Journal of Mathematical Imaging and Vision (2023) 65:414–436
https://doi.org/10.1007/s10851-022-01122-x

Anisotropic Variational Models for Image Denoising Based on Directional Hessian

Guangyu Yang[1] · Weibo Wei[1] · Zhenkuan Pan[1]

Received: 20 July 2021 / Accepted: 18 September 2022 / Published online: 27 October 2022

Abstract

Anisotropic and high-order diffusion variational models have excellent performances in image coherence and smoothness preserving, respectively. In order to preserve these merits simultaneously in one variational model for image restoration, we propose three second-order anisotropic variational models making use of directional Hessian. The first one is the double-orientational bounded Hessian (DOBH) model; it is an extension to the isotropic bounded Hessian (BH) model. The second is the double-orientational total generalized variation (DOTGV), which is an extension to the total generalized variation (TGV) model. The third is the double-orientational total variation and bounded Hessian (DOTBH) model, which is a hybrid one combining the first-order and second-order directional regularizers. The second-order directional derivatives are designed by Hession and directional vectors which are derived from classic structure tensors. In order to cope with complex calculations of these models, alternating direction method of multipliers (ADMM) algorithms are designed, respectively. Thus, the proposed models can be decomposed into a set of simple sub-problems of optimization, which can be solved by fast FFT method or soft thresholding formulas. In order to improve computational efficiency, fast ADMM algorithms with restart strategy are designed and implemented finally. Experimental results demonstrate better performances compared with previous classical models, especially in large-scale texture restoration.

Keywords Image denoising · Directional Hessian · Structure tensor · Anisotropic diffusion · ADMM

1 Introduction

Image denoising is a long-term topic in image processing due to its fundamental significance in computer vision. Variational method has become one of the influential approaches in these areas [1,2] in recent 30 years. The regularizers in variational models of image denoising are core parts since they determine different feature preserving performances. Investigations on regularizers of high-order isotropic diffusion for image smoothness preserving and first-order anisotropic diffusion for image coherence preserving have made many achievements. In this paper, we will focus on variational models with second-order anisotropic diffusion regularizers for more feature preserving.

Let Ω be an image domain, $f : \Omega \rightarrow \mathbb{R}$ and $u : \Omega \rightarrow \mathbb{R}$ be noised and restored image, respectively, a typical variational model for image denoising is usually stated as

$$\min_{u} \{\lambda F(u, f) + R(u)\}, \tag{1.1}$$

where $\lambda > 0$ is a trade-off parameter. The first term in (1.1) is a fidelity term which ensures u to be close to f. The second term in (1.1) is a regularizer which determines what kinds of image features can be preserved. The earliest regularizer for linear isotropic diffusion is Tikhonov regularized term [3] which leads to smeared edges. The nonlinear diffusion Total variation (TV) regularizer proposed in [4] can overcome this problem, but leads to staircase effects. So various regularizers with high-order derivatives and anisotropic strategies for smoothness and edge preserving have attracted a lot of investigations [5–8].

Anisotropic diffusion is an extension to isotropic diffusion in scale space [9,10]; it depends on diffusion directions, which can be estimated via structure tensor naturally [11–13]. Some researchers proposed other direction estimation framework; for example, [14] designed a parameterized single orientation estimation approach for directional total

✉ Zhenkuan Pan
zkpan@126.com

[1] College of Computer Science & Technology, Qingdao University, Qingdao 266071, China

variation (DTV) model and [15] proposed to estimate the main direction according to edge gradient called edge adaptive directional total variation (EADTV) model. In addition, TV regularization has been extended by the orientational information to capture the local features in [16]. Furthermore, these directional regularizers were generalized from first-order to higher-order form, such as directional total generalized variation (DTGV) model [17] for texture image diffusion single direction globally. [18] proposed a new anisotropic higher-order total directional variation regularize and extended it to third-order derivative to solve inverse imaging problems.

There are also some researches on smoothing images in multi-directional approach. Some authors proposed to estimate directions by decomposing them into horizontal and vertical as a two-dimensional problem. This concept was first applied to the denoising of isotropic TV model to improve efficiency of the algorithm [19]. Then, it was extended to higher-order in [20], the variation regularizer with double-orientational estimation based on BH model was established to eliminate the staircase effect. By adding edge detection functions, a new nonlinear anisotropic high-order model was developed in [21], which has a better edge preserving ability.

In recent years, some other methods with anisotropic properties were also emerged. Anisotropic total variation (ATV) model based on local weighting matrix and gradient operator was constructed in [22]. There were also non-local techniques and other latest methods which utilize edge-preserving in image restoration [23–35]. The representative one in this category is non-local total variation (NLTV) regularizer [23]. In [24], the discrete domain extension of NLTV for vector-valued images was studied. [25] proposed a novel approach to non-local adaptive nonparametric filtering. With introducing of new various theories, anisotropic diffusional equation has been widely used in the field of image processing [26–29].

In this paper, inspired by structure tensor, three second-order anisotropic variational regularizers are developed based on directional Hessian. The orthogonal system of the diffusion tensor including directional Hessian is constructed which tends to describing local neighborhoods in double orientations. The required directional information can be calculated efficiently by eigenvectors without explicit orientational analysis. Meanwhile, corresponding eigenvalues as the diffusion intensity are recommended to divide image features. It is worth noting that our methods which make orientations seem more useful for describing local structures. Unlike estimating directions of the image from both horizontal and vertical directions, a set of orthogonal directions according to gradients changes are obtained by directional Hessian. Compared with parameterized single-orientation estimational approach, diffusion intensity can be designed flexibly and visually. Direction estimational methods which can improve robustness are adept in handling two directions of mutual occlusion.

By introducing proper auxiliary variables, original models are decomposed into several simple minimization subproblems and can be solved with alternating direction method of multipliers (ADMM). In addition, effective acceleration methods are applied to optimize iterative process until the convergence is guaranteed.

Both the theoretical and numerical details are described within variation formulations. In the contrast experiments, except for classical second-order variational models such as BH and TGV models, anisotropic first-order and second-order models such as TDV, non-local TV and BM3D are equally competitive to the proposed models. Experimental results showed that proposed models have better performance in preserving image features. Especially they are more suitable for processing large-scale texture image restoration.

2 Background and Foundation

2.1 Related Models

Since the functional regularizers are particularly concerned, related anisotropic works are reviewed in this section.

2.1.1 Structure Tensor

Weickert's structure tensor allows both orientation estimation and image structure analysis. For image function $u : \Omega \to \mathbb{R}$, it is defined as

$$J_\rho(\nabla u_\sigma) = K_\rho * J_\sigma(\nabla u_\sigma) = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}, \tag{2.1.1}$$

with

$$J_\sigma(\nabla u_\sigma) := \nabla u_\sigma \otimes \nabla u_\sigma, \tag{2.1.2}$$

where ∇u_σ is vector-valued structure descriptor, J_σ is resulted from the tensor product $\otimes$, and the symmetric matrix $J_\rho(\nabla u_\sigma) = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix}$ is positive semidefinite. σ, ρ are non-negative parameters, $u_\sigma = K_\sigma * u$. K_ρ, K_σ are Gaussian kernels with standard deviations as σ, ρ, respectively. Usually, σ and ρ fulfill $\sigma \ll \rho$, where σ is associated to noise scale, and ρ integrates orientational information. For $\nabla u_\sigma \neq 0$, structure tensor $J_\rho(u)$ has two orthogonal eigenvectors $\mathbf{T}$ and $\mathbf{N}$ which approximate isophote direction and gradient direction. Eigenvectors reflect the minimum and maximum variational directions of the local image, respectively. The corresponding nonzero real eigenvalues are λ_1 and λ_2.

The structure tensor in (2.1.1) can be rewritten as

$$J_\rho(\nabla u_\sigma) = \lambda_1(\mathbf{T}\otimes\mathbf{T}) + \lambda_2(\mathbf{N}\otimes\mathbf{N}), \tag{2.1.3}$$

with

$$\mathbf{U} = \begin{bmatrix} 2J_{12} \\ J_{22} - J_{11} + \sqrt{(J_{11}-J_{22})^2 + 4(J_{12})^2} \end{bmatrix}, \mathbf{N} = \frac{\mathbf{U}}{|\mathbf{U}|}, \tag{2.1.4}$$

$$\mathbf{V} = \begin{bmatrix} J_{11} - J_{22} - \sqrt{(J_{11}-J_{22})^2 + 4(J_{12})^2} \\ 2J_{12} \end{bmatrix}, \mathbf{T} = \frac{\mathbf{V}}{|\mathbf{V}|}. \tag{2.1.5}$$

where **N** is normal direction and **T** is tangential direction. Corresponding eigenvalues λ_1 and λ_2 are

$$\begin{cases} \lambda_1 = \frac{1}{2}\left(J_{11} + J_{22} + \sqrt{(J_{11}-J_{22})^2 + 4(J_{12})^2}\right) \\ \lambda_2 = \frac{1}{2}\left(J_{11} + J_{22} \text{ - } \sqrt{(J_{11}-J_{22})^2 + 4(J_{12})^2}\right) \end{cases}. \tag{2.1.6}$$

Eigenvalues of $J_\rho(u)$ provide useful information about consistency of the structure, and the coherence is expressed as

$$(\lambda_1 - \lambda_2)^2 = (J_{11} - J_{22})^2 + 4J_{12}{}^2. \tag{2.1.7}$$

For any $x \in \Omega$, structure image information is classified according to the size of λ_1 and λ_2.

If $\lambda_1 \approx 0, \lambda_2 \approx 0$, then x is likely to belong to a homogeneous region;

If $\lambda_1 \gg \lambda_2 \approx 0$, then x is likely to lie on an edge;

If $\lambda_1 \approx \lambda_2 > 0$, then x is likely to be a corner point.

Since the choice of diffusion tensor should reflect local image structure that it reveals the same set of eigenvectors **T** and **N** as $J_\rho(u)$. The desired goal of the filter affects the choice of corresponding eigenvalues set as g_T and g_N. It is required to adaptively use different diffusion coefficients in various directions to preserve image features. The choice of eigenvalues reflects the degree of diffusion in distinct intensity.

Weickert studied two representatives of anisotropic diffusion processes according to diverse selection of eigenvalues.

The first one is called edge-enhancing anisotropic diffusion, which has the advantage of dealing with noisy edges. The corresponding designs for g_T and g_N are

$$\begin{cases} g_T = 1, \\ g_N = \begin{cases} 1 & (p \le 0) \\ 1 - \exp\left(\frac{-C_m}{(p/\lambda_0)^m}\right) & (p > 0) \end{cases}, \end{cases} \tag{2.1.8}$$

where m = 4, C$_4$= 3.31488, $\lambda_0 > 0$.

Constant C_m is calculated in such a way that flux $\Phi(p) = pg(p)$ is increasing for p $\in [0, \lambda_0]$ and decreasing for p $\in [\lambda_0, \infty]$.

The second one is coherence-enhancing anisotropic diffusion, which can be well adapted to processing one-dimensional features. The corresponding designs for g_T and g_N are

$$\begin{cases} g_T = \begin{cases} l & if \quad \lambda_1 = \lambda_2 \\ l + (1-l)\exp\left(\frac{-C}{(\lambda_1-\lambda_2)^{2m}}\right) & else \end{cases} \\ g_N = l \end{cases} \tag{2.1.9}$$

where $l = 0.001, C = 1, m = 1$.

2.1.2 Directional Total Variation

Further, in [14], the discrete DTV introduced for image denoising in a single dominant direction was expressed as

$$\min\left\{E(u) = \lambda \int_\Omega \left|\nabla_{\theta,a} u\right| dx + \frac{1}{2}\int_\Omega (u-f)^2 dx\right\}. \tag{2.1.10}$$

The circular unit ball generated by L^2-norm is transformed into an ellipse. Major semi-axis is $a > 1$ rotated by θ which penalizes variations for a. R_θ and Λ_a denote rotation and scaling matrices, respectively, as

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \Lambda_a = \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.1.11}$$

Intrinsic gradient and divergence functions are

$$\begin{cases} \nabla_{\theta,a} u = R_\theta \Lambda_a \nabla u \\ \nabla_{\theta,a} \mathbf{v} = \nabla \cdot (R_{-\theta}\Lambda_a \mathbf{v}) \end{cases}. \tag{2.1.12}$$

[15] proposed to spatially vary the direction θ in multiple directions as

$$(\cos\theta, \sin\theta) = \frac{(\nabla u_\sigma)^T}{|\nabla u_\sigma|}. \tag{2.1.13}$$

2.1.3 Directional Total Generalized Variation

The main advantage of TGV model is to reconstruct piecewise polynomial functions. Compared with traditional TV regularization which can only reconstruct piecewise constant function, this method has superior restorational performance.

The regularization term given by DTGV [17] is

$$\min\left\{\frac{1}{2}(u-f)^2 + \lambda_1 \int_\Omega \left|\nabla_{\theta,\alpha} u \text{ - } p\right| dx + \lambda_2 \int_\Omega |\varepsilon(p)| dx\right\}. \tag{2.1.14}$$

Directional symmetrized derivative $\varepsilon(\cdot)$ is the adjoint operator. [17] focused on DTGV analysis under continuous settings and only single directional estimation was designed for entire image.

2.2 Anisotropic Diffusion

In order to improve the quality of image denoising, orientational information is introduced to the image denoising process. In various image filtering and analysis tasks, orientational estimation is essential.

Local orientational estimation is an important issue in image processing and computer vision. A group of methods obtained the local geometry by calculating structure tensor, eigenvalues and orthogonal eigenvectors of structure tensor. The diffusion tensor are used to defined to control the direction of flux in partial differential equation. [36] divided these methods into divergence-based, trace-based, and curvature-based methods, all of them have been deeply studied in [37–39]. The curvature-based method related to line integral convolution [9] is more suitable for recovery of sharp nodes than the other two methods. Many authors dealt with single-orientational estimation through the methods mentioned above for subsequent denoising [17,18].

