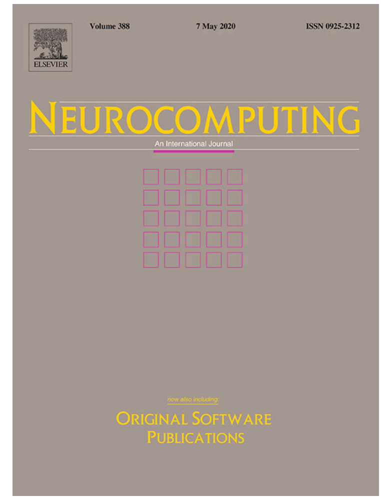
Journal Pre-proofs



Single Image Restoration Through $_2$ -relaxed Truncated $_0$ Analysis-based Sparse Optimization in Tight Frames

Liangtian He, Yilun Wang, Jun Liu, Chao Wang, Shaobing Gao

PII: S0925-2312(21)00309-X
DOI: https://doi.org/10.1016/j.neucom.2021.02.053
Reference: NEUCOM 23518

To appear in: *Neurocomputing*

Received Date: 16 July 2020
Accepted Date: 22 February 2021

Single Image Restoration Through ℓ_2-relaxed Truncated ℓ_0 Analysis-based Sparse Optimization in Tight Frames

Liangtian He[a], Yilun Wang[b], Jun Liu[c], Chao Wang[d], Shaobing Gao*[e]

[a]*Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Mathematical Sciences, Anhui University, Hefei, 230601, P. R. China.*
[b]*audience.ai (part of American Family Insurance), Chicago, IL, 60654, United States.*
[c]*School of Mathematics and Statistics, Northeast Normal University, Changchun, 130024, P. R. China*
[d]*College of Mathematics, Physics, and Information, Zhejiang Ocean University, Zhoushan, 316022, P. R. China.*
[e]*Colledge of Computer Science, Sichuan University, Chengdu, 610065, P. R. China.*

Abstract

Image restoration problems, i.e., recovery of an original high-quality image from the degraded observation, arise in various science and engineer areas. Over the past decades, the framelet-based methods are particularly investigated and adopted, owing to the excellent ability of sparse approximating the piecewise-smooth functions such as natural images. In this paper, we propose a novel tight frame-based ℓ_2-relaxed truncated ℓ_0 analysis-sparsity model that simultaneously exploiting the sparsity and support priors. The resulting nonconvex nonsmooth optimization problem is addressed by using the proposed *proximal alternating adaptive hard thresholding* (PAAHT) method. We also proved that the sequence generated by the proposed algorithm sublinearly converges. Numerical experiments on several typical image restoration problems demonstrate that the proposed method is more effective than the standard sparsity-inducing algorithms and outperforms several state-of-the-art methods in both objective and perceptual quality.

Keywords: Wavelet tight frame, Single image restoration, Adaptive hard

☆**Corresponding author.**
Email addresses: `helt@ahu.edu.cn` (Liangtian He), `yilun.wang@gmail.com` (Yilun Wang), `liuj292@nenu.edu.cn` (Jun Liu), `wangchao1321654@163.com` (Chao Wang), `gaoshaobing@scu.edu.cn` (Shaobing Gao*)

thresholding, Nonconvex nonsmooth optimization, Alternating minimization

1. Introduction

Due to the defects of imaging systems and the interference of external factors, the recorded image will inevitably be degraded during image acquisition, transmission, and storage. Image restoration aims to recover the clean image from the degraded observation, which plays an important role in many science and engineer areas such as optics, magnetic resonance imaging, computer tomography, astronomy, to name just a few. Mathematically speaking, the digital image can be viewed as a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ by stacking the columns one by one, and the generic image formation process can be formulated as the following large scale linear system:

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \omega, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the original image to be recovered, $\mathbf{y} \in \mathbb{R}^n$ is the observed low-quality image, $\omega \in \mathbb{R}^n$ is usually assumed to be the additive white gaussian noise (AWGN) with variance σ_w^2, $\mathbf{A} \in \mathbb{R}^{m\times n}$ is the degradation matrix corresponding to various image restoration applications. For instance, if $\mathbf{A}$ is an identity matrix, the problem is specially named as image denoising. If $\mathbf{A}$ is a matrix representing the blur convolution operator, the problem (1) becomes image deblurring; when $\mathbf{A}$ is a diagonal matrix whose diagonal elements are either 0 or 1, i.e., keeping or removing the corresponding pixels, then it changes to image inpainting.

Over the past decades, a variety of effective methods have been proposed to solve the above imaging inverse problems (1), which can be roughly grouped into two categories, i.e., model-based optimization methods [6, 14, 18, 24, 28, 30, 32, 35, 37, 55, 56] and discriminative learning methods [4, 11, 25, 57, 58, 59]. The model-based methods obtain the solution by directly solving a formulated optimization model, which usually involves an iterative procedure. On the contrary, given a training dataset containing degraded-clean image pairs, the discriminative learning methods could learn a nonlinear function to match the degraded observation to its underlying high-quality version. These two kinds of image restoration methods have their respective merits and drawbacks. The model-based methods are briefly flexible to tackle different restoration applications by only replacing the degraded operator. In contrast, discriminative learning methods are usually restricted to certain tasks and tend to deliver more promising performances. In this work, we limit our attention to the model-based image restoration methods since we aim to provide an effective algorithm with the flexibility to handle single image restoration for various tasks.

From the perspective of Maximum A Posterior (MAP) estimation, one can infer the solution from its degraded observation by solving the following regularization model.

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}} \left\{ J(\mathbf{x}) \triangleq \lambda \Psi_{\text{reg}}(\mathbf{x}) + \Phi_{\text{fid}}(\mathbf{x}, \mathbf{y}) \right\}, \tag{2}$$

where $\Psi_{\text{reg}}(\mathbf{x})$ is the regularizer term that regularizes the solution by enforcing certain prior constraints, $\Phi_{\text{fid}}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2$ is the data fidelity term which measures how $\mathbf{x}$ is fit to the observation $\mathbf{y}$, and λ is the regularization parameter that balances these two terms. How to choose an effective functional $\Psi_{\text{reg}}(\mathbf{x})$ in (2) is an active research topic in the field of imaging sciences. The transform domain sparsity-inducing regularization models, e.g., the well-known total variation (TV) and wavelet frame-based approaches [7, 8, 20, 26, 33, 34, 35, 60], have been specially investigated. These algorithms' key motivation is that the latent image is sparse (or compressible) in the transform domain.

In this paper, we focus on the tight frame-based method. Specifically, besides the commonly utilized sparsity prior, we further significantly improve its restoration quality by employing the associated support prior and formulating a more elegant and effective regularization model. Concretely, the main contributions of this work are summarized as follows.

- We propose a generic, simple yet effective tight frame-based ℓ_2-relaxed truncated ℓ_0 analysis-sparsity restoration model, which considers the support prior together with the sparsity prior. Compared with the conventional sparsity-only regularization model, the formulated one is noticeably helpful to preserve the sharp edges and homogeneous areas better simultaneously.
- We propose an efficient algorithm called proximal alternating adaptive hard thresholding (PAAHT) to address the resulting ℓ_0-based nonconvex nonsmooth optimization problem. Furthermore, we proved that the sequence generated by the proposed algorithm sublinearly converges.
- Comprehensive numerical experiments on several typical image restoration problems reveal that our proposed algorithm offers a noticeable boost to the conventional transform domain sparsity-inducing methods. The recovery performance is highly comparable (often better) to existing state-of-the-art methods.

The rest of this paper is outlined as follows. In the next section, we first briefly revisit some closely related works. In Section 3, we introduce the proposed tight frame-based ℓ_2-relaxed truncated ℓ_0 analysis-sparsity restoration model. The corresponding efficient algorithm PAAHT is developed to solve the resulting nonconvex-nonsmooth optimization problem. Furthermore, we also established a convergence analysis of the proposed algorithm. In Section 4, extensive experiments are conducted to illustrate the proposed algorithm's performance for several typical image restoration tasks. Some concluding remarks and possible future work are given in section 5.

2. Related work

In this section, we will briefly review several closely related works that serve as the basis of our proposed method to make the paper self-contained. To avoid

Notation	Representation
$< \mathbf{x}, \mathbf{y} >$	Inner product of vectors $\mathbf{x}$ and $\mathbf{y}$
$\mathbf{A}$	Degradation matrix corresponding to different image restoration tasks
$\mathbf{A}^T$	The transpose matrix of $\mathbf{A}$
$\mathbf{A}^{-1}$	The inverse matrix of $\mathbf{A}$
$\mathbf{I}$	Identity matrix
$\mathcal{D}$	Analysis sparsity operator
$\mathcal{D}^*$	Synthesis sparsity operator
$\lvert x \rvert$	Absolute value of scaler x
$\lvert\lvert \mathbf{x} \rvert\rvert_1$	Sum of absolute values of vector $\mathbf{x}$
$\lvert\lvert \mathbf{x} \rvert\rvert_2$	Euclid norm of vector $\mathbf{x}$
$\lvert\lvert \mathbf{x} \rvert\rvert_0$	Number of nonzero entries of vector $\mathbf{x}$
$\lvert\lvert \mathbf{x} \rvert\rvert_\infty$	The largest value (in magnitude) of vector $\mathbf{x}$
$\{\mathbf{x}^k\}$	The sequence of vectors
$\mathcal{S}(\mathbf{x})$	The index set of nonzero entries of vector $\mathbf{x}$
S	Detected support index set of analysis coefficients in the transform domain
T	Supplementary set of the detected support set S
$\mathbf{x}_T$	Truncated form of vector $\mathbf{x}$ indexed in T
λ, μ	Regularization parameters
d_k	Proximal penalty parameters
η	Thresholding parameter

Table 1: Main notations and representations.

confusion to the notations appearing in this paper, we give the main notations and representations in Table 1.

2.1. ℓ_1 and ℓ_0 analysis sparsity model

There are mainly two kinds of sparsity priors: one is synthesis sparsity prior, which assumes that the image is a linear combination of few basis functions from a dictionary; Another is the analysis sparsity prior, which assumes that the image is locally highly incoherent (generates nearly-zero responses) with most of the elements of a (typically) redundant set of kernels [35]. We focus on the analysis sparsity prior in this work. The reasons are as follows: Firstly, compared with the synthesis sparsity prior, its counterpart analysis sparsity prior may be easier to justify from the viewpoint of empirical Bayes since the analysis coefficients are directly observable. Secondly, numerical experiments have shown that the quality of the restored image by analysis sparsity method is often better than that of the synthesis sparsity approach [7].

The analysis sparsity-based ℓ_1 norm model is generally formulated as follows:

$$\text{argmin}_{\mathbf{x}} \lambda ||\mathcal{D}\mathbf{x}||_1 + \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2, \tag{3}$$

where $\mathcal{D}$ is an analysis operator such as the gradient operator[1], wavelet tight frame, shearlet and other sparsity transforms. The ℓ_1 norm model is extensively adopted attributing to its convex property, and it can be efficiently solved with convergence guarantee via many methods such as split Bregman [6], ADMM [3], primal-dual [9], etc. However, this is grounded on the consideration that the ℓ_1 norm regularization is capable of obtaining the sparsest solution if the operator $\mathbf{A}$ in (1) satisfies certain conditions [5, 12]. Unfortunately, such assumptions may be violated in real image restoration applications. Thus the ℓ_1 norm model often obtains a suboptimal solution.

During the last years, the ℓ_0 pseudo-norm regularization has attracted much more researchers' attention. As we know, the most direct way to induce sparsity is penalizing the ℓ_0 pseudo-norm of the solution, and it is reasonable to penalize the ℓ_0 pseudo-norm of the analysis coefficients instead of the ℓ_1 norm.

$$\text{argmin}_{\mathbf{x}} \lambda ||\mathcal{D}\mathbf{x}||_0 + \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2. \tag{4}$$

Many efforts have been made to address the above nonconvex and NP-hard analysis-sparsity ℓ_0 model. For instance, In [29], Lu et al. proposed the penalty decomposition (PD) method to solve the wavelet frame-based analysis ℓ_0 sparsity problem. Their numerical results demonstrated significant improvements over some commonly used ℓ_1 minimization models in terms of the quality of recovered images. However, the computational cost of PD method is relatively high. Dong et al. in [19] proposed a more efficient algorithm called the mean doubly augmented Lagrangian (MDAL) method to solve the same optimization

[1]The standard total variation (TV) regularization is defined as the ℓ_1 norm of first derivative of the underlying image in horizonal and vertical directions , which could be interpreted as the simplest analysis operator.

problem. However, the convergence analysis has not been established for the MDAL method.

2.2. ℓ_2-relaxed ℓ_0 analysis sparsity model

Utilizing the ℓ_0 pseudo-norm (i.e., counting the nonzero analysis coefficients) to directly characterize real-world images conveys an ignorable problem, that is, real images are not strictly sparse for a given analysis dictionary, but just approximately so, i.e., they are compressible. Therefore, it is not justified to directly use the ℓ_0 minimization of the analysis coefficients. In [34], Portilla proposed to model the analysis coefficients as a strictly sparse vector plus a Gaussian correction term.

$$\mathcal{D}\mathbf{x} = \mathbf{a} + \mathbf{r}, \tag{5}$$

where $\mathbf{a}$ is a strictly sparse vector $p(\mathbf{a}) \propto \exp(-\frac{1}{\alpha}||\mathbf{a}||_0)$, $\mathbf{r}$ is the Gaussian residual term $p(\mathbf{r}) \propto \exp(-\frac{1}{2\sigma_r^2}||\mathbf{r}||_0)$, and the formulated ℓ_2-relaxed ℓ_0 analysis sparsity optimization problem has the form.

$$\text{argmin}_{\mathbf{x},\mathbf{a}}||\mathbf{a}||_0 + \lambda||\mathcal{D}\mathbf{x} - \mathbf{a}||_2^2 + \mu||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2, \tag{6}$$

where $\lambda = \frac{\alpha}{2\sigma_r^2}$, $\mu = \frac{\alpha}{2\sigma_w^2}$. The joint optimization problem (6) in [10] is tackled by iteratively marginally minimizing $(\mathbf{x}, \mathbf{a})$, but the convergence analysis is not provided. In [44], Yang et.al proposed the *proximal alternating iterative hard thresholding* (PAIHT) method, and proved that the generated sequence $(\mathbf{x}^k, \mathbf{a}^k)$ converges to the critical point of the joint ℓ_2-relaxed ℓ_0 minimization problem.

In [35], the analysis coefficients are interpreted as a Bernoulli mixture of a high-probability low-variance Gaussian and a low-probability high-variance plateau. Given a set of analysis operators ,$\mathcal{D}_j, j = 1, 2, \ldots, J$, Portilla et.al proposed a variant of the ℓ_0 pseudo-norm:

$$p(\mathbf{x}) \propto \exp(-\sum_{j=1}^{J} \frac{1}{\alpha_j}||\mathcal{D}_j\mathbf{x}||_{(0,K_j)}) \tag{7}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{D}_j\mathbf{x} \in \mathbb{R}^{M_j}$, with $M_j \geq n$, and α_j controls the relative influence on the prior of each linear representation $\mathcal{D}_j\mathbf{x}$. Here $||\mathbf{v}||_{(0,K)}$ is defined by,

$$
\begin{aligned}
||\mathbf{v}||_{(0,K)} &= \sum_{m=1}^{M} \min(|\frac{v_m}{K}|^2, 1) \\
&= ||\Theta_h(\mathbf{v}, K)||_0 + \frac{1}{K^2}||\mathbf{v} - \Theta_h(\mathbf{v}, K)||_2^2 \\
&= \min_{\mathbf{a}}||\mathbf{a}||_0 + \frac{1}{K^2}||\mathbf{v} - \mathbf{a}||_2^2 \qquad (8)
\end{aligned}
$$

where $\Theta_h(\mathbf{v}, K)$ denotes the hard thresholding operator that set the components smaller than K in magnitude as zero, and it is easily to derive that $\lim_{K\to 0}||\mathbf{v}||_{(0,K)} = ||\mathbf{v}||_0$. Accordingly, Portilla et.al proposed a variant of the ℓ_2-relaxed ℓ_0 analysis-sparsity model.

$$
\text{argmin}_{\mathbf{x}} \frac{1}{2\sigma_w^2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \sum_{j=1}^{J} \frac{1}{\alpha_j} \min_{\mathbf{a}_j} \left\{ ||\mathbf{a}_j||_0 + \frac{1}{K_j}||\mathcal{D}_j\mathbf{x} - \mathbf{a}_j||_2^2 \right\} \qquad (9)
$$

Similarly, the above nonvonvex optimization problem (9) can be efficiently solved by alternate marginal minimizing $\{\mathbf{a}_j\}$ and $\mathbf{x}$, and the succession $\{\mathbf{x}^k\}$ is proved to be unconditionally convergent [35].

