [40] (PSNR=24.95dB; FSIM=0.8756), CoS [42] (PSNR= 26.71dB; FSIM=0.9214) and the proposed GSR (PSNR= **31.58**dB; FSIM=**0.9599**).

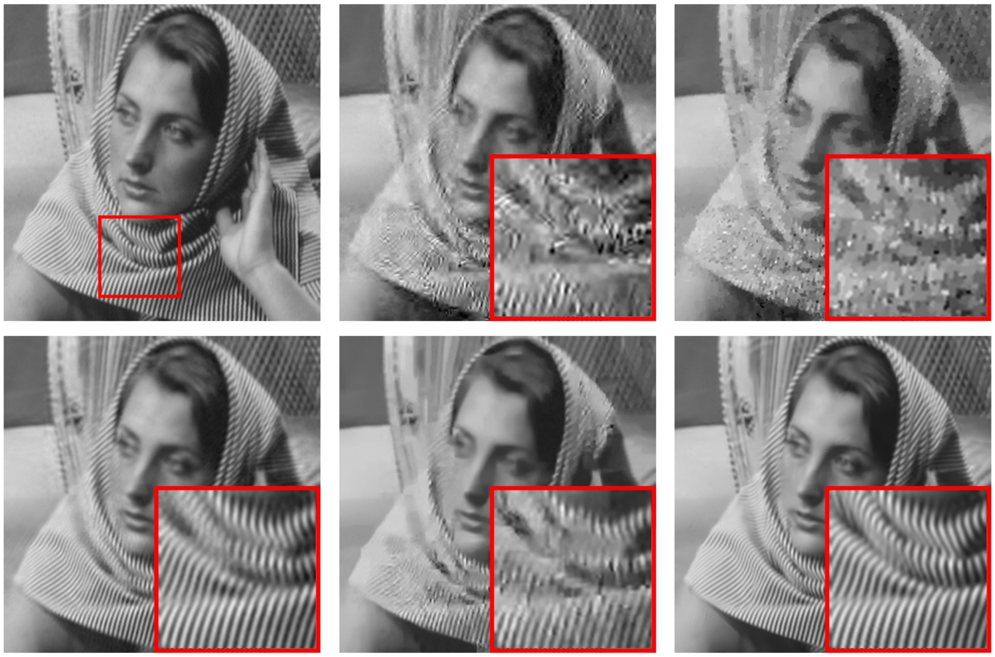


Figure 13: Visual quality comparison of image CS recovery on gray image *Barbara* in the case of ratio = 20%. From left to right and top to bottom: original image, the CS recovered images by DWT (PSNR=23.96dB; FSIM=0.8547), TV [41] (PSNR =23.79dB; FSIM =0.8199), MH [40] (PSNR=31.09dB; FSIM=0.9419), CoS [42] (PSNR= 26.60dB; FSIM=0.8742) and the proposed GSR (PSNR=**34.59**dB; FSIM =**0.9703**).

The PSNR and FSIM comparisons for six gray test images in the cases of 20%, 30%, and 40% measurements are provided in Table 6. GSR achieves the highest PSNR and FSIM among the six comparative algorithms over all the cases, which can improve roughly 7.9 dB, 7.3 dB, 4.4 dB, and 3.7 dB on average, in comparison with DWT, TV, MH, CoS, respectively, greatly improving existing CS recovery results.

Some visual results of the recovered images by various algorithms are presented in Figs. 12~13. Obviously, DWT and TV generate the worst perceptual results. The CS recovered images by MH and CoS possess much better visual quality than those of DWT and TV, but still suffer from some undesirable artifacts, such as ringing effects and lost details. The proposed algorithm GSR not only eliminates the ringing effects, but also preserves sharper edges and finer details, showing much clearer and better visual results than the other competing methods. Our work also offers a fresh and successful instance to corroborate the CS theory applied for natural images.

