Framelet-Based image restoration for impulse noise removal problem

Xiao-Juan Yang †[1]

† School of Mathematics, Nanjing Normal University Taizhou College, Taizhou 225300, P. R. China

Abstract

We consider the image recovery problem where the observed image is simultaneously corrupted by blur and impulse noise. Due to the edge preserving property of the total variation (TV), and the property of providing good sparse approximation to piecewise smooth functions of the wavelet frames, we propose a Framelet-based hybrid regularization model to significantly lessen staircase artifacts while well preserving the valuable edge information of the image. We take advantage of an alternating direction method of multiplier (ADMM) to efficiently find a solution of this model. Because of the convex of our model, the convergence of our method can be guaranteed. Experimental results are finally presented to show the efficiency of our method in terms of the peak-signal-to-noise ratio (PSNR), structure similarity index measure (SSIM) and the relative error (ReErr).

Keywords: total variation, impulse noise, alternating direction method of multiplier, Framelet.

1 Introduction

Image restoration plays an important role in many applications whose aim is to recover a clean image from the degraded observation. In this paper, we restrict our attention to image deblurring under impulse noise. There are two common types of impulse noise: salt-and-pepper noise, corrupting a portion of all pixels with minimal or maximal intensities and leaving the remainder unaffected, and random-valued noise, the same as salt-and-pepper noise except that corrupted pixels have random intensity values between minimal and maximal ones.

Assume that $u \in \mathbb{R}^{n^2}$ is the true image with pixel value in $[0, 255]$ and $H : \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$ is a blurring operator. The observed corrupted image $g \in \mathbb{R}^{n^2}$ can be formatted as

$$g = N_{imp}(Hu)$$

where N_{imp} represents the degradation by impulse noise.

The earlier technique for removing impulse noise is a median filter [33], which is efficient and easy to implement. Some modified versions of the median filter are given in [3, 26, 32]. Since these filters accomplish the filtering task via replacing each pixel in the image by average value around it, they cannot preserve image edges well.

There are some regularization methods that preserve the edge information in the restored solution [11]. Bar, Sochen, and Kiryati [1,2] proposed a model which is composed of the Mumford-Shah regularizer and the modified l^1 data-fidelity term to deblur the image with impulse noise. Nikolova [29] proposed a variational model which combines an l^1 data-fidelity term and an edge-preserving regularization term.

Among the edge-preserving regularization techniques, the most commonly used approach, the total variation (TV) regularization [34], which measures the l^1 norm of the magnitude of the gradient (the first-order difference) of the image. The advantage of using the TV regularization

[1] Corresponding Author, email:yxjwyz@163.com

is that edges of the image can be well prevered. Yang, Zhang, and Yin [41] proposed an efficient algorithm (FTVd) for solving a TV-l^1 model

$$\min_u \{\|Hu - g\|_1 + \lambda \|u\|_{TV}\} \tag{1}$$

which combines TV regularization and l^1 data-fidelity term, where$\|\cdot\|_{TV}$ is the discrete TV regularization term. The discrete gradient operator $\nabla : \mathbb{R}^{n^2} \to \mathbb{R}^{n^2}$ with periodic boundary condition is defined by

$$(\nabla u)_{j,k} = ((\nabla u)^x_{j,k}, (\nabla u)^y_{j,k})$$

with

$$(\nabla u)^x_{j,k} = \begin{cases} u_{j+1,k} - u_{j,k} & \text{if } 1 \le j < n, \\ u_{j,1} - u_{j,n} & \text{if } j = n, \end{cases}$$

$$(\nabla u)^y_{j,k} = \begin{cases} u_{j,k+1} - u_{j,k} & \text{if } 1 \le k < n, \\ u_{1,k} - u_{n,k} & \text{if } k = n, \end{cases}$$

for $j, k = 1, \cdots, n$. Here $u_{j,k}$ refers to the $((k-1)n + j)$th entry of the vector u (it is the (j, k)th pixel location of the image). The discrete total variation of u is defined by

$$\|u\|_{TV} = \sum_{1 \le j,k \le n} \sqrt{|(\nabla u)^x_{j,k})|^2 + |(\nabla u)^y_{j,k})|^2}. \tag{2}$$

The definition in (2) is often referred to as isotropic total variation. The anisotropic total variation is defined as follows

$$\|u\|_{TV} = \sum_{1 \le j,k \le n} |(\nabla u)^x_{j,k})| + |(\nabla u)^y_{j,k})|.$$

Dong et al. [20] used a primal-dual approach for the model (1). To reduce the computational cost, Wu et al. [37] introduced the augmented Lagrangian method to solve this model at the cost of lower restored quality.

We known that staircase artifacts may appear in the reconstructed image with the TV regularization. One approach reducing the staircase effect is adding a higher order regularization term to the original ROF model by [10]. The other is based on wavelet frames. It is well known that images, especially natural images, can be regarded as piecewise smooth functions, and wavelet frames can usually provide good sparse approximations to piecewise smooth functions. The ability to approximate images sparsely is an important characteristic of wavelets, see [28]. Therefore, the framelet-based models can significantly improve image restoration quality [4–6,12]. Dong et al. [19] employed the framelet-based regularization. Li et al. [27] combined a framelet-based regularization to deblurred images with mixed Gaussian-impulse noise.

Our method is related to the sparse approximation of images by wavelets. There are mainly three formulations utilizing the sparseness of the wavelet frame coefficients, namely analysis based approach, synthesis based approach, and balanced approach. Chen [7] combine the synthesis based approach and TV regularization to create a model for image deblurring under impulse noise. In our work, we combine the ideas behind TV regularization and framelet in a different way. We will focus on the analysis approach because it provides a direct link to the local geometry of u [8]. The analysis approach is often modeled as a regularization term as follows

$$R(u) = \|Wu\|_1,$$

where W is the analysis operator and Wu is the corresponding wavelet frame coefficients. We combine framelet and TV regularization together to propose the following model

$$\min_{u}\{\|Hu-g\|_1+\alpha\|u\|_{TV}+\beta\|Wu\|_1\}, \tag{3}$$

Throughout this paper, we only focus on the use of the isotropic formulation of TV. Actually, it is also suitable for anisotropic TV. For ease of notation, we denote the isotropic formulation of TV as follow

$$\|u\|_{TV}:=\sum\|u\|_2=\sum\sqrt{|(\nabla u)^x|^2+|(\nabla u)^y|^2}, \tag{4}$$

where Σ denotes the summation taken over all pixels.

The computational challenge of (3) comes mainly from the non-differentiability of the TV norm and l_1 regularized terms. In order to solve the compound regularization model, we use the alternating direction method of multipliers (ADMM) [35, 38, 39, 41]. Experimental results demonstrate the efficiency of our method for preserving both the structure details and the sharp edges.

The paper is organized as follows. In Section 2 we give some preliminaries of framelets and examples of framelets used in this paper. In Section 3, we present the detailed algorithm for the model (3). Some numerical experiments are given to illustrate the performance of the proposed algorithm in Section 4.

2 Framelets

In this section, we present some preliminaries of tight framelets. For simplicity, we only present the univariate framelets and the framelets for two variables can be constructed by tensor product of univariate framelets. As for the details about how to construct the tight framelets and the detailed frame theory, the interested reader can find further details in [12, 13, 18].

A system $X\subset L_2(\mathbb{R})$ is called a tight frame of $L_2(\mathbb{R})$ if

$$f=\sum_{g\in X}<f,g>g,\quad \forall f\in L_2(\mathbb{R}). \tag{5}$$

This is equivalent to

$$\|f\|_2^2=\sum_{g\in X}|<f,g>|^2\quad \forall f\in L_2(\mathbb{R}), \tag{6}$$

where $<\cdot,\cdot>$ and $\|\cdot\|_2=<\cdot,\cdot>^{\frac{1}{2}}$ are the inner product and norm of $L_2(\mathbb{R})$. We can see that an orthonormal basis is a tight frame, since the identities (5) and (6) hold for arbitrary orthonormal bases in $L_2(\mathbb{R})$. Hence tight frames are generalization of orthonormal bases that bring in the redundancy which is often useful in applications such as denoising, see, e.g., [17].

For given $\Psi:=\{\psi_1,\cdots,\psi_q\}\subset L_2(\mathbb{R})$, the corresponding wavelet (or affine) system $X(\Psi)$ generated by Ψ is defined by the collection of dilations and shifts of Ψ as

$$X(\Psi)=\{2^{k/2}\psi_l(2^kx-j):1\leq l\leq q;k,j\in Z\}.$$

When $X(\Psi)$ forms a tight frame of $L_2(\mathbb{R})$, each function $\psi_l, l=1,\cdots,q$, is called a (tight) framelet and the whole $X(\Psi)$ is called a tight wavelet frame.

The construction of framelet Ψ, which are desirably (anti)symmetric and compactly supported functions, are usually based on a multiresolution analysis (MRA) that is generated by some refinable function ϕ with refinement mask a_0 satisfying the refinement equation

$$\hat{\phi}(2\cdot) = \hat{a}_0 \hat{\phi}(\cdot),$$

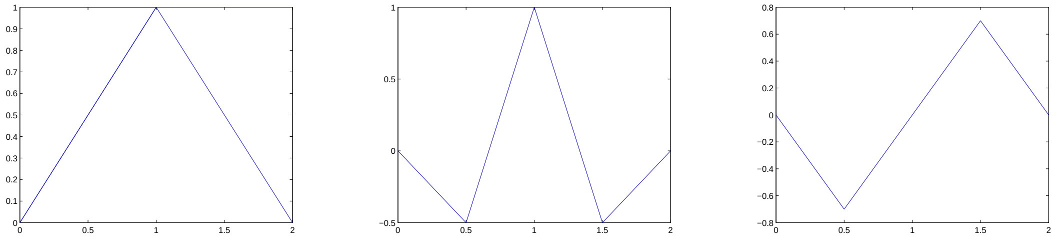
where $\hat{\phi}$ is the Fourier transform of Ψ, and $\hat{a}_0$ which is 2π periodic is the Fourier transform of a_0. The idea of an MRA-based construction of framelet Ψ is to find masks a_l, which are finite sequence, such that

$$\hat{\psi}_l(2\cdot) = \hat{a}_l \hat{\phi}(\cdot), \quad l = 1, \cdots, q,$$

where $\hat{a}_l$ which is 2π periodic is the Fourier transform of a_l. The sequence $a_1, \cdots, a_q$ are called wavelet frame masks, or the high filters of the system, and the refinement mask a_0 is called low pass filter. The unitary extension principle (UEP) of [36] asserts that the system $X(\Psi)$ generated by a finite set Ψ forms a tight frame in $L_2(\mathbb{R})$ provided that the masks $a_0, a_1, \cdots, a_q$ are finitely supported and their Fourier series satisfy

$$\sum_{l=0}^{q} |\hat{a}_l(\xi)|^2 = 1 \quad \text{and} \quad \sum_{l=0}^{q} \hat{a}_l(\xi)\overline{\hat{a}_l(\xi+\nu)} = 0,$$

for all $\nu \in \{0, \pi\} \backslash \{0\}$ and $\xi \in [-\pi, \pi]$., the framelets we adopt in this paper are constructed from piecewise linear B-spline first given in [36]. The refinement mask is $\hat{a}_0(\xi) = \cos^2(\xi/2)$, whose corresponding lowpass filter is $\hat{a}_0 = \frac{1}{4}[1, 2, 1]$. Those two framelet masks are $\hat{a}_1(\xi) = -\frac{\sqrt{2}i}{2}\sin^2(\xi/2)$ and $\hat{a}_2(\xi) = \sin^2(\xi/2)$ whose corresponding lowpass filters are $\hat{a}_1 = \frac{\sqrt{2}}{4}[1, 0, -1]$ and $\hat{a}_2 = \frac{1}{4}[-1, 2, -1]$, respectively. we can see that the masks a_1 and a_2 correspond to the first order and second order diffreence operators respectively up to a scaling. The associated refinable function and framelets are given in Fig.1



(a) ϕ (b) ψ_1 (c) ψ_2

Fig.1. Piecewise linear framelets.

