



A support-denoiser-driven framework for single image restoration



Liangtian He^a, Yilun Wang^b, Shaobing Gao^{c,*}

^a Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Mathematical Sciences, Anhui University, Hefei, 230601, PR China

^b American Family Insurance, Chicago, IL, 60654, United States

^c College of Computer Science, Sichuan University, Chengdu, 610065, PR China

ARTICLE INFO

Article history:

Received 6 May 2020

Received in revised form 14 January 2021

Keywords:

Wavelet tight frame
Nonlocal patched denoiser
Hybrid model
Single image restoration
Plug-and-play ADMM

ABSTRACT

Model-based methods have been the powerful strategies for solving a variety of imaging inverse problems. Particularly, the sparsity-enforcing regularization models have been especially widely investigated and adopted over the past decades. Along this research direction, one of the most important topics is the model formulation that is able to incorporate the more suitable image priors. In this paper, we propose a universal and flexible image restoration model that exploits the *local* sparsity, support and *nonlocal* denoiser priors simultaneously. While the proposed model is nonconvex as a whole, we show that it can be naturally tackled via a multi-stage convex relaxation procedure based on an extended alternating direction method of multiplier (ADMM) algorithm. Comprehensive numerical experiments demonstrate the effectiveness of our proposed algorithm over many existing state-of-the-art methods, in both objective and perceptual quality.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Image restoration including image denoising, deblurring, inpainting, etc., is aiming to recover the underlying true image from its degraded observation. Mathematically, the degraded process is often formulated by applying a degrade operator to the clear true image as below.

$$\mathbf{f} = \mathbf{A}\mathbf{u}^* + \omega. \quad (1)$$

where \mathbf{A} is an ill-posed matrix representing a degradation operator, and image is assumed to be composed of N pixels. Let $\mathbf{u}^* \in \mathbb{R}^N$, $\mathbf{f} \in \mathbb{R}^N$ and $\omega \in \mathbb{R}^N$ be lexicographically stacked representations of the original clear image, the corrupted image and the additive noise,¹ respectively.

Over the past decades, a large number of effective algorithms have been proposed to solve the ill-posed problem (1), and these algorithms can be broadly categorized as model-based optimization methods and discriminative learning methods [1]. Specifically, the model-based methods obtain the results by solving some optimization problems, which usually involve an iterative loop [2–10]. In contrast, the discriminative learning methods first learn the prior parameters on a training set containing degraded and clear image pairs, then infer the underlying image with a predefined nonlinear

* Corresponding author.

E-mail addresses: helt@ahu.edu.cn (L. He), yilun.wang@gmail.com (Y. Wang), gaoshaobing@scu.edu.cn (S. Gao).

¹ We restrict our discussion to the Gaussian noise with entries drawn at random from the normal distribution $\mathcal{N}(0, \sigma^2)$.

mapping function [11–13]. One major difference between these two categories is that the former can be extended to solve the various image restoration tasks by altering the degradation operator in (1), while the later can be only tailored to a specialized task. In a nutshell, while the discriminative learning methods enjoy a fast testing speed and often deliver a more pleasing output, they commonly lack the flexibility and generality. By contrast, the model-based methods have a promising advantage of flexibility, as well as a highly competitive performance. Therefore, we restrict our attention to model-based methods in this work, since we aim to provide a unified algorithm for multiple image restoration tasks solely based on a single degraded image.

Among the various image restoration applications, image denoising is often considered as a pre-processing step for other imaging tasks. While the denoising and other restoration scenarios (e.g., deblurring) look very similar, they are fundamentally different from the perspective of manifold [14]. In particular, many of the current state-of-the-art (SOTA) denoising approaches such as BM3D [15] and WNNM [16] denoisers are nonlocal patch-based and they are too complex and sophisticated to be easily formalized as the solutions of a certain explicit optimization problem. Therefore, one cannot directly exploit the virtue of these denoising algorithms for the model-based image restoration methods.

Very recently, there were several attempts to incorporate a denoiser prior into the model-based optimization methods to deal with other inverse imaging problems such as deblurring and inpainting. For example, Danielyan et al. [17] adapted the BM3D denoiser for image deblurring by adopting the augmented Lagrangian method. More recently, the denoising prior has been successfully incorporated into a plug-and-play ADMM algorithmic framework. This idea first appeared in [18] where a special variable splitting trick was utilized to make any existing Gaussian denoiser as one modular of the ADMM iterative scheme. In [19], Ono used the primal–dual algorithm to tackle the resulting optimization problem. Some of the other related works can be referred to [20–24] and the references therein. All of the methods mentioned above have shown that the denoiser prior is a good alternative to model image characteristic, and existing denoising algorithms can be employed to solve other image restoration tasks by decoupling the fidelity term and the regularization term.

There have been enduring efforts on constructing appropriate regularization models for effectively improving the restoration quality. In this paper, we move forward along this direction with the goal to design a model-based single image restoration method, which is capable of smoothing the recovered image so that the noise can be suppressed to the greatest extend while simultaneously preserving the critical features and tiny details. To the best of our knowledge, this is the first attempt incorporating the conventional sparsity prior with the *explicit* formulation, and the innovative denoiser prior with the *implicit* formulation into a unified model. Specifically, our main contributions are listed as follows.

- Compared with the conventional sparsity-only regularization models that often have very limited capability in texture recovery, the resulted hybrid model utilizing both the support prior and the denoiser prior is able to better preserve the complex structures in the restored image.
- In contrast to the standard plug-and-play scheme where a single regularizer term corresponding to a specified denoiser is applied, the additional sparsity and support priors demonstrate to be quite useful for preserving the sharp edges and homogeneous areas.
- Extensive experimental results on several typical image restoration problems demonstrate the advantages of the hybrid priors. Specifically, an efficient multi-stage convex relaxation algorithm is proposed and achieves a highly competitive (and often better) performance compared to many SOTA model-based methods.

The remainder of this paper is organized as follows. Section 2 briefly revisits the plug-and-play ADMM algorithm, as well as the support-driven sparsity regularization model. Section 3 introduces our proposed flexible image restoration model which simultaneously exploits the local sparsity, support and nonlocal denoiser priors. Moreover, a corresponding efficient algorithmic framework is also developed. Section 4 devotes the comprehensive experiments and the detailed discussions on several image restoration tasks. Section 5 concludes this paper and discusses some open questions for future developments.

2. Preliminaries

To make the paper self-contained, we first briefly introduce the ADMM algorithm, plug-and-play ADMM algorithm and support-driven sparsity regularization model that serve as the basis of our proposed method.

2.1. ADMM algorithm

The ADMM algorithm can efficiently solve a kind of linear inverse problem of the following form.

$$\underset{\mathbf{u}}{\operatorname{argmin}} h(\mathbf{u}) + \sum_{j=1}^J g_j(\mathbf{H}_j \mathbf{u}), \quad (2)$$

where h and $g_j, j = 1, 2, \dots, J$ are closed, proper, convex functions, and matrix $\mathbf{H}_j, j = 1, 2, \dots, J$ is the appropriate dimension [25]. Let us introduce an auxiliary variable and denote

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_J \end{bmatrix}, \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \vdots \\ \mathbf{v}_J \end{bmatrix}, g(\mathbf{v}) = \sum_{j=1}^J g_j(\mathbf{v}_j). \quad (3)$$

The iterative loop of ADMM is

$$\left\{ \begin{array}{l} \mathbf{u}^{k+1} = \operatorname{argmin}_{\mathbf{u}} h(\mathbf{u}) + \sum_{j=1}^J \|\mathbf{H}_j \mathbf{u} - \mathbf{v}_j^k - \mathbf{d}_j^k\|_2^2 \\ \mathbf{v}_1^{k+1} = \operatorname{argmin}_{\mathbf{v}_1} g_1(\mathbf{v}_1) + \frac{\mu}{2} \|\mathbf{H}_1 \mathbf{u}^{k+1} - \mathbf{v}_1 - \mathbf{d}_1^k\|_2^2 \\ \vdots \quad \vdots \quad \vdots \\ \mathbf{v}_J^{k+1} = \operatorname{argmin}_{\mathbf{v}_J} g_J(\mathbf{v}_J) + \frac{\mu}{2} \|\mathbf{H}_J \mathbf{u}^{k+1} - \mathbf{v}_J - \mathbf{d}_J^k\|_2^2 \\ \mathbf{d}^{k+1} = \mathbf{d}^k - \mathbf{H} \mathbf{u}^{k+1} + \mathbf{v}^{k+1}, \end{array} \right. \quad (4)$$

where $\mathbf{d} = [\mathbf{d}_1^T, \mathbf{d}_2^T, \dots, \mathbf{d}_J^T]^T$ are the scaled Lagrangian multipliers, μ is the penalty parameter.

Lemma 2.1 (Boyd et al. [25]). Consider an optimization problem of the form (2), where matrix \mathbf{H} has full column rank, and $h : \mathbb{R}^N \rightarrow \mathbb{R}$ and $g : \mathbb{R}^M \rightarrow \mathbb{R}$ are closed, proper, and convex functions. Let $\mathbf{v}^0, \mathbf{d}^0 \in \mathbb{R}^M$ and $\mu > 0$ be given. If the sequences $(\mathbf{u}^k, k = 0, 1, \dots)$, $(\mathbf{v}^k, k = 0, 1, \dots)$ and $(\mathbf{d}^k, k = 0, 1, \dots)$ are generated according to (4), $(\mathbf{u}^k, k = 0, 1, \dots)$ will converge to a solution of (2) if existing. Furthermore, if a solution does not exist, then at least one of the sequences $(\mathbf{v}^k, k = 0, 1, \dots)$ or $(\mathbf{d}^k, k = 0, 1, \dots)$ diverges.

2.2. Plug-and-play ADMM algorithm with denoiser prior

In [18], Venkatakrishnan et al. proposed a plug-and-play denoiser algorithmic framework based on the ADMM method. For the optimization model

$$\operatorname{argmin}_{\mathbf{u}} \lambda \mathcal{R}(\mathbf{u}) + \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{f}\|_2^2. \quad (5)$$

By introducing a special variable splitting technique $\mathbf{u} = \mathbf{v}$ in (5), one can obtain an equivalently constrained optimization form as follows.

$$(\mathbf{u}^*, \mathbf{v}^*) = \operatorname{argmin}_{\mathbf{u}, \mathbf{v}} \lambda \mathcal{R}(\mathbf{v}) + \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{f}\|_2^2, \quad \text{s.t. } \mathbf{u} = \mathbf{v}. \quad (6)$$

Then the optimum of (6) can be obtained using the ADMM iterative scheme.

$$\left\{ \begin{array}{l} \mathbf{u}^{k+1} = \operatorname{argmin}_{\mathbf{u}} \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{f}\|_2^2 + \frac{\mu}{2} \|\mathbf{u} - \mathbf{v}^k + \mathbf{d}^k\|_2^2 \\ \mathbf{v}^{k+1} = \operatorname{argmin}_{\mathbf{v}} \lambda \mathcal{R}(\mathbf{v}) + \frac{\mu}{2} \|\mathbf{u}^{k+1} - \mathbf{v} + \mathbf{d}^k\|_2^2 \\ \mathbf{d}^{k+1} = \mathbf{d}^k + \mathbf{u}^{k+1} - \mathbf{v}^{k+1}, \end{array} \right. \quad (7)$$

where \mathbf{d} is the scaled Lagrange multiplier. Specifically, the optimization problem (7) involving the regularization term can be rewritten as

$$\begin{aligned} \mathbf{v}^{k+1} &= \operatorname{argmin}_{\mathbf{v}} \mathcal{R}(\mathbf{v}) + \frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{v} - (\mathbf{u}^{k+1} + \mathbf{d}^k)\|_2^2 \\ &= D_\sigma(\mathbf{u}^{k+1} + \mathbf{d}^k, \sqrt{\lambda/\mu}), \end{aligned} \quad (8)$$

where D_σ is a Gaussian denoiser. From the Bayesian point of view, by treating $(\mathbf{u}^{k+1} + \mathbf{d}^k)$ as a noisy image with noise level $\sqrt{\lambda/\mu}$, then (8) can be viewed as a denoising problem. It is worth noting that the explicit form of $\mathcal{R}(\mathbf{u})$ is not required, and varied existing powerful denoisers (e.g., NLM and BM3D) without specified formulations can be easily plugged into this optimization based imaging model.

