Joint Patch-Group Based Sparse Representation for Image Inpainting

Zhiyuan Zha[1,2] ZHAZHIYUAN.MMD@GMAIL.COM
Xin Yuan[3] XYUAN@BELL-LABS.COM
Bihan Wen[4] BWEN3@ILLINOIS.EDU
Jiantao Zhou[5] JTZHOU@UMAC.MO
Ce Zhu[1] * ECZHU@UESTC.EDU.CN

[1] *School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China.*

[2] *School of Electronic Science and Engineering, Nanjing University, Nanjing, 210023, China.*

[3] *Nokia Bell Labs, 600 Mountain Avenue, Murray Hill, NJ, 07974, USA.*

[4] *Department of Electrical and Computer Engineering, Univerisity of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA.*

[5] *Department of Computer and Information Science, University of Macau, Macau 999078, China.*

Editors: Jun Zhu and Ichiro Takeuchi

Abstract

Sparse representation has achieved great successes in various machine learning and image processing tasks. For image processing, typical patch-based sparse representation (PSR) models usually tend to generate undesirable visual artifacts, while group-based sparse representation (GSR) models produce over-smooth phenomena. In this paper, we propose a new sparse representation model, termed joint patch-group based sparse representation (JPG-SR). Compared with existing sparse representation models, the proposed JPG-SR provides a powerful mechanism to integrate the local sparsity and nonlocal self-similarity of images. We then apply the proposed JPG-SR model to a low-level vision problem, namely, image inpainting. To make the proposed scheme tractable and robust, an iterative algorithm based on the alternating direction method of multipliers (ADMM) framework is developed to solve the proposed JPG-SR model. Experimental results demonstrate that the proposed model is efficient and outperforms several state-of-the-art methods in both objective and perceptual quality.

Keywords: Sparse representation, joint patch-group based sparse representation, nonlocal self-similarity, image inpainting, ADMM.

1. Introduction

As a very popular technique in image processing, sparse representation has attracted significant interests (Elad and Aharon, 2006; Aharon et al., 2006; Caiafa et al., 2017; Mairal et al., 2009; Dong et al., 2011; Zhang et al., 2014a; Liu and Tsang, 2017). Generally speaking, methods of sparse representation can be classified into two categories: patch-based sparse

* Corresponding Author. This work was supported by the NSFC (61571102) and the applied research programs of science and technology., Sichuan Province (No. 2018JY0035). The first two authors contributed to this work equally.

representation (PSR) and group-based sparse representation (GSR). PSR assumes that each patch of an image can be perfectly modeled by a sparse linear combination of learnable basis elements. These elements, called atoms, compose a dictionary (Aharon et al., 2006). The dictionary is usually learned from a natural image dataset. Compared with the traditionally analytic dictionaries, such as DCT and wavelet, dictionaries learned directly from images are superior to be adapted to image local structures, and thus could improve the sparsity which results in better performance. For instance, the seminal work of KSVD dictionary learning method proposed by Aharon et al. (2006) has not only achieved promising denoising results, but also been extended to various image processing and computer vision tasks (Zhang and Li, 2010; Lun et al., 2016). However, it has been shown that the PSR with an over-complete dictionary is usually unstable and tends to generate undesirable visual artifacts (Elad and Yavneh, 2009). Moreover, it is computationally expensive to learn an off-the-shelf dictionary, and the PSR model usually ignores the correlation among similar patches (Mairal et al., 2009; Zhang et al., 2014a; Zha et al., 2017).

Inspired by the success of the nonlocal self-similarity (NSS) prior in images (Buades et al., 2005; Dabov et al., 2007; Li et al., 2016), instead of using a single patch as the basic unit in sparse representation, recent advances in GSR considers similar patch group as the basic unit, while similar to PSR, it can be sparsely represented by a set of sparse codes in the group domain, i.e., each group can also be precisely represented by a sparse linear combination of basis elements of the dictionary Mairal et al. (2009). The GSR have demonstrated great potential in various image processing tasks (Mairal et al., 2009; Zhang et al., 2014a; Dabov et al., 2007). For example, Dabov et al. (2007) proposed the BM3D method to combine NSS prior with transform domain filtering, which is still one of the state-of-the-art denoising methods. Mairal et al. (2009) proposed the learned simultaneous sparse coding (LSSC) to improve denoising performance of KSVD Aharon et al. (2006) via GSR. Zhang et al. (2014a) proposed a group-based sparse representation model for image restoration. Though GSR models have shown great successes in various image processing tasks, they may suffer over-smooth effect in the recovered image (Li et al., 2016).

Bearing the above concern in mind, we aim to address the following questions in this paper.

1) Is it possible to mitigate the drawbacks of PSR and GSR models, respectively?

2) Is it possible to build a joint model to integrate the PSR and GSR models? If yes, can we solve the model efficiently?

We answer these questions by developing a new sparse representation model, dubbed joint patch-group based sparse representation (JPG-SR). Compared with previous sparse representation models, the proposed JPG-SR is capable of integrating the local sparsity with NSS of the image. To make the optimization tractable, we develop an iterative algorithm based on the alternating direction method of multipliers (ADMM) framework to solve the proposed model. Through applying the proposed JPG-SR model to a low-level vision task, i.e., image inpainting, we demonstrate that the JPG-SR not only retains the advantages of PSR and GSR, but also alleviates their drawbacks, respectively.

The flowchart of the proposed JPG-SR model for image inpainting is illustrated in Fig. 1. Note that the PSR (the top row in Fig. 1) and the GSR (the bottom row in Fig. 1)

are fed into an ADMM framework to recover the image (right part in Fig. 1) in one shot. Experimental results demonstrate that the proposed model is efficient and outperforms many state-of-the-art approaches.

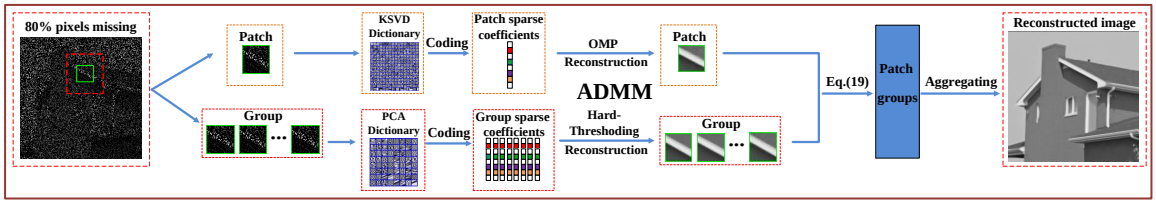


Figure 1: Flowchart of the proposed JPG-SR model for image inpainting. The corrupted image (left) are fed into our JPG-SR model on two paths, i.e., PSR on top and GSR in the bottom. These two paths are jointly optimized by the proposed ADMM framework producing the finally recovered image (right).

The remainder of this paper is organized as follows. Section 2 introduces the related work about sparse representation. Section 3 develops our new sparse representation model, i.e., the Joint Patch-Group Sparse Representation model. Section 4 develops an optimization method to solve the proposed JPG-SR model for the image inpainting task. Section 5 presents the experimental results. Finally, several concluding remarks are given in Section 6.

2. Background

2.1. Patch-based Sparse Representation

Following the notations in Elad and Aharon (2006), the basic unit of sparse representation for images is patch. Mathematically, for an (vectorized) image $\boldsymbol{x} \in \mathbb{R}^N$, let $\boldsymbol{x}_i = \boldsymbol{R}_i\boldsymbol{x}, i = 1, 2, ...n$, denotes an (vectorized) image patch of size $\sqrt{b} \times \sqrt{b}$ extracted from location i. Given a dictionary $\mathbf{D} \in \mathbb{R}^{b\times M}$, $b \leq M$, the sparse representation processing of each patch $\boldsymbol{x}_i$ is to find a sparse vector where most coefficients are zero. Specifically, each patch $\boldsymbol{x}_i$ can be sparsely represented as $\boldsymbol{x}_i \approx \mathbf{D}\boldsymbol{\alpha}_i$ by solving the following ℓ_0 minimization problem,

$$\hat{\boldsymbol{\alpha}}_i = \arg\min_{\boldsymbol{\alpha}_i} \left(\sum\nolimits_{i=1}^{n} \frac{1}{2}\|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_0 \right), \tag{1}$$

where λ is a regularization parameter, and $||\ ||_0$ signifies the ℓ_0-norm (quasi-norm), i.e., counting the nonzero entries in $\boldsymbol{\alpha}_i$. Then the entire image $\boldsymbol{x}$ can be sparsely represented by a set of sparse codes $\{\boldsymbol{\alpha}_i\}_{i=1}^n$. Concatenating n patches, let $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{b\times n}$ denote all the patches extracted from the image. Since $\mathbf{D}$ is shared by these patches, we have

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\boldsymbol{\alpha}} \left(\frac{1}{2}\|\mathbf{X} - \mathbf{D}\boldsymbol{\alpha}\|_F^2 + \lambda\|\boldsymbol{\alpha}\|_0 \right), \tag{2}$$

where $||\ ||_F^2$ denotes the Frobenius norm, $\boldsymbol{\alpha} \in \mathbb{R}^{M\times n}$ is the coefficient matrix, and the ℓ_0-norm is imposed on each column of $\boldsymbol{\alpha}$ (corresponding to each patch).

