



## Image denoising via a new anisotropic total-variation-based model

Zhi-Feng Pang<sup>a,\*</sup>, Ya-Mei Zhou<sup>a</sup>, Tingting Wu<sup>b</sup>, Ding-Jie Li<sup>c</sup><sup>a</sup> School of Mathematics and Statistics, Henan University, Kaifeng, 475004, China<sup>b</sup> College of Science, Nanjing University of Posts and Telecommunications, Nanjing, 210023, China<sup>c</sup> Department of Radiation Oncology, Affiliated Cancer Hospital of Zhengzhou University, Zhengzhou, 450008, China

## ARTICLE INFO

## Keywords:

Image denoising

Anisotropic total variation

Alternating direction method of multipliers (ADMM)

Weighted matrix

## ABSTRACT

To keep local structures when denoising the degraded image, we propose a new anisotropic total variation (TV)-based restored model based on the combination of the gradient operator  $\nabla$  and the adaptive weighted matrix  $T$  into the  $\ell^1$ -norm regularized term. The weighted matrix  $T$  depends on the edge indicator function along the  $x$  and  $y$ -axis directions, so this matrix can rotate the direction of the gradient operator tending to bigger weight and the proposed model can thus describe the local features in image. In order to cope with this nonsmooth model, we employ the alternating direction method of multipliers (ADMM) to solve it. Relying on the convexity, the convergence of the proposed numerical algorithm is provided as well. Denoising experiments on the artificial images and benchmark images show the effectiveness of the proposed model by comparing it to other well-known total-variation-based models in terms of the restored quality.

## 1. Introduction

Image restoration plays an important role in various applied areas such as medical and astronomical imaging, video coding, computer vision and many others [1,2]. One of the most challenging tasks for this problem is of efficiently keeping image details such as edges and textures when restoring the degraded image. Since this course lacks some prior information, it is a classic ill-posed problem. Then we need to employ some mathematical schemes to overcome it. One of core techniques is the regularization method modeled by

$$\min_u \lambda F(u, f) + \mathcal{R}(u), \quad (1)$$

where  $f: \Omega \rightarrow \mathbb{R}$  denotes the degraded image and  $u: \Omega \rightarrow \mathbb{R}$  is the expected restored image. Here  $\Omega \subset \mathbb{R}^2$  denotes the image domain. In the model (1), the first term is the fidelity term and the second term is the regularization term. The purpose of the regularization parameter  $\lambda > 0$  is to balance the influence of these two terms on the computed solution in a suitable manner.

For the model (1), how to choose an effective regularization term is the key problem. The choice of this term mainly depends on different restoration tasks. The earliest scheme is a quadric functional of the  $\ell^2$  norm,  $\|L u\|_2^2$ , often called Tikhonov regularization [3], where  $L$  is the differential operator. Obviously, Tikhonov regularization has excellent performance in noise-removing since it corresponds to a Gaussian filter operation. However, it often overly smoothes image edges which are important features in image recognition. In order to overcome this drawback, a very successful scheme is the seminal work of Rudin,

Osher and Fatemi (ROF) [4], where they employed the total variation (TV) minimization to remove the white Gaussian noise. Since the space of total variation exhibits jump discontinuities, TV-based models are quite efficient for regularizing image without smoothing image edges and therefore used in many fields of image processing such as image reconstruction [5,6], inpainting [7], and segmentation [8], etc. However, to TV-based models, one of main disadvantage is the promotion of piecewise constant structures in the solution  $u$  since it penalizes the small jump, a phenomenon known as staircasing effect. To do this, by using some properties of fourth-order linear or nonlinear diffusion damps oscillations much faster than second-order diffusion, some higher order extensions of the total variation have been proposed in literatures [9–11]. There are some unsuitable facts that the high-order TV-based models often blur image edges. So other improved models have been proposed such as the total generalized variation (TGV) model [11], TV-Stokes two-step model [12], hybrid TV-based model [13,14], nonlocal-based model [15,16], Schatten-norm [17] and fractional order derivative-based model [18,19], etc. These schemes were reported to give better restored performance than the standard TV-based regularizer in some regions of image. However, these model still may not be ideally suited for the regularizer of ill-posed inverse problems when describing the image local details.

In fact, above TV-based models are the isotropic model since they use the same weight for the subvariable in the gradient operator  $\nabla := (\nabla_x, \nabla_y)^T$ . This implies that  $\nabla_x$  and  $\nabla_y$  give the same penalties along the  $x$ -axis and the  $y$ -axis directions. Actually, since the image includes many different features, we often wish that the proposed model diffuse

\* Corresponding author.

E-mail addresses: [zhifengpang@163.com](mailto:zhifengpang@163.com) (Z.-F. Pang), [ya\\_mei\\_zhou@163.com](mailto:ya_mei_zhou@163.com) (Y.-M. Zhou), [wutt@njupt.edu.cn](mailto:wutt@njupt.edu.cn) (T. Wu), [150304089@qq.com](mailto:150304089@qq.com) (D.-J. Li).

along the tangent direction of local features. Then this setting for two subvariables in the gradient operator with the same weight cannot efficiently couple with local features. In order to deal with this unpleasant effect, we need to add different weight into each component of the gradient operator. So we propose the following anisotropic TV-based (ATV) model as

$$\min_{u(x,y)} \frac{\lambda}{2} \|u(x,y) - f(x,y)\|_2^2 + \|\mathbf{T}(x,y)\nabla u(x,y)\|_{2,1}, \quad (2)$$

where  $\mathbf{T}(x,y) := \text{diag}(t_1(x,y), t_2(x,y))$  and  $\|\cdot\|_2$  and  $\|\cdot\|_{2,1}$  respectively denote the  $\ell^2$ -norm and the  $\ell^{2,1}$ -norm. Obviously, the model (2) tends to the ROF model when setting  $t_1(x,y)$  and  $t_2(x,y)$  to be the same constant. Actually, from the viewpoint of geometry, coupling  $\mathbf{T}(x,y)$  with  $\nabla u(x,y)$  together can be regarded as rotating the vector  $\nabla u(x,y)$  with the angle  $\alpha - \beta$  as shown in Fig. 1. It implies that this model can efficiently describe local features when choosing suitable  $t_1(x,y)$  and  $t_2(x,y)$  comparing with the classic TV-based models. So how to choose a suitable  $\mathbf{T}(x,y)$  is very important. To this aim, we choose a group of images with seven different affine angles such as  $0, \pi/12, \pi/6, \pi/4, 2\pi/3, 5\pi/12, \pi/2$  and degrade them by adding to the white Gaussian noise with different standard deviations as  $\sigma = 0.01, 0.05, 0.1$ . The original images and the relationships between  $t_1(x,y)$  and  $t_2(x,y)$  are shown in Fig. 2 when restoring degraded images. As we can see from the second row in Fig. 2 that the ratio value of  $\log(t_1(x,y)/t_2(x,y))$  depends on the change of angles when we want to obtain a better restored image. Especially, the ratio is almost symmetrical. These imply that our proposed model will give more robust restored results when choosing the adaptive matrix  $\mathbf{T}(x,y)$  by coupling with the local structure information at the pixel point  $(x,y)$ . Among the featured schemes, the gradient information can relate the structure of objects in an image, identify features of interest for recognition/classification directly or provide the basis of further processing for various computer vision tasks. Thus, here we still employ it as the adaptively weigh function

$$\mathbf{T}(x,y) = \begin{bmatrix} t_1(x,y) & 0 \\ 0 & t_2(x,y) \end{bmatrix} := \begin{bmatrix} \frac{1}{1 + \kappa |G_{\hat{\sigma}}(x,y) * \nabla_x f(x,y)|} & 0 \\ 0 & \frac{1}{1 + \kappa |G_{\hat{\sigma}}(x,y) * \nabla_y f(x,y)|} \end{bmatrix} \quad (3)$$

to indicate changes of  $t_1(x,y)$  and  $t_2(x,y)$ , where  $\kappa > 0$  and  $\hat{\sigma}$  are parameters. In the formula (3), we can find that the gradient direction  $\nabla u$  will tend to the horizontal direction  $\nabla_x$  when  $\nabla_x f < \nabla_y f$ .<sup>1</sup> This states the change in the difference direction of  $\nabla u$  for the degraded image to be superior and vice versa. It is worth noting that this setting for  $\mathbf{T}(x,y)$  does not effect the convergence analysis of our proposed numerical method since our proposed model (2) is the pointwise.<sup>1</sup>

As some commendations, our proposed model is closely related to the works in [20,21]. In [20], authors proposed to replace the isotropic TV semi-norm by an anisotropic term that mirrors the directional structure of either the noisy original data or the smoothed image, where the anisotropy depends on eigenvalues of the structure tensor. They also discussed the existence theory by using the concept of relaxation. In [21], another anisotropic model was proposed by introducing a new exponentially directional weighted function (EDWF), which is based on the difference between  $\ell^1$  and  $\ell^2$  norms over the exponential function. Based on the weighted difference of anisotropic and isotropic total variation model, a  $\ell^2 - \alpha \ell^1$ -based anisotropic model in [22] was developed for the image processing, where  $\alpha$  is chosen according to the gradient distributions with in  $[0,1]$ . Furthermore, with the help of the Wulff shape and the rotation transformation, an anisotropic TV-based model in [23,24] was obtained by containing rotated (linearly transformed)

<sup>1</sup> In order to simplify notations, we omit the variables  $(x,y)$  for all of functions  $f(x,y)$ ,  $u(x,y)$  and operators  $\mathbf{T}(x,y)$ ,  $\nabla u(x,y)$  to be  $f$ ,  $u$  and  $\mathbf{T}$ ,  $\nabla u$  in the following contents.

rectangular shapes which can avoid the round-off effects at vertices. However, different to our proposed model (2), these mentioned models are often nonconvex, especially when choosing the restored image as the original inputting image for the next iteration in the corresponding numerical implementation [20,21]. To be the contrast, our proposed model (2) has more flexible anisotropy via adding different weights into the gradient operator.

To the numerical implementation, it is difficult to solve the proposed model (2) since two linear operators  $\mathbf{T}$  and  $\nabla$  are included into the non-differentiable  $\ell^1$ -norm. In the past several decades, a plethora of approaches have been proposed such as primal-based methods [3,4,25,26], dual-based methods [27–30] and primal–dual-based methods [31–33], etc. Here we only consider to transform the model (2) into the constrained optimization problem by introducing some auxiliary variables. Then we employ the augmented Lagrangian (AL) method to obtain the AL function. Since this function as a saddle point problem includes three primal variables and two dual variables, we can use the alternating direction method of multipliers (ADMM) to solve it. As a summary, our main contributions are two-folds in the following.

- We propose a novel anisotropic TV-based model (2) for the image denoising problem. Comparing with some related TV-based models, our proposed model is particularly suitable for describing local features of image since we add the different weights to the gradient operator based on the local information.
- To deal with the nonsmoothing in the model (2), we employ the ADMM to solve it and also give the related convergence analysis. Furthermore, experimental results demonstrate that the proposed model (2) is able to improve image quality significantly compared with the state-of-the-art TV-based models.

The rest of this paper is organized as follows. In Section 2, we give a brief review of the TV-based models, which are very relevant to our proposed model. In Section 3, an alternating minimization algorithm is employed to find the minimizer of the proposed minimization problem. We also analyze the convergence of the proposed alternating minimization algorithm in this section. Some numerical experiments are given to illustrate the performance of the proposed algorithm in Section 4. Concluding remarks are given in the last section.

## 2. Preliminaries

In this section, we firstly introduce some notations such as the discrete gradient operator with the assumption of periodic boundary condition and soft-thresholding operator, then summarize some related gradient-based restored models closely related to our proposed model.