D. Effect of Number of Best Matched Patches

This subsection will give the detailed description about how sensitive the performance is affected by c, which is the number of best matched patches.

To investigate the sensitivity of c, experiments with respect to various c, ranging from 20 to 120, in the case of image inpainting and image deblurring for three test images are conducted. The performance comparison with various c is shown in Fig. 14. From Fig.14, it is concluded that the performance of our proposed algorithm is not quite sensitive to c because all the curves are almost flat. The highest performance for each case is usually achieved with c in the range [40, 80]. Therefore, in this paper, c is empirically set to be 60.

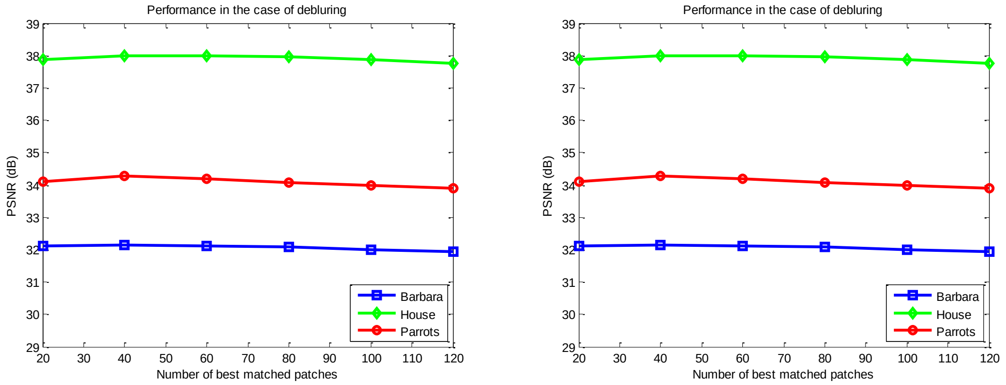


Figure 14: Performance comparison with various c for three test images. From left to right: in the case of image text removal and in the case of image deblurring with 9×9 uniform blur kernel and $\sigma = 0.5$.

E. Effect of Sparsity Parameter

This subsection gives some discussion about how sensitive the performance is affected by the sparsity parameter λ.

To investigate the effect of the sparsity parameter λ for the performance, two scenarios of deblurring experiments are conducted with various blur kernels and noise variances, i.e., scenario 4 and scenario 5 in Table 3. Fig. 15 provides ISNR (dB) evolution with respect to λ in the cases of scenario 4 (PSF = $[1\ 4\ 6\ 4\ 1]^T\,[1\ 4\ 6\ 4\ 1]$, σ = 7) and scenario 5 (PSF = Gaussian with std = 1.6, σ = 2) for two test images. From Fig. 15, three conclusions can be observed. First, as expected, there is an optimal λ that achieves the highest ISNR by balancing image noise suppression with image details preservation (see Fig. 16(c)). That means, if λ is set too small, the image noise can't be suppressed (see Fig. 16(b)); if λ is set too large, the image details will be lost (see Fig. 16(d)). Second, in each case, the optimal λ for each test image is almost the same. For instance, in the case of σ = 7, the optimal λ is 12.2, and in the case of σ = 2, the optimal λ is 0.8. This is very important for parameter optimization, since the optimal λ in each case can be determined by only one test image and then applied to other test images. Third, it is obvious to see that λ has a great relationship with σ, i.e., a larger σ corresponds to a larger λ.