The standard structure tensor method can be extended with a combination of higher-order derivatives to obtain its advantages. Odd-order filters and even-order filters can be used for edge detection and straight line detection, respectively. Mixed-order filters can be used for phase invariance. On the basis of studying local orientation, a unified multiple orientations mathematical model was established. Novel estimational schemes for an arbitrary number of superimposed orientations in bivariate images as well as double orientations in signals of arbitrary signal dimensionality were derived in [40–44]. In general, double-orientational approach for bivariate images is more valid, robust, and accurate.

A continuous diffusion equation is denoted as follows:

$$\frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{D}\nabla u). \tag{2.2.1}$$

where $\mathbf{D}$ is diffusion tensor. It can be rewritten as follows:

$$\frac{\partial u}{\partial t} = c_T I_{TT} + c_N I_{NN}, \tag{2.2.2}$$

where I_{TT}, I_{NN} are second-order derivatives of corresponding two directions and $\mathbf{T}, \mathbf{N}$ are two orthogonal vectors.

We define the $DH = I_{TT} + I_{NN}$ as directional Hessian. I_{TT}, I_{NN} are given as follows:

$$\begin{cases} I_{\mathbf{TT}} = \mathbf{T}^T\mathbf{HT} \\ I_{\mathbf{NN}} = \mathbf{N}^T\mathbf{HN} \end{cases}. \tag{2.2.3}$$

As for double-orientation tensors, the derivatives along the direction of $\mathbf{T}$ and $\mathbf{N}$ can be expressed as

$$\begin{cases} U_N = \mathbf{N} \cdot \nabla u \\ U_T = \mathbf{T} \cdot \nabla u \end{cases}. \tag{2.2.4}$$

Energy functional which realizes linear anisotropic diffusion is defined as

$$\min \left\{ \begin{aligned} E(u) &= \frac{1}{2}\int_\Omega g_T |u_T|^2 dx + \frac{1}{2}\int_\Omega g_N |u_N|^2 dx \\ &= \frac{1}{2}\int_\Omega g_T \nabla^T u \mathbf{J}_T \nabla u dx + \frac{1}{2}\int_\Omega g_N \nabla^T u \mathbf{J}_N \nabla u dx \end{aligned} \right\} \tag{2.2.5}$$

where $\mathbf{J}_T = \mathbf{TT}^T, \mathbf{J}_N = \mathbf{NN}^T$.

Equation (2.2.5) can be rewritten as

$$\min \left\{ E(u) = \frac{1}{2}\int_\Omega \nabla^T u \mathbf{J} \nabla u dx \right\}, \tag{2.2.6}$$

where $\mathbf{J} = g_T\mathbf{J}_T + g_N\mathbf{J}_N, \mathbf{J}_T = \mathbf{TT}^T, \mathbf{J}_N = \mathbf{NN}^T$.

The energy functional which realizes nonlinear anisotropic diffusion is defined as

$$\min \left\{ E(u) = \frac{1}{2}\int_\Omega g_T |u_T| dx + \frac{1}{2}\int_\Omega g_N |u_N| dx \right\}. \tag{2.2.7}$$

Minimizing the functional (2.2.5) by steepest descent is as follows:

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{J}\nabla u) & x \in \Omega, t > 0 \\ u(x,0) = f(x) & x \in \Omega \cup \partial\Omega, t = 0 \\ (\mathbf{J}\nabla u) \cdot \mathbf{n} = 0 & x \in \Omega \cup \partial\Omega, t = 0 \end{cases}. \tag{2.2.8}$$

Minimizing the functional (2.2.7) by steepest descent is as follows:

$$\begin{cases} \frac{\partial u}{\partial t} = \nabla \cdot (\mathbf{A}(\mathbf{u})\nabla u) & x \in \Omega, t > 0 \\ u(x,0) = f(x) & x \in \Omega \cup \partial\Omega, t = 0 \\ \nabla u \cdot \mathbf{n} = 0 & x \in \partial\Omega, t \geq 0 \end{cases}, \tag{2.2.9}$$

where $\mathbf{A}(\mathbf{u}) = g_T \frac{\mathbf{TT}^T}{|U_T|} + g_N \frac{\mathbf{NN}^T}{|U_N|}$.

If $\mathbf{R} = [\mathbf{N}\ \mathbf{T}]^T$ represents morphological transform matrix and $g = diag(g_N\ g_T)$ represents diffusion function matrix. (2.2.8) and (2.2.9) can be rewritten as

$$\begin{cases} \min \left\{ E(u) = \int_\Omega |g\mathbf{R}\nabla u|^2 \right\} \\ \min \left\{ E(u) = \int_\Omega |g\mathbf{R}\nabla u| \right\} \end{cases}. \tag{2.2.10}$$

The corresponding gradient descent equations are

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot \left(\mathbf{R}^T g^T g \mathbf{R} \nabla u \right) \\ \dfrac{\partial u}{\partial t} = \nabla \cdot \left(\dfrac{\mathbf{R}^T g^T g \mathbf{R} \nabla u}{|g \mathbf{R} \nabla u|} \right) \end{cases}. \tag{2.2.11}$$

The relation between eigenvectors in double-orientational estimation and single directional estimation [15] is

$$\begin{cases} \mathbf{T} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \\ \mathbf{N} = \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \end{cases}, \tag{2.2.12}$$

where $\theta = \frac{1}{2} \arctan \frac{2J_{12}}{J_{11} - J_{22}}$.

The choice of corresponding eigenvalues sets as g_T and g_N which depend on the desired goal of filter is used to represent image structure.

g_T and g_N can be estimated directly from the degraded images, such as homogeneous region, edge and corner point. It is worth noting that many corners are contained within edges. Several design methods commonly used in the literatures are introduced here.

In [18], eigenvalues of diffusion tensor are defined as

$$\mathbf{b} = (1, b_2(x)), \tag{2.2.13}$$

with

$$b_2(x) = \frac{\lambda_1(x) - \lambda_2(x)}{\lambda_1(x) + \lambda_2(x) + \varepsilon}, \tag{2.2.14}$$

where $\varepsilon > 0$, λ_1 and λ_2 are eigenvalues of structure tensor.

There are other approaches like weight function to design g_T and g_N. One of them can be defined as follows:

$$\begin{cases} g_T = o_1(x, y)c_1 + o_2(x, y)\gamma_1 \\ g_N = o_1(x, y)c_2 + o_2(x, y)\gamma_2 \end{cases}, \tag{2.2.15}$$

where c_1 and c_2 are two directional characteristic weights to smooth region, γ_1 and γ_2 are two directional characteristic weights to edge region. $o_1(x, y)$ is decreasing function of $|\nabla u|$ and $o_2(x, y)$ is increasing function of $|\nabla u|$.

For most directional images, the above selections for g_T and g_N can achieve the similar effects. The classical design in (2.1.7) is used in the following part due to its simple and efficient characters.

2.3 Discretization

Before proceeding, some notations are introduced briefly. Let $\Omega \to \mathbb{R}^{M \times N} \frac{n!}{r!(n-r)!}$ represents two-dimensional gray image space $\mathbb{R}$ with size $M \times N$. First-order forward differences are, respectively,

$$\partial_x^+ u_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } 1 \le i \le M, 1 \le j \le N \\ u_{1,j} - u_{i,j} & \text{if } i = M, 1 \le j \le N \end{cases}.$$
$$\partial_y^+ u_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } 1 \le i \le M, 1 \le j \le N \\ u_{i,1} - u_{i,j} & \text{if } 1 \le i \le M, j = N \end{cases}, \tag{2.3.1}$$

First-order backward differences are, respectively, as follows:

$$\partial_x^- u_{i,j} = \begin{cases} u_{i,j} - u_{i-1,j} & \text{if } 1 < i \le M, 1 \le j \le N \\ u_{i,j} - u_{M,j} & \text{if } i = 1, 1 \le j \le N \end{cases}.$$
$$\partial_y^- u_{i,j} = \begin{cases} u_{i,j} - u_{i,j-1} & \text{if } 1 \le i \le M, 1 \le j \le N \\ u_{i,j} - u_{i,N} & \text{if } 1 \le i \le M, j = 1 \end{cases}, \tag{2.3.2}$$

The boundary conditions are essential in the process of solving Euler–Lagrange equations. The periodic boundary conditions are used in the FFT algorithm for solving process of subsequent models. The gradient, Laplacian, and divergence are discretized as follows:

$$\nabla u = \begin{bmatrix} \partial_x^+ u \\ \partial_y^+ u \end{bmatrix},$$
$$\nabla \cdot \nabla z = \left(\partial_x^- \partial_x^+ + \partial_y^- \partial_y^+ \right) z, \tag{2.3.3}$$
$$\nabla(\nabla \cdot \mathbf{n}) = \begin{bmatrix} \partial_x^- \partial_x^+ n_1 + \partial_x^- \partial_y^+ n_2 \\ \partial_y^- \partial_x^+ n_1 + \partial_y^- \partial_y^+ n_2 \end{bmatrix}.$$

For discrete frequencies, $i \in [1, M]$ and $j \in [1, N]$ are the discrete time domain indicators, $r \in [0, M-1]$ and $s \in [0, N-1]$ are the discrete frequency domain indexes. For $f_{i,j} \in \mathbb{R}^{M \times N}$, $\mathcal{F}$ is obtained, shifting operators $S_1{}^{\pm}$, $S_2{}^{\pm}$ are introduced as follows:

$$\mathcal{F}(S_1{}^{\pm} f_{i,j}) = e^{\pm\sqrt{-1}z_i} \mathcal{F}(f_{i,j}),\ z_i = \frac{2\pi}{N} s$$
$$\mathcal{F}(S_2{}^{\pm} f_{i,j}) = e^{\pm\sqrt{-1}z_j} \mathcal{F}(f_{i,j}),\ z_j = \frac{2\pi}{N} r, \tag{2.3.4}$$

where $e^{\pm\sqrt{-1}z_i} = \cos z_i \pm \sqrt{-1} \sin z_i$ and $e^{\pm\sqrt{-1}z_j} = \cos z_j \pm \sqrt{-1} \sin z_j$.

FFT is used to solve high-order Euler equations by upper transformation to get algebraic equations. Then, optimized variables are obtained by inverse DFT. $\Re(\cdot)$ is used to represent the real part of a complex number in variable result.

3 The Proposed Models

In this section, it is the first attempt to established novel anisotropic variation models based on directional Hessian for image denoising. Then, fast ADMM with restart is employed to design an iterative scheme to solve the energy equation. In addition, gradient descent method, soft threshold formula, FFT and other composite numerical approximation methods are also introduced in this section.

3.1 DOTV Model

Enlightened by DTV model, first-order anisotropic variation which named double-orientational TV (DOTV) model is proposed.

$$\begin{aligned}&\min\left\{\begin{array}{l} E_1(u)=\gamma_T\displaystyle\int_\Omega g_T\left|u_T\right|dx+\gamma_N\int_\Omega g_N\left|u_N\right|dx \\ +\dfrac{1}{2}\displaystyle\int_\Omega (f-u)^2dx,\end{array}\right\}\\ &s.t.\ u_T=\mathbf{T}\cdot\nabla u,\ u_N=\mathbf{N}\cdot\nabla u.\end{aligned}\tag{3.1.1}$$

where u_T and u_N are scalar products. $\mathbf{T}$ and $\mathbf{N}$ are orthogonal eigenvectors in the structure tensor corresponding to the maximum and minimum local directional variation. g_T and g_N are their associated eigenvalues which need to be designed. γ_T, γ_N are penalty parameters imposed on the regularize terms, respectively. By introducing three auxiliary variables $\mathbf{w}=\nabla u$, $w_T=\mathbf{T}\cdot\mathbf{w}$ and $w_N=\mathbf{N}\cdot\mathbf{w}$, (3.1.1) can be transformed into following the augmented Lagrangian formulation.

$$\begin{aligned}&(u,\mathbf{w},w_T,w_N)\\ &=\mathop{\arg\min}_{u,\mathbf{w},w_T,w_N} E_1\\ &=\mathop{\arg\min}_{u,\mathbf{w},w_T,w_N}\left\{\begin{array}{l}\gamma_T\displaystyle\int_\Omega g_T\left|u_T\right|dx+\gamma_N\int_\Omega g_N\left|u_N\right|dx\\ +\dfrac{1}{2}\displaystyle\int_\Omega (f-u)^2dx+\int_\Omega \boldsymbol{\lambda}\cdot(\mathbf{w}-\nabla u)\,dx\\ +\dfrac{\mu}{2}\displaystyle\int_\Omega |\mathbf{w}-\nabla u|^2dx\\ +\displaystyle\int_\Omega \lambda_T\cdot\left(w_T-\mathbf{T}\cdot\mathbf{w}\right)dx\\ +\dfrac{\mu_T}{2}\displaystyle\int_\Omega \left(w_T-\mathbf{T}\cdot\mathbf{w}\right)^2dx\\ +\displaystyle\int_\Omega \lambda_N\cdot\left(w_N-\mathbf{N}\cdot\mathbf{w}\right)dx\\ +\dfrac{\mu_N}{2}\displaystyle\int_\Omega \left(w_N-\mathbf{N}\cdot\mathbf{w}\right)^2dx\end{array}\right\},\end{aligned}\tag{3.1.2}$$

where μ, μ_T, μ_N are positive penalty parameters, $\boldsymbol{\lambda}$, λ_T, λ_N are Lagrange multipliers. The solution of (3.1.2) can be decomposed into four sub-problems:

$$\varepsilon_1(u)=\min_u\left\{\begin{array}{l}E(u)=\dfrac{1}{2}\displaystyle\int_\Omega (f-u)^2dx\\ \qquad+\dfrac{\mu}{2}\displaystyle\int_\Omega\left|\mathbf{w}-\nabla u+\dfrac{\boldsymbol{\lambda}}{\mu}\right|^2dx\end{array}\right\},\tag{3.1.3}$$

$$\begin{aligned}&\varepsilon_2(\mathbf{w})\\ &=\min_{\mathbf{w}}\left\{\begin{array}{l}E(\mathbf{w})=\dfrac{\mu}{2}\displaystyle\int_\Omega\left|\mathbf{w}-\nabla u+\dfrac{\boldsymbol{\lambda}}{\mu}\right|^2\\ \quad+\dfrac{\mu_T}{2}\displaystyle\int_\Omega\left(w_T-\mathbf{T}\cdot\mathbf{w}+\dfrac{\lambda_T}{\mu_T}\right)^2dx\\ \quad+\dfrac{\mu_N}{2}\displaystyle\int_\Omega\left(w_N-\mathbf{N}\cdot\mathbf{w}+\dfrac{\lambda_N}{\mu_N}\right)^2dx\end{array}\right\},\end{aligned}\tag{3.1.4}$$

$$\begin{aligned}&\varepsilon_3(w_T)\\ &=\min_{w_T}\left\{\begin{array}{l}E(w_T)=\gamma_T\displaystyle\int_\Omega g_T\left|w_T\right|dx\\ \quad+\dfrac{\mu_T}{2}\displaystyle\int_\Omega\left(w_T-\mathbf{T}\cdot\mathbf{w}+\dfrac{\lambda_T}{\mu_T}\right)^2dx\end{array}\right\},\end{aligned}\tag{3.1.5}$$

$$\begin{aligned}&\varepsilon_4(w_N)\\ &=\min_{w_N}\left\{\begin{array}{l}E(w_N)=\gamma_N\displaystyle\int_\Omega g_N\left|w_N\right|dx\\ \quad+\dfrac{\mu_N}{2}\displaystyle\int_\Omega\left(w_N-\mathbf{N}\cdot\mathbf{w}+\dfrac{\lambda_N}{\mu_N}\right)^2dx\end{array}\right\}.\end{aligned}\tag{3.1.6}$$