2.2. Group-based Sparse Representation

Instead of using a single patch as the basic unit in sparse representation, recent studies have shown that GSR using patch groups can produce more promising results for various image processing tasks than typical PSR models (Mairal et al., 2009; Zhang et al., 2014a; Dabov et al., 2007; Li et al., 2016). Hereby, we briefly introduce the GSR model.

In particular, image $\boldsymbol{x}$ is firstly divided into n overlapped patches $\boldsymbol{x}_i$ of size $\sqrt{b}\times\sqrt{b}, i = 1, 2, ..., n$. Then, different from PSR, for each exemplar patch $\boldsymbol{x}_i$, its most similar m patches (by KNN method (Keller et al., 1985)) are selected from a $W \times W$ sized searching window to form a set $\mathcal{S}_{G_i}$. Following this, all the patches in $\mathcal{S}_{G_i}$ are stacked into a matrix $\mathbf{X}_{G_i} \in \mathbb{R}^{b\times m}$, which contains every patch in $\mathcal{S}_{G_i}$ as its column, *i.e.*, $\mathbf{X}_{G_i} = \{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, ..., \boldsymbol{x}_{i,m}\}$. This matrix $\mathbf{X}_{G_i}$ consisting of all the patches with similar structures is thus called a group, where $\boldsymbol{x}_{i,j}$ denotes the j-th patch (column) in the i-th group. Finally, similar to PSR, given a dictionary $\mathbf{D}_{G_i} \in \mathbb{R}^{b\times K}$, each group $\mathbf{X}_{G_i}$ can be sparsely represented and solved by

$$\hat{\boldsymbol{\beta}}_{G_i} = \arg\min_{\boldsymbol{\beta}_{G_i}} \left(\frac{1}{2}\|\mathbf{X}_{G_i} - \mathbf{D}_{G_i}\boldsymbol{\beta}_{G_i}\|_F^2 + \lambda\|\boldsymbol{\beta}_{G_i}\|_0 \right). \tag{3}$$

In order to put all groups in one shot, let $\mathbf{Q}_i \in \mathbb{R}^{n\times m}$ denotes the searching and extracting operations of the similar patches for the i-th patch, i.e., $\mathbf{X}_{G_i} = \mathbf{X}\mathbf{Q}_i$. Concatenating n patch groups, we have $\mathbf{X}_G = \mathbf{X}[\mathbf{Q}_1, \ldots, \mathbf{Q}_n] = \mathbf{X}\mathbf{Q} \in \mathbb{R}^{b\times mn}$. Due to the fact that each group has its own dictionary and they are not necessarily shared, let $\mathbf{D}_G = [\mathbf{D}_{G_1}, \ldots, \mathbf{D}_{G_n}] \in \mathbb{R}^{b\times(nK)}$ and $\bar{\boldsymbol{\beta}}_G = [\bar{\boldsymbol{\beta}}_{G_1}, \ldots, \bar{\boldsymbol{\beta}}_{G_n}]$, where $\{\bar{\boldsymbol{\beta}}_{G_i}\}_{i=1}^n \in \mathbb{R}^{nK\times m}$ is an extended (longer with more rows) version of $\boldsymbol{\beta}_{G_i} \in \mathbb{R}^{K\times m}$, with $\boldsymbol{\beta}_{G_i}$ in the corresponding locations (from $((i-1)K+1)$-th row to (iK)-th row) but zeros elsewhere, i.e., corresponding to $\mathbf{D}_{G_i}$ in $\mathbf{D}_G$. The problem we are going to solve now becomes

$$\hat{\bar{\boldsymbol{\beta}}}_G = \arg\min_{\bar{\boldsymbol{\beta}}_G} \left(\frac{1}{2}\|\mathbf{X}_G - \mathbf{D}_G\bar{\boldsymbol{\beta}}_G\|_F^2 + \lambda\|\bar{\boldsymbol{\beta}}_G\|_0 \right), \tag{4}$$

where the ℓ_0-norm is again imposed on each column and this holds true for the following derivations in this paper. It is worth noting that both $\mathbf{X}$ in PSR and $\mathbf{X}_G$ in GSR are constructed from the same image $\boldsymbol{x}$.

3. Joint Patch-Group based Sparse Representation Model

The PSR model usually generates some undesirable visual artifacts, while GSR may lead to over-smooth effect in various image processing tasks. To cope with these problems, instead of using Eq. (2) or Eq. (4) individually, we propose the joint patch-group based sparse representation (JPG-SR) model in this section.

Before doing this, we introduce some preliminary transformations to link the PSR model in Eq. (2) with the GSR model in Eq. (4). Recall that each patch (column) in the patch group $\mathbf{X}_G$ is from $\mathbf{X}$ and it can be sparsely represented by Eq. (2). Therefore, in addition to the sparse representation in Eq. (4), we can also have

$$\mathbf{X}_G = \mathbf{D}\boldsymbol{\alpha}_G, \tag{5}$$

where $\boldsymbol{\alpha}_G \in \mathbb{R}^{M\times(mn)}$ is composed of the corresponding columns in $\boldsymbol{\alpha}$; in other words, $\boldsymbol{\alpha}_G$ is an expanded version of $\boldsymbol{\alpha}$ in Eq. (2), where each column is reproduced by m times according to the patch searching in $\mathbf{X}_G$.

Following this, we propose the JPG-SR model by solving

$$\begin{aligned} \hat{\mathbf{C}} &= \arg\min_{\mathbf{C}} \frac{1}{2}\|\mathbf{X}_G - \mathbf{U}\mathbf{C}\|_F^2 + \lambda\|\boldsymbol{\alpha}_G\|_0 + \rho\|\bar{\boldsymbol{\beta}}_G\|_0, \\ \mathbf{U} &= [\mathbf{D} \;\; \mathbf{D}_G], \quad \mathbf{C} = \begin{bmatrix} \boldsymbol{\alpha}_G \\ \bar{\boldsymbol{\beta}}_G \end{bmatrix}, \end{aligned} \tag{6}$$

where λ and ρ are regularization parameters, balancing the two sparsity inducing penalties ($\|\boldsymbol{\alpha}_G\|_0$ and $\|\bar{\boldsymbol{\beta}}_G\|_0$) and the fidelity term, i.e., $\frac{1}{2}\|\mathbf{X}_G - \mathbf{U}\mathbf{C}\|_F^2$. $||\boldsymbol{\alpha}_G||_0$ corresponds to the patch sparsity prior to retain the image local consistency, reducing the over-smooth effect, while $||\bar{\boldsymbol{\beta}}_G||_0$ is associated with group sparsity prior to keep image nonlocal consistency, suppressing undesirable visual artifacts. In this way, the proposed JPG-SR provides a good way of integrating local sparsity and NSS of images[1].

In Eq. (6), after $\mathbf{C}$ is estimated, we can obtain $\mathbf{X}_G$. Following this, the original image $\boldsymbol{x}$ can be recovered by aggregating the patches. It is worth noting that the desired signal $\mathbf{X}_G$ is only updated when both the patch sparsity prior ($\boldsymbol{\alpha}_G$) and the group sparsity prior ($\bar{\boldsymbol{\beta}}_G$) are available and therefore our model is a joint one.

4. Joint Patch-Group based Sparse Representation for Image Inpainting

We are now considering the proposed JPG-SR model to a low-level vision task, image inpainting.

4.1. Image Inpainting

The goal of image inpainting is to reconstruct a high quality image $\boldsymbol{x}$ from its degraded observation $\boldsymbol{y}$, which is a typical ill-posed inverse problem and can be mathematically expressed as

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{\varphi}, \tag{7}$$

where $\mathbf{H}$ is a diagonal matrix whose diagonal entries are either 1 or 0, keeping or killing corresponding pixels. $\boldsymbol{\varphi}$ is usually assumed to be a zero-mean white Gaussian noise.