2.3. *Truncated ℓ_0 regularization model*

For an effective regularizer, it is crucial to exploit more appropriate prior knowledge of latent image. Particularly, for the ℓ_0 analysis sparsity model, He et al. [27] further employed "twin" of the analysis sparsity prior, i.e., the associated support prior (locations of nonzero analysis coefficients) can be incorporated to significantly boost the recovery performance, and proposed the support-driven truncated ℓ_0 analysis sparsity model.

$$
\text{argmin}_{\mathbf{x}} \lambda||(\mathcal{D}\mathbf{x})_T||_0 + \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2, \qquad (10)
$$

where $(\mathcal{D}\mathbf{x})_T$ is the truncated version of $\mathcal{D}\mathbf{x}$, i.e., a subvector indexed in T of $\mathcal{D}\mathbf{x}$ after truncation, S denotes the index set of nonzero analysis coefficients, and T is the supplementary set of S, i.e., $S = T^C$. Compared with the standard ℓ_0 regularization model, it is obvious that the nonzero analysis coefficients (particularly those with large magnitudes) should not be forced toward 0 by penalization, i.e., the entries corresponding to support set should be truncated out of the regularizer term, leading to a more reasonable and effective regularization model [27].

The challenges of (10) mainly lie in two aspects. Firstly, the detection of the support index set S, as the ground truth image can not be available in practice. Empirically, the strategy of *first-pass estimation* is an effective way to perform the support detection, i.e., a relatively reliable support set can be detected with an open interface. The basic idea is as follows. The degraded image is initially operated by employing an existing algorithm, which can be resorted to a current off-the-shelf, state-of-the-art method, such as the powerful nonlocal patched methods. Then given a relatively high-quality reference image $\mathbf{x}^{\text{ref}}$ of the *first-pass estimation*, S can be roughly computed as the indices of analysis coefficients whose magnitudes are greater than a prescribed threshold.

$$S := \left\{ i : |(\mathcal{D}\mathbf{x}^{\text{ref}})_i| > \frac{||\mathcal{D}\mathbf{x}^{\text{ref}}||_\infty}{\eta} \right\}, \tag{11}$$

where the thresholding parameter $\eta > 0$. Secondly, due to the nonconvex and nonsmooth property of the truncated ℓ_0 optimization problem (10), an efficient algorithmic solver with convergence guarantee is not a trivial task. In [27], He et al. proposed a variant of the MDAL method to solve (10), though the numerical results show a stable behavior, the strict convergence discussion has not been established.

2.4. Nonlocal patched methods

Started with nonlocal means method (NLM) [2] for image denoising, a flurry of nonlocal patched approaches have been proposed over the past decades, which are built upon the nonlocal self-similarity (NSS) prior, i.e., similar structures in patches can be found across a natural image. Compared to the local transform domain sparsity-inducing methods, the nonlocal patched approaches often deliver more impressive performances. In the past few years, NSS prior is also combined with the ideas of a low-rank model and adaptive dictionary learning, and has generated a diversity of state-of-the-art image restoration algorithms [10, 13, 14, 18, 22, 43, 45-51].

3. Our proposed model and algorithmic solver

3.1. ℓ_2-relaxed truncated ℓ_0 analysis sparsity model

The ℓ_2-relaxed ℓ_0 analysis-sparsity model [34, 35] demonstrates that analysis coefficients are compressible, rather than strictly sparse. Thus it is not justified to penalize the analysis coefficients into ℓ_0 pseudo-norm directly. However, the associated support prior is not exploited in the formulated model. By contrast, while the truncated ℓ_0 analysis-sparsity model proposed by He et al. [27] exploits the support prior in the regularizer term, it directly penalizes the analysis coefficients. More importantly, the variant of the MDAL method in [27] for the truncated ℓ_0 minimization problem lacks a strict convergence guarantee. In the context of analysis sparsity image modeling, both the ℓ_2-relaxed formulation and support prior are beneficial to the final recovery performance in their own aspects. Therefore, it is preferable to incorporate them together into a unified model.

In this work, we propose a generic, simple yet effective single image restoration model taking the ℓ_2-relaxed ℓ_0 analysis-sparsity formulation and associated support prior into consideration simultaneously.

$$\text{argmin}_{\mathbf{x},\mathbf{a}} \psi(\mathbf{x},\mathbf{a}) := \lambda ||\mathbf{a}_T||_0 + \frac{\mu}{2}||\mathcal{D}\mathbf{x} - \mathbf{a}||_2^2 + \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 \tag{12}$$

where $\mathcal{D}$ denotes the tight frame analysis operators[2], $p(\mathbf{a}) \propto \exp(-\frac{1}{\alpha}||\mathbf{a}_T||_0)$, $p(\mathbf{r}) \propto \exp(-\frac{1}{2\sigma_r^2}||\mathbf{r}||_0)$, and $||\mathbf{a}_T||_0$ denotes the truncated ℓ_0 pseudo-norm which

[2]We focus on the $\mathcal{D}$ as tight frames, i.e., $\mathcal{D}^*\mathcal{D} = \mathbf{I}$ in this paper, due to its simplicity and high efficiency. More importantly, leading to a convergent algorithmic solver.

incorporates the support prior of $\mathcal{D}\mathbf{x}$ into the regularizer term. $\lambda = \frac{2\sigma_w^2}{\alpha}$ and $\mu = \frac{\sigma_w^2}{\sigma_r^2}$ are the regularization parameters.

3.2. Proposed algorithm

To the formulated nonconvex nonsmooth ℓ_2-relaxed truncated ℓ_0 minimization problem, we propose a simple iterative procedure consisting of the following two steps.

$$\begin{cases} \mathbf{x}^{k+1} = \mathrm{argmin}_{\mathbf{x}} \frac{1}{2}||\mathbf{A}\mathbf{x} - \mathbf{y}||_2^2 + \frac{\mu}{2}||\mathcal{D}\mathbf{x} - \mathbf{a}^k||_2^2 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad k = 0, 1, 2, \dots \\ \mathbf{a}^{k+1} = \mathrm{argmin}_{\mathbf{a}} \lambda||\mathbf{a}_T||_0 + \frac{\mu}{2}||\mathcal{D}\mathbf{x}^{k+1} - \mathbf{a}||_2^2 + \frac{d_k}{2}||\mathbf{a} - \mathbf{a}^k||_2^2 \end{cases} \tag{13}$$

where $d_k > 0$ denotes the proximal penalty parameters, and $\mathbf{a}^0$ is the initialization vector.

For the first subproblem of (13), according to the first-order optimality condition, it is easy to obtain that

$$\mathbf{x}^{k+1} = (\mathbf{A}^T\mathbf{A} + \mu\mathcal{D}^*\mathcal{D})^{-1}(\mathbf{A}^T\mathbf{y} + \mu\mathcal{D}^*\mathbf{a}^k). \tag{14}$$

For the second subproblem of (13), it is obvious to attain that

$$\begin{aligned} \mathbf{a}^{k+1} &= \mathrm{argmin}_{\mathbf{a}} \lambda||\mathbf{a}_T||_0 + \frac{\mu}{2}||\mathcal{D}\mathbf{x}^{k+1} - \mathbf{a}||_2^2 + \frac{d_k}{2}||\mathbf{a} - \mathbf{a}^k||_2^2 \\ &= \mathrm{argmin}_{\mathbf{a}} \lambda||\mathbf{a}_T||_0 + \frac{\mu + d_k}{2}||\mathbf{a} - \frac{\mu\mathcal{D}\mathbf{x}^{k+1} + d_k\mathbf{a}^k}{\mu + d_k}||_2^2. \end{aligned} \tag{15}$$

Note that (15) can be equivalently decomposed as

$$\mathrm{argmin}_{\mathbf{a}} \underbrace{\lambda||\mathbf{a}_T||_0 + \frac{\mu + d_k}{2}||\mathbf{a}_T - \left(\frac{\mu\mathcal{D}\mathbf{x}^{k+1} + d_k\mathbf{a}^k}{\mu + d_k}\right)_T||_2^2}_{\text{Part A}} + \underbrace{\frac{\mu + d_k}{2}||\mathbf{a}_S - \left(\frac{\mu\mathcal{D}\mathbf{x}^{k+1} + d_k\mathbf{a}^k}{\mu + d_k}\right)_S||_2^2}_{\text{Part B}}. \tag{16}$$

Then it is obvious that the first part is a standard ℓ_0 minimization problem, which can be solved via the hard thresholding operator. The second part is a quadratic optimization problem, its optimum is $\mathbf{a}_S = (\frac{\mu\mathcal{D}\mathbf{x}^{k+1} + d_k\mathbf{a}^k}{\mu + d_k})_S$. In summary, the solution of $\mathbf{a}$-subproblem can be derived as below.

$$\mathbf{a}^{k+1} = \mathcal{H}_{T,\lambda,\mu,d_k}\left(\frac{\mu\mathcal{D}\mathbf{x}^{k+1} + d_k\mathbf{a}^k}{\mu + d_k}\right), \tag{17}$$

where the *support-guided adaptive hard thresholding operator* is defined as

$$\mathcal{H}_{T,\lambda,\mu,d_k}(z_i) = \begin{cases} 0, & \text{if} \quad i \in T \quad \text{and} \quad |z_i| < \sqrt{\frac{2\lambda}{\mu+d_k}}. \\ z_i, & \text{otherwise.} \end{cases} \tag{18}$$

Concluding the above analysis, the iterative loop of (13) can be rewritten as

$$\begin{cases} \mathbf{x}^{k+1} = (\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}(\mathbf{A}^T\mathbf{y} + \mu\mathcal{D}^*\mathbf{a}^k) \\ \mathbf{a}^{k+1} = \mathcal{H}_{T,\lambda,\mu,d_k}\left(\frac{\mu\mathcal{D}\mathbf{x}^{k+1}+d_k\mathbf{a}^k}{\mu+d_k}\right) \end{cases} \quad k = 0, 1, 2, \ldots \tag{19}$$

For convenience, since the second step in the iterative loop is connected with the adaptive hard thresholding operator, the proposed algorithm in this paper is termed as *proximal alternating adaptive hard thresholding* (PAAHT) method.

Algorithm 1 PAAHT for ℓ_2-relaxed truncated ℓ_0 regularization model

Input: Given the observed image $\mathbf{y}$ and the degraded operator $\mathbf{A}$; Tight frame operator $\mathcal{D}$; Regularization parameters $\lambda > 0$, $\mu > 0$; Proximal penalty parameters $d_k \in [d_{\min}, d_{\max}]$; Thresholding parameter $\eta > 0$. Support detection (11) via a reference image $\mathbf{x}^{\text{ref}}$ with an open interface, e.g., the off-the-shelf image restoration methods. Set $k = 0$, $\mathbf{a}^0 = \mathcal{D}\mathbf{x}^{\text{ref}}$.

1: **for** $k = 0$ **to** k_{max}, **do**

2: $\mathbf{x}^{k+1} = (\mathbf{A}^T\mathbf{A} + \mu\mathcal{D}^*\mathcal{D})^{-1}(\mathbf{A}^T\mathbf{y} + \mu\mathcal{D}^*\mathbf{a}^k)$.

3: $\mathbf{a}^{k+1} = \mathcal{H}_{T,\lambda,\mu,d_k}\left(\frac{\mu\mathcal{D}\mathbf{x}^{k+1}+d_k\mathbf{a}^k}{\mu+d_k}\right)$.

4: Update μ and d_k if necessary.

5: If stopping criterion satisfies go to Output.

6: **end for**

Output: $\mathbf{a}^{k+1}$, $\mathbf{x}^{k+1}$.

3.3. Convergence analysis

It is noted that Yang et.al in [44] has proposed the PAIHT method for the tight frame-based ℓ_2-relaxed ℓ_0 analysis-sparsity optimization problem (6) with strict convergence guarantee. Following his work, the convergence properties of the PAAHT method for the proposed tight frame-based ℓ_2-relaxed truncated ℓ_0 analysis-sparsity minimization problem (12) is established as follows.

Theorem 3.1. *Assume that $\mathbf{A}^T\mathbf{A}$ is a positive definite matrix, i.e., $\mathbf{A}^T\mathbf{A} \succ 0$. Given T ($T = S^C$), the sequence generated by iterative procedure (19) PAAHT converges and there exist a positive integer k_0 such that $\mathcal{S}(\mathbf{a}^k) = \mathcal{S}(\mathbf{a}^{k_0})$ for all $k \geq k_0$. Let $(\mathbf{x}^*, \mathbf{a}^*)$ denote the limit of the sequence $\{(\mathbf{x}^k, \mathbf{a}^k)\}$, it also obtains*

$$||\mathbf{a}^k - \mathbf{a}^*||_2 \leq \mathcal{O}(1/\sqrt{k}), \quad ||\mathbf{x}^k - \mathbf{x}^*||_2 \leq \mathcal{O}(1/\sqrt{k}), \quad k \to \infty.$$

Proof: By substituting the solution of first subproblem (14) into the iterative scheme (13), it obtains

$$\begin{aligned}
\lambda||\mathbf{a}_T||_0 \quad &+ \quad \frac{\mu}{2}||\mathbf{a} - \mathbf{a}^k + \mathbf{a}^k - \mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}(\mathbf{A}^T\mathbf{y} + \mu\mathcal{D}\mathbf{a}^k)||_2^2 + \frac{d_k}{2}||\mathbf{a} - \mathbf{a}^k||_2^2 \\
&= \quad \lambda||\mathbf{a}_T||_0 + \frac{\mu}{2}||\mathbf{a} - \mathbf{a}^k + (\mathbf{I} - \mu\mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D})\mathbf{a}^k - \mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}||_2^2 \\
&+ \quad \frac{d_k}{2}||\mathbf{a} - \mathbf{a}^k||_2^2.
\end{aligned}$$

Since $\mathcal{D}$ is a tight frame analysis operator, i.e., $\mathcal{D}^*\mathcal{D} = \mathbf{I}$, the nonzero eigenvalues of $\mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D}$ coincide with eigenvalues of $(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}$. Additionally, based on the condition $\mathbf{A}^T\mathbf{A} \succ 0$, it gives

$$0 < \mathbf{I} - \mu\mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D} \leq 1.$$

Therefore, $\mathbf{I} - \mu\mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D}$ is a symmetric positive definite matrix. According to the Cholesky factorization [21], there exists a unique lower triangular matrix $\mathbf{L}$ such that

$$\mathbf{L}^T\mathbf{L} = \mathbf{I} - \mu\mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D}$$

For convenience, we define

$$\mathcal{G}(\mathbf{a}) = \frac{\mu}{2}||\mathbf{L}^T\mathbf{a} - \mathbf{L}^{-1}\mathbf{f}||_2^2, \quad \text{where} \quad \mathbf{f} = \mathcal{D}^*(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}.$$