The corresponding Euler equations of (3.1.3)- (3.1.6) can be solved by numerical approximation. The iterative solution process is decomposed into following five sub-processes.

Sub 1: Fixed $\mathbf{w}^k, w_T^k, w_N^k$ to calculate u^{k+1}. Euler–Lagrange equation of (3.1.3) is

$$u^{k+1}-f-\mu\Delta u^{k+1}+\mu\left(\nabla\cdot\left(\mathbf{w}^k+\frac{\boldsymbol{\lambda}^k}{\mu}\right)\right)=0.\tag{3.1.7}$$

(3.1.7) can be efficiently solved by FFT

$$u_{i,j}^{k+1}=\Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{1i,j})}{\varepsilon_1}\right)\right),\tag{3.1.8}$$

with $G_{1i,j}=f_{i,j}-\nabla\cdot\boldsymbol{\lambda}_{i,j}^k-\mu\nabla\cdot\mathbf{w}_{i,j}^k$ and $\varepsilon_1=1-2\mu(\cos\frac{2\pi s}{N}+\cos\frac{2\pi r}{M}-2)$.

Sub 2: Fixed $u^{k+1}, {w_T}^k, {w_N}^k$ to calculate $\mathbf{w}^{k+1}$. Euler–Lagrange equation of (3.1.4) is

$$\begin{aligned}&\underbrace{\left(\mu I+\mu_T\mathbf{T}\mathbf{T}^T+\mu_N\mathbf{N}\mathbf{N}^T\right)}_{\mathrm{K}}\mathbf{w}^{k+1}\\ &=\underbrace{\mu\left(\nabla u^{k+1}-\frac{\boldsymbol{\lambda}^k}{\mu}\right)+\mu_T\mathbf{T}\left(w_T^k+\frac{\lambda_T^k}{\mu_T}\right)+\mu_N\mathbf{N}\left(w_N^k+\frac{\lambda_N^k}{\mu_N}\right)}_{\mathrm{Z}}.\end{aligned}\tag{3.1.9}$$

The solution of $\mathbf{w}^{k+1}$ can be achieved as

$$\mathbf{w}^{k+1}=\mathrm{K}^{-1}\mathrm{Z}.\tag{3.1.10}$$

Sub 3: Fixed u^{k+1}, $\mathbf{w}^{k+1}$, w_N^k to calculate w_T^{k+1}. (3.1.5) can be solved via generalized soft thresholding equation

$$
\begin{aligned}
w_T^{k+1} &= \max\left(\left|\mathbf{T}\cdot\mathbf{w}^{k+1}-\frac{\lambda_T^k}{\mu_T}\right|-\frac{\gamma_T}{\mu_T}g_T,0\right)\\
&\quad\times\frac{\mathbf{T}\cdot\mathbf{w}^{k+1}-\frac{\lambda_T^k}{\mu_T}}{\left|\mathbf{T}\cdot\mathbf{w}^{k+1}-\frac{\lambda_T^k}{\mu_T}\right|},0\frac{0}{|0|}.
\end{aligned}
\tag{3.1.11}
$$

Sub 4: Fixed u^{k+1}, $\mathbf{w}^{k+1}$, w_T^{k+1} to calculate w_N^{k+1}. (3.1.6) can be solved via generalized soft thresholding equation

$$
\begin{aligned}
w_N^{k+1} &= \max\left(\left|\mathbf{N}\cdot\mathbf{w}^{k+1}-\frac{\lambda_N^k}{\mu_N}\right|-\frac{\gamma_N}{\mu_N}g_N,0\right)\\
&\quad\times\frac{\mathbf{N}\cdot\mathbf{w}^{k+1}-\frac{\lambda_N^k}{\mu_N}}{\left|\mathbf{N}\cdot\mathbf{w}^{k+1}-\frac{\lambda_N^k}{\mu_N}\right|},0\frac{0}{|0|}.
\end{aligned}
\tag{3.1.12}
$$

Sub 5: Update Lagrange multipliers $\boldsymbol{\lambda}^{k+1}$, λ_T^{k+1}, λ_N^{k+1} in (3.1.2).

$$
\begin{cases}
\boldsymbol{\lambda}^{k+1}=\boldsymbol{\lambda}^k+\mu\left(\mathbf{w}^{k+1}-\nabla u^{k+1}\right)\\
\lambda_T^{k+1}=\lambda_T^k+\mu_T\left(w_T^{k+1}-\mathbf{T}\cdot\mathbf{w}^{k+1}\right)\\
\lambda_N^{k+1}=\lambda_N^k+\mu_N\left(w_N^{k+1}-\mathbf{N}\cdot\mathbf{w}^{k+1}\right)
\end{cases}.
\tag{3.1.13}
$$

3.2 DOBH Model

Then, high-order anisotropic variation based on the bounded Hessian named double-orientational bounded Hessian (DOBH) model is proposed.

$$
\begin{aligned}
&\min\left\{\begin{aligned}&E_2(u)=\gamma_t\int_\Omega g_T\left|u_{TT}\right|dx+\gamma_n\int_\Omega g_N\left|u_{NN}\right|dx\\&+\frac{\gamma}{2}\int_\Omega(u-f)^2dx,\end{aligned}\right\}\\
&s.t.\ v_T=u_{TT}=\mathbf{T}\cdot(\mathbf{HT}),v_N=u_{NN}=\mathbf{N}\cdot(\mathbf{HN}),\mathbf{H}=\nabla^2u.
\end{aligned}
\tag{3.2.1}
$$

where v_T and v_N are scalar products. $\mathbf{T}$, $\mathbf{N}$ are eigenvectors and g_T, g_N are eigenvalues. $\mathbf{H}$ represents a matrix-valued function.

By introducing auxiliary variables $\mathbf{H}=\nabla^2u$, $v_T=\mathbf{T}\cdot(\mathbf{HT})$ and $v_N=\mathbf{N}\cdot(\mathbf{HN})$, the constrained optimizational problem (3.2.1) can be efficiently solved with following the augmented Lagrangian formulation.

$$
\begin{aligned}
\left(u,\mathbf{H},v_T,v_N\right)&=\underset{u,\mathbf{H},v_T,v_N}{\arg\min}\ E_2\\
&=\underset{u,\mathbf{H},v_T,v_N}{\arg\min}\left\{\begin{aligned}&\gamma_t\int_\Omega g_T\left|v_T\right|dx+\gamma_n\int_\Omega g_N\left|v_N\right|dx\\&+\frac{\gamma}{2}\int_\Omega(u-f)^2dx+\int_\Omega\breve{}_{\mathbf{H}}\left(\mathbf{H}-\nabla^2u\right)dx\\&+\frac{\mu_H}{2}\int_\Omega\left(\mathbf{H}-\nabla^2u\right)^2dx\\&+\int_\Omega\lambda_t\left(v_T-\mathbf{T}\cdot(\mathbf{HT})\right)dx\\&+\frac{\mu_t}{2}\int_\Omega\left(v_T-\mathbf{T}\cdot(\mathbf{HT})\right)^2dx\\&+\int_\Omega\lambda_n\left(v_N-\mathbf{N}\cdot(\mathbf{HN})\right)dx\\&+\frac{\mu_n}{2}\int_\Omega\left(v_N-\mathbf{N}\cdot(\mathbf{HN})\right)^2dx\end{aligned}\right\},
\end{aligned}
\tag{3.2.2}
$$

where μ_H,μ_t,μ_n are positive penalty parameters, $\boldsymbol{\lambda}_H,\lambda_t,\lambda_n$ are Lagrange multipliers. (3.2.2) is changed into following sub-problems.

$$
\varepsilon_1(u)=\min_u\left\{\begin{aligned}&E(u)=\frac{\gamma}{2}\int_\Omega(u-f)^2dx\\&+\frac{\mu_H}{2}\int_\Omega\left(\mathbf{H}-\nabla^2u+\frac{\boldsymbol{\lambda}_H}{\mu_H}\right)^2dx\end{aligned}\right\},
\tag{3.2.3}
$$

$$
\begin{aligned}
&\varepsilon_2(\mathbf{H})\\
&=\min_{\mathbf{H}}\left\{\begin{aligned}&E(\mathbf{H})=\frac{\mu_H}{2}\int_\Omega\left(\mathbf{H}-\nabla^2u+\frac{\boldsymbol{\lambda}_H}{\mu_H}\right)^2dx\\&+\frac{\mu_t}{2}\int_\Omega\left(v_T-\mathbf{T}\cdot(\mathbf{HT})+\frac{\lambda_t}{\mu_t}\right)^2dx\\&+\frac{\mu_n}{2}\int_\Omega\left(v_N-\mathbf{N}\cdot(\mathbf{HN})+\frac{\lambda_n}{\mu_n}\right)^2dx\end{aligned}\right\},
\end{aligned}
\tag{3.2.4}
$$

$$
\begin{aligned}
&\varepsilon_3(v_T)\\
&=\min_{v_T}\left\{\begin{aligned}&E(v_T)=\gamma_t\int_\Omega g_T\left|v_T\right|dx\\&+\frac{\mu_t}{2}\int_\Omega\left(v_T-\mathbf{T}\cdot(\mathbf{HT})+\frac{\lambda_t}{\mu_t}\right)^2dx\end{aligned}\right\},
\end{aligned}
\tag{3.2.5}
$$

$$
\begin{aligned}
&\varepsilon_4(v_N)\\
&=\min_{v_N}\left\{\begin{aligned}&E(v_N)=\gamma_n\int_\Omega g_N\left|v_N\right|dx\\&+\frac{\mu_n}{2}\int_\Omega\left(v_N-\mathbf{N}\cdot(\mathbf{HN})+\frac{\lambda_n}{\mu_n}\right)^2dx\end{aligned}\right\}.
\end{aligned}
\tag{3.2.6}
$$

The iterative solving process is decomposed into following five sub-processes.

Sub 1: Fixed $\mathbf{H}^k$, v_T^k, v_N^k to calculate u^{k+1}. Euler–Lagrange equation of (3.2.3) is

$$\gamma\left(u^{k+1}-f\right)+\mu_H\left(\nabla^2 u^{k+1}-\mathbf{H}^k-\frac{\boldsymbol{\lambda}_H{}^k}{\mu_H}\right) + \frac{1}{\mu_H}(u^{k+1}-u^k)=0. \tag{3.2.7}$$

By introducing auxiliary vector variable

$$\mathbf{F}^k=\begin{pmatrix}\mathbf{F}_\mathbf{1}^k\\ \mathbf{F}_\mathbf{2}^k\end{pmatrix}=\mathbf{H}^k+\frac{\boldsymbol{\lambda}_H^k}{\mu_H}=\begin{pmatrix}\mathbf{H}_\mathbf{1}^k+\frac{\boldsymbol{\lambda}_{H1}^k}{\mu_H} & \mathbf{H}_\mathbf{2}^k+\frac{\boldsymbol{\lambda}_{H2}^k}{\mu_H}\\ \mathbf{H}_\mathbf{3}^k+\frac{\boldsymbol{\lambda}_{H3}^k}{\mu_H} & \mathbf{H}_\mathbf{4}^k+\frac{\boldsymbol{\lambda}_{H4}^k}{\mu_H}\end{pmatrix},$$

(3.2.7) can be transformed into following iterative formulation

$$\gamma\left(u^{k+1}-f\right)+\mu_H\begin{pmatrix}\nabla\cdot\left(\nabla u_{x_1}^{k+1}+\mathbf{F}_\mathbf{1}^k\right)\\ \nabla\cdot\left(\nabla u_{x_2}^{k+1}+\mathbf{F}_\mathbf{2}^k\right)\end{pmatrix} + \frac{1}{\mu_H}(u^{k+1}-u^k)=0. \tag{3.2.8}$$

(3.2.8) can be rewritten as

$$\gamma\left(u^{k+1}-f\right)+\mu_H\Delta^2 u+\nabla\cdot\begin{pmatrix}\nabla\cdot\mathbf{F}_\mathbf{1}^k\\ \nabla\cdot\mathbf{F}_2^k\end{pmatrix}+\tfrac{1}{\mu_H}(u^{k+1}-u^k)=0. \tag{3.2.9}$$

(3.2.9) can be efficiently solved by FFT

$$u_{i,j}^{k+1}=\Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{2i,j})}{\varepsilon_2}\right)\right), \tag{3.2.10}$$

with $G_{2i,j}=\gamma f_{i,j}-\mu_H\nabla\cdot\begin{pmatrix}\nabla\cdot\mathbf{F}_\mathbf{1i,j}^k\\ \nabla\cdot\mathbf{F}_\mathbf{2i,j}^k\end{pmatrix}+\mu_H u$ and $\varepsilon_2=\gamma+4\mu_H(\cos\frac{2\pi s}{N}+\cos\frac{2\pi r}{M}-2)^2+\mu_H$.

