Signal Processing 157 (2019) 14–24

Contents lists available at ScienceDirect

Signal Processing

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

Weighted-l_1-method-noise regularization for image deblurring

Chunyu Yang, Weiwei Wang*, Xiangchu Feng, Xin Liu

School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 27 July 2018
Revised 15 October 2018
Accepted 12 November 2018
Available online 13 November 2018

Keywords:
Image deblurring
Method noise regularization
Split Bregman method

ABSTRACT

Various image prior based regularization techniques have been proposed for image deblurring. By utilizing existing image smoothing operators, the method-noise provides a new way to formulate image regularizers. The method noise is defined as the difference of an image and its smoothed version, obtained by an image smoothing operator such as the non-local means(NLM). Therefore, the method noise mainly contains edges, small scaled details and noise (if exists). The l_2-NLM method noise regularization has been successfully used in image denoising. However, the restored image exists over-smoothed edges and noise in smooth areas cannot be perfectly removed. In this work, we propose a weighted-l_1-method-noise regularization model for image deblurring. We analyze the advantages of the proposed model in terms of variational form and its solution. Specifically, the l_1 penalty of the method noise is better than the l_2 penalty in removing noise in smooth areas. The incorporated gradient based weight can better preserve image edges. Experimental results show that the proposed method can obtain better results than other method noise based regularization methods.

1. Introduction

Image deblurring aims to estimate a sharp clean image from a blurred noisy image, which is one of the most basic problems in image processing. In general, image blurring can be described by the following mathematical formulation [1]:

$$\boldsymbol{f} = \boldsymbol{A} * \boldsymbol{u} + \boldsymbol{k} \tag{1}$$

where $\boldsymbol{f}$ and $\boldsymbol{u} \in R^N (N = m \times n)$ denotes the observed image and the original image respectively, $\boldsymbol{A}$ stands for the blurring convolution kernel, $*$ denotes the convolution operation, and $\boldsymbol{k} \in R^N$ is the white Gaussian noise with variance σ^2.

In recent years, many image deblurring methods have been presented based on the following variational framework [2–6]:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|\boldsymbol{A} * \boldsymbol{u} - \boldsymbol{f}\|_2^2 + \lambda J(\boldsymbol{u}) \tag{2}$$

where $\|\cdot\|_2^2$ stands for square of l_2 norm in R^N. The first term is the data fitting term which ensures the blurred version of the restored image close to the observed image. The second term $J(\boldsymbol{u})$ is called regularization term, which imposes smoothness or some kind of structure constraint on the estimated image and can be deduced from the prior of the real image. $\lambda > 0$ is a tuning parameter to balance the data fitting term and the regularization term.

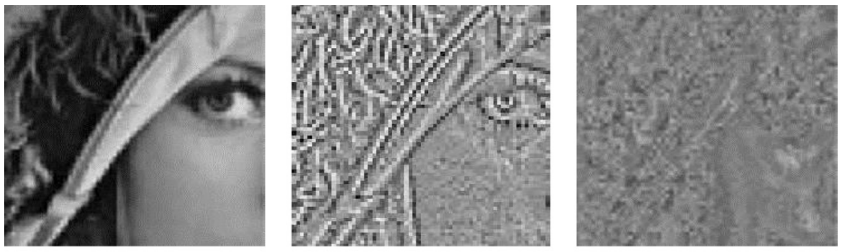
Various regularization terms such as the well-known total variation (TV) [2], non-local TV [3], sparse in a transform domain [5], and their modifications [4,6,24–26] were proposed. However, these methods exist some shortcomings. For example, the TV based methods suffer from staircasing effect in smooth area while the sparsity based methods show visible distortion around the edges. How to restore the real image and preserve as much important structures like edges and details without spurious distortion is still challenging.

Buades et al. [7] uses the residual of the real image and its nonlocal means (NLM) [8] estimation as the regularization term for image denoising and obtains good results. Specifically, the regularization term is $J(\boldsymbol{u}) = \left\|\boldsymbol{u} - NLM_f(\boldsymbol{u})\right\|_2^2$, where $NLM_f(\boldsymbol{u})$ denotes the nonlocal means of $\boldsymbol{u}$ and the weight is computed from the observed image $\boldsymbol{f}$, Wang and Bao [9] adapts this regularization term by using the real image rather than the observed image to compute the weight.

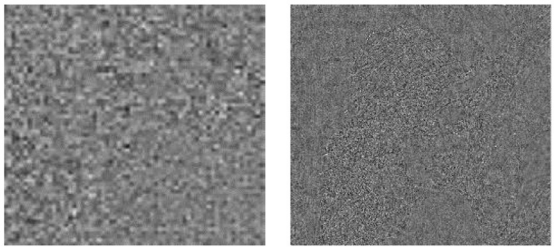
The general form of this kind of regularity is $J(\boldsymbol{u}) = \|\boldsymbol{u} - D(\boldsymbol{u})\|$, where $D(\cdot)$ is an image smoothing operator, $\boldsymbol{u} - D(\boldsymbol{u})$ is called method noise, and $\|\cdot\|$ is a vector norm measuring the method noise. Existing examples of $D(\cdot)$ include TV minimization [10], neighborhood filtering (NF) [11], and the block-matching and 3D filtering (BM3D) [12]. Fig. 1 shows the method noise $\boldsymbol{u} - D(\boldsymbol{u})$, with $\boldsymbol{u}$ being part of the clean image "Lenna", and $D(\cdot)$ being TV minimization, NF, NLM and BM3D, respectively. One can observe visually significant structure, such as large-scale edges and small-scale details, in the method noise of TV minimization. Though less

* Corresponding author.
E-mail addresses: chunyuyang666@163.com (C. Yang), wwwang@mail.xidian.edu.cn (W. Wang), xcfeng@mail.xidian.edu.cn (X. Feng), liuxin386@126.com (X. Liu).

https://doi.org/10.1016/j.sigpro.2018.11.009
0165-1684/

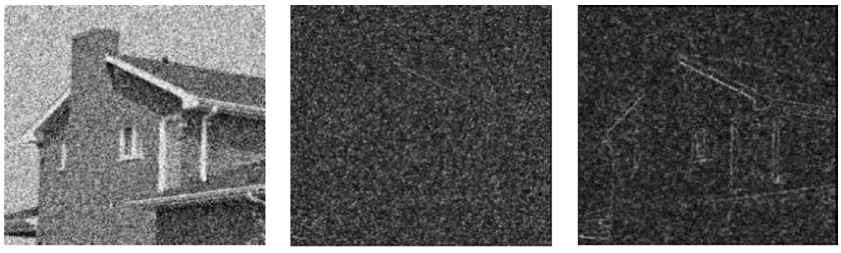


(a) Original image (b) TV (c) NF

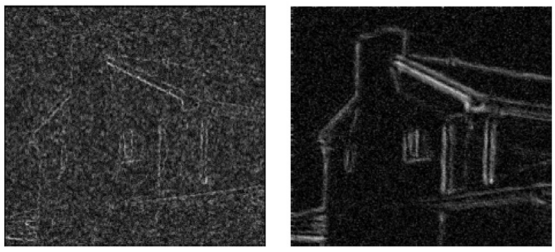


(d) NLM (e) BM3D

Fig. 1. Method noise experiment on a natural image.



(a) Noisy image (b) Roberts (c) Prewitt



(d) Sobel (e) GSG

Fig. 2. Edges detected by some gradient operators from a noisy image (the standard deviation of noise is 20).

visual meaningful structure is noticeable in the method noise of NF, NLM and BM3D, there still exists noticeable discrepancies. In the scenario of image deblurring, we expect that, in smooth area of $\boldsymbol{u}$, $D(\boldsymbol{u})$ is close to $\boldsymbol{u}$; or equivalently, the method noise should be close to zero; while around edges, considering the smoothing effect of $D(\cdot)$, we should allow proper discrepancy between $D(\boldsymbol{u})$ and $\boldsymbol{u}$. This suggests that a sparse penalty of the method noise should be better than the l_2 penalty of the method noise.