Then it obtains $\nabla\mathcal{G}(\mathbf{a}^k) = \mu(\mathbf{L}\mathbf{L}^T\mathbf{a}^k - \mathbf{f})$. Thus the second subproblem about $\mathbf{a}$ in (13) is equal to

$$\mathbf{a}^{k+1} \in \mathrm{argmin}_{\mathbf{a}}\lambda||\mathbf{a}_T||_0 + \frac{\mu}{2}||\mathbf{a} - \mathbf{a}^k + \frac{1}{\mu}\nabla\mathcal{G}(\mathbf{a}^k)||_2^2 + \frac{d_k}{2}||\mathbf{a} - \mathbf{a}^k||_2^2. \tag{20}$$

Let $\rho_{\max}(\mathbf{L}\mathbf{L}^T)$ denotes the largest eigenvalue of $\mathbf{L}\mathbf{L}^T$. Then $\nabla\mathcal{G}$ is Lipschitz continuous with constant $\mu\rho_{\max}(\mathbf{L}\mathbf{L}^T)(\leq \mu)$ and

$$\mathcal{G}(\mathbf{a}^{k+1}) \leq \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu\rho_{\max}(\mathbf{L}\mathbf{L}^T)}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2. \tag{21}$$

Let us define

$$\mathcal{F}(\mathbf{a}) = \lambda||\mathbf{a}_T||_0 + \mathcal{G}(\mathbf{a}).$$

Based on (20) and (21), it can derive that

$$\begin{aligned}
\mathcal{F}(\mathbf{a}^{k+1}) &= \lambda||(\mathbf{a}^{k+1})_T||_0 + \mathcal{G}(\mathbf{a}^{k+1}) \\
&\leq \lambda||(\mathbf{a}^{k+1})_T||_0 + \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu\rho_{\max}(\mathbf{L}\mathbf{L}^T)}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \\
&\leq \lambda||(\mathbf{a}^{k+1})_T||_0 + \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu + d_k}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \\
&\leq \lambda||(\mathbf{a}^k)_T||_0 + \mathcal{G}(\mathbf{a}^k) = \mathcal{F}(\mathbf{a}^k)
\end{aligned} \tag{22}$$

From the inequality (22), it indicates that sequence $\{\mathcal{F}(\mathbf{a}^k)\}$ is nonincreasing and

$$\mathcal{F}(\mathbf{a}^k) - \mathcal{F}(\mathbf{a}^{k+1}) \geq P - Q = \frac{\mu + d_k + \mu\rho_{\max}(\mathbf{L}\mathbf{L}^T)}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \tag{23}$$

where

$$\begin{cases} P = \lambda||(\mathbf{a}^{k+1})_T||_0 + \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu\rho_{\max}(\mathbf{L}\mathbf{L}^T)}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \\ Q = \lambda||(\mathbf{a}^{k+1})_T||_0 + \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu+d_k}{2}||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \end{cases}$$

Since $\mathcal{G}(\mathbf{a})$ is bounded below, it is obvious that $\{\mathcal{F}(\mathbf{a}^k)\}$ is bounded below. Therefore, sequence $\{\mathcal{F}(\mathbf{a}^k)\}$ converges to a finite value as $k \to \infty$. Together with (22), it obtains that

$$\lim_{k\to\infty} ||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2. \tag{24}$$

Based on (20), it gets

$$\mathbf{a}^{k+1} \in \text{argmin}_{\mathbf{a}} \lambda||\mathbf{a}_T||_0 + \frac{\mu + d_k}{2}||\mathbf{a}||_2^2 - (\mu + d_k)\left\langle \mathbf{a}, \mathbf{a}^k - \frac{1}{\mu + d_k}\nabla\mathcal{G}(\mathbf{a}^k)\right\rangle. \tag{25}$$

Let g_i^k denotes the i-th element of vector $\nabla\mathcal{G}(\mathbf{a}^k)$. For the entrywise minimization problem (25),

$$a_i^{k+1} \in \text{argmin}_a \lambda||a_T||_0 + \frac{\mu + d_k}{2}a^2 - (\mu + d_k)\left\langle a, a_i^k - \frac{g_i^k}{\mu + d_k}\right\rangle.$$

According to the support-guided adaptive hard-threshloding operator (18), it can easily obtains that

$$\begin{cases} a_i^{k+1} = 0, \quad \text{if} \quad i \in T \quad \text{and} \quad |a_i^k - \frac{g_i^k}{\mu+d_k}| < \sqrt{\frac{2\lambda}{\mu+d_k}}. \\ a_i^{k+1} = a_i^k - \frac{g_i^k}{\mu+d_k}, \quad \text{otherwise.} \end{cases} \tag{26}$$

Let $\mathcal{S}(\mathbf{a}^k)$ denotes the support set of the vector $\mathbf{a}^k$. If $\mathcal{S}(\mathbf{a}^k) \neq \mathcal{S}(\mathbf{a}^{k+1})$, it can derive that

$$||(\mathbf{a}^{k+1} - \mathbf{a}^k)_S||_2^2 \geq \min\left\{\frac{2\lambda}{\mu + d_{k-1}}, \frac{2\lambda}{\mu + d_k}\right\} \tag{27}$$

by discussing the following four cases:

1. If $i \in \overline{\mathcal{S}(\mathbf{a}^k)} \cap \overline{\mathcal{S}(\mathbf{a}^{k+1})} \cap S$. It obtains that $a_i^k = a_i^{k+1} = 0$. Thus, $|a_i^{k+1} - a_i^k| = 0$.

2. If $i \in \mathcal{S}(\mathbf{a}^k) \cap \overline{\mathcal{S}(\mathbf{a}^{k+1})} \cap S$. It obtains $a_i^k \neq 0$ and $a_i^{k+1} = 0$. Since $a_i^k \neq 0$

only holds in the case that

$$a_i^k = a_i^{k-1} - \frac{g_i^{k-1}}{\mu + d_{k-1}} \geq \sqrt{\frac{2\lambda}{\mu + d_{k-1}}}.$$

Otherwise $a_i^k = 0$ is a contradiction. Thus $|a_i^{k+1} - a_i^k| \geq \sqrt{\frac{2\lambda}{\mu+d_{k-1}}}$.

3. If $i \in \overline{\mathcal{S}(\mathbf{a}^k)} \cap \mathcal{S}(\mathbf{a}^{k+1}) \cap S$. It obtains $a_i^k = 0$ and $a_i^{k+1} \neq 0$. Similarly,

$$a_i^{k+1} = a_i^k - \frac{g_i^k}{\mu + d_k} \geq \sqrt{\frac{2\lambda}{\mu + d_k}}.$$

Thus $|a_i^{k+1} - a_i^k| \geq \sqrt{\frac{2\lambda}{\mu+d_k}}$.

4. If $i \in \mathcal{S}(\mathbf{a}^k) \cap \mathcal{S}(\mathbf{a}^{k+1}) \cap S$. It obtains $a_i^k \neq 0$ and $a_i^{k+1} \neq 0$. Clearly, $|a_i^{k+1} - a_i^k| \geq 0$.

Therefore, according to (24) and (27), recall that $(\mathbf{a}^{k+1})_T = (\mathbf{a}^k)_T = \mathbf{0}$, it gets that $\mathcal{S}(\mathbf{a}^k)$ does not change when k is sufficiently large. Assuming there exists a positive integer k_0 such that $\mathcal{S}(\mathbf{a}^k) = \mathcal{S}(\mathbf{a}^{k_0})$ holds for all $k \geq k_0$. Then for any $k \geq k_0$, the iterative scheme (20) becomes the equivalently constraint minimization problem as follows

$$\mathbf{a}^{k+1} = \mathrm{argmin}_{\mathbf{a}\in\Omega} \left\{ \mathcal{G}(\mathbf{a}^k) + < \mathbf{a}^{k+1} - \mathbf{a}^k, \nabla\mathcal{G}(\mathbf{a}^k) > + \frac{\mu + d_k}{2} ||\mathbf{a}^{k+1} - \mathbf{a}^k||_2^2 \right\} \tag{28}$$

where the constraint set $\Omega = \{\mathbf{a} : \mathcal{S}(\mathbf{a}) = \mathcal{S}(\mathbf{a}^{k_0})\}$. Let $\mathbf{a}^*$ denotes the optimum of minimization problem

$$\min_{\mathbf{a}\in\Omega} \mathcal{G}(\mathbf{a}).$$

Then with the similar proof of Theorem 2.2 in [21], it derives that the sequence $\{\mathbf{a}^k\}$ generated by (27) satisfies

$$\mathcal{G}(\mathbf{a}^{k+l}) - \mathcal{G}(\mathbf{a}^*) \leq \frac{L}{2l} ||\mathbf{a}^k - \mathbf{a}^*||_2^2, \tag{29}$$

where $L = \mu + \sup_{k>0} d_k$. Let $\rho_{\min}(\mathbf{L}\mathbf{L}^T)$ denote the smallest eigenvalue of $\mathbf{L}\mathbf{L}^T$. Since $\mathcal{G}(\mathbf{a})$ is strongly convex with modulus $\mu\rho_{\min}(\mathbf{L}\mathbf{L}^T)$, and $\nabla\mathcal{G}(\mathbf{a}^*) = 0$, it gets

$$||\mathbf{a}^{k+l} - \mathbf{a}^*||_2^2 \leq \frac{2}{\mu\rho_{\min}(\mathbf{L}\mathbf{L}^T)}(\mathcal{G}(\mathbf{a}^{k+l}) - \mathcal{G}(\mathbf{a}^*)) \tag{30}$$

where l is a positive integer. Combining (29) and (30), it obtains

$$||\mathbf{a}^{k_0+l} - \mathbf{a}^*||_2^2 \leq \frac{L}{l\mu\rho_{\min}(\mathbf{L}\mathbf{L}^T)}||\mathbf{a}^{k_0} - \mathbf{a}^*||_2^2,$$

which implies

$$||\mathbf{a}^k - \mathbf{a}^*||_2 \leq \mathcal{O}(1/\sqrt{k}), \quad k \to \infty.$$

Therefore, it obtains

$$\lim_{k\to\infty} \mathbf{a}^k = \mathbf{a}^*.$$

Together with (14), it derives that $\mathbf{x}^* = \lim_{k\to\infty} \mathbf{x}^k$ and

$$||\mathbf{x}^k - \mathbf{x}^*||_2 \leq \rho_{\mathrm{s}}||\mathbf{a}^k - \mathbf{a}^*||_2 \leq \mathcal{O}(1/\sqrt{k}), \quad k \to \infty.$$

where ρ_{s} is the spectral norm of matrix $\mu(\mathbf{A}^T\mathbf{A} + \mu\mathbf{I})^{-1}\mathcal{D}$. $\square$

4. Numerical experiments

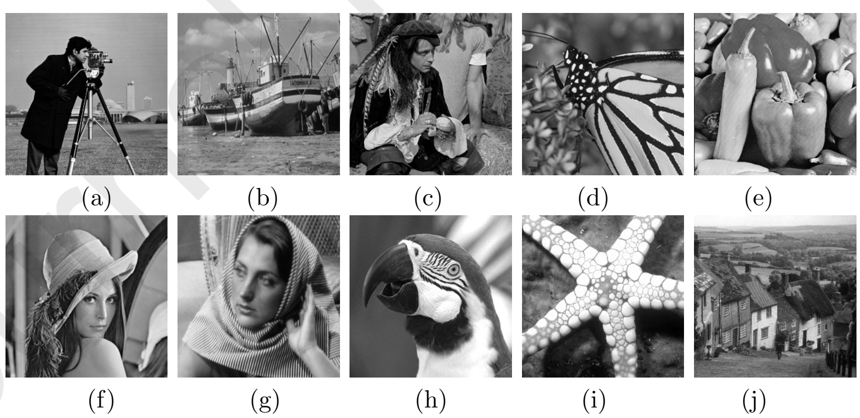


Figure 1: 10 typical test images. (a) Cameraman, (b) Boat, (c) Man, (d) Monarch, (e) Peppers. (f) Lena, (g) Barbara, (h) Parrot, (i) Starfish, (j) Goldhill. All sizes of the test images are 256×256.

Scenario	PSF	σ
1	$1/(z_1^2+z_2^2), z_1, z_2 = -7, \ldots, 7$	$\sqrt{2}$
2	$1/(z_1^2+z_2^2), z_1, z_2 = -7, \ldots, 7$	2
3	uniform 9	$\sqrt{2}$
4	uniform 9	2
5	fspecial(gaussian,25,1.6)	$\sqrt{2}$
6	fspecial(gaussian,25,1.6)	2
7	fspecial(motion,15,30)	$\sqrt{2}$
8	fspecial(motion,15,30)	2

Table 2: Eight typical deblurring scenarios with different blur PSFs and two noise variances ($\sigma = \sqrt{2}$ and $\sigma = 2$) in the first set.

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

Table 3: Six benchmark deblurring scenarios with different blur PSFs and noise variances in the second set.

4.1. Experimental settings

In this section, comprehensive experiments are conducted to demonstrate the performance of our proposed model and corresponding algorithm. Specifically, two fundamental and widely investigated image restoration applications: image denoising and deblurring, are considered to illustrate the effectiveness of our proposed PAAHT algorithm. The ten standard natural images, which consist of complex components in different scales and with different patterns (pixel intensity ranges from 0 to 255), as shown in Figure 1, are used for our denoising test and deblurring in the first set.

4.2. Evaluation metrics

To quantitatively measure the quality of the recovered images, the Peak Signal-to-Noise Ratio (PSNR) and Improved Signal-to-Noise Ratio (ISNR) are

Method	Regularizer term	Sparsity prior	Support prior	ℓ_2-relaxed	Convergence guarantee
SB [6]	ℓ_1	√	×	×	√
MDAL [19]	ℓ_0	√	×	×	×
L0-AbS [34]	ℓ_2-relaxed ℓ_0	√	×	√	×
ℓ_2-r-ℓ_0 [35]	ℓ_2-relaxed ℓ_0	√	×	√	√
PAIHT [44]	ℓ_2-relaxed ℓ_0	√	×	√	√
SDSR [27]	truncated ℓ_0	√	√	×	×
PAAHT	ℓ_2-relaxed truncated ℓ_0	√	√	√	√

Table 4: The comparison of several closely-related tight frame-based image restoration methods.

computed, which are defined as

$$\mathrm{PSNR}(\mathbf{x}, \bar{\mathbf{x}}) := 10\log_{10}\left(\frac{255^2 n}{||\mathbf{x}-\bar{\mathbf{x}}||_2^2}\right),$$

and

$$\mathrm{ISNR}(\mathbf{x}, \bar{\mathbf{x}}, \mathbf{y}) := 10\log_{10}\left(\frac{||\mathbf{y}-\bar{\mathbf{x}}||_2^2}{||\mathbf{x}-\bar{\mathbf{x}}||_2^2}\right),$$

where $\bar{\mathbf{x}}$, $\mathbf{y}$ and $\mathbf{x}$ are the ground-truth image, corrupted image and recovered image with pixel number n, respectively. Additionally, the perceptual quality metric Structural SIMilarity (SSIM) is also calculated to evaluate the visual quality.

$$\mathrm{SSIM}(\mathbf{x}, \bar{\mathbf{x}}) := \frac{(2\mu_{\mathbf{x}}\mu_{\bar{\mathbf{x}}} + C_1)(2\sigma_{\mathbf{x}\bar{\mathbf{x}}} + C_2)}{(\mu_{\mathbf{x}}^2 + \mu_{\bar{\mathbf{x}}}^2 + C_1)(\sigma_{\mathbf{x}}^2 + \sigma_{\bar{\mathbf{x}}}^2 + C_2)}.$$

where $\mu_{\mathbf{x}}$ and $\mu_{\bar{\mathbf{x}}}$ are the averages of $\mathbf{x}$ and $\bar{\mathbf{x}}$, respectively. $\sigma_{\mathbf{x}}$ and $\sigma_{\bar{\mathbf{x}}}$ are the variances of $\mathbf{x}$ and $\bar{\mathbf{x}}$, respectively. $\sigma_{\mathbf{x}\bar{\mathbf{x}}}$ is the covariance of $\mathbf{x}$ and $\bar{\mathbf{x}}$. The positive constants C_1 and C_2 act as stabilizing constants for near-zero denominator values. The higher SSIM value means the better visual quality, and note that the SSIM value would equal to 1 in the cases of ideal restorations.