### 2.1. Notations

Let us describe some notations used throughout this paper and assume the size of image to be  $m \times n$ . We denote by  $X$  the Euclidean space  $\mathbb{R}^{m \times n}$  and set  $Y = X \times X$ . Inner products and Euclidean norms are defined as

$$(\bar{u}, \bar{v}) = \sum_{i=1}^m \sum_{j=1}^n \bar{u}_{i,j} \bar{v}_{i,j}, \quad \|\bar{u}\|_2 := \sqrt{\sum_{i=1}^m \sum_{j=1}^n \bar{u}_{i,j}^2},$$

$$\langle \hat{\mathbf{g}}, \hat{\mathbf{h}} \rangle = \sum_{i=1}^m \sum_{j=1}^n \sum_{s=1}^2 \hat{\mathbf{g}}_{i,j,s}^2 \hat{\mathbf{h}}_{i,j,s}^2, \quad \|\hat{\mathbf{g}}\|_{2,1} = \sum_{i=1}^m \sum_{j=1}^n \sqrt{\sum_{s=1}^2 \hat{\mathbf{g}}_{i,j,s}^2}$$

for  $\bar{u}, \bar{v} \in X$  and  $\hat{\mathbf{g}} := (g_1, g_2)^T \in Y$ ,  $\hat{\mathbf{h}} := (h_1, h_2)^T \in Y$ . In the following, we define the discrete gradient  $\nabla \bar{u}_{i,j} = (D_x^+ \bar{u}_{i,j}, D_y^+ \bar{u}_{i,j})$  as forward difference operators

$$D_x^+ \bar{u}_{i,j} = \begin{cases} \bar{u}_{i+1,j} - \bar{u}_{i,j} & \text{if } 1 \leq i < m, 1 \leq j \leq n \\ \bar{u}_{1,j} - \bar{u}_{i,j} & \text{if } i = m, 1 \leq j \leq n, \end{cases}$$

$$D_y^+ \bar{u}_{i,j} = \begin{cases} \bar{u}_{i,j+1} - \bar{u}_{i,j} & \text{if } 1 \leq i \leq m, 1 \leq j < n \\ \bar{u}_{i,1} - \bar{u}_{i,j} & \text{if } 1 \leq i \leq m, j = n. \end{cases}$$

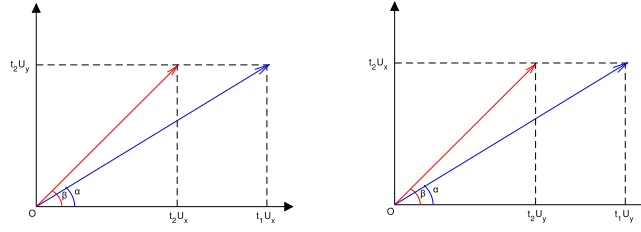


Fig. 1. The gradient  $\nabla u(x, y)$  with different weighted matrix  $T(x, y)$ : Left:  $t_1(x, y)u_x(x, y) > t_2(x, y)u_y(x, y)$ ; Right:  $t_2(x, y)u_x(x, y) < t_1(x, y)u_y(x, y)$ . Here we also set  $\beta = \arctan \frac{\nabla_y u(x, y)}{\nabla_x u(x, y)}$  and  $\alpha = \arctan \frac{t_2(x, y)\nabla_y u(x, y)}{t_1(x, y)\nabla_x u(x, y)}$ .

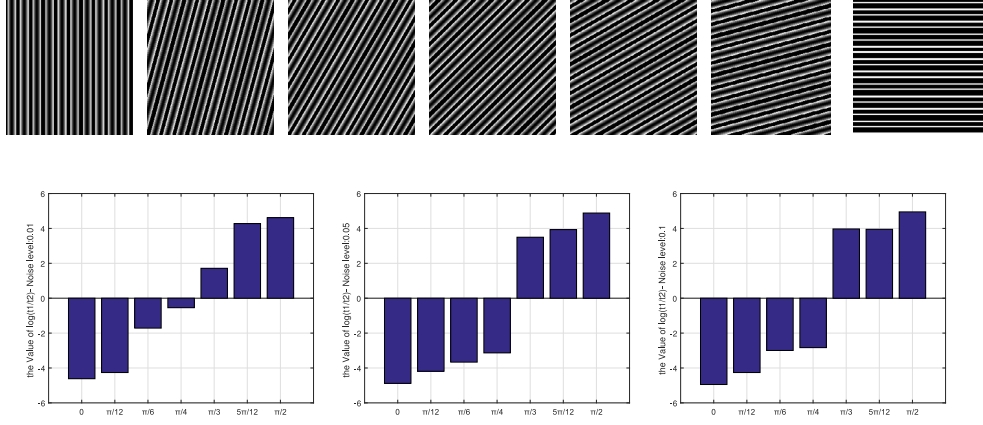


Fig. 2. First row: testing affine images with different angles 0,  $\pi/12$ ,  $\pi/6$ ,  $\pi/4$ ,  $2\pi/3$ ,  $5\pi/12$ ,  $\pi/2$ ; Second row: the ratio of  $\log(t_1/t_2)$ .

Similarly, we can also define backward difference operators as

$$D_x^- \bar{u}_{i,j} = \begin{cases} \bar{u}_{i,j} - \bar{u}_{i-1,j} & \text{if } 1 < i \leq m, 1 \leq j \leq n \\ \bar{u}_{i,j} - \bar{u}_{m,j} & \text{if } i = 1, 1 \leq j \leq n, \end{cases}$$

$$D_y^- \bar{u}_{i,j} = \begin{cases} \bar{u}_{i,j} - \bar{u}_{i,j-1} & \text{if } 1 \leq i \leq m, 1 < j \leq n \\ \bar{u}_{i,j} - \bar{u}_{i,n} & \text{if } 1 \leq i \leq m, j = 1. \end{cases}$$

Using the divergence theorem in the case of the discrete setting as

$$-\operatorname{div} \rho \cdot \bar{u} = \rho \cdot \nabla \bar{u}$$

for  $\bar{u} \in X$  and  $\rho = (\rho^1, \rho^2) \in Y$ , where  $-\operatorname{div}$  denotes the adjoint operator of  $\nabla$ , then we have

$$\operatorname{div} \rho_{i,j} = D_x^- \rho_{i,j}^1 + D_y^- \rho_{i,j}^2.$$

## 2.2. Soft-thresholding operator

The soft-thresholding operator has been employed in many splitting type methods since we can use it to obtain the explicit solution in many  $\ell^1$ – $\ell^2$  optimization problems. Actually, this operator is closely related to the proximal operator [32,34]. Define the following abbreviation as

$$\Gamma_0 := \{g : \Omega \rightarrow \mathbb{R} \cup \{+\infty\} : g \text{ is proper, lower semicontinuous, convex}\}$$

Let  $g \in \Gamma_0$  and  $\tau > 0$ , the mapping

$$\operatorname{Prox}_g(\mathbf{y}, \tau) := \operatorname{argmin}_{\mathbf{x}} \left\{ g(\mathbf{x}) + \frac{\tau}{2} \|\mathbf{x} - \mathbf{y}\|_2^2 \right\} \quad (4)$$

is called the proximal operator of the function  $g$  at  $\mathbf{y}$  with the scaling parameter  $\tau$ . It is obvious that the proximal operator is used to make an approximation to a value, while making a compromise between the accuracy of the approximation and a cost associated to the new value [35]. There are many choices for the function  $g(\mathbf{x})$  such as the  $\ell^p$ -norm and the indicator function for a convex set. However, we only consider  $g(\mathbf{x}) = \|\mathbf{x}\|_{2,1}$  for  $\mathbf{x} := (x_1, x_2)^T \in Y$ ,  $\mathbf{y} := (y_1, y_2)^T \in Y$ . In

this case, the proximal problem (4) is separable, so we can obtain the explicit scheme as

$$\operatorname{Prox}_{\|\cdot\|_2}(\mathbf{y}_{i,j}, \tau) = \max \left\{ \|\mathbf{y}_{i,j}\|_2 - \frac{1}{\tau}, 0 \right\} \frac{\mathbf{y}_{i,j}}{\|\mathbf{y}_{i,j}\|_2}, \quad (5a)$$

where  $\|\mathbf{y}_{i,j}\|_2 = \sqrt{y_{i,j,1}^2 + y_{i,j,2}^2}$ . Since many operators in the image processing are the piecewise point, for the aim of the simplification, we rewrite the above operator into a general form as

$$\operatorname{Prox}_{\|\cdot\|_{2,1}}(\mathbf{y}, \tau) = \left\{ \operatorname{Prox}_{\|\cdot\|_2}(\mathbf{y}_{1,1}, \tau), \operatorname{Prox}_{\|\cdot\|_2}(\mathbf{y}_{1,2}, \tau), \dots, \operatorname{Prox}_{\|\cdot\|_2}(\mathbf{y}_{m,n}, \tau) \right\} \quad (5)$$

without any confusions.

## 2.3. Related models

Since our proposed model is based on the gradient operator, this subsection mainly reviews some related gradient-based models used in many classic image denoising works [4,9,10,25,36,37].

### 2.3.1. ROF model

Since its introduction in 1992, the Rudin–Osher–Fatemi (ROF) model [4], also known as the total variation denoising, has found numerous applications. One way to put this model is that the total variation of an image is used as a regularizer for an image denoising optimization problem

$$\min_u \frac{\lambda}{2} \|u - f\|_2^2 + \|\nabla u\|_{2,1}. \quad (6)$$

This model is closely related to the fact that the  $BV(\Omega)$  space in the continuous space as

$$BV(\Omega) := \sup \left\{ \int_{\Omega} u \operatorname{div} \mathbf{p}, \mathbf{p} \in C_0^\infty(Y, \mathbb{R}^2) \text{ and } \|\mathbf{p}\|_\infty \leq 1 \right\} < \infty \quad (7)$$

does not penalize discontinuous, so it can efficiently preserve image edges. However, one problem in this model is the occurring staircasing effect, i.e., the creation of flat areas separated by jumps.

### 2.3.2. Variable exponent TV-based model (VETV)

In order to overcome the staircasing effect of the ROF model, one of schemes [36–38] uses

$$\min_u \frac{\lambda}{2} \|u - f\|_2^2 + \|\nabla u\|_2^{p(x,y)} \quad (8)$$

to restore image, where  $p(x, y)$  is a smooth monotonically decreasing function such that  $\lim_{(x,y) \rightarrow 0} p(x, y) = 2$  and  $\lim_{(x,y) \rightarrow \infty} p(x, y) = 1$ . This model is a variable exponent model. It chooses diffusion speed through exponent and then can reduce the staircasing effect. One of the choices of  $p(x, y)$  is

$$p(x, y) = 1 + \frac{1}{1 + \bar{\kappa} |G_{\bar{\sigma}}(x, y) * \nabla f(x, y)|},$$

where  $G_{\bar{\sigma}}(x, y) = \frac{1}{\sqrt{2\pi\bar{\sigma}}} \exp\left(-\frac{\|x\|_2^2 + \|y\|_2^2}{2\bar{\sigma}^2}\right)$  is the Gaussian kernel and  $*$  is the convolution product,  $\bar{\kappa}$  and  $\bar{\sigma}$  are a fixed parameter.<sup>2</sup> Clearly, the image gradient of  $f$  is large in the regions with edges, so the model (8) approximates the ROF model, then the edges can be preserved. In relatively smooth regions, the image gradient of  $f$  is small, so the model (8) approximates isotropic smoothing, then it can process into piecewise smooth regions. In other regions, the diffusion is properly adjusted based on the image  $f$ .