Given the degraded image $\boldsymbol{y}$ in Eq. (7) and leveraging the proposed JPG-SR in Eq. (6), we aim to recover the original image $\boldsymbol{x}$ by solving the following minimization problem,

$$\hat{\mathbf{C}} = \arg\min_{\mathbf{C}} \frac{1}{2}||\mathbf{Y}_G - \mathbf{H}_G\mathbf{U}\mathbf{C}||_F^2 + \lambda||\boldsymbol{\alpha}_G||_0 + \rho||\boldsymbol{\beta}_G||_0, \tag{8}$$

where $\mathbf{Y}_G$ is obtained from $\boldsymbol{y}$ in the same procedure of $\mathbf{X}_G$, and similarly for $\mathbf{H}_G$, which is obtained from $\mathbf{H}$.

1. Solving Eq. (6) appears to give the same results of Eq. (2) and Eq. (4) if the global minimum can be found and the joint optimization is not necessary any more in such case. However, the proposed JPG-SR model in Eq. (6) is non-convex, and therefore, this is different from the simple average of two results using PSR and GSR independently. In our experiments, we notice that this joint estimation plays a pivot role in the performance improvement and shows good convergence behavior.

4.2. ADMM Based Algorithm to Solve the Proposed JPG-SR Model

Since Eq. (8) is a large-scale non-convex optimization problem, in order to make the optimization tractable, we employ the alternating direction method of multipliers (ADMM) (He et al., 2002; Boyd et al., 2011) framework, whose underlying principle is to split the unconstrained minimization problem into different constrained sub-problems. We first give a brief introduction to the ADMM method by considering a constrained optimization problem,

$$\min_{\mathbf{Z}\in\mathbb{R}^N,\mathbf{C}\in\mathbb{R}^M} f(\mathbf{Z})+g(\mathbf{C}), \qquad \text{s.t.} \qquad \mathbf{Z}=\mathbf{U}\mathbf{C}, \tag{9}$$

where $\mathbf{U}\in\mathbb{R}^{M\times N}$ and $f:\mathbb{R}^N\to\mathbb{R}$, $g:\mathbb{R}^M\to\mathbb{R}$. The basic ADMM regime is shown in Algorithm 1, where t denotes the iteration number.

Algorithm 1 The ADMM Algorithm.

Require: $\mathbf{C}$ and $\mathbf{Z}$.

1: Initial $\mu>0$, $\mathbf{J}$
2: **for** $t=0$ **to** Max-Iter **do**
3: $\quad\mathbf{Z}^{t+1}=\arg\min_{\mathbf{Z}} f(\mathbf{Z})+\frac{\mu}{2}||\mathbf{Z}-\mathbf{U}\mathbf{C}^t-\mathbf{J}^t||_2^2$.
4: $\quad\mathbf{C}^{t+1}=\arg\min_{\mathbf{C}} g(\mathbf{C})+\frac{\mu}{2}||\mathbf{Z}^{t+1}-\mathbf{U}\mathbf{C}-\mathbf{J}^t||_2^2$.
5: $\quad\mathbf{J}^{t+1}=\mathbf{J}^t-(\mathbf{Z}^{t+1}-\mathbf{U}\mathbf{C}^{t+1})$.
6: **end for**

Now, let us come back to Eq. (8) and invoke ADMM to solve it. We first translate Eq. (8) into an equivalent constrained form by introducing an auxiliary variable $\mathbf{Z}$,

$$\hat{\mathbf{C}}=\arg\min_{\mathbf{C},\mathbf{Z}}\frac{1}{2}||\mathbf{Y}_G-\mathbf{H}_G\mathbf{Z}||_F^2+\lambda||\boldsymbol{\alpha}_G||_0+\rho||\bar{\boldsymbol{\beta}}_G||_0, \quad \text{s.t.} \quad \mathbf{Z}=\mathbf{U}\mathbf{C}. \tag{10}$$

Through defining $f(\mathbf{Z})=\frac{1}{2}||\mathbf{Y}_G-\mathbf{H}_G\mathbf{Z}||_F^2$, $g(\mathbf{C})=\lambda||\boldsymbol{\alpha}_G||_0+\rho||\bar{\boldsymbol{\beta}}_G||_0$, and employing Line 3 in Algorithm 1, we have

$$\begin{aligned}\hat{\mathbf{Z}}^{t+1}&=\arg\min_{\mathbf{Z}} f(\mathbf{Z})+\frac{\mu}{2}||\mathbf{Z}-\mathbf{U}\mathbf{C}^t-\mathbf{J}^t||_F^2\\&=\arg\min_{\mathbf{Z}}\frac{1}{2}||\mathbf{Y}_G-\mathbf{H}_G\mathbf{Z}||_F^2+\frac{\mu}{2}\left|\left|\mathbf{Z}-[\mathbf{D}\ \mathbf{D}_G]\begin{bmatrix}\boldsymbol{\alpha}_G^t\\ \bar{\boldsymbol{\beta}}_G^t\end{bmatrix}-\mathbf{J}\right|\right|_F^2.\end{aligned} \tag{11}$$

By splitting the second term in Eq. (11), we have

$$\hat{\mathbf{Z}}^{t+1}=\arg\min_{\mathbf{Z}}\frac{1}{2}||\mathbf{Y}_G-\mathbf{H}_G\mathbf{Z}||_F^2+\frac{\mu_1}{2}||\mathbf{Z}-\mathbf{D}\boldsymbol{\alpha}_G^t-\mathbf{J}_1^t||_F^2+\frac{\mu_2}{2}||\mathbf{Z}-\mathbf{D}_G\bar{\boldsymbol{\beta}}_G^t-\mathbf{J}_2^t||_F^2, \tag{12}$$

where μ_1 and μ_2 are small positive constants. Instead of μ, we introduce μ_1 and μ_2 to make the solution of Eq. (12) more feasible.

Next, invoking Line 4 in Algorithm 1, we have

$$\begin{aligned}\mathbf{C}^{t+1}&=\arg\min_{\mathbf{C}} g(\mathbf{C})+\frac{\mu}{2}||\mathbf{Z}^{t+1}-\mathbf{U}\mathbf{C}-\mathbf{J}^t||_F^2\\&=\arg\min_{\boldsymbol{\alpha}_G,\bar{\boldsymbol{\beta}}_G}\lambda||\boldsymbol{\alpha}_G||_0+\rho||\boldsymbol{\beta}_G||_0+\frac{\mu_1}{2}||\mathbf{Z}^{t+1}-\mathbf{D}\boldsymbol{\alpha}_G^t-\mathbf{J}_1^t||_F^2+\frac{\mu_2}{2}||\mathbf{Z}^{t+1}-\mathbf{D}_G\bar{\boldsymbol{\beta}}_G^t-\mathbf{J}_2^t||_F^2.\end{aligned} \tag{13}$$

We now decouple the minimization problem of $\mathbf{C}$ in Eq. (13) with respect to $\boldsymbol{\alpha}_G$ and $\bar{\boldsymbol{\beta}}_G$, and solve them separately, i.e.,

$$\boldsymbol{\alpha}_G^{t+1} = \arg\min_{\boldsymbol{\alpha}} \lambda||\boldsymbol{\alpha}_G||_0 + \frac{\mu_1}{2}||\mathbf{Z}^{t+1} - \mathbf{D}\boldsymbol{\alpha}_G - \mathbf{J}_1^t||_F^2, \tag{14}$$

$$\bar{\boldsymbol{\beta}}_G^{t+1} = \arg\min_{\bar{\boldsymbol{\beta}}_G} \rho||\bar{\boldsymbol{\beta}}_G||_0 + \frac{\mu_2}{2}||\mathbf{Z}^{t+1} - \mathbf{D}_G\bar{\boldsymbol{\beta}}_G - \mathbf{J}_2^t||_F^2. \tag{15}$$

Following this, we update $\mathbf{J}^t$ by invoking line 5 in Algorithm 1,

$$\mathbf{J}_1^{t+1} = \mathbf{J}_1^t - (\mathbf{Z}^{t+1} - \mathbf{D}\boldsymbol{\alpha}_G^{t+1}), \tag{16}$$

$$\mathbf{J}_2^{t+1} = \mathbf{J}_2^t - (\mathbf{Z}^{t+1} - \mathbf{D}_G\bar{\boldsymbol{\beta}}_G^{t+1}). \tag{17}$$