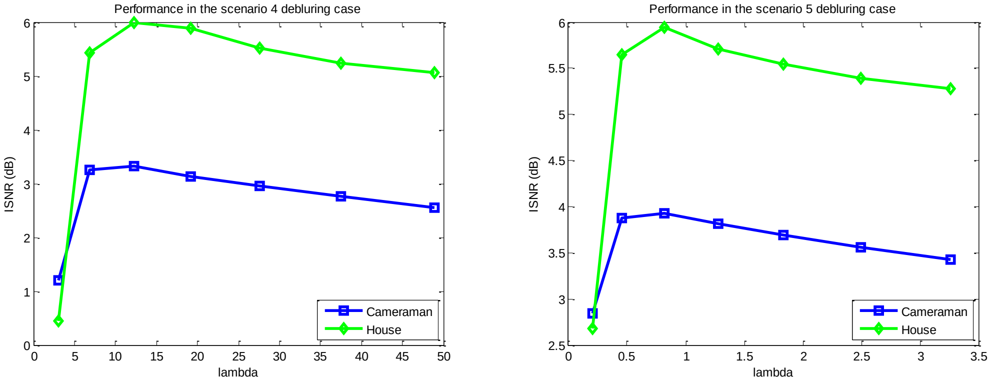
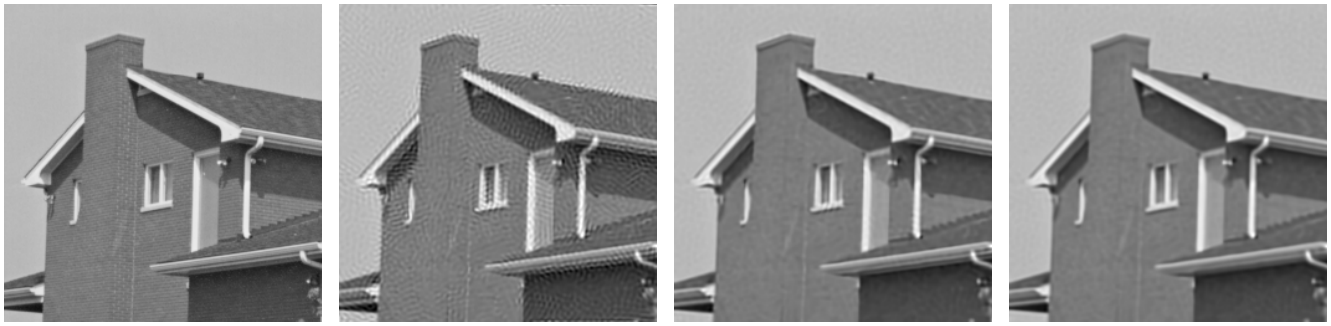


Figure 15: PSNR evolution with respect to sparsity parameter λ in the cases of in the cases of scenario 4 (PSF = $[1\ 4\ 6\ 4\ 1]^T\,[1\ 4\ 6\ 4\ 1]$, σ = 7) and scenario 5 (PSF = Gaussian with std = 1.6, σ = 2) for two test images.



(a) (b) (c) (d)

Figure 16: Visual quality comparison of proposed algorithm with various λ in the case of scenario 5 (PSF = Gaussian with std = 1.6, σ = 2) with respect to image *House*. (a) Original image; (b) Deblurred result with λ = 0.2, ISNR=2.68dB; (c) Deblurred result with λ = 0.8, ISNR=**5.95**dB; (d) Deblurred result with λ = 3.2, ISNR=5.28dB.

F. *Algorithm Complexity and Computational Time*

The complexity of GSR is provided as follows. Assume that the number of image pixels is *N*, that the average time to compute similar patches for each reference patch is T_S . The SVD of each group $\boldsymbol{x}_{G_k}$ with size of $B_s \times c$ is $\mathcal{O}(B_S \times c^2)$. Hence, the total complexity of GSR for image restoration is $\mathcal{O}(N(B_S c^2 + T_S))$. For a 256×256 image, the proposed algorithm GSR requires about 8~9 minutes for image inpainting, 6~7 minutes for image deblurring and 7~8 minutes for CS recovery, on an Intel Core2 Duo 2.96G PC under Matlab R2011a environment.