4.3. Implementation details

Compared with other analysis operators, e.g., the gradient operator, the wavelet frame can adaptively choose proper differential operators in different regions of a given image according to the order of the singularity of the latent solutions. For the sake of simplicity and efficiency, the sparsifying operator $\mathcal{D}$ in our proposed model (12) is chosen as the tight wavelet frame. Specifically, the linear B-spline framelet with decomposition level 1 is utilized. The parameters of the proposed algorithm and other competing approaches are all carefully tuned to achieve the optimal PSNR and SSIM values. Throughout the numerical experiments, the stopping criterion of our proposed algorithm is set as when the maximum number of allowed outer iterations k_{max} has been reached, or the relative differences between consecutive iterations satisfy

$$\min\left\{\frac{||\mathbf{x}^k-\mathbf{x}^{k-1}||_2}{||\mathbf{x}^k||_2}, \frac{||\mathbf{A}\mathbf{x}^k-\mathbf{y}||_2}{||\mathbf{y}||_2}\right\} < \varepsilon.$$

Empirically, we set $k_{max} = 500$ and $\varepsilon = 1 \times 10^{-8}$. For the nonlocal patched methods, all public source codes are downloaded from the authors' website, and the default parameters are adopted. We believe our comparison would be fair under these specific settings.

Note that the reference image (11) of our proposed algorithmic framework can be resorted to the off-the-shelf, state-of-the-art image restoration methods. The WNNM and IDD-BM3D have remained the state-of-the-art denoising and deblurring methods in the literature since its publication. For the sake of consistency, following previous works, the results of WNNM and IDD-BM3D are adopted as the reference images for support detection (11) in the cases of denoising and deblurring, respectively. The regularization parameters λ and μ are adjusted to give the best performance in terms of PSNR and SSIM values. Empirically, the penalty parameter d_k is set as $d_k = 2\mu$, the thresholding parameters in both cases are fixed as $\eta = 300$. All the implementations are carried out in Matlab R2018a installed on a desktop with AMD Ryzen 7 3700X 8-Core Processor (3.60GHz) and 32 GB memory. The Matlab source code of our proposed algorithm can be downloaded at the website at `https://github.com/jackygsb/PAAHT-Code`.

4.4. Image denoising

In the experiments of image denoising, the noisy images are synthesized by adding Gaussian white noise with four standard deviations $\sigma = 30, 50, 70, 100$. We compare our proposed PAAHT algorithm against several representative

Image	σ	BM3D [15]	EPLL [52]	NCSR [18]	WNNM [22]	PGPD [43]	PCLR [10]	SSC-GSM [14]	**PAAHT**
Cameraman	30	28.64	28.36	28.63	28.81	28.54	28.82	28.60	**28.89**
		0.8373	0.8316	0.8396	0.8403	0.8259	0.8430	0.8298	**0.8473**
	50	26.12	26.02	26.20	26.46	26.46	**26.56**	26.56	26.56
		0.7826	0.7617	0.7844	0.7849	0.7774	**0.7944**	0.7861	0.7936
	70	24.61	24.51	24.61	24.91	24.94	**25.03**	24.88	25.02
		0.7426	0.7082	0.7501	0.7453	0.7379	0.7551	0.7464	**0.7588**
	100	23.08	22.86	22.97	23.41	23.23	23.49	23.26	**23.51**
		0.6926	0.6351	0.7065	0.6971	0.6776	0.7159	0.6914	**0.7207**
Boat	30	27.38	27.48	27.22	27.56	27.41	**27.68**	27.27	27.63
		0.7796	0.7838	0.7693	0.7836	0.7744	0.7865	0.7668	**0.7896**
	50	24.91	25.09	24.74	25.21	25.07	25.26	24.97	**25.29**
		0.6748	0.6833	0.6642	0.6918	0.6833	0.6928	0.6760	**0.6992**
	70	23.57	23.61	23.32	23.82	23.70	23.82	23.66	**23.86**
		0.6125	0.6091	0.6018	0.6282	0.6195	0.6261	0.6163	**0.6314**
	100	22.29	22.20	21.99	22.43	22.39	22.40	22.34	**22.45**
		0.5468	0.5257	0.5377	0.5513	0.5473	0.5555	0.5502	**0.5557**
Man	30	26.40	26.52	26.40	26.59	26.45	26.60	26.49	**26.64**
		0.7591	0.7677	0.7574	0.7667	0.7579	0.7649	0.7600	**0.7728**
	50	24.08	24.30	24.04	24.32	24.29	24.36	24.25	**24.37**
		0.6581	0.6648	0.6502	0.6710	0.6628	0.6646	0.6543	**0.6756**
	70	22.83	22.95	22.68	22.98	22.94	23.01	22.92	**23.03**
		0.5914	0.5917	0.5827	0.6063	0.5954	0.5946	0.5867	**0.6077**
	100	21.57	21.59	21.31	21.65	21.63	**21.71**	21.57	21.66
		0.5195	0.5095	0.5048	0.5304	0.5179	0.5196	0.5071	**0.5325**
Monarch	30	28.36	28.35	28.47	28.92	28.49	28.83	28.83	**28.94**
		0.8822	0.8789	0.8856	0.8918	0.8853	0.8930	0.8923	**0.8936**
	50	25.82	25.78	25.77	26.32	26.00	26.25	**26.37**	26.36
		0.8200	0.8124	0.8260	0.8343	0.8269	0.8370	0.8336	**0.8378**
	70	24.24	24.07	24.04	24.63	24.34	24.59	**24.76**	24.67
		0.7674	0.7533	0.7768	0.7861	0.7756	0.7903	**0.7932**	0.7912
	100	22.52	22.23	22.13	22.95	22.56	22.93	**23.06**	22.98
		0.7021	0.6771	0.7124	0.7256	0.7029	**0.7364**	0.7351	0.7348
Peppers	30	28.66	28.61	28.46	28.84	28.70	**28.92**	28.70	28.87
		0.8169	0.8152	0.8119	0.8195	0.8164	**0.8225**	0.8165	0.8203
	50	26.17	26.27	26.03	26.43	26.31	**26.56**	26.47	26.45
		0.7579	0.7520	0.7570	0.7628	0.7578	**0.7695**	0.7662	0.7646
	70	24.74	24.68	24.41	24.79	24.79	**24.99**	24.90	24.82
		0.7147	0.7004	0.7154	0.7145	0.7118	**0.7288**	0.7244	0.7212
	100	23.17	23.08	22.74	23.21	23.21	**23.47**	23.34	23.23
		0.6582	0.6368	0.6648	0.6607	0.6530	**0.6849**	0.6749	0.6713
Lena	30	29.46	29.19	29.36	29.73	29.60	29.69	29.53	**29.76**
		0.8584	0.8477	0.8597	0.8667	0.8622	0.8647	0.8598	**0.8683**
	50	26.90	26.69	26.95	27.27	27.15	27.19	27.09	**27.29**
		0.7920	0.7732	0.8011	0.8068	0.7990	0.8054	0.7964	**0.8100**
	70	25.48	25.08	25.35	25.82	25.60	25.63	25.75	**25.84**
		0.7407	0.7107	0.7529	0.7622	0.7645	0.7574	0.7560	**0.7675**
	100	23.87	23.46	23.64	24.35	24.02	24.08	24.30	**24.38**
		0.6740	0.6345	0.7048	0.7025	0.6780	0.7033	0.7016	**0.7135**
Barbara	30	29.08	27.59	28.77	29.48	28.93	28.82	29.22	**29.50**
		0.8618	0.8209	0.8554	0.8720	0.8565	0.8556	0.8616	**0.8727**
	50	26.42	24.87	26.15	26.85	26.27	26.19	26.46	**26.88**
		0.7698	0.6943	0.7575	0.7901	0.7613	0.7633	0.7657	**0.7906**
	70	24.86	23.30	24.38	25.22	24.72	24.59	25.14	**25.24**
		0.6973	0.6030	0.6768	0.7203	0.6887	0.6886	0.7117	**0.7211**
	100	23.20	21.90	22.72	23.48	23.11	23.03	23.52	**23.53**
		0.6092	0.5135	0.5965	0.6273	0.6039	0.6057	0.6335	**0.6288**
Parrot	30	30.45	30.11	30.44	30.85	30.43	30.75	30.83	**30.88**
		0.8803	0.8667	0.8814	0.8834	0.8778	0.8823	0.8799	**0.8875**
	50	27.99	27.61	27.93	28.31	28.02	28.29	**28.43**	28.36
		0.8407	0.8154	0.8478	0.8450	0.8382	0.8479	0.8407	**0.8532**
	70	26.39	25.93	26.15	26.60	26.36	26.64	**26.80**	26.66
		0.8068	0.7695	0.8188	0.8137	0.8026	0.8193	0.8080	**0.8266**
	100	24.70	24.03	24.26	24.89	24.59	24.79	**25.05**	24.93
		0.7592	0.7047	0.7789	0.7716	0.7462	0.7809	0.7576	**0.7951**
Starfish	30	27.72	27.75	27.75	28.11	27.67	27.98	27.70	**28.15**
		0.8087	0.8110	0.8082	0.8183	0.8045	0.8152	0.7974	**0.8205**
	50	25.20	25.32	25.19	25.56	25.22	25.48	25.31	**25.59**
		0.7119	0.7161	0.7094	0.7289	0.7110	0.7265	0.7093	**0.7304**
	70	23.85	23.82	23.79	24.08	23.87	23.99	23.88	**24.10**
		0.6488	0.6410	0.6471	0.6603	0.6474	0.6582	0.6421	**0.6625**
	100	22.52	22.40	22.46	22.69	22.58	22.61	22.56	**22.70**
		0.5810	0.5596	0.5818	0.5861	0.5786	0.5891	0.5752	**0.5902**
Goldhill	30	26.99	26.93	26.86	27.11	26.90	27.07	26.96	**27.16**
		0.7125	0.7136	0.7032	0.7180	0.7029	0.7102	0.7139	**0.7229**
	50	25.12	25.02	24.90	25.21	25.11	25.14	24.90	**25.23**
		0.6141	0.6087	0.5935	0.6201	0.6090	0.6098	0.5999	**0.6273**
	70	24.00	23.86	23.73	24.07	24.01	23.95	23.91	**24.08**
		0.5516	0.5405	0.5287	0.5554	0.5460	0.5394	0.5366	**0.5596**
	100	22.86	22.72	22.52	22.83	**22.88**	22.82	22.86	22.85
		0.4873	0.4711	0.4655	0.4846	0.4801	0.4754	0.4718	**0.4851**
Avg.	30	28.31	28.09	28.24	28.60	28.31	28.52	28.41	**28.64**
		0.8197	0.8137	0.8172	0.8260	0.8164	0.8238	0.8178	**0.8296**
	50	25.87	25.70	25.79	26.19	25.99	26.13	26.08	**26.24**
		0.7422	0.7282	0.7391	0.7536	0.7427	0.7511	0.7428	**0.7582**
	70	24.45	24.18	24.25	24.69	24.53	24.62	24.66	**24.73**
		0.6874	0.6627	0.6851	0.6992	0.6889	0.6958	0.6921	**0.7048**
	100	22.98	22.65	22.67	23.19	23.02	23.13	23.19	**23.22**
		0.6230	0.5868	0.6254	0.6337	0.6186	0.6367	0.6298	**0.6428**

Table 5: 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.

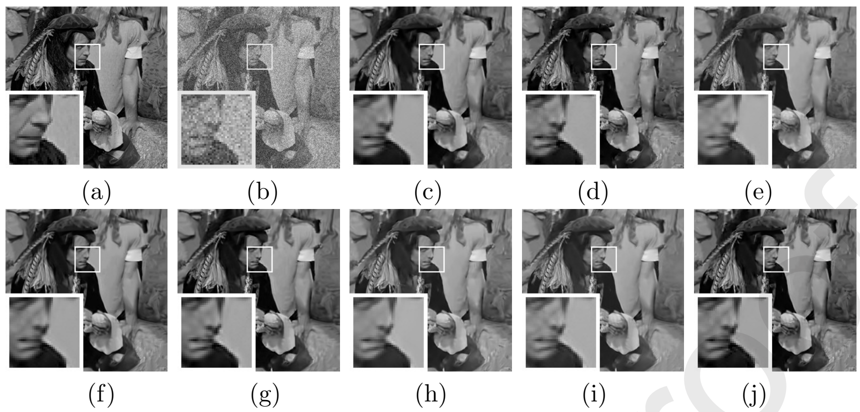


Figure 2: Denoising performance comparison on the *Man* image (256×256) with moderate noise corruption ($\sigma = 30$). (a) original image, (b) noisy image (PSNR=10.17; SSIM=0.1400), (c) BM3D [15] (PSNR=26.40; SSIM=0.7591), (d) EPLL [52] (PSNR=26.52; SSIM=0.7677), (e) NCSR [18] (PSNR=26.40; SSIM=0.7574), (f) WNNM [22] (PSNR=26.59; SSIM=0.7667), (g) PGPD [43] (PSNR=26.45; SSIM=0.7579), (h) PCLR [10] (PSNR=26.60; SSIM=0.7649), (i) SSC-GSM [14] (PSNR=26.49; SSIM=0.7600), (j) PAAHT (PSNR=26.64; SSIM=0.7728).

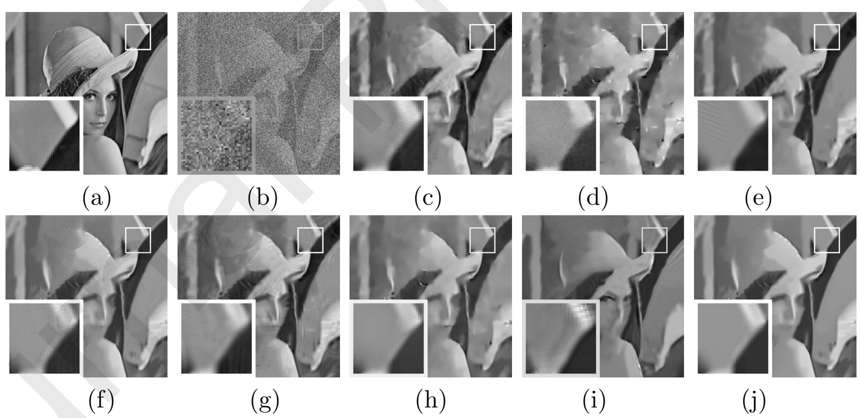


Figure 3: Denoising performance comparison on the *Lena* image (256×256) with heavy noise corruption ($\sigma = 100$). (a) original image, (b) noisy image (PSNR=8.10; SSIM=0.0606), (c) BM3D [15] (PSNR=23.87; SSIM=0.6740), (d) EPLL [52] (PSNR=23.46; SSIM=0.6345), (e) NCSR [18] (PSNR=23.64; SSIM=0.7048), (f) WNNM [22] (PSNR=24.35; SSIM=0.7025), (g) PGPD [43] (PSNR=24.02; SSIM=0.6780), (h) PCLR [10] (PSNR=24.08; SSIM=0.7033), (i) SSC-GSM [14] (PSNR=24.30; SSIM=0.7016), (j) PAAHT (PSNR=24.38; SSIM=0.7135).