2.3. Support-driven sparsity regularization model

The ℓ_1 norm sparsity-enforcing regularization model has been one of most promising methods to solve the ill-conditioned imaging inverse problems.

$$\operatorname{argmin}_{\mathbf{u}} \lambda \|\Gamma \mathbf{u}\|_1 + \frac{1}{2} \|\mathbf{f} - \mathbf{A}\mathbf{u}\|_2^2, \quad (9)$$

where Γ is a sparsifying operator such that most entries of $\Gamma\mathbf{u}$ are zeros or close to zero. As we know, the most direct way to induce sparsity is to penalize the ℓ_0 pseudo-norm of the solution.

$$\underset{\mathbf{u}}{\operatorname{argmin}} \lambda \|\Gamma\mathbf{u}\|_0 + \frac{1}{2} \|\mathbf{f} - \mathbf{Au}\|_2^2. \quad (10)$$

However, the optimization problem (10) is NP-hard and computationally intractable. Thus ones often turn to ℓ_1 norm based model (9) for computational efficiency. However, it is well known that a gap exists between the ℓ_1 norm model and ℓ_0 pseudo-norm model in many imaging tasks such as image deblurring. Therefore, many alternatives have been proposed to bridge the gap between ℓ_1 norm and ℓ_0 pseudo-norm. In [26], Candes et al. proposed the weighted ℓ_1 regularization model.

$$\underset{\mathbf{u}}{\operatorname{argmin}} \lambda \|W(\Gamma\mathbf{u})\|_1 + \frac{1}{2} \|\mathbf{f} - \mathbf{Au}\|_2^2, \quad (11)$$

where W is a diagonal matrix, whose values are inversely proportional to the true element magnitude, i.e., $W_i = \frac{1}{|(\Gamma\mathbf{u})_i| + \epsilon}$, with added positive parameter ϵ to avoid dividing by zero. In the spirit of constructing a more effective yet simple weighted ℓ_1 model [27], He et al. proposed the truncated ℓ_1 regularization model [28].

$$\underset{\mathbf{u}}{\operatorname{argmin}} \lambda \|(\Gamma\mathbf{u})_T\|_1 + \frac{1}{2} \|\mathbf{f} - \mathbf{Au}\|_2^2, \quad (12)$$

where \mathbf{a}_T denotes the elements of \mathbf{a} indexed in T , i.e., $(\Gamma\mathbf{u})_T$ is a subvector of $\Gamma\mathbf{u}$ after truncation. T is the supplementary set of S , i.e., $T = S^C$, where S is the support index set of coefficients in the transform domain. While the true S or T is not known beforehand due to the ground truth image being unavailable in practice, we can estimate the support information based on a relatively high quality reference image, which could be a solution of an existing off-the-shelf image recovery method. Specifically, given a reference image \mathbf{u}^{ref} , the indexes of the support set can be roughly estimated as follows.

$$S := \left\{ i : |(\Gamma\mathbf{u}^{\text{ref}})_i| > \frac{\|\Gamma\mathbf{u}^{\text{ref}}\|_\infty}{\rho} \right\}, \quad (13)$$

where the thresholding parameter $\rho > 0$. Note that compared to the weighted ℓ_1 model, it adopts a specific yet effective 0-1 greedy weighting scheme in (12).

Lemma 2.2 (Zhang [29]). $\|\mathbf{x}\|_1$ is the tightest convex relaxation of $\|\mathbf{x}\|_0$.

Theorem 2.1. Given T , $\|\mathbf{x}_T\|_1$ is the tightest convex relaxation of $\|\mathbf{x}_T\|_0$.

Proof. Let us denote $\mathbf{x} = (x_1, x_2, \dots, x_N) \in \mathbb{R}^N$, $\|\mathbf{x}\|_0 = \#\{i \in \{1, 2, \dots, N\} : x_i \neq 0\}$. Without the loss of generality, we define the previous K elements as the nonzero values, i.e., $S = \{1, 2, \dots, K\}$. Therefore, $T = \{K+1, K+2, \dots, N\}$, $x_T = (x_{K+1}, x_{K+2}, \dots, x_N)$, and $\|\mathbf{x}_T\|_1 = \sum_{i=K+1}^N |x_i|$. According to Lemma 2.2, we can directly derive that $\|\mathbf{x}_T\|_1 = \sum_{i=K+1}^N |x_i|$ is the tightest convex relaxation of $\|\mathbf{x}_T\|_0 = \#\{i \in \{K+1, K+2, \dots, N\} : x_i \neq 0\}$, where $\#$ denotes the cardinality of a given set. \square

3. Proposed model using hybrid local and nonlocal image priors

3.1. Proposed model and algorithmic framework

The existing plug-and-play ADMM algorithm with nonlocal patched denoiser has demonstrated an impressive performance, especially for texture recovery. However, they may oversmooth the irregular structures and result in undesirable artifacts on the smooth regions in the case of the underlying image does not exhibit a self-similarity property. By contrast, the support-driven sparsity regularization methods are capable of preserving the salient edges well, while alleviating the unwanted artifacts on the smooth regions simultaneously. However, for the images of complex structures, due to the relative inefficiency of the local sparsifying operator to deal with heavy textures, the textures and tiny details are often falsely removed. Therefore, in the context of image modeling, both the local prior and nonlocal prior are beneficial to the final recovery performance in their own aspects. Thus it is desirable to exploit them together and strive to achieve a good balance between them.

In this work, we put forward a generic, simple, yet effective image restoration model integrating the *local* sparsity, support and *nonlocal* denoising priors. These priors work together and we can obtain some satisfactory recovered results.

$$\underset{\mathbf{u}}{\operatorname{argmin}} \lambda_1 \mathcal{R}(\mathbf{u}) + \lambda_2 \|(\Gamma\mathbf{u})_T\|_1 + \frac{1}{2} \|\mathbf{Au} - \mathbf{f}\|_2^2, \quad (14)$$

where $\mathcal{R}(\mathbf{u})$ is a regularizer term with no explicit formulation, Γ denotes a local sparsifying operator,² and $\|(\Gamma\mathbf{u})_T\|_1$ denotes the truncated ℓ_1 norm which takes support prior into consideration. Similar to (6), we associate an auxiliary

² We focus on the sparsity operator Γ as wavelet tight frame [30] ($\Gamma^T \Gamma = I$, where I denotes the unit matrix) throughout this paper, due to its simplicity and efficiency.

variable \mathbf{v}_1 with the image \mathbf{u} , in order to plug the denoiser into the optimization procedure. Moreover, we introduce another auxiliary variable $\Gamma\mathbf{u} = \mathbf{v}_2$, and obtain the following constraint optimization problem.

$$\underset{\mathbf{u}, \mathbf{v}_1, \mathbf{v}_2}{\operatorname{argmin}} \lambda_1 \mathcal{R}(\mathbf{v}_1) + \lambda_2 \|(\mathbf{v}_2)_T\|_1 + \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{f}\|_2^2, \quad \text{s.t. } \mathbf{u} = \mathbf{v}_1, \Gamma\mathbf{u} = \mathbf{v}_2. \quad (15)$$

Correspondingly, the ADMM iterative loop consists of the following steps until convergence.

$$\begin{cases} \mathbf{u}^{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{A}\mathbf{u} - \mathbf{f}\|_2^2 + \frac{\mu_1}{2} \|\mathbf{u} - \mathbf{v}_1^k + \mathbf{d}_1^k\|_2^2 + \frac{\mu_2}{2} \|\Gamma\mathbf{u} - \mathbf{v}_2^k + \mathbf{d}_2^k\|_2^2 \\ \mathbf{v}_1^{k+1} = \underset{\mathbf{v}_1}{\operatorname{argmin}} \lambda_1 \mathcal{R}(\mathbf{v}_1) + \frac{\mu_1}{2} \|\mathbf{u}^{k+1} - \mathbf{v}_1 + \mathbf{d}_1^k\|_2^2 \\ \mathbf{v}_2^{k+1} = \underset{\mathbf{v}_2}{\operatorname{argmin}} \lambda_2 \|(\mathbf{v}_2)_T\|_1 + \frac{\mu_2}{2} \|\Gamma\mathbf{u}^{k+1} - \mathbf{v}_2 + \mathbf{d}_2^k\|_2^2 \\ \mathbf{d}_1^{k+1} = \mathbf{d}_1^k + \mathbf{u}^{k+1} - \mathbf{v}_1^{k+1} \\ \mathbf{d}_2^{k+1} = \mathbf{d}_2^k + \Gamma\mathbf{u}^{k+1} - \mathbf{v}_2^{k+1}. \end{cases} \quad (16)$$

Specifically, the subproblem associated with \mathbf{u} is solved with the following normal equation.

$$(\mathbf{A}^T \mathbf{A} + \mu_1 \mathbf{I} + \mu_2 \Gamma^T \Gamma) \mathbf{u}^{k+1} = \mathbf{A}^T \mathbf{f} + \mu_1 (\mathbf{v}_1^k - \mathbf{d}_1^k) + \mu_2 \Gamma^T (\mathbf{v}_2^k - \mathbf{d}_2^k). \quad (17)$$

The \mathbf{v}_1 subproblem is solved with the Gaussian denoiser D_σ that is same as (8). Specifically,

$$\begin{aligned} \mathbf{v}_1^{k+1} &= \underset{\mathbf{v}_1}{\operatorname{argmin}} \mathcal{R}(\mathbf{v}_1) + \frac{1}{2(\sqrt{\lambda_1/\mu_1})^2} \|\mathbf{v}_1 - (\mathbf{u}^{k+1} + \mathbf{d}^k)\|_2^2 \\ &= D_\sigma(\mathbf{u}^{k+1} + \mathbf{d}^k, \sqrt{\lambda_1/\mu_1}). \end{aligned} \quad (18)$$

Given T based on (13), the \mathbf{v}_2 subproblem

$$\mathbf{v}_2^{k+1} = \underset{\mathbf{v}_2}{\operatorname{argmin}} \lambda_2 \|(\mathbf{v}_2)_T\|_1 + \frac{\mu_2}{2} \|\Gamma\mathbf{u}^{k+1} - \mathbf{v}_2 + \mathbf{d}_2^k\|_2^2, \quad (19)$$

can be equivalently decomposed as

$$\mathbf{v}_2^{k+1} = \underset{\mathbf{v}_2}{\operatorname{argmin}} \lambda_2 \|(\mathbf{v}_2)_T\|_1 + \frac{\mu_2}{2} \|(\Gamma\mathbf{u}^{k+1} - \mathbf{v}_2 + \mathbf{d}_2^k)_T\|_2^2 + \frac{\mu_2}{2} \|(\Gamma\mathbf{u}^{k+1} - \mathbf{v}_2 + \mathbf{d}_2^k)_S\|_2^2. \quad (20)$$

Then, the solution of \mathbf{v}_2^{k+1} can be easily derived as below

$$\mathbf{v}_2^{k+1} = \mathcal{S}_{T, \lambda_2, \mu_2}(\Gamma\mathbf{u}^{k+1} + \mathbf{d}_2^k), \quad (21)$$

where the *support-driven adaptive soft shrinkage operator* is defined as

$$\mathcal{S}_{T, \lambda_2, \mu_2}(x_i) = \begin{cases} (|x_i| - \frac{\lambda_2}{\mu_2}) \cdot \frac{x_i}{|x_i|}, & \text{if } i \in T. \\ x_i, & \text{otherwise.} \end{cases} \quad (22)$$

Note that the salient edges of an image commonly correspond to the large coefficients in the transform domain. Compared with the standard soft shrinkage operator, the above support-driven selective soft shrinkage operator (22) only performs the thresholding on the components indexed in T , and thus the nonzero large components are not shrunk. The resulting benefit is that the sharp edges can be well preserved, while simultaneously reducing the artifacts on the smooth regions.