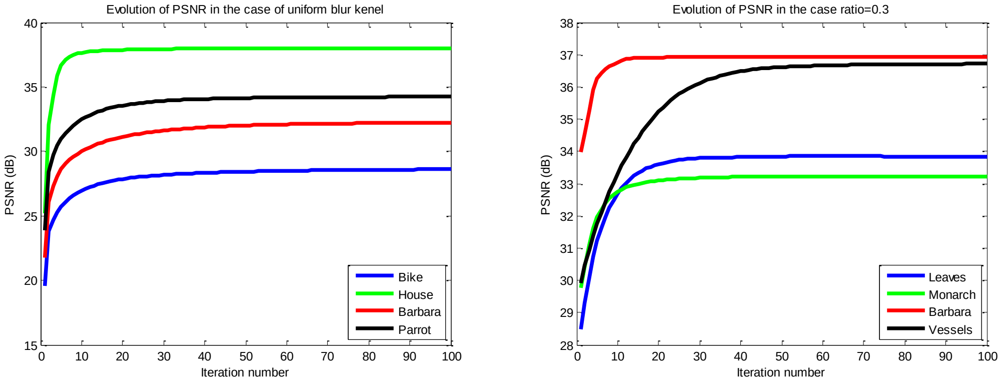


Figure 17: Stability of the proposed algorithm. From left to right: progression of the PSNR (dB) results achieved by proposed GSR for test images with respect to the iteration number in the cases of image deblurring with uniform blur kernel and image CS recovery with ratio=0.3.

G. *Algorithm Stability*

Since the objective function (12) is non-convex, it is difficult to give its theoretical proof for global convergence. Here, we only provide empirical evidence to illustrate the stability of the proposed GSR. Take the cases of image CS recovery and image deblurring as examples. Fig. 17 plots the evolutions of PSNR versus iteration numbers for test images in the cases of image deblurring with uniform blur kernel and CS recovery with ratio=0.3. It is observed that with the growth of iteration number, all the PSNR curves increase monotonically and ultimately become flat and stable, exhibiting good stability of the proposed GSR model.

H. *Comparison between ℓ_0 and ℓ_1 Minimization*

In order to make a comparison between ℓ_0 and ℓ_1 minimization, split Bregman iteration (SBI) is also used to solve Eq. (21). The only difference from solving Eq. (12) described in Table 1 is $\hat{\boldsymbol{\alpha}}_{G_k}$ in Eq. (40) is computed by the operator of soft thresholding, rather than hard thresholding. Take the cases of image deblurring with uniform blur kernel for two images *Barbara* and *Parrot* as examples. Fig. 18 plots their progression curves of the PSNR (dB) results achieved by proposed GSR-driven ℓ_0 and ℓ_1 minimization with respect to the iteration number. The result achieved by GSR-driven ℓ_0 minimization with SBI is denoted by SBI+L0 (blue solid line), while the result achieved by GSR-driven ℓ_1 minimization with SBI is denoted by SBI+L1 (green dotted line). It is obvious that SBI+L0 has better performance than SBI+L1 with more than 1.5 dB on average, which fully demonstrates and the superiority of ℓ_0 minimization (Eq. (12)) over ℓ_1 minimization (Eq. (21)), and validates the effectiveness of our proposed approach to solve Eq. (12) again. Our study also assures the feasibility of using the ℓ_0 minimization for image restoration problems.