In this work we propose a novel regularization term. Basically, we use an l_1 -norm penalty of the method noise to enforce its sparsity. To better preserve edges in the recovered image, we introduce an image gradient-based weight in the method noise penalty. Specifically, in smooth area, the gradient is close to zero and the weight is close to one, the penalty of the method noise enforces the sparsity of the method noise; while around edges, the gradient is large and the weight is close to zero, which greatly reduce the penalty of the method noise and allow the edges be well preserved in the restored image. We use the global sparse gradient (GSG) model [13] to estimate image gradient because it can detect image gradient better than the classical gradient operators, such as Roberts operator [14], Prewitt operator [15], and Sobel operator [16], especially when the image is corrupted by noise. The proposed model is nonlinear and non-smooth, we propose a Bregman operator splitting (BOS) based algorithm [19] to solve our model. Numerical experiments show that our deblurring method outperforms some state-of-the-art methods such as l_2-NLM method and Plug-and-play Priors [17,18].

The rest of the paper is organized as follows. Section 2 reviews the related work: the l_2-NLM model [8] and the GSG model [13]. We present our model and give a new algorithm in Section 3. In

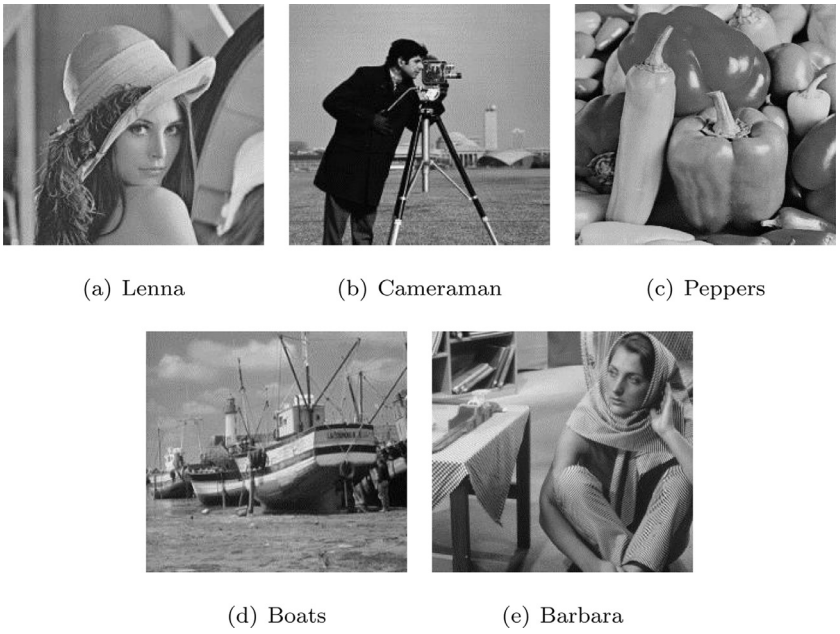


Fig. 3. Tested images.

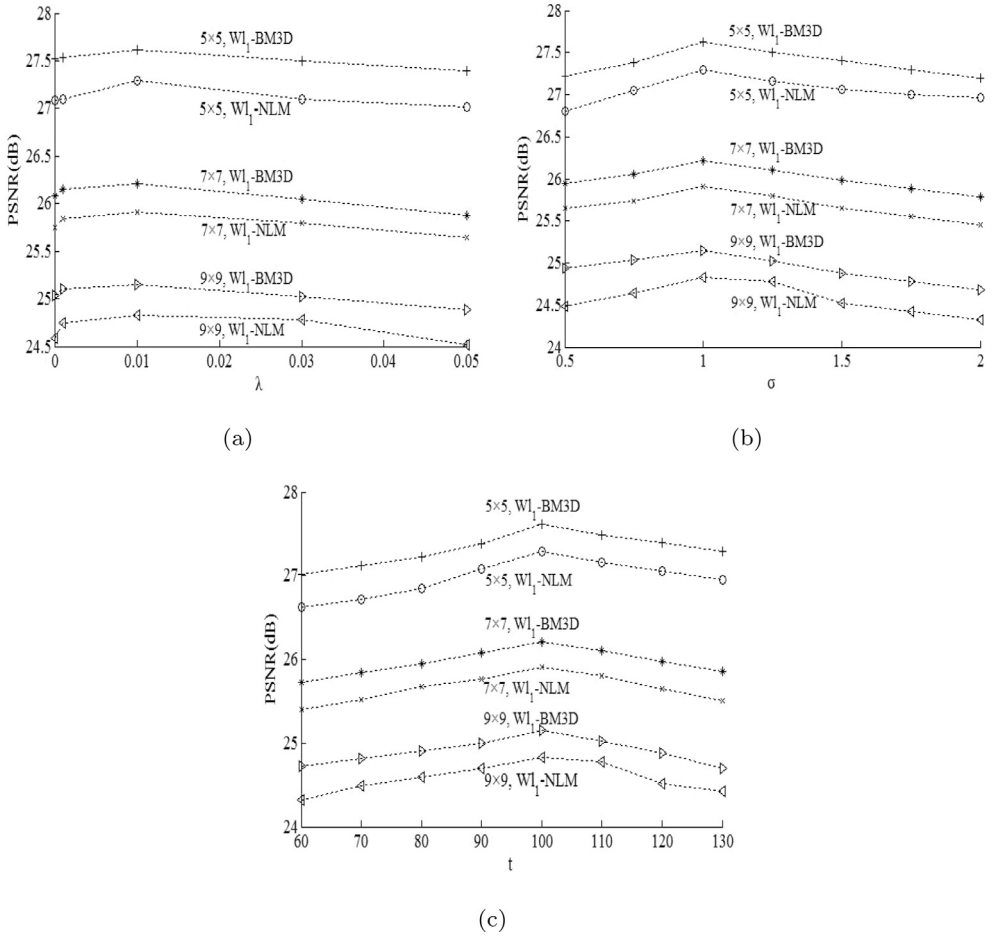
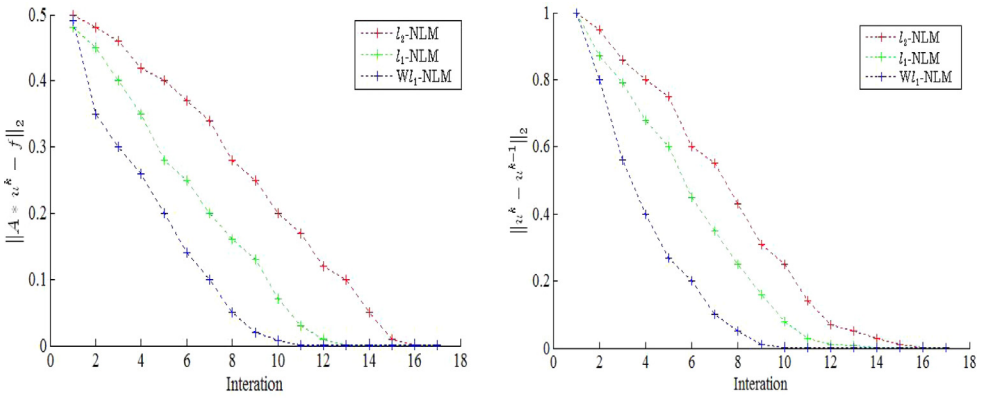
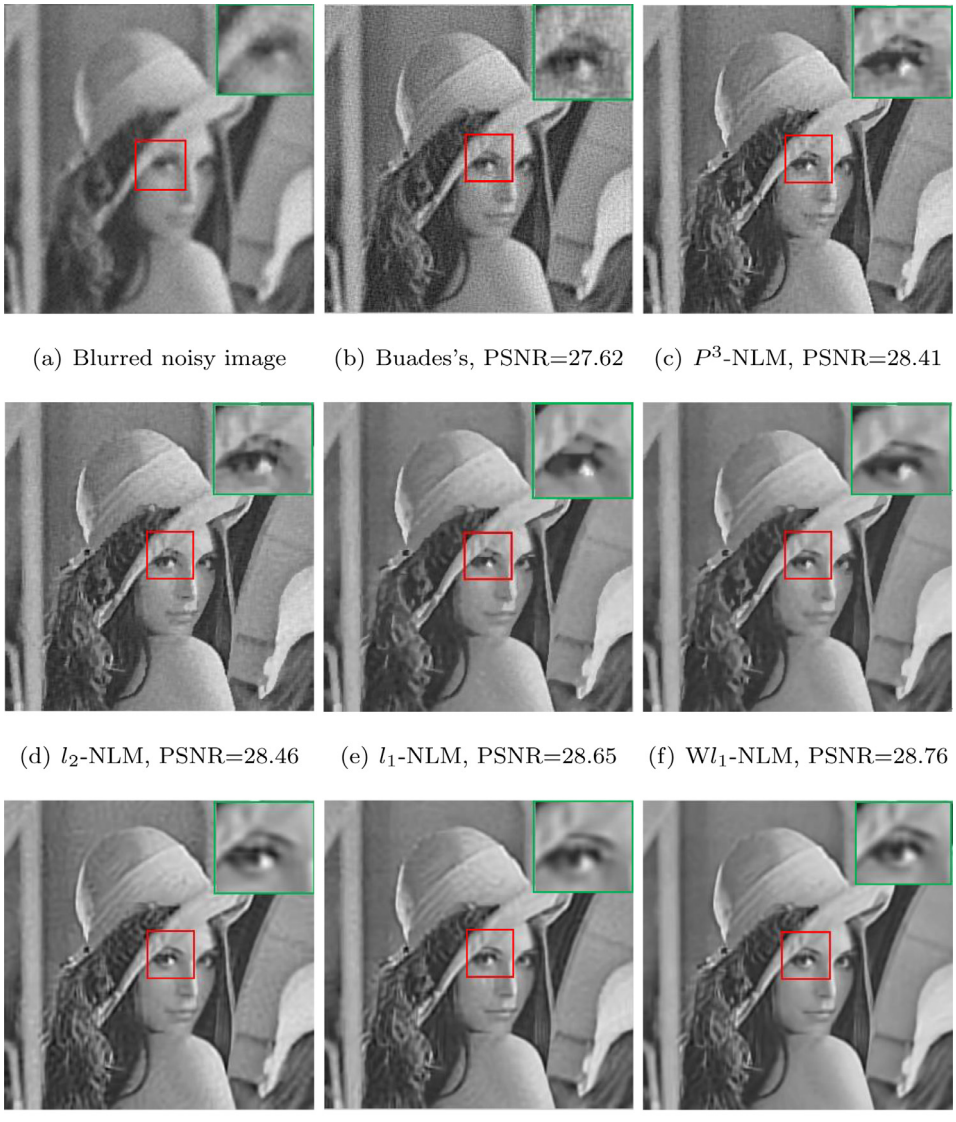