Based on the above analyses, we summarize the whole algorithmic framework into Algorithm 1, which is a multi-stage process, alternatively implementing the support detection and solving the resulting hybrid minimization problem (14).

Remark. Note that Algorithm 1 is a multi-stage procedure, and the initial reference image can be resorted to an off-the-shelf, SOTA restoration method. The solution of this resulting hybrid model (14) can act as the new reference image for the next stage. Then, the updated support set T will lead to a new hybrid model (14), which results in an iterative procedure. Moreover, it should be emphasized that while the hybrid model (14) of each stage is a convex optimization problem if the denoiser satisfies some assumptions [23]. However, the Algorithm 1 actually solves a nonconvex optimization problem (despite its explicit formulation is not necessarily available) as a whole, because the support detection is a nonconvex operation, which is also a key characteristic of our algorithm. In general, the proposed algorithm can be considered as a multi-stage convex relaxation procedure of a nonconvex image recovery model [29].

Theorem 3.1. Consider the application of ADMM to (14), which results in the iterative loop, where \mathbf{v}_2 is computed with the proximal mapping (21), and \mathbf{v}_1 is obtained with a Gaussian denoiser (18). If the denoising operator D_σ has symmetric gradient and is non-expansive [23], the inner iterative loop of the proposed algorithm globally converges to a solution of (14) if one exists.

Algorithm 1 Single image restoration via the Support-Denoiser-driven hybrid model (**HSSD**).

1. **Input:** Given the observed image \mathbf{f} and the degraded operator \mathbf{A} ;
Local sparsifying operator Γ ; Denoiser D_σ ; Regularization parameters λ_1, λ_2 ; Penalty parameters μ_1, μ_2 ; Thresholding parameter ρ .
2. **Initialization:** $\mathbf{d}_1^0 = \mathbf{f}, \mathbf{d}_2^0 = \Gamma\mathbf{f}, \mathbf{v}_1^0 = \mathbf{f}, \mathbf{v}_2^0 = \Gamma\mathbf{f}$. Obtaining a reference image for support detection (determine the T) with an open interface, e.g., the off-the-shelf image restoration methods.
3. **Outer loop (stage):** iteration on $j = 1, \dots, J$.
 - (a) Perform support detection on reference image via (13).
 - (b) **Inner loop:** (solving the minimization problem (14))
 - While** the stopping condition is not satisfied,
 - iteration on $k = 0, 1, 2, \dots, K$. **Do**
 - (I) Update \mathbf{u}^{k+1} via (17).
 - (II) Update \mathbf{v}_1^{k+1} via (18).
 - (III) Update \mathbf{v}_2^{k+1} via (21).
 - (IV) Update $\mathbf{d}_1^{k+1} = \mathbf{d}_1^k + \mathbf{u}^{k+1} - \mathbf{v}_1^{k+1}$.
 - (V) Update $\mathbf{d}_2^{k+1} = \mathbf{d}_2^k + \Gamma\mathbf{u}^{k+1} - \mathbf{v}_2^{k+1}$.
 - End**
 - (c) The intermediate result \mathbf{u}^k acts as the new reference image for support detection in next stage.
 4. **Output:** Final restored image \mathbf{u}^J .

Proof. The proof mainly consists of verifying the conditions of [Lemma 2.1](#). The first condition for convergence is that \mathbf{H} has full column rank. The inner iterative loop of the proposed algorithm is an instance of ADMM with

$$\mathbf{H} = \begin{bmatrix} \Gamma \\ \mathbf{I} \end{bmatrix},$$

which has full column rank, where \mathbf{I} denotes the unit matrix. The second condition regards to functions g_1 and g_2 . According to [Lemma 2.2](#) and [Theorem 2.1](#), it is easy to see that given T , $\|(\Gamma\mathbf{u})_T\|_1$ is a closed, proper and convex function. Furthermore, if the denoiser has symmetric gradient and is non-expansive, it also corresponds to the proximity operator of a closed, proper and convex function [23]. Therefore, the function being minimized is the sum of proper, closed, convex functions, thus the inner iterative loop of the proposed algorithm globally converges if a solution exists according to the theorem of ADMM [25].

4. Numerical experiments

4.1. Experimental settings

The wavelet frame [30] is selected as the local sparsifying operator in our proposed model. Compared to other local sparsifying operators such as the gradient operator of well-known total variation model [31], it allows to adaptively choose proper differential operators according to the order of the singularity of the underlying solutions for different regions of a given image. Specifically, the linear B-spline framelet with the decomposition level 1 is adopted [30]. The parameters of the proposed algorithm, as well as other competing approaches, are manually tuned to achieve the best performance. For the plug-and-play ADMM algorithm, it does not strictly guarantee that the better the denoiser's performance is, the better the performance of the entire scheme is [14,18,32]. So for the sake of consistency and simplicity, unless otherwise specified, following previous works, we also use the BM3D denoising prior (with the default parameters) which is well known to be one of the most efficient Gaussian denoising algorithms in all the experiments [17,18].

The stopping criterion of the proposed Algorithm 1 is:

$$\frac{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|_2}{\|\mathbf{u}^k\|_2} < 1 \times 10^{-5}. \quad (23)$$

In addition, we empirically find that the performance of Algorithm 1 is satisfying even the outer loop (stage) is executed only once, and a larger J value does not always lead to a noticeable improvement. Hence, as a trade-off, we empirically choose $J = 1$ in order to greatly reduce the computational cost of the proposed algorithm.

Table 1

Comparison of the PSNR (dB) and SSIM results of the different denoising methods. The highest PSNR and SSIM scores are highlighted in bold. The sizes of the test images are all 256×256 .

Image	σ	BM3D [15]	EPLL [33]	NCSR [34]	WNNM [16]	PGPD [35]	PCLR [36]	HSSD
Cameraman	30	28.64 0.8373	28.36 0.8316	28.63 0.8396	28.81 0.8403	28.54 0.8259	28.82 0.8430	28.84 0.8468
		26.12 0.7826	26.02 0.7617	26.20 0.7844	26.46 0.7849	26.46 0.7774	26.56 0.7944	26.57 0.7949
	70	24.61 0.7426	24.51 0.7082	24.61 0.7501	24.91 0.7453	24.94 0.7379	25.03 0.7551	25.03 0.7606
		23.08 0.6926	22.86 0.6351	22.97 0.7065	23.41 0.6971	23.23 0.6776	23.49 0.7159	23.52 0.7246
	Parrot	30.45 0.8803	30.11 0.8667	30.44 0.8814	30.85 0.8834	30.43 0.8778	30.75 0.8823	30.74 0.8876
		27.99 0.8407	27.61 0.8154	27.93 0.8478	28.31 0.8450	28.02 0.8382	28.29 0.8479	28.24 0.8541
		26.39 0.8068	25.93 0.7695	26.15 0.8188	26.60 0.8137	26.36 0.8026	26.64 0.8193	26.53 0.8262
		24.70 0.7592	24.03 0.7047	24.26 0.7789	24.89 0.7716	24.59 0.7462	24.79 0.7809	24.75 0.7890
Monarch	30	28.36 0.8822	28.35 0.8789	28.47 0.8856	28.92 0.8918	28.49 0.8853	28.83 0.8930	28.81 0.8937
		25.82 0.8200	25.78 0.8124	25.77 0.8260	26.32 0.8343	26.00 0.8269	26.25 0.8370	26.25 0.8376
	70	24.24 0.7674	24.07 0.7533	24.04 0.7768	24.63 0.7861	24.34 0.7756	24.59 0.7903	24.58 0.7905
		22.52 0.7021	22.23 0.6771	22.13 0.7124	22.95 0.7256	22.56 0.7029	22.93 0.7364	22.92 0.7367
	Lena	29.46 0.8584	29.19 0.8477	29.36 0.8597	29.73 0.8667	29.60 0.8622	29.69 0.8647	29.68 0.8684
		26.90 0.7920	26.69 0.7732	26.95 0.8011	27.27 0.8068	27.15 0.7990	27.19 0.8054	27.09 0.8096
		25.48 0.7407	25.08 0.7107	25.35 0.7529	25.82 0.7622	25.60 0.7645	25.63 0.7574	25.69 0.7649
		23.87 0.6740	23.46 0.6345	23.64 0.7048	24.35 0.7025	24.02 0.6780	24.08 0.7033	24.28 0.7149

Note that the solution to the proposed model can also be solved via other optimization methods such as the primal-dual (PD) method [19]. Here, we choose the ADMM method mainly due to its simplicity, high efficiency and competitive convergence property. All the experimental test images are shown in Fig. 1 (Note that the sizes of these images are different, and the pixel intensity ranges from 0 to 255). For color images, the image restoration operations are only implemented to the luminance component. For convenience, we denote our method **Hybrid Sparsity Support Denoiser** priors based image restoration as **HSSD** in the following experiments. The Matlab source code of our proposed algorithm can be downloaded at the website: <https://github.com/jackygsb>.

4.2. Image denoising

We compare the proposed HSSD method with several representative model-based image denoising algorithms, including the **EPLL** [33], **BM3D** [15], **NCSR** [34], **WNNM** [16], **PGPD** [35], **PCLR** [36]. Apart from these generic image denoising methods, a model-based class-specific image denoising algorithm [37], denoted as **TIP'17**, is also included such that we can preliminarily investigate the performance of our algorithm for class-specific denoising. We adopt the denoising results of [16] and [37] as the initial reference images of HSSD to estimate the support set S , in the cases of generic and class-specific image denoising, respectively. The threshold parameter ρ is related to the noise level, and we set ρ to be 300, 200, 150 and 100 for $\sigma \leq 30$, $30 < \sigma \leq 50$, $50 < \sigma \leq 70$, $70 < \sigma \leq 100$, respectively. The maximum iterative number K is set to be 50, 100, 150, and 200 respectively, on these noise levels. The regularization parameters λ_1 and λ_2 are adjusted to give the best performance in terms of PSNR and SSIM [38] values.

The PSNR and SSIM results of these tested methods are reported in Table 1. The highest PSNR and SSIM values for each image and on each noise level are highlighted in bold. From Table 1, we can see that the proposed algorithm achieves a highly competitive denoising performance. Additionally, compared with the model-based image denoising algorithms, though our algorithm does not consistently outperform the competitive algorithms in terms of PSNR values, it consistently



Fig. 1. All of the experimental test images.

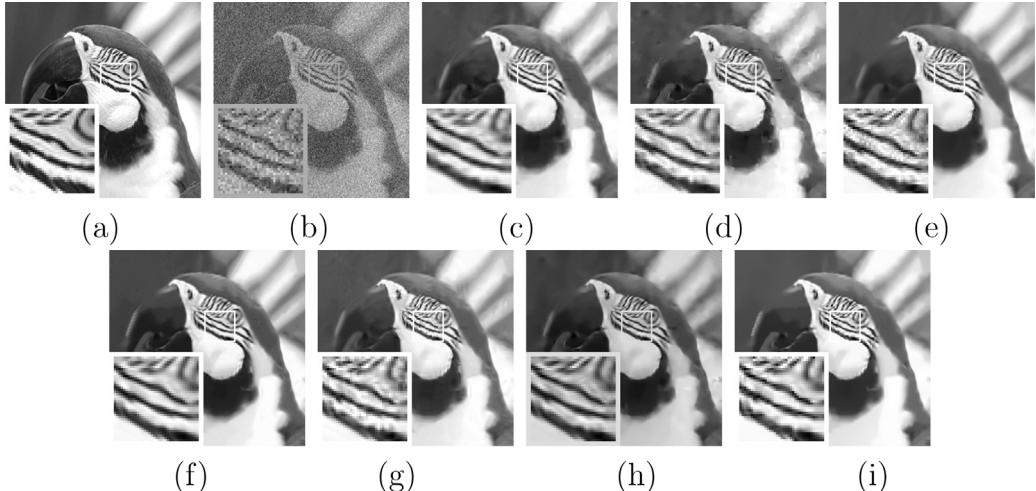


Fig. 2. Denoising performance comparison on the *Parrot* image (256×256) with heavy noise corruption ($\sigma = 50$). (a) original image, (b) noisy image (PSNR = 14.12; SSIM = 0.1466), (c) BM3D [15] (PSNR = 27.99; SSIM = 0.8407), (d) EPLL [33] (PSNR = 27.61; SSIM = 0.8154), (e) NCSR [34] (PSNR = 27.93; SSIM = 0.8478), (f) WNNM [16] (PSNR = **28.31**; SSIM = 0.8450), (g) PGPD [35] (PSNR = 28.02; SSIM = 0.8382), (h) PCLR [36] (PSNR = 28.29; SSIM = 0.8479), (i) HSSD (PSNR = 28.24; SSIM = **0.8541**).

achieves the highest SSIM values among all of the tested images on four noise levels, which is more consistent with our human eye perception.