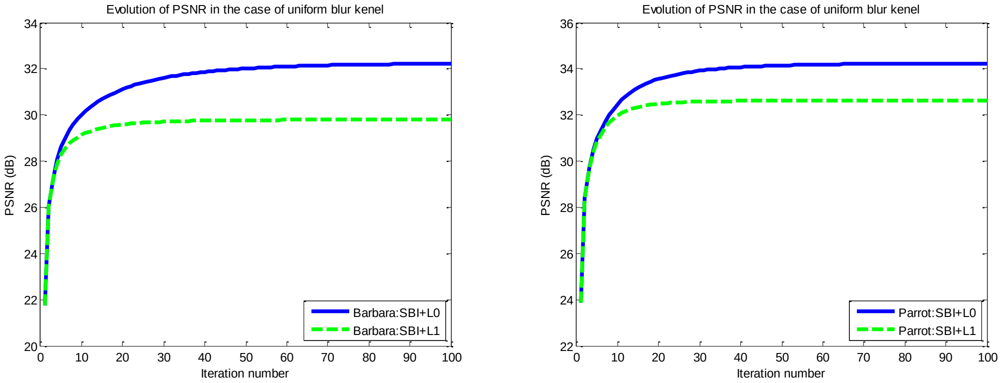


Figure 18: Comparison between GSR-driven ℓ_0 and ℓ_1 minimization solved by SBI. From left to right: progression of the PSNR (dB) results achieved by proposed GSR-driven ℓ_0 and ℓ_1 minimization with respect to the iteration number for images *Barbara* and *Parrot* in the cases of image deblurring with uniform blur kernel.

I. *Comparison between SBI and IST*

In our previous work [47], the convex optimization approach iterative shrinkage/thresholding (IST) is utilized to solve our proposed GSR-driven ℓ_0 minimization for image CS recovery. Here, we make a comparison between SBI and IST. Take the cases of image CS recovery with ratio=0.3 for two images *Monarch* and *Leaves* as examples. Fig. 19 plots their progression curves of the PSNR (dB) results achieved by solving GSR-driven ℓ_0 minimization with SBI and IST. The result achieved by ℓ_0 minimization with SBI is denoted by SBI+L0 (red solid line), while the result achieved by ℓ_0 minimization with IST is denoted by IST+L0 (black dotted line). Obviously, SBI is more efficient and effective to solve our proposed GSR-driven ℓ_0 minimization problem than IST.

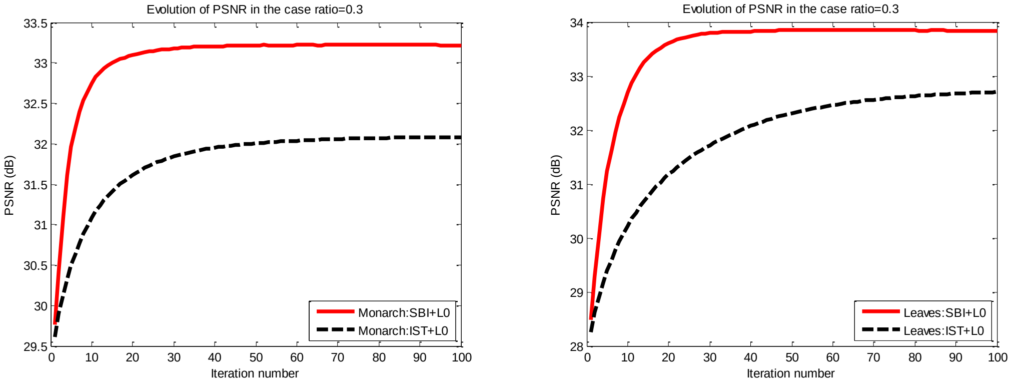


Figure 19: Comparison between SBI and IST for solving GSR-driven ℓ_0 minimization. From left to right: progression of the PSNR (dB) results achieved by proposed GSR-driven ℓ_0 minimization with respect to the iteration number for images *Monarch* and *Leaves* in the cases of image CS recovery with ratio=0.3.