The numerical computation of the wavelet framelet transform is done by using the WF decomposition algorithm given in [21]. We conduct the decomposition algorithm without downsampling by means of refinement and frmelet masks. We denote the fast framelet transform as W and the inverse framelet transform as W^T, which is the adjoint operator of W. The frame coefficients of u can be computed by $x = Wu$, and we will have the perfect reconstruction formula $u = W^T x$, i.e., $u = W^T W u$. Hence W is a tight frame if and only if $W^T W = I$, where I is the identity matrix. Unlike the orthonormal case, we emphasize that $WW^T \neq I$ in general.

3 Numerical Algorithm

As the proposed model is a convex optimization problem, there are efficient solvers for finding a global minimizer of (3). The solvers include the Bregman method [23], Proximal splitting methods [15], primal-dual methods [16, 22], and alternating direction method of multipliers [24]. Here we just adopt the alternating direction method of multipliers.

We begin by reformulating (3) as the following constrained optimization problem

$$\min_u\{\|z\|_1 + \alpha \sum \|x\|_2 + \beta\|y\|_1\}, \quad \text{subject to} \quad Hu - g = z, \nabla u = x, Wu = y \tag{7}$$

The resulting augmented Lagrangian function is given by

$$\begin{aligned} L(z, x, y, u, \lambda_1, \lambda_2, \lambda_3) = \|z\|_1 + \alpha \sum \|x\|_2 + \beta\|y\|_1 \ &+ \frac{\gamma_1}{2}\|z - (Hu - g)\|_2^2 - \lambda_1^T(z - (Hu - g)) \\ + \frac{\gamma_2}{2}\|x - \nabla u\|_2^2 - \lambda_2^T(x - \nabla u) &+ \frac{\gamma_3}{2}\|y - Wu\|_2^2 - \lambda_3^T(y - Wu) \end{aligned} \tag{8}$$

where λ_1, λ_2, and λ_3 are the vectors corresponding to the Lagrange multipliers to the linear constraints, and γ_1, γ_2, and γ_3 are the penalty parameter of the alternating direction method of multipliers to control the speed of the convergence; see, for instance, [30]. The ADMM iteration associated to the minimizing L in (8) with respect to (z, x, y, u) and with the updates of the multipliers λ_1, λ_2, and λ_3 is described as follows

$$\begin{cases} (z^{k+1}, x^{k+1}, y^{k+1}, u^{k+1}) = \arg\min_{z,x,y,u} L(z, x, y, u, \lambda_1^k, \lambda_2^k, \lambda_3^k), \\ \lambda_1^{k+1} = \lambda_1^k - \xi\gamma_1(z^{k+1} - (Hu^{k+1} - g)), \\ \lambda_2^{k+1} = \lambda_2^k - \xi\gamma_2(u^{k+1} - \nabla u^{k+1}), \\ \lambda_3^{k+1} = \lambda_3^k - \xi\gamma_3(y^{k+1} - Wu^{k+1}). \end{cases} \tag{9}$$

Since the variables z, x, y, and u are decoupled, this allows us to solve them more easily on their corresponding subproblems in the ADMM. We now investigate these subproblems one by one for the impulsive noise removal problem.

In step 1, we solve for the variable z in the subproblem. This subproblem corresponds to the following optimization problem

$$\min_z\{\|z\|_1 + \frac{\gamma_1}{2}\|z - (Hu^k - g)\|_2^2 - (\lambda_1^k)^T(z - (Hu^k - g))\}. \tag{10}$$

We can directly obtain the solution of (10) by using one-dimensional shrinkage operator

$$z^{k+1} = \text{sgn}(Hu^k - g + \frac{\lambda_1^k}{\gamma_1}) \circ \max\{|Hu^k - g + \frac{\lambda_1^k}{\gamma_1}| - \frac{1}{\gamma_1}, 0\}, \tag{11}$$

where sign refers to 1 if the entry is greater than or equal to zero, and -1 if the entry is negative, and $\circ$ represents the point-wise product.

In step 2, we solve for the variable x in the subproblem. This subproblem corresponds to the following optimization problem

$$\min_x\{\alpha \sum \|x\|_2 + \frac{\gamma_2}{2}\|x - \nabla u^k\|_2^2 - (\lambda_2^k)^T(x - \nabla u^k)\}. \tag{12}$$

The minimizer of the subproblem (12) can be obtained by using two-dimensional shrinkage formula

$$x^{k+1} = \frac{\nabla u^k + \frac{\lambda_2^k}{\gamma_2}}{\|\nabla u^k + \frac{\lambda_2^k}{\gamma_2}\|} \max\{\|\nabla u^k + \frac{\lambda_2^k}{\gamma_2}\| - \frac{\alpha}{\gamma_2}, 0\} \tag{13}$$

where the convention $0 \cdot (0/0) = 0$ is followed.

In step 3, we solve for the variable y in the subproblem. This subproblem corresponds to the following optimization problem

$$\min_y \{\beta\|y\|_1 + \frac{\gamma_3}{2}\|y - Wu^k\|_2^2 - (\lambda_3^k)^T(y - Wu^k)\}. \tag{14}$$

Similarity, the minimizer of the subproblem (14) can be obtained by using two-dimensional shrinkage formula

$$y^{k+1} = \text{sgn}(Wu^k + \frac{\lambda_3^k}{\gamma_3}) \circ \max\{\|Wu^k + \frac{\lambda_3^k}{\gamma_3}\| - \frac{\beta}{\gamma_3}, 0\} \tag{15}$$

In step 4, we solve for the variable u in the subproblem. This subproblem corresponds to the following optimization problem

$$\begin{aligned} &\min_u \{\frac{\gamma_1}{2}\|z^{k+1} - (Hu - g)\|_2^2 - (\lambda_1^k)^T(z^{k+1} - (Hu - g)) \\ &+ \frac{\gamma_2}{2}\|x^{k+1} - \nabla u\|_2^2 - (\lambda_2^k)^T(x^{k+1} - \nabla u) + \frac{\gamma_3}{2}\|y^{k+1} - Wu\|_2^2 - (\lambda_3^k)^T(y^{k+1} - Wu)\}. \end{aligned} \tag{16}$$

The minimizer can be obtained by equivalently solving a linear system

$$\begin{aligned} &(\gamma_1 H^T H + \gamma_2 \nabla^T \nabla + \gamma_3 I)u \\ &= \gamma_1 H^T(g + z^{k+1}) - H^T\lambda_1^k + \gamma_2 \nabla^T x^{k+1} - \nabla^T \lambda_2^k + \gamma_3 W^T y^{k+1} - W^T \lambda_3^k \end{aligned} \tag{17}$$

where we have used $W^T W = I$.

Note that H and ∇ have block circulant with circulant blocks (BCCB) structure when periodic boundary conditions are used [31, 38, 40]. We know that the computations with BCCB matrices can be very efficient by using fast Fourier transforms (FFTs). We can write

$$u^{k+1} = \mathcal{F}^{-1}(\zeta) \tag{18}$$

with

$$\zeta = \frac{\gamma_1 \mathcal{F}(H^T)(\mathcal{F}(g) + \mathcal{F}(z^{k+1}) - \mathcal{F}(\lambda_1^k)) + \gamma_2 \mathcal{F}(\nabla^T)(\mathcal{F}(x^{k+1}) - \mathcal{F}(\lambda_2^k)) + \gamma_3 \mathcal{F}(W^T)(\mathcal{F}(y) - \mathcal{F}(\lambda_3^k))}{\gamma_1 \mathcal{F}(H^T)\mathcal{F}(H) + \gamma_2 \mathcal{F}(\nabla^T)\mathcal{F}(\nabla) + \gamma_3 I}. \tag{19}$$

Based on the above analysis, we obtain the following algorithm for solving the impulse noise removal problem

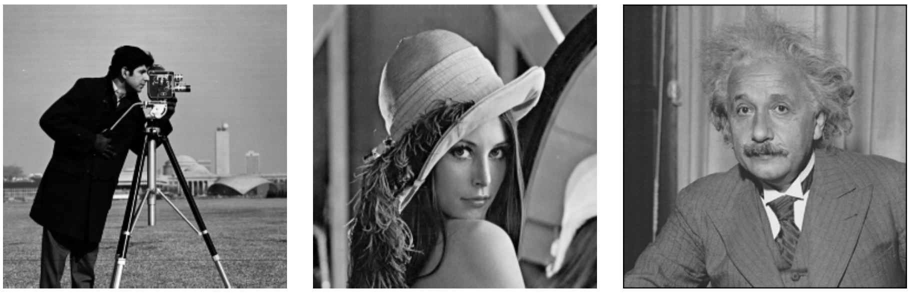
Algorithm 1 ADMM for the problem (3)

```
1. Input: $g, H, \alpha > 0, \beta > 0, \epsilon > 0, \lambda_1 > 0, \lambda_2 > 0$, and $\lambda_3 > 0$.
2. Initialization:$u^0 = g, x^0 = \nabla u^0, y^0 = Wu^0, \lambda_1 = \lambda_2 = \lambda_3 = 0$.
3. For $k = 0, 1, \cdots$, do:
4. Compute $(z^{k+1}, x^{k+1}, y^{k+1}, u^{k+1})$ as an minimizer of the Lagrangian function
   with Lagrange multipliers $\lambda_1, \lambda_2$ and $\lambda_3$, i.e.,
   compute $z$: $z^{k+1} = \mathrm{sgn}(Hu^k - g + \frac{\lambda_1^k}{\gamma_1}) \circ \max\{|Hu^k - g + \frac{\lambda_1^k}{\gamma_1}| - \frac{1}{\gamma_1}, 0\}$;
   compute $x$: $x^{k+1} = \frac{\nabla u^k + \frac{\lambda_2^k}{\gamma_2}}{\|\nabla u^k + \frac{\lambda_2^k}{\gamma_2}\|} \max\{\|\nabla u^k + \frac{\lambda_2^k}{\gamma_2}\| - \frac{\alpha}{\gamma_2}, 0\}$;
   compute $y$: $y^{k+1} = \mathrm{sgn}(Wu^k + \frac{\lambda_3^k}{\gamma_3}) \circ \max\{\|Wu^k + \frac{\lambda_3^k}{\gamma_3}\| - \frac{\beta}{\gamma_3}, 0\}$;
   compute $u$: $u^{k+1} = \mathcal{F}^{-1}(\zeta)$ with $\zeta$ in (19);
   update $\lambda_1$: $\lambda_1^{k+1} = \lambda_1^k - \xi\gamma_1(z^{k+1} - (Hu^{k+1} - g))$,
   update $\lambda_2$: $\lambda_2^{k+1} = \lambda_2^k - \xi\gamma_2(u^{k+1} - \nabla u^{k+1})$;
   update $\lambda_3$: $\lambda_3^{k+1} = \lambda_3^k - \xi\gamma_3(y^{k+1} - Wu^{k+1})$.
     if $\|u^k - u^{k-1}\|/\|u^{k-1}\| < \epsilon$;
         return
     End if
5. End For
6. Output $u = u^{k+1}$.
```

In the Algorithm 1, $\epsilon > 0$ is a given tolerance.

4 Numerical experiments

In this section, we illustrate the performance of the proposed algorithm and compare it with the FTVd algorithm [41]. We use three test images of size $n \times m$: Cameraman (256×256), Lena (256×256) and Einstein (256×256), which are shown in Figure 2, and the pixels of all images are scaled to between 0 and 255. In our experiments, we consider two different blurring operators: a 7×7 Gaussian blur with standard deviation 5 and a average blur 9 (generated by the MATLAB function fspecial). We generate the blurry image with periodic boundary condition and corrupt the blurry image with impulse noise. As observed in [41], the influence of boundary conditions on



(a) Cameraman (b) Lena (c) Einstein

Fig.2. Original images.

image quality is negligible since the sizes of the tested images are much larger than those of the tested blurring kernels. For much discussions about boundary conditions, see [31].