Figs. 2 and 3 present the visual comparison of generic denoising results for two images (*Parrot* and *Cameraman*) at moderate ($\sigma = 50$) and heavy ($\sigma = 100$) noise levels. As observed from the highlighted window, HSSD is very effective in reconstructing both the smooth and the texture/edge regions. We also compare our algorithm with the SOTA methods for class-specific image denoising scenarios. Figs. 4 and 5 display the visual comparison for two typical *face* images³ at noise levels $\sigma = 20$ and $\sigma = 50$. It can be observed that our algorithm delivers the best visual quality in both the smooth and the texture/edge regions. In contrast, restored images by the competing methods often suffer from the noticeable artifacts especially around the smooth areas.

4.3. Image deblurring

Two sets of experiments are conducted to demonstrate the performance of the proposed algorithm for simulated image deblurring. In the first set, two commonly used blur kernels (see Table 2) are adopted for simulations. Then, additive Gaussian noise with noise level $\sigma = \sqrt{2}$ is added to the blurred images. In the second set, six typical deblurring experiments (as shown in Table 3) with respect to four standard gray images are conducted for more comprehensive comparison.

We compare our algorithm with several popular deblurring methods, including the framelet-based ℓ_1 regularization (**Split Bregman**) [30], framelet-based truncated ℓ_1 regularization (**Truncated ℓ_1**) [28], **IDD-BM3D** [17], **TVMM** [31], **L0-Spar** [39], **NCSR** [34], machine learning based method **MLP** [40], plug-and-play ADMM framework using the BM3D denoiser (**ADMM-BM3D**) [18], plug-and-play ADMM framework using the GMM denoiser (**ADMM-GMM**) [32], plug-and-play scheme using the CNN denoiser (**IRCNN**) [1] and plug-and-play scheme using the CNN and WNNM denoisers (**LNIR**) [41], iterative denoising and backward projections using the CNN denoiser (**IDBP-CNN**) [42]. As far as we know,

³ The test images can be available on the website: <https://saeed-anwar.github.io/publications/>.

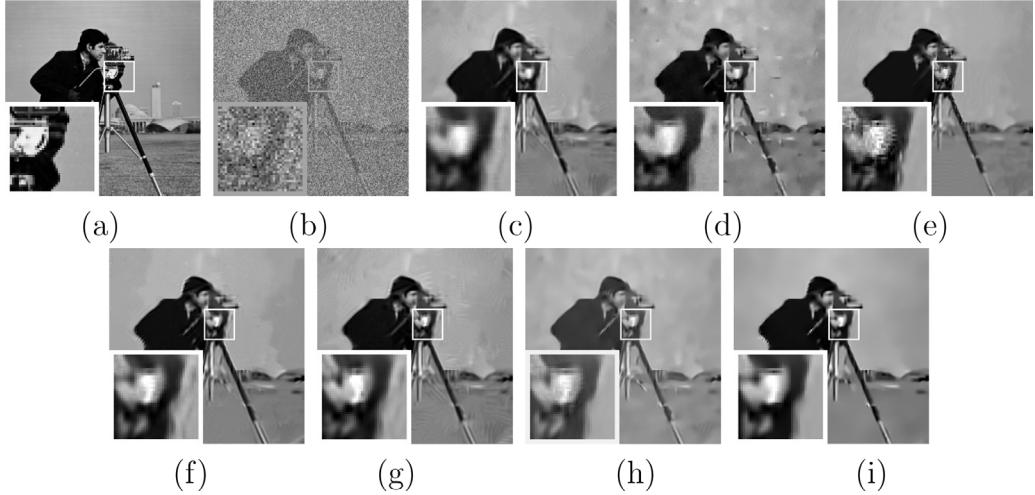


Fig. 3. Denoising performance comparison on the *Cameraman* image (256×256) with heavy noise corruption ($\sigma = 100$). (a) original image, (b) noisy image (PSNR = 8.10; SSIM = 0.0762), (c) BM3D [15] (PSNR = 23.08; SSIM = 0.6926), (d) EPLL [33] (PSNR = 22.86; SSIM = 0.6351), (e) NCSR [34] (PSNR = 22.97; SSIM = 0.7065), (f) WNNM [16] (PSNR = 23.41; SSIM = 0.6971), (g) PGPD [35] (PSNR = 23.23; SSIM = 0.6776), (h) PCLR [36] (PSNR = 23.49; SSIM = 0.7159), (i) HSSD (PSNR = **23.52**; SSIM = **0.7246**).

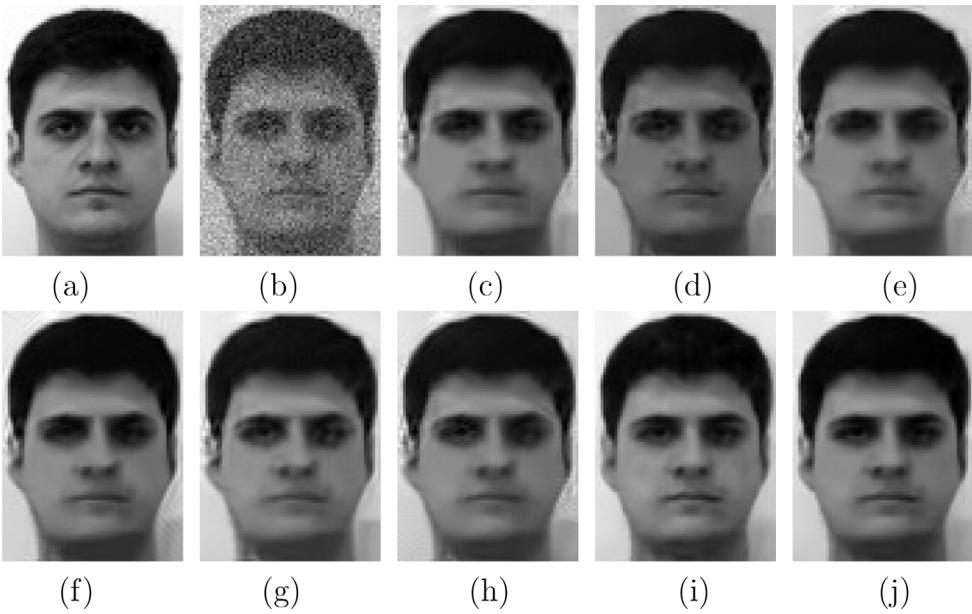


Fig. 4. Denoising performance comparison. (a) original image; (b) noisy image ($\sigma = 20$; PSNR = 22.14; SSIM = 0.4807); (c) BM3D [15] (PSNR = 31.65; SSIM = 0.9172); (d) EPLL [33] (PSNR = 31.86; SSIM = 0.9192); (e) NCSR [34] (PSNR = 31.57; SSIM = 0.9148); (f) WNNM [16] (PSNR = 32.05; SSIM = 0.9173); (g) PGPD [35] (PSNR = 31.88; SSIM = 0.9162); (h) PCLR [36] (PSNR = 31.94; SSIM = 0.9186); (i) TIP'17 [37] (PSNR = 32.60; SSIM = 0.9282); (j) HSSD (PSNR = **32.96**; SSIM = **0.9335**).

ADMM-BM3D, IRCNN, IDBP-CNN provide the SOTA image deblurring performance among the denoiser-based plug-and-play methods. We adopt the deblurring result of IDD-BM3D method as the initial reference image for estimating the support set S , and the thresholding parameter in (13) is set as $\rho = 300$. The maximum iterative number is set as $K = 100$. The regularization parameters λ_1 and λ_2 are proportional to the noise level. Empirically, λ_1 can be set as $\lambda_1 = \sigma^2/3$, and λ_2 can be set as $\lambda_2 = \lambda_1/2$.

The PSNR and SSIM results on the ten gray test images in the first set of experiments are reported in Table 2. From Table 2, we can observe that the proposed algorithm outperforms the competing methods in terms of both PSNR and SSIM values. On average, our proposed algorithm outperforms IDD-BM3D and ADMM-BM3D by (0.37 dB, 0.53 dB) and (0.35 dB, 0.49 dB) in terms of PSNR values, and by (0.0164, 0.0138) and (0.0106, 0.0116) in terms of SSIM values for the

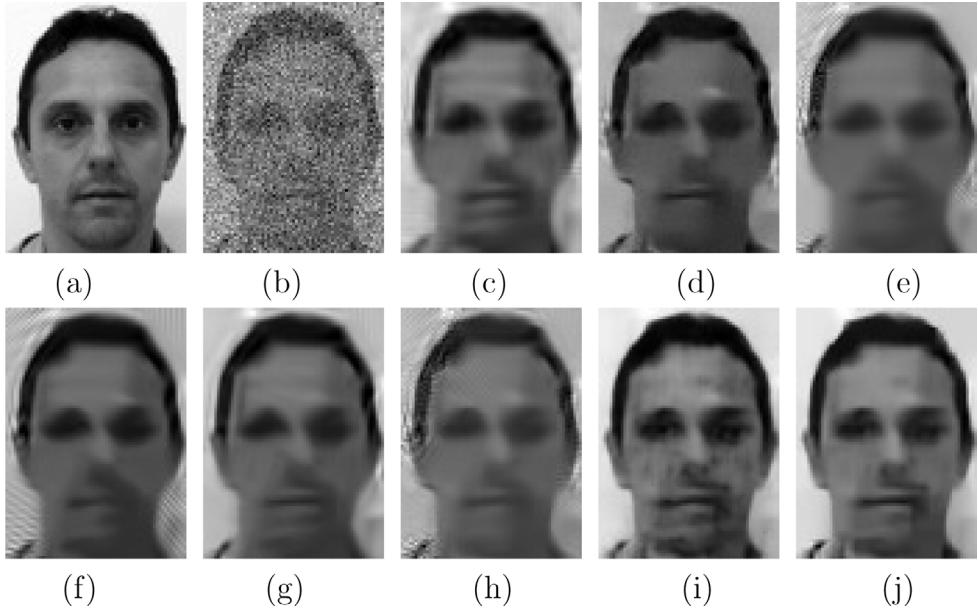


Fig. 5. Denoising performance comparison. (a) original image; (b) noisy image ($\sigma = 50$; PSNR = 14.18; SSIM = 0.1951); (c) BM3D [15] (PSNR = 26.55; SSIM = 0.8136); (d) EPLL [33] (PSNR = 26.71; SSIM = 0.8039); (e) NCSR [34] (PSNR = 25.61; SSIM = 0.7817); (f) WNNM [16] (PSNR = 26.25; SSIM = 0.7946); (g) PGPD [35] (PSNR = 26.72; SSIM = 0.8122); (h) PCLR [36] (PSNR = 25.68; SSIM = 0.7908); (i) TIP'17 [37] (PSNR = 27.85; SSIM = 0.8329); (j) HSSD (PSNR = **28.14**; SSIM = **0.8471**).