Fig. 4. Effects of parameters λ, δ and t on the performance of our method. (a), (b) and (c) shows the effect of the parameter λ, δ and t, respectively.



(a) $\left\|\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}\right\|_2$ (b) $\left\|\boldsymbol{u}^k - \boldsymbol{u}^{k-1}\right\|_2$

Fig. 5. The errors $\left\|\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}\right\|_2$ and $\left\|\boldsymbol{u}^k - \boldsymbol{u}^{k-1}\right\|_2$ vs. the iteration times for three methods: l_2-NLM, l_1-NLM and Wl_1-NLM.

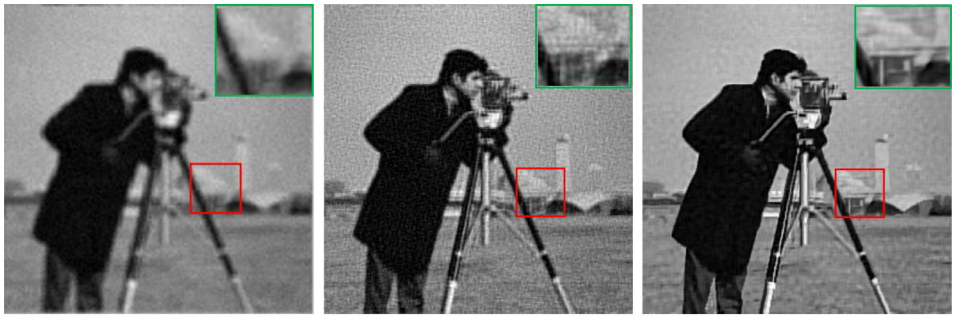


(a) Blurred noisy image (b) Buades's, PSNR=27.62 (c) P^3-NLM, PSNR=28.41

(d) l_2-NLM, PSNR=28.46 (e) l_1-NLM, PSNR=28.65 (f) Wl_1-NLM, PSNR=28.76

(g) P^3-BM3D, PSNR=28.67 (h) l_1-BM3D, PSNR=28.93 (i) Wl_1-BM3D, PSNR=29.05

Fig. 6. Restored results from 5×5 average blurred image Lenna by different methods.



(a) Blurred noisy image (b) Buades's, PSNR=25.94 (c) P^3-NLM, PSNR=26.89

(d) l_2-NLM, PSNR=26.94 (e) l_1-NLM, PSNR=27.17 (f) Wl_1-NLM, PSNR=27.29

(g) P^3-BM3D, PSNR=27.35 (h) l_1-BM3D, PSNR=27.58 (i) Wl_1-BM3D, PSNR=27.62

Fig. 7. Restored results from 5×5 average blurred image Cameraman by different methods.

Section 4, we present some experimental results obtained by our method and evaluate our method by both objective metrics and visual effects. Conclusions are drawn in Section 5.

2. Related work and discussion

The non-local means based regularization model [7] is:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \left\| \boldsymbol{u} - NLM_{\boldsymbol{f}}(\boldsymbol{u}) \right\|_2^2 \quad s.t. \|A\boldsymbol{u} - \boldsymbol{f}\|_2^2 \le \sigma^2 \tag{3}$$

where $NLM_{\boldsymbol{f}}(\boldsymbol{u})$ denotes the non-local means of $\boldsymbol{u}$ by using weight computed from $\boldsymbol{f}$. Since the observed image $\boldsymbol{f}$ is blurred and maybe corrupted by noise, the weight computed from it is unreliable. Wang and Bao [9] proposed to improve the above model by computing the weights from the restored image $\boldsymbol{u}$:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|\boldsymbol{u} - NLM_{\boldsymbol{u}}(\boldsymbol{u})\|_2^2 \quad s.t. \|A\boldsymbol{u} - \boldsymbol{f}\|_2^2 \le \sigma^2 \tag{4}$$

To solve the problem in Eq. (4), they propose the following BOS algorithm, which combines the Bregman iteration and operator splitting into a unified framework (more detailed explanations on BOS can be found in [9,19,20]):

$$\begin{cases} \boldsymbol{v}^{k+1} = \boldsymbol{u}^k - \delta \boldsymbol{A}^T * (\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}^k) \\ \boldsymbol{u}^{k+1} = \arg\min_{\boldsymbol{u}} (\left\| \boldsymbol{u} - NLM_{\boldsymbol{v}^{k+1}}(\boldsymbol{v}^{k+1}) \right\|_2^2 + \lambda \left\| \boldsymbol{u} - \boldsymbol{v}^{k+1} \right\|_2^2) \\ \boldsymbol{f}^{k+1} = \boldsymbol{f}^k + \boldsymbol{f} - \boldsymbol{A} * \boldsymbol{u}^{k+1} \end{cases} \tag{5}$$

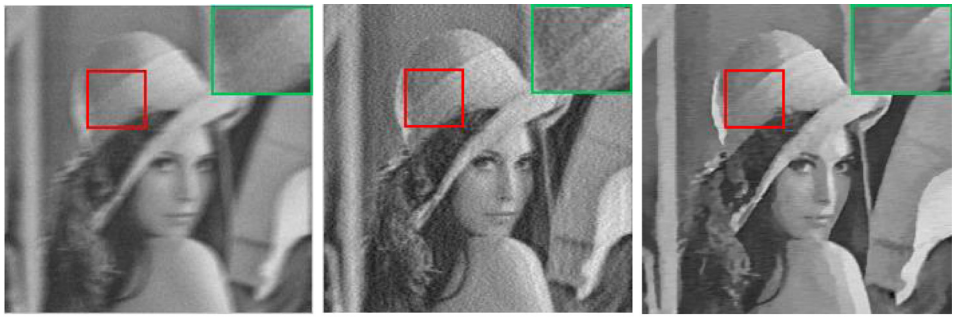
where δ is a positive number, λ is a scaling parameter which balances the two terms of the object function, $\boldsymbol{u}^k$ denotes an intermediate restored image, and $\boldsymbol{f}^k$, $\boldsymbol{v}^k$ are the iterative observed image and intermediate blurred image. The minimization problem in Eq. (5) has an analytic solution:

$$\boldsymbol{u}^{k+1} = \frac{1}{1+\lambda} NLM_{\boldsymbol{v}^{k+1}}(\boldsymbol{v}^{k+1}) + \frac{\lambda}{1+\lambda} \boldsymbol{v}^{k+1} \tag{6}$$