The quality of the restored images is measured by Peak-signal-to-noise ratio (PSNR), Structural similarity index (SSIM) and the Relative error (ReErr). They are defined as follows:

$$\mathrm{PSNR} = 20\log_{10}\frac{255}{\frac{1}{mn}\|u-f^*\|_2},$$

$$\mathrm{SSIM} = \frac{(2\mu_f^*\mu_u + C_1)(2\sigma_{f^*u} + C_2)}{(\mu_{f^*}^2 + \mu_u^2 + C_1)(\sigma_{f^*}^2 + \sigma_u^2 + C_2)}.$$

$$\mathrm{ReErr} = \tfrac{\|u-f^*\|_2}{\|f^*\|_2},$$

Where f^* is the orginal image, and u is the restored image. μ_{f^*} and μ_u are averages of f^* and u respectively, σ_{f^*} and σ_u are the variance of f^* and u respectively, σ_{f^*u} is the covariance of f^* and u and the positive constants C_1 and C_2 can be thought of as stabilizing constants for near-zero denominator values. In general, a high PSNR-value or a small ReErr-value indicates the restoration is more accurate. However, we know that the PSNR-value or a small ReErr-value are not always in agreement with visual perception. The SSIM index is a well-known quality metric used to measure the similarity between two images. The SSIM map is whiter, the restored image is closer to the clean image.

The stopping criterion is the relative difference successively (RDS) of the restored image, which is commonly used in [14, 25, 41] and is defined by

$$\mathrm{RDS}(u^{k+1}, u^k) = \tfrac{\|u^{k+1}-u^k\|_2}{\|u^{k+1}\|_2}.$$

We stop all methods when $RDS < 1 \times 10^{-3}$.

All the experiments were performed using MATLAB 7.7.0 on a computer equipped with an Intel (R) Core (TM) 2.60 GHz processor, with 4.00 GB of RAM, and running Windows XP.

4.1 Deblurring image with salt-pepper noise

In the first experiment, the image is blurred by a 7 × 7 Gaussian blur with standard deviation 5 and then corrupt with salt-and pepper noise with noise levels $20\%, 30\%, 40\%, 50\%$ and 60%.

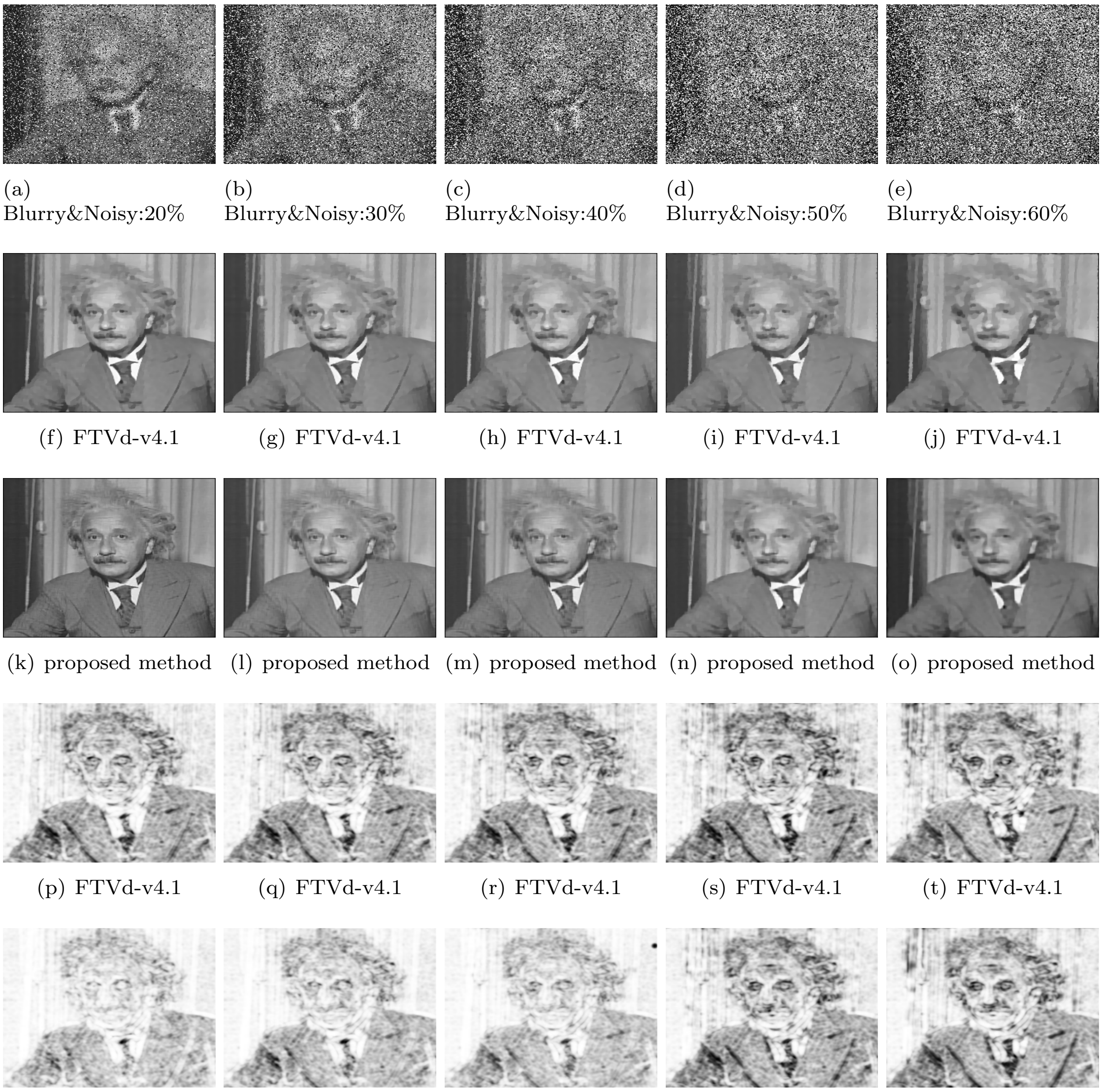
There are three parameters and five parameters in the algorithm FTVd and the proposed algorithm respectively. In order to have fair comparisons, we determine the best parameters such that the SSIM index is biggest. The FTVd algorithm used in this paper for comparison is FTVd_v4.1, which is the latest version for FTVd. For the noise levels $20\%, 30\%, 40\%, 50\%$ and 60%, the parameters (μ, β_1, β_2) are tuned to get the best results, and they are set to be $(13, 1, 20), (13, 1, 20), (13, 1, 20), (13, 1, 20), (8, 1, 20)$ for the Einstein image, $(25, 1, 20), (25, 1, 20),$ $(13, 1, 20), (13, 5, 20), (10, 5, 20)$ for the Lena image and $(25, 1, 20), (25, 1, 20), (25, 1, 20), (25, 1, 20),$ $(10, 1, 20)$ for the Cameraman image respectively. For Algorithm 1, we set $(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3)$ to be $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.003, 0.008, 5, 0.05, 0.15)$, $(0.05, 0.009,$ $15, 0.05, 0.05)$, $(0.05, 0.02, 15, 0.15, 0.05)$ for the Einstein image, $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.05, 0.009,$ $15, 0.15, 0.15)$, $(0.05, 0.009, 15, 0.15, 0.15)$, $(0.05, 0.008, 15, 0.15, 0.15)$, $(0.03, 0.02, 5, 0.05, 0.05)$ for the Lena image, and $(0.003, 0.008, 15, 0.15, 0.15)$, $(0.01, 0.009, 15, 0.05, 0.15)$, $(0.003, 0.02, 15, 0.15, 0.15)$,

$(0.03, 0.02, 5, 0.15, 0.15)$, $(0.03, 0.02, 10, 0.05, 0.05)$ for the Cameraman for the noise levels $20\%, 30\%, 40\%, 50\%$ and 60% respectively.The restored images by all two methods are shown in Figs 3-5(f)-(o) and the SSIM maps of the restored images in Figs 3-5(p)-(y).

Table 1
PSNR(dB) values, SSIM and ReErr for various methods for the test images corrupted by salt-and-pepper noise levels from 20% to 60%, blurred by Gaussian kernel of size 7×7, and standard deviation 5.

Images	Noise levels	FTVd			Proposed method		
		PSNR	SSIM	ReErr	PSNR	SSIM	ReErr
Einstein	20%	31.184	0.819	0.057	34.332	0.902	0.041
	30%	30.916	0.812	0.059	32.737	0.878	0.048
	40%	30.337	0.800	0.063	31.715	0.845	0.058
	50%	29.404	0.783	0.070	30.702	0.835	0.063
	60%	26.415	0.745	0.099	28.333	0.786	0.082
Lena	20%	28.776	0.867	0.077	32.288	0.934	0.052
	30%	28.398	0.857	0.081	29.302	0.885	0.073
	40%	27.946	0.847	0.085	28.715	0.871	0.078
	50%	26.925	0.818	0.096	27.899	0.853	0.085
	60%	25.986	0.792	0.107	26.691	0.818	0.104
Cameraman	20%	29.952	0.869	0.074	32.804	0.937	0.048
	30%	28.125	0.857	0.081	31.833	0.914	0.059
	40%	27.952	0.842	0.090	30.162	0.883	0.082
	50%	27.042	0.817	0.102	28.530	0.834	0.095
	60%	24.792	0.786	0.114	23.915	0.781	0.120

In the second experiment, the image is blurred by a Average blur 9 and then corrupt with salt-and pepper noise with noise levels $20\%, 30\%, 40\%, 50\%$ and 60%. For all the noise levels, the parameters (μ, β_1, β_2) in FTVd_v4.1 are set to be $(13, 1, 20)$ for the Einstein image, $(25, 120), (25, 1, 20), (25, 1, 20), (13, 1, 20), (13, 1, 20)$ for the Lena image, and $(25, 1, 20)$ for the Cameraman image. For the proposed Algorithm 1, we set $(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3)$ to be $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.01, 0.008, 15, 0.15, 0.15)$, $(0.006, 0.02, 15, 0.05, 0.15)$, $(0.01, 0.009, 10, 0.15, 0.05)$ for the Einstein image, $(0.01, 0.009, 15, 0.15, 0.15)$, $(0.01, 0.009, 15, 0.15, 0.15)$, $(0.01, 0.009, 15, 0.15, 0.15)$, $(0.003, 0.008, 15, 0.15, 0.05)$, $(0.006, 0.009, 10, 0.05, 0.05)$ for the Lena image and $(0.003, 0.009, 15, 0.15, 0.15)$, $(0.003, 0.009, 15, 0.15, 0.15)$, $(0.005, 0.02, 10, 0.15, 0.15)$, $(0.009, 0.02, 10, 0.15, 0.15)$, $(0.005, 0.009, 5, 0.15, 0.05)$ for the Cameraman image for the noise levels $20\%, 30\%, 40\%, 50\%$ and 60% respectively. The restored images by all two methods are shown in Figs 6-8(f)-(o) and the SSIM maps of the restored images in Figs 6-8(p)-(y).