### 2.3.3. High-order TV-based model (HOTV)

By using some properties of fourth-order linear or nonlinear diffusion damps oscillations much faster than second-order diffusion, one of high-order TV models [9,10] as

$$\min_u \frac{\lambda}{2} \|u - f\|_2^2 + \|\nabla^2 u\|_{F,1} \quad (9)$$

has been proposed to overcome the staircasing effect related to the ROF model, where  $\|\nabla^2 u\|_F = \sqrt{u_{xx}^2 + u_{xy}^2 + u_{yx}^2 + u_{yy}^2}$  denotes the Frobenius norm. However, this model often leads to blur the image edge.

### 2.3.4. Total generalized variation-based model (TGV)

As an extensions of the TV-based models, the total generalized variation (TGV)-based models have been used in many fields of image processing. The original definition of the TGV [39] is defined in the framework of the dual formulation incorporating the space of symmetric  $k$ -tensors. In order to more efficiently solve the TGV-based models, the authors in [25] employed the Legendre–Fenchel transformation to transform the original form into the dual form and then proposed a new TGV-based restored model as

$$\min_{u,w} \frac{\lambda}{2} \|u - f\|_2^2 + \alpha_0 \|\nabla u - w\|_{2,1} + \alpha_1 \|\mathcal{E}(w)\|_{F,1}, \quad (10)$$

where  $\alpha_0$  and  $\alpha_1$  are the positive parameters,  $\mathcal{E}(w)$  is defined by

$$\mathcal{E}(w) = \frac{1}{2} \begin{bmatrix} 2\nabla_x w_1 & (\nabla_y w_1 + \nabla_x w_2) \\ (\nabla_y w_1 + \nabla_x w_2) & 2\nabla_y w_2 \end{bmatrix}$$

for  $w = (w_1, w_2)^T$ . The main advantage of TGV model is the ability to reconstruct piecewise polynomial functions. So it has a superior performance in image restoration compared to the classic TV regularization that can only reconstruct piecewise constant functions.

<sup>2</sup> As the convention, we set  $\bar{\kappa} = 4\sqrt{mn}$ , where  $mn$  denotes the image size as  $m \times n$ . For the standard variance  $\bar{\sigma}$ , we only set it to be a constant as  $\bar{\sigma} = 0.5$  in the numerical implementation.

### 2.3.5. Anisotropic total variation filtering model (ATVF)

One of schemes closely related to our proposed model (2) is the anisotropic total variation filtering [20] based on the linear structure tensor of  $u$  or the degraded image  $f$ . Since our proposed model (2) is convex, we only consider the case of choosing  $f$ . Especially, we first need to define the linear structure tensor  $J_\rho(f)$  as

$$J_\rho(f) := \begin{bmatrix} \mathcal{K}_\rho * \nabla_x^2 f & \mathcal{K}_\rho * \nabla_x f \nabla_y f \\ \mathcal{K}_\rho * \nabla_y f \nabla_x f & \mathcal{K}_\rho * \nabla_y^2 f \end{bmatrix},$$

where  $*$  denotes the convolution operator and  $\mathcal{K}_\rho(x, y) := \frac{1}{2\pi\rho^2} \exp\left(-\frac{x^2+y^2}{2\rho^2}\right)$ . Since  $J_\rho(f)$  is positive semi-definite and symmetric, we can get the singular value decomposition as

$$J_\rho(f) = U^T \Lambda U^T, \quad (11)$$

where  $U$  is an orthogonal matrix and  $\Lambda = \text{diag}(\sigma_1, \sigma_2)$  is the diagonal matrix of the order eigenvalues. By setting a threshold operation  $\gamma(\cdot)$ , they then defined a new linear structure tensor [20] as

$$\mathcal{J}_\rho(f) = U^T \Lambda_\gamma U := U^T \text{diag}(\gamma(\sigma_1 - \sigma_2), 1) U.$$

By setting  $\mathcal{A} = U \text{diag}(\sqrt{\gamma(\sigma_1 - \sigma_2)}, 1)$ , then we can transform the model proposed in [20] into the following form

$$\min_u \frac{\lambda}{2} \|u - f\|_2^2 + \|\mathcal{A} \nabla u\|_{2,1}. \quad (12)$$

Obviously, the main difference between our proposed model (2) and the model (12) is that our proposed mainly depends on the edge not the structure tensor.

## 3. The proposed anisotropic TV model

This section mainly focuses on the numerical method to solve the ATV model (2). Since it is nonsmooth and convex, we firstly employ the alternating direction method of multipliers (ADMM) to solve it and then give some convergence analysis. The ADMM was originally introduced in early 1970s [27,28], and has been studied extensively in the field of machine learning, computer vision, image and signal processing, networking. The basic motivation of the ADMM is to first split the original nonsmooth minimization problem into several sub-problems via introducing some auxiliary variables, and then solve each subproblem separately by employing some efficient numerical methods. This method is closely related to many other methods such as dual decomposition, the method of multipliers, Douglas–Rachford splitting, Spingarn’s method of partial inverses, Dykstra’s alternating projections, see for the instance the work in [39].

### 3.1. ATV-model

In order to use the ADMM, we need to firstly transform the proposed model (2) into the following constrained optimization problem

$$\begin{cases} \min_{w,v,u} \frac{\lambda}{2} \|u - f\|_2^2 + \|w\|_{2,1} \\ \text{s.t. } w := (w_1, w_2)^T = T v \text{ and } v := (v_1, v_2)^T = \nabla u. \end{cases} \quad (13)$$

The aim of this transformation is to move out two  $\nabla$  and  $T$  from the  $\ell^1$ -norm in order to use some efficient numerical methods. Then, based on the augmented Lagrange (AL) method, we rewrite the problem (13) as the saddle-point problem

$$\min_{w,v,u} \max_{\alpha,\beta} \mathcal{L}(w, v, u, \alpha, \beta) = \frac{\lambda}{2} \|u - f\|_2^2 + \langle \alpha, v - \nabla u \rangle + \frac{\gamma_1}{2} \|v - \nabla u\|_2^2 + \|w\|_{2,1} + \langle \beta, w - T v \rangle + \frac{\gamma_2}{2} \|w - T v\|_2^2, \quad (14)$$

where  $\gamma_1 > 0$ ,  $\gamma_2 > 0$  are the penalty parameters and  $\alpha = (\alpha_1, \alpha_2) > 0$ ,  $\beta = (\beta_1, \beta_2) > 0$  are the Lagrange multipliers (i.e., dual variables) responding to the splitting variables  $w$  and  $v$ , respectively. In the problem (14), it includes five subvariables, so we need to solve one of subvariables and simultaneously fix others when using the ADMM as



**Algorithm 1:** ADMM to solve the problem (14)

1. Initialize:  $\lambda > 0$  and choose original value of  $u^0$ ,  $\mathbf{v}^0$ ,  $\alpha^0$ , and  $\beta^0$ ;
2. For  $k = 1, 2, \dots$ , obtaining  $(u^{k+1}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1})$  by

$$\begin{cases} \mathbf{w}^{k+1} := \underset{\mathbf{w}}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}, \mathbf{v}^k, u^k, \alpha^k, \beta^k), & (a) \\ \mathbf{v}^{k+1} := \underset{\mathbf{v}}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}^{k+1}, \mathbf{v}, u^k, \alpha^k, \beta^k), & (b) \\ u^{k+1} := \underset{u}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}^{k+1}, \mathbf{v}^{k+1}, u, \alpha^k, \beta^k), & (c) \\ \alpha^{k+1} = \alpha^k + \gamma_1 (\mathbf{v}^{k+1} - \nabla u^{k+1}), & (d) \\ \beta^{k+1} = \beta^k + \gamma_2 (\mathbf{w}^{k+1} - \mathbf{T} \mathbf{v}^{k+1}). & (e) \end{cases} \quad (15)$$

3. End for until stopping rule meets;
4. Set  $u := u^{k+1}$  as the denoising image.

In the following, we consider how to solve subproblems (15)(a)–(c).

- For the subproblem (15)(a), it is the classic  $\ell^2 - \ell^1$  problem as

$$\mathbf{w}^{k+1} := \underset{\mathbf{w}}{\operatorname{argmin}} \frac{\gamma_2}{2} \left\| \mathbf{w} - \left( \mathbf{T} \mathbf{v}^k - \frac{\beta^k}{\gamma_2} \right) \right\|_2^2 + \|\mathbf{w}\|_{2,1}.$$

By using the soft-thresholding operator (5), its closed-form solution can be written as

$$\mathbf{w}^{k+1} = \operatorname{Prox}_{\|\cdot\|_{2,1}} \left( \left\| \mathbf{T} \mathbf{v}^k - \frac{\beta^k}{\gamma_2} \right\|_{2,1}, \gamma_2 \right). \quad (16)$$

- For the subproblem (15)(b), it is a smoothing optimization problem. So its optimal solution  $\mathbf{v}^{k+1}$  satisfies the following linear system

$$\begin{cases} \alpha_1^k + \gamma_1 (v_1^{k+1} - \nabla_x u^k) - t_1 \beta_1^k + \gamma_2 t_1 (t_1 v_1^{k+1} - w_1^{k+1}) = 0, \\ \alpha_2^k + \gamma_1 (v_2^{k+1} - \nabla_y u^k) - t_2 \beta_2^k + \gamma_2 t_2 (t_2 v_2^{k+1} - w_2^{k+1}) = 0. \end{cases} \quad (17)$$

With a simple computation, its explicit solution can be obtained by

$$\begin{cases} v_1^{k+1} = \frac{\gamma_1 \nabla_x u^k + t_1 \beta_1^k + \gamma_2 t_1 w_1^{k+1} - \alpha_1^k}{\gamma_1 + \gamma_2 t_1^2}, \\ v_2^{k+1} = \frac{\gamma_1 \nabla_y u^k + t_2 \beta_2^k + \gamma_2 t_2 w_2^{k+1} - \alpha_2^k}{\gamma_1 + \gamma_2 t_2^2}. \end{cases} \quad (18)$$

- For the subproblem (15)(c), it is a least squares optimal problem and the corresponding Euler–Lagrange equation is

$$(\lambda I - \gamma_1 \Delta) u^{k+1} = \lambda f - \operatorname{div} \alpha^k - \gamma_1 \operatorname{div} \mathbf{v}^{k+1}. \quad (19)$$

For this linear equation system, different boundary conditions lead to different numerical methods. In fact, we shall notice that the Laplacian operator  $-\Delta$  is positive semi-definite when using the zero Neumann boundary condition or the zero Dirichlet boundary condition. In this case, we can use the preconditioned conjugate gradient (PCG) method [40] to solve it since  $\lambda$  and  $\gamma_1$  are both positive scalars, the matrix on the left-hand-side of the above system is symmetric and positive definite. However here we assume that the boundary condition is periodic, so the problem (19) can be efficiently solved by

$$u^{k+1} = \mathcal{F}^{-1} \left( \frac{\lambda \mathcal{F}(f) - \mathcal{F}(\operatorname{div} \alpha^k) - \gamma_1 \mathcal{F}(\operatorname{div} \mathbf{v}^{k+1})}{\lambda \mathcal{F}(I) - \gamma_1 \mathcal{F}(\Delta)} \right), \quad (20)$$

where  $\mathcal{F}(\cdot)$  and  $\mathcal{F}^{-1}(\cdot)$  are the fast Fourier transformation (FFT) and the inverse transformation, respectively.