Sub 2: Fixed u^{k+1}, v_T^k, v_N^k to calculate $\mathbf{H}^{k+1}$. By introducing variables $\mathbf{G}_H^k=\nabla^2 u^{k+1}-\frac{\boldsymbol{\lambda}_H^k}{\mu_H}$, $G_T^k=v_T^k+\frac{\lambda_t^k}{\mu_t}$, $G_N^k=v_N^k+\frac{\lambda_n^k}{\mu_n}$, the Euler–Lagrange equation of (3.2.4) is

$$\mu_H\left(\mathbf{H}^{k+1}-\mathbf{G}_H{}^k\right)+\mu_t\left(\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)-G_T^k\right)\begin{pmatrix}T_1\mathbf{T}^T\\ T_2\mathbf{T}^T\end{pmatrix} + \mu_n\left(\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)-G_N^k\right)\begin{pmatrix}N_1\mathbf{N}^T\\ N_2\mathbf{N}^T\end{pmatrix}=0. \tag{3.2.11}$$

with

$$\begin{aligned}
&\mathbf{H}^{k+1}-\mathbf{G}_H^k=\begin{pmatrix}H_1^{k+1}-G_{H_1}^k & H_2^{k+1}-G_{H_2}^k\\ H_3^{k+1}-G_{H_3}^k & H_4^{k+1}-G_{H_4}^k\end{pmatrix},\\
&\left(\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)-G_T^k\right)\begin{pmatrix}T_1\mathbf{T}^T\\ T_2\mathbf{T}^T\end{pmatrix}\\
&\quad=(T_1^2H_1^{k+1}+T_1T_2H_3^{k+1}+T_1T_2H_2^{k+1}\\
&\qquad+T_2^2H_4^{k+1}-G_T^k)\begin{pmatrix}T_1^2 & T_1T_2\\ T_1T_2 & T_2^2\end{pmatrix},\\
&\left(\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)-G_N^k\right)\begin{pmatrix}N_1\mathbf{N}^T\\ N_2\mathbf{N}^T\end{pmatrix}\\
&\quad=(N_1^2H_1^{k+1}+N_1N_2H_3^{k+1}+N_1N_2H_2^{k+1}\\
&\qquad+N_2^2H_4^{k+1}-G_N^k)\begin{pmatrix}N_1^2 & N_1N_2\\ N_1N_2 & N_2^2\end{pmatrix}.
\end{aligned} \tag{3.2.12}$$

Substitute (3.2.12) into (3.2.11), then $\mathbf{H}^{k+1}$ can be obtained.

Sub 3: Fixed u^{k+1}, $\mathbf{H}^{k+1}$, v_N^k to calculate v_T^{k+1}. (3.2.5) can be solved via generalized soft thresholding equation

$$\begin{aligned}v_T^{k+1}&=\max\left(\left|\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)-\frac{\lambda_t^k}{\mu_t}\right|-\gamma_t\frac{g_T}{\mu_t},0\right)\\ &\quad\times\frac{\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)-\frac{\lambda_t^k}{\mu_t}}{\left|\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)-\frac{\lambda_t^k}{\mu_t}\right|}.\end{aligned} \tag{3.2.13}$$

Sub 4: Fixed u^{k+1}, $\mathbf{H}^{k+1}$, v_T^{k+1} to calculate v_N^{k+1}. (3.2.6) can be solved via generalized soft thresholding equation

$$\begin{aligned}v_N^{k+1}&=\max\left(\left|\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)-\frac{\lambda_n^k}{\mu_N}\right|-\gamma_n\frac{g_N}{\mu_n},0\right)\\ &\quad\times\frac{\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)-\frac{\lambda_n^k}{\mu_n}}{\left|\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)-\frac{\lambda_n^k}{\mu_n}\right|}.\end{aligned} \tag{3.2.14}$$

Sub 5: Update Lagrange multipliers $\boldsymbol{\lambda}_H^{k+1}$, λ_t^{k+1}, λ_n^{k+1} in (3.2.2)

$$\begin{cases}\lambda_\mathrm{t}^{k+1}=\lambda_t^k+\mu_t\left(v_T^{k+1}-\mathbf{T}\cdot\left(\mathbf{H}^{k+1}\mathbf{T}\right)\right)\\ \lambda_n^{k+1}=\lambda_n^k+\mu_n\left(v_N^{k+1}-\mathbf{N}\cdot\left(\mathbf{H}^{k+1}\mathbf{N}\right)\right).\\ \boldsymbol{\lambda}_H^{k+1}=\boldsymbol{\lambda}_H^k+\mu_H\left(\mathbf{H}^{k+1}-\nabla^2u^{k+1}\right)\end{cases} \tag{3.2.15}$$

3.3 DOTGV Model

Inspired by TGV model, higher derivative of this result obtained by first-order anisotropic variation is intended to estimate orientations. A novel high-order anisotropic variation named double-orientational total generalized variation (DOTGV) model is proposed.

$$\min\left\{\begin{aligned}&E_3(u)=\frac{1}{2}\int_\Omega(u-f)^2dx+\gamma_w\int_\Omega g_T\left|P_T-\mathbf{T}\cdot\nabla u\right|dx\\ &+\gamma_v\int_\Omega g_N\left|P_N-\mathbf{N}\cdot\nabla u\right|dx\\ &+\gamma_s\int_\Omega g_T\left|\mathbf{T}\cdot\nabla P_T\right|dx+\gamma_q\int_\Omega g_N\left|\mathbf{N}\cdot\nabla P_N\right|dx.\end{aligned}\right\} \tag{3.3.1}$$

where $\gamma_w, \gamma_v, \gamma_s, \gamma_q$ are penalty parameters. By introducing auxiliary variables $w = P_T - \mathbf{T} \cdot \nabla u$, $v = P_N - \mathbf{N} \cdot \nabla u$, $s = \mathbf{T} \cdot \nabla P_T$ and $q = \mathbf{N} \cdot \nabla P_N$, constrained optimization problem (3.3.1) can be efficiently solved with following the augmented Lagrangian formulation.

$$
\begin{aligned}
(u, w, v, s, q, P_T, P_N) &= \underset{u,w,v,s,q,P_T,P_N}{\arg\min} \; E_3 \\
&= \underset{u,w,v,s,q,P_T,P_N}{\arg\min} \left\{ \begin{array}{l}
\frac{1}{2} \int_\Omega (u-f)^2 dx + \gamma_w \int_\Omega g_T \, |w| \, dx \\
+ \gamma_v \int_\Omega g_N \, |v| \, dx \\
+ \gamma_s \int_\Omega g_T \, |s| \, dx + \gamma_q \int_\Omega g_N \, |q| \, dx \\
+ \frac{\mu_w}{2} \int_\Omega \left(w - P_T + \mathbf{T} \cdot \nabla u + \frac{\lambda_w}{\mu_w} \right)^2 dx \\
+ \frac{\mu_v}{2} \int_\Omega \left(v - P_N + \mathbf{N} \cdot \nabla u + \frac{\lambda_v}{\mu_v} \right)^2 dx \\
+ \frac{\mu_s}{2} \int_\Omega \left(s - \mathbf{T} \cdot \nabla P_T + \frac{\lambda_s}{\mu_s} \right)^2 dx \\
+ \frac{\mu_q}{2} \int_\Omega \left(q - \mathbf{N} \cdot \nabla P_N + \frac{\lambda_q}{\mu_q} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.2}
$$

where $\mu_w, \mu_v, \mu_s, \mu_q$ are positive penalty parameters, λ_w, λ_v, λ_s, λ_q are Lagrange multipliers. (3.3.2) is changed into following sub-problems.

$$
\begin{aligned}
&\varepsilon_1(u) \\
&= \min_u \left\{ \begin{array}{l}
E(u) = \frac{1}{2} \int_\Omega (u-f)^2 dx \\
+ \frac{\mu_w}{2} \int_\Omega \left(w^k - P_T^k + \mathbf{T} \cdot \nabla u + \frac{\lambda_w^k}{\mu_w} \right)^2 dx \\
+ \frac{\mu_v}{2} \int_\Omega \left(v^k - P_N^k + \mathbf{N} \cdot \nabla u + \frac{\lambda_v^k}{\mu_v} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.3}
$$

$$
\begin{aligned}
&\varepsilon_2(w) \\
&= \min_w \left\{ \begin{array}{l}
E(w) = \gamma_w \int_\Omega g_T \, |w| \, dx + \frac{\mu_w}{2} \\
\int_\Omega \left(w - P_T + \mathbf{T} \cdot \nabla u + \frac{\lambda_w}{\mu_w} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.4}
$$

$$
\begin{aligned}
&\varepsilon_3(v) \\
&= \min_v \left\{ \begin{array}{l}
E(v) = \gamma_v \int_\Omega g_N \, |v| \, dx \\
+ \frac{\mu_v}{2} \int_\Omega \left(v - P_N + \mathbf{N} \cdot \nabla u + \frac{\lambda_v}{\mu_v} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.5}
$$

$$
\begin{aligned}
&\varepsilon_4(s) \\
&= \min_s \left\{ \begin{array}{l}
E(w_s) = \gamma_s \int_\Omega g_T \, |s| \, dx \\
+ \frac{\mu_s}{2} \int_\Omega \left(s - \mathbf{T} \cdot \nabla P_T + \frac{\lambda_s}{\mu_s} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.6}
$$

$$
\begin{aligned}
&\varepsilon_4(q) \\
&= \min_q \left\{ \begin{array}{l}
E(q) = \gamma_q \int_\Omega g_N \, |q| \, dx \\
+ \frac{\mu_q}{2} \int_\Omega \left(q - \mathbf{N} \cdot \nabla P_N + \frac{\lambda_q}{\mu_q} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.7}
$$

$$
\begin{aligned}
&\varepsilon_5(P_T) \\
&= \min_{P_T} \left\{ \begin{array}{l}
E(P_T) \\
= \frac{\mu_w}{2} \int_\Omega \left(w - P_T + \mathbf{T} \cdot \nabla u + \frac{\lambda_w}{\mu_w} \right)^2 dx \\
+ \frac{\mu_s}{2} \int_\Omega \left(s - \mathbf{T} \cdot \nabla P_T + \frac{\lambda_s}{\mu_s} \right)^2 dx
\end{array} \right\},
\end{aligned}
\tag{3.3.8}
$$

$$
\begin{aligned}
&\varepsilon_6(P_N) \\
&= \min_{P_N} \left\{ \begin{array}{l}
E(P_N) = \frac{\mu_v}{2} \int_\Omega \left(v - P_N + \mathbf{N} \cdot \nabla u + \frac{\lambda_v}{\mu_v} \right)^2 dx \\
+ \frac{\mu_q}{2} \int_\Omega \left(q - \mathbf{N} \cdot \nabla P_N + \frac{\lambda_q}{\mu_q} \right)^2 dx
\end{array} \right\}.
\end{aligned}
\tag{3.3.9}
$$

The iterative solving process is decomposed into following eight sub-processes.

Sub 1: Fixed $w^k, v^k, s^k, q^k, P_T^k, P_N^k$ to calculate u^{k+1}. Euler–Lagrange equation of (3.3.3) is

$$
\begin{aligned}
&\underbrace{\mu_w \nabla \cdot \begin{pmatrix} T_1 \left(w^k - P_T^k + \frac{\lambda_w^k}{\mu_w} \right) \\ T_2 \left(w^k - P_T^k + \frac{\lambda_w^k}{\mu_w} \right) \end{pmatrix} + \mu_v \nabla \cdot \begin{pmatrix} N_1 \left(v^k - P_N^k + \frac{\lambda_v^k}{\mu_v} \right) \\ N_2 \left(v^k - P_N^k + \frac{\lambda_v^k}{\mu_v} \right) \end{pmatrix}}_{Gu} \\
&\qquad + \left(u^{k+1} - f \right) + (\mu_w \mathbf{T} \cdot \mathbf{T} + \mu_v \mathbf{N} \cdot \mathbf{N}) \, \Delta u^{k+1} = 0.
\end{aligned}
\tag{3.3.10}
$$

(3.3.10) can be efficiently solved by FFT

$$
u_{i,j}^{k+1} = \Re \left(\mathcal{F}^{-1} \left(\frac{\mathcal{F}(G_{3i,j})}{\varepsilon_3} \right) \right),
\tag{3.3.11}
$$

with $G_{3i,j} = \gamma f_{i,j} - Gu$ and

$$
\varepsilon_3 = 1 - 2 \left(\mu_w \mathbf{T} \cdot \mathbf{T} + \mu_v \mathbf{N} \cdot \mathbf{N} \right) \left(\cos \frac{2\pi s}{N} + \cos \frac{2\pi r}{M} - 2 \right).
$$