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

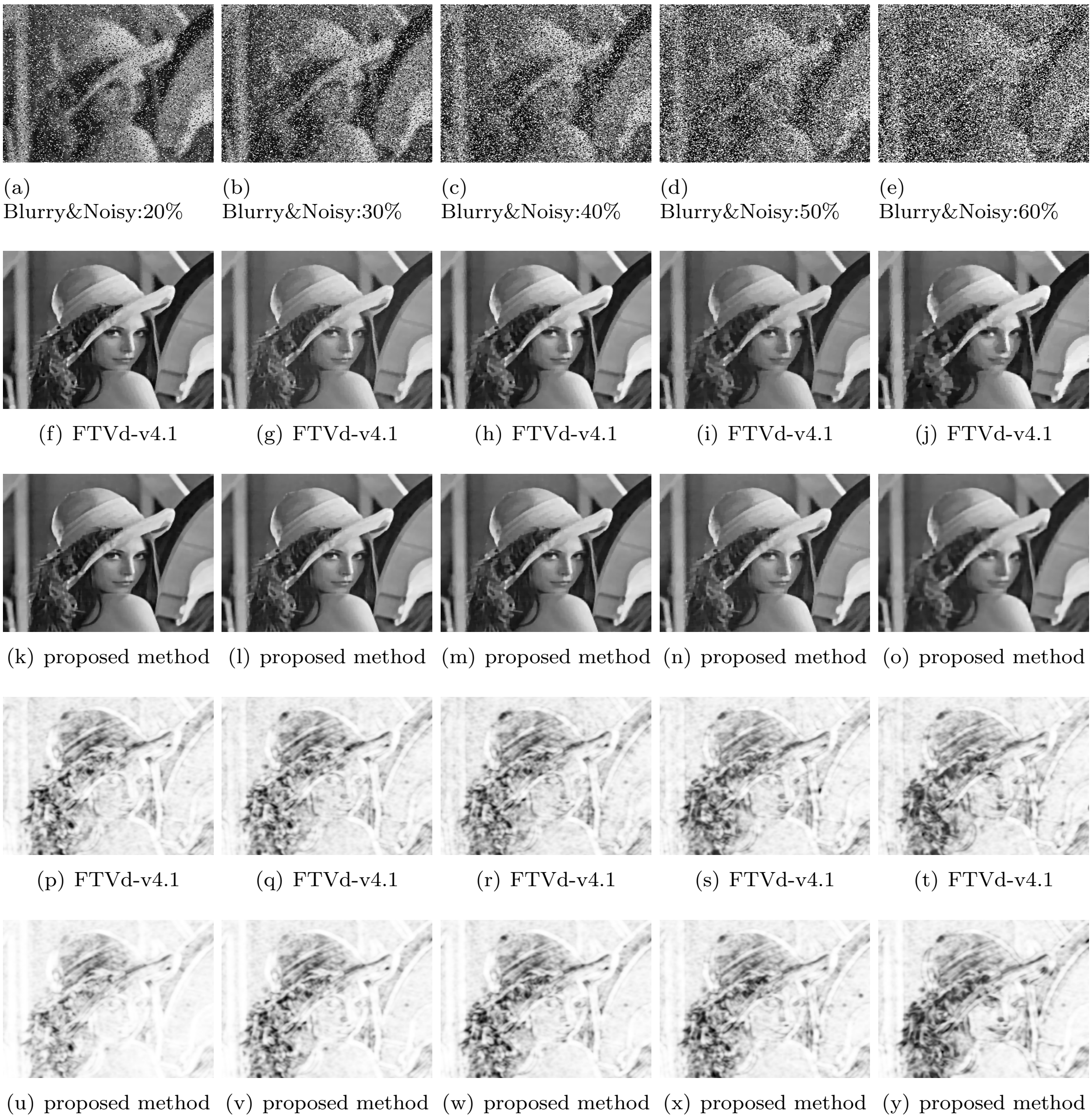
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.3. Recovered images of different methods on the image Einstein corrupted by Gaussian blur $G([7,7],5)$ and salt-and-pepper noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

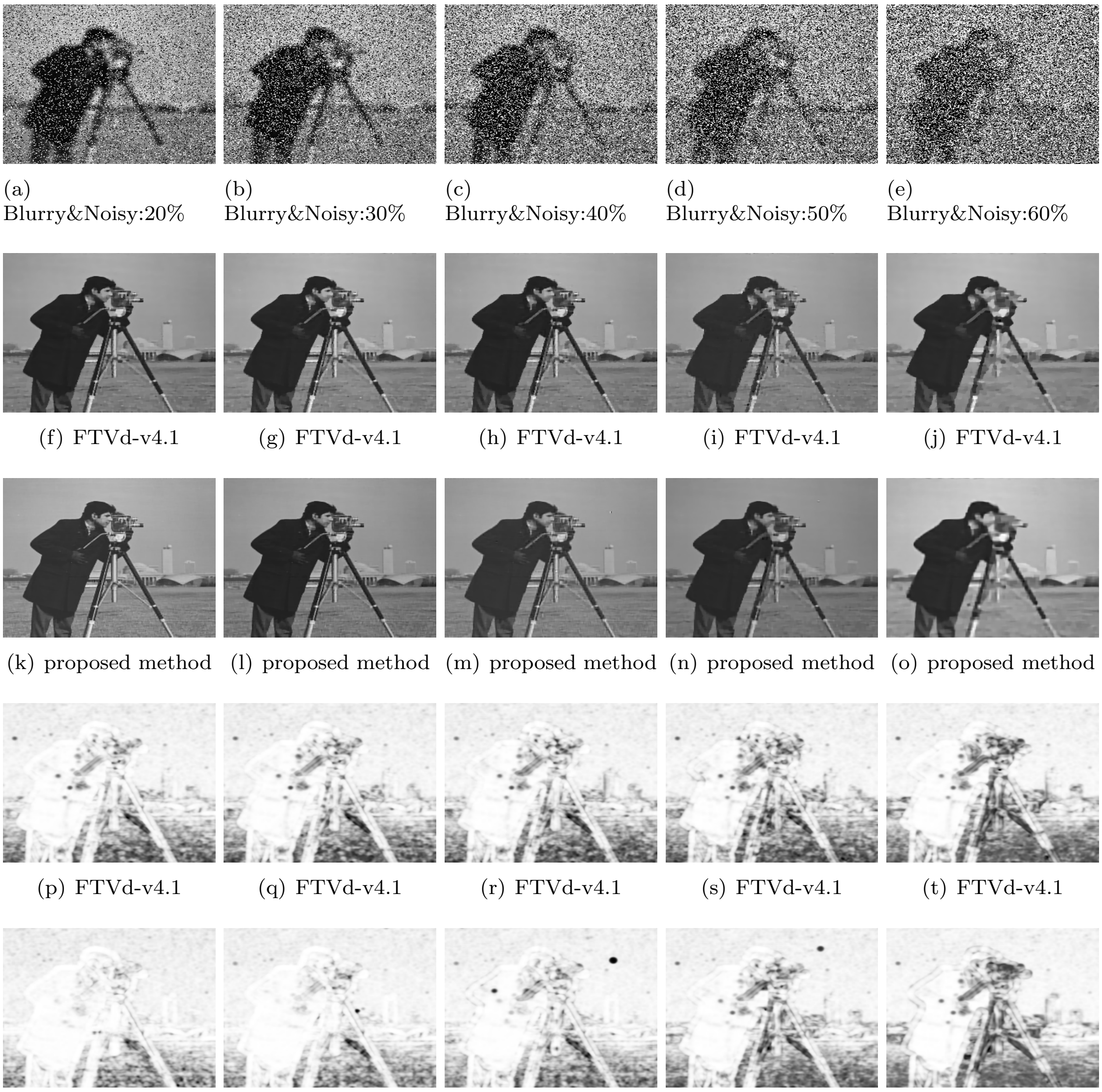
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.4. Recovered images of different methods on the image Lena corrupted by Gaussian blur $G([7,7],5)$ and salt-and-pepper noise with noise levels from 20% to 60%..



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.5. Recovered images of different methods on the image Cameraman corrupted by Gaussian blur $G([7,7],5)$ and salt-and-pepper noise with noise levels from 20% to 60%.

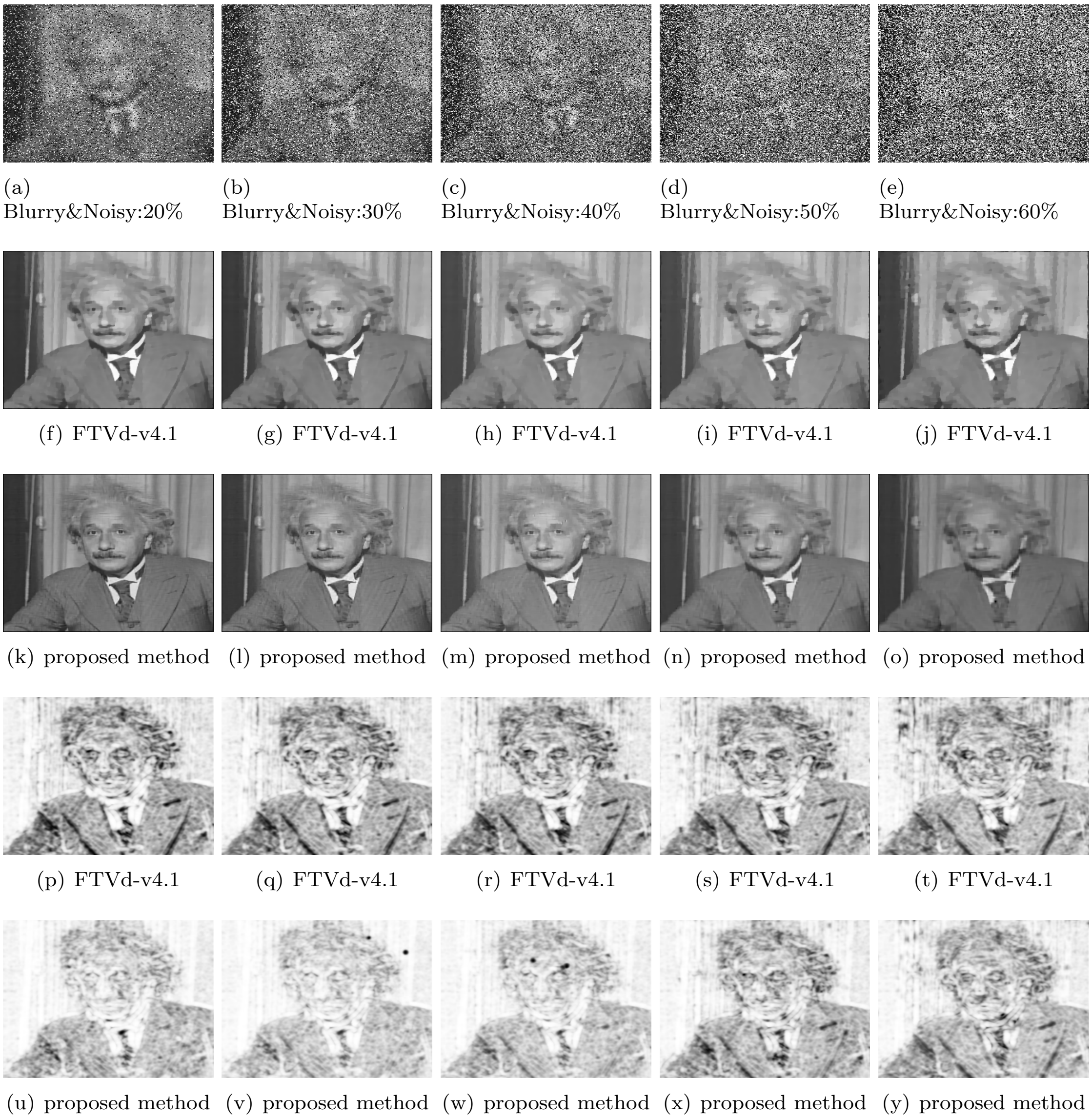
Table 2
PSNR(dB) values, SSIM and ReErr for various methods for the test images corrupted by salt-and-pepper noise levels from 20% to 60%, blurred by Average kernel of size 9.