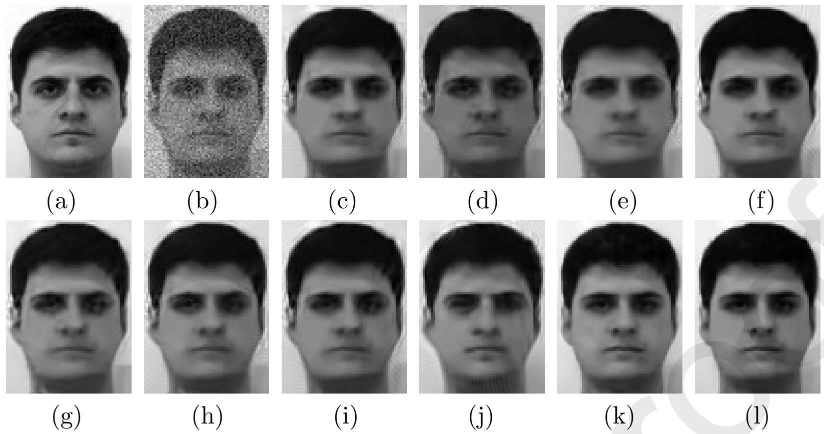


Figure 4: Denoising performance comparison. (a) original image; (b) noisy image ($\sigma =$ 20; PSNR=22.14; SSIM=0.4807); (c) BM3D [15] (PSNR=31.65; SSIM=0.9172); (d) E-PLL [52] (PSNR=31.86; SSIM=0.9192); (e) NCSR [18] (PSNR=31.57; SSIM=0.9148); (f) WNNM [22] (PSNR=32.05; SSIM=0.9173); (g) PGPD [43] (PSNR=31.88; S-SIM=0.9162); (h) PCLR [10] (PSNR=31.94; SSIM=0.9186); (i) SAIST [13] (P-SNR=31.74; SSIM=0.9127); (j) SSC-GSM [14] (PSNR=32.10; SSIM=0.9171); (k) TIP'17 [1] (PSNR=32.60; SSIM=0.9282); (l) PAAHT (PSNR= 32.92; SSIM=0.9327).

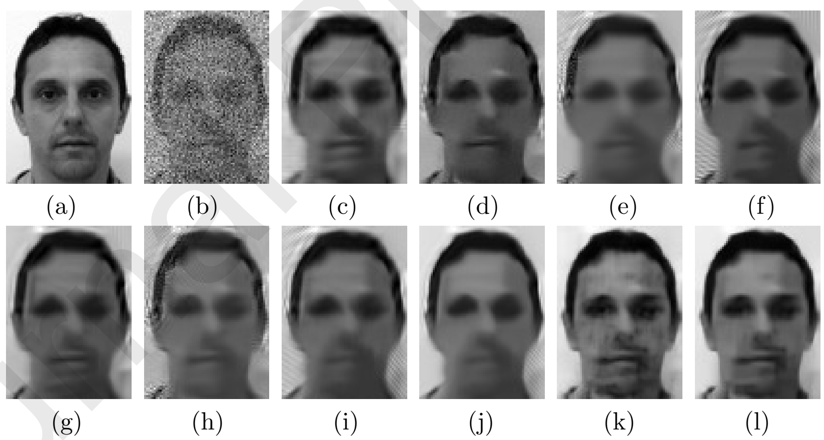


Figure 5: Denoising performance comparison. (a) original image; (b) noisy image ($\sigma =$ 50; PSNR=14.18; SSIM=0.1951); (c) BM3D [15] (PSNR=26.55; SSIM=0.8136); (d) E-PLL [52] (PSNR=26.71; SSIM=0.8039); (e) NCSR [18] (PSNR=25.61; SSIM=0.7817); (f) WNNM [22] (PSNR=26.25; SSIM=0.7946); (g) PGPD [43] (PSNR=26.72; S-SIM=0.8122); (h) PCLR [10] (PSNR=25.68; SSIM=0.7908); (i) SAIST [13] (P-SNR=26.33; SSIM=0.7886); (j) SSC-GSM [14] (PSNR=27.35; SSIM=0.8145); (k) TIP'17 [1] (PSNR=27.85; SSIM=0.8329); (l) PAAHT (PSNR=28.05; SSIM=0.8433).

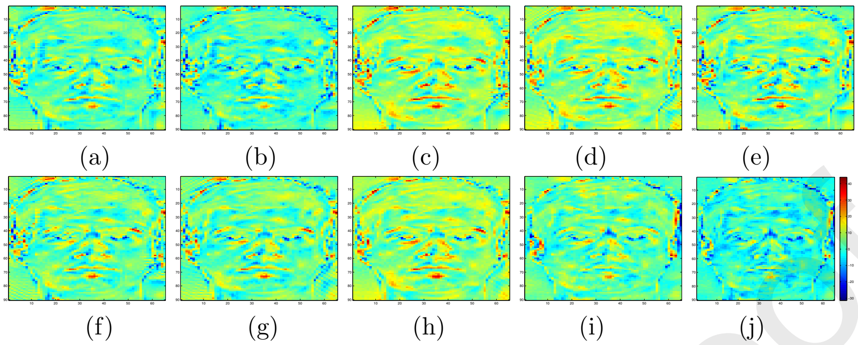


Figure 6: The difference maps comparion of different methods of Figure 4. (a) BM3D [15] (b) EPLL [52] (c) NCSR [18] (d) WNNM [22] (e) PGPD [43] (f) PCLR [10] (g) SAIST [13] (h) SSC-GSM [14] (i) TIP'17 [1] (j) PAAHT.

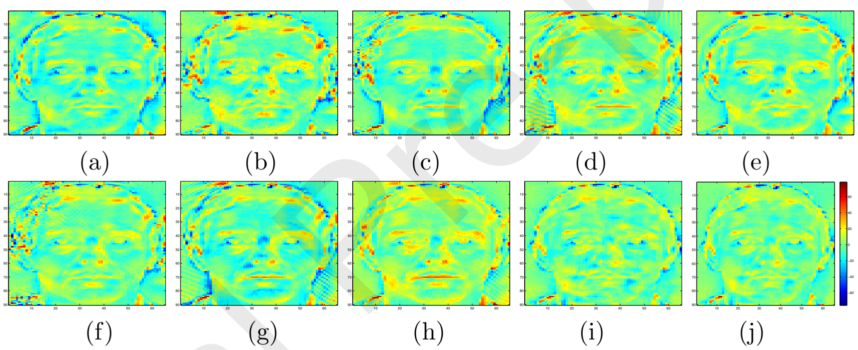


Figure 7: The difference maps comparion of different methods of Figure 5. (a) BM3D [15] (b) EPLL [52] (c) NCSR [18] (d) WNNM [22] (e) PGPD [43] (f) PCLR [10] (g) SAIST [13] (h) SSC-GSM [14] (i) TIP'17 [1] (j) PAAHT.

σ	BM3D	EPLL	NCSR	WNNM	PGPD	PCLR	SAIST	SSC-GSM	TNRD	DnCNN	FFDNet	GSRC	PAAHT
35	28.40	28.03	28.29	28.69	28.41	28.59	28.44	28.54	-	28.82	28.92	28.55	28.81
	0.8139	0.8018	0.8107	0.8190	0.8125	0.8187	0.8152	0.8123	-	-	0.8315	0.8185	0.8223
50	26.72	26.35	26.56	27.05	26.81	26.94	26.79	26.90	26.81	27.18	27.32	26.95	27.09
	0.7681	0.7475	0.7658	0.7772	0.7666	0.7763	0.7696	0.7675	-	-	0.7907	0.7735	0.7819
75	24.91	24.48	24.60	25.23	24.98	25.10	24.96	25.14	-	25.20	25.49	25.12	25.26
	0.7065	0.6738	0.7081	0.7202	0.7070	0.7208	0.7148	0.7135	-	-	0.7357	0.7211	0.7269

Table 6: Average PSNR and SSIM values of different methods for noise levels 35, 50 and 75 on Set12 dataset.

Noise level σ	PSNR (mean/variance)	SSIM (mean/variance)
35	28.7798/0.0190	$0.8323/4.6248 \times 10^{-4}$
50	27.1128/0.0192	$0.7997/4.3591 \times 10^{-4}$
75	25.0954/0.0221	$0.7592/5.5508 \times 10^{-4}$

Table 7: The mean and variance values of 100 times repetitive and random experiments by PAAHT for noise levels 35, 50 and 70. Test image: *Cameraman* (256×256).

state-of-the-art denoising methods, i.e., **BM3D** [15], **EPLL** [52], **NCSR** [18], **WNNM** [22], **PGPD** [43], **PCLR** [10], **SAIST** [13], **SSC-GSM** [14], **TNRD** [4], **DnCNN** [58], **FFDNet** [59], **GSRC** [45]. The PSNR and SSIM results of several tested methods are reported in Table 5, The highest PSNR and S-SIM values for each image and on each noise level is highlighted in bold. As can be seen, for the collection of 10 test images, the proposed PAAHT achieves highly comparable denoising performance to other competitors. On average, the PAAHT algorithm consistently achieves the best denoising performance on all the four noise levels , in terms of both PSNR and SSIM values. Table 6 reports the average PSNR and SSIM values of different methods for noise levels 35, 50 and 75 on the benchmark Set12 dataset. It can be observed that PAAHT surpasses all the model-based denoising methods, while is slightly inferior to the FFDNet, which is an impressive deep learning based denoising method. Table 7 gives the mean and variance values of 100 times repetitive and random experiments by PAAHT for noise levels 35, 50 and 70, note that other scenarios have the similar observations, it can be concluded that our algorithm has a reliable stability.

Figures 2 and 3 display the visual comparison of denoising results for two typical images (*Man* and *Lena*) at moderate ($\sigma = 30$) and heavy ($\sigma = 100$) noise levels, respectively. It can be observed from Figure 2 that the PAAHT delivers the best visual quality at the moderate noise level. By contrast, restored images by other competing algorithms tend to suffer from noticeable artifacts especially around the smooth areas. When the noise contamination is very severe, the superiority of PAAHT to other competing approaches is easier to justify. As observed from the highlighted window of Figure 3, PAAHT is visually more pleasant and particularly effective in reconstructing both the smooth and the texture/edge regions. Additionally, besides the natural image denoising, the proposed PAAHT algorithm is also preliminarily compared with the state-of-the-art method [1] for class-specific image denoising scenarios. Figures 4 and 5 show the visual comparison for face images [3]. Moreover, for a better visual comparison, following the implementation in [31], the difference maps of the recovered images in Figures 4 and 5 are plotted in Figures 6 and 7, respectively. It can be clearly observed that the denoised image by the proposed PAAHT has much fewer artifacts than other methods, and is visually especially pleasant around the smooth areas.

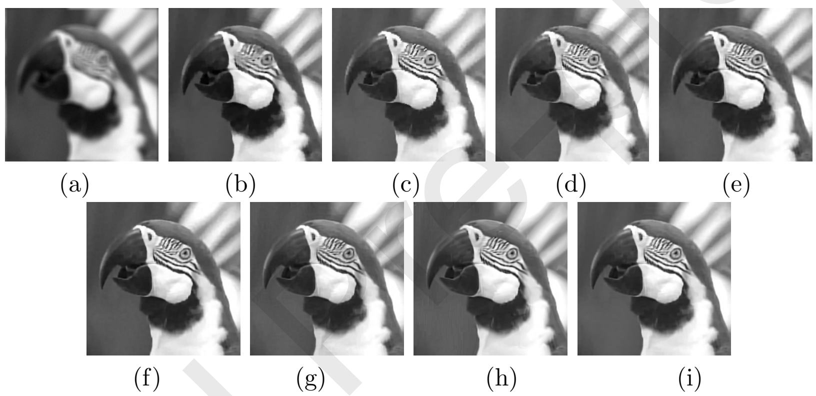


Figure 8: Visual quality comparison of image deburring. (a) Blurred and noisy image (*Parrot* (256×256), Uniform kernel: (9×9, $\sigma = \sqrt{2}$)). (b) recovery image by Split Bregman [6] (PSNR=27.39; SSIM=0.8708), (c) MDAL [19](PSNR=28.64; SSIM=0.8868), (d) PAIHT [44] (PSNR=28.67; SSIM=0.8828), (e) ℓ_2-r-ℓ_0 [35] (PSNR=29.14; SSIM=0.8882), (f) SDSR [27] (PSNR=30.24; SSIM=0.8982), (g) IDD-BM3D [16] (PSNR=29.98; SSIM=0.8914), (h) SNSS [46] (PSNR=30.39; SSIM=0.8933), (i) PAAHT (PSNR=30.31; SSIM=0.9010).

4.5. Image deblurring

In the experiments of image deblurring, two sets of blur kernels are considered. In the first set, four blur kernels (see Table 2), with additive Gaussian