Sub 2: Fixed $u^{k+1}, v^k, s^k, q^k, P_T^k, P_N^k$ to calculate w^{k+1}. (3.3.4) can be solved via generalized soft thresholding equation

$$
\begin{aligned}
w^{k+1} &= \max \left(\left| P_T^k + \mathbf{T} \cdot \nabla u^{k+1} - \frac{\gamma_w \lambda_w^k}{\mu_w} \right| - \frac{g_T}{\mu_w}, 0 \right) \\
&\quad \times \frac{P_T^k + \mathbf{T} \cdot \nabla u^{k+1} - \frac{\lambda_w^k}{\mu_w}}{\left| P_T^k + \mathbf{T} \cdot \nabla u^{k+1} - \frac{\lambda_w^k}{\mu_w} \right|}, 0 \frac{0}{|0|}.
\end{aligned}
\tag{3.3.12}
$$

Sub 3: Fixed u^{k+1}, w^{k+1}, s^k, q^k, ${P_T}^k$, ${P_N}^k$ to calculate v^{k+1}. (3.3.5) can be solved via generalized soft thresholding equation

$$\begin{aligned} v^{k+1} &= \max\left(\left|P_N^k + \mathbf{N}\cdot\nabla u^{k+1} - \frac{\lambda_v^k}{\mu_v}\right| - \frac{\gamma_v g_N}{\mu_v}, 0\right) \\ &\times \frac{P_N^k + \mathbf{N}\cdot\nabla u^{k+1} - \frac{\lambda_v^k}{\mu_v}}{\left|P_N^k + \mathbf{N}\cdot\nabla u^{k+1} - \frac{\lambda_v^k}{\mu_v}\right|}, 0\frac{0}{|0|}. \end{aligned} \tag{3.3.13}$$

Sub 4: Fixed u^{k+1}, w^{k+1}, v^{k+1}, q^k, P_T^k, P_N^k to calculate s^{k+1}. (3.3.6) can be solved via generalized soft thresholding equation

$$\begin{aligned} s^{k+1} &= \max\left(\left|\mathbf{T}\cdot\nabla P_T^k - \frac{\lambda_s^k}{\mu_s}\right| - \frac{\gamma_s g_T}{\mu_s}, 0\right) \\ &\times \frac{\mathbf{T}\cdot\nabla P_T^k - \frac{\lambda_s^k}{\mu_s}}{\left|\mathbf{T}\cdot\nabla P_T^k - \frac{\lambda_s^k}{\mu_s}\right|}, 0\frac{0}{|0|}. \end{aligned} \tag{3.3.14}$$

Sub 5: Fixed u^{k+1}, w^{k+1}, v^{k+1}, s^{k+1}, P_T^k, P_N^k to calculate q^{k+1}. (3.3.7) can be solved via generalized soft thresholding equation

$$\begin{aligned} q^{k+1} &= \max\left(\left|\mathbf{N}\cdot\nabla P_N^k + \frac{\lambda_q^k}{\mu_q}\right| - \frac{\gamma_q g_N}{\mu_q}, 0\right) \\ &\times \frac{\mathbf{N}\cdot\nabla P_N^k + \frac{\lambda_q^k}{\mu_q}}{\left|\mathbf{N}\cdot\nabla P_N^k + \frac{\lambda_q^k}{\mu_q}\right|}, 0\frac{0}{|0|}. \end{aligned} \tag{3.3.15}$$

Sub 6: Fixed u^{k+1}, w^{k+1}, v^{k+1}, s^{k+1}, q^{k+1}, P_N^k to calculate P_T^{k+1}. Euler–Lagrange equation of (3.3.8) is

$$\begin{aligned} &\underbrace{-\mu_w\left(w^{k+1} + \mathbf{T}\cdot\nabla u^{k+1} + \frac{\lambda_w^k}{\mu_w}\right) + \mu_s\nabla\cdot\begin{pmatrix} T_1\left(s^{k+1} + \frac{\lambda_s^k}{\mu_s}\right) \\ T_2\left(s^{k+1} + \frac{\lambda_s^k}{\mu_s}\right)\end{pmatrix}}_{GP_T} \\ &\quad + \mu_w P_T^{k+1} - \mu_s\mathbf{T}\cdot\mathbf{T}\Delta P_T^{k+1} = 0. \end{aligned} \tag{3.3.16}$$

(3.3.16) can be efficiently solved by FFT

$$ {P_T}_{i,j}^{k+1} = R\left(F^{-1}\left(\frac{F(G_{4i,j})}{\varepsilon_4}\right)\right), \tag{3.3.17}$$

with $G_{4i,j} = -\mu_s G{P_T}_{i,j}$ and $\varepsilon_4 = 1 - 2\mu_s\left(T_1^2 + T_2^2\right)$ $(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2)$.

Sub 7: Fixed u^{k+1}, w^{k+1}, v^{k+1}, s^{k+1}, q^{k+1}, P_T^{k+1} to calculate P_N^{k+1}. Euler–Lagrange equation of (3.3.9) is

$$\begin{aligned} &\underbrace{-\mu_v\left(v^{k+1} + \mathbf{N}\cdot\nabla u^{k+1} + \frac{\lambda_v^k}{\mu_v}\right) + \mu_q\nabla\cdot\begin{pmatrix} T_1\left(q^{k+1} + \frac{\lambda_q^k}{\mu_q}\right) \\ T_2\left(q^{k+1} + \frac{\lambda_q^k}{\mu_q}\right)\end{pmatrix}}_{GP_N} \\ &\quad + \mu_v P_N^{k+1} - \mu_q\mathbf{N}\cdot\mathbf{N}\Delta P_N^{k+1} = 0. \end{aligned} \tag{3.3.18}$$

(3.3.18) can be efficiently solved by FFT

$$ {P_N}_{i,j}^{k+1} = \Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{5i,j})}{\varepsilon_5}\right)\right), \tag{3.3.19}$$

with $G_{5i,j} = -\mu_s G{P_N}_{i,j}$ and

$$\varepsilon_5 = 1 - 2\mu_q\left(N_1^2 + N_2^2\right)(\cos\frac{2\pi s}{N} + \cos\frac{2\pi r}{M} - 2).$$

Sub 8: Update Lagrange multipliers λ_w^{k+1},λ_v^{k+1}, λ_s^{k+1},λ_q^{k+1} in (3.3.2)

$$\begin{cases} \lambda_w^{k+1} = \lambda_w^k + \mu_w\left(w^{k+1} - P_T^{k+1} + \mathbf{T}\cdot\nabla u^{k+1}\right) \\ \lambda_v^{k+1} = \lambda_v^k + \mu_v\left(v^{k+1} - P_N^{k+1} + \mathbf{N}\cdot\nabla u^{k+1}\right) \\ \lambda_s^{k+1} = \lambda_s^k + \mu_s\left(s^{k+1} - \mathbf{T}\cdot\nabla P_T^{k+1}\right) \\ \lambda_q^{k+1} = \lambda_q^k + \mu_q\left(q^{k+1} - \mathbf{N}\cdot\nabla P_N^{k+1}\right) \end{cases}. \tag{3.3.20}$$

3.4 DOTBH Model

In the end, the regularizer combines anisotropic TV and bounded Hessian which called double-orientational TV and bounded Hessian (DOTBH) model is proposed.

$$\begin{aligned} \min E_4(u) &= \min\left\{E_1(u) + E_2(u)\right\} \\ &= \min\left\{\begin{aligned} &\gamma_T\int_\Omega g_T|u_T|dx + \gamma_N\int_\Omega g_N|u_N|dx \\ &+\gamma_t\int_\Omega g_T|u_{TT}|dx + \gamma_n\int_\Omega g_N|u_{NN}|dx \\ &+\frac{\gamma}{2}\int_\Omega (f-u)^2dx \end{aligned}\right\} \\ &s.t.\ u_T = \mathbf{T}\cdot\nabla u, u_N = \mathbf{N}\cdot\nabla u, \\ &v_T = u_{TT} = \mathbf{T}\cdot(\mathbf{HT}), v_N = u_{NN} = \mathbf{N}\cdot(\mathbf{HN}). \end{aligned} \tag{3.4.1}$$

The constrained optimization problem (3.4.1) can be efficiently solved with following the augmented Lagrangian formulation.

$$\begin{aligned} &(u, \mathbf{w}, w_T, w_N, \mathbf{H}, v_T, v_N) \\ &= \underset{u,\mathbf{w},w_T,w_N\mathbf{H},v_T,v_N}{\arg\min} E_4 = \underset{u,\mathbf{w},w_T,w_N}{\arg\min} E_1 + \underset{u,\mathbf{H},v_T,v_N}{\arg\min} E_2. \end{aligned} \tag{3.4.2}$$

Iterative evaluation of auxiliary variables $\mathbf{w}, w_T, w_N$, $\mathbf{H}, v_T, v_N, \lambda_T, \lambda_{N,}, \lambda_t, \lambda_n$ can refer to the DOTV and DOBH corresponding to (3.3.1) and (3.2.15). Euler–Lagrange equation of (3.4.3) to calculate is

$$\gamma\left(u^{k+1}-f\right)+\mu_H\Delta^2 u-\mu\Delta u^{k+1}+\nabla\cdot\begin{pmatrix}\nabla\cdot\mathbf{F}_\mathbf{1}^k\\ \nabla\cdot\mathbf{F}_2^k\end{pmatrix}$$
$$+\mu\left(\nabla\cdot\left(\mathbf{w}^k+\frac{\boldsymbol{\lambda}^k}{\mu}\right)\right)+\frac{1}{\mu_H}(u^{k+1}-u^k)=0, \tag{3.4.3}$$

with $\mathbf{F}^k=\begin{pmatrix}\mathbf{F}_\mathbf{1}^k\\ \mathbf{F}_\mathbf{2}^k\end{pmatrix}=\mathbf{H}^k+\frac{\overset{\smile k}{\mathbf{H}}}{\mu_H}$ (3.4.3) can be efficiently solved by FFT

$$u_{i,j}{}^{k+1}=\Re\left(\mathcal{F}^{-1}\left(\frac{\mathcal{F}(G_{6i,j})}{\varepsilon_6}\right)\right), \tag{3.4.4}$$

with

$$G_{6i,j}=\gamma f_{i,j}-\mu_H\nabla\cdot\begin{pmatrix}\nabla\cdot\mathbf{F}_{\mathbf{1i,j}}^k\\ \nabla\cdot\mathbf{F}_{\mathbf{2i,j}}^k\end{pmatrix}+\mu_H u-\nabla\cdot\boldsymbol{\lambda}_{i,j}^k$$
$$-\mu\nabla\cdot\mathbf{w}_{i,j}^k,$$

and

$$\varepsilon_6=\gamma+4\mu_H(\cos\frac{2\pi s}{N}+\cos\frac{2\pi r}{M}-2)^2+\mu_H$$
$$-2\mu(\cos\frac{2\pi s}{N}+\cos\frac{2\pi r}{M}-2).$$

3.5 Accelerating the Algorithm

In order to monotonically decrease cost function, FISTA [45], TWIST [46], Fast ADMM [47] can be adapted to accelerate convergence. In this section, the algorithm of Fast ADMM with restart [47] is introduced as follows. The accelerated method of Fast ADMM is a simple ADMM with a predictor-corrector-type acceleration step. Fast ADMM with restart algorithm enforce stability by a restart rule. A simple restart rule is introduced to ensure the convergence of general objective and monotonous decrease of function value. The weakly convex problem is settled by introducing a step-skipping process to selectively accelerate certain iterative applications.

The overall implementation of fast ADMM with restart algorithm is applied to the proposed models. The restart rule relies on a combined residual which measures both the original and double errors. $\mathbf{d}$ denotes the accelerated rules satisfied the judgment condition $c_k<\eta c_{k-1}$ with $\eta\in(0,1)$. $\mathbf{e}$ denotes the restarted rules. During the acceleration, accelerated rules and restarted rules should be done as follows:

$$\mathbf{d}^{k+1}=\mathbf{d}^{k+1}+\frac{\alpha^k-1}{\alpha^{k+1}}(\mathbf{d}^{k+1}-\mathbf{d}^k), \tag{3.5.1}$$

with $\alpha^{k+1}=\frac{1+\sqrt{1+4(\alpha^k)^2}}{2}$, where $\alpha^1=1$,

$$\begin{cases}\alpha^{k+1}=1\\ \mathbf{e}^{k+1}=\mathbf{e}^k\\ c^{k+1}=\dfrac{c^{k+1}}{\eta}\end{cases}. \tag{3.5.2}$$

Accelerated process for DOTV is as follows:

$$c_1^{k+1}=\mu\left\|\mathbf{w}^{k+1}-\mathbf{w}^k\right\|^2+\mu_T\left\|w_T^{k+1}-w_T^k\right\|^2$$
$$+\mu_N\left\|w_N^{k+1}-w_N^k\right\|^2+\mu^{-1}\left\|{}^{\smile k+1}-{}^{\smile k}\right\|^2 \tag{3.5.3}$$
$$+\mu_T^{-1}\left\|\lambda_T^{k+1}-\lambda_T^k\right\|^2+\mu_N^{-1}\left\|\lambda_N^{k+1}-\lambda_N^k\right\|^2,$$

$$\mathbf{d_1}=(\mathbf{w}, w_T, w_N, \boldsymbol{\lambda}, \lambda_T, \lambda_N), \tag{3.5.4}$$
$$\mathbf{e_1}=(\mathbf{w}, w_T, w_N, \boldsymbol{\lambda}, \lambda_T, \lambda_N). \tag{3.5.5}$$

Accelerated process for DOBH is as follows:

$$c_2^{k+1}=\mu_H\left\|\mathbf{H}^{k+1}-\mathbf{H}^k\right\|^2+\mu_t\left\|v_T^{k+1}-v_T{}^k\right\|^2$$
$$+\mu_n\left\|v_N^{k+1}-v_N^k\right\|^2+\mu_H^{-1}\left\|\boldsymbol{\lambda}_H^{k+1}-\boldsymbol{\lambda}_H^k\right\|^2 \tag{3.5.6}$$
$$+\mu_t^{-1}\left\|\lambda_t^{k+1}-\lambda_t^k\right\|^2+\mu_n^{-1}\left\|\lambda_n^{k+1}-\lambda_n^k\right\|^2,$$

$$\mathbf{d_2}=(\mathbf{H}, v_T, v_N, \boldsymbol{\lambda}_H, \lambda_t, \lambda_n), \tag{3.5.7}$$
$$\mathbf{e_2}=(\mathbf{H}, v_T, v_N, \boldsymbol{\lambda}_H, \lambda_t, \lambda_n). \tag{3.5.8}$$