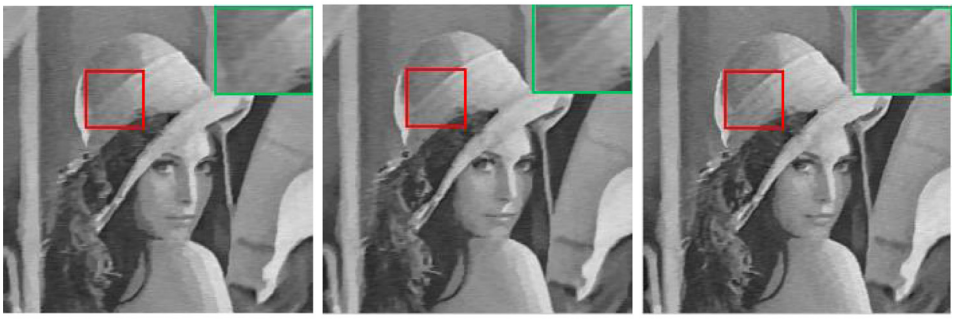
Putting together, l_2-NLM algorithm can be summarized as follows:

$$\begin{cases} \boldsymbol{v}^{k+1} = \boldsymbol{u}^k - \delta \boldsymbol{A}^T * (\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}^k) \\ \boldsymbol{u}^{k+1} = \frac{1}{1+\lambda} NLM_{\boldsymbol{v}^{k+1}}(\boldsymbol{v}^{k+1}) + \frac{\lambda}{1+\lambda} \boldsymbol{v}^{k+1} \\ \boldsymbol{f}^{k+1} = \boldsymbol{f}^k + \boldsymbol{f} - \boldsymbol{A} * \boldsymbol{u}^{k+1} \end{cases} \tag{7}$$

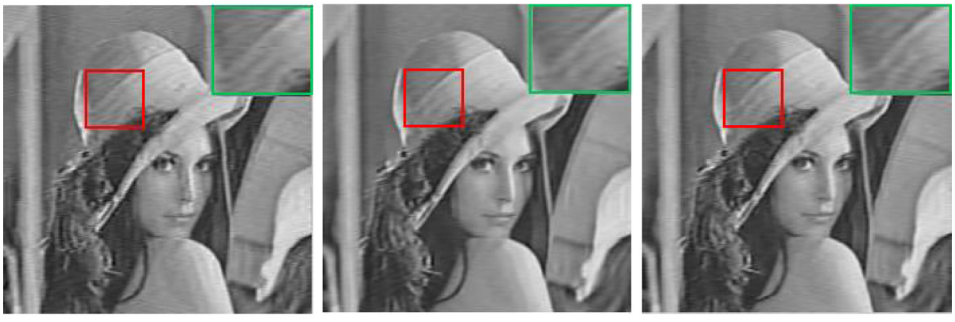
A direct extension of the above algorithm can be obtained by replacing the non-local mean operator NLM with a general image



(a) Blurred noisy image (b) Buades's, PSNR=26.33 (c) P^3-NLM, PSNR=27.09



(d) l_2-NLM, PSNR=27.14 (e) l_1-NLM, PSNR=27.40 (f) Wl_1-NLM, PSNR=27.54



(g) P^3-BM3D, PSNR=27.61 (h) l_1-BM3D, PSNR=27.83 (i) Wl_1-BM3D, PSNR=27.97

Fig. 8. Restored results from motion blurred ($\theta = 10^0$, $L = 5$) image "Lenna" by different methods.

smoothing operator $D(u)$, which leads to the following algorithm:

$$\begin{cases} \boldsymbol{v}^{k+1} = \boldsymbol{u}^k - \delta \boldsymbol{A}^T * (\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}^k) \\ \boldsymbol{u}^{k+1} = \dfrac{1}{1+\lambda} D_{\boldsymbol{v}^{k+1}}(\boldsymbol{v}^{k+1}) + \dfrac{\lambda}{1+\lambda} \boldsymbol{v}^{k+1} \\ \boldsymbol{f}^{k+1} = \boldsymbol{f}^k + \boldsymbol{f} - \boldsymbol{A} * \boldsymbol{u}^{k+1} \end{cases} \tag{8}$$

We call it l_2-Denoiser. When $D(u)$ is taken as a particular smoothing operator such as BM3D, we name it l_2-BM3D.

The GSG model [13] is a state-of-the-art method for detecting gradients directly from a noisy image. Different from the traditional local gradient operators, such as Roberts operator [14], Prewitt operator [15], Sobel operator [16], the GSG model utilizes the first order Taylor expansion to estimate the gradient. The specific model is as follows:

$$\nabla \boldsymbol{u} = \underset{\nabla \boldsymbol{u} \in R^{N\times 2}}{\arg\min} \sum_{i,j=1}^{N} \frac{1}{N} \varphi_{ij}^s (z_i - z_j + \nabla u_i (x_j - x_i)) + \lambda \|\nabla \boldsymbol{u}\|_1 \tag{9}$$

where $z_i = z(x_i)$, $z(x)$, $u(x)$ and $k(x)$, $x \in R^N$, respectively represents the noisy image, the clean image and the noise, which is assumed to be i.i.d.additive Gaussian noise, or:

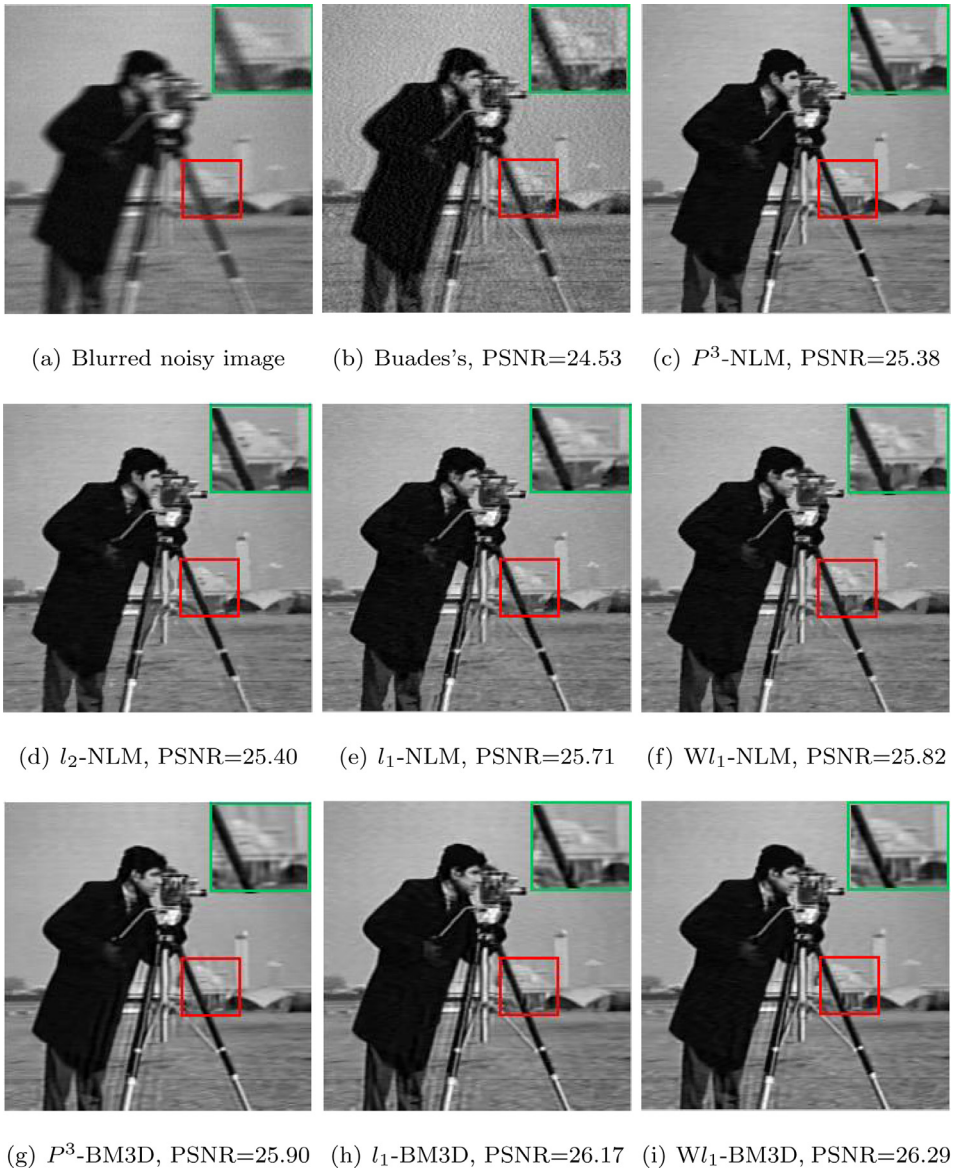
$$z(x) = u(x) + k(x) \tag{10}$$

The expression $z_i - z_j + \nabla u_i(x_j - x_i)$ denotes the error between z_i and its first order Taylor expansion at x_j, so the first term $\sum_{i,j=1}^{N} \frac{1}{N} \varphi_{ij}^s (z_i - z_j + \nabla u_i (x_j - x_i))$ denotes the total approximation error, where the kernel $\varphi_{ij}^s = \exp(-\frac{\|x_j - x_i\|_2^2}{2s^2})$ is incorporated to emphasize the role of the nearby points and s is a scale parameter controlling the rate of decay of the kernel. The regularization term $\|\nabla \boldsymbol{u}\|_1$ is used to suppress the noise. To solve problem (10), Zhang et al. [13] proposed the following iterative algorithm by using the forward-backward splitting technique [21]:

$$\nabla u_i^{(k+1)} = \mathrm{T}_{\lambda\alpha}(\mathrm{Q}_i^{(k)}) = (\mathrm{T}_{\lambda\alpha}\mathrm{Q}_{1,i}^{(k)}, \mathrm{T}_{\lambda\alpha}\mathrm{Q}_{2,i}^{(k)}) \tag{11}$$

where α is a constant parameter, $\mathrm{Q}_i^{(k)} = \nabla u_i^{(k)} - \alpha \frac{2}{N} \varphi_{ij}^s (z_i - z_j + \nabla u_i^{(k)} (x_j - x_i)(x_j - x_i)^T)$, T_β: $R^n \to R^n$ is the well-known soft thresholding operator [22]:

$$T_\beta(x_i) = max\{|x_i| - \beta, 0\} \cdot sgn(x_i) \tag{12}$$



(a) Blurred noisy image (b) Buades's, PSNR=24.53 (c) P^3-NLM, PSNR=25.38

(d) l_2-NLM, PSNR=25.40 (e) l_1-NLM, PSNR=25.71 (f) Wl_1-NLM, PSNR=25.82

(g) P^3-BM3D, PSNR=25.90 (h) l_1-BM3D, PSNR=26.17 (i) Wl_1-BM3D, PSNR=26.29

Fig. 9. Restored results from motion blurred ($\theta = 10^0$, $L = 5$) image Cameraman by different methods.

Fig. 2 shows gradients obtained by the GSG method and some commonly used gradient operators from the noisy image "House". One can see that the GSG method is much more stable than other methods in detecting the image edges.

3. The proposed model and algorithm

3.1. The proposed model

The NLM-based regularization model shows that the method noise $\boldsymbol{u} - D(\boldsymbol{u})$ can be used to regularize images in applications of image restoration, where $D(\cdot)$ is an existing image smoothing operator such as NLM, NF, and BM3D. Based on the observation and analysis we made in the introduction, we propose to use an l_1-norm penalty of the method noise to enforce its sparsity. To better preserve edges in the recovered image, we also introduce a gradient-based weight in the method noise to adaptively adjust the penalty. Specifically, in smooth area, the weight should be close to one and enforce the sparsity of the method noise; while around edges, the weight should be close to zero and help to preserve edges. Because the global sparse gradient (GSG) model [13] can detect image gradient better than the classical gradient operators, especially when the image is corrupted by heavy noise, we use GSG gradient operator to compute the gradient-based weight.

Motivated by the above analysis, we present the following model:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|w(|\nabla \boldsymbol{u}|) \odot (\boldsymbol{u} - D(\boldsymbol{u}))\|_1 \quad s.t. \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \le \sigma^2 \tag{13}$$

where $\odot$ stands for Hadamard product, $\|.\|_1$ is the l_1 norm, and the weight is defined by

$$w(|\nabla \boldsymbol{u}|) = 1/\left(1 + \frac{|\nabla \boldsymbol{u}|}{t}\right) \tag{14}$$

in which t is a scaling constant. Note that such a weight is used in the anisotropic diffusion [21] to protect edges. It is easy to see $w(|\nabla \boldsymbol{u}|) \in (0, 1)$. In smooth areas of the image, $|\nabla \boldsymbol{u}| \approx 0$, $w(|\nabla \boldsymbol{u}|) \approx 1$; while around edges, $|\nabla \boldsymbol{u}| \to \infty$, $w(|\nabla \boldsymbol{u}|) \to 0$. The objective function essentially enforces that the method noise in the smooth area to be zero, or the restored image close to the smoothed image; while around edges, the small weight tends to protect edges from being over-smoothed. The constraint ensures

the fidelity of the restored image to the observed image. We call our model Wl_1-Denoiser. We will further analyze the advantages of our method in detail (see Section 3.3) from the solution of our model.

3.2. Minimization algorithm

Note that the proposed model (13) is non-linear and non-smooth. To make the solution easier, we use the GSG model [13] and the Tikhonov regularization method as in [23] to obtain an initial image $\boldsymbol{u}^0$ and its gradient $\nabla\boldsymbol{u}^0$ from the observed image $\boldsymbol{f}$, and we set the weight $w(|\nabla\boldsymbol{u}|) = w\left(\left|\nabla\boldsymbol{u}^0\right|\right)$ in our model.

By using the BOS technique, we give the following iterative algorithm to solve the problem (13):

$$\boldsymbol{v}^{k+1} = \boldsymbol{u}^k - \delta\boldsymbol{A}^T * (\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}^k) \tag{15}$$

$$\boldsymbol{u}^{k+1} = \arg\min_{\boldsymbol{u}} \left\| w\left(\left|\nabla\boldsymbol{u}^0\right|\right) \odot \left(\boldsymbol{u} - D(\boldsymbol{v}^{k+1})\right)\right\|_1 + \lambda\left\|\boldsymbol{u} - \boldsymbol{v}^{k+1}\right\|_2^2 \tag{16}$$

$$\boldsymbol{f}^{k+1} = \boldsymbol{f}^k + \boldsymbol{f} - \boldsymbol{A} * \boldsymbol{u}^{k+1} \tag{17}$$

To solve the subproblem (16), we first let $p = \boldsymbol{u} - D(\boldsymbol{v}^{k+1})$ and rewrite (16) in the following form:

$$\boldsymbol{p}^{k+1} = \arg\min_{\boldsymbol{p}} \lambda\left\|\boldsymbol{p} - \left(\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})\right)\right\|_2^2 + \left\| w\left(\left|\nabla\boldsymbol{u}^0\right|\right) \odot \boldsymbol{p}\right\|_1 \tag{18}$$

then we solve problem (18) by the soft thresholding operator [22]:

$$\boldsymbol{p}^{k+1} = \mathrm{T}_{\frac{w\left(|\nabla\boldsymbol{u}^0|\right)}{2\lambda}}\left(\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})\right) \tag{19}$$

and obtain $\boldsymbol{u}^{k+1}$ as:

$$\boldsymbol{u}^{k+1} = \mathrm{T}_{\frac{w\left(|\nabla\boldsymbol{u}^0|\right)}{2\lambda}}\left(\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})\right) + D(\boldsymbol{v}^{k+1}) \tag{20}$$

where T_β: $R^n \to R^n$ is the well-known soft thresholding operator :

$$T_\beta(x_i) = max\{|x_i| - \beta, 0\} \cdot sgn(x_i) \tag{21}$$

The pseudo-code is given in Algorithm 1, where *Atol* and *Btol* are two error tolerances, and *Max*_iteration is the maximum iteration times.