VI. CONCLUSIONS

This paper establishes a novel and general framework for high-quality image restoration using group-based sparse representation (GSR) modeling, which sparsely represents natural images in the domain of group, and explicitly and effectively characterizes the intrinsic local sparsity and nonlocal self-similarity of natural images simultaneously in a unified manner. An effectual self-adaptive group dictionary learning technique with low complexity is designed. To achieve high sparsity degree and high recovery quality, this paper proposes to exploit the convex optimization algorithms to solve the non-convex ℓ_0 minimization problem directly. Our study not only assures the feasibility of using the ℓ_0 minimization for image restoration problems, but also demonstrates the superiority of the ℓ_0 minimization over the ℓ_1 minimization, which is very interesting and surprising. Experimental results on three applications: image inpainting, deblurring and CS recovery have shown that the proposed GSR achieves significant performance improvements over many current state-of-the-art schemes and exhibits good stability. It is worth emphasizing that GSR greatly improves existing CS recovery results, which will promote further research and development of CS theory applied in natural images. Future work includes the extensions of GSR on a variety of applications, such as image deblurring with mixed Gaussian and impulse noise, and video restoration and so on.

REFERENCES

[1] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D.*, vol. 60, pp. 259–268, Nov. 1992.

[2] M. R. Banham and A. K. Katsaggelos, “Digital image restoration,” *IEEE Trans. Signal Processing Mag.*, vol. 14, no. 2, pp. 24–41, Mar. 1997.

[3] L. Zhang, W. Dong, D. Zhang, G. Shi, “Two-stage image denoising by principle component analysis with local pixel grouping,” *Pattern Recognition*, vol. 43, pp.1531–5549, Apr. 2010.

[4] A. Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vis.*, vol. 20, no. 1–2, pp. 89–97, Jan–Mar 2004.

[5] W. Dong, G. Shi, and X. Li, “Nonlocal image restoration with bilateral variance estimation: a low-rank approach,” *IEEE Trans. on Image Processing*, vol. 22, no. 2, pp. 700–711, Feb. 2013.

[6] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” *IEEE Trans. on Image Process.*, vol. 16, no. 2, pp. 349–366, Feb. 2007.

[7] S. Roth and M. J. Black, “Fields of experts,” *International Journal of Computer Vision*, vol. 82, no. 2, pp. 205–229, 2009.

[8] A. Buades, B. Coll, and J. M. Morel, “A non-local algorithm for image denoising,” *Proc. of Int. Conf. on Computer Vision and Pattern Recognition*, pp. 60–65, 2005.

[9] J. Bioucas-Dias, M. Figueiredo, J. Oliveira, “Total-variation image deconvolution: A majorization-minimization approach,” *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.

[10] X. Li, “Image recovery from hybrid sparse representation: a deterministic annealing approach,” *IEEE J. of Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 953–962, Sep. 2011

[11] W. Dong, L. Zhang, G. Shi, X. Wu, “Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization,” *IEEE Trans. on Image Processing*, vol. 20, no. 7, pp. 1838–1857, Jul. 2011.

[12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” *Proc. IEEE Int. Conf. Comput. Vis.*, Tokyo, Japan, pp. 2272–2279, Sep. 2009.

[13] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, “Framelet based blind motion deblurring from a single image,” *IEEE Transactions on Image Processing*, vol. 21, no. 2, pp. 562–572, Feb. 2012.

[14] A. Danielyan, V. Katkovnik, and K. Egiazarian, “BM3D frames and variational image deblurring,” *IEEE Trans. Image Process.*, vol. 21,no. 4, pp. 1715–1728, Apr. 2012.

[15] W. Dong, L. Zhang, G. Shi and X. Li, “Nonlocally centralized sparse representation for image restoration,” *IEEE Trans. on Image Processing*, vol. 22, no. 4, pp. 1620-1630, Apr. 2013.

[16] J. Bioucas-Dias and M. Figueiredo, “A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration,” *IEEE Trans. on Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[17] A. Beck and M. Teboulle, “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems,” *IEEE Trans. on Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[18] M. Afonso, J. Bioucas-Dias and M. Figueiredo, “Fast image recovery using variable splitting and constrained optimization,” *IEEE Trans. on Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[19] M. Duarte, M. Davenport, D. Takhar, J. Laska, and T. Sun, “Single-pixel imaging via compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[20] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma, and W. Gao, “Compressed sensing recovery via collaborative sparsity,” *Proc. of IEEE Data Compression Conference*, pp. 287–296, Snowbird, Utah, USA, Apr. 2012.