Images	Noise levels	FTVd			Proposed method		
		PSNR	SSIM	ReErr	PSNR	SSIM	ReErr
Einstein	20%	30.452	0.798	0.062	34.127	0.890	0.043
	30%	30.141	0.790	0.064	33.276	0.878	0.047
	40%	29.816	0.781	0.067	32.394	0.850	0.054
	50%	28.621	0.763	0.077	31.040	0.824	0.062
	60%	27.381	0.737	0.088	29.015	0.785	0.074
Lena	20%	28.095	0.847	0.083	30.860	0.909	0.061
	30%	27.647	0.835	0.088	30.240	0.899	0.066
	40%	27.184	0.821	0.093	29.524	0.883	0.073
	50%	26.740	0.809	0.098	29.240	0.853	0.087
	60%	25.954	0.786	0.109	27.404	0.818	0.100
Cameraman	20%	29.316	0.851	0.084	32.231	0.927	0.053
	30%	27.960	0.843	0.088	30.703	0.914	0.061
	40%	28.343	0.832	0.094	28.721	0.879	0.077
	50%	26.935	0.811	0.103	27.591	0.850	0.089
	60%	24.957	0.770	0.122	25.188	0.800	0.113

4.2 Deblurring image with random-valued noise

In the third experiment, the image is blurred by a 7×7 Gaussian blur with standard deviation 5 and then corrupt with random-valued noise with noise levels $20\%, 30\%, 40\%, 50\%$ and 60%.

For the algorithm FTVd_v4.1, the parameters (μ, β_1, β_2) are tuned to get the best results, and they are set to be $(13, 1, 20)$ for the Einstein image for all the noise levels. The parameters (μ, β_1, β_2) are set to be $(10, 1, 20), (10, 1, 20), (13, 1, 20), (13, 1, 20), (10, 1, 20)$ for the Lena image, and the parameters (μ, β_1, β_2) are set to be $(25, 10, 20), (13, 5, 20), (13, 5, 20), (10, 1, 20), (4, 5, 10)$ for the Cameraman image. For the proposed Algorithm 1, we set $(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3)$ to be $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.005, 0.008, 5, 0.15, 0.05), (0.009, 0.02, 10, 0.15, 0.15), (0.04, 0.02, 20, 0.05, 0.15), (0.003, 0.02, 5, 0.15, 0.05)$ for the Einstein image, $(0.003, 0.008, 15, 0.15, 0.15), (0.003, 0.009, 15, 0.15, 0.15), (0.04, 0.02, 20, 0.05, 0.15)$, $(0.005, 0.02, 10, 0.15, 0.05), (0.003, 0.02, 5, 0.15, 0.05)$ for the Lena image, and $(0.003, 0.008, 15, 0.05, 0.15)$, $(0.003, 0.008, 15, 0.15, 0.05), (0.02, 0.02, 10, 0.15, 0.15), (0.01, 0.02, 5, 0.15, 0.05), (0.05, 0.02, 5, 0.15, 0.05)$ for the Cameraman image for the noise levels $20\%, 30\%, 40\%, 50\%$ and 60% respectively. The restored images by all two methods are shown in Figs 9-11(f)-(o) and the SSIM maps of the restored images in Figs 9-11(p)-(y).



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

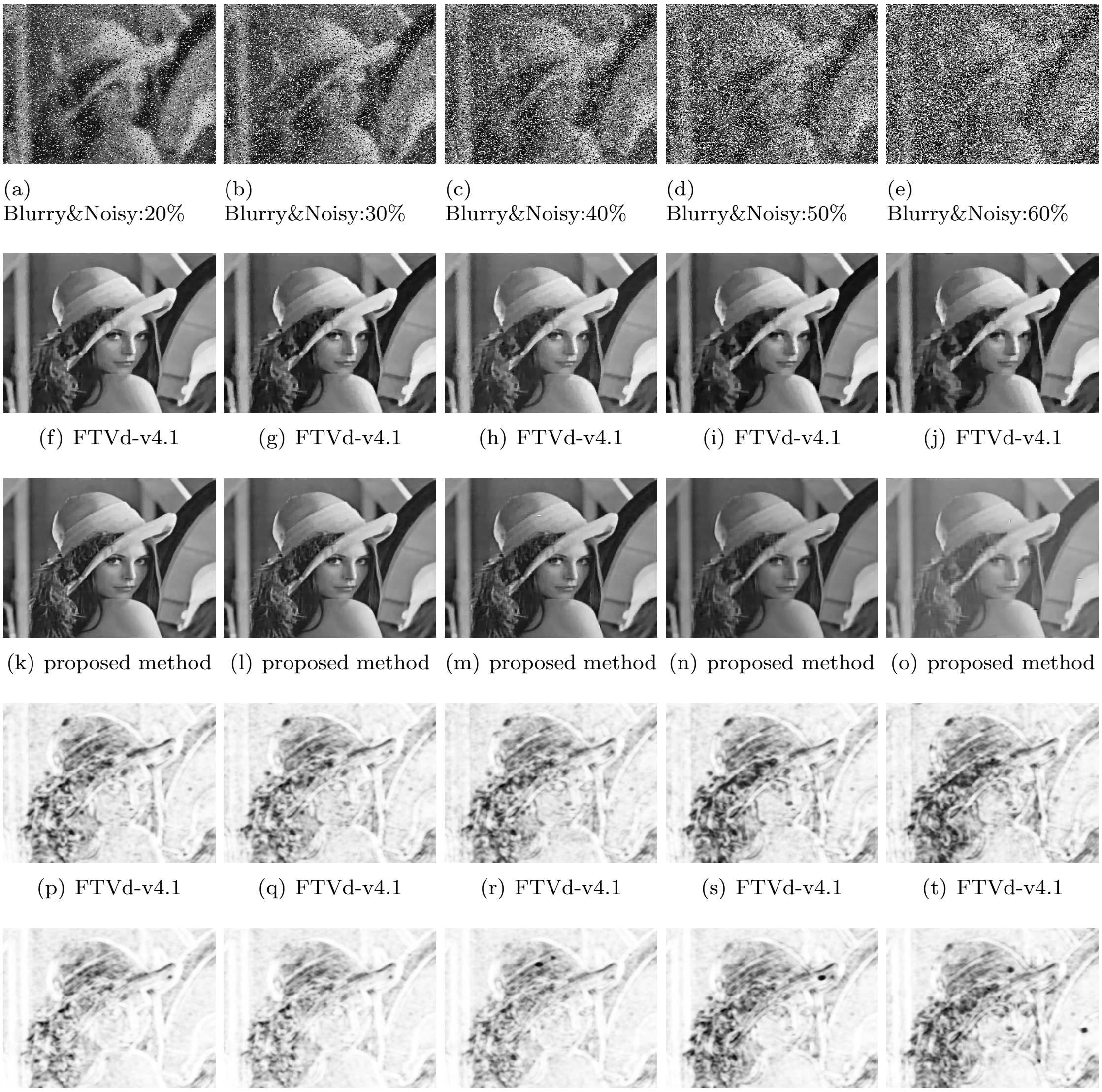
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.6. Recovered images of different methods on the image Einstein corrupted by Average(9) and salt-and-pepper noise with noise levels from 20% to 60%..



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

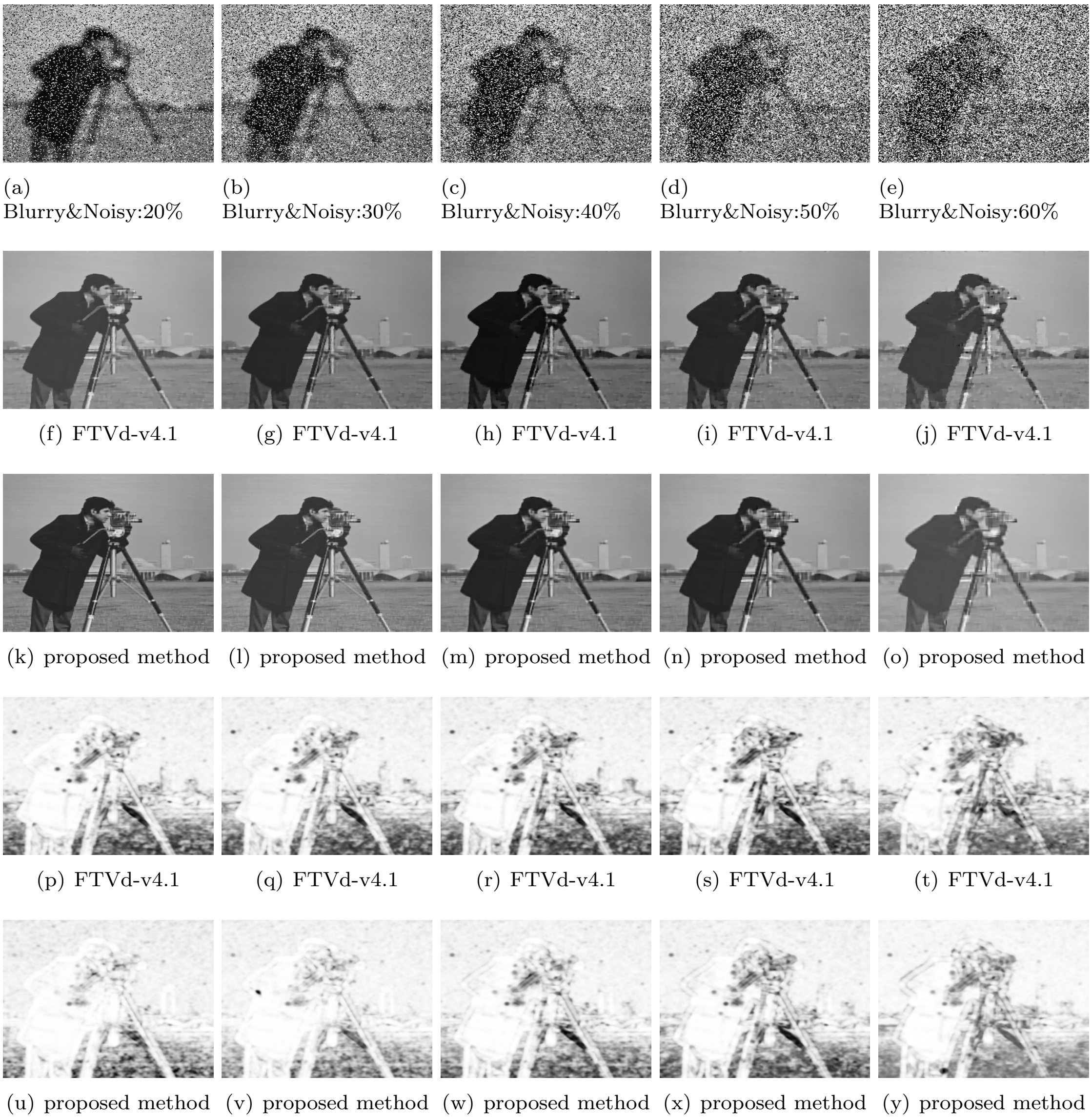
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.7. Recovered images of different methods on the image Lena corrupted by Average(9) and salt-and-pepper noise with noise levels from 20% to 60%..



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.8. Recovered images of different methods on the image Cameraman corrupted by Average(9) and salt-and-pepper noise with noise levels from 20% to 60%..