Accelerated process for DOTGV is as follows:

$$c_3^{k+1}=\mu_w\left\|w^{k+1}-w^k\right\|^2+\mu_v\left\|v^{k+1}-v^k\right\|^2$$
$$+\mu_s\left\|s^{k+1}-s^k\right\|^2+\mu_q\left\|q^{k+1}-q^k\right\|^2$$
$$+\mu_w^{-1}\left\|\lambda_w^{k+1}-\lambda_w^k\right\|^2+\mu_v^{-1}\left\|\lambda_v^{k+1}-\lambda_v^k\right\|^2 \tag{3.5.9}$$
$$+\mu_s^{-1}\left\|\lambda_s^{k+1}-\lambda_s^k\right\|^2+\mu_q^{-1}\left\|\lambda_q^{k+1}-\lambda_q^k\right\|^2,$$

$$\mathbf{d_3}=(w, v, s, q, \lambda_w, \lambda_v, \lambda_s, \lambda_q), \tag{3.5.10}$$
$$\mathbf{e_3}=(w, v, s, q, \lambda_w, \lambda_v, \lambda_s, \lambda_q). \tag{3.5.11}$$

Accelerated process for DOTBH is as follows:

$$c_4^{k+1}=\mu_H\left\|\mathbf{H}^{k+1}-\mathbf{H}^k\right\|^2+\mu_t\left\|v_T^{k+1}-v_T^k\right\|^2$$
$$+\mu_n\left\|v_N^{k+1}-v_N^k\right\|^2+\mu_H^{-1}\left\|\overset{\smile k+1}{\mathbf{H}}-\overset{\smile k}{\mathbf{H}}\right\|^2$$
$$+\mu_t^{-1}\left\|\lambda_t^{k+1}-\lambda_t^k\right\|^2+\mu_n^{-1}\left\|\lambda_n^{k+1}-\lambda_n^k\right\|^2 \tag{3.5.12}$$
$$+\mu\left\|\mathbf{w}^{k+1}-\mathbf{w}^k\right\|^2+\mu_T\left\|w_T^{k+1}-w_T^k\right\|^2$$
$$+\mu_N\left\|w_N^{k+1}-w_N^k\right\|^2+\mu^{-1}\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2$$
$$+\mu_T^{-1}\left\|\lambda_T^{k+1}-\lambda_T^k\right\|^2+\mu_N^{-1}\left\|\lambda_N^{k+1}-\lambda_N^k\right\|^2,$$

Table 1 Fast ADMM with restart for proposed models

Algorithm 1. Fast ADMM with restart for DOTV model

```
Step 1: Initialization: Set $(w_T, w_N, \mathbf{w}, \lambda_T, \lambda_N, \boldsymbol{\lambda}, c_1) = 0, (\gamma_T, \gamma_N, \mu, \mu_T, \mu_N, \alpha, \eta, iteration) > 0$
Step 2: repeat
Step 3: Compute coefficient $\mathbf{T}, \mathbf{N}, g_T, g_N$ according to (2.1.4), (2.1.5) and (2.1.8)
Step 4: Compute coefficient $u, \mathbf{w}, w_T, w_N$ for solving sub-problems according to (3.1.7)-(3.1.12)
Step 5: Update Lagrange multiplier $\boldsymbol{\lambda}, \lambda_T, \lambda_N$ according to (3.1.13)
Step 6: Accelerated process:
    If $c_1^{k+1} < \eta c_1^k$ then compute accelerated rules (3.5.1) with $\mathbf{d}_1$, else compute restart rules (3.5.2) with $e_1$.
Step 7: until convergence of $u$
Step 8: return $u$
Step 9: end function
```

Algorithm 2. Fast ADMM with restart for DOBH model

```
Step 1: Initialization: Set $(v_T, v_N, \mathbf{H}, \lambda_t, \lambda_n, \boldsymbol{\lambda}_H, c_2) = 0,(\gamma_t, \gamma_n, \gamma, \mu_H, \mu_t, \mu_n, \alpha, \eta, iteration) > 0$
Step 2: repeat
Step 3: Compute coefficient $\mathbf{T}, \mathbf{N}, g_T, g_N$ according to (2.1.4), (2.1.5) and (2.1.8)
Step 4: Compute coefficient $u, v_T, v_N, \mathbf{H}$ for solving sub-problems according to (3.2.7)-(3.2.14)
Step 5: Update Lagrange multiplier $\boldsymbol{\lambda}_H, \lambda_t, \lambda_n$ according to (3.2.15)
Step 6: Accelerated process:
    If $c_2^{k+1} < \eta c_2^k$ then compute accelerated rules (3.5.1) with $\mathbf{d}_2$, else compute restart rules (3.5.2) with $e_2$.
Step 7: until convergence of $u$
Step 8: return $u$
Step 9: end function
```

Algorithm 3. Fast ADMM with restart for DOTGV model

```
Step 1: Initialization: Set $(w, v, s, q, \lambda_w, \lambda_v, \lambda_s, \lambda_q, c_3) = 0,(\gamma_w, \gamma_v, \gamma_s, \gamma_q, \mu_w, \mu_v, \mu_s, \mu_q, \alpha, \eta, iteration) > 0$
Step 2: repeat
Step 3: Compute coefficient $\mathbf{T}, \mathbf{N}, g_T, g_N$ according to (2.1.4), (2.1.5) and (2.1.8)
Step 4: Compute coefficient $u, w, v, s, q, P_T, P_N$ for solving sub-problems according to (3.3.10)-(3.3.19)
Step 5: Update Lagrange multiplier $\lambda_t, \lambda_n, \lambda_s, \lambda_q$ according to (3.3.20)
Step 6: Accelerated process:
    If $c_3^{k+1} < \eta c_3^k$ then compute accelerated rules (3.5.1) with $\mathbf{d}_3$, else compute restart rules (3.5.2) with $e_3$.
Step 7: until convergence of $u$
Step 8: return $u$
Step 9: end function
```

Algorithm 4. Fast ADMM with restart for DOTBH model

```
Step 1: Initialization: Set $(w_T, w_N, \mathbf{w}, \lambda_T, \lambda_N, \boldsymbol{\lambda}, v_T, v_N, \mathbf{H}, \lambda_t, \lambda_n, \boldsymbol{\lambda}_H, c_4) = 0,(\gamma_T, \gamma_N, \mu, \mu_T, \mu_N, \gamma_t, \gamma_n, \gamma, \mu_H, \mu_t, \mu_n, \alpha, \eta, iteration) > 0$
Step 2: repeat
Step 3: Compute coefficient $\mathbf{T}, \mathbf{N}, g_T, g_N$ according to (2.1.4), (2.1.5) and (2.1.8)
Step 4: Compute coefficient $u, w_T, w_N, \mathbf{w}, v_T, v_N, \mathbf{H}$ for solving sub-problems according to (3.1.7)-(3.2.14)
Step 5: Update Lagrange multiplier $\boldsymbol{\lambda}_H, \lambda_t, \lambda_n, \boldsymbol{\lambda}, \lambda_T, \lambda_N$ according to (3.1.13) and (3.2.15)
Step 6: Accelerated process:
    If $c_4^{k+1} < \eta c_4^k$ then compute accelerated rules (3.5.1) with $\mathbf{d}_4$, else compute restart rules (3.5.2) with $e_4$.
Step 7: until convergence of $u$
Step 8: return $u$
Step 9: end function
```

$$\mathbf{d_4} = (\mathbf{H}, v_T, v_N, \boldsymbol{\lambda}_H, \lambda_t, \lambda_n, \mathbf{w}, w_T, w_N, \boldsymbol{\lambda}, \lambda_T, \lambda_N), \tag{3.5.13}$$

$$\mathbf{e_4} = (\mathbf{H}, v_T, v_N, \boldsymbol{\lambda}_H, \lambda_t, \lambda_n, \mathbf{w}, w_T, w_N, \boldsymbol{\lambda}, \lambda_T, \lambda_N). \tag{3.5.14}$$

In each iteration of the algorithm, combined residual c^{k+1} is computed first. If $c^{k+1} < \eta c^k$ then compute the accelerated rules (3.5.1) with $\mathbf{d}^{k+1}$, otherwise compute the restart rules (3.5.2) with $\mathbf{e}^{k+1}$.

In the proposed models, the stopping criteria are defined as

$$\frac{\left|E^{k+1} - E^k\right|}{E^k} < threshold, \tag{3.5.15}$$

where E^{k+1} and E^k are the energy values of the current step and preceding step. The value of $threshold$ for following experiments is 10^{-6} in all cases.

The solving processes for the proposed methods are summarized in Table 1.

4 Numerical Experiments

In this section, the qualitative and quantitative comparison of proposed methods is studied, respectively. The performance of our anisotropic variation models is demonstrated for the applications of image denoising. All experiments are per-

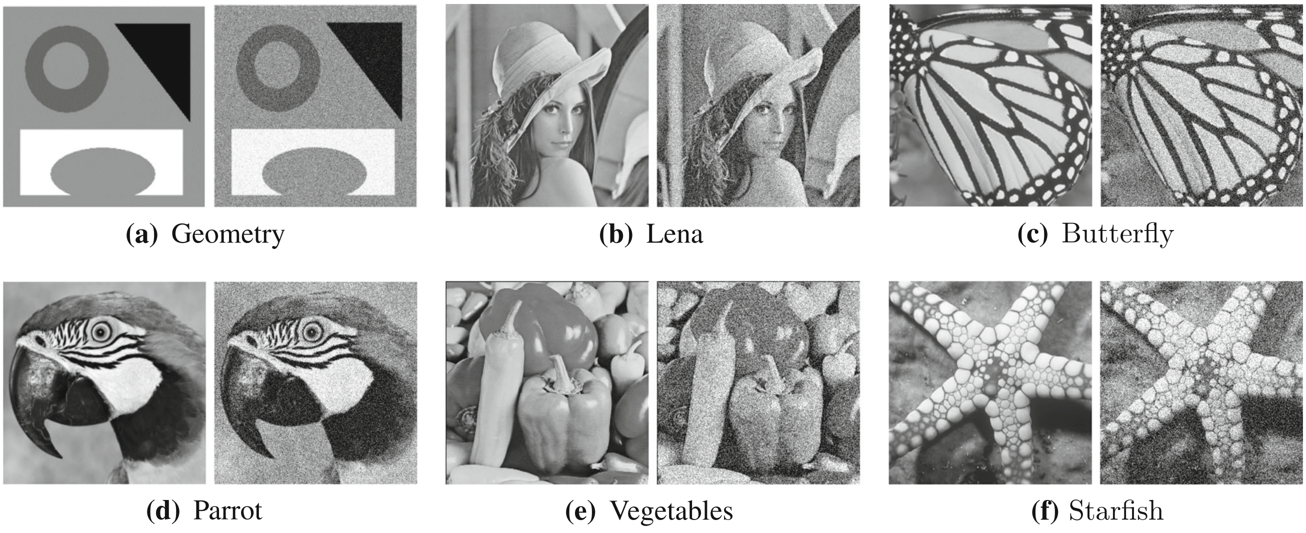

(a) Geometry **(b)** Lena **(c)** Butterfly **(d)** Parrot **(e)** Vegetables **(f)** Starfish

Fig. 1 Testing images and noisy images used in the numerical implementations. (**a**), (**b**) are corrupted by 10% Gaussian noise, (**c**)–(**f**) are corrupted by 15% Gaussian noise