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

Scenario	Method	*C.man*	*Boat*	*Man*	*Monarch*	*Peppers*	*Lena*	*Barbara*	*Parrots*	*Starfish*	*Goldhill*	**Avg.**
1	Split Bregman [6]	29.55 0.8830	29.87 0.8716	29.07 0.8688	31.76 0.9408	30.58 0.8845	31.98 0.9133	28.72 0.8568	32.47 0.9201	30.65 0.8991	28.84 0.8062	30.35 0.8844
	MDAL [19]	30.15 0.8927	30.06 0.8767	29.03 0.8716	31.28 0.9431	31.43 0.8860	32.02 0.9184	29.10 0.8765	32.71 0.9252	30.99 0.9074	29.10 0.8124	30.59 0.8910
	PAIHT [44]	29.82 0.8896	30.08 0.8830	28.96 0.8708	31.10 0.9404	31.13 0.8846	31.82 0.9164	29.72 0.8867	32.83 0.9247	31.10 0.9088	28.82 0.8002	30.54 0.8905
	ℓ_2-r-ℓ_0 [35]	30.40 0.8928	30.53 0.8863	29.29 0.8764	31.44 0.9394	31.37 0.8853	32.50 0.9216	30.34 0.9025	33.25 0.9251	31.57 0.9107	29.30 0.8154	31.00 0.8956
	SDSR [27]	31.51 0.9033	31.67 0.9058	30.28 0.8945	32.97 0.9554	32.39 0.8975	33.61 0.9339	32.36 0.9328	34.21 0.9328	32.61 0.9248	29.99 0.8389	32.16 0.9120
	IDD-BM3D [16]	31.08 0.8916	30.96 0.8911	29.65 0.8810	32.47 0.9452	31.98 0.8893	33.27 0.9241	32.77 0.9296	33.90 0.9233	31.95 0.9129	29.51 0.8223	31.75 0.9010
	PAAHT	**31.73** **0.9099**	**31.77** **0.9100**	**30.35** **0.8980**	**33.34** **0.9562**	**32.48** **0.8989**	**33.98** **0.9371**	**33.43** **0.9402**	**34.71** **0.9381**	**32.92** **0.9290**	**29.99** **0.8398**	**32.47** **0.9157**
2	Split Bregman [6]	28.47 0.8654	28.80 0.8421	28.08 0.8398	30.62 0.9269	29.71 0.8694	30.98 0.8952	27.35 0.8142	31.40 0.9090	29.59 0.8749	28.10 0.7739	29.31 0.8611
	MDAL [19]	29.10 0.8742	29.06 0.8530	28.04 0.8407	30.19 0.9295	30.67 0.8736	31.04 0.9003	27.82 0.8395	31.64 0.9142	29.99 0.8820	28.31 0.7823	29.59 0.8689
	PAIHT [44]	28.83 0.8723	28.95 0.8536	27.93 0.8421	29.89 0.9245	29.97 0.8676	30.60 0.8971	28.27 0.8439	31.67 0.9116	29.92 0.8877	28.10 0.7737	29.41 0.8674
	ℓ_2-r-ℓ_0 [35]	29.27 0.8739	29.42 0.8571	28.26 0.8455	30.47 0.9296	30.54 0.8723	31.51 0.9067	29.01 0.8659	32.33 0.9154	30.47 0.8874	28.47 0.7796	29.98 0.8733
	SDSR [27]	30.42 0.8879	30.26 0.8810	28.96 0.8686	31.67 0.9441	31.49 0.8842	32.35 0.9176	30.63 0.9108	32.69 0.9206	31.29 0.9059	29.10 0.8085	30.89 0.8929
	IDD-BM3D [16]	30.01 0.8760	29.79 0.8664	28.56 0.8556	31.23 0.9329	31.08 0.8745	32.13 0.9083	31.31 0.9087	32.47 0.9113	30.77 0.8935	28.77 0.7947	30.61 0.8822
	PAAHT	**30.70** **0.8954**	**30.56** **0.8875**	**29.25** **0.8754**	**32.11** **0.9475**	**31.60** **0.8859**	**32.80** **0.9233**	**31.95** **0.9220**	**33.30** **0.9279**	**31.74** **0.9122**	**29.24** **0.8149**	**31.33** **0.8992**
3	Split Bregman [6]	26.74 0.8335	26.95 0.7854	25.90 0.7556	27.47 0.8752	28.57 0.8274	28.59 0.8438	26.35 0.7678	27.39 0.8708	27.09 0.7934	27.47 0.7357	27.25 0.8089
	MDAL [19]	27.64 0.8545	27.56 0.8177	26.15 0.7690	27.95 0.8921	29.17 0.8395	28.93 0.8589	26.59 0.7842	28.64 0.8868	27.81 0.8232	27.72 0.7496	27.82 0.8276
	PAIHT [44]	26.89 0.8423	26.91 0.7940	25.88 0.7650	26.94 0.8737	28.64 0.8294	28.46 0.8456	26.73 0.7853	28.67 0.8828	27.49 0.8154	27.40 0.7280	27.40 0.8162
	ℓ_2-r-ℓ_0 [35]	27.37 0.8443	27.50 0.8076	26.11 0.7682	27.56 0.8851	29.03 0.8374	29.01 0.8587	26.98 0.7969	29.14 0.8882	27.78 0.8210	27.78 0.7442	27.83 0.8252
	SDSR [27]	29.04 0.8726	28.54 0.8402	**26.98** **0.7998**	**29.59** **0.9138**	**30.02** **0.8544**	30.05 0.8752	28.10 0.8291	30.24 0.8982	28.84 0.8492	**28.30** 0.7712	28.97 0.8502
	IDD-BM3D [16]	28.54 0.8586	28.06 0.8219	26.55 0.7799	29.04 0.9034	29.62 0.8427	29.71 0.8658	27.99 0.8227	29.98 0.8914	28.35 0.8321	27.92 0.7526	28.58 0.8371
	PAAHT	**29.05** **0.8757**	**28.57** **0.8430**	26.95 0.7960	29.51 0.9136	29.99 0.8536	**30.06** **0.8767**	**28.22** **0.8335**	**30.31** **0.9010**	**28.88** **0.8516**	28.28 **0.7730**	**28.98** **0.8518**
4	Split Bregman [6]	26.09 0.8152	26.37 0.7585	25.37 0.7296	26.86 0.8589	28.01 0.8121	28.09 0.8278	25.69 0.7395	26.83 0.8601	26.51 0.7692	26.92 0.7048	26.67 0.7876
	MDAL [19]	27.01 0.8360	26.76 0.7826	25.51 0.7436	27.10 0.8738	28.51 0.8254	28.31 0.8429	25.86 0.7559	27.87 0.8751	27.01 0.7951	27.18 0.7215	27.11 0.8052
	PAIHT [44]	26.09 0.8189	26.19 0.7581	25.20 0.7332	26.11 0.8530	27.98 0.8120	27.91 0.8289	26.01 0.7598	27.86 0.8708	26.88 0.7912	26.88 0.6996	26.71 0.7926
	ℓ_2-r-ℓ_0 [35]	26.53 0.8206	26.70 0.7710	25.44 0.7359	26.64 0.8655	28.34 0.8211	28.29 0.8399	26.31 0.7684	28.20 0.8769	26.99 0.7906	27.16 0.7080	27.06 0.7998
	SDSR [27]	28.13 0.8533	27.70 0.8124	26.32 0.7740	28.66 **0.8987**	29.31 **0.8398**	29.31 0.8587	27.20 0.8012	29.47 0.8875	28.00 0.8239	27.70 0.7428	28.18 0.8292
	IDD-BM3D [16]	27.69 0.8393	27.27 0.7935	25.94 0.7540	28.24 0.8875	28.97 0.8263	29.03 0.8473	27.25 0.7949	29.20 0.8784	27.60 0.8066	27.39 0.7262	27.86 0.8154
	PAAHT	**28.14** **0.8577**	**27.72** **0.8150**	**26.32** **0.7745**	**28.71** 0.9006	**29.38** 0.8393	**29.41** **0.8616**	**27.49** **0.8078**	**29.57** **0.8915**	**28.10** **0.8275**	**27.76** **0.7465**	**28.26** **0.8322**

Table 8: Comparison of PSNR (dB) and SSIM results by different deblurring methods. Bold values denote the highest PSNR and SSIM values.

Scenario	Method	*C.man*	*Boat*	*Man*	*Monarch*	*Peppers*	*Lena*	*Barbara*	*Parrots*	*Starfish*	*Goldhill*	**Avg.**
5	Split Bregman [6]	27.00 0.8601	28.01 0.8396	27.46 0.8376	30.35 0.9335	29.14 0.8781	30.71 0.9025	25.12 0.7640	30.31 0.9163	29.08 0.8768	27.66 0.7699	28.48 0.8578
	MDAL [19]	27.34 0.8687	28.09 0.8487	27.14 0.8349	29.39 0.9326	29.18 0.8775	30.21 0.9038	25.61 0.7752	29.96 0.9180	29.31 0.8923	27.20 0.7690	28.34 0.8621
	PAIHT [44]	27.03 0.8598	27.83 0.8394	27.16 0.8342	28.96 0.9268	28.80 0.8733	29.99 0.8992	25.56 0.7837	30.43 0.9170	29.54 0.8900	27.57 0.7610	28.29 0.8584
	ℓ_2-r-ℓ_0 [35]	27.34 0.8628	28.20 0.8436	27.40 0.8345	29.65 0.9303	29.32 0.8765	30.89 0.9071	25.72 0.7841	31.00 0.9185	30.01 0.8875	27.90 0.7681	28.74 0.8613
	SDSR [27]	28.36 0.8781	**29.08** 0.8641	28.12 0.8543	**31.34** 0.9442	**30.24** **0.8862**	31.59 0.9149	27.22 0.8245	31.85 0.9244	30.83 0.9019	28.31 0.7876	29.68 0.8780
	IDD-BM3D [16]	28.10 0.8687	28.73 0.8528	27.83 0.8441	30.90 0.9380	29.97 0.8799	31.41 0.9089	27.08 0.8205	31.55 0.9179	30.36 0.8924	28.18 0.7787	29.41 0.8702
	PAAHT	**28.45** **0.8804**	29.06 **0.8659**	**28.14** **0.8554**	31.27 **0.9453**	30.17 0.8859	**31.66** **0.9166**	**27.26** **0.8276**	**31.88** **0.9274**	**31.16** **0.9083**	**28.33** **0.7886**	**29.74** **0.8801**
6	Split Bregman [6]	26.73 0.8496	27.59 0.8215	27.08 0.8206	29.84 0.9245	28.85 0.8694	30.31 0.8914	24.69 0.7480	29.94 0.9087	28.69 0.8631	27.44 0.7533	28.12 0.8450
	MDAL [19]	27.08 0.8574	27.68 0.8306	26.83 0.8189	29.05 0.9249	29.33 0.8708	29.94 0.8943	24.57 0.7467	29.56 0.9099	28.68 0.8744	27.07 0.7563	27.98 0.8484
	PAIHT [44]	26.64 0.8484	27.36 0.8209	26.79 0.8182	28.41 0.9160	28.38 0.8629	29.59 0.8891	24.78 0.7525	29.96 0.9093	28.99 0.8770	27.24 0.7484	27.81 0.8443
	ℓ_2-r-ℓ_0 [35]	26.91 0.8506	27.66 0.8234	26.93 0.8153	29.03 0.9210	28.93 0.8676	30.38 0.8962	24.79 0.7536	30.48 0.9114	29.36 0.8713	27.60 0.7506	28.21 0.8461
	SDSR [27]	27.85 0.8646	**28.49** 0.8450	27.63 0.8349	30.57 0.9361	**29.85** 0.8780	30.94 0.9025	**26.32** 0.7926	31.18 0.9157	30.12 0.8857	27.88 0.7693	29.08 0.8624
	IDD-BM3D [16]	27.63 0.8609	28.05 0.8311	27.28 0.8201	30.24 0.9338	29.55 0.8725	30.84 0.9019	26.02 0.7853	30.99 0.9161	29.69 0.8757	27.71 0.7549	28.80 0.8552
	PAAHT	**27.93** **0.8703**	28.43 **0.8481**	**27.64** **0.8365**	**30.71** **0.9383**	29.80 **0.8785**	**31.13** **0.9075**	26.20 **0.7942**	**31.34** **0.9206**	**30.39** **0.8927**	**27.92** **0.7708**	**29.15** **0.8658**
7	Split Bregman [6]	28.76 0.8672	28.53 0.8383	27.86 0.8231	29.37 0.9035	29.82 0.8614	30.37 0.8837	27.25 0.8226	31.00 0.9065	28.96 0.8614	28.21 0.7793	29.01 0.8547
	MDAL [19]	30.31 0.8886	29.58 0.8672	28.39 0.8425	30.70 0.9286	31.29 0.8744	30.98 0.9053	27.89 0.8521	32.34 0.9202	30.18 0.8895	28.86 0.7994	30.05 0.8768
	PAIHT [44]	28.57 0.8611	28.54 0.8389	27.36 0.8112	29.71 0.9150	30.59 0.8664	30.91 0.9016	28.13 0.8555	31.96 0.9130	30.09 0.8912	28.40 0.7900	29.43 0.8644
	SDSR [27]	31.71 0.9059	31.17 0.8964	**29.66** **0.8779**	**32.66** **0.9467**	**32.41** **0.8904**	32.91 **0.9276**	31.42 0.9234	33.50 0.9278	31.96 0.9153	**29.88** **0.8336**	31.73 0.9045
	IDD-BM3D [16]	30.97 0.8841	30.35 0.8776	28.85 0.8548	31.65 0.9287	31.60 0.8746	32.24 0.9062	31.76 0.9170	32.47 0.9081	31.01 0.8950	29.26 0.8114	31.02 0.8858
	PAAHT	**31.88** **0.9105**	**31.19** **0.8971**	29.64 0.8778	32.54 0.9427	32.31 0.8879	**33.03** 0.9244	**32.35** **0.9292**	**33.54** **0.9282**	**32.02** **0.9155**	29.86 0.8334	**31.84** **0.9047**
8	Split Bregman [6]	27.57 0.8447	27.57 0.8042	26.86 0.7899	28.16 0.8857	28.77 0.8420	29.31 0.8610	26.21 0.7853	29.87 0.8957	27.99 0.8333	27.37 0.7425	27.97 0.8284
	MDAL [19]	29.12 0.8661	28.49 0.8368	27.27 0.8038	29.39 0.9105	30.21 0.8548	29.92 0.8836	26.67 0.8092	31.10 0.9069	28.90 0.8585	28.05 0.7643	28.91 0.8495
	PAIHT [44]	27.67 0.8488	27.60 0.8129	26.53 0.7887	28.19 0.8946	29.16 0.8423	29.50 0.8779	26.85 0.8134	30.58 0.8971	28.69 0.8624	27.44 0.7530	28.22 0.8391
	SDSR [27]	30.37 0.8879	**29.81** **0.8707**	28.27 0.8466	**31.19** **0.9322**	**31.24** **0.8736**	31.51 **0.9084**	29.62 0.8961	**32.07** **0.9151**	30.48 0.8906	28.85 0.7992	30.34 0.8820
	IDD-BM3D [16]	29.65 0.8675	29.06 0.8510	27.58 0.8215	30.30 0.9144	30.47 0.8564	31.01 0.8880	30.31 0.8940	30.93 0.8974	29.67 0.8698	28.34 0.7784	29.73 0.8638
	PAAHT	**30.59** **0.8959**	29.84 0.8722	**28.34** **0.8475**	31.20 0.9326	31.19 0.8719	**31.78** 0.9101	**30.86** **0.9090**	32.12 0.9192	**30.62** **0.8935**	**28.89** **0.8016**	**30.54** **0.8854**

Table 9: Comparison of PSNR (dB) and SSIM results by different deblurring methods. Bold values denote the highest PSNR and SSIM values.

Image	Method	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
	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 [33]	7.41	5.17	8.54	2.57	3.36	1.30
	L0-AbS[34]	7.70	5.55	9.10	2.93	3.49	1.77
	IDD-BM3D [16]	8.85	7.12	10.45	3.98	4.31	4.89
	NCSR [18]	8.78	6.69	10.33	3.78	4.60	4.50
Cameraman	ADMM-BM3D [41]	8.18	6.13	9.58	3.26	3.93	4.88
(256 x 256)	ADMM-GMM [38]	8.34	6.39	9.73	3.49	4.18	4.90
	MLP [36]	-	-	-	-	4.48	-
	IRCNN [53]	9.08	7.33	10.30	4.29	**4.70**	-
	IDBP-CNN [40]	9.08	7.28	10.55	4.25	-	-
	LNIR [23]	-	-	-	-	**4.70**	-
	GSR [55]	8.39	6.39	10.08	3.33	3.94	4.76
	PAAHT	**9.50**	**7.66**	**15.37**	**4.32**	4.67	**6.44**
	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 [33]	7.98	6.57	10.39	4.12	4.54	2.44
	L0-AbS [34]	8.40	7.12	11.06	4.55	4.80	2.15
	IDD-BM3D [16]	9.95	8.55	12.89	5.79	5.74	7.13
	NCSR [18]	9.96	8.48	13.12	5.81	5.67	6.94
House	ADMM-BM3D [41]	9.64	8.02	12.95	5.23	5.06	7.37
(256 x 256)	ADMM-GMM [38]	9.84	8.40	12.87	5.57	5.55	6.65
	MLP [36]	-	-	-	-	5.62	-
	IRCNN [53]	9.69	8.63	11.58	6.05	5.98	-
	IDBP-CNN [40]	9.93	8.45	11.91	5.85	-	-
	LNIR [23]	-	-	-	-	**6.26**	-
	GSR [55]	10.02	8.56	13.44	6.00	5.95	7.18
	PAAHT	**10.50**	**9.04**	**16.57**	**6.31**	6.00	**8.16**
	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 [33]	6.36	4.98	7.47	3.52	3.61	2.79
	L0-AbS [34]	6.66	5.71	7.79	4.09	4.22	1.93
	IDD-BM3D [16]	7.97	6.61	8.91	4.97	4.85	6.34
Lena	NCSR [18]	8.03	6.54	9.25	4.93	4.86	6.19
	ADMM-BM3D [41]	8.00	6.56	9.00	4.88	4.67	6.42
(512 x 512)	ADMM-GMM [38]	8.01	6.53	8.95	4.93	4.81	6.09
	IRCNN [53]	8.06	6.79	8.88	5.13	-	-
	IDBP-CNN [40]	8.24	6.64	9.05	5.05	-	-
	GSR [55]	8.24	6.76	9.43	5.17	4.96	6.57
	PAAHT	**8.57**	**7.16**	**12.26**	**5.55**	**5.12**	**7.35**
	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 [33]	3.10	1.33	3.49	0.41	0.75	0.59
	L0-AbS [34]	3.51	1.53	3.98	0.73	0.81	1.17
	IDD-BM3D [16]	7.64	3.96	6.05	1.88	1.16	5.45
Barbara	NCSR [18]	7.76	3.64	5.92	2.06	1.43	5.50
	ADMM-BM3D [41]	7.32	2.99	6.05	1.55	1.40	5.76
(512 x 512)	ADMM-GMM [38]	5.91	2.19	5.37	1.42	1.24	5.14
	IRCNN [53]	7.54	4.68	5.92	1.82	-	-
	IDBP-CNN [40]	6.89	4.41	6.07	**2.40**	-	-
	GSR [55]	**8.98**	**4.80**	7.15	2.19	**1.58**	6.20
	PAAHT	8.04	4.24	**9.08**	2.00	1.21	**6.70**

Table 10: ISNR (dB) comparison for image deblurring in the second set. The highest PSNR values are highlighted in bold.