In this following we want to prove the convergence of Algorithm 1, which states that the proposed iterative scheme (15)(a)–(e) converges to the saddle point of the min–max problem (14) and also to the solution of original problem (2). The main idea of the following proof has been used in [41–43]. However, the decoupled operations for

these methods usually distributes in different terms, where there only includes an operator in every decoupled term. This is different to the proposed model (2), where two operators  $\nabla$  and  $\mathbf{T}$  are coupled into the  $\ell^1$ -norm. So, for the aim of the completion, we give the proof details in the following.

**Theorem 3.1.** *If there exists a saddle point  $(\mathbf{w}^*, \mathbf{v}^*, u^*, \alpha^*, \beta^*)$  of the problem (14), then it is unique and the sequence  $\{(\mathbf{w}^k, \mathbf{v}^k, u^k, \alpha^k, \beta^k)\}$  generated by the scheme (15)(a)–(e) converges to the solution of (13) for every choosing original point  $(\mathbf{w}^0, \mathbf{v}^0, u^0, \alpha^0, \beta^0)$ .*

**Proof.** See Appendix  $\square$

**Remark 3.1.** Here we need some comments on Algorithm 1 to solve the proposed model (2). In fact, another splitting form can be chosen as

$$\begin{cases} \min_{\tilde{\mathbf{v}}, u} \frac{\lambda}{2} \|u - f\|_2^2 + \|\tilde{\mathbf{v}}\|_{2,1} \\ \text{s.t. } \tilde{\mathbf{v}} := (\tilde{v}_1, \tilde{v}_2)^T = \mathbf{T} \nabla u, \end{cases}$$

which augmented Lagrangian function is

$$\min_{\tilde{\mathbf{v}}, u} \max_{\rho} \frac{\lambda}{2} \|u - f\|_2^2 + \|\tilde{\mathbf{v}}\|_{2,1} + \langle \rho, \tilde{\mathbf{v}} - \mathbf{T} \nabla u \rangle + \frac{\rho}{2} \|\tilde{\mathbf{v}} - \mathbf{T} \nabla u\|_2^2. \quad (21)$$

In the saddle point problem (21), the main attention is focused on how to solve the subproblem  $u$ .

$$u^* = \underset{u}{\operatorname{argmin}} \frac{\lambda}{2} \|u - f\|_2^2 + \langle \rho, \tilde{\mathbf{v}} - \mathbf{T} \nabla u \rangle + \frac{\rho}{2} \|\tilde{\mathbf{v}} - \mathbf{T} \nabla u\|_2^2.$$

To this end, we need the related Euler–Lagrangian equation

$$(\lambda I - \operatorname{div}(\mathbf{T}^T \mathbf{T} \nabla)) u^* = \lambda f + \operatorname{div} \mathbf{T}^T \rho + \rho \operatorname{div} \mathbf{T}^T \tilde{\mathbf{v}}. \quad (22)$$

Here  $\mathbf{T}^T$  denotes the transpose of the matrix operator  $\mathbf{T}$ . It is obvious that we cannot obtain a block circulant matrix with circulant blocks (BCCB) in the left side for all boundary conditions due to the different  $t_1$  and  $t_2$  in  $\mathbf{T}$ . Here we mainly want to use the FFT to solve the related subproblem (19), so our splitting scheme employs two auxiliary variables  $\mathbf{w}$  and  $\mathbf{v}$  in (13) to decouple the operators  $T$  and  $\nabla$ .

#### 4. Numerical comparisons

In this section, some experimental results are conducted to evaluate the performance of the proposed model (ATV) (2) and the numerical method. From various classic and recent state-of-the-art image restoration approaches such as the TV-based methods, learning-based methods and wavelet-based methods, etc., we only focus on the comparisons between the ATV (2) and other TV-based models such as the ROF model (6) used in [4], the variable exponent total variation model (VETV) (8) used in [38], the high-order total variation model (HOTV) (9) used in [9], the total generalized variation model (TGV) (10) used in [11] and the anisotropic total variation filtering model (12) used in [20]. All experiments are performed via using MATLAB (R2017a) on a windows10 (64bit) desktop computer with an Intel Core i7 2.40 GHz processor and 8.0GB of RAM. Furthermore, all of numerical methods will be stopped when the relative difference between two successive iteration satisfies

$$\frac{\|u^k - u^{k-1}\|_2}{\|u^{k-1}\|_2} \leq 10^{-5}$$

or after a maximum number of 500 iterations. Furthermore, testing images are normalized in the range of [0,1] before degrading them and then use Matlab's function 'imnoise' to add the white Gaussian noise to them. Here we set the Greek alphabet  $\sigma$  as the standard deviation of the noise. The restored quality is measured by SNR (signal-noise ratio) and SSIM (structural similarity index). It was demonstrated that the higher value of SSIM or SNR implies a better restoration corresponds to subjective quality of visual perception. Here we still use Matlab's functions *snr* and *ssim* with the standard parameter to compute the corresponding values.



Fig. 3. Eight testing images with the size  $256 \times 256$  are used in the numerical implementations.

#### 4.1. Choosing parameters

The restored effectiveness depends on the regularization parameter  $\lambda$  that controls how much filtering is introduced by the regularization. Often the key issue in connection with these methods is to find a regularization parameter that gives a good balance, filtering out enough noise without losing too much information in the computed solution. So we want to consider both the SNR and visual impression. Although the SNR is a good indicator of reconstruction quality, it is not always true that a restored image with a higher SNR with respect to the original image is more visually appealing than another with a lower SNR. Our experience was that the best visual quality for a given image and method usually occurred at a  $\lambda$  slightly larger than the one giving the highest SNR, which tended to appear noisier than one expects. Specifically, we set the related parameters into a bigger range as  $[a_0, b_0]$ , and then find a suitable subset as  $[a_1, b_1] \subset [a_0, b_0]$ . In the next we find a more suitable subset in  $[a_2, b_2]$ . By repeating this strategy, we can find a suitable range  $[a_n, b_n]$ , where the difference between the successive SNR is below 0.001 when setting  $\lambda = a_n$  and  $\lambda = b_n$ . At this moment, we set the final value of  $\lambda$  to be  $(a_n + b_n)/2$ .

Except for the regularization parameter  $\lambda$ , there are some penalty parameters in the ADMM such as  $\gamma_i$ , where  $i$  depends on the number of introduced variables. For example, we set  $i = 2$  for our proposed model and  $i = 1$  for the ROF and the HOTV. To the TGV, there are two balance parameter  $\alpha_0$  and  $\alpha_1$ , here we set  $\alpha_0 = 0.15$  and  $\alpha_1 = 0.8$ . In fact, it is fair for choosing different values for them, but it is the emphasis point for our paper to some comparisons among aforementioned models.

#### 4.2. Numerical implementations

We compare our method and competing approaches on eight high-resolution images as shown in Fig. 3. These images have been used in a previous study evaluating a denoising approach with the TV-based models such as the ROF model (6), the VETV model (8), the HOTV model (9), the TGV model (10), the ATVF (12) and are selected based on their resolution and rich features.

##### 4.2.1. Artificial image

We first consider to restore the piecewise constant image as ‘Texmos’, which is degraded by different noisy levels as  $\sigma = 0.01, 0.05, 0.1$ . Related data of SNR and SSIM can be found from Table 1. It can be seen that our proposed model (2) performs more robust than other related TV-based models except for the high noisy image. At the same time, ATVF (12) is more effective for the high noisy image. In addition, we also found that the ROF model (6) performs slightly better than others high-order TV-based models (8)–(10). This illustrates that the

Table 1

Related data for restoring the degraded image of Texmos with the different noisy level.

Noise	0.01			0.05			0.1		
Model	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	12.6	24.0244	0.8551	6.4	17.1302	0.7137	5.1	14.0129	0.6163
VETV	10.7	22.9916	0.7907	5.7	16.8490	0.6694	4.4	13.9547	0.6012
HOTV	32.6	22.2242	0.7371	14.3	16.2267	0.5672	10.3	13.6323	0.4913
TGV	4.9	22.2185	0.7374	2.3	16.2066	0.5500	1.5	13.6380	0.6790
ATVF	12.7	25.3032	0.8603	4.7	18.2980	0.7767	3.7	<b>15.2308</b>	<b>0.7088</b>
ATV	6.9	<b>26.3270</b>	<b>0.8979</b>	2.9	<b>18.4092</b>	<b>0.7832</b>	2.0	15.1011	0.7038

TV-based model is suitable to restore the piecewise constant image since this norm tends to the sparse in the framework of the gradient transformation. In order to efficiently show the restored abilities, we choose the 90th column of the restored images as shown in Fig. 4. We can observe from the slice plots that our proposed model (2) leads to the restored images with neat edges and correct amplitude. These facts are more obvious in edges of large scale constant regions and the low noisy image. These facts are due to the diffusion tending to the main direction when choosing the adaptive weighed matrix  $\mathbf{T}$  to the gradient operator  $\nabla$  in the proposed model (2).

##### 4.2.2. Benchmark images

Since many images are not piecewise constant images surrounded by edges, we in the following test four benchmark images as Lena, Cameraman, Butterfly and Peppers as shown in Figs. 5 and 6, where Lena is comprised by more smoothing or detailed regions and Cameraman includes many approximated cartoon regions. So these images are suitable to evaluate our proposed model. To this end, we generate testing data by adding the noise with as  $\sigma = 0.01, 0.05, 0.1$ . On these degraded images, we evaluate aforementioned TV-based models. As we can see from Table 2 that the proposed model (2) and the ATVF (12) are more effective for the small details like edges and lines. Especially, the restored abilities of our proposed model (2) is more robust for the low noisy image (Lena, Butterfly, Peppers) or the approximated cartoon image (Cameraman). To also give a visual impression of these comparisons among different models, we depict the respective results for these two images, where we only show a part of the restored image as shown in the first line of Figs. 5 and 6. Since the differences between the restored  $u$  and the original clean image  $\bar{u}$  are best visible in full resolution, we choose a close-up of the head region of Lena, Cameraman, and Butterfly, Peppers. Obviously, it is easy to find that our proposed model can efficiently keep edges and details. These facts are easy to be observed in the differences shown in the second line of Figs. 5 and 6 since our proposed method gives darker colorbar also includes less mottled effect.