Algorithm 1 Wl_1-Denoiser for image deblurring.

Input: Observed image $\boldsymbol{f}$, convolution kernel $\boldsymbol{A}$, initial image $\boldsymbol{u}^0$, values of parameters λ, δ, *Atol* and *Btol*.

1: compute: $w\left(\left|\nabla\boldsymbol{u}^0\right|\right) = 1/\left(1 + \frac{|\nabla\boldsymbol{u}^0|}{t}\right)$.
2: Initialize: $k = 0$, $\boldsymbol{f}^0 = \boldsymbol{f}$.
3: **while** not converge **do**
4: fix $\boldsymbol{u}$ and $\boldsymbol{f}$, update $\boldsymbol{v}$ by Eq. (15);
5: fix $\boldsymbol{v}$ and $\boldsymbol{f}$, update $\boldsymbol{u}$ by Eq. (19);
6: fix $\boldsymbol{u}$ and $\boldsymbol{v}$, update $\boldsymbol{f}$ by Eq. (17);
7: $k = k + 1$;
8: check the convergence condition:
9: $k \le Max\_iteration$ && $\left\|\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}\right\|_2 > Atol$ && $\left\|\boldsymbol{u}^k - \boldsymbol{u}^{k-1}\right\|_2 < Btol$
10: **end while**

Output: $\boldsymbol{u}^k$.

The computational complexity of the proposed algorithm is provided as follows. The computational complexity of the weight $w(|\nabla u^0|)$ is $O(N^2)$ [13]. The computational complexity for updating $\boldsymbol{v}$ is $O(N)$, The soft thresholding operator has a computational complexity of $O(N^2)$. The computational complexity for updating $\boldsymbol{f}$ is $O(N)$. The complexity of these computation ($D(\cdot)$ excluded) is $O(N^2)$. The total computational complexity of Algorithm 1 depends on the image smoothing operator $D(\cdot)$ used. In the experiment, we will show that our method is more efficient than other methods in the sense that the errors $\left\|\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}\right\|_2$ and $\left\|\boldsymbol{u}^k - \boldsymbol{u}^{k-1}\right\|_2$ descend more quickly as the iteration progresses.

3.3. Advantages of our model

Note that, if the weight function $w\left(\left|\nabla\boldsymbol{u}^0\right|\right) = 1$, our Wl_1-Denoiser model degenerates into the following l_1-Denoiser model:

$$\hat{\boldsymbol{u}} = \arg\min_{\boldsymbol{u}} \|\boldsymbol{u} - D(\boldsymbol{u})\|_1 \quad s.t. \|\boldsymbol{A}\boldsymbol{u} - \boldsymbol{f}\|_2^2 \le \sigma^2 \tag{22}$$

similar to (13), (22) can be solved by using the BOS technique. The formula for updating $\boldsymbol{v}^{k+1}$ and $\boldsymbol{f}^{k+1}$ is the same as Eqs. (15) and (17), respectively. The formula for updating $\boldsymbol{u}^{k+1}$ is:

$$\boldsymbol{u}^{k+1} = \mathrm{T}_{\frac{1}{2\lambda}}\left(\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})\right) + D(\boldsymbol{v}^{k+1}) \tag{23}$$

Furthermore, $\frac{1}{2\lambda} \to +\infty$ when $\lambda \to 0$. By the definition of Soft thresholding operator [22], Eq. (23) degenerates into:

$$\boldsymbol{u}^{k+1} = D(\boldsymbol{v}^{k+1}) \tag{24}$$

This leads to the following BOS iteration:

$$\begin{cases} \boldsymbol{v}^{k+1} = \boldsymbol{u}^k - \delta\boldsymbol{A}^T * (\boldsymbol{A} * \boldsymbol{u}^k - \boldsymbol{f}^k) \\ \boldsymbol{u}^{k+1} = D(\boldsymbol{v}^{k+1}) \\ \boldsymbol{f}^{k+1} = \boldsymbol{f}^k + \boldsymbol{f} - \boldsymbol{A} * \boldsymbol{u}^{k+1} \end{cases} \tag{25}$$

The iteration scheme in Eq. (25) is consistent with the Plug-and-Play Priors algorithm [17,18], which we call P^3-Denoiser.

Compare Eq. (19) of our method with Eq. (23) of l_1-Denoiser, Eq. (24) of P^3-Denoiser, and Eq. (6) of l_2-NLM, one can appreciate the advantages of our method over others from the solution process. In the iteration schemes of all methods, $\boldsymbol{v}^{k+1}$ is essentially an intermediate blurred image and $\boldsymbol{u}^{k+1}$ further refines $\boldsymbol{v}^{k+1}$. Different models lead to different refining strategies. Our method improves the quality of $\boldsymbol{u}^{k+1}$ in this way: it first applies an image smoothing operator $D(\cdot)$ to the intermediate blurred image $\boldsymbol{v}^{k+1}$, yielding a smoothed clean version $D(\boldsymbol{v}^{k+1})$ and a residual or the method noise $\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})$, which mainly contains the edges, small scaled details and noise of $\boldsymbol{v}^{k+1}$. To restore the edges back while discarding the noise, it applies the soft-thresholding operator $\mathrm{T}_{\frac{w\left(|\nabla\boldsymbol{u}^0|\right)}{2\lambda}}$ to the method noise $\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})$ with gradient tuning thresholds $\frac{w\left(|\nabla\boldsymbol{u}^0|\right)}{2\lambda}$. In smooth area of the image, the method noise mainly contains noise and $w(|\nabla\boldsymbol{u}^0|) \approx 1$, the soft-thresholding operator filters out noise and small scaled details; around edges, the method noise mainly contains edges, and $w(|\nabla\boldsymbol{u}^0|) \approx 0$, the soft thresholding operator preserves the edges very well. By adding the edges $\mathrm{T}_{\frac{w\left(|\nabla\boldsymbol{u}^0|\right)}{2\lambda}}\left(\boldsymbol{v}^{k+1} - D(\boldsymbol{v}^{k+1})\right)$ back to the smooth and clean image $D(\boldsymbol{v}^{k+1})$, our method obtains a refined version $\boldsymbol{u}^{k+1}$ of $\boldsymbol{v}^{k+1}$. The method l_1-Denoiser refines $\boldsymbol{v}^{k+1}$ in a similar way; the major difference is that it uses a uniform threshold value in filtering the method noise, thus may wrongly filter out some edges and cannot preserve edges as well as our method. The P^3-Denoiser actually does not restore edges back. Actually, it refines $\boldsymbol{v}^{k+1}$ directly by applying a smoothing operator $D(\cdot)$, thus may cause over smoothing of the edges. The method l_2-NLM refines $\boldsymbol{v}^{k+1}$ by the weighted average of $\boldsymbol{v}^{k+1}$ and its smoothed clean version $NLM_{\boldsymbol{v}^{k+1}}(\boldsymbol{v}^{k+1})$, which may also cause edges to be over smoothed.

Table 1
Deblurring performance of different methods for average blur.