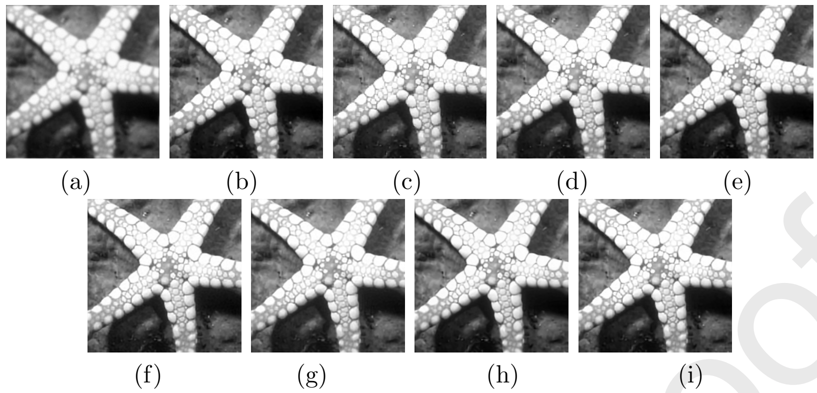


Figure 9: Visual quality comparison of image deburring. (a) Blurred and noisy image (*Starfish* (256 × 256), Gaussian kernel: fspecial (Gaussian, 25, 1.6), $\sigma = \sqrt{2}$)). (b) recovery image by Split Bregman [6] (PSNR=29.08; SSIM=0.8768), (c) MDAL [19](PSNR=29.31; SSIM=0.8923), (d) PAIHT [44] (PSNR=29.54; SSIM=0.8900), (e) ℓ_2-r-ℓ_0 [35] (PSNR=30.01; SSIM=0.8875), (f) SDSR [27] (PSNR=30.83; S-SIM=0.9019), (g) IDD-BM3D [16] (PSNR=30.36; SSIM=0.8924), (h) SNSS [46] (P-SNR=31.06; SSIM=0.9034), (i) PAAHT (PSNR=31.16; SSIM=0.9083).

noises $\sigma_w = \sqrt{2}, 2$ are exploited for the simulation. In the second set, six typical deblurring experiments (see Table 3) concerning four standard grayscale images, which are often used as the benchmark for image deblurring, are also provided for more comprehensive comparisons.

The proposed PAAHT algorithm for the ℓ_2-relaxed framelet-based truncated ℓ_0 regularization model compares with several closely-related and representative transform domain sparse representation image deblurring methods. The competing algorithms include **TVMM** [33] for the total variation regularization model, the **Split Bregman** [6] method for framelet-based ℓ_1 regularization model, **MDAL** [19] for the ℓ_0 regularization model, **L0-AbS** [34], $\boldsymbol{\ell_2}$**-r-**$\boldsymbol{\ell_0}$ [35] and **PAIHT** [44] for the ℓ_2-relaxed framelet-based ℓ_0 regularization model, **S-DSR** [27] for the framelet-based truncated ℓ_0 regularization model. To better illustrating the differences of these framelet-based methods, a brief comparison of them are summarized in Table 4. Besides these popular local-based image restoration methods, the proposed PAAHT is also compared with several other leading deblurring methods in the literature, including **IDD-BM3D** [16],

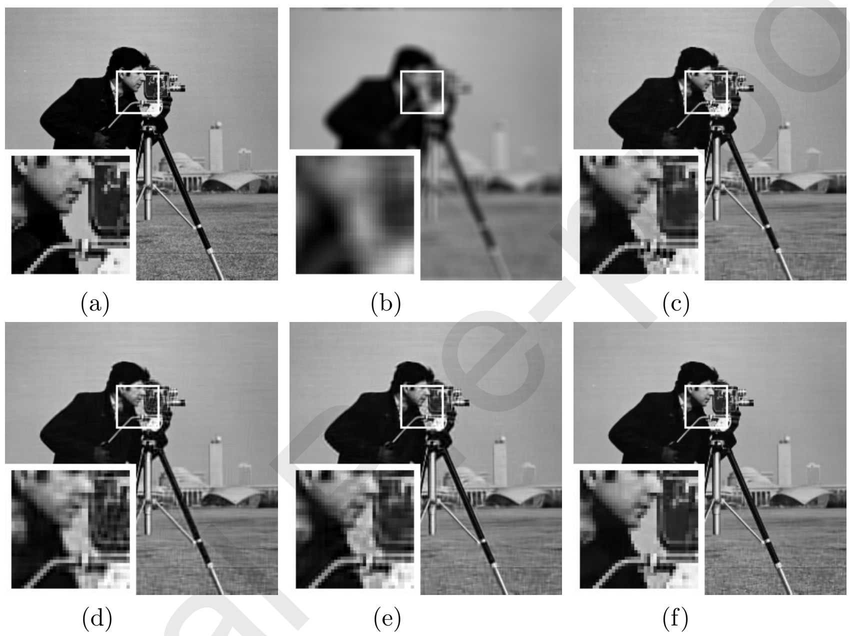


Figure 10: Visual quality comparison of image deburring. (a) Original clean image *Cameraman* (256×256), (b) Blurred and noisy image (scenario: 3), (c) recovery image by IDD-BM3D [16] (ISNR=10.45), (d) ADMM-BM3D [41] (ISNR=9.58), (e) GSR [55] (ISNR=10.08), (f) PAAHT (ISNR=15.37). Best viewed on high-resolution display.

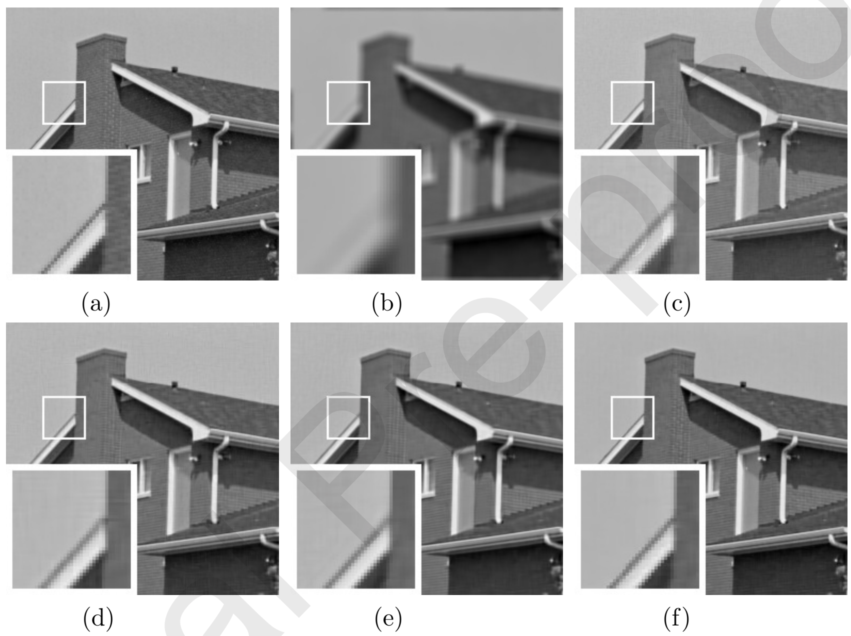
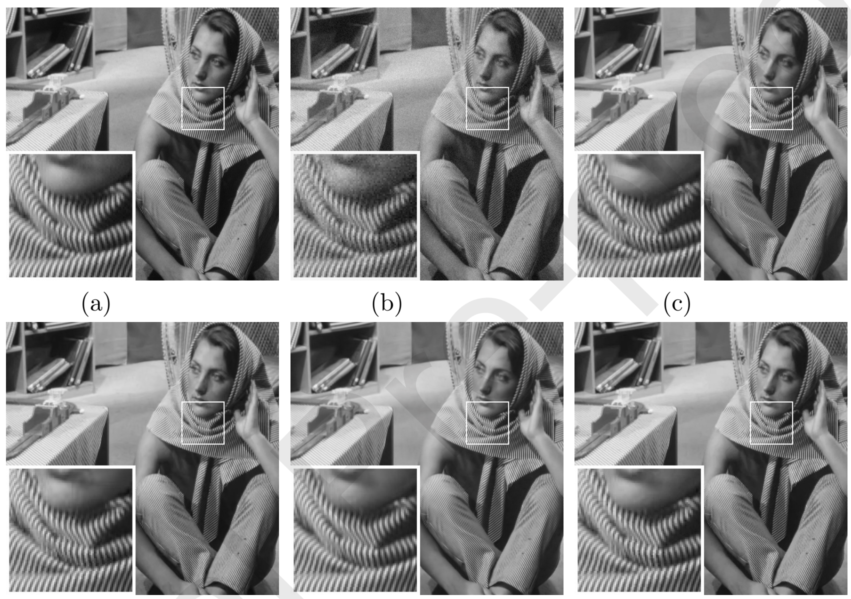


Figure 11: Visual quality comparison of image deburring. (a) Original clean image *House* (256×256), (b) Blurred and noisy image (scenario: 3), (c) recovery image by IDD-BM3D [16] (ISNR=12.89), (d) ADMM-BM3D [41] (ISNR=12.95), (e) GSR [55] (ISNR=13.44), (f) PAAHT (ISNR=16.57). Best viewed on high-resolution display.



(a) (b) (c) (d) (e) (f)

Figure 12: Visual quality comparison of image deburring. (a) Original clean image *Barbara* (512×512), (b) Blurred and noisy image (scenario: 6), (c) recovery image by IDD-BM3D [16] (ISNR=5.45), (d) ADMM-BM3D [41] (ISNR=5.76), (e) GSR [55] (ISNR=6.20), (f) PAAHT (ISNR=6.70). Best viewed on high-resolution display.

ASDS [17], **NCSR** [18], **JSM** [54], **GSR** [55], **SNSS** [46], machine learning based method **MLP** [36], plug-and-play ADMM framework using the BM3D denoiser (**ADMM-BM3D**) [41], plug-and-play ADMM framework using the GMM denoiser (**ADMM-GMM**) [38], plug-and-play scheme using the CNN denoiser (**IRCNN**) [53] and plug-and-play scheme using the CNN and WNNM denoisers (**LNIR**) [23], iterative denoising and backward projections using the CNN denoiser (**IDBP-CNN**) [40].

The PSNR and SSIM values on ten grayscale test images in the first set of experiments are reported in Table 8 and Table 9. One can observe that SDSR and PAAHT produce very similar results, attributing to the usage of sparsity and support priors simultaneously, and obtain noticeable improvements over other competing methods. Not surprisingly, on average, PAAHT moderately outperforms SDSR to a certain extent, demonstrating the benefits of exploiting ℓ_2-relaxed formulation in the regularization model. In summary, it can be concluded that the sparsity prior, support prior, and ℓ_2-relaxed formulation are complementary to improve the restoration quality, and turning off either one of them often leads to a worse result. The visual comparisons of these deblurring methods are shown in Figures 8 and 9, from which one can observe that the PAAHT algorithm produces cleaner and sharper image edges and textures than other competing methods.

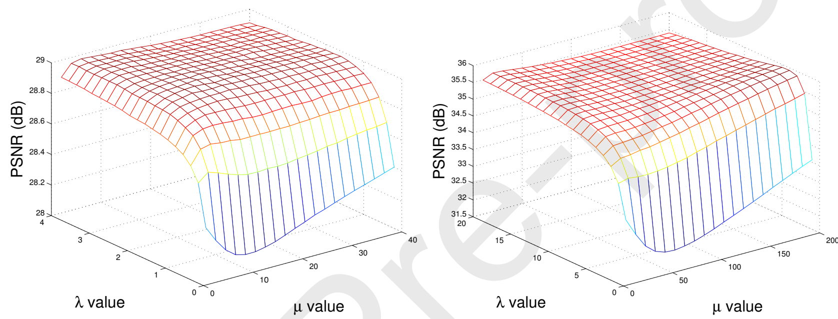
Table 10 lists the comparison of ISNR results for six benchmark deblurring experiments in the second set. It is clearly observed that PAAHT achieves the highest ISNR results in most cases, as labeled in bold. Particularly, for several scenarios, the performance gain over the other leading deblurring methods is significant (for scenario 3, up to 4.9 dB and 3.2 dB against IDD-BM3D on test images *Cameraman* and *Lena*, respectively). Additionally, Table 11 reports the average PSNR and SSIM values of different methods for scenarios 3 and 6 on Set12 dataset, it can be clearly observed that PAAHT produces the best results and outperforms other competing algorithms by a large margin. The visual comparisons of the deblurring methods for the second set are presented in Figures 10-12. From the enlarged regions, one can visually observe that

Scenario	IDD-BM3D	ADMM-BM3D	ASDS	JSM	GSR	PAAHT	
3	32.33	32.20	31.59	31.54	32.43	36.60	
	0.9003	0.8973	0.9035	0.8943	0.9082	0.9522	
6	34.92	34.96	34.46	34.71	35.03	36.31	
	0.9276	0.9312	0.9180	0.9241	0.9287	0.9425	

Table 11: Average PSNR and SSIM values of different methods for scenarios 3 and 6 on Set12 dataset.

the proposed PAAHT algorithm generates a much clearer image than other competing approaches on both the homogenous areas and salient edges.

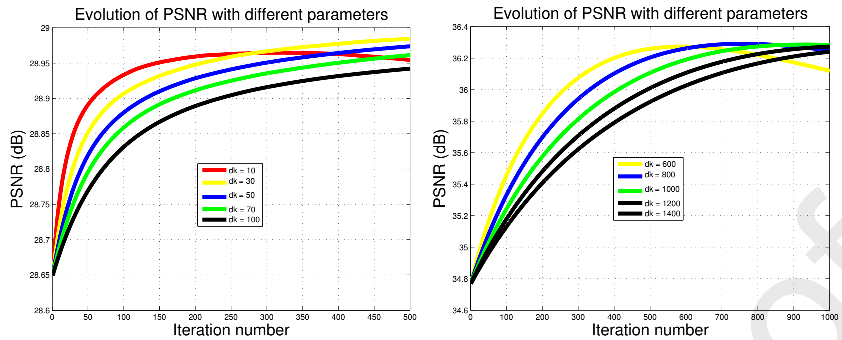
4.6. Effect of the parameters



(a) (b)

Figure 13: Evolution of PSNR values with regularization parameters λ and μ in the cases of deblurring in the second set. (a) Scenario 2 ($d_k = 20$), (b) Scenario 6 ($d_k = 100$). Test image: *Cameraman*. It can be observed that the final performance of our algorithm is not very sensitive to the regularization parameters λ and μ within an appropriate range.