[21] D. Geman and G. Reynolds, “Constrained restoration and the recovery of discontinuities,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 14, no. 3, pp. 367–383, Mar. 1992.

[22] D. Mumford and J. Shah, “Optimal approximation by piecewise smooth functions and associated variational problems,” *Comm. on Pure and Appl. Math.*, vol. 42, pp. 577–685, Jul. 1989.

[23] A. M. Bruckstein, D.L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[24] M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation,” *IEEE Trans. on Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[25] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[26] J. Yang, J. Wright, T. Huang, and Y. Ma, “Image super-resolution via sparse representation,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[27] K. Engan, S. O. Aase, and J. H. Hakon-Husoy, “Method of optimal directions for frame design,” *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, pp. 2443–2446, 1999.

[28] Joel A. Tropp and Anna C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[29] J. Tropp, A. Gilbert, M. Strauss, “Algorithms for simultaneous sparse approximation part I: greedy pursuit,” *Signal processing*, vol. 86, no. 3 pp. 572–588, Mar. 2006.

[30] L. Xu, C. Lu, Y. Xu, and J. Jia, “Image smoothing via L0 gradient minimization,” *ACM Trans. on Graphics*, vol. 30, no. 6, Dec. 2011.

[31] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image restoration by sparse 3-D transform-domain collaborative filtering,” *Proc. of SPIE Electron. Imaging*, San Jose, CA, 2008, vol. 6812, Art. ID 6812–1D.

[32] S. Kindermann, S. Osher, and P.W. Jones, “Deblurring and denoising of images by nonlocal functionals,” *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1091–1115, 2005.

[33] A. Elmoataz, O. Lezoray, and S. Bougleux, “Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing,” *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1047–1060, Jul. 2008.

[34] G. Peyr é, “Image processing with nonlocal spectral bases,” *Multiscale Model. Simul.*, vol. 7, no. 2, pp. 703–730, 2008.

[35] X. Zhang, M. Burger, X. Bresson and S. Osher, “Bregmanized nonlocal regularization for deconvolution and sparse reconstruction,” *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 253–276, 2010.

[36] M. Jung, X. Bresson, T. F. Chan, and L. A. Vese, “Nonlocal Mumford-Shah regularizers for color image restoration,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1583–1598, Jun. 2011.

[37] S. Mallat and G. Yu, “Super-resolution with sparse mixing estimators,” *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2889–2900, Nov. 2010.

[38] S. Chen, D. Donoho, M. Saunders, “Atomic decompositions by basis pursuit,” *SIAM Review*, vol. 43, pp. 129–159, 2001.

[39] M. K. Varanasi, B. Aazhang, “Parametric generalized Gaussian density estimation,” *J. Acoust. Soc. Amer.*, vol. 86, no. 4, pp. 1404–1415, 1989.

[40] C. Chen, E. W. Tramel, and J. E. Fowler, “Compressed-Sensing Recovery of Images and Video Using Multi-hypothesis Predictions,” *Proc. of the 45th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, pp. 1193–1198, Nov. 2011.

[41] C. Li, W. Yin, and Y. Zhang, “TVAL3: TV Minimization by Augmented Lagrangian and Alternating Direction Algorithm,” 2009.

[42] J. Zhang, D. Zhao, C. Zhao, R. Xiong, S. Ma and W. Gao, “Image compressive sensing recovery via collaborative sparsity,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 380–391, Sep. 2012.

[43] T. Goldstein and S. Osher, “The split Bregman algorithm for L1 regularized problems,” *SIAM J. Imaging Sci.*, vol. 2, pp. 323–343, Apr. 2009.

[44] J. Zhang, D. Zhao, R. Xiong, S. Ma, and W. Gao, “Image Restoration Using Joint Statistical Modeling in Space-Transform Domain,” *IEEE Transactions on Circuits System and Video Technology* (2014), DOI: 10.1109/TCSVT.2014.2302380.