Blur mask size	Image		Buades	P^3-NLM	l_2-NLM	l_1-NLM	Wl_1-NLM	P^3-BM3D	l_1-BM3D	Wl_1-BM3D
5×5	Lenna	PSNR	27.62	28.41	28.46	28.65	28.76	28.67	28.93	**29.05**
		SSIM	0.747	0.790	0.786	0.805	0.813	0.810	0.826	**0.835**
	Cameraman	PSNR	25.94	26.89	26.94	27.17	27.29	27.35	27.58	**27.62**
		SSIM	0.759	0.795	0.800	0.811	0.832	0.824	0.845	**0.853**
	Peppers	PSNR	29.03	29.78	29.81	30.02	30.11	30.04	30.23	**30.43**
		SSIM	0.801	0.847	0.851	0.879	0.884	0.869	0.884	**0.897**
	Boats	PSNR	26.19	27.07	27.12	27.34	27.45	27.38	27.62	**27.71**
		SSIM	0.732	0.765	0.768	0.783	0.795	0.783	0.802	**0.811**
	Barbara	PSNR	27.25	28.10	28.13	28.30	28.43	28.49	28.64	**28.72**
		SSIM	0.745	0.788	0.789	0.805	0.811	0.805	0.827	**0.839**
	Ave.	PSNR	27.20	28.05	28.09	28.29	28.40	28.39	28.60	**28.71**
		SSIM	0.756	0.797	0.798	0.816	0.827	0.818	0.836	**0.846**
7×7	Lenna	PSNR	26.44	26.94	27.09	27.31	27.43	27.4	27.62	**27.78**
		SSIM	0.701	0.744	0.747	0.763	0.770	0.769	0.801	**0.815**
	Cameraman	PSNR	24.44	25.25	25.33	25.80	25.91	26.87	26.13	**26.21**
		SSIM	0.709	0.759	0.760	0.775	0.787	0.779	0.798	**0.802**
	Peppers	PSNR	27.63	28.59	28.68	28.72	28.8	28.79	29.00	**29.12**
		SSIM	0.771	0.818	0.821	0.837	0.845	0.84	0.865	**0.879**
	Boats	PSNR	24.36	25.84	25.97	26.07	26.22	26.19	26.41	**26.54**
		SSIM	0.668	0.713	0.715	0.731	0.744	0.735	0.752	**0.768**
	Barbara	PSNR	25.65	26.68	26.72	27.04	27.16	27.23	27.30	**27.43**
		SSIM	0.702	0.750	0.752	0.767	0.776	0.773	0.789	**0.783**
	Ave.	PSNR	25.70	26.67	26.75	26.99	27.10	27.01	27.28	**27.41**
		SSIM	0.710	0.757	0.759	0.775	0.784	0.779	0.801	**0.810**
9×9	Lenna	PSNR	25.62	26.31	26.37	26.59	26.70	26.69	26.93	**27.02**
		SSIM	0.661	0.719	0.721	0.743	0.752	0.748	0.769	**0.780**
	Cameraman	PSNR	23.67	24.56	24.60	24.72	24.83	24.78	25.04	**25.15**
		SSIM	0.694	0.734	0.738	0.762	0.77	0.761	0.775	**0.789**
	Peppers	PSNR	26.58	27.30	27.32	27.59	27.72	27.70	27.91	**28.03**
		SSIM	0.752	0.80	0.802	0.820	0.834	0.829	0.842	**0.855**
	Boats	PSNR	24.1	24.85	24.89	25.07	25.21	25.14	25.38	**25.50**
		SSIM	0.665	0.671	0.673	0.686	0.699	0.681	0.699	**0.706**
	Barbara	PSNR	25.13	25.94	26	26.15	26.26	26.22	26.45	**26.54**
		SSIM	0.663	0.718	0.720	0.739	0.745	0.744	0.761	**0.771**
	Ave.	PSNR	25.02	25.79	25.83	26.02	26.14	26.10	26.34	**26.45**
		SSIM	0.687	0.728	0.731	0.750	0.760	0.753	0.769	**0.780**

4. Experimental results and analysis

Extended experiments show the effectiveness and efficiency of our method. Here we present our experimental results in two blur scenarios: average blur and motion blur, to demonstrate the effectiveness of the proposed method. Fig. 3 shows some of the test images (grayscale, 256×256) used in our experiment. The average blur is simulated by applying box filter of size 5×5, 7×7 and 9×9, respectively to the test images. The larger the size, the heavier the image is blurred. The motion blur consists of two causes: rotation unclockwise through an angle θ (in degree) and shifting by L (in pixel). We consider three examples of motion blur: $(\theta, L) = (5^0, 5)$, and $(\theta, L) = (10^0, 5)$, $(\theta, L) = (10^0, 10)$. To show robustness of our method to noise, we also add i.i.d. white Gaussian noise with variance $\sigma^2 = 25$ into the blurred images.

We compare the proposed method with several state-of-the-art deblurring methods, including Buades's non-local means regularization method [7], l_2-Denoiser [9], l_1-Denoiser and P^3-Denoiser [17,18], with $D(u)$ being taken as NLM and BM3D. we assess the methods by two objective image quality metrics: the Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Metric (SSIM). We also present the restored image for visual assessment.

The initial image gradient is estimated by using the GSG method and the parameters are set as the same as given in [13]. For all experiments, the maximum iteration number $Max_{\text{iteration}} = 20$. The parameters of other methods are set to get the optimal results.

We select the parameters λ, δ and t in our method manually. Taking the average blurred "Cameraman" as an example, we show how these parameters influence the PSNR metric of the restored images obtained by different methods in Fig. 4, where (a), (b) and (c) shows the effect of the parameter λ, δ and t, respectively. Beside each curve, we state the size of the average blur mask and our method with particular $D(\cdot)$. For example, (5×5, Wl_1 -BM3D) means the size of the average blur mask is 5×5, and the image is restored by using our method Wl_1-BM3D, in which the image smoothing operator BM3D is used. We select the optimal parameter values: $\lambda = 0.01$, $\delta = 1$ and $t = 100$, which yields the best PSNR values. Experiments show that these parameter values also apply for other images, so we use these values for all test images.

To show the efficiency of our method, we display how the errors $\left\|A * u^k - f\right\|_2$ and $\left\|u^k - u^{k-1}\right\|_2$ descend in the iteration for three methods: l_2-NLM, l_1-NLM and Wl_1-NLM in Fig. 5. The test image is "Lenna" and the blur mask size is 7×7. One can see that errors of our method descend more quickly than that of other methods. In other words, our method needs fewer iterations to obtain an estimated image within the same error tolerances. The curves also show our method converges numerically. In all experiments, we set the error tolerances $Atol = 10^{-4}$ and $Btol = 10^{-4}$.

To evaluate our method objectively, we report the PSNRs and SSIMs of the restored images by different methods in Table 1 for average blur, and Table 2 for motion blur. The best results are in bold font. It can be observed that the PSNRs and SSIMs of the proposed method are superior than that of other methods. Take the average blur 7×7 as an example. In terms of average PSNR, our method Wl_1-NLM improves Buades's method, P^3-NLM, l_2-NLM and l_1-NLM by 1.40 dB, 0.43 dB, 0.35 dB and 0.11 dB, respectively; our method Wl_1-BM3D improves P^3-BM3D and l_1-BM3D by 0.40 dB and 0.13 dB. In terms of average SSIM, the proposed Wl_1-NLM improves these methods by 0.074, 0.030, 0.029 and 0.011;

Table 2
Deblurring performance of different methods for motion blur.

Motion blur	Image		Buades	P^3-NLM	l_2-NLM	l_1-NLM	Wl_1-NLM	P^3-BM3D	l_1-BM3D	Wl_1-BM3D
$\theta = 5^0$	Lenna	PSNR	27.90	28.79	28.81	29.03	29.15	29.24	29.46	**29.58**
$L = 5$		SSIM	0.749	0.808	0.809	0.835	0.846	0.852	0.873	**0.884**
	Cameraman	PSNR	27.24	28.02	28.00	28.26	28.38	28.42	28.61	**28.72**
		SSIM	0.718	0.764	0.768	0.790	0.802	0.798	0.810	**0.822**
	Peppers	PSNR	28.87	29.74	29.79	29.97	30.07	30.02	30.26	**30.40**
		SSIM	0.782	0.830	0.832	0.856	0.864	0.872	0.892	**0.905**
	Boats	PSNR	27.05	27.94	27.95	28.12	28.20	28.26	28.48	**28.58**
		SSIM	0.705	0.754	0.753	0.772	0.785	0.791	0.815	**0.824**
	Barbara	PSNR	26.72	27.61	27.68	27.89	28.00	28.14	28.33	**28.49**
		SSIM	0.724	0.760	0.764	0.784	0.796	0.808	0.831	**0.841**
	Ave.	PSNR	27.56	28.42	28.45	28.65	28.76	28.81	29.02	**29.15**
		SSIM	0.736	0.783	0.785	0.807	0.819	0.824	0.844	**0.855**
$\theta = 10^0$	Lenna	PSNR	26.33	27.09	27.14	27.40	27.54	27.61	27.83	**27.97**
$L = 5$		SSIM	0.681	0.719	0.721	0.746	0.757	0.762	0.788	**0.801**
	Cameraman	PSNR	24.53	25.38	25.40	25.71	25.82	25.90	26.17	**26.29**
		SSIM	0.698	0.742	0.747	0.776	0.790	0.792	0.831	**0.844**
	Peppers	PSNR	27.86	28.64	28.62	28.90	29.05	29.11	29.32	**29.48**
		SSIM	0.747	0.790	0.792	0.816	0.828	0.832	0.857	**0.873**
	Boats	PSNR	25.16	25.92	25.91	26.08	26.19	26.17	26.39	**26.51**
		SSIM	0.677	0.730	0.739	0.757	0.771	0.769	0.792	**0.805**
	Barbara	PSNR	25.98	26.74	26.76	26.95	27.09	27.12	27.35	**27.49**
		SSIM	0.661	0.728	0.729	0.750	0.768	0.771	0.794	**0.810**
	Ave.	PSNR	26.97	26.75	26.77	27.00	27.13	27.18	27.41	**27.55**
		SSIM	0.692	0.742	0.746	0.769	0.782	0.785	0.812	**0.826**
$\theta = 10^0$	Lenna	PSNR	26.04	26.80	26.79	27.01	27.12	27.17	27.40	**27.52**
$L = 10$		SSIM	0.654	0.705	0.702	0.729	0.740	0.738	0.760	**0.771**
	Cameraman	PSNR	24.59	25.32	25.35	25.58	25.71	25.84	26.08	**26.18**
		SSIM	0.677	0.721	0.725	0.749	0.761	0.768	0.788	**0.800**
	Peppers	PSNR	27.31	28.06	28.08	28.27	28.39	28.42	28.65	**28.76**
		SSIM	0.701	0.758	0.761	0.784	0.797	0.804	0.825	**0.837**
	Boats	PSNR	24.78	25.51	25.54	25.74	25.87	25.91	26.16	**26.19**
		SSIM	0.667	0.714	0.715	0.735	0.746	0.741	0.760	**0.772**
	Barbara	PSNR	25.40	26.16	26.17	26.36	26.48	26.49	26.72	**26.86**
		SSIM	0.645	0.698	0.701	0.720	0.732	0.736	0.759	**0.769**
	Ave.	PSNR	25.62	26.37	26.39	26.59	26.71	26.77	27.00	**27.10**
		SSIM	0.668	0.719	0.720	0.743	0.755	0.757	0.778	**0.789**