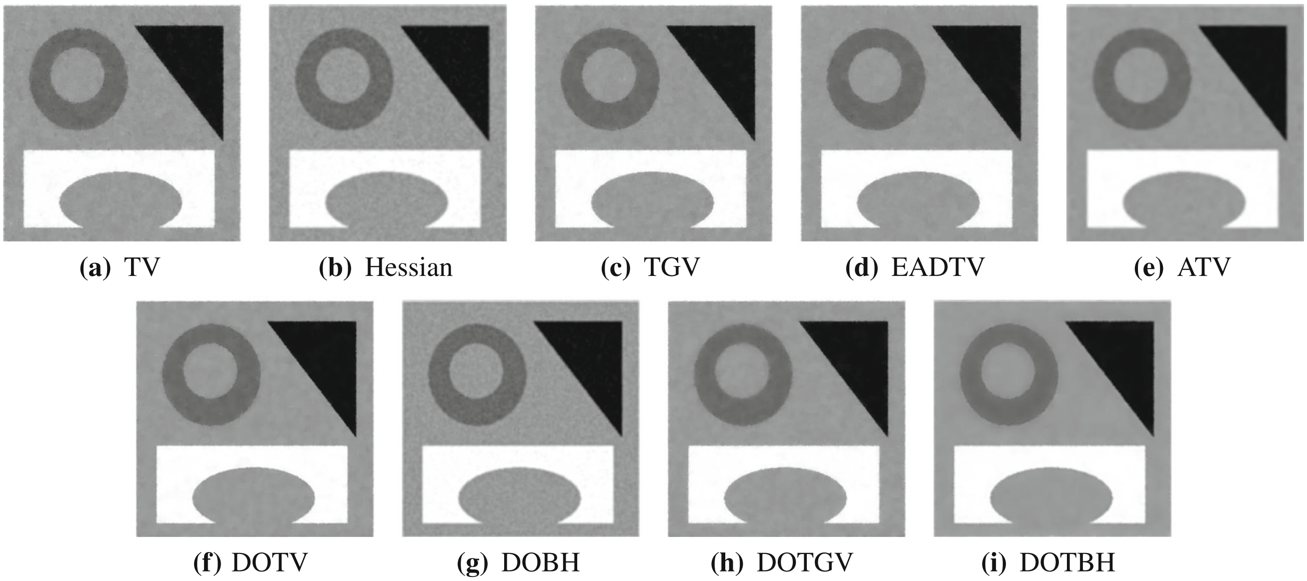
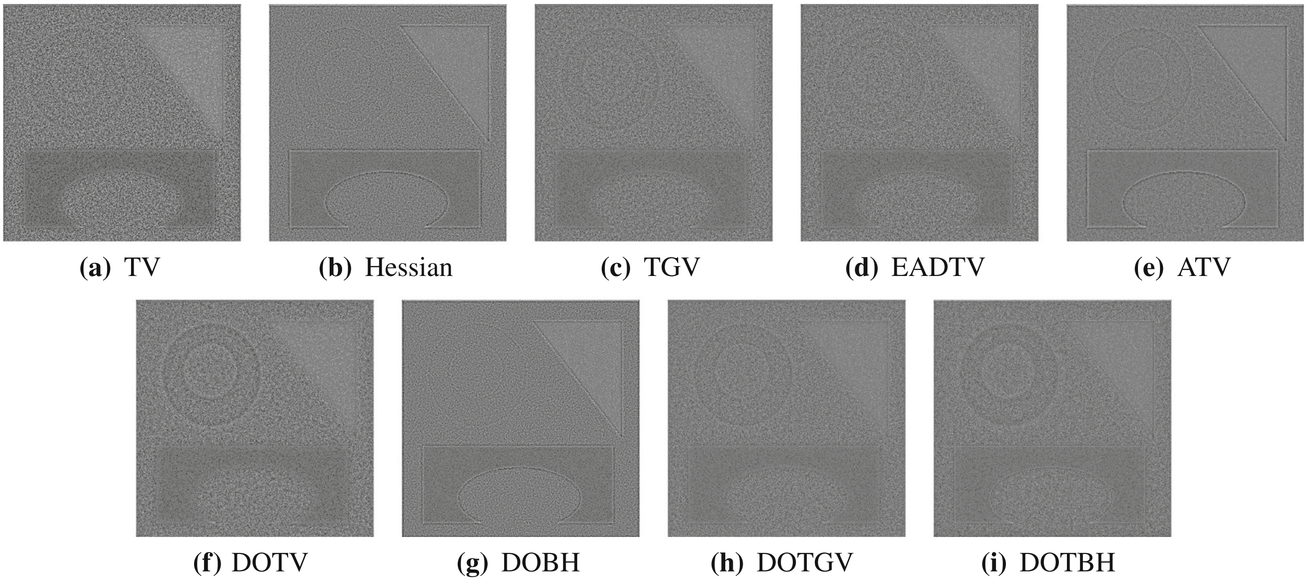
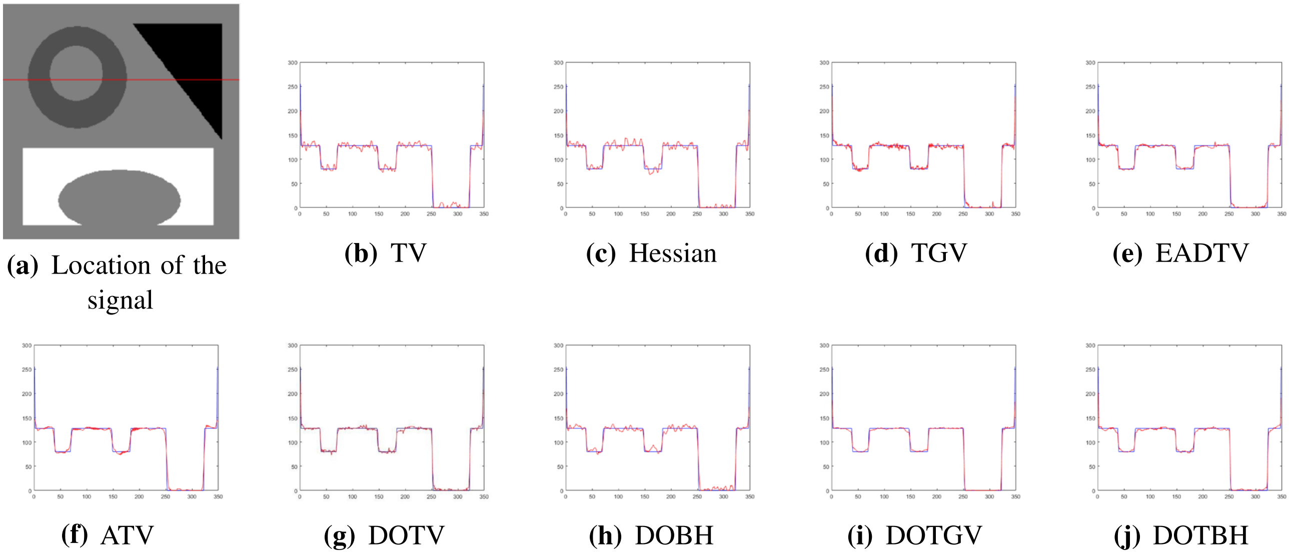

(a) TV **(b)** Hessian **(c)** TGV **(d)** EADTV **(e)** ATV **(f)** DOTV **(g)** DOBH **(h)** DOTGV **(i)** DOTBH

Fig. 2 Denoised images of geometry with 10% Gaussian noise


(a) TV **(b)** Hessian **(c)** TGV **(d)** EADTV **(e)** ATV

(f) DOTV **(g)** DOBH **(h)** DOTGV **(i)** DOTBH

Fig. 3 Residual images of denoised results in Fig. 2



(a) Location of the signal **(b)** TV **(c)** Hessian **(d)** TGV **(e)** EADTV

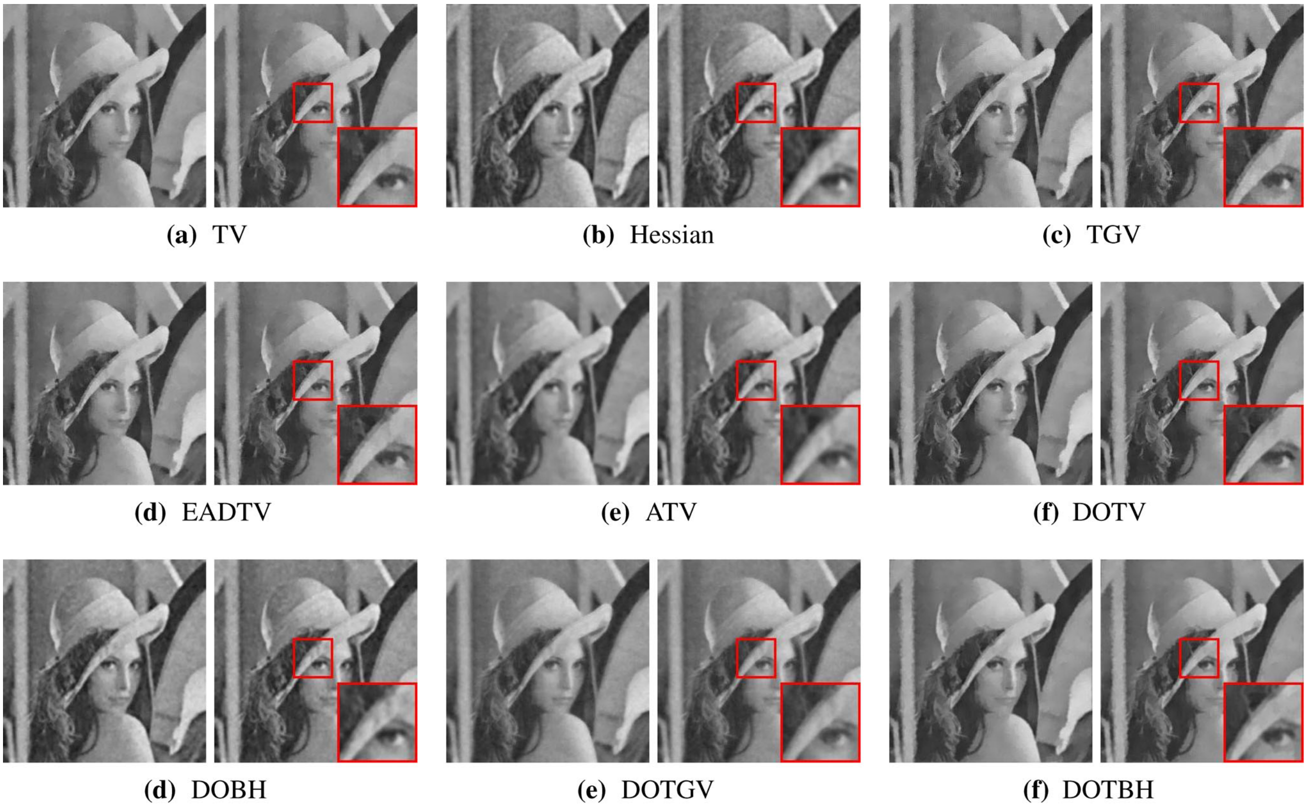
(f) ATV **(g)** DOTV **(h)** DOBH **(i)** DOTGV **(j)** DOTBH

Fig. 4 Location of the signal and plots of middle slices of the denoised images in Fig. 2

formed using Matlab 2018a on a Windows 10 PC with an Intel(R) Core (TM) i5-8500T at 2.1 GHz and 8 GB RAM.

Some remarks on the choice of tuning parameters are given before our numerical results are presented. Gaussian kernels with standard deviations σ, ρ are associated with noise scale, the bigger the noise, the larger the value should be. We set $\sigma = 0.5$, $\rho = 4$, which commonly bring for excellent restorational results. The parameters used in DOTV model are $\gamma_T = \gamma_N = 25, \mu_T = \mu_N = \mu = 1$. The parameters used in DOBH model are $\gamma_t = \gamma_n = 1, \gamma = 0.5, \mu_H = 1, \mu_t = 10, \mu_n = 0.1$. The parameters used in DOTGV model are $\gamma = \gamma_w = \gamma_s = 1, \gamma_v = 25, \gamma_q = 10, \mu_w = 0.1, \mu_v = \mu_q = 5, \mu_s = 1$. The parameters used in DOTV model refer to DOTV and DOBH models.

The test images and noisy images are shown in Fig. 1. The merits of proposed models are demonstrated through comparison with classic isotropic models including TV [4], BH [6], TGV [7] models, anisotropic models including EADTV [15], ATV [22], TDV [18] models and non-local denoising approach such as BM3D [25] model. All the models are objectively compared by peak signal-to-noise ratio (PSNR) and structure similarity index map (SSIM). Meanwhile, the preserving properties of structure features are compared subjectively.

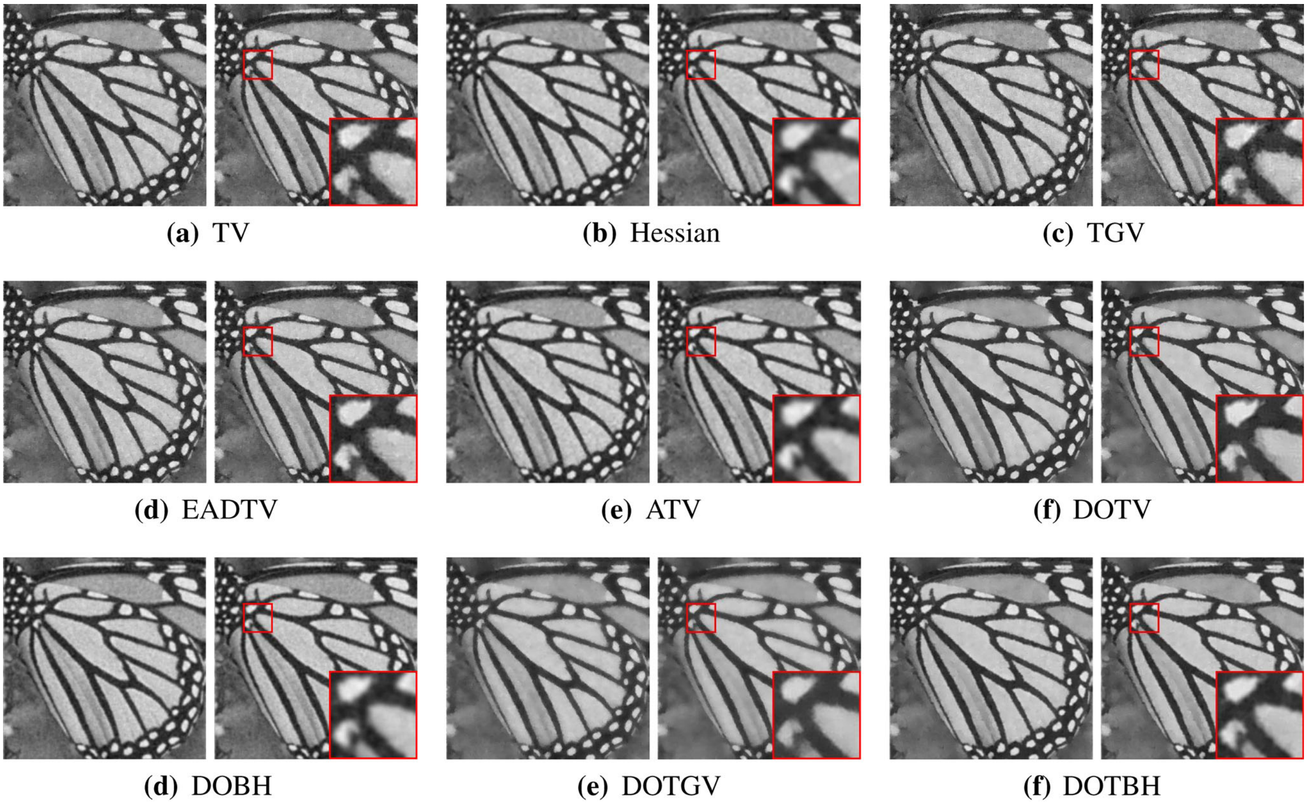


(a) TV **(b)** Hessian **(c)** TGV

(d) EADTV **(e)** ATV **(f)** DOTV

(d) DOBH **(e)** DOTGV **(f)** DOTBH

Fig. 5 The restored images and the close-up region of Lena with 10% Gaussian noise