This subsection will give a detailed description about how sensitive the performance of PAAHT algorithm is affected by the regularization parameters λ and μ, and penalty parameter d_k. The evolution of PSNR values versus the λ and μ choices in the cases of deblurring (two scenarios of deblurring experiments in the second set are conducted with various blur kernels and noise variances, i.e., scenario 2 and scenario 6 in Table 3) is provided in Figure 13. It is obvious to see that regularization parameters λ and μ have great relationships with the noise level σ_w, i.e., a larger σ_w corresponds to larger λ and μ optimums. It can also be concluded that the small changes of λ and μ do not dramatically



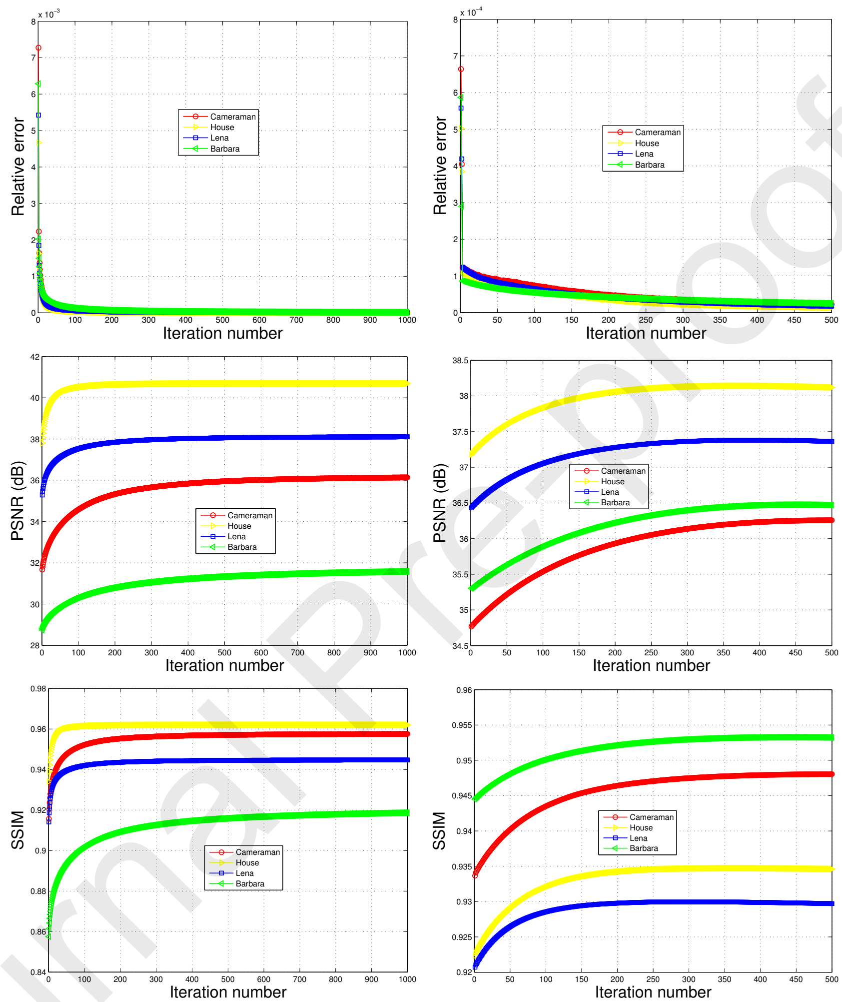
(a) (b)

Figure 14: Evolution of PSNR values with penalty parameters d_k in the cases of deblurring in the second set. (a) Scenario 2 ($\lambda = 2.0$, $\mu = 10\lambda = 20$), (b) Scenario 6 ($\lambda = 10.0$, $\mu = 10\lambda = 100$). Test image: *Cameraman*. It can be observed that appropriate choice of d_k can achieve a more rapid convergence.

impact the final recovery performance. The curves of PSNR values versus the d_k choices in the cases of deblurring are displayed in Figure 14. one can also clearly observe that a larger σ_w corresponds to larger d_k, the final performance of PAAHT algorithm is not very sensitive to the panelty parameter d_k within an appropriate range, and appropriate choice of d_k leads to a more rapid convergence. Note that the other cases have similar observations, and it can conclude that the PAAHT algorithm is quite robust for these involved parameters.

4.7. Convergence property

This subsection provides the empirical evidence to illustrate the convergence behavior of PAAHT algorithm. Take the cases of image deblurring as examples. Figure 15 plots the relative errors between consecutive iterations, the PSNR and SSIM values versus the iteration numbers. It can be obviously observed that with the growth of iteration number, all the relative error curves decrease monotonically, the PSNR and SSIM curves increase monotonically, and ultimately become flat and stable, exhibiting good convergence behavior of the proposed PAAHT algorithm.



(a) (b)

Figure 15: The evolution of relative error between consecutive iterations, PSNR and SSIM values of the proposed method in the cases of deblurring in the second set. (a) scenario: 3, (b) scenario: 6.

4.8. Computational complexity and running time

It is noted that the computational cost of the whole algorithm consists of two parts. For example, for the image deblurring, assume that the number of image pixels is N, and the image boundary condition is periodic. Thus the involved matrix inversion can be efficiently computed using the fast Fourier transforms (FFTs). The complexity of the off-the-shelf image restoration method to produce an initial reference image is denoted as $\mathbf{M}$. Then, the total complexity of our algorithm is $\mathbf{M} + \mathcal{O}(N\log N)$. Empirically, for a 256×256 grayscale image, the iterative loop of our algorithm takes around $2 \sim 3$ seconds for image denoising and 10 seconds for image deblurring, respectively, on an AMD Ryzen 7 3700X 8-Core Processor (3.60GHz) and 32 GB memory PC under Matlab R2018a environment.

5. Conclusions

This paper establishes a novel tight frame-based ℓ_2-relaxed truncated ℓ_0 analysis-sparsity model, which incorporates the sparsity prior, support prior, and ℓ_2-relaxed formulation simultaneously. Besides, an efficient algorithm called *proximal alternating adaptive hard-thresholding* (PAAHT) is proposed to solve the corresponding nonconvex nonsmooth minimization problem. It is proved that the sequence generated by the proposed algorithm sublinearly converges. Extensive experimental experiments on two fundamental image restoration applications: denoising and deblurring recovery show that the proposed PAAHT algorithm exhibits good convergence behavior, offers a noticeable boost compared with the conventional tight frame-based methods, and often achieves significant performance improvements over many current state-of-the-art schemes. Future work includes the extensions of PAAHT to more applications, such as class-specific image restoration problems, image deblurring with impulsive and Poisson noises.

6. Acknowledgement

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 (12001005, 11701079, 61806134, 62076170), the start-up funding of Anhui University (Y040418173), the Natural Science Research Project of Anhui Universities (KJ2019A0005, KJ2019A0032), the Project of Natural Science Foundation of Anhui Province (2008085QF286), the Fundamental Research Funds for the Central Universities (2412020FZ023); Jilin Provincial Department of Education (JJKH20190293KJ); Sichuan University Innovation Spark Project (2019SCUH0007).

References

[1] S. Anwar, F. Porikli and C. P. Huynh. Category-specific object image denoising. *IEEE Trans. Image Process*, 2017, 26(11), 5506-5518.

[2] A. Buades, B. Coll, J. M. Morel. A non-local algorithm for image denoising. *in Proc. IEEE CVPR*, 2005, 2, 60-65.

[3] 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,* 2010, 3(1), 1-122.

[4] Y. Chen, T. Pock. Trainable Nonlinear Reaction Diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2017, 39(6), 1256 - 1272.

[5] E. J. Candes, M. B. Wakin, An introduction to compressive sampling. *IEEE Signal Processing Magzine,* 2008, 25, 21-30.

[6] J. Cai, S. Osher, Z. Shen. Split Bregman methods and frame based image restoration. *Multiscale. Model. Simul,* 2009, 8(2), 337-369.

[7] J. Cai, H. Ji, Z. Shen, G. Ye, Data-driven tight frame construction and image denoising, *Appl. Comput. Harmon. Anal,* 2014, 37(1), 89–105.

[8] J. Cai, B. Dong, Z. Shen, Image restorations: A wavelet frame based model for piecewise smooth functions and beyond, *Appl. Comput. Harmon. Anal,* 2016, 41(1), 94–138.

[9] A. Chambolle, T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging and Vision,* 2010, 40(1), 120-145.

[10] F. Chen, L. Zhang, H. Yu. External patch prior guided internal clustering for image denoising. *in Proc. IEEE ICCV*, 2015, 603-611.

[11] C. Dong, C. C. Loy, K. He, et al. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2016, 38(2), 295-307.

[12] D. Donoho, Compressed sensing. *IEEE Transactions on Information Theory,* 2006, 52, 1289-1306.

[13] W. Dong, G. Shi, X. Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Trans. Image Process*, 2013, 22(2), 700-711.

[14] W. Dong, G. Shi, Y. Ma, X. Li. Image restoration via simultaneous sparse coding: where structured sparsity meets Gaussian scale mixture. *Int. J. Comput. Vis*, 2015, 114(2), 217-232.

[15] K. Dabov, A. Foi, V. Katkovnik, et al. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. Image Process,* 2007, 16(8), 2080-2095.

[16] A. Danielyan, V. Katkovnik, K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Trans. Image Process,* 2012, 21(4), 1715-1728.

[17] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Trans. Image Process,* 2011, 20(7), 1838-1857.

[18] W. Dong, L. Zhang, G. Shi. Nonlocally centralized sparse representation for image restoration. *IEEE Trans. Image Process,* 2013, 22(4), 1620-1630.

[19] B. Dong, Y. Zhang, An efficient algorithm for ℓ_0 minimization in wavelet frame based image restoration. *J. Sci. Comput,* 2013, 54(2-3), 350-368.

[20] B. Dong, Q. Jiang, Z. Shen, Image restoration: wavelet frame shrinkage, nonlinear evolution PDEs, and beyond, *Multiscale. Model. Simul,* 2017, 15(1), 606-660.

[21] G. H. Golub, C. F. Van Loan, Matrix Computations, *JHU Press*, 2012.

[22] S. Gu, L. Zhang, W. Zuo, et al. Weighted nuclear norm minimization with application to image denoising. *in Proc. IEEE CVPR*, 2014, 2862-2869.

[23] S. Gu, R. Timofte, and L. V. Gool. Integrating local and non-local denoiser priors for image restoration. *in Proc. IEEE ICPR*, 2018, 2923-2928.

[24] G. Hou, Z. Pan, G. Wang. An efficient nonlocal variational method with application to underwater image restoration. *Neurocomputing*, 2019, 369, 106-121.

[25] L. Huang, Y. Xia. Joint blur kernel estimation and CNN for blind image restoration. *Neurocomputing*, 2020, 396, 324-345.

[26] L. He, Y. Wang. Iterative support detection-based split Bregman method for wavelet frame-based image inpainting , *IEEE Trans. Image Process,* 2014, 23(12), 5470-5485.

[27] L. He, Y. Wang, Z. Xiang. Support driven wavelet frame-based image deblurring, *Inform. Sciences,* 2019, 479, 250-269.

[28] K. Jon, Y. Sun, Q. Li, et al. Image restoation using overlapping group sparsity on hyper-Laplacian prior of image gradient. *Neurocomputing*, 2021, 420, 57-69.

[29] S. Lu, Y. Zhang. Sparse approximation via penalty decomposition methods. *SIAM Journal on Optimization,* 2013, 23(4), 2448-2478.

[30] J. Liu, T. Huang, I. W. Selesnick, et al. Image restoration using total variation with overlapping group sparsity. *Inform. Sciences,* 2015, 295, 232-246.

[31] C. Lanaras, E. Baltsavias, K. Schindler. Hyperspectral super-resolution by coupled spectral unmixing. *in Proc. IEEE ICCV,* 2015, 3586-3594.

[32] J. Liu, M. Yan, T. Zeng. Surface-aware blind image deblurring. *IEEE Trans. Pattern Anal. Mach. Intell*, 2019, Doi: 10.1109/TPAMI.2019.2941472.

[33] J. Oliveira, J. M. Bioucas-Dias, M. Figueiredo. Adaptive total variation image deblurring: A majorization-minimization approach, *Signal Process,* 2009, 89(9), 1683-1693.

[34] J. Portilla. Image restoration through ℓ_0 analysis-based sparse optimization in tight frames. *in Proc. IEEE ICIP,* 2009, 3909-3912.

[35] J. Portilla, A. Tristán-Vegaz, I. W. Selesnick. Efficient and robust image restoration using multiple-feature L2-relaxed sparse analysis priors. *IEEE Trans. Image Process,* 2015, 24(12), 5046-5059.

[36] C. J. Schuler, H. C. Burger, S. Harmeling, et.al. A machine learning approach for non-blind image deconvolution. *in Proc. IEEE CVPR*, 2013, 1067-1074.

[37] L. Tang, Y. Ren, Z. Fang, C. He. A generalized hybrid nonconvex variational regularization model for staircase reduction in image restoration. *Neurocomputing*, 2019, 359, 15-31.

[38] 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, 3518-3522.

[39] A. M. Teodoro, J. M. Bioucas-Dias, M. A. T. Figueiredo. A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising. *IEEE Trans. Image Process*, 2019, 28(1), 451-463.

[40] T. Tirer, R. Giryes. Image restoration by iterative denoising and backward projections. *IEEE Trans. Image Process*, 2019, 28(3), 1220-1234.

[41] S. V. Venkatakrishnan, C. A. Bouman, B. Wohlberg. Plug-and-play priors for model based reconstruction. *in Proc. IEEE GCSIP,* 2013, 945-948.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process,* 2004, 13(4), 600-612.

[43] J. Xu, L. Zhang, W. Zuo, D. Zhang, and X. Feng. Patch group based nonlocal self-similarity prior learning for image denoising. *in Proc. IEEE ICCV*, 2015, 244-252.

[44] F. Yang, Y. Shen, Z. Liu. The proximal alternating iterative hard thresholding method for ℓ_0 minimization, with complexity $\mathcal{O}(1/\sqrt{k})$. *J. Comput. Appl. Math,* 2017, 311, 115-129.

[45] Z. Zha, X.Yuan, B. Wen, J. Zhou, C. Zhu. Group sparsity residual constraint with non-local priors for image restoration. *IEEE Trans. Image Process,* 2020, 29, 8960-8975.

[46] Z. Zha, X. Yuan, J. Zhou, Ce Zhu, B. Wen. Image restoration via simultaneous nonlocal self-similarity priors. *IEEE Trans. Image Process,* 2020, 29, 8561-8576.

[47] Z. Zha, X. Yuan, B. Wen, J. Zhang, J. Zhou, Ce Zhu. Image restoration using joint patch-group based sparse representation. *IEEE Trans. Image Process,* 2020, 29, 7735-7750.

[48] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, C. Zhu. A benchmark for sparse coding: When group sparsity meets rank minimization. *IEEE Trans. Image Process,* 2020, 29, 5094-5109.

[49] Z. Zha, X. Yuan, B. Wen, J. Zhou, J. Zhang, C. Zhu. From rank estimation to rank approximation: rank residual constraint for image restoration. *IEEE Trans. Image Process,* 2019, 29, 3254-3269.

[50] Z. Zha, B. Wen, X. Yuan, J. Zhou, Ce Zhu. Reconciliation of group sparsity and low-rank models for image restoration. *IEEE International Conference on Multimedia and Expo (ICME)*, 2020, 1-6.

[51] Z. Zha, X. Yuan, J. Zhou, Ce Zhu, B. Wen. A hybrid structural sparse error model for image deblocking. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, 2493-2497.

[52] D. Zoran, Y. Weiss. From learning models of natural image patches to whole image restoration. *in Proc. IEEE ICCV*, 2011, 479-486.

[53] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep CNN denoiser prior for image restoration. *in Proc. IEEE CVPR*, 2017, 3929-3938.

[54] J. Zhang, D. Zhao, R. Xiong, S. Ma, W. Gao. Image Restoration Using Joint Statistical Modeling in a Space-Transform Domain. *IEEE Transactions on Circuits and Systems for Video Technology,* 2014, 24(6), 915 - 928.

[55] J. Zhang, D. Zhao, W. Gao. Group-based sparse representation for image restoration. *IEEE Trans. Image Process,* 2014, 23(8), 3336-3351.

[56] Z. Zha, X. Zhang, Y. Wu, et al. Non-convex weighted ℓ_p nuclear norm based ADMM framework for image restoration. *Neurocomputing*, 2018, 311, 209-224.

[57] Y. Zhou, J. Dong, Y. Yang. Deep fractal residual network for fast and accurate single image super resolution. *Neurocomputing*, 2020, 398, 389-398.

[58] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising, *IEEE Trans. Image Process,* 2018, 26(7), 3142-3155.

[59] K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a fast and flexible solution for CNN based image denoising, *IEEE Trans. Image Process,* 2018, 27(9), 4608-4622.

[60] X. Zhang, M. Burger, X. Bresson, S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.* 2010, 3(3), 253-276.