our method Wl_1-BM3D improves P^3-BM3D and l_1-BM3D by 0.031 and 0.009, respectively. As for motion blur, our method Wl_1-NLM and Wl_1-BM3D outperform other methods. In all, by utilizing gradient weighted sparse penalty of the method noise, the proposed method Wl_1-Denoiser can perform better than l_1-Denoiser and l_2-Denoiser in protecting edges while removing noise.

For visual assessment, we show the restored images of Lenna and Cameraman in Figs. 6,7 for average blur, and Figs. 8,9 for motion blur. The size of the average blur mask is 5×5, and the parameters of the motion blur are $\theta = 10^0$, $L = 5$. Either using NLM or BM3D as the image smoothing operator, our Wl_1-Denoiser obtain better results than l_1-Denoiser, l_2-Denoiser and P^3-Denoiser. For example, the image (Fig. 6(i)) restored by our method Wl_1-BM3D is visually much better than the images (Fig. 6(h)) restored by l_1-BM3D. In the smooth areas of the image in figure Fig. 6(i), one can notice that the noise is removed perfectly, while in the smooth areas of the image in figure Fig. 6(h), one can notice some annoying distortions. Moreover, the eye (marked in red box) looks more natural than that obtained by other methods.

5. Conclusion

Image prior is significant to regularization methods for image deblurring. But due to the great variety of images, the exact image prior is not available. By utilizing existing image smoothing operators, the method noise provides a new way to create image regularizers. In this work, we propose a weighted-l_1-method noise regularization model for image deblurring. The proposed method has two-fold advantages: the l_1 penalty of the method-noise is better than the l_2 penalty in removing noise in smooth areas; The incorporated gradient-based weight can better preserve image edges. The advantages are verified by experimental results. In our paper, we only considered the deblurring problem with a given convolution kernel. In case that the convolution kernel is unknown, how to incorporate estimation of the convolution kernel in our method will be explored in our future work.

References

[1] H. Yang, Z.B. Zhang, Y.J. Guan, An adaptive parameter estimation for guided filter based image deconvolution, Signal Process. 138 (2017) 16–26.
[2] R.H. Chan, M. Tao, X. Yuan, Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers, SIAM J. Imaging Sci. 6 (1) (2013).
[3] S. Tang, W. Gong, W. Li, et al., Non-blind image deblurring method by local and nonlocal total variation models, Signal Process. 94 (2014) 339–349.
[4] H. Zhang, L. Tang, Z. Fang, et al., Nonconvex and nonsmooth total generalized variation model for image restoration, Signal Process. 143 (2018) 69–85.
[5] L. Xu, S. Zheng, J. Jia, Unnatural l_0 sparse representation for natural image deblurring, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[6] M. Kang, M. Jung, M. Kang, Rician denosing and deblurring using sparse representation prior and nonconvex total variation, J. Vis. Commun. Image Represent. 54 (2018) 80–89.
[7] A. Buades, B. Coll, J.M. Morel, Image enhancement by non-local reverse heat equation, Preprint CMLA, vol 22, 2006.
[8] A. Buades, B. Coll, J.M. Morel, A non-local algorithm for image denoising, in: IEEE Conference on Computer Vision and Pattern Recognition, 2005.
[9] Z.M. Wang, H. Bao, A new regularization model based on non-local means for image deblurring, Appl. Mech. Mater. Trans Tech Publications 411 (2013) 1164–1169.
[10] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (4) (1992) 259–268.
[11] S. Parameswaran, E. Luo, T. Nguyen, Patch matching for image denoising using neighborhood-based collaborative filtering, IEEE Trans. Circuits Syst. Video Technol. 99 (2016) 156–168.
[12] K. Dabov, A. Foi, V. Katkovnik, Image denoising by sparse 3-d transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[13] R. Zhang, X. Feng, L. Yang, et al., Global sparse gradient guided variational retinex model for image enhancement, Signal Process. Image Commun. 58 (2017) 270–281.
[14] J. Xu, L. Wang, Z. Shi, A switching weighted vector median filter based on edge detection, Signal Process. 98 (2014) 359–469.
[15] X. Yang, M. Liu, J. Xu, et al., Image segmentation and recognition algorithm of greenhouse whitefly and thrip adults for automatic monitoring device, Trans. Chin. Soc. Agric. Eng. 34 (1) (2018) 164–170.
[16] Y. Zhang, X. Han, H. Zhang, et al., Edge detection algorithm of image fusion based on improved sobel operator, in: Information Technology and Mechatronics Engineering Conference, IEEE, 2017.
[17] U.S. 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.
[18] S.H. Chan, X. Wang, O.A. Elgendy, Plug-and-play ADMM for image restoration: fixed-point convergence and applications, IEEE Trans. Comput. Imaging 3 (1) (2017) 84–98.
[19] X. Zhang, M. Burger, X. Bresson, et al., Bregmanized nonlocal regularization for deconvolution and sparse reconstruction, SIAM J. Imaging Sci. 3 (3) (2010) 253–276.
[20] Q. Cheng, H. Shen, L. Zhang, et al., Inpainting for remotely sensed images with a multichannel nonlocal total variation model, IEEE Trans. Geosci. Remote Sens. 52 (1) (2014) 175–187.
[21] T. Liu, T.K. Pong, Further properties of the forward-backward envelope with applications to difference-of-convex programming, Comput. Optim. Appl. 67 (3) (2017) 489–520.
[22] P.Y. Chen, I.W. Selesnick, Translation-invariant shrinkage/thresholding of group sparse signals, Signal Process. 94 (1) (2014) 476–489.
[23] M. Vauhkonen, D. Vadasz, P.A. Karjalainen, et al., Tikhonov regularization and prior information in electrical impedance tomography, IEEE Trans. Med. Imaging 17 (2) (1998) 285–293.
[24] J. Zhang, D. Zhao, W. Gao, Group-based sparse representation for image restoration, IEEE Trans. Image Process. 23 (8) (2014) 3336–3351.
[25] W. Dong, et al., Nonlocally centralized sparse representation for image restoration, IEEE Trans. Image Process. 22 (4) (2013) 1620–1630.
[26] A. Danielyan, V. Katkovnik, K. Egiazarian, BM3d frames and variational image deblurring, IEEE Trans. Image Process. 21 (4) (2012) 1715–1728.