In summary, it can be seen that the minimization of Eq. (10) involves three minimization sub-problems, i.e., $\mathbf{Z}$, $\boldsymbol{\alpha}_G$ and $\bar{\boldsymbol{\beta}}_G$ sub-problems. Fortunately, there is an efficient solution to each sub-problem, which will be discussed below. Furthermore, in image inpainting problem considered in this work, each problem can be solved patch by patch. Take the i-th patch $\boldsymbol{x}_i$ as an example, $\boldsymbol{y}_i = \mathbf{H}_i\boldsymbol{x}_i$, where $\mathbf{H}_i$ denotes the masks to kill or keep pixels in the i-th patch. In the PSR model, recall that $\boldsymbol{\alpha}_G$ is an expanded version of $\boldsymbol{\alpha}$ and we have $\boldsymbol{x}_i = \mathbf{D}\boldsymbol{\alpha}_i$. After this, the $\boldsymbol{\alpha}_i$ is solved, we can straightforwardly obtain $\boldsymbol{\alpha}_G$. In the GSR model, let $\tilde{\boldsymbol{\beta}}_i$ concatenate all the group coefficients including the i-th patch; we thus have $\boldsymbol{x}_i = \mathbf{D}_G\tilde{\boldsymbol{\beta}}_i$. In the following, we consider to solve the problem for each patch and the superscript t is omitted for conciseness. More specifically, we translate the $\boldsymbol{\alpha}_G$ sub-problem to $\{\boldsymbol{\alpha}_i\}_{i=1}^n$ subproblem and translate the $\bar{\boldsymbol{\beta}}_G$ sub-problem to $\{\boldsymbol{\beta}_{G_i}\}_{i=1}^n$ sub-problem and translate the $\mathbf{Z}$ sub-problem to $\{\boldsymbol{z}_i\}_{i=1}^n$ sub-problem.

4.2.1. $\mathbf{Z}$ Sub-problem

Given $\boldsymbol{\alpha}_G$ and $\bar{\boldsymbol{\beta}}_G$, $\mathbf{Z}$ sub-problem in Eq. (12) for each patch $\boldsymbol{z}_i$, becomes

$$\min_{\boldsymbol{z}_i} \ell_1(\boldsymbol{z}_i) = \min_{\boldsymbol{z}_i} \frac{1}{2}||\boldsymbol{y}_i - \mathbf{H}_i\boldsymbol{z}_i||_2^2 + \frac{\mu_1}{2}||\boldsymbol{z}_i - \mathbf{D}\boldsymbol{\alpha}_i - \boldsymbol{j}_{1,i}||_2^2 + \frac{\mu_2}{2}||\boldsymbol{z}_i - \mathbf{D}_G\tilde{\boldsymbol{\beta}}_i - \boldsymbol{j}_{2,i}||_2^2. \tag{18}$$

This is a quadratic form and it has a closed-form solution,

$$\hat{\boldsymbol{z}}_i = [\mathbf{H}_i^T\mathbf{H}_i + (\mu_1+\mu_2)\mathbf{I}]^{-1}[\mathbf{H}_i^T\boldsymbol{y}_i + \mu_1(\mathbf{D}\boldsymbol{\alpha}_i + \boldsymbol{j}_{1,i}) + \mu_2(\mathbf{D}_G\tilde{\boldsymbol{\beta}}_i + \boldsymbol{j}_{2,i})], \tag{19}$$

where $\mathbf{I}$ is an identity matrix with desired dimensions and $\{\boldsymbol{j}_{1,i}, \boldsymbol{j}_{2,i}\}$ are the corresponding elements from $\{\mathbf{J}_1, \mathbf{J}_2\}$, respectively.

Note that each $\boldsymbol{z}_i$ is *jointly* estimated in Eq. (19) using both PSR ($\boldsymbol{\alpha}_G$) and GSR ($\bar{\boldsymbol{\beta}}_G$) in one shot. This is different from the simple average of two results using PSR and GSR independently. In our experiments, we notice that this joint estimation plays a pivot role in the performance improvement of our proposed model.

4.2.2. α_G Sub-problem

Recall that $\boldsymbol{\alpha}_G$ is an expanded version of $\boldsymbol{\alpha}$, and thus the $\boldsymbol{\alpha}_G$ can be solved by the $\boldsymbol{\alpha}$ problem. According to Eq. (14), for i-th patch, $\boldsymbol{\alpha}$ sub-problem can be rewritten as

$$\min_{\boldsymbol{\alpha}_i} \ell_2(\boldsymbol{\alpha}_i) = \min_{\boldsymbol{\alpha}_i} \left(\frac{1}{2}||\mathbf{D}\boldsymbol{\alpha}_i - \boldsymbol{r}_i||_2^2 + \frac{\lambda}{\mu_1}||\boldsymbol{\alpha}_i||_0 \right), \tag{20}$$

where $\boldsymbol{r}_i = \boldsymbol{z}_i - \boldsymbol{j}_{1,i}$. This is a sparse representation problem, and we hereby directly solve the constrained form,

$$\min_{\boldsymbol{\alpha}_i} ||\boldsymbol{\alpha}_i||_0 \quad s.t. \quad ||\boldsymbol{r}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 \leq \delta, \tag{21}$$

where δ is a small constant, and apparently Eq. (21) can be efficiently solved by the orthogonal matching pursuit (OMP) algorithm (Tropp and Gilbert, 2007). Due to its effectiveness and efficiency, the KSVD algorithm (Aharon et al., 2006) is employed to learn the dictionary $\mathbf{D}$. For a given problem, OMP constructs a sparse solution via iteratively building up an approximation, rather than minimizing an objective function. The vector $\boldsymbol{r}_i$ in Eq. (21) is approximated as a linear combination of a few columns of $\mathbf{D}$, where the active set of columns to be used is built column by column, in a greedy fashion. At each iteration a new column is added to the active set the column that best correlates with the current residual. Although OMP is a heuristic method, in some cases it works marvelously (Donoho and Tsaig, 2008).

Algorithm 2 Image Inpainting Using JPG-SR Model.

Require: The observed image $\boldsymbol{y}$ and measurement matrix $\mathbf{H}$.

Initial t, $\mathbf{Z}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_G$, $\mathbf{J}$, b, m, W, μ_1, μ_2, ρ.
for $t = 0$ **to** Max-Iter **do**
  Update $\mathbf{Z}^{t+1}$ by Eq. (19);
  $\mathbf{R}^{t+1} = \mathbf{Z}^{t+1} - \mathbf{J}_1^t$;
  Construct dictionary $\mathbf{D}$ by $\mathbf{R}^{t+1}$ with KSVD.
  for Each patch $\boldsymbol{r}_i$ **do**
    Update $\boldsymbol{\alpha}_i^{t+1}$ by computing Eq. (21);
  end for
  $\mathbf{R}_G^{t+1} = \mathbf{Z}^{t+1} - \mathbf{J}_2^t$;
  for Each group $\mathbf{R}_{G_i}$ **do**
    Construct dictionary $\mathbf{D}_{G_i}$ from $\mathbf{R}_{G_i}^{t+1}$ using PCA;
    Update $\boldsymbol{\beta}_{G_i}^{t+1}$ by computing Eq. (27);
  end for
  Update $\boldsymbol{\alpha}^{t+1}$ by concatenating all $\boldsymbol{\alpha}_i$;
  Update $\mathbf{D}_G^{t+1}$ by concatenating all $\mathbf{D}_{G_i}$;
  Update $\boldsymbol{\beta}_G^{t+1}$ by concatenating all $\boldsymbol{\beta}_{G_i}$;
  Update $\mathbf{J}_1^{t+1}$ by Eq. (16);
  Update $\mathbf{J}_2^{t+1}$ by Eq. (17);
end for
Output: The final restored image $\hat{\boldsymbol{x}}$ by aggregating patches in $\mathbf{Z}$.

4.2.3. $\bar{\beta}_G$ Sub-problem

Given $\mathbf{Z}$, according to Eq. (15), $\bar{\boldsymbol{\beta}}_G$ sub-problem can be rewritten as

$$\min_{\bar{\boldsymbol{\beta}}_G} \ell_3(\bar{\boldsymbol{\beta}}_G) = \min_{\bar{\boldsymbol{\beta}}_G} \frac{1}{2}||\mathbf{D}_G\bar{\boldsymbol{\beta}}_G - \mathbf{R}_G||_F^2 + \frac{\rho}{\mu_2}||\bar{\boldsymbol{\beta}}_G||_0) \tag{22}$$

where $\mathbf{R}_G = \mathbf{Z} - \mathbf{J}_2$.