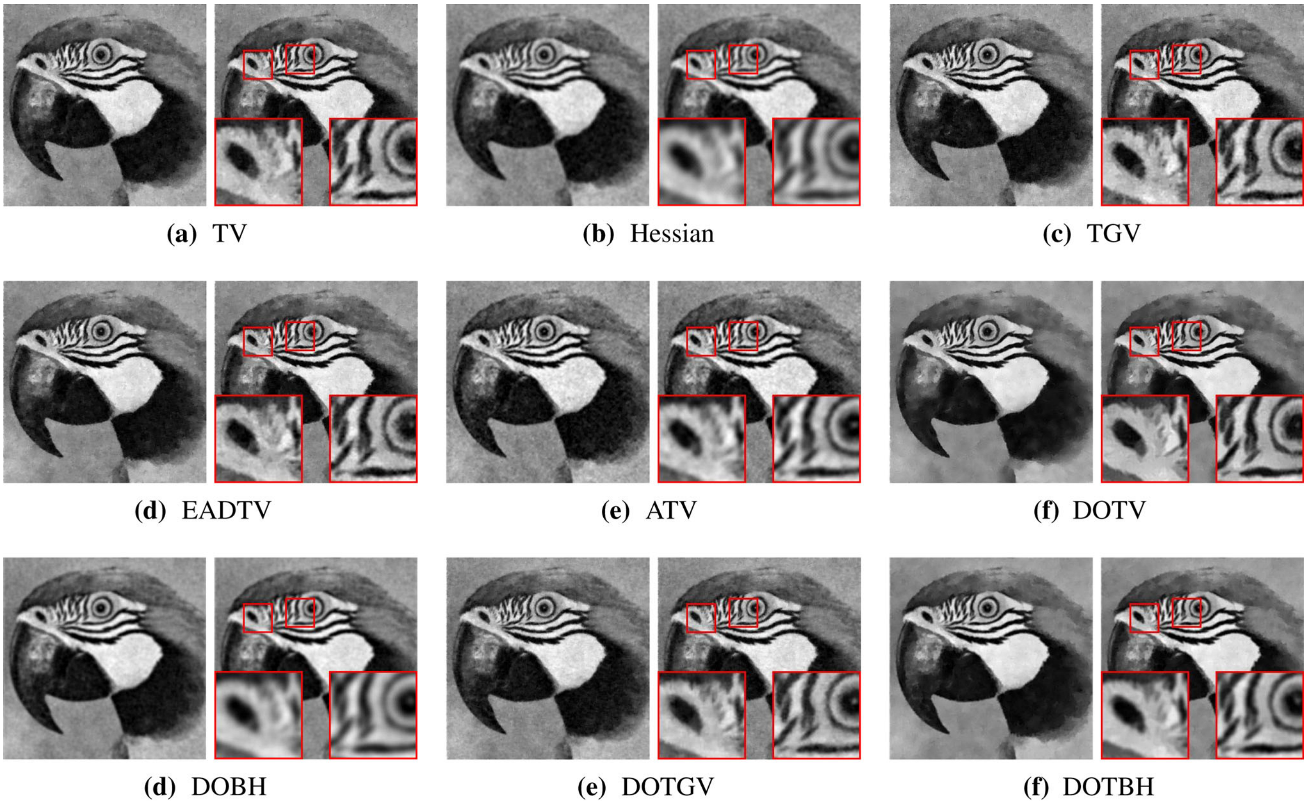


(a) TV **(b)** Hessian **(c)** TGV

(d) EADTV **(e)** ATV **(f)** DOTV

(d) DOBH **(e)** DOTGV **(f)** DOTBH

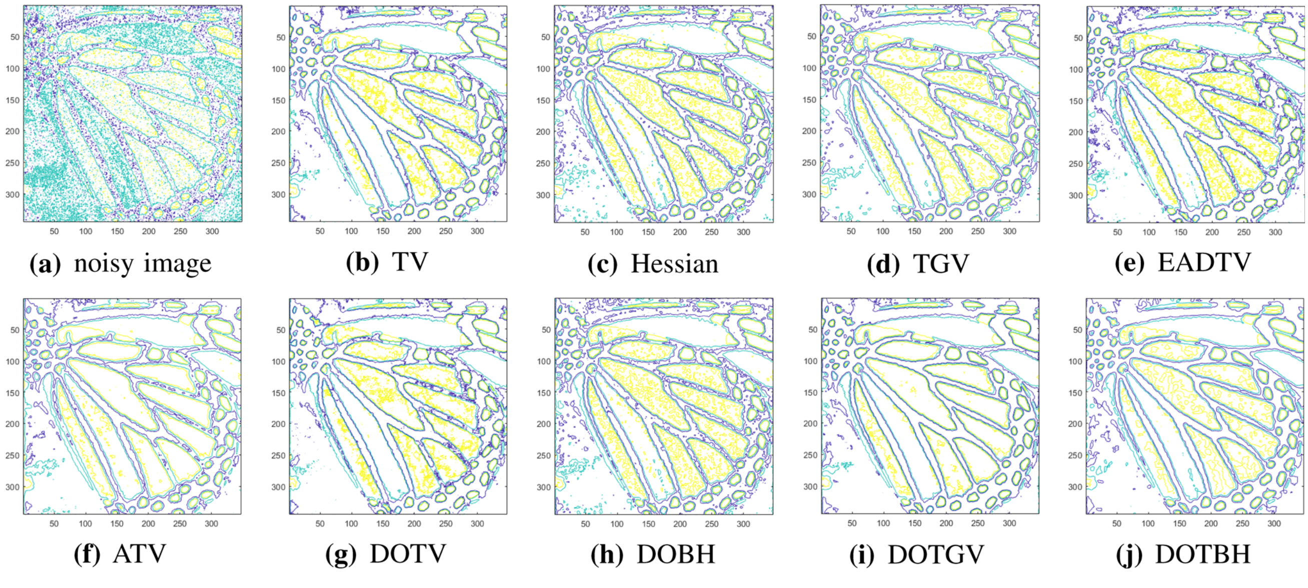
Fig. 6 The denoised images and close-up region of Butterfly with 15% Gaussian noise


(a) TV **(b)** Hessian **(c)** TGV

(d) EADTV **(e)** ATV **(f)** DOTV

(d) DOBH **(e)** DOTGV **(f)** DOTBH

Fig. 7 Denoised images and close-up regions of Parrot with 15% Gaussian noise



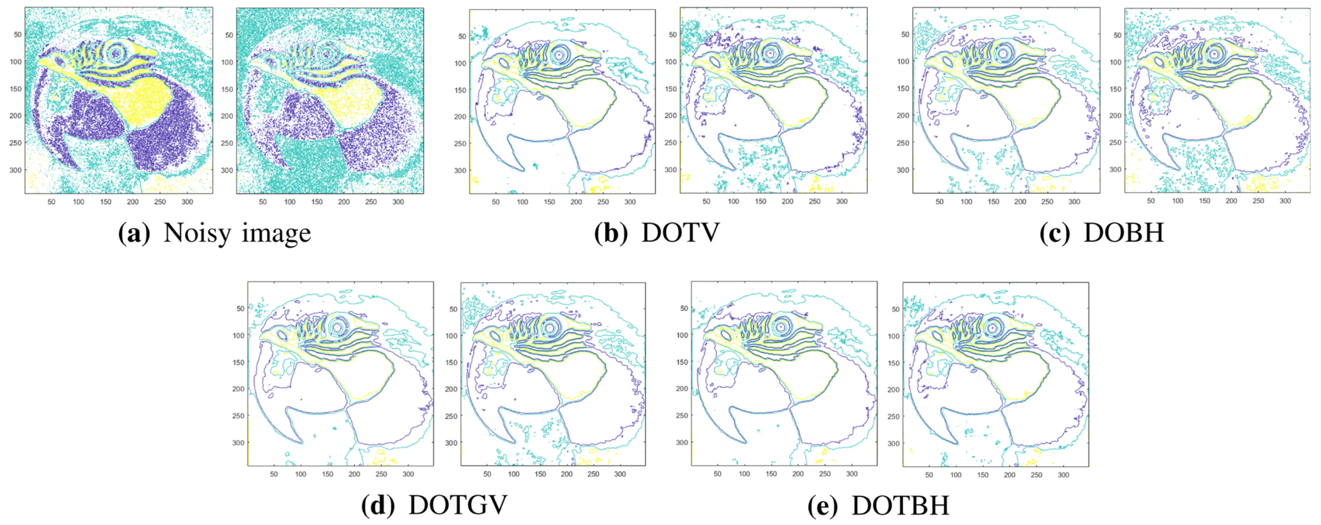
(a) noisy image **(b)** TV **(c)** Hessian **(d)** TGV **(e)** EADTV

(f) ATV **(g)** DOTV **(h)** DOBH **(i)** DOTGV **(j)** DOTBH

Fig. 8 Contour of denoised image for all the models of Butterfly with 15% Gaussian noise in Fig. 6

The piecewise constant image of geometry contains different geometric shapes. The noisy image with 10% Gaussian noise and denoised images that processed the models including TV, BH, TGV, EADTV, ATV and proposed models are listed in Fig. 2. Figure 3 presents residual images obtained by different models. The way we get residuals is $f - u + 100$. A column of signals of denoised images are expressed where the red line is located in Fig. 4(a). In Figure. 4(b)-(j), the blue slice is extracted from the original image and the red slice is extracted from the denoised image.

It can be seen from the experimental results that all methods can be used for denoising and restoration. In general,



(a) Noisy image **(b)** DOTV **(c)** DOBH

(d) DOTGV **(e)** DOTBH

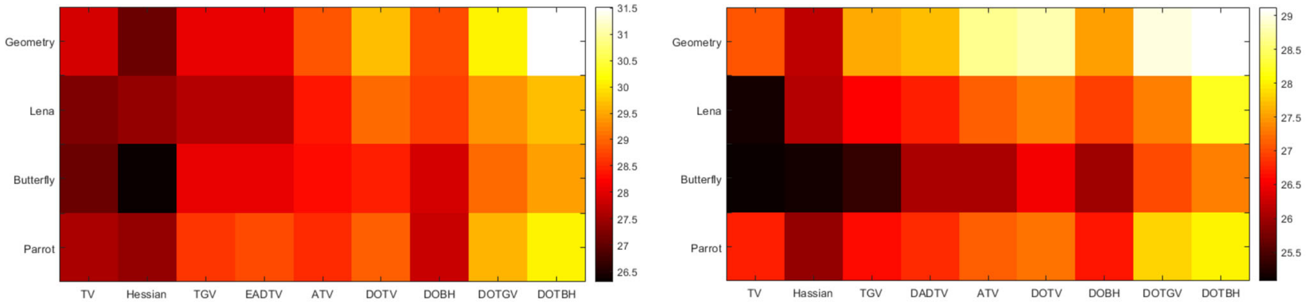
Fig. 9 Contour of denoised image for proposed models of Parrot with 10% and 15% Gaussian noise

Table 2 Comparison of some objective quantities of images with 10% Gaussian noise

Model	Geometry		Lena		Butterfly		Parrot	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
TV	27.9027	0.8405	27.2496	0.6832	27.1026	0.8846	27.5584	0.7927
Hessian	27.0467	0.8046	27.4272	0.7930	26.2998	0.8501	27.3852	0.7796
TGV	28.0342	0.8508	27.6229	0.7703	28.0209	0.8605	28.5907	0.7970
EADTV	28.0328	0.8885	27.6735	0.7942	28.0841	0.8693	28.7672	0.8312
ATV	28.8471	0.8930	28.3824	0.8074	28.3087	0.8864	28.5210	0.8564
DOTV	29.6827	0.8968	29.0300	0.9015	28.4893	0.8764	28.9277	0.8620
DOBH	28.8086	0.8596	28.6673	0.8227	27.9200	0.8552	27.8088	0.8065
DOTGV	30.0639	0.9038	29.3861	0.8961	29.0201	0.8862	29.6366	0.8998
DOTBH	31.5212	0.9286	29.7229	0.9082	29.4134	0.9096	30.0868	0.9037

Table 3 Comparison of some objective quantities of images with 15% Gaussian noise

Model	Geometry		Lena		Butterfly		Parrot	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
TV	27.0747	0.7544	25.1558	0.6221	25.0755	0.7024	26.7665	0.7252
Hessian	26.1976	0.7032	26.0945	0.7001	25.1601	0.7241	25.8986	0.6624
TGV	27.5523	0.8179	26.5565	0.7413	25.3638	0.7490	26.6039	0.7668
EADTV	27.6976	0.8329	26.7516	0.7767	26.0245	0.7527	26.8086	0.7971
ATV	28.6704	0.8630	27.1436	0.8039	26.0394	0.8096	27.1174	0.8025
DOTV	28.7496	0.8558	27.3157	0.8045	26.4973	0.8064	27.2606	0.8190
DOBH	27.4853	0.8273	26.9397	0.7802	26.0122	0.7865	26.6709	0.7735
DOTGV	28.9485	0.8654	27.3094	0.8153	27.0056	0.8288	27.8050	0.8524
DOTBH	29.1244	0.9036	28.2344	0.8776	27.3009	0.8599	28.0028	0.8605



(a) Hot map of values of PSNR with 10% Gaussian noise.

(b) Hot map of values of PSNR with 15% Gaussian noise.

Fig. 10 Hot map of values of PSNR

Table 4 Comparison of CPU time and total iterations of different models on ADMM and Fast ADMM with restart