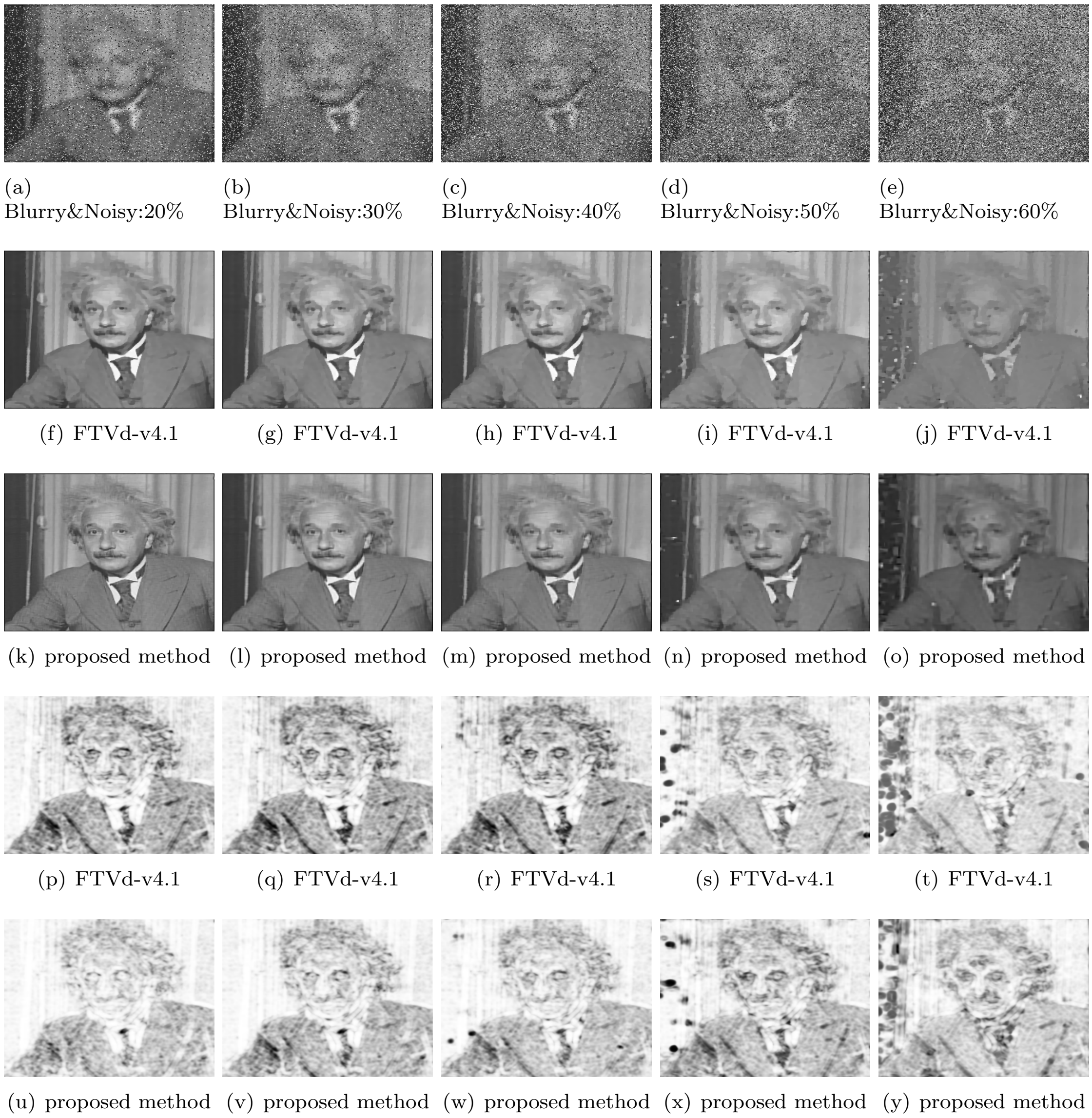
Table 3
PSNR(dB) values, SSIM and ReErr for various methods for the test images corrupted by random-valued noise levels from 20% to 60%, blurred by Gaussian kernel of size 7 × 7, and standard deviation 5.

Images	Noise levels	FTVd			Proposed method		
		PSNR	SSIM	ReErr	PSNR	SSIM	ReErr
Einstein	20%	31.164	0.819	0.057	34.298	0.902	0.041
	30%	30.771	0.812	0.060	32.731	0.879	0.048
	40%	30.087	0.797	0.065	31.716	0.848	0.056
	50%	27.299	0.757	0.089	28.823	0.794	0.080
	60%	26.250	0.667	0.153	26.669	0.709	0.125
Lena	20%	28.445	0.860	0.080	31.956	0.930	0.053
	30%	27.940	0.848	0.085	30.588	0.890	0.073
	40%	27.448	0.835	0.090	28.099	0.861	0.083
	50%	26.313	0.772	0.127	26.298	0.800	0.113
	60%	23.686	0.640	0.212	24.894	0.662	0.194
Cameraman	20%	28.554	0.857	0.079	32.681	0.935	0.048
	30%	26.594	0.832	0.092	29.950	0.904	0.065
	40%	26.250	0.810	0.101	27.935	0.855	0.089
	50%	23.741	0.725	0.145	23.808	0.735	0.130
	60%	20.122	0.618	0.186	20.248	0.608	0.183

In the last experiment, the image is blurred by a Average blur 9 and then corrupt with random-valued noise with noise levels $20\%, 30\%, 40\%, 50\%$ and 60%. The parameters (μ, β_1, β_2) in FTVd_$v4.1$ are set to be $(25,1,20), (25,1,20), (13,1,20), (13,1,20), (13,1,20)$, for the Einstein image, $(13,1,20), (13,1,20), (13,1,20), (13,1,20), (10,10,20)$ for the Lena image, and $(25,1,20), (25,1,20), (25,1,20), (8,5,20), (4,10,10)$, for the Cameraman image. For the proposed Algorithm 1, we set $(\alpha, \beta, \gamma_1, \gamma_2, \gamma_3)$ to be $(0.003, 0.008, 15, 0.05, 0.15)$,$(0.003, 0.008, 15, 0.05, 0.15)$,$(0.003, 0.02, 15, 0.15, 0.15)$,$(0.03, 0.009, 5, 0.05, 0.05)$,$(0.003, 0.02, 5, 0.15, 0.05)$ for the Einstein image, $(0.005, 0.008, 15, 0.05, 0.15)$, $(0.005, 0.008, 15, 0.05, 0.15)$, $(0.005, 0.008, 15, 0.15, 0.05)$, $(0.03, 0.009, 5, 0.05, 0.05)$, $(0.03, 0.02, 5, 0.05, 0.05)$ for the Lena image and $(0.003, 0.008, 15, 0.15, 0.15)$, $(0.003, 0.008, 15, 0.15, 0.15)$, $(0.003, 0.008, 10, 0.15, 0.05)$, $(0.1, 0.005, 10, 0.05, 0.05)$, $(0.1, 0.05, 20, 0.15, 0.15)$ for the Cameraman image for the noise levels $20\%, 30\%, 40\%, 50\%$ and 60% respectively. The restored images by all two methods are shown in Figs 12-14(f)-(o) and the SSIM maps of the restored images in Figs 12-14(p)-(y).

Table 4
PSNR(dB) values, SSIM and ReErr for various methods for the test images corrupted by random-valued noise levels from 20% to 60%, blurred by Average kernel of size 9.

Images	Noise levels	FTVd			Proposed method		
		PSNR	SSIM	ReErr	PSNR	SSIM	ReErr
Einstein	20%	30.326	0.795	0.063	33.735	0.889	0.044
	30%	30.018	0.787	0.065	32.548	0.876	0.050
	40%	29.386	0.779	0.070	31.319	0.845	0.057
	50%	27.407	0.750	0.088	28.365	0.780	0.079
	60%	24.969	0.674	0.131	25.268	0.711	0.113
Lena	20%	27.927	0.843	0.085	31.240	0.916	0.058
	30%	27.538	0.833	0.089	30.491	0.899	0.065
	40%	26.831	0.813	0.097	28.040	0.861	0.084
	50%	25.393	0.776	0.114	25.791	0.794	0.109
	60%	23.557	0.661	0.190	22.347	0.685	0.166
Cameraman	20%	29.545	0.841	0.087	31.872	0.929	0.052
	30%	27.353	0.841	0.090	30.251	0.901	0.068
	40%	26.310	0.821	0.097	27.856	0.861	0.086
	50%	23.453	0.734	0.130	23.200	0.739	0.130
	60%	20.123	0.606	0.186	20.218	0.611	0.184



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

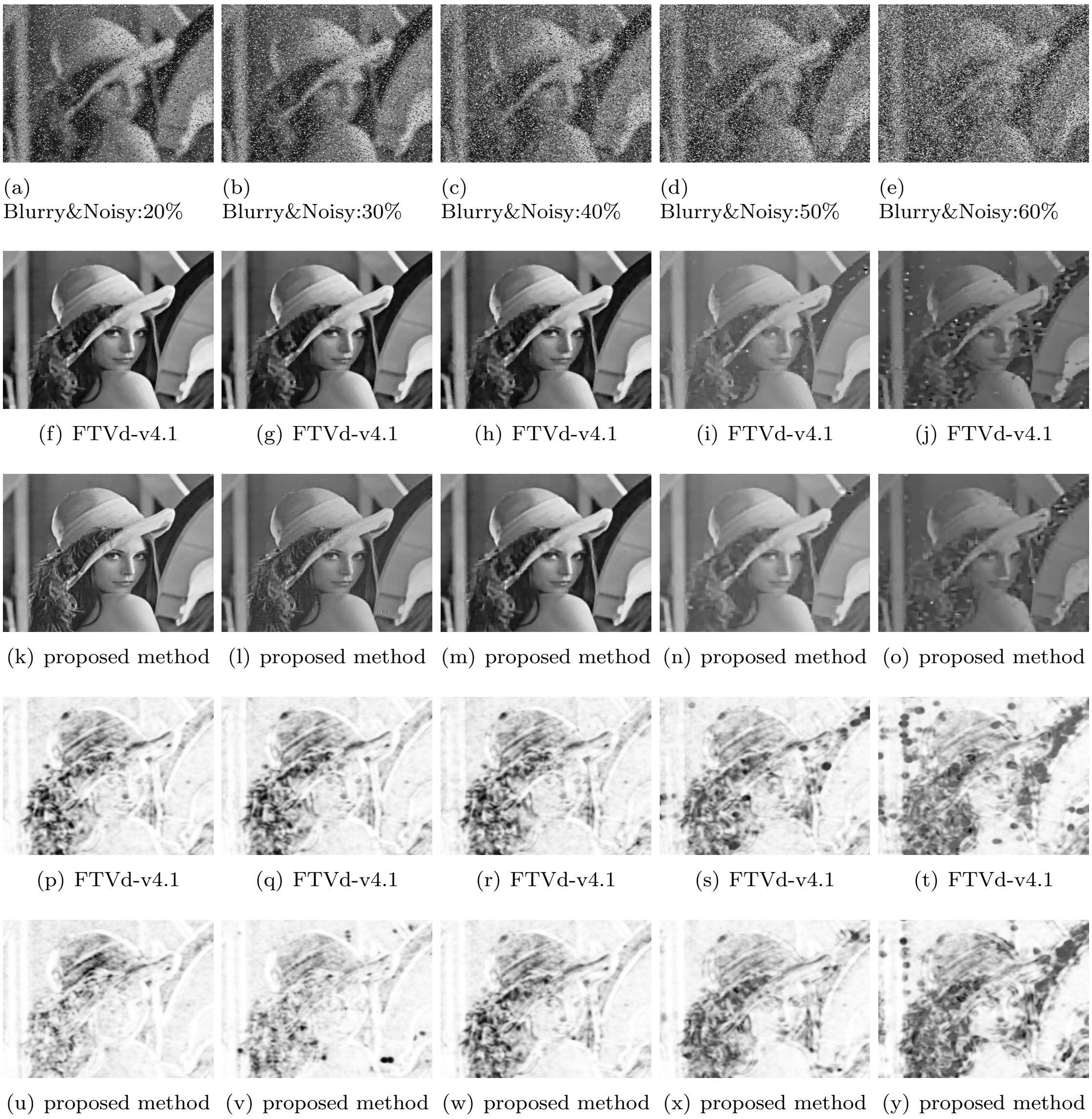
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.9. Recovered images of different methods on the image Einstein corrupted by Gaussian blur $G([7,7],5)$ and random-valued noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