Figure 2: Test images used in the experiment. Top row: from left to right, Cowboy, Light, Mickey, House, Leaves, Peppers, Fence; bottom row: from left to right, Butterfly, Lena, Corn, Zebra, Barbara, Starfish, Tower.

Recalling the relationship of $\bar{\boldsymbol{\beta}}_G$, $\tilde{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}$, for each patch, we can get the other two after solving any one of them. Now, instead of considering each patch as in the $\boldsymbol{\alpha}$ sub-problem, we consider each *patch group* here. For i-th patch group, we aim to solve

$$\hat{\boldsymbol{\beta}}_{G_i} = \arg\min_{\boldsymbol{\beta}_{G_i}} \left(\frac{1}{2} ||\mathbf{R}_{G_i} - \mathbf{D}_{G_i}\boldsymbol{\beta}_{G_i}||_F^2 + \frac{\rho}{\mu_2}||\boldsymbol{\beta}_{G_i}||_0 \right). \tag{23}$$

One important issue of solving sub-problem $\boldsymbol{\beta}_G$ is the selection of the dictionary. To adapt to the local image structures, instead of learning an over-complete dictionary for each group as in Mairal et al. (2009), we learn the principle component analysis (PCA) based dictionary (Dong et al., 2011) for each group. Due to the orthogonality of the dictionary $\mathbf{D}_{G_i}$ and based on the orthogonal invariance, Eq. (23) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{\beta}}_{G_i} &= \min_{\boldsymbol{\beta}_{G_i}} \left(\frac{1}{2} ||\boldsymbol{\gamma}_{G_i} - \boldsymbol{\beta}_{G_i}||_F^2 + \frac{\rho}{\mu_2}||\boldsymbol{\beta}_{G_i}||_0 \right) \\ &= \min_{\boldsymbol{\beta}_i} \left(\frac{1}{2} ||\boldsymbol{\gamma}_i - \boldsymbol{\beta}_i||_2^2 + \frac{\rho}{\mu_2}||\boldsymbol{\beta}_i||_0 \right), \end{aligned} \tag{24}$$

where $\mathbf{R}_{G_i} = \mathbf{D}_{G_i}\boldsymbol{\gamma}_{G_i}$, and $\{\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i\}$ denote the vectorization form of the matrix $\{\boldsymbol{\gamma}_{G_i}, \boldsymbol{\beta}_{G_i}\}$, respectively.

In order to obtain the solution of Eq. (24) effectively, we have the following Lemma.

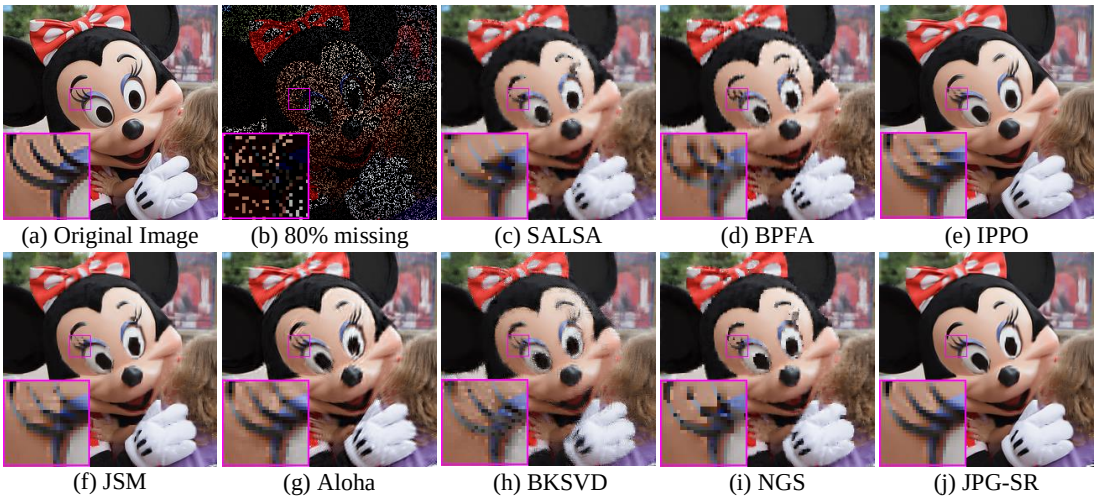
Lemma 1 *The minimization problem*

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left(\frac{1}{2} ||\mathbf{x} - \mathbf{a}||_2^2 + \tau ||\mathbf{x}||_0 \right) \tag{25}$$

has a closed-form solution

$$\hat{\mathbf{x}} = \text{hard}(\mathbf{a}, \sqrt{2\tau}) = \mathbf{a} \odot \mathbf{1}(\text{abs}(\mathbf{a}) - \sqrt{2\tau}), \tag{26}$$

where $\odot$ *denotes the element-wise (Hadamard) product,* abs( ) *calculates the absolute value of each entry (element-wise), and* $\mathbf{1}(\cdot)$ *denotes indicator function, i.e.,* $\mathbf{1}(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & otherwise \end{cases}$.



(a) Original Image (b) 80% missing (c) SALSA (d) BPFA (e) IPPO

(f) JSM (g) Aloha (h) BKSVD (i) NGS (j) JPG-SR

Figure 3: Visual comparison of *Mickey* by image inpainting with 80% missing pixels. (a) Original image; (b) Degraded image with 80% pixels missing; (c) SALSA (PSNR= 24.46dB); (d) BPFA (PSNR= 24.53dB); (e) IPPO (PSNR= 26.33dB); (f) JSM (PSNR= 26.09dB); (g) Aloha (PSNR=25.33dB); (h) BKSVD (PSNR= 23.72dB); (i) NGS (PSNR= 24.50dB); (j) JPG-SR (PSNR= **26.80dB**).

Proof See Afonso et al. (2010). ■

Therefore, based on Lemma 1, the closed-form solution of Eq. (24) is

$$\hat{\boldsymbol{\beta}}_i = \text{hard}(\boldsymbol{\gamma}_i, \sqrt{2\rho/\mu_2}) = \boldsymbol{\gamma}_i \odot \mathbf{1}(\text{abs}(\boldsymbol{\gamma}_i) - \sqrt{2\rho/\mu_2}). \tag{27}$$

This process is performed across all n patch groups to achieve $\boldsymbol{\beta}_G$, which is the final solution for $\bar{\boldsymbol{\beta}}_G$ sub-problem in Eq. (15).

After solving the above three sub-problems, we summarize the overall algorithm to solve Eq. (8) in Algorithm 2.

5. Experimental Results

In order to demonstrate the effectiveness of the proposed JPG-SR, we report the performance of JPG-SR for image inpainting and compare it with several state-of-the-art methods, including SALSA (Afonso et al., 2011), BPFA (Zhou et al., 2012), IPPO (Ram et al., 2013), JSM (Zhang et al., 2014b), Aloha (Jin and Ye, 2015), BKSVD (Serra et al., 2017) and NGS (Liu et al., 2017) methods. Note that the BKSVD method is the classical patch-based sparse representation method and JSM method is based on group sparse representation method. The experimental images are shown in Fig. 2. Throughout the numerical experiments, we choose the following stoping criterion of iteration for the proposed inpainting algorithm, i.e.,

$$\frac{||\hat{\boldsymbol{x}}^t - \hat{\boldsymbol{x}}^{t-1}||_2^2}{||\hat{\boldsymbol{x}}^{t-1}||_2^2} < \epsilon, \tag{28}$$

where ϵ is a small constant. The source code of the proposed JPG-SR for image inpainting can be downloaded at: https://drive.google.com/open?id=1Bj-UHVCQWMmJUw4NnV1hp11_IDFh5Og3.