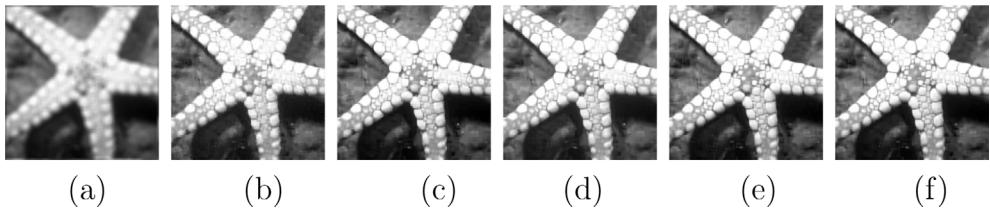


Fig. 6. Visual quality comparison of image deburring. (a) Blurred and noisy image (*Starfish* (256 × 256), Uniform kernel: (9 × 9, $\sigma = \sqrt{2}$)); (b) recovery image by Split Bregman [30] (PSNR = 27.09; SSIM = 0.7934), (c) Truncated ℓ_1 [28] (PSNR = 28.26; SSIM = 0.8318), (d) IDD-BM3D [17] (PSNR = 28.35; SSIM = 0.8321), (e) ADMM-BM3D [18] (PSNR = 28.46; SSIM = 0.8412), (f) HSSD (PSNR = **28.88**; SSIM = **0.8552**).

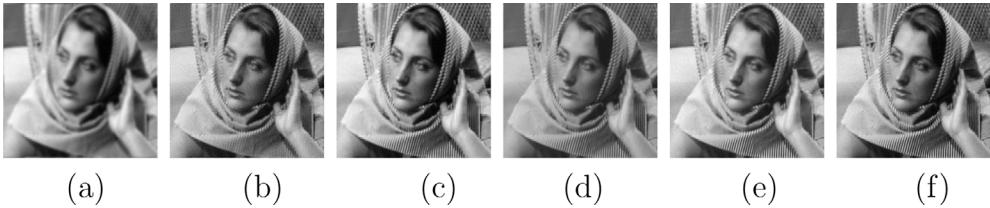


Fig. 7. Visual quality comparison of image deburring. (a) Blurred and noisy image (*Barbara* (256 × 256), Gaussian kernel: fspecial (Gaussian, 25, 1.6, $\sigma = \sqrt{2}$)). (b) recovery image by Split Bregman [30] (PSNR = 25.12; SSIM = 0.7640), (c) Truncated ℓ_1 [28] (PSNR = 27.06; SSIM = 0.8207), (d) IDD-BM3D [17] (PSNR = 27.08; SSIM = 0.8205), (e) ADMM-BM3D [18] (PSNR = 26.76; SSIM = 0.8097), (f) HSSD (PSNR = **27.69**; SSIM = **0.8332**).

two blur kernels, respectively. The visual comparisons of the deblurring methods are shown in Figs. 6 and 7, from which we can see that the proposed model using hybrid priors produces the sharper image edges and cleaner textures than other competing methods. The local-based sparsity prior, support prior, and nonlocal patched denoiser prior are complementary to improve the restoration quality, and turning off either one of them leads to a worse result.

Table 4 lists the comparison of ISNR results for six typical deblurring experiments in the second set. Here, the BM3D and WNNM denoisers are exploited in our algorithm (which we term as HSSD-BM3D and HSSD-WNNM, respectively) for the purpose of verifying the role of the denoiser. Note that the PSNR results for the second set have been reported

Table 2

Comparison of PSNR (dB) and SSIM results by different deblurring methods in the first set. Bold values denote the highest PSNR and SSIM values. The sizes of the test images are all 256×256 .

Scenario	Image	Split Bregman [30]	Truncated ℓ_1 [28]	IDD-BM3D [17]	ADMM-BM3D [18]	HSSD-BM3D
Uniform Kernel $(9 \times 9, \sigma = \sqrt{2})$	<i>Cameraman</i>	26.74 0.8335	28.64 0.8653	28.54 0.8586	28.24 0.8528	28.98 0.8763
	<i>Boat</i>	26.95 0.7854	28.12 0.8265	28.06 0.8219	27.98 0.8274	28.54 0.8452
	<i>Man</i>	25.90 0.7556	26.60 0.7849	26.55 0.7799	26.42 0.7837	26.95 0.8031
	<i>Monarch</i>	27.47 0.8752	29.20 0.9061	29.04 0.9034	28.31 0.8959	29.50 0.9154
	<i>Peppers</i>	28.57 0.8274	29.67 0.8450	29.62 0.8427	29.41 0.8438	29.97 0.8553
	<i>Lena</i>	28.59 0.8438	29.73 0.8685	29.71 0.8658	29.63 0.8657	30.07 0.8793
	<i>Barbara</i>	26.35 0.7678	27.94 0.8232	27.90 0.8227	27.87 0.8202	28.07 0.8306
	<i>Parrots</i>	27.39 0.8708	30.09 0.8962	29.98 0.8914	29.99 0.8900	30.33 0.9022
	<i>Starfish</i>	27.09 0.7934	28.26 0.8318	28.35 0.8321	28.46 0.8412	28.88 0.8552
	<i>Goldhill</i>	27.47 0.7357	27.95 0.7563	27.92 0.7526	27.90 0.7579	28.25 0.7723
	Avg.	27.25 0.8089	28.62 0.8404	28.58 0.8371	28.42 0.8379	28.95 0.8535
Gaussian Kernel $(\text{Gaussian}(25, 1.6), \sigma = \sqrt{2})$	<i>Cameraman</i>	27.00 0.8601	28.26 0.8766	28.10 0.8687	27.93 0.8653	28.51 0.8810
	<i>Boat</i>	28.01 0.8396	28.82 0.8570	28.73 0.8528	28.62 0.8545	29.08 0.8671
	<i>Man</i>	27.46 0.8376	27.87 0.8464	27.83 0.8441	27.67 0.8432	28.04 0.8549
	<i>Monarch</i>	30.35 0.9335	31.11 0.9408	30.90 0.9380	29.94 0.9330	31.19 0.9460
	<i>Peppers</i>	29.14 0.8781	30.01 0.8807	29.97 0.8799	30.37 0.8804	30.28 0.8868
	<i>Lena</i>	30.71 0.9025	31.43 0.9112	31.41 0.9089	31.22 0.9101	31.60 0.9171
	<i>Barbara</i>	25.12 0.7640	27.06 0.8207	27.08 0.8205	26.76 0.8097	27.69 0.8332
	<i>Parrots</i>	30.31 0.9163	31.78 0.9239	31.55 0.9179	31.50 0.9186	31.90 0.9273
	<i>Starfish</i>	29.08 0.8768	30.57 0.8969	30.36 0.8924	30.67 0.8994	31.00 0.9060
	<i>Goldhill</i>	27.66 0.7699	28.17 0.7802	28.18 0.7787	28.04 0.7779	28.30 0.7883
	Avg.	28.48 0.8578	29.51 0.8734	29.41 0.8702	29.27 0.8692	29.76 0.8808

in [17,32] and [42]. For fair comparison, the PSNR values of other competing methods are directly obtained from [17,32] and [42]. We optimize the parameters of the proposed deblurring method for each experiment. From Table 4, we can see that the proposed HSSD-WNNM outperforms HSSD-BM3D slightly, while it requires much more computational time (see Section 4.5 for details). In addition, our algorithm achieves the highest ISNR values over other competing methods in most cases, including the plug-and-play ADMM scheme using the deep learning denoisers. Particularly, the performance gain over the well-known IDD-BM3D and NCSR methods on image *Barbara* is significant (up to 1.34 dB and 1.66 dB on scenario 2, respectively). Moreover, our proposed HSSD-BM3D outperforms the ADMM-BM3D by a large margin in terms of PSNR values in most cases, which demonstrates the benefits of incorporating the local sparsity and support priors into the regularized model. The visual comparisons of the deblurring methods for the second set are presented in Figs. 8 and 9. From the enlarged regions, we can observe that our proposed algorithm generates much cleaner image than other competing approaches on both homogeneous areas and salient edges.

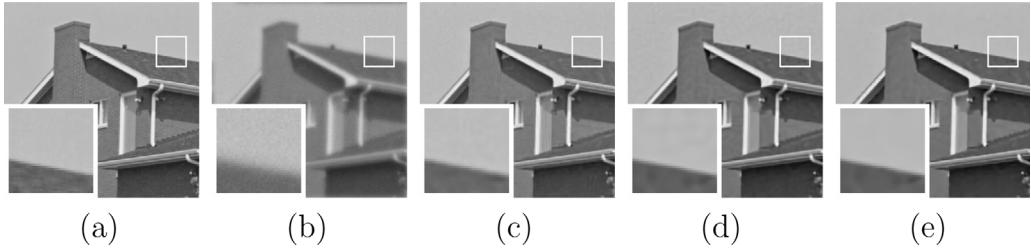


Fig. 8. Visual quality comparison of image deburring. (a) Original clean image *House* (256×256), (b) Blurred and noisy image (scenario: 2), (c) recovery image by IDD-BM3D [17] ($\text{PSNR} = 34.01$), (d) ADMM-BM3D [18] ($\text{PSNR} = 33.48$), (e) HSSD ($\text{PSNR} = 34.47$). Best viewed on high-resolution display.



Fig. 9. Visual quality comparison of image deburring. (a) Original clean image *Lena* (256×256), (b) Blurred and noisy image (scenario: 4), (c) recovery image by IDD-BM3D [17] ($\text{PSNR} = 33.78$), (d) ADMM-BM3D [18] ($\text{PSNR} = 33.69$), (e) HSSD ($\text{PSNR} = 34.22$). Best viewed on high-resolution display.

Table 3

Six typical deblurring scenarios with different blur PSFs and noise variances in the second set.

Scenario	PSF	σ^2
1	$1/(z_1^2 + z_2^2)$, $z_1, z_2 = -7, \dots, 7$	2
2	$1/(z_1^2 + z_2^2)$, $z_1, z_2 = -7, \dots, 7$	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

4.4. Image inpainting

In this subsection, two interesting types of image inpainting with different masks are considered, i.e., image restoration from partial random samples and the text (scratch) removal. The proposed algorithm is compared with several representative model-based methods including **ISDSB** [44], **FOE** [45], **JSM** [46], **BPFA** [47], **GSR** [43], **ADMM-BM3D** [18], **Truncated ℓ_1** [28] and **TIP'19** [48]. We adopt the inpainting result of GSR method as the initial reference image for estimating the support set S , and the thresholding parameter in (13) is set as $\rho = 300$. The maximum iterative number is set as $K = 300$. The regularization parameters λ_1 and λ_2 are set as $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$, respectively.

Table 5 lists the PSNR and SSIM results of the competing methods for image inpainting from partial random samples on eight standard grayscale images at different percentages of random samples. Note that the PSNR and SSIM results have already been reported in [48]. For fair comparison, the quantitative values of other competing methods are directly obtained from [48]. It can be clearly seen that our algorithm outperforms the competing methods in most cases in terms of both the PSNR and SSIM values. In particular, the performance gain by the propose algorithm over [48] on ratio 10% is significant (up to 2.20 dB and 3.38 dB on images *Cameraman256* and *Lena512*, respectively).

Table 6 lists the PSNR results by our proposed method for image inpainting from partial random samples on five standard color images at different percentages of random samples. It can be observed that the proposed algorithm achieves a highly competitive inpainting performance compared to GSR, and outperforms the ADMM-BM3D [18] and Truncated ℓ_1 [28] by a large margin in terms of the PSNR values. Particularly, the performance gain over the GSR method on color image *Foreman* is quite significant (up to 4.6 dB on data percentage 80%). These results demonstrate the advantage

Table 4

ISNR (dB) comparison for image deblurring in the second set. The highest PSNR values are highlighted in bold.