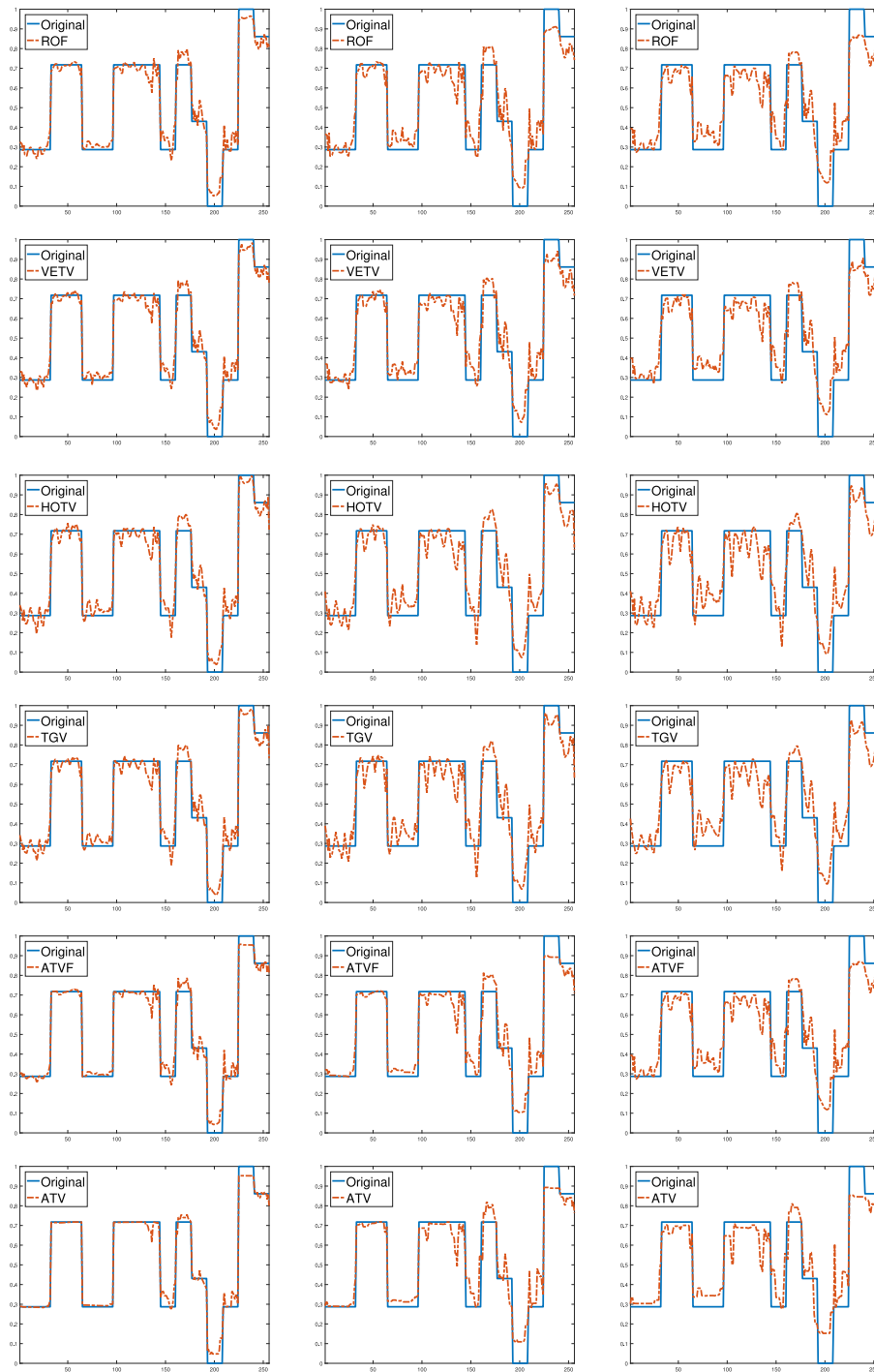


Fig. 4. The comparisons of the 90th column plots generated from the restored image and the original image. Plots from top to bottom are restored results by using models as ROF, VETV, HOTV, TGV, ATVF, and the proposed model (ATV). 1st column: restored image with  $\sigma = 0.01$ ; 2st column: restored image with  $\sigma = 0.05$ ; 3st column: restored image with  $\sigma = 0.1$ .

#### 4.2.3. Extend to remained images.

In this subsection, we test other three images and find that the same conclusion about restoration capability can be obtained from Table 3. This still shows that the our proposed model (2) can achieve well enough restored results for any image types at different noise density. Fig. 7 shows output images of assorted images only for noise density with  $\sigma = 0.05$ . It can be seen that the proposed model (2) successfully preserve the details while removing the most noise and only scattered tiny noise spots left. In order to show the restored abilities, we simultaneously plot the contour curves of the restored image.

These plots clearly reveal that the proposed model, as compared to other gradient-based models, generates images with smoother surfaces, and with smooth and uniformly separated level lines, especially visible around the approximated constant regions and clear edges. Now we discuss here the CPU-time of the aforementioned denoising models. To get some insights, we plot them in Fig. 8. As can be seen all of models are very fast in getting the numerical solution. Specially, the ROF is the fastest of them and our proposed also spends short CPU time. The reason is that the ROF and the ATV are relatively simple when employing the numerical methods to solve them. On the other

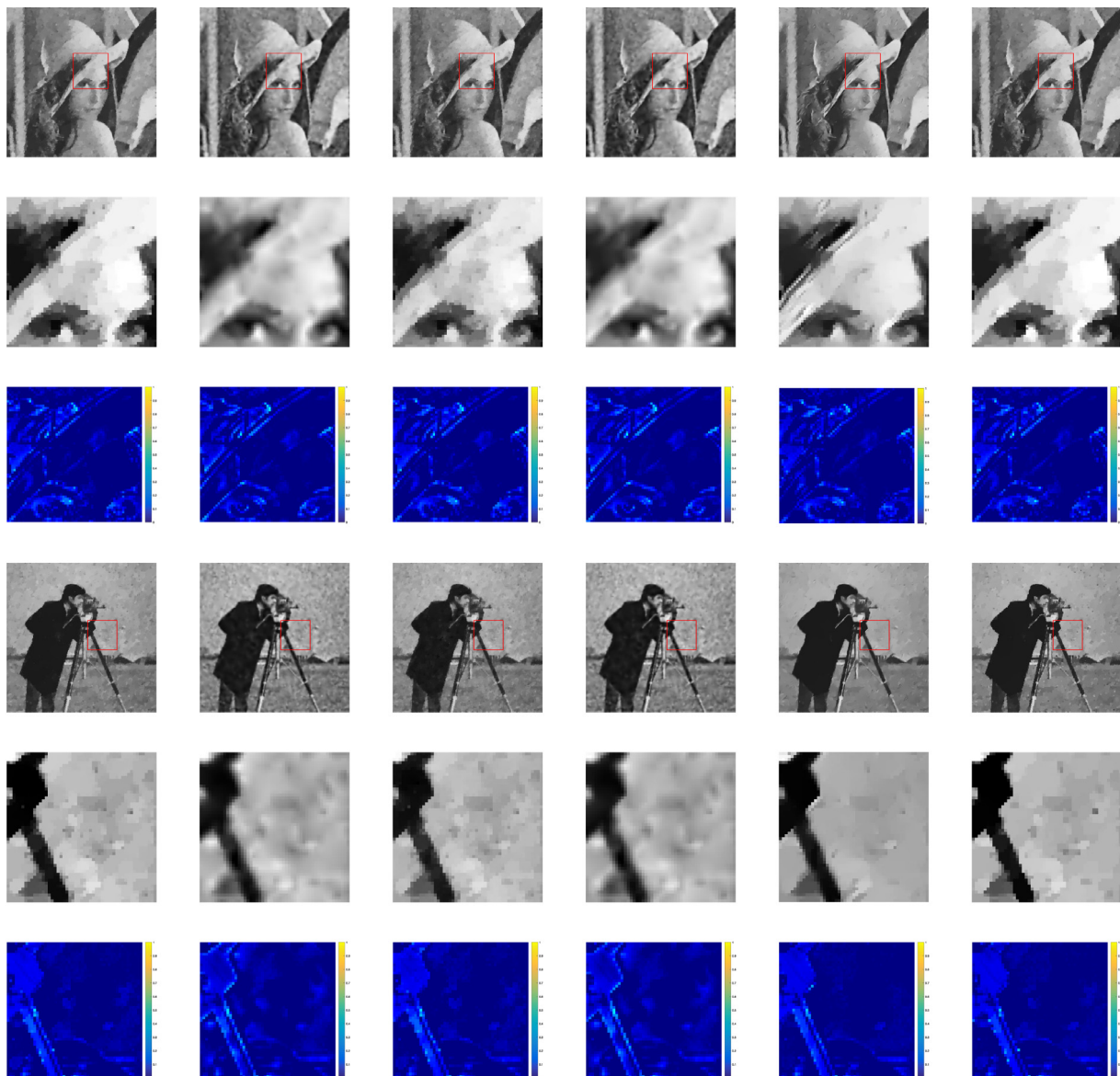


Fig. 5. Denoising images by using different models. Images from left to right are restored results by using the models as ROF, VETV, HOTV, TGV, ATVF, and the proposed model (ATV). The first row to every image are a part of the restored image in order to show the restored abilities. The second rows to every image are part of the restored images. The third row to every image are the difference between restored images and the original clean image. The colorbar shows some more efficient restorations if the color is more shaded. Here we only show the related results by restoring the noisy image with  $\sigma = 0.05$ .

hand, in order to evaluate the convergence of numerical methods, Fig. 9 only shows the convergence curves of SNR for restoring noisy images with  $\sigma = 0.05$ . It is easy to find that these curves have the nearly same restored ability. However, our proposed model can obtained the highest SNR than other models. This also implies that our proposed model is more robust than other models when restoring degraded images.

## 5. Conclusion

In this paper, a new noise removal model was proposed via coupling the local weighted matrix with the gradient operator. In order to describe the local features via the weighted matrix  $T = \text{diag}(t_1, t_2)$ , we use the edge indicator function to obtain the weighted parameters  $t_1$  and  $t_2$  by respectively using the gradient information of  $\nabla_x f$  and  $\nabla_y f$ . Since the proposed model can be fall into the framework of the ADMM, so the convergence of the numerical method can be efficiently kept. The experimental results exhibited some significant

improvements in performance over several other related gradient-based methods. In addition, we also noticed that the weighed parameter  $T$  closely depends on image structure and noise intensity. This motives us to find some preprocessing scheme to improve our proposed model via weakening the effect of the noise and the image structure in our future work.

## Acknowledgments

The authors would like to thank the editor and the reviewers for their valuable comments, suggestions, and making this paper much improved. This work was partially supported by the National Basic Research Program of China (973 Program) (No. 2015CB856003), Natural Science Foundation of China (Grant No. 11501301), the 1311 Talent Plan of NUPT, China and Hunan Provincial Key Laboratory of Mathematical Modeling and Analysis in Engineering (Changsha University of Science & Technology, China).



**Table 2**  
Comparisons of SNR and SSIM by using different denoising models.

Noise	0.01			0.05			0.1		
Image	Fig(b): Lena								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	14.4	21.9949	0.7921	6.1	17.7309	0.6599	4.8	15.6197	0.5829
VETV	11.7	21.7629	0.7660	5.4	17.7051	0.6455	4.2	15.6169	0.5811
HOTV	27.7	22.1256	0.7854	9.7	17.9046	0.6554	6.7	15.8970	0.5971
TGV	4.2	22.1698	0.7867	1.5	17.9272	0.6524	1.0	<b>15.9215</b>	<b>0.5991</b>
ATVF	13.9	<b>22.8128</b>	<b>0.8198</b>	4.8	<b>18.1367</b>	<b>0.6699</b>	3.4	15.7406	0.5868
ATV	12.3	22.2767	0.8035	5.3	18.0352	<b>0.6699</b>	3.1	15.6536	0.5930
Image	Fig(c): Cameraman								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	15.0	21.3905	0.7399	6.3	16.6918	0.5850	4.7	14.5344	0.5179
VETV	12.2	20.9429	0.6957	5.4	16.5339	0.5665	4.1	14.4632	0.5075
HOTV	30.6	21.0614	0.7057	10.7	16.4046	0.5370	6.7	14.4140	0.4923
TGV	4.8	21.0501	0.6991	1.6	16.3996	0.5377	1.6	14.4179	0.4936
ATVF	14.8	<b>22.1273</b>	0.7608	4.6	<b>17.3020</b>	0.6012	2.9	<b>14.7847</b>	0.5354
ATV	12.3	21.9328	<b>0.7637</b>	4.0	17.1081	<b>0.6135</b>	2.7	14.7323	<b>0.5428</b>
Image	Fig(d): Butterfly								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	21.9	23.9426	0.9059	11.4	18.3312	0.7876	8.9	15.5353	0.7007
VETV	19.0	23.2525	0.8584	10.0	18.0586	0.7600	7.8	15.4515	0.6797
HOTV	51.7	23.7777	0.8820	22.3	18.6079	0.7666	16.5	15.9797	0.6899
TGV	8.1	23.6260	0.8753	3.4	18.4881	0.7657	2.5	15.9011	0.6840
ATVF	22.1	24.6443	0.9177	9.1	<b>19.3652</b>	0.8202	6.4	<b>16.3034</b>	0.7329
ATV	17.3	<b>24.7020</b>	<b>0.9185</b>	7.0	19.1240	<b>0.8273</b>	5.3	16.1619	<b>0.7377</b>
Image	Fig(e): Peppers								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	13.7	22.9078	0.8186	6.1	18.2672	0.6870	4.7	15.9573	0.6057
VETV	10.4	22.7127	0.7989	5.3	18.2665	0.6792	4.6	15.9147	0.5672
HOTV	25.5	22.9410	0.7971	9.5	18.6965	0.6975	6.5	16.4319	0.6364
TGV	4.0	23.2312	0.8253	1.6	18.6845	0.6890	1.0	<b>16.4322</b>	<b>0.6422</b>
ATVF	13.0	<b>23.8367</b>	<b>0.8437</b>	4.3	18.6487	0.6952	3.0	15.9539	0.6174
ATV	11.4	23.5088	0.8381	5.1	<b>18.7675</b>	<b>0.7058</b>	4.1	16.2374	0.6333