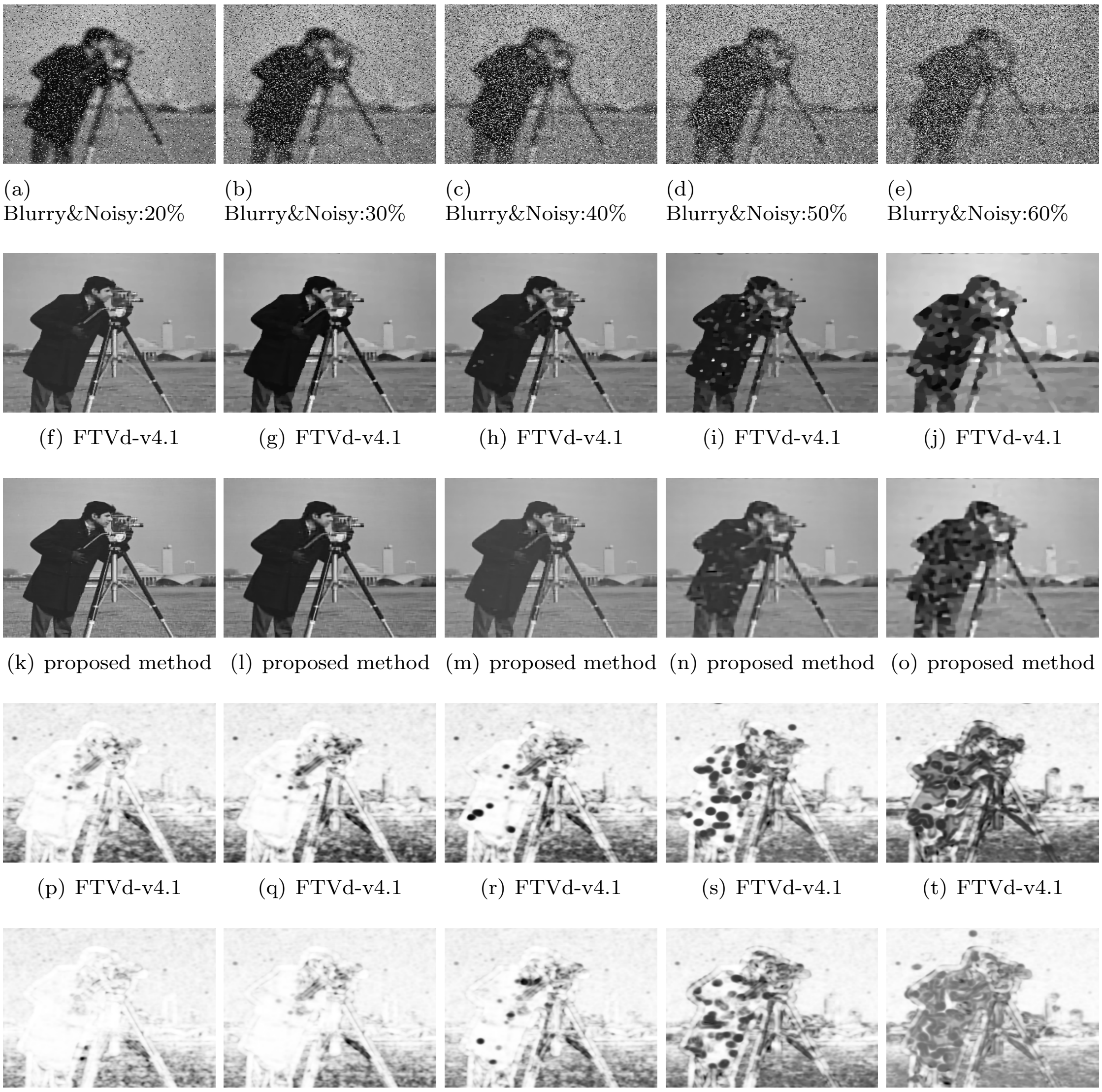
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.10. Recovered images of different methods on the image Lena corrupted by Gaussian blur $G([7,7],5)$ and random-valued noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

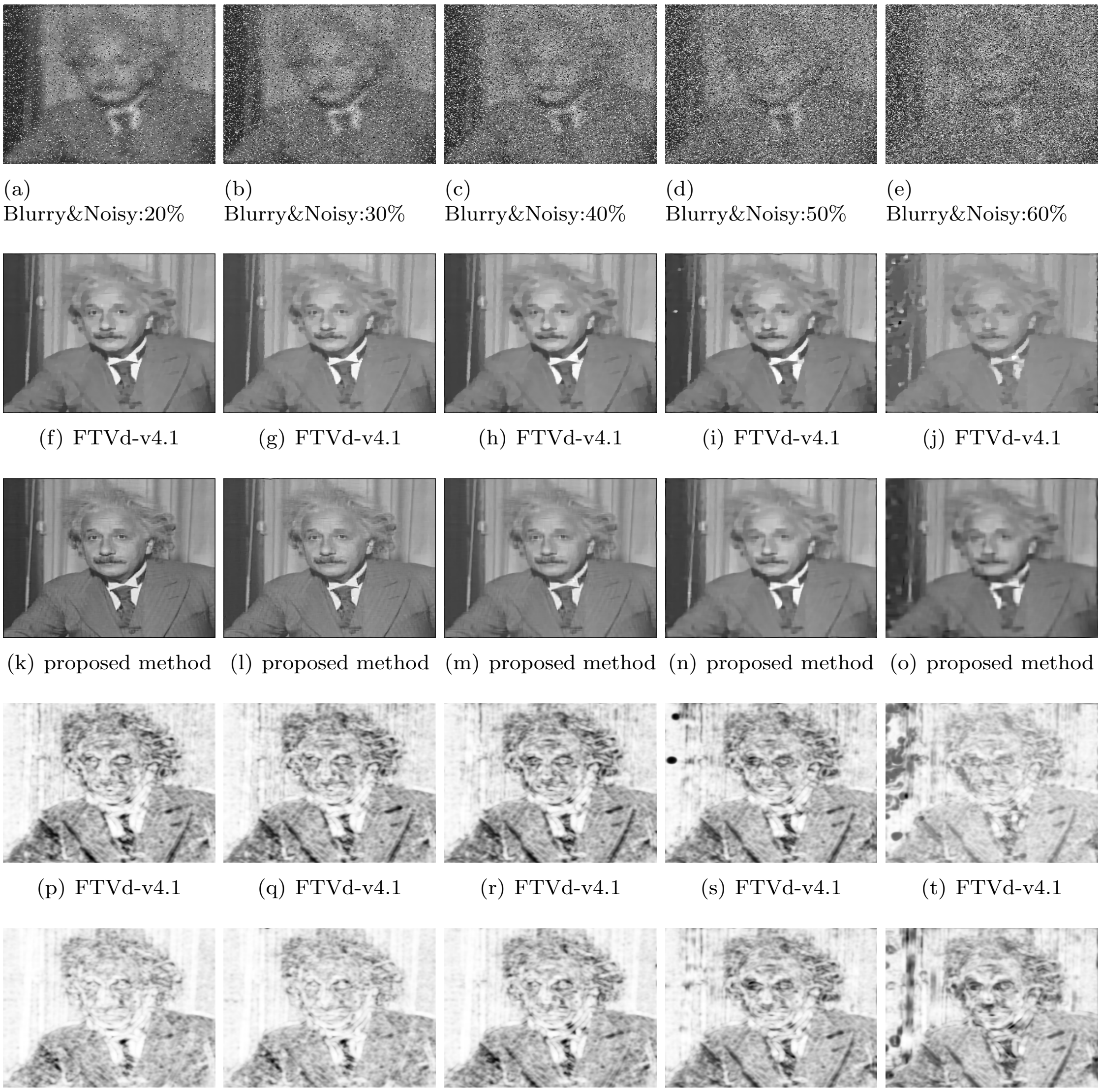
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.11. Recovered images of different methods on the image Cameraman corrupted by Gaussian blur $G([7,7],5)$ and random-valued noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

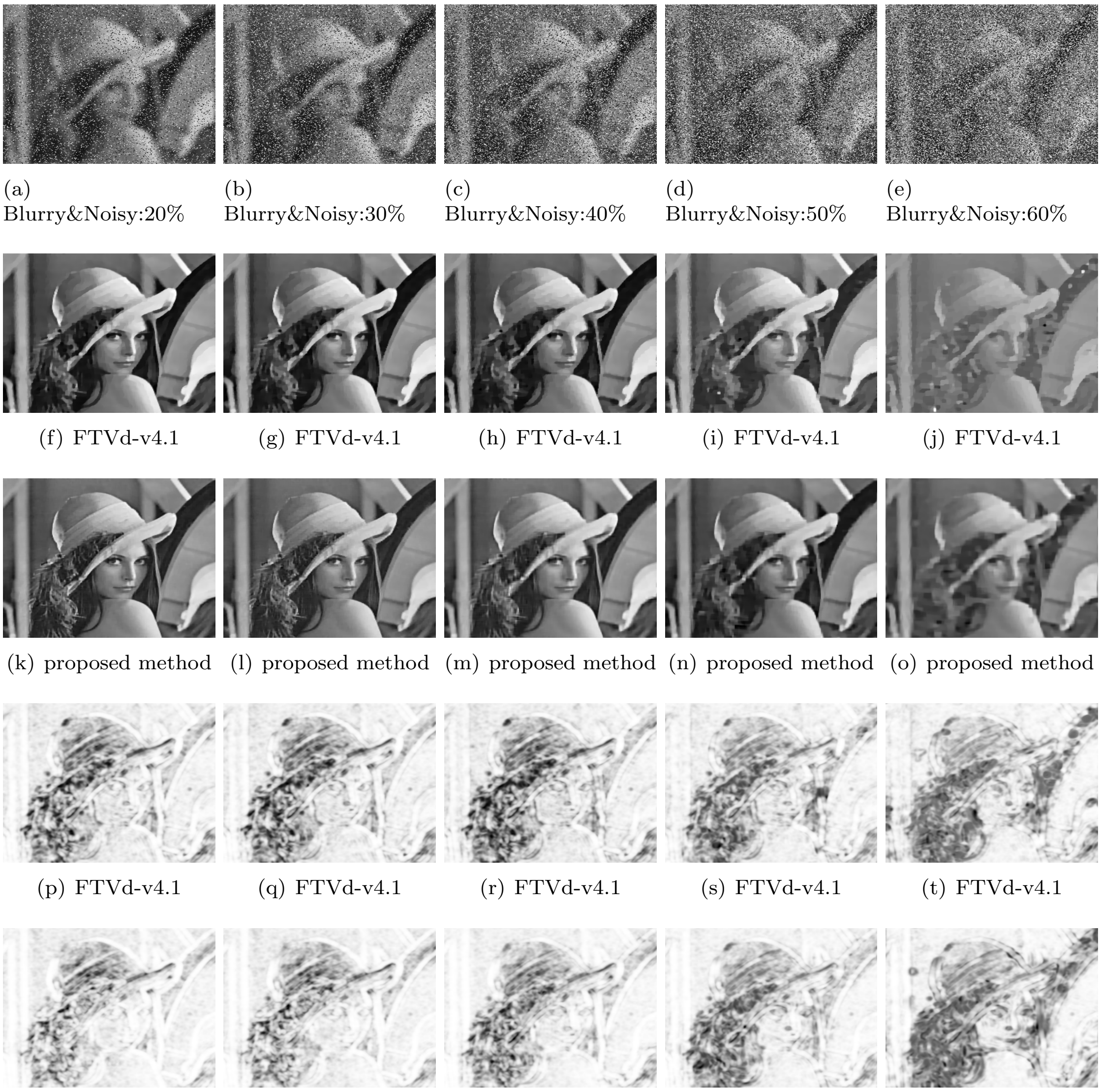
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.12. Recovered images of different methods on the image Einstein corrupted by Average(9) and random-valued noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