Image	Method	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Cameraman (256 × 256)	BSNR	31.87	25.85	40.00	18.53	29.19	17.76
	Input PSNR	22.23	22.16	20.76	24.62	23.36	29.82
	TVMM [31]	7.41	5.17	8.54	2.57	3.36	1.30
	L0-Spar [39]	7.70	5.55	9.10	2.93	3.49	1.77
	IDD-BM3D [17]	8.85	7.12	10.45	3.98	4.31	4.89
	NCSR [34]	8.78	6.69	10.33	3.78	4.60	4.50
	ADMM-BM3D [18]	8.18	6.13	9.58	3.26	3.93	4.88
	ADMM-GMM [32]	8.34	6.39	9.73	3.49	4.18	4.90
	MLP [40]	–	–	–	–	4.48	–
	IRCNN [1]	9.08	7.33	10.30	4.29	4.70	–
	IDBP-CNN [42]	9.08	7.28	10.55	4.25	–	–
	LNIR [41]	–	–	–	–	4.70	–
HSSD-BM3D	9.20	7.47	10.90	4.11	4.68	5.50	–
	HSSD-WNNM	9.30	7.61	10.93	4.12	4.82	5.59
House (256 × 256)	BSNR	29.16	23.14	40.00	15.99	26.61	15.15
	Input PSNR	25.61	25.46	24.11	28.06	27.81	29.98
	TVMM [31]	7.98	6.57	10.39	4.12	4.54	2.44
	L0-Spar [39]	8.40	7.12	11.06	4.55	4.80	2.15
	IDD-BM3D [17]	9.95	8.55	12.89	5.79	5.74	7.13
	NCSR [34]	9.96	8.48	13.12	5.81	5.67	6.94
	ADMM-BM3D [18]	9.64	8.02	12.95	5.23	5.06	7.37
	ADMM-GMM [32]	9.84	8.40	12.87	5.57	5.55	6.65
	MLP [40]	–	–	–	–	5.62	–
	IRCNN [1]	9.69	8.63	11.58	6.05	5.98	–
	IDBP-CNN [42]	9.93	8.45	11.91	5.85	–	–
	LNIR [41]	–	–	–	–	6.26	–
HSSD-BM3D	10.31	9.01	13.43	6.14	5.91	7.37	–
	HSSD-WNNM	10.44	9.11	13.63	6.26	6.18	7.43
Lena (512 × 512)	BSNR	29.89	23.87	40.00	16.47	27.18	15.52
	Input PSNR	27.25	27.04	25.84	28.81	29.16	30.03
	TVMM [31]	6.36	4.98	7.47	3.52	3.61	2.79
	L0-Spar [39]	6.66	5.71	7.79	4.09	4.22	1.93
	IDD-BM3D [17]	7.97	6.61	8.91	4.97	4.85	6.34
	NCSR [34]	8.03	6.54	9.25	4.93	4.86	6.19
	ADMM-BM3D [18]	8.00	6.56	9.00	4.88	4.67	6.42
	ADMM-GMM [32]	8.01	6.53	8.95	4.93	4.81	6.09
	IRCNN [1]	8.06	6.79	8.88	5.13	–	–
	IDBP-CNN [42]	8.24	6.64	9.05	5.05	–	–
	HSSD-BM3D	8.42	7.05	9.46	5.41	5.09	6.60
	HSSD-WNNM	8.51	7.12	9.55	5.49	5.13	6.68
Barbara (512 × 512)	BSNR	30.81	24.79	40.00	17.35	28.07	16.59
	Input PSNR	23.34	23.25	22.49	24.22	23.77	29.78
	TVMM [31]	3.10	1.33	3.49	0.41	0.75	0.59
	L0-Spar [39]	3.51	1.53	3.98	0.73	0.81	1.17
	IDD-BM3D [17]	7.64	3.96	6.05	1.88	1.16	5.45
	NCSR [34]	7.76	3.64	5.92	2.06	1.43	5.50
	ADMM-BM3D [18]	7.32	2.99	6.05	1.55	1.40	5.76
	ADMM-GMM [32]	5.91	2.19	5.37	1.42	1.24	5.14
	IRCNN [1]	7.54	4.68	5.92	1.82	–	–
	IDBP-CNN [42]	6.89	4.41	6.07	2.40	–	–
	HSSD-BM3D	8.18	5.05	6.73	2.13	1.50	5.83
	HSSD-WNNM	8.62	5.30	7.01	2.27	1.68	6.06

of exploiting the local (sparsity and support) and nonlocal image (denoiser) priors simultaneously. The visual quality comparisons in the case of image inpainting from 20% random samples and in the case of text (scratch) removal for two standard color test images are displayed in Figs. 10 and 11, respectively. It can be observed that our algorithm is superior to other competing algorithms, in terms of both the sharpness of the edges and smoothness of the homogenous regions in the recovered images.



Fig. 10. Visual quality comparison of image inpainting from partial random samples for color image. (a) Original clean image *Lena*, (b) the degraded image with only 20% random samples available ($\text{PSNR} = 6.85$), (c) recovery image by GSR [43] ($\text{PSNR} = 30.74$), (d) ADMM-BM3D [18] ($\text{PSNR} = 30.47$), (e) Truncated ℓ_1 [28] ($\text{PSNR} = 30.14$), (f) HSSD ($\text{PSNR} = \mathbf{31.01}$). (g) Original clean image *Foreman*, (h) the degraded image with only 20% random samples available ($\text{PSNR} = 4.57$), (i) recovery image by GSR [43] ($\text{PSNR} = 32.43$), (j) ADMM-BM3D [18] ($\text{PSNR} = 33.00$), (k) Truncated ℓ_1 [28] ($\text{PSNR} = 32.11$), (l) HSSD ($\text{PSNR} = \mathbf{34.25}$). Bold values denote the best PSNR values.



Fig. 11. Visual quality comparison in the case of text/scratch removal for color image. (a) Original clean image *Lena*, (b) the degraded image ($\text{PSNR} = 13.32$), (c) recovery image by GSR [43] ($\text{PSNR} = 37.08$), (d) ADMM-BM3D [18] ($\text{PSNR} = 36.95$), (e) Truncated ℓ_1 [28] ($\text{PSNR} = 36.67$), (f) HSSD ($\text{PSNR} = \mathbf{37.32}$). (g) Original clean image *Foreman*, (h) the degraded image ($\text{PSNR} = 9.00$), (i) recovery image by GSR [43] ($\text{PSNR} = 37.11$), (j) ADMM-BM3D [18] ($\text{PSNR} = 35.54$), (k) Truncated ℓ_1 [28] ($\text{PSNR} = 36.31$), (l) HSSD ($\text{PSNR} = \mathbf{38.13}$). Bold values denote the best PSNR values.

4.5. Computational complexity and running time

Note that the computational complexity of the whole algorithm consists of two parts. For example, for the image deblurring, assume that the image boundary condition is periodic. The computational complexity of the off-the-shelf image restoration method to produce an initial reference image is denoted as C_1 , the number of image pixels is N , the computational complexity of the Gaussian denoiser for a single iteration is denoted as C_2 , and the outer stage number and inner iteration number are J and K , respectively. Then, the total complexity of our algorithm is $C_1 + \mathcal{O}(JK(C_2 + N \log N))$. Empirically, for a 256×256 gray-scale image, the **Inner loop** of the proposed algorithm takes around $2 \sim 3$ minutes for image deblurring and $6 \sim 7$ minutes for image inpainting, respectively, on an Intel(R) Core(TM) i5-4590 CPU (3.30 GHz), 4 GB of memory PC under Matlab R2010a environment.

The proposed HSSD algorithm iteratively solves a shrinkage operator and a nonlocal patched denoising problem. Since the adopted denoising algorithm is time consuming, the proposed algorithm is relatively slow. Table 7 gives the average computational CPU time of our algorithm for denoising, deblurring and inpainting on images of size 256×256 . As expected, HSSD-WNNM is considerably slower than HSSD-BM3D, as WNNM denoising algorithm requires much more time. A possible way to accelerate the proposed method is to improve the processing speed of the denoiser, by tuning the parameters of the denoiser at a cost of slightly reducing the denoising performance, which we remain it as an important future work.

Table 5

Comparison of the PSNR (dB) and SSIM inpainting results for the grayscale images. The highest PSNR and SSIM scores are highlighted in bold.

Ratio	Method	Cameraman256	House256	Peppers256	Lena512	Couple512	Boat512	Man512	Goldhill512
10%	ISDSB [44]	19.03 0.669	21.62 0.714	18.90 0.657	23.83 0.732	21.80 0.548	21.77 0.584	22.96 0.619	23.73 0.582
	FOE [45]	22.59 0.768	27.92 0.826	24.37 0.813	27.91 0.820	25.18 0.710	25.00 0.710	26.14 0.742	27.14 0.716
	JSM [46]	22.23 0.755	29.34 0.840	25.31 0.817	28.21 0.814	25.06 0.701	25.03 0.688	25.82 0.709	26.52 0.678
	BPFA [47]	21.69 0.698	26.09 0.770	22.97 0.775	27.39 0.801	24.46 0.673	24.75 0.679	25.33 0.709	26.49 0.679
	GSR [43]	22.12 0.776	31.37 0.870	26.05 0.851	29.18 0.851	25.58 0.763	25.56 0.741	26.38 0.763	27.16 0.732
	TIP'19 [48]	20.45 0.749	29.88 0.775	25.68 0.842	25.90 0.832	25.33 0.753	24.66 0.721	25.83 0.758	26.86 0.682
	HSSD	22.65 0.779	31.18 0.863	26.54 0.848	29.28 0.846	25.88 0.767	25.87 0.742	26.53 0.763	27.42 0.736
	ISDSB [44]	21.73 0.757	25.07 0.799	22.11 0.792	27.30 0.815	24.29 0.677	24.10 0.691	25.78 0.729	26.29 0.693
	FOE [45]	24.72 0.842	31.12 0.879	27.56 0.883	30.93 0.879	27.94 0.821	27.76 0.804	28.39 0.831	29.31 0.808
	JSM [46]	25.27 0.844	33.17 0.895	28.66 0.890	31.55 0.886	28.50 0.839	28.08 0.810	28.48 0.821	29.43 0.802
20%	BPFA [47]	24.10 0.806	30.19 0.872	25.74 0.864	30.09 0.879	27.38 0.801	27.75 0.799	28.01 0.815	29.10 0.789
	GSR [43]	24.76 0.852	34.36 0.922	28.21 0.899	32.80 0.909	28.94 0.864	28.97 0.844	28.98 0.853	30.26 0.838
	TIP'19 [48]	24.57 0.845	34.55 0.924	28.03 0.899	32.81 0.908	29.21 0.866	29.32 0.843	29.13 0.856	29.47 0.825
	HSSD	25.18 0.855	34.51 0.920	28.84 0.900	32.88 0.909	29.25 0.868	29.21 0.845	29.15 0.854	30.34 0.839
	ISDSB [44]	24.00 0.822	29.27 0.865	24.74 0.865	29.42 0.861	26.11 0.761	26.05 0.767	27.57 0.797	28.17 0.769
	FOE [45]	26.34 0.879	33.26 0.911	29.72 0.911	32.81 0.909	29.98 0.879	29.88 0.856	30.05 0.879	31.03 0.859
	JSM [46]	27.43 0.890	35.57 0.926	30.46 0.921	33.90 0.919	30.56 0.891	30.52 0.872	30.50 0.878	31.19 0.861
30%	BPFA [47]	25.68 0.855	33.25 0.915	28.22 0.901	33.07 0.909	29.67 0.867	29.89 0.854	29.98 0.871	31.03 0.849
	GSR [43]	27.56 0.898	36.58 0.948	30.60 0.929	35.10 0.933	31.01 0.905	31.36 0.892	30.96 0.900	32.21 0.886
	TIP'19 [48]	27.80 0.898	35.97 0.948	31.46 0.930	35.16 0.932	31.21 0.904	31.51 0.893	31.28 0.901	32.26 0.883
	HSSD	27.92 0.901	36.71 0.948	31.15 0.930	35.18 0.934	31.23 0.907	31.56 0.893	31.13 0.900	32.30 0.886
	ISDSB [44]	25.64 0.865	31.63 0.895	26.67 0.901	31.26 0.895	28.02 0.827	28.01 0.828	29.24 0.847	29.70 0.823
	FOE [45]	28.15 0.911	35.51 0.937	31.41 0.933	34.45 0.930	31.94 0.917	31.57 0.888	31.60 0.913	32.50 0.896
	JSM [46]	29.02 0.920	37.20 0.949	32.04 0.940	35.69 0.940	32.19 0.922	32.46 0.909	32.11 0.914	32.94 0.899
40%	BPFA [47]	27.24 0.890	35.51 0.939	30.34 0.921	34.87 0.925	31.44 0.901	31.57 0.882	31.40 0.901	32.46 0.881
	GSR [43]	29.15 0.925	38.45 0.962	32.23 0.946	36.91 0.949	32.71 0.931	33.43 0.923	32.78 0.929	33.82 0.916
	TIP'19 [48]	29.53 0.924	39.26 0.965	32.69 0.943	37.05 0.947	32.87 0.947	33.49 0.930	32.92 0.924	33.76 0.913
	HSSD	29.68 0.926	38.52 0.962	32.76 0.946	36.93 0.949	32.99 0.932	33.48 0.923	32.81 0.929	33.87 0.916

Table 6

Comparison of the PSNR (dB) inpainting results for the color images. The highest PSNR values are highlighted in bold.