**Table 3**  
Comparisons of SNR and SSIM by using different denoising models.

Noise	0.01			0.05			0.1		
Image	Fig(f): Parrot								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	21.1	25.5509	0.8604	9.5	20.3753	0.7593	7.3	17.6557	0.7011
WTV	15.2	24.9507	0.8234	7.7	20.1752	0.7362	6.0	17.5595	0.6909
HOTV	54.7	25.3393	0.8043	15.7	20.5567	0.7513	11.6	17.5915	0.6271
TGV	5.8	25.6615	0.8574	2.4	20.5487	0.7478	1.7	17.8759	0.6968
ATVF	20.9	<b>26.4442</b>	<b>0.8773</b>	7.4	<b>21.1924</b>	0.7902	5.1	<b>18.2241</b>	0.7245
ATV	17.2	26.0356	0.8740	6.3	20.8612	<b>0.7948</b>	5.9	18.0534	<b>0.7277</b>
Image	Fig(g): Phantom								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	22.7	16.4138	0.4618	11.9	9.6729	0.3626	9.2	7.0388	0.3188
WTV	16.6	15.6318	0.4203	10.7	9.3947	0.3086	8.5	6.8831	0.2654
HOTV	43.7	15.6695	0.4124	28.0	9.2482	0.2877	20.4	6.7871	0.2445
TGV	8.9	15.5935	0.4057	4.2	9.2160	0.2879	3.1	6.7715	0.2420
ATVF	25.8	16.5656	0.4486	9.3	9.9784	0.4084	7.4	7.3408	0.3443
ATV	15.6	<b>17.2225</b>	<b>0.4816</b>	5.5	<b>10.3870</b>	<b>0.4241</b>	3.8	<b>7.6208</b>	<b>0.3828</b>
Image	Fig(h): Frame								
Models	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM	$\lambda$	SNR	SSIM
TV	11.0	29.5008	0.8538	5.1	22.7030	0.8064	4.3	18.9340	0.7581
WTV	7.6	29.1869	0.8475	4.4	22.5778	0.7899	8.5	18.8507	0.7588
HOTV	18.3	29.0157	0.8343	6.8	22.5954	0.7829	5.0	18.9849	0.7514
TGV	2.8	29.0283	0.8338	1.0	22.6022	0.7855	0.8	18.9864	0.7432
ATVF	9.4	30.3719	0.8607	3.2	23.2058	0.8038	2.3	18.9791	0.7417
ATV	8.6	<b>30.4019</b>	<b>0.8616</b>	3.1	<b>23.3126</b>	<b>0.8255</b>	2.4	<b>19.3001</b>	<b>0.7881</b>

### Conflict of interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence

our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

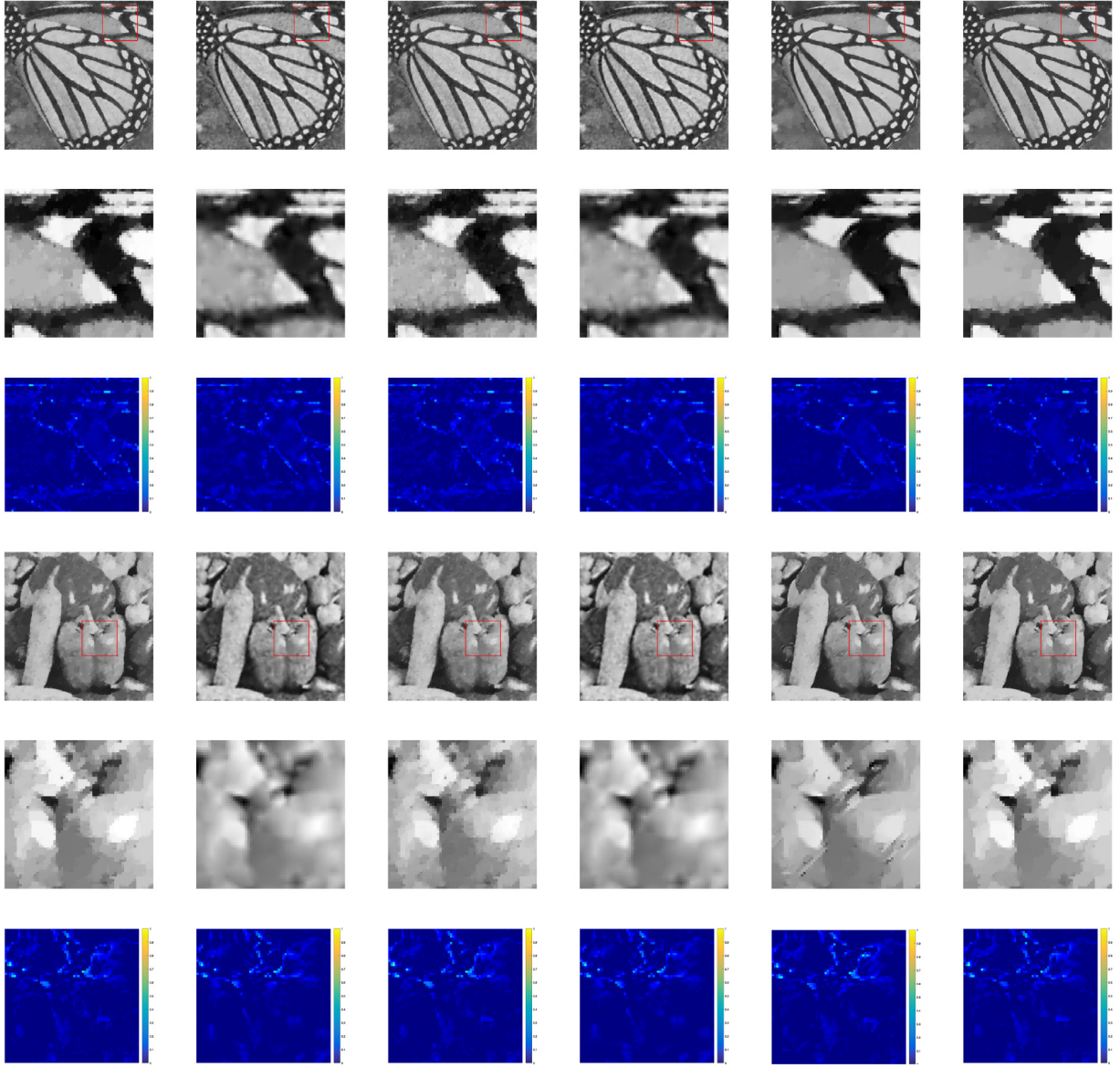


Fig. 6. Denoising images by using different models. Images from left to right are restored results by using the models as ROF, VETV, HOTV, TGV, ATVF, and the proposed model (ATV). The first row to every image are a part of the restored image in order to show the restored abilities. The second rows to every image are part of the restored images. The third rows to every image are the difference between restored images and the original clean image. The colorbar shows some more efficient restorations if the color is more shaded. Here we only show the related results by restoring the noisy image with  $\sigma = 0.05$ .

### Appendix. Proof of Theorem 3.1

**Proof.** In order to prove the convergence of the proposed algorithm, we firstly notice that the sequence  $\left\{ \left( \mathbf{w}^{k+1}, \mathbf{v}^{k+1}, u^{k+1}, \alpha^{k+1}, \beta^{k+1} \right) \right\}$  generated by the subproblems (15)(a)–(e) satisfies

$$\begin{cases} 0 = r^{k+1} + \beta^k + \gamma_2 (\mathbf{w}^{k+1} - \mathbf{T}\mathbf{v}^k), & (a) \\ 0 = \alpha^k + \gamma_1 (\mathbf{v}^{k+1} - \nabla u^k) - \mathbf{T}^* \beta^k - \gamma_2 \mathbf{T}^* (\mathbf{w}^{k+1} - \mathbf{T}\mathbf{v}^{k+1}), & (b) \\ 0 = \lambda (u^{k+1} - f) + \text{div} \alpha^k + \gamma_1 \text{div} (\mathbf{v}^{k+1} - \nabla u^{k+1}), & (c) \\ \alpha^{k+1} = \alpha^k + \gamma_1 (\mathbf{v}^{k+1} - \nabla u^{k+1}), & (d) \\ \beta^{k+1} = \beta^k + \gamma_2 (\mathbf{w}^{k+1} - \mathbf{T}\mathbf{v}^{k+1}), & (e) \end{cases} \quad (23)$$

where  $\mathbf{T}^*$  and  $\text{div}$  denote the adjoint operators of  $\mathbf{T}$  and  $\nabla$ , respectively, and  $r^{k+1} \in \partial \|\mathbf{w}^{k+1}\|_{2,1}$ . Here  $\partial \|\cdot\|_{2,1}$  denotes the subdifferential set (see [32] for the definition). From the assumption that the point  $(\mathbf{w}^*, \mathbf{v}^*, u^*, \alpha^*, \beta^*)$  is a saddle point of the problem (14), we also obtain

that

$$\begin{cases} 0 = r^* + \beta^* + \gamma_2 (\mathbf{w}^* - \mathbf{T}\mathbf{v}^*), & (a) \\ 0 = \alpha^* + \gamma_1 (\mathbf{v}^* - \nabla u^*) - \mathbf{T}^* \beta^* - \gamma_2 \mathbf{T}^* (\mathbf{w}^* - \mathbf{T}\mathbf{v}^*), & (b) \\ 0 = \lambda (u^* - f) + \text{div} \alpha^* + \gamma_1 \text{div} (\mathbf{v}^* - \nabla u^*), & (c) \\ \alpha^* = \alpha^* + \gamma_1 (\mathbf{v}^* - \nabla u^*), & (d) \\ \beta^* = \beta^* + \gamma_2 (\mathbf{w}^* - \mathbf{T}\mathbf{v}^*), & (e) \end{cases} \quad (24)$$

where  $r^* \in \partial (\|\mathbf{w}^{k+1}\|_{2,1})$ . Set  $r_e^k = r^k - r^*$ ,  $\mathbf{w}_e^k = \mathbf{w}^k - \mathbf{w}^*$ ,  $\mathbf{v}_e^k = \mathbf{v}^k - \mathbf{v}^*$ ,  $u_e^k = u^k - u^*$ ,  $\alpha_e^k = \alpha^k - \alpha^*$ ,  $\beta_e^k = \beta^k - \beta^*$ , subtracting (24)(a)–(e) from (23)(a)–(e) and then respectively taking the inner product  $\langle \cdot, \cdot \rangle$  of the left- and right-hand sides by  $\mathbf{w}_e^{k+1}$ ,  $\mathbf{v}_e^{k+1}$ ,  $u_e^{k+1}$ ,  $\alpha_e^k$ ,  $\beta_e^k$ , we