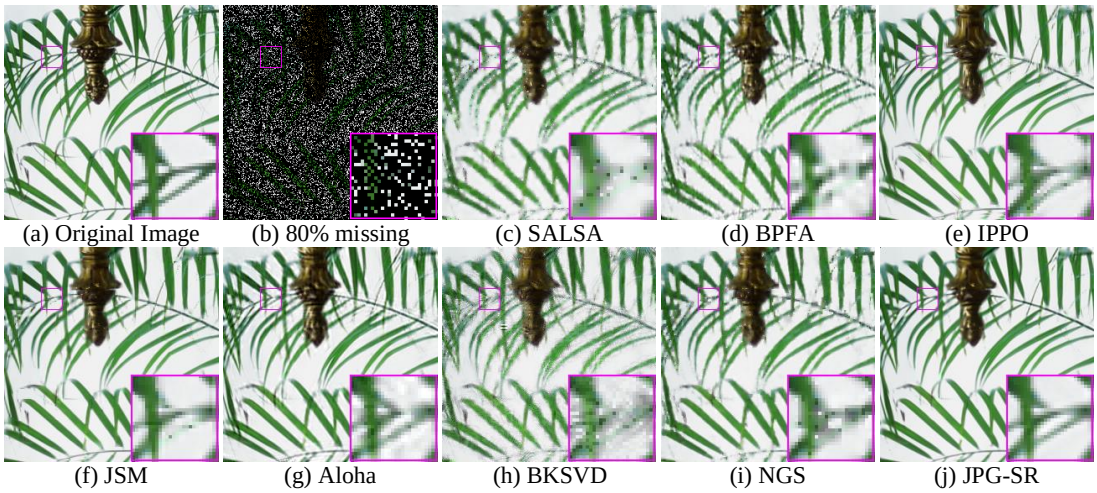


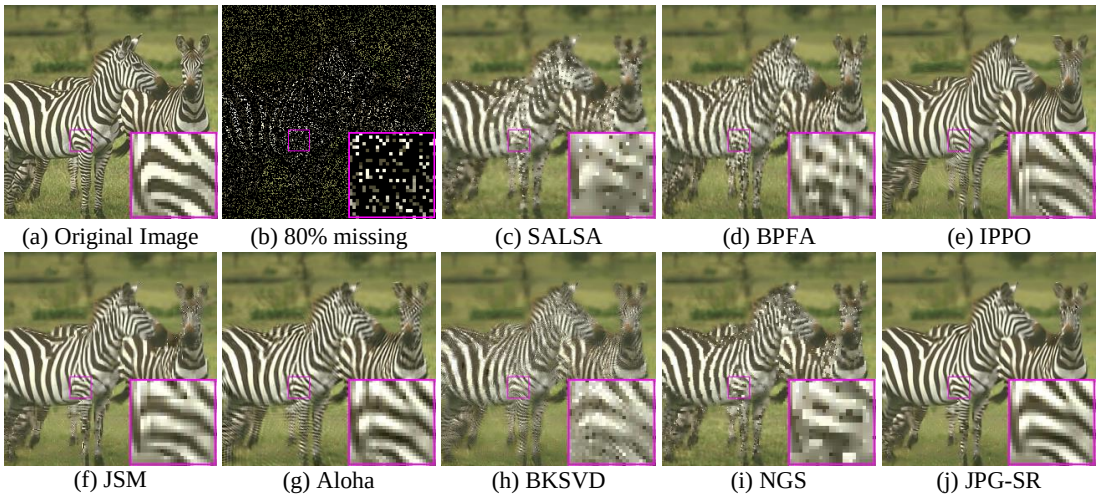
Figure 4: Visual comparison of *Leaves* by image inpainting with 80% missing pixels. (a) Original image; (b) Degraded image with 80% pixels missing; (c) SALSA (PSNR= 22.03dB); (d) BPFA (PSNR= 23.78dB); (e) IPPO (PSNR= 25.56dB); (f) JSM (PSNR= 26.18dB); (g) Aloha (PSNR=25.90dB); (h) BKSVD (PSNR= 22.05dB); (i) NGS (PSNR= 23.87dB); (j) JPG-SR (PSNR= **27.38dB**).

5.1. Comparison with State-of-the-art Methods

In this subsection, two interesting examples of image inpainting with different masks are conducted, i.e., partial random samples and text inlayed samples. The parameters are set as follows. The size of each patch $\sqrt{b} \times \sqrt{b}$ is set to be 8×8. The size of searching window $W \times W$ is set to 25×25 and the matched patch number in each group $m = 60$. (μ_1, μ_2, ϵ) are set to (0.003, 0.07, 0.0011), (0.005, 0.1, 0.00092), (0.01, 0.09, 0.0011), (0.03, 0.09, 0.0014) and (0.005, 0.1, 0.0005) when 80%, 70%, 60%, 50% pixels missing, and text inlayed, respectively.

Table 1 lists the PSNR comparison results for the 14 color images using all competing methods. It can be seen that the proposed JPG-SR consistently outperforms other competing methods in terms of PSNR metric (the only exception is the image *House* with 50% pixels missing for which JSM and Aloha slightly outperform JPG-SR by below 0.1dB). The average gains of the proposed JPG-SR over SALSA, BPFA, IPPO, JSM, Aloha, BKSVD and NGS methods are as much as 4.73dB, 2.36dB, 0.95dB, 1.04dB, 1.34dB, 3.31dB and 3.15dB, respectively.

The visual comparisons of images *Mickey*, *Leaves*, *Zebra*, *Tower* and *Light* with 80% pixels missing are shown in Fig. 3, Fig. 4, Fig. 5, Fig. 6 and Fig. 7, respectively. Obviously, it can be seen that SALSA and BKSVD could not reconstruct sharp edges and fine details. The BPFA, IPPO, JSM, Aloha and NGS methods produce images with a much better visual quality than SALSA and BKSVD, but still suffer from some undesirable artifacts, such as the ringing effects. Note that BKSVD produces the obvious undesirable visual artifacts since it is a PSR method. Although JSM method can obtain more visual results than BKSVD, it often generates over-smooth phenomena. By contrast, the proposed JPG-SR not only preserves sharp edges and fine details, but also eliminates the ringing effects.

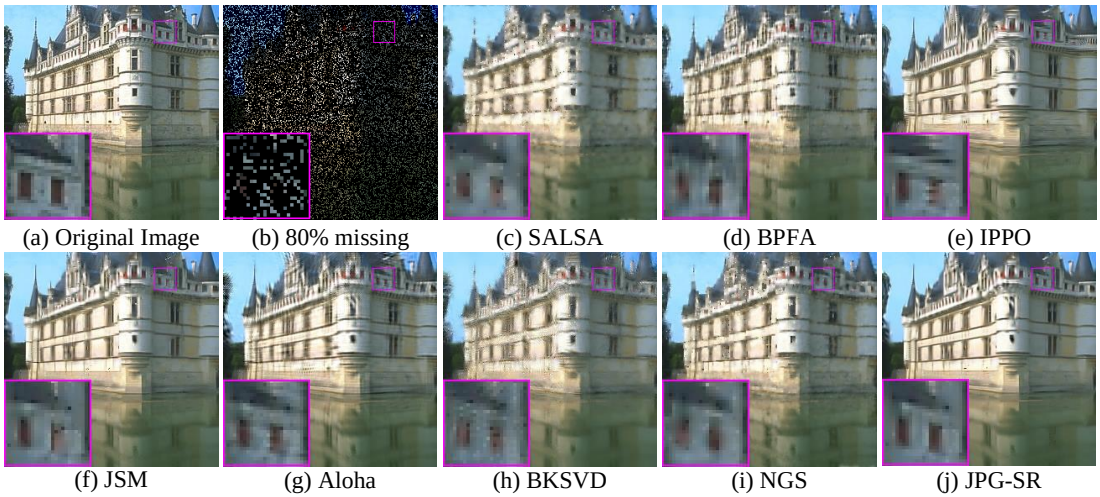


(a) Original Image (b) 80% missing (c) SALSA (d) BPFA (e) IPPO

(f) JSM (g) Aloha (h) BKSVD (i) NGS (j) JPG-SR

Figure 5: Visual comparison of *Zebra* by image inpainting with 80% missing pixels. (a) Original image; (b) Degraded image with 80% pixels missing; (c) SALSA (PSNR= 19.68dB); (d) BPFA (PSNR= 20.90dB); (e) IPPO (PSNR= 22.71dB); (f) JSM (PSNR= 21.88dB); (g) Aloha (PSNR=22.72dB); (h) BKSVD (PSNR= 19.37dB); (i) NGS (PSNR= 20.49dB); (j) JPG-SR (PSNR= **23.09dB**).

Therefore, these experimental findings clearly demonstrate that the effectiveness of the proposed JPG-SR model.