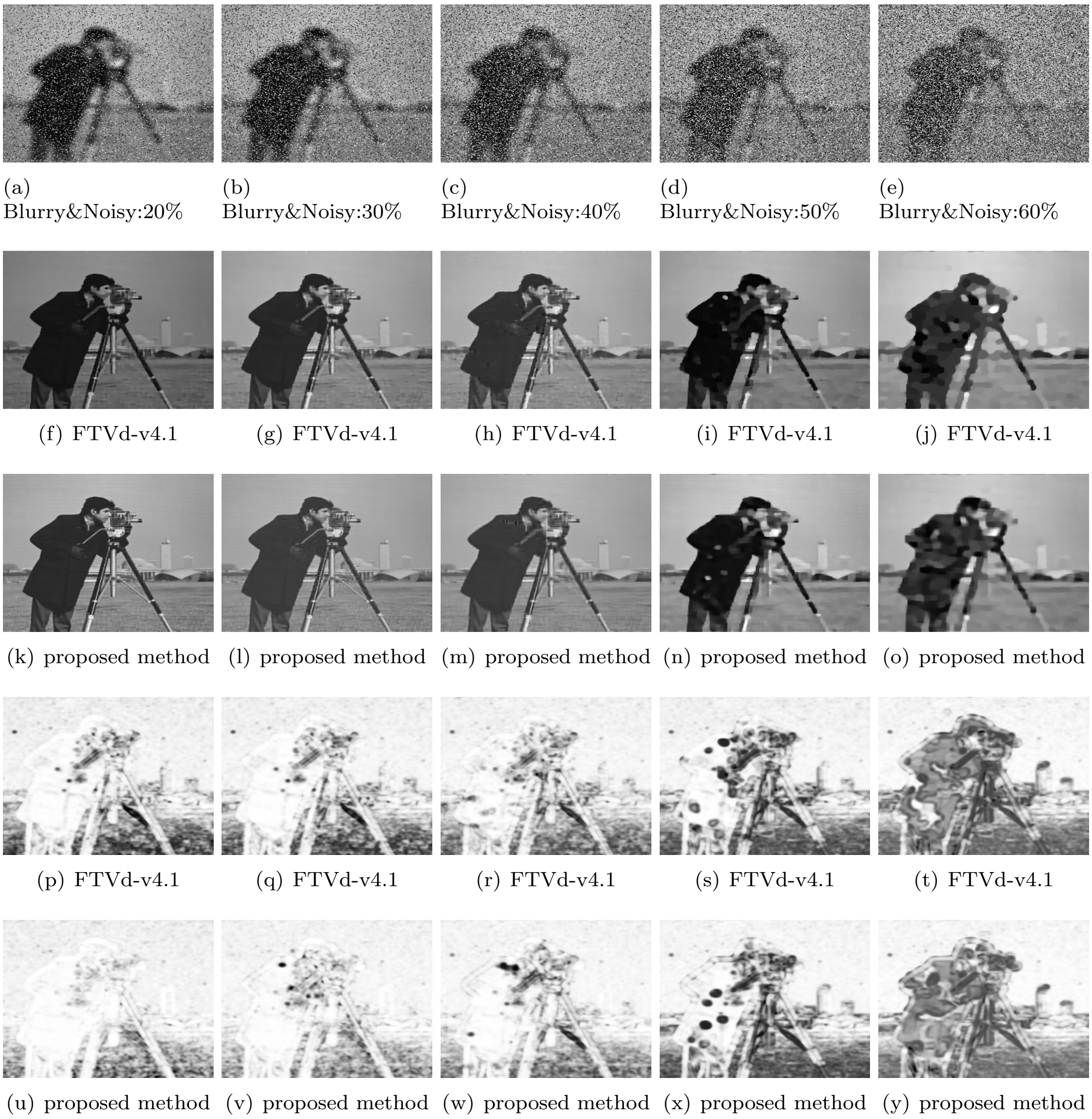
(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.13. Recovered images of different methods on the image Lena corrupted by Average(9) and random-valued noise with noise levels from 20% to 60%.



(a) Blurry&Noisy:20% (b) Blurry&Noisy:30% (c) Blurry&Noisy:40% (d) Blurry&Noisy:50% (e) Blurry&Noisy:60%

(f) FTVd-v4.1 (g) FTVd-v4.1 (h) FTVd-v4.1 (i) FTVd-v4.1 (j) FTVd-v4.1

(k) proposed method (l) proposed method (m) proposed method (n) proposed method (o) proposed method

(p) FTVd-v4.1 (q) FTVd-v4.1 (r) FTVd-v4.1 (s) FTVd-v4.1 (t) FTVd-v4.1

(u) proposed method (v) proposed method (w) proposed method (x) proposed method (y) proposed method

Fig.14. Recovered images of different methods on the image Cameraman corrupted by Average(9) and random-valued noise with noise levels from 20% to 60%.

From Figs. 3-14(f)-(0), we observe that the images restored by the proposed method have much smoother background than the FTVd-v4.1 method, and at the same time the edges can be preserved as well. After a visual inspection of the images, it is easy to check that compared with the FTVd-v4.1 method, the proposed approach yields better results since it avoids the staircase effect and preserves edges. In all experiment, we use the SSIM map to reveal areas of high or low similarity between two images. We present the SSIM maps of the restored images in Figs. 3-14(p)-(y). The reason for that choice is that in contrast with traditional quality measures like PSNR, the SSIM index also assesses the conservation of the structural information of the reconstructed image. We see from Figs. 3-14(p)-(y) that the SSIM map of the restored images by the proposed method is whiter than those by the FTVd-v4.1 method, i.e., our method can get better restoration results.

From the recovered quantities shown in Tables 1-4, we can judge the recovered PSNRs, SSIM and relative errors by the proposed method are much better than those obtained by the FTVd-v4.1 method.

References

[1] L. Bar, N. Soch and N.Kiryati, Image deblurring in the presence of salt-and-pepper noise. In Scale Space and PDE Methods in computer Vision, Lecture Notes in Comput. Sci.3459, Springer, Berlin, (2005)107-118.

[2] L. Bar, N. Soch and N.Kiryati,Image deblurring in the presence of salt-and-pepper noise. Int. J. Comput. Vis.,70(2006)279-298.

[3] T. Chen, H. Wu. Space variant median filters for the restoration of the impulse noisecorrupted images. IEEE Trans. Circuits Syst. II, 48(2001)784-789.

[4] R. Chan, T. Chan, L.Shen and Z.Shen. Wavelet algorithms for high-resolution image reconstruction. SIAM Journal on Scientific Computing. 24(4)(2003)1408-1432.

[5] R. Chan, S. Riemenschneider, L.Shen and Z.Shen. Tight frame: an efficient way for high-resolution image reconstruction. Applied and Computational Harmonic Analysis. 17(1)(2004)91-115.

[6] J. Cai, Z. Shen. Framelet based deconvolution. J.Comp. Math. 28(2010)289-308.

[7] F. Chen, Y. L. Jiao, G. R. Ma and Q. Q. Qin. Hybrid regularization image deblurring in the presence of impulsive noise. J.Vis. Commun. Image R. 24(2013)1349-1359.

[8] J. F. Cai, B. Dong, S. Osher and Z. Shen. Image restoration: Total variation, wavelet frames, and beyond. Journal of the American Mathematical. 25 (2012)1033-1089.

[9] A. Chai, Z. Shen. Deconvolution: A wavelet frame approach. Numerische Mathematik. 106(4)(2007)529-587.

[10] T. Chan, A. Marquina and P. Mulet. High- order total variation-based image restoration. SIAM Journal on Scientific Computing 22(2001)503-516.

[11] P. Charbonnier, L. B. Feraud, G. Aubert, M. Barlaud. Deterministic edge-preserving regularization in computed imaging, IEEE Transactions on Image Processing., 6(1997)298-311.

[12] A.Chai, Z.Shen. Deconvolution: a wavelet frame approach. Numer. Math. 106(2007)529-587.

[13] J.Chai, Z. Shen, Framelet based deconvolution. J. Comput. Math. 28(2010)289-308.

[14] J. F. Cai, R. H. Chan and M. Nikolova, Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise, Inverse Probl. Imaging, 2(2008)187-204.

[15] P.L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, New York,(2011)185-211.

[16] A. Chambolle, T. Pock, A first-order primal-dual algorithm algorithm for convex problems with applications to imaging, J. Math. Imaging Vision, 3(2011)120-145.

[17] I. Daubechies, Ten Lectures on Wavelets, CBMS Conf. Ser. in Appl. Math., vol. 61, SIAM, Philadelphia, 1992.

[18] B. Dong, Z.Shen, MRA based wavelet frames and applications, in: IAS Lecture Notes Series, Summer Program on the Mathematics of image Processing, Park City Mathematics INstitute.

[19] B.Dong, H. Ji, J.Li, Z.Shen and Y. Hu. Wavelet frame based blind image inpainting. Appl. Comput. Harmon. Anal., 32(2012)268-279.

[20] Y. Dong, M.HinterMüller and M. Neri, An efficient primal-dual method for L^1TV image restoration, SIAM J.Imaging Sci., 2(2009)1168-1189.

[21] I. Daubechies, B. Han, A. Ron, Z. Shen, Framelets: MRA-based constructions of wavelet frames, Appl. Comput. Harmon. Anal. 14(2003)1-14.

[22] E. Esser, X. Zhang, and T. F. Chan, A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science, SIAM J. Imaging Sci., 3(2010)1015-1046.

[23] T. Goldstein, S. Osher, The split Bregman method for L1-regularized problems, SIAM J. Imaging Sci., 2(2009)323-343.

[24] D. Gabay, B. Mercier, a dual algorithm for the solution of nonlinear variational problems via finite element approximation, Comput. Math. Appl., 2(1976)17-40.

[25] X. Guo, F. Li and M. K. Ng, A fast $l_1 - TV$ algorithm for image restoration, SIAM J. Sci. Comput., 31(2009)2322-2341.

[26] H. Hwang, R.Haddad. Adaptive median filters: New algorithms and results, IEEE Trans. Image Process., 4(1995)2322-2341.

[27] Y. R. Li, L. Shen, D.Q. Dai and B.W. Suter. Framelet algorithms for de-blurring images corrupted by impulse plus Gaussian noise. IEEE Trans. Image Process. 20(2011)1822-1837.

[28] S. Mallat, A Wavelet Tour of Signal Processing, second ed., Academic Press, 1999.

[29] M. Nikolova. A variational approach to remove outliers and impulse noise. J.Math.Imaging Vision, 20(2004)99-120.

[30] M. K. Ng, P. Weiss, and X. Yuan, Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods, SIAM J. Sci. Comput., 32(2010)2710-2736.

[31] M. K. Ng, R. H. Chan and W. C. Tang, A fast algorithm for deblurring models with Neumann boundary conditions, SIAM J. Sci. Comput., 21(2000)851-866.

[32] G. Pok, J. Liu and A. Nair, Selective removal of impulse noise based on homogeneity level information, IEEE Trans. Image Process., 12(2003)85-92.

[33] W. Partt. Median Filtering. Technical report, Image Processing Institute, University of Southern California, Los Angeles, CA, (1975).

[34] L.I. Rudin, S. Osher, E. Fatemi. Nonlinear total variation based noise removal algorithms, Physica D, 60(1992)259-268.

[35] M.Tao, J.Yang, Alternating direction algorithms for total variation deconvolution in image reconstruction, Prepring, 2009.

[36] A. Ron, Z. Shen, Affine Systems in $L_2(\mathbb{R})$: The Analysis of the Analysis Operator, Journal of Functional Analysis. 148(2) 93(1997)408-447.

[37] C. Wu, J. Zhang, X. C. Tai, Augmented Lagrangian method for total variation restoration with non-quadratic fidelity, Inverse problems and imaging, 5 (2011)237-261.

[38] Y. Wang, J.Yang, W.Yin and Y.Zhang. A new alternating minimization algorithm for total variation image reconstruction. SIAM J Imaging Sci,1(3)(2008)248-272.

[39] Z. Wen, D. Goldfarb, W.Yin, Alternating direction augmented Lagrangian methods for semidefinite programming. Math Program Comput, 2(3-4)(2010)203-230.

[40] C. L. Wu and X. C. Tai, Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high Order Models, SIAM J. Imaging Sci., 3(3)(2010) 300-339.

[41] J. Yang, Y. Zhang, W. Yin, An efficient TVL_1 algorithm for deblurring multichannel images corrupted by impulsive noise, SIAM Journal on Scientific Computing, 31(2009)2842-2865.