Ratio	Method	Lena	House	Foreman	Monarch	Leaves
20%	Input	6.85	6.16	4.57	6.55	3.95
	GSR [43]	30.74	35.59	32.43	26.03	27.44
	ADMM-BM3D [18]	30.47	34.31	33.00	24.53	26.31
	Truncated ℓ_1 [28]	30.14	35.00	32.11	25.94	26.87
30%	HSSD	31.01	35.82	34.25	26.19	27.60
	Input	7.42	6.74	5.14	7.11	4.53
	GSR [43]	32.71	37.62	34.42	28.93	31.17
	ADMM-BM3D [18]	32.45	37.11	35.20	27.89	29.85
50%	Truncated ℓ_1 [28]	32.16	37.19	33.89	28.63	29.85
	HSSD	32.96	38.03	35.70	28.99	31.27
	Input	8.92	8.22	6.61	8.62	6.02
	GSR [43]	36.14	41.61	36.61	32.84	35.86
80%	ADMM-BM3D [18]	35.97	40.89	38.58	31.90	34.68
	Truncated ℓ_1 [28]	35.33	41.15	36.35	32.32	33.56
	HSSD	36.38	41.63	39.52	32.94	35.97
	Input	12.87	12.19	10.54	12.64	9.98
80%	GSR [43]	42.49	47.60	40.73	39.71	44.58
	ADMM-BM3D [18]	42.18	47.20	44.56	38.22	42.87
	Truncated ℓ_1 [28]	42.05	47.25	41.12	39.20	41.88
	HSSD	42.87	47.65	45.47	39.54	44.20

Table 7

Average running time (in seconds) of our algorithm for denoising ($\sigma = 50$), deblurring and inpainting (ratio: 20) on image of size 256×256 .

Scenario	Method	Total	Denoiser	Framelet
Denoising	HSSD-BM3D	167.75	158.61	1.39
	HSSD-WNNM	3552.4	3542.0	1.39
Deblurring	HSSD-BM3D	147.79	137.86	1.42
	HSSD-WNNM	3696.3	3685.1	1.42
Inpainting	HSSD-BM3D	338.26	319.11	1.49
	HSSD-WNNM	3436.13	3425.5	1.49

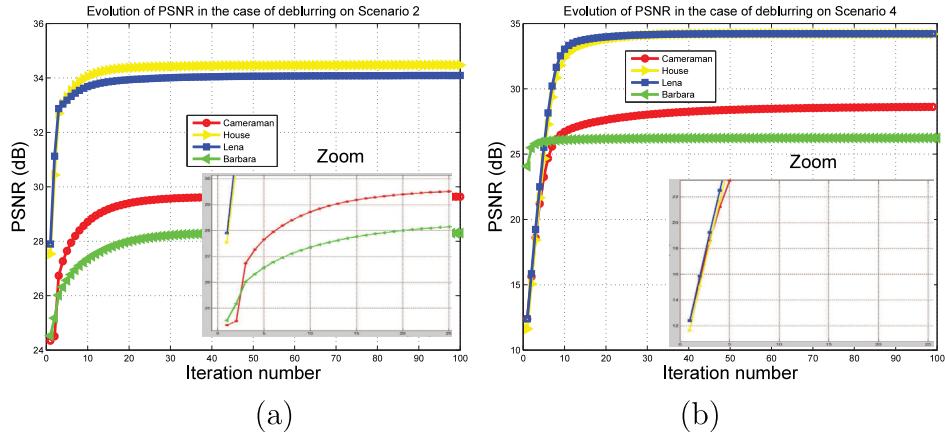


Fig. 12. Progress of PSNR (dB) v.s. the iteration number in the cases of deblurring by the proposed method. (a) Scenario 2, (b) Scenario 4. The initial few iterations are zoomed in for better visual comparison.

4.6. Algorithm stability

Figs. 12 and 13 plots the evolutions of the PSNR values versus the iteration numbers in the cases of deblurring and inpainting, respectively. It can be clearly observed that with the growth of iteration number, all the PSNR curves increase monotonically and ultimately become flat, which indicates a reliable converging behavior.

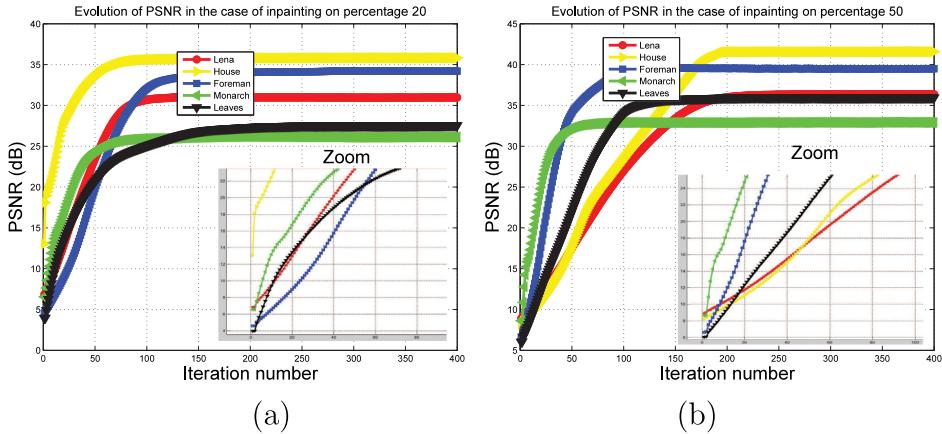


Fig. 13. Progress of PSNR (dB) v.s. the iteration number in the cases of inpainting by the proposed method. (a) Percentage 20, (b) Percentage 50. The initial few iterations are zoomed in for better visual comparison.

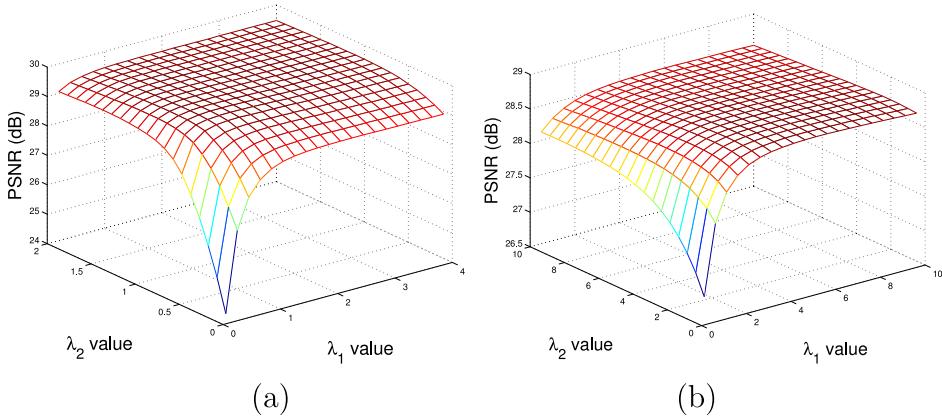


Fig. 14. Evolution of PSNR values with regularization parameters λ_1 and λ_2 in the cases of deblurring. (a) Scenario 2, (b) Scenario 4. Test image: *Cameraman*.

4.7. Effect of the parameters

This section discusses how the performance of our algorithm is affected by the two regularization parameters λ_1 , λ_2 , two penalty parameters μ_1 , μ_2 and the thresholding parameter ρ . The curves of PSNR values versus the λ_1 and λ_2 choices in the cases of deblurring are presented in Fig. 14. It can be observed that the small changes of λ_1 and λ_2 do not result in a dramatic impact on the final recovery. The curves of PSNR values versus the μ_1 and μ_2 choices in the cases of inpainting are displayed in Fig. 15. It can be seen that appropriate choice of μ_1 and μ_2 can achieve a more rapid convergence. The PSNR curves versus the different ρ values are provided in Fig. 16, which indicates the final performance of our algorithm is not very sensitive to the thresholding parameter ρ within an appropriate range. It is worth emphasizing that we also have the similar observations from other cases (due to the limited page space, these results are omitted here), which demonstrate that the proposed algorithm is quite robust regarding these involved parameters.

4.8. Limitations and possible improvements

Despite the good recovery performance of our algorithm for single image restoration, it has its own applicable conditions and limitations. Specifically, we will elaborate on the following three aspects, i.e., initial reference image choice, support detection scheme and sparse representation capacity.

As a multi-convex relaxation algorithm, the proposed algorithm prefers the high quality initial reference images instead of the low quality initial images as expected. While the proposed algorithm is tolerable with certain amount of false detections of nonzero components, the ratio of false detections cannot be too high expectedly [27,28]. When the quality of the reference image is quite poor, the detected support set might be completely unreliable, resulting in a very low

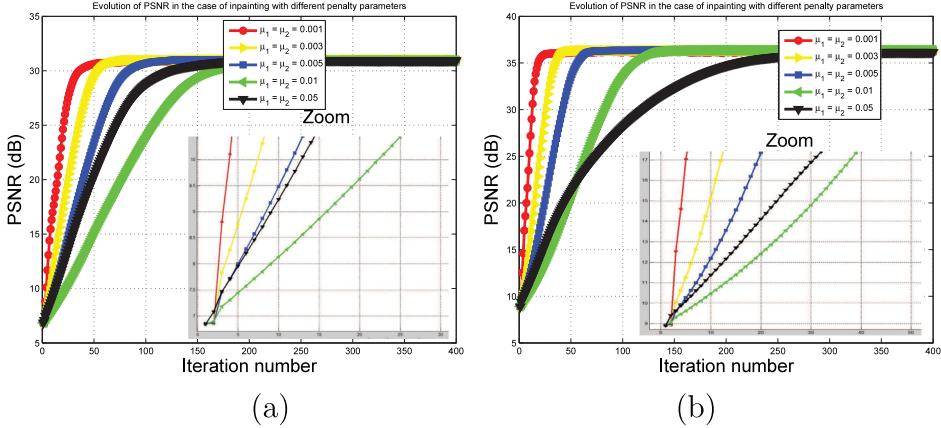


Fig. 15. Evolution of PSNR values with penalty parameters μ_1 and μ_2 in the cases of inpainting. (a) Data percentage 20, (b) Data percentage 50. Test image: *Lena*. The initial few iterations are zoomed in for better visual comparison.