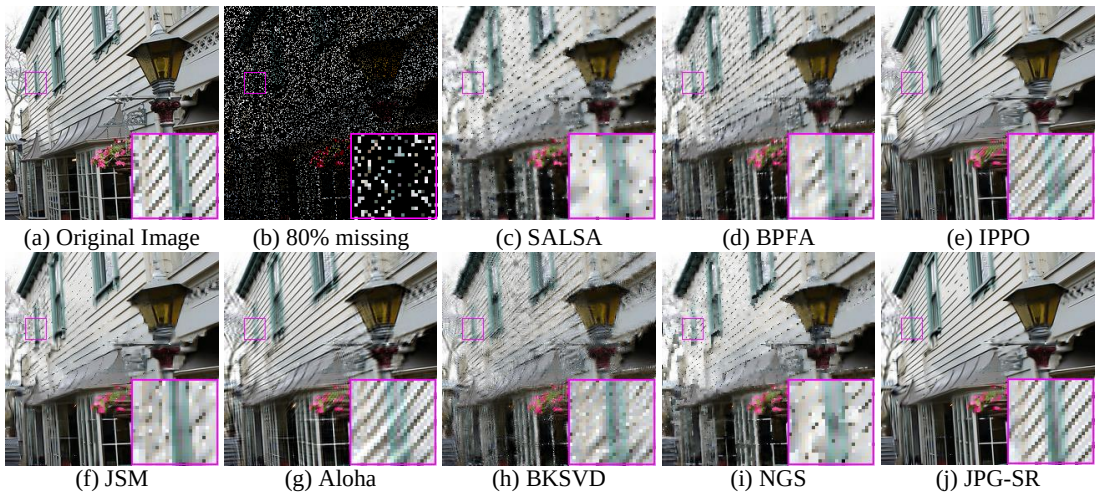
(a) Original Image (b) 80% missing (c) SALSA (d) BPFA (e) IPPO

(f) JSM (g) Aloha (h) BKSVD (i) NGS (j) JPG-SR

Figure 6: Visual comparison of *Tower* by image inpainting with 80% missing pixels. (a) Original image; (b) Degraded image with 80% pixels missing; (c) SALSA (PSNR= 22.85dB); (d) BPFA (PSNR= 23.94dB); (e) IPPO (PSNR= 24.50dB); (f) JSM (PSNR= 24.59dB); (g) Aloha (PSNR=23.88dB); (h) BKSVD (PSNR= 22.92dB); (i) NGS (PSNR= 23.47dB); (j) JPG-SR (PSNR= **24.87dB**).

Table 1: PSNR (dB) comparison of SALSA, BPFA, IPPO, JSM, ALoha, BKSVD, NGS and JPG-SR for image inpainting.

Miss pixels	Methods	Barbara	Butterfly	Fence	House	Cowboy	Light	Mickey	Leaves	Lena	Peppers	Starfish	Tower	Corn	Zebra	Average
80%	SALSA	22.62	22.85	21.80	29.54	23.72	18.27	24.46	22.03	28.20	28.58	25.70	22.85	24.28	19.68	23.90
	BPFA	25.11	24.04	26.24	30.80	24.93	19.23	24.53	23.78	29.50	29.58	26.79	23.94	25.54	20.90	25.35
	IPPO	28.32	25.13	27.98	33.65	25.38	21.49	26.33	25.56	30.64	30.48	26.30	24.50	25.14	22.71	26.68
	JSM	26.95	25.57	28.59	34.31	25.40	20.23	26.09	26.18	30.46	30.48	27.07	24.59	25.58	21.88	26.67
	Aloha	29.59	24.88	28.88	33.79	25.06	21.50	25.33	25.90	30.89	29.95	26.33	23.88	25.60	22.72	26.73
	BKSVD	25.21	22.00	24.20	28.15	24.12	18.77	23.72	22.05	28.16	27.75	25.36	22.92	23.69	19.37	23.96
	NGS	23.88	23.85	25.26	31.34	24.21	18.52	24.50	23.87	28.87	29.35	26.17	23.47	24.74	20.49	24.89
	JPG-SR	**30.11**	**26.58**	**29.70**	**35.05**	**25.67**	**22.13**	**26.80**	**27.38**	**31.25**	**31.35**	**27.80**	**24.87**	**26.47**	**23.09**	**27.73**
70%	SALSA	23.38	25.06	23.57	31.58	25.70	19.32	25.98	24.36	28.82	30.36	27.55	24.22	26.11	21.41	25.53
	BPFA	28.32	26.68	28.87	33.75	26.76	21.58	26.16	26.98	31.62	31.74	28.93	25.66	27.82	22.78	27.69
	IPPO	30.89	27.68	30.08	36.64	27.40	23.47	28.59	28.58	32.97	33.05	28.91	26.11	27.77	24.76	29.07
	JSM	30.48	27.97	30.46	36.71	27.11	23.12	28.25	29.28	32.69	33.47	29.36	26.64	27.66	23.95	29.08
	Aloha	32.40	27.29	30.57	36.68	27.24	23.17	27.11	29.04	32.80	32.76	28.22	25.77	27.95	24.55	28.97
	BKSVD	27.58	25.00	28.35	32.93	25.99	20.85	26.17	25.29	30.96	30.96	27.79	25.07	25.83	23.06	26.85
	NGS	26.11	26.36	27.32	33.91	26.19	20.78	26.68	26.44	30.77	31.59	28.35	25.22	26.77	22.71	27.09
	JPG-SR	**33.42**	**29.26**	**31.54**	**37.02**	**27.78**	**24.28**	**29.06**	**30.75**	**33.39**	**34.37**	**30.21**	**26.94**	**28.88**	**25.09**	**30.14**
60%	SALSA	24.57	26.79	25.45	32.76	26.99	20.49	27.41	26.29	31.49	32.18	29.09	25.73	27.75	22.80	27.13
	BPFA	31.06	28.88	30.79	36.40	28.42	23.65	27.83	29.83	33.54	34.20	30.98	27.28	30.07	24.53	29.82
	IPPO	33.55	29.85	32.14	38.25	29.58	25.13	30.76	30.88	34.89	35.16	31.09	27.81	29.75	26.79	31.12
	JSM	33.21	29.83	32.23	38.55	28.89	24.83	29.85	31.47	34.56	35.47	31.40	28.09	29.45	25.90	30.98
	Aloha	35.13	29.16	32.33	38.68	28.92	24.47	28.59	31.41	34.72	35.00	30.19	27.16	29.83	26.24	30.85
	BKSVD	29.86	27.70	30.72	33.48	28.14	23.00	28.53	28.61	33.48	33.44	29.99	26.68	28.35	25.27	29.09
	NGS	28.24	28.37	30.11	36.29	27.78	22.78	28.09	28.87	32.81	33.59	30.26	27.04	28.55	24.39	29.08
	JPG-SR	**35.76**	**31.23**	**33.12**	**38.82**	**29.71**	**25.73**	**31.15**	**33.08**	**35.46**	**36.25**	**32.35**	**28.58**	**30.86**	**27.09**	**32.09**
50%	SALSA	25.66	28.52	27.25	35.17	28.59	21.47	28.98	28.11	33.08	34.01	30.90	27.01	29.39	24.42	28.75
	BPFA	34.01	30.98	32.82	39.24	30.21	25.71	29.43	32.79	35.61	36.44	33.13	28.83	32.10	26.37	31.98
	IPPO	35.91	31.69	33.95	40.02	31.30	26.70	32.74	33.32	36.50	36.91	33.10	29.57	31.76	28.42	32.99
	JSM	35.87	31.47	33.75	**40.53**	30.75	26.48	31.96	33.78	36.39	37.35	33.24	29.48	31.33	27.77	32.87
	Aloha	37.46	30.78	33.79	**40.58**	30.46	25.84	30.33	34.01	36.41	36.88	31.85	28.71	31.89	27.67	32.62
	BKSVD	33.58	29.64	32.44	36.50	29.75	24.68	29.95	31.25	35.44	35.87	31.99	28.27	30.28	26.97	31.19
	NGS	30.93	30.28	32.00	38.85	29.32	24.62	29.75	31.23	34.56	35.59	32.10	28.53	30.31	26.03	31.01
	JPG-SR	**37.76**	**32.89**	**34.63**	40.51	**31.85**	**27.51**	**33.39**	**35.48**	**37.19**	**37.87**	**34.29**	**30.15**	**33.00**	**29.10**	**33.97**
Inlay text	SALSA	29.18	29.81	26.77	36.70	30.17	24.94	30.67	29.03	33.84	36.16	32.60	28.75	30.96	25.19	30.34
	BPFA	34.27	31.71	32.23	39.24	31.13	28.60	31.70	31.78	35.27	37.50	33.88	30.94	32.16	27.04	32.68
	IPPO	37.65	33.98	35.10	41.06	32.61	29.92	34.04	35.26	37.29	39.42	35.35	31.91	32.48	29.99	34.72
	JSM	37.79	33.19	35.41	41.21	32.42	29.65	32.99	35.40	36.98	39.27	35.17	32.48	32.26	29.11	34.52
	Aloha	39.16	31.58	34.94	41.37	30.94	28.38	30.49	34.74	36.03	37.40	32.06	30.34	32.04	28.87	33.45
	BKSVD	35.16	29.09	31.69	37.06	31.20	27.77	31.43	29.74	34.66	34.90	32.83	30.35	29.70	27.91	31.68
	NGS	33.57	31.78	28.73	35.95	30.80	27.26	31.10	30.05	34.71	35.61	33.02	30.21	31.65	26.24	31.48
	JPG-SR	**39.28**	**34.56**	**36.61**	**41.56**	**33.06**	**30.31**	**34.66**	**36.32**	**37.36**	**39.71**	**36.03**	**32.98**	**32.87**	**30.20**	**35.39**