[45] L. Zhang, L. Zhang, X. Mou and D. Zhang, “FSIM: A Feature SIMilarity index for image quality assessment,” *IEEE Trans. Image Processing*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.

[46] J. Portilla, “Image restoration through l0 analysis-based sparse optimization in tight frames,” *IEEE Int. conf. Image Process.*, pp. 3909-3912, Nov. 2009.

[47] J. Zhang, D. Zhao, F. Jiang and W. Gao, “Structural group sparse representation for image compressive sensing recovery,” *Proc. of IEEE Data Compression Conference*, pp. 331–340, Snowbird, Utah, USA, Mar. 2013.

[48] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro and L. Carin, “Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images,” *IEEE Trans. Image Processing*, vol. 21, no. 1, pp. 130–144, Jan. 2012.

[49] J. Zhang, C. Zhao, D. Zhao, and W. Gao, “Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization,” *Signal Processing* (2013), DOI: 10.1016/j.sigpro.2013.09.025.

[50] H. Ji, C. Liu, Z. Shen, and Y. Xu, “Robust video denoising using low rank matrix completion,” *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 1791–1798, 2010.

[51] H. Ji, S. Huang, Z. Shen, and Y. Xu, “Robust video restoration by joint sparse and low rank matrix approximation,” *SIAM J. Imaging Sci.*, vol. 4, no. 4, pp. 1122–1142, 2011.

APPENDIX A

Proof of **Theorem 1**:

Due to the assumption that each $\boldsymbol{e}(j)$ is independent, we obtain that each $\boldsymbol{e}(j)^2$ is also independent. Since $E[\boldsymbol{e}(j)]=0$ and $\mathrm{Var}[\boldsymbol{e}(j)]=\sigma^2$, we have the mean of each $\boldsymbol{e}(j)^2$, that is

$$E[\boldsymbol{e}(j)^2]=\mathrm{Var}[\boldsymbol{e}(j)]+[E[\boldsymbol{e}(j)]]^2=\sigma^2,\ \ j=1,...,N. \tag{41}$$

By invoking *Law of Large Numbers* in probability theory, for any $\varepsilon>0$, it yields $\lim_{N\to\infty} P\left\{ \left| \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{e}(j)^2 - \sigma^2 \right| < \frac{\varepsilon}{2} \right\} = 1$, i.e.,

$$\lim_{N\to\infty} P\left\{ \left| \frac{1}{N}\|\boldsymbol{x}-\boldsymbol{r}\|_2^2 - \sigma^2 \right| < \frac{\varepsilon}{2} \right\} = 1\ , \tag{42}$$

Further, let $\boldsymbol{x}_G, \boldsymbol{r}_G$ denote the concatenation of all $\boldsymbol{x}_{G_k}$ and $\boldsymbol{r}_{G_k}$, $k=1,2,...,n$, respectively, and denote each element of $\boldsymbol{x}_G-\boldsymbol{r}_G$ by $\boldsymbol{e}_G(i), i=1,...,K$. Due to the assumption, we conclude that $\boldsymbol{e}_G(i)$ is independent with zero mean and variance σ^2.

Therefore, the same manipulations with Eq. (40) applied to $\boldsymbol{e}_G(i)^2$ lead to $\lim_{K\to\infty} P\left\{ \left| \frac{1}{K}\sum_{i=1}^{K} \boldsymbol{e}_G(i)^2 - \sigma^2 \right| < \frac{\varepsilon}{2} \right\} = 1$, namely,

$$\lim_{K\to\infty} P\left\{ \left| \frac{1}{K}\sum_{k=1}^{n} \left\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\right\|_F^2 - \sigma^2 \right| < \frac{\varepsilon}{2} \right\} = 1\ . \tag{43}$$

Considering Eqs. (42) and (43) together, we prove Eq. (34). □