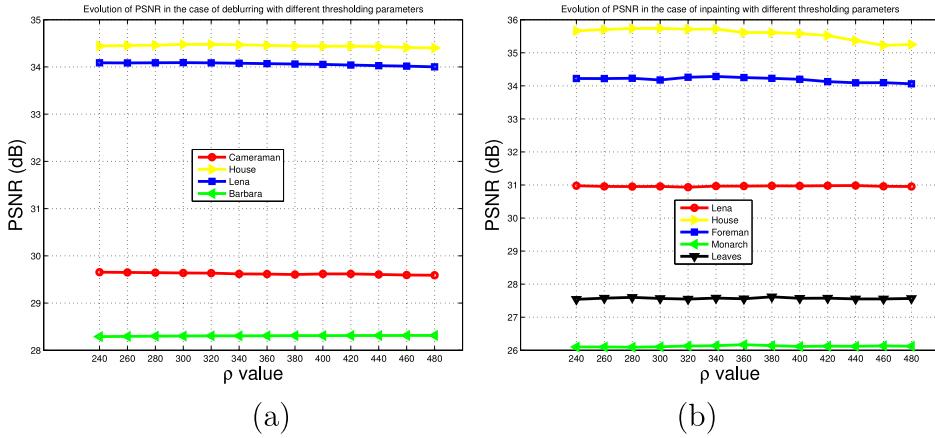


Fig. 16. Evolution of PSNR values with thresholding parameters ρ within an appropriate range. (a) Scenario 2, (b) Data percentage 20.

quality recovery result. On the other hand, since the initial reference image can be provided by any off-the-shelf image restoration method, we do have the flexibility to choose appropriate high quality initial reference image.

Meanwhile, while in general the proposed algorithm can further improve the recovery performance based on a reasonable quality reference image, it cannot be always the case. For instance, from Table 6, we can see that our algorithm slightly lags behind the GSR in few cases. The reason is that the wavelet frame and BM3D adopted in our algorithm, just like any data representation model, have their own limitation in terms of the representation capacity, especially for characterizing heavily complex images. Therefore, in practice, we can compare the quality between the recovered image and reference image, and choose the better one as the final output, this is justified when computational aspects are less important. In addition, the simple prespecified linear B-spline framelet is used in this work. However, more sophisticated sparsifying transforms, such as the data-driven sparsifying operator [49] can be adopted depending on different problems, aiming to achieve more pleasing recovery results via these refined sparsifying transforms.

5. Conclusion

In this paper, we propose an elegant and flexible model-based single image restoration method using the hybrid priors. More specifically, we incorporate the *local* prior (sparsity and support priors of wavelet tight frame) with the explicit formulation and the *nonlocal* prior (BM3D/WNNM denoiser prior) with the implicit optimization objective into a unified model, which is able to simultaneously preserve the salient edges, homogenous regions and complex textures. Comprehensive experimental results show that the proposed algorithm in general performs at the level of SOTA.

Meanwhile, the proposed algorithm still needs some improvements in several aspects. While a sufficient condition on which the inner loop of our algorithm can achieve a global convergence behavior is provided, the assumptions are still relatively too strict. Specifically, for the general denoisers, it is a challenge to verify whether they satisfy the required

conditions. Thus, a possible future work could focus on the convergence analysis, including exploring the verifiable and weaker conditions for the convergence guarantee. In addition, we currently manually tuned the parameters in the experiments and thus it is expected to adjust the regularization parameters λ_1 and λ_2 , penalty parameters μ_1 and μ_2 in an automatic way.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments and constructive suggestions that have much improved the presentation of this paper. This work was partially supported by the NSFC, China (12001005, 61806134, 62076170), the start-up funding of Anhui University, China (Y040418173), the Natural Science Research Project of Anhui Universities, China (KJ2019A0005, KJ2019A0032), the Project of Natural Science Foundation of Anhui Province, China (2008085QF286), the Science and Technology Planning Project of Sichuan Province, China (2020YFG0324), Sichuan University, China Innovation Spark Project (2019SCUH0007).

References

- [1] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: Proc. IEEE CVPR, 2017, pp. 3929–3938.
- [2] L. Deng, H. Guo, T. Huang, A fast image recovery algorithm based on splitting deblurring and denoising, *J. Comput. Appl. Math.* 287 (2015) 88–97.
- [3] J. Huang, T. Huang, A nonstationary accelerating alternating direction method for frame-based Poissonian image deblurring, *J. Comput. Appl. Math.* 352 (2019) 181–193.
- [4] X. Lv, Y. Song, F. Li, An efficient nonconvex regularization for wavelet frame and total variation based image restoration, *J. Comput. Appl. Math.* 290 (2015) 553–566.
- [5] F. Li, X. Lv, Z. Deng, Regularized iterative Weiner filter method for blind image deconvolution, *J. Comput. Appl. Math.* 336 (2019) 425–438.
- [6] J. Mei, Y. Dong, T. Huang, Simultaneous image fusion and denoising by using fractional-order gradient information, *J. Comput. Appl. Math.* 351 (2019) 212–227.
- [7] A. Mazier, A. Bilger, A.E. Forte, et al., Inverse simulation for retrieving the undeformed position for hyperelastic materials: application to breast simulations, in: 14th World Congress in Computational Mechanics, WCCM, 2020.
- [8] A. Míra, A. Carton, S. Muller, Y. Payan, A biomechanical breast model evaluated with respect to MRI data collected in three different positions, *Clin. Biomech.* 60 (2018) 191–199.
- [9] S. Wang, W. Guo, T. Huang, G. Raskutti, A fast image recovery algorithm based on splitting deblurring and denoising, *J. Comput. Appl. Math.* 311 (2015) 551–564.
- [10] M. Sellier, An iterative method for the inverse elasto-static problem, *J. Fluids Struct.* 27 (8) (2011) 1461–1470.
- [11] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6) (2017) 1256–1272.
- [12] H. Liu, Z. Fu, J. Han, et al., Single image super-resolution using multi-scale deep encoder-decoder with phase congruency edge map guidance, *Inform. Sci.* 473 (2019) 44–58.
- [13] A. Yang, B. Yang, Z. Ji, et al., Lightweight group convolutional network for single image super-resolution, *Inform. Sci.* 516 (2020) 220–233.
- [14] Y. Romano, M. Elad, P. Milanfar, The little engine that could: Regularization by denoising (RED), *SIAM J. Imag. Sci.* 10 (4) (2017) 1804–1844.
- [15] K. Dabov, A. Foi, V. Katkovnik, et al., Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.* 16 (8) (2007) 2080–2095.
- [16] S. Gu, L. Zhang, W. Zuo, et al., Weighted nuclear norm minimization with application to image denoising, in: Proc. IEEE CVPR, 2014, pp. 2862–2869.
- [17] A. Danielyan, V. Katkovnik, K. Egiazarian, BM3D frames and variational image deblurring, *IEEE Trans. Image Process.* 21 (4) (2012) 1715–1728.
- [18] S.V. Venkatakrishnan, C.A. Bouman, B. Wohlberg, Plug-and-play priors for model based reconstruction, in: Proc. IEEE GCSIP, 2013, pp. 945–948.
- [19] S. Ono, Primal-dual plug-and-play image restoration, *IEEE Signal Process. Lett.* 24 (8) (2017) 1108–1112.
- [20] G.T. Buzzard, S.H. Chan, S. Sreehari, et al., Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium, *SIAM J. Imag. Sci.* 11 (3) (2018) 2001–2020.
- [21] S.H. Chan, X. Wang, O.A. Elgendy, Plug-and-play ADMM for image restoration: Fixed point convergence and applications, *IEEE Trans. Comput. Imag.* 3 (1) (2017) 84–98.
- [22] U. Kamilov, H. Mansour, B. Wohlberg, A plug-and-play priors approach for solving nonlinear imaging inverse problems, *IEEE Signal Process. Lett.* 24 (12) (2017) 1872–1876.
- [23] S. Sreehari, S.V. Venkatakrishnan, B. Wohlberg, et al., Plug-and-play priors for bright field electron tomography and sparse interpolation, *IEEE Trans. Comput. Imag.* 2 (4) (2016) 408–423.
- [24] A.M. Teodoro, J.M. Bioucas-Dias, M.A.T. Figueiredo, A convergent image fusion algorithm using sceneadapted Gaussian-mixture-based denoising, *IEEE Trans. Image Process.* 28 (1) (2019) 451–463.
- [25] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (1) (2010) 1–122.
- [26] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, *J. Fourier Anal. Appl.* 14 (5–6) (2008) 877–905.
- [27] Y. Wang, W. Yin, Sparse signal reconstruction via iteration support detection, *SIAM J. Imag. Sci.* 3 (3) (2010) 462–491.
- [28] L. He, Y. Wang, C. Bao, N. Zhou, Wavelet frame truncated ℓ_1 -regularized image deblurring, in: Proc. IEEE WCSP, 2016, pp. 1–5.
- [29] T. Zhang, Analysis of multi-stage convex relaxation for sparse regularization, *J. Mach. Learn. Res.* 11 (2010) 1081–1107.
- [30] J. Cai, S. Osher, Z. Shen, Split Bregman methods and frame based image restoration, *Multiscale Model. Simul.* 8 (2) (2009) 337–369.
- [31] J. Oliveira, J.M. Bioucas-Dias, M. Figueiredo, Adaptive total variation image deblurring: A majorization-minimization approach, *Signal Process.* 89 (9) (2009) 1683–1693.
- [32] A.M. Teodoro, J.M. Bioucas-Dias, M.A.T. Figueiredo, Image restoration and reconstruction using variable splitting and class-adapted image priors, in: Proc. IEEE Int. Conf. Image Process., ICIP, 2016, pp. 3518–3522.
- [33] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: Proc. IEEE ICCV, 2011, pp. 479–486.
- [34] W. Dong, L. Zhang, G. Shi, Nonlocally centralized sparse representation for image restoration, *IEEE Trans. Image Process.* 22 (4) (2013) 1620–1630.
- [35] J. Xu, L. Zhang, W. Zuo, D. Zhang, X. Feng, Patch group based nonlocal self-similarity prior learning for image denoising, in: Proc. IEEE ICCV, 2015, pp. 244–252.
- [36] F. Chen, L. Zhang, H. Yu, External patch prior guided internal clustering for image denoising, in: Proc. IEEE ICCV, 2015, pp. 603–611.

- [37] S. Anwar, F. Porikli, C.P. Huynh, Category-specific object image denoising, *IEEE Trans. Image Process.* 26 (11) (2017) 5506–5518.
- [38] Z. Wang, A.C. Bovik, H.R. Sheikh, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [39] J. Portilla, Image restoration through ℓ_0 analysis-based sparse optimization in tight frames, in: Proc. IEEE ICIP, 2009, pp. 3909–3912.
- [40] C.J. Schuler, H.C. Burger, S. Harmeling, et al., A machine learning approach for non-blind image deconvolution, in: Proc. IEEE CVPR, 2013, pp. 1067–1074.
- [41] S. Gu, R. Timofte, L.V. Gool, Integrating local and non-local denoiser priors for image restoration, in: Proc. IEEE ICPR, 2018, pp. 2923–2928.
- [42] T. Tirer, R. Giryes, Image restoration by iterative denoising and backward projections, *IEEE Trans. Image Process.* 28 (3) (2019) 1220–1234.
- [43] J. Zhang, D. Zhao, W. Gao, Group-based sparse representation for image restoration, *IEEE Trans. Image Process.* 23 (8) (2014) 3336–3351.
- [44] L. He, Y. Wang, Iterative support detection-based split Bregman method for wavelet frame-based image inpainting, *IEEE Trans. Image Process.* 23 (12) (2014) 5470–5485.
- [45] S. Roth, M.J. Black, Fields of experts: A framework for learning image priors, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit, 2005, pp. 860–867.
- [46] J. Zhang, D. Zhao, R. Xiong, S. Ma, et al., Image restoration using joint statistical modeling in a space-transform domain, *IEEE Trans. Circuits Syst. Video Technol.* 24 (6) (2014) 915–928.
- [47] M. Zhou, H. Chen, J. Paisley, et al., Nonparametric Bayesian dictionary learning for analysis of noisy and incomplete images, *IEEE Trans. Image Process.* 21 (1) (2012) 130–144.
- [48] M. Zhang, C. Desrosiers, High-quality image restoration using low-rank patch regularization and global structure sparsity, *IEEE Trans. Image Process.* 28 (2) (2019) 868–879.
- [49] J. Cai, H. Ji, Z. Shen, G. Ye, Data-driven tight frame construction and image denoising, *Appl. Comput. Harmon. Anal.* 37 (2014) 89–105.