(a) Original Image (b) 80% missing (c) SALSA (d) BPFA (e) IPPO

(f) JSM (g) Aloha (h) BKSVD (i) NGS (j) JPG-SR

Figure 7: Visual comparison of *Light* by image inpainting with 80% missing pixels. (a) Original image; (b) Degraded image with 80% pixels missing; (c) SALSA (PSNR= 18.27dB); (d) BPFA (PSNR= 19.23dB); (e) IPPO (PSNR= 21.49dB); (f) JSM (PSNR= 20.23dB); (g) Aloha (PSNR= 21.50dB); (h) BKSVD (PSNR= 18.77dB); (i) NGS (PSNR= 18.52dB); (j) JPG-SR (PSNR= **22.13dB**).

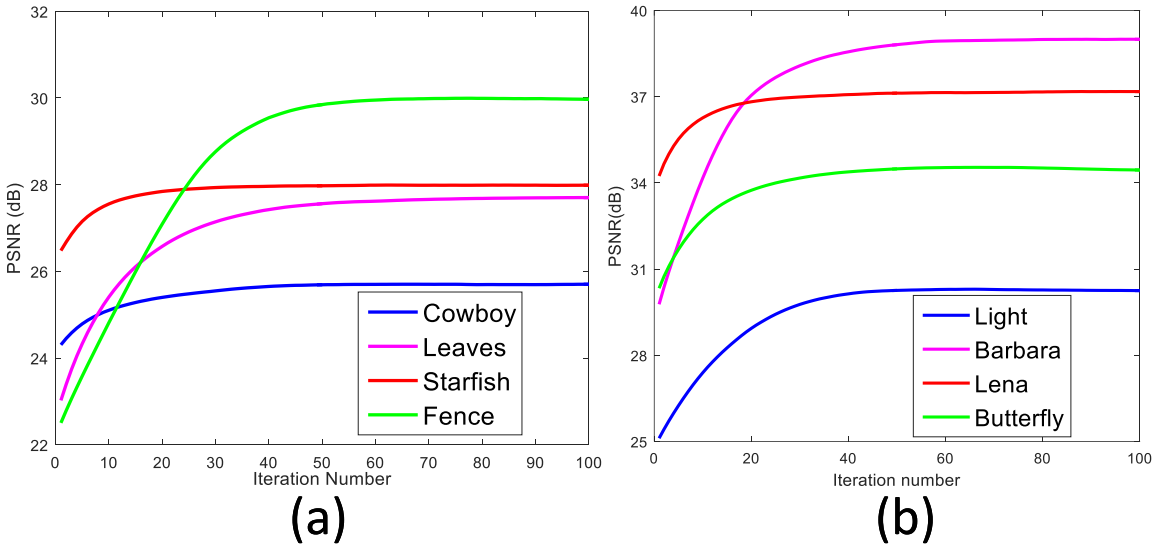


Figure 8: Convergence analysis. (a) PSNR results versus iteration number for image inpainting with 80% pixels missing. (b) PSNR results versus iteration number for image inpainting with text inlayed.

5.2. Convergence

Since the proposed model is non-convex, it is difficult to give its theoretical proof for global convergence. Hereby, we provide the empirical evidence to illustrate the good convergence behavior of our proposed model. Fig. 8 illustrates the convergence performance of the proposed scheme. It shows the curves of the PSNR values versus the iteration numbers for image inpainting with 80% pixels missing and text inlayed, respectively. It can be seen that with the increase of iteration number, the PSNR curves of the reconstructed images gradually increase and eventually become flat and stable. Obviously, the proposed scheme enjoys a good convergence performance.

6. Conclusion

A new sparse representation model, called joint patch-group based sparse representation (JPG-SR) has been proposed in this paper. Compared with existing sparse representation models, the proposed JPG-SR integrates both local sparsity and nonlocal self-similarity of the image. We have developed an iterative algorithm based on the ADMM framework to solve the proposed model for image inpainting. Experimental results have demonstrated that the proposed model is efficient and outperforms many state-of-the-art methods both quantitatively and qualitatively.

References

Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9):2345–2356, 2010.

Manya V Afonso, José M Bioucas-Dias, and Mário AT Figueiredo. An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems. *IEEE Transactions on Image Processing*, 20(3):681–695, 2011.

Michal Aharon, Michael Elad, and Alfred Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

Cesar F Caiafa, Olaf Sporns, Andrew Saykin, and Franco Pestilli. Unified representation of tractography and diffusion-weighted mri data using sparse multidimensional arrays. In *Advances in Neural Information Processing Systems*, pages 4340–4351, 2017.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011.

D. L. Donoho and Y. Tsaig. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, Nov 2008. ISSN 0018-9448. doi: 10.1109/TIT.2008.929958.

Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

Michael Elad and Irad Yavneh. A plurality of sparse representations is better than the sparsest one alone. *IEEE Transactions on Information Theory*, 55(10):4701–4714, 2009.

Bingsheng He, Li-Zhi Liao, Deren Han, and Hai Yang. A new inexact alternating directions method for monotone variational inequalities. *Mathematical Programming*, 92(1):103–118, 2002.

Kyong Hwan Jin and Jong Chul Ye. Annihilating filter-based low-rank hankel matrix approach for image inpainting. *IEEE Transactions on Image Processing*, 24(11):3498–3511, 2015.

James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

Mading Li, Jiaying Liu, Zhiwei Xiong, Xiaoyan Sun, and Zongming Guo. Marlow: A joint multiplanar autoregressive and low-rank approach for image completion. In *European Conference on Computer Vision*, pages 819–834. Springer, 2016.

Hangfan Liu, Ruiqin Xiong, Xinfeng Zhang, Yongbing Zhang, Siwei Ma, and Wen Gao. Nonlocal gradient sparsity regularization for image restoration. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(9):1909–1921, 2017.

Weiwei Liu and Ivor W Tsang. Making decision trees feasible in ultrahigh feature and label dimensions. *The Journal of Machine Learning Research*, 18(1):2814–2849, 2017.

Daniel PK Lun et al. Robust fringe projection profilometry via sparse representation. *IEEE Transactions on Image Processing*, 25(4):1726–1739, 2016.

Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

Idan Ram, Michael Elad, and Israel Cohen. Image processing using smooth ordering of its patches. *IEEE transactions on image processing*, 22(7):2764–2774, 2013.

Juan G Serra, Matteo Testa, Rafael Molina, and Aggelos K Katsaggelos. Bayesian k-svd using fast variational inference. *IEEE Transactions on Image Processing*, 26(7):3344–3359, 2017.

Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

Zhiyuan Zha, Xin Liu, Xiaohua Huang, Henglin Shi, Yingyue Xu, Qiong Wang, Lan Tang, and Xinggan Zhang. Analyzing the group sparsity based on the rank minimization methods. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 883–888. IEEE, 2017.

Jian Zhang, Debin Zhao, and Wen Gao. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*, 23(8):3336–3351, 2014a.

Jian Zhang, Debin Zhao, Ruiqin Xiong, Siwei Ma, and Wen Gao. Image restoration using joint statistical modeling in a space-transform domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):915–928, 2014b.

Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698. IEEE, 2010.

Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*, 21(1):130–144, 2012.