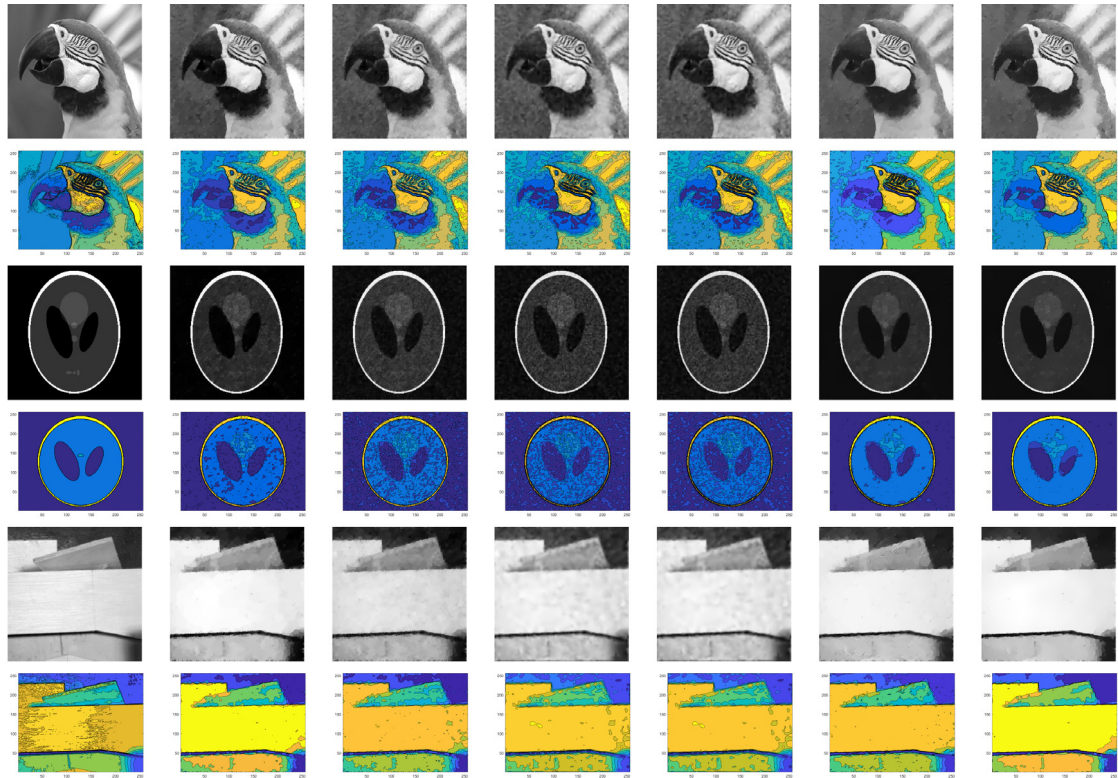


Fig. 7. Denoising images by using different models. Images from left to right are original images and the denoising images by using ROF, VETV, HOTV, TGV, ATVF, and the proposed model (ATV). The first rows to every image are the restored image. The second rows to every image are the contour plots of restored images. Here we only show the related results by restoring the noisy image with  $\sigma = 0.05$ .

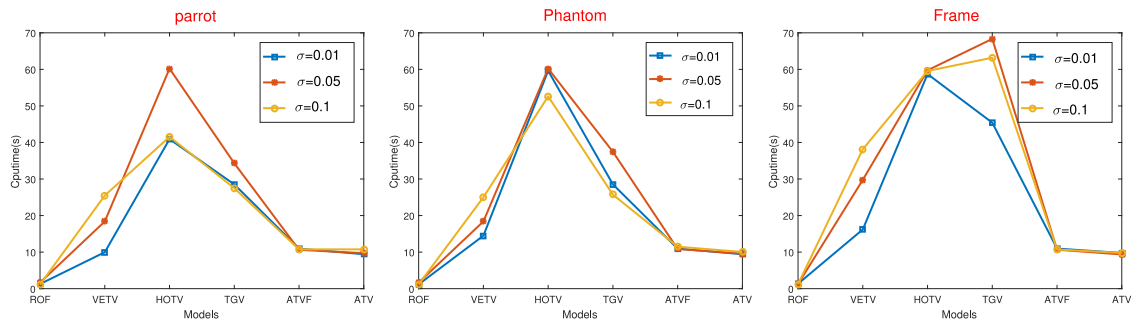


Fig. 8. Curves of the CPU time of ROF, VETV, HOTV, TGV, ATVF, and the proposed model (ATV).

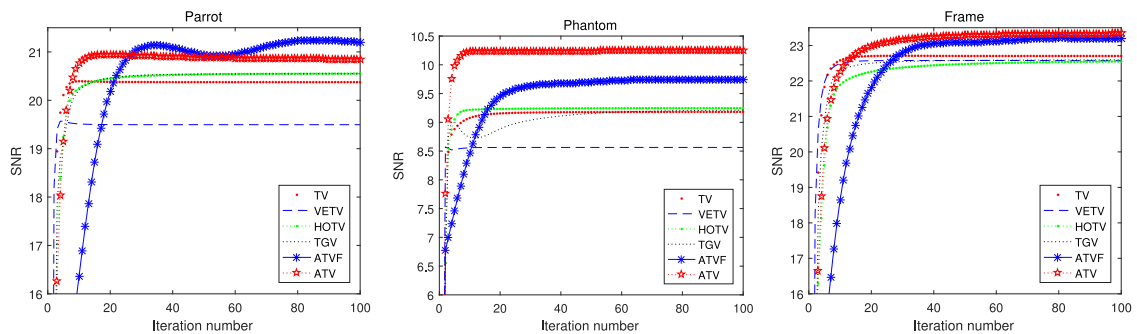


Fig. 9. Curves of the SNR VS the iteration only to the case for the noise level as  $\sigma = 0.05$ .

have

$$\begin{cases} 0 = \langle r_e^{k+1}, w_e^{k+1} \rangle + \langle \beta_e^k, w_e^{k+1} \rangle + \gamma_2 \|w_e^{k+1}\|_2^2 - \gamma_2 \langle \mathbf{T}v_e^k, w_e^{k+1} \rangle, & (a) \\ 0 = \langle \alpha_e^k, v_e^{k+1} \rangle + \gamma_1 \|v_e^{k+1}\|_2^2 - \langle \mathbf{T}^* \beta_e^k, v_e^{k+1} \rangle - \gamma_1 \langle \nabla u_e^k, v_e^{k+1} \rangle \\ \quad - \gamma_2 \langle \mathbf{T}^* w_e^{k+1}, v_e^{k+1} \rangle + \gamma_2 \|\mathbf{T}v_e^{k+1}\|_2^2, & (b) \\ 0 = \lambda \|u_e^{k+1}\|_2^2 + \gamma_1 \|\nabla u_e^{k+1}\|_2^2 + \gamma_1 \langle \text{div} v_e^{k+1}, u_e^{k+1} \rangle + \langle \text{div} \alpha_e^k, u_e^{k+1} \rangle, & (c) \\ \langle \alpha_e^{k+1}, \alpha_e^k \rangle = \|\alpha_e^k\|_2^2 + \gamma_1 \langle v_e^{k+1} - \nabla u_e^{k+1}, \alpha_e^k \rangle, & (d) \\ \langle \beta_e^{k+1}, \beta_e^k \rangle = \|\beta_e^k\|_2^2 + \gamma_2 \langle w_e^{k+1} - \mathbf{T}v_e^{k+1}, \beta_e^k \rangle. & (e) \end{cases} \quad (25)$$

Using the fact

$$\begin{cases} \frac{1}{2\gamma_1} [\|\alpha_e^k\|_2^2 - \|\alpha_e^{k+1}\|_2^2] = \langle \nabla u_e^{k+1} - v_e^{k+1}, \alpha_e^k \rangle - \frac{\gamma_1}{2} \|v_e^{k+1} - \nabla u_e^{k+1}\|_2^2 & (a) \\ \frac{1}{2\gamma_2} [\|\beta_e^k\|_2^2 - \|\beta_e^{k+1}\|_2^2] = \langle \mathbf{T}v_e^{k+1} - w_e^{k+1}, \beta_e^k \rangle - \frac{\gamma_2}{2} \|\mathbf{T}v_e^{k+1} - w_e^{k+1}\|_2^2 & (b) \end{cases} \quad (26)$$

and sum them with (25)(a)–(c), we can obtain

$$\begin{aligned} & \frac{1}{2\gamma_1} [\|\alpha_e^k\|_2^2 - \|\alpha_e^{k+1}\|_2^2] + \frac{1}{2\gamma_2} [\|\beta_e^k\|_2^2 - \|\beta_e^{k+1}\|_2^2] \\ &= \langle r_e^{k+1}, w_e^{k+1} \rangle + \langle \beta_e^k, w_e^{k+1} \rangle + \gamma_2 \|w_e^{k+1}\|_2^2 - \gamma_2 \langle \mathbf{T}v_e^k, w_e^{k+1} \rangle + \langle \alpha_e^k, v_e^{k+1} \rangle \\ & \quad + \gamma_1 \|v_e^{k+1}\|_2^2 - \langle \mathbf{T}^* \beta_e^k, v_e^{k+1} \rangle - \gamma_1 \langle \nabla u_e^k, v_e^{k+1} \rangle - \gamma_2 \langle \mathbf{T}^* w_e^{k+1}, v_e^{k+1} \rangle \\ & \quad + \gamma_2 \|\mathbf{T}v_e^{k+1}\|_2^2 + \lambda \|u_e^{k+1}\|_2^2 + \gamma_1 \|\nabla u_e^{k+1}\|_2^2 + \gamma_1 \langle \text{div} v_e^{k+1}, u_e^{k+1} \rangle \\ & \quad + \langle \text{div} \alpha_e^k, u_e^{k+1} \rangle \\ & \quad + \langle \nabla u_e^{k+1} - v_e^{k+1}, \alpha_e^k \rangle - \frac{\gamma_1}{2} \|v_e^{k+1} - \nabla u_e^{k+1}\|_2^2 + \langle \mathbf{T}v_e^{k+1} - w_e^{k+1}, \beta_e^k \rangle \\ & \quad - \frac{\gamma_2}{2} \|\mathbf{T}v_e^{k+1} - w_e^{k+1}\|_2^2 \end{aligned} \quad (23a)$$

$$\begin{aligned} &= \langle r_e^{k+1}, w_e^{k+1} \rangle + \lambda \|u_e^{k+1}\|_2^2 + \frac{\gamma_1}{2} \|v_e^{k+1} - \nabla u_e^{k+1}\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^{k+1} - w_e^{k+1}\|_2^2 \\ & \quad + \gamma_2 \langle w_e^{k+1}, \mathbf{T}v_e^{k+1} - \mathbf{T}v_e^k \rangle \\ & \quad + \gamma_1 \langle \nabla u_e^{k+1} - \nabla u_e^k, v_e^{k+1} \rangle \end{aligned} \quad (23b)$$

$$\begin{aligned} &= \langle r_e^{k+1}, w_e^{k+1} \rangle + \lambda \|u_e^{k+1}\|_2^2 + \frac{\gamma_1}{2} \|v_e^{k+1} - \nabla u_e^{k+1}\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^{k+1} - w_e^{k+1}\|_2^2 \\ & \quad + \frac{\gamma_2}{2} [\|w_e^{k+1}\|_2^2 + \|\mathbf{T}v_e^{k+1}\|_2^2 - \|\mathbf{T}v_e^{k+1} - w_e^{k+1}\|_2^2] \\ & \quad - \frac{\gamma_2}{2} [\|w_e^{k+1}\|_2^2 + \|\mathbf{T}v_e^k\|_2^2 - \|\mathbf{T}v_e^k - w_e^{k+1}\|_2^2] \\ & \quad + \frac{\gamma_1}{2} [\|\nabla u_e^{k+1}\|_2^2 + \|\nabla u_e^k\|_2^2 - \|\nabla u_e^{k+1} - \nabla u_e^k\|_2^2] - \frac{\gamma_1}{2} [\|\nabla u_e^k\|_2^2 \\ & \quad + \|\nabla u_e^{k+1}\|_2^2 - \|\nabla u_e^k - \nabla u_e^{k+1}\|_2^2]. \end{aligned} \quad (27)$$

By summing the above equation from  $k = 0$  to  $k = K$ , we get

$$\begin{aligned} & \frac{1}{2\gamma_1} \|\alpha_e^0\|_2^2 - \frac{1}{2\gamma_1} \|\alpha_e^{K+1}\|_2^2 + \frac{1}{2\gamma_2} \|\beta_e^0\|_2^2 - \frac{1}{2\gamma_2} \|\beta_e^{K+1}\|_2^2 + \frac{\gamma_1}{2} \|\nabla u_e^0\|_2^2 \\ & \quad - \frac{\gamma_1}{2} \|\nabla u_e^{K+1}\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^0\|_2^2 - \frac{\gamma_2}{2} \|\mathbf{T}v_e^{K+1}\|_2^2 \\ &= \sum_{k=0}^K \left[ \langle r_e^{k+1}, w_e^{k+1} \rangle + \lambda \|u_e^{k+1}\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^k - w_e^{k+1}\|_2^2 \right. \\ & \quad \left. + \frac{\gamma_1}{2} \|\nabla u_e^k - \nabla u_e^{k+1}\|_2^2 \right]. \end{aligned} \quad (28)$$

Using the fact that the monotonicity of subdifferential as

$$\langle \zeta_x - \zeta_y, x - y \rangle \geq 0 \quad \text{for } \zeta_x \in \partial f(x) \text{ and } \zeta_y \in \partial f(y) \quad (29)$$

and omitting some constant terms, it follows that

$$\begin{aligned} & \frac{1}{2\gamma_1} \|\alpha_e^0\|_2^2 + \frac{1}{2\gamma_2} \|\beta_e^0\|_2^2 + \frac{\gamma_1}{2} \|\nabla u_e^0\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^0\|_2^2 \\ & \geq \sum_{k=0}^K \left[ \lambda \|u_e^{k+1}\|_2^2 + \frac{\gamma_2}{2} \|\mathbf{T}v_e^k - w_e^{k+1}\|_2^2 + \frac{\gamma_1}{2} \|\nabla u_e^k - \nabla u_e^{k+1}\|_2^2 \right]. \end{aligned}$$

Using the boundedness in the left side of above inequation, we can deduce the following fact as

$$\begin{cases} \lim_{k \rightarrow \infty} \|u_e^{k+1}\|_2^2 = 0, & (a) \\ \lim_{k \rightarrow \infty} \|\mathbf{T}v_e^k - w_e^{k+1}\|_2^2 = 0, & (b) \\ \lim_{k \rightarrow \infty} \|v_e^{k+1} - \nabla u_e^k\|_2^2 = 0. & (c) \end{cases} \quad (30)$$

This implies that the sequence  $\{(u^k, v^k, w^k)\}$  generated by the iteration scheme (15)(a)–(c) converges to the saddle point  $\{(u^*, v^*, w^*)\}$ . That is to say

$$\begin{cases} \lim_{k \rightarrow \infty} u^k = u^*, & (a) \\ \lim_{k \rightarrow \infty} v^k = v^*, & (b) \\ \lim_{k \rightarrow \infty} w^k = w^*. & (c) \end{cases} \quad (31)$$

Then we obtain that the sequence  $\{(w^k, v^k, u^k, \alpha^k, \beta^k)\}$  generated by the iteration scheme (15)(a)–(e) converges to the saddle point  $\{(w^*, v^*, u^*, \alpha^*, \beta^*)\}$ .  $\square$

## References

- [1] G. Aubert, P. Kornprobst, *Mathematical Problem in Image Processing: Partial Differential Equations and the Calculus of Variations*, Springer, New York, NY, 2008.
- [2] O. Scherzer, *Handbook of Mathematical Methods in Imaging*, Springer, New York, 2015.
- [3] C. Vogel, *Computational Methods for Inverse Problems*, SIAM, 2002.
- [4] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1–4) (1992) 259–268.
- [5] L. Ritschl, F. Bergner, C. Fleischmann, M. Kachelriebe, Improved total variation-based CT image reconstruction applied to clinical data, *Phys. Med. Biol.* 56 (2017) 1545.
- [6] K. Block, M. Uecker, J. Frahm, Undersampled radial MRI with multiple coils: iterative image reconstruction using a total variation constraint, *Magn. Reson. Med.* 57 (2007) 1086–1098.
- [7] M. Afonso, J. Sanches, Blind inpainting using  $\ell_0$  and total variation regularization, *IEEE Trans. Image Process.* 24 (7) (2015) 2239–2253.
- [8] M. Donoser, M. Urschler, M. Hirzer, Saliency driven total variation segmentation, in: *12th International Conference on Computer Vision*, 2009, p. 11367754.
- [9] M. Lysaker, A. Lundervold, X. Tai, Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time, *IEEE Trans. Image Process.* 12 (12) (2003) 1579–1590.
- [10] Y. You, M. Kaveh, Fourth-order partial differential equation for noise removal, *IEEE Trans. Image Process.* 9 (10) (2000) 1723–1730.
- [11] K. Bredies, K. Kunisch, T. Pock, Total generalized variatio, *SIAM J. Imaging Sci.* 3 (3) (2010) 492–526.
- [12] T. Rahman, X. Tai, S. Osher, A TV-stokes denoising algorithm, in: *Proceedings of Scale Space and Variational Methods in Computer Vision*, 2007, pp. 473–482.
- [13] K. Papafitsoros, C. Schonlieb, A combined first and second order variational approach for image reconstruction, *J. Math. Imaging Vision* 48 (2) (2014) 308–338.
- [14] F. Li, C. Shen, J. Fan, C. Shen, Image restoration combining a total variational filter and a fourth-order filter, *J. Vis. Commun. Image Represent.* 18 (4) (2007) 322–330.
- [15] A. Buades, B. Coll, J. Morel, A review of image denoising methods, with a new one, *Multiscale Model. Simul.* 4 (2) (2006) 490–530.
- [16] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, *Multiscale Model. Simul.* 7 (3) (2008) 1005–1028.
- [17] S. Lefkimmatis, M. Unser, Poisson image reconstruction with Hessian Schatten-norm regularization, *IEEE Trans. Image Process.* 22 (11) (2013) 4314–4327.
- [18] J. Bai, X. Feng, Fractional-order anisotropic diffusion for image denoising, *IEEE Trans. Image Process.* 16 (10) (2007) 2492–2502.
- [19] J. Zhang, Z. Wei, A class of fractional-order multi-scale variational models and alternating projection algorithm for image denoising, *Appl. Math. Model.* 35 (5) (2011) 2516–2528.
- [20] M. Grasmair, F. Lenzen, Anisotropic total variation filtering, *Appl. Math. Optim.* 62 (3) (2010) 323–339.
- [21] M. Kumar, M. Diwakar, A new exponentially directional weighted function based CT image denoising using total variation, *J. King Saud Univ. -Comput. Inf. Sci.* (2017) <http://dx.doi.org/10.1016/j.jksuci.2016.12.002>.



- [22] Y. Lou, T. Zeng, S. Osher, J. Xin, A weighted difference of anisotropic and isotropic total variation model for image processing, *SIAM J. Imag. Sci.* 8 (3) (2015) 1798–1823.
- [23] S. Setzer, G. Steidl, T. Teuber, Restoration of images with rotated shapes, *Numer. Algorithms* 48 (1–3) (2008) 49–66.
- [24] B. Berkels, M. Burger, M. Droske, O. Nemitz, M. Rumpf, Cartoon extraction based on anisotropic image classification vision, modeling, and visualization, in: *Vision, Modeling, and Visualization*, 2009.
- [25] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, *SIAM J. Imaging Sci.* 1 (3) (2008) 248–272.
- [26] N. Parikh, S. Boyd, Proximal algorithms, *Found. Trends Optim.* 1 (3) (2014) 123–231.
- [27] A. Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vision* 20 (1–2) (2004) 89–97.
- [28] J. Aujol, Some first-order algorithms for total variation based image restoration, *J. Math. Imaging Vision* 34 (3) (2009) 307–327.
- [29] M. Zhu, S. Wright, T. Chan, Duality-based algorithms for total-variation-regularized image restoration, *Comput. Optim. Appl.* 47 (3) (2010) 377–400.
- [30] S. Setzer, Operator splittings, Bregman methods and frame shrinkage in image processing, *Int. J. Comput. Vis.* 92 (3) (2011) 265–280.
- [31] A. Chambolle, T. Pock, A first-order primal–dual algorithm for convex problems with applications to imaging, *J. Math. Imaging Vision* 40 (1) (2011) 120–145.
- [32] P. Combettes, J. Pesquet, Proximal splitting methods in signal processing. fixed-point algorithms for inverse problems, in: *Science and Engineering*, Springer, New York, 2011, pp. 185–212.
- [33] E. Esser, X. Zhang, T. Chan, A general framework for a class of first order primal–dual algorithms for convex optimization in imaging science, *SIAM J. Imaging Sci.* 3 (4) (2010) 1015–1046.
- [34] L. Briceno-Arias, P. Combettes, J. Pesquet, N. Pustelnik, Proximal algorithms for multicomponent image recovery problems, *J. Math. Imaging Vision* 41 (1–2) (2011) 3–22.
- [35] S. Bubeck, Convex optimization: algorithms and complexity, *Found. Trends Optim.* 8 (3–4) (2015) 231–357.
- [36] E. Bollt, R. Chartrand, S. Esodoglu, P. Schultz, K. Vixie, Graduated adaptive image denoising: local compromise between total variation and isotropic diffusion, *Adv. Comput. Math.* 31 (1–7) (2009) 61–85.
- [37] F. Li, Z. Li, L. Pi, Variable exponent functionals in image restoration, *Appl. Math. Comput.* 216 (3) (2010) 870–882.
- [38] Y. Chen, S. Levine, M. Rao, Variable exponent, linear growth functionals in image restoration, *SIAM J. Appl. Math.* 66 (4) (2006) 1383–1406.
- [39] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (1) (2011) 1–122.
- [40] G. Golub, C. Loan, *Matrix Computation*, Johns Hopkins Press, 2012.
- [41] R. Glowinski, S. Osher, T. Yin, *Splitting Methods in Communication, Imaging, Science, and Engineering*, Springer, 2016.
- [42] J. Yang, X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, *Math. Comput.* 82 (281) (2013) 301–329.
- [43] J. Cai, S. Osher, Z. Shen, Convergence of the linearized Bregman iteration for  $\ell_1$ -norm minimization, *Math. Comp.* 78 (268) (2009) 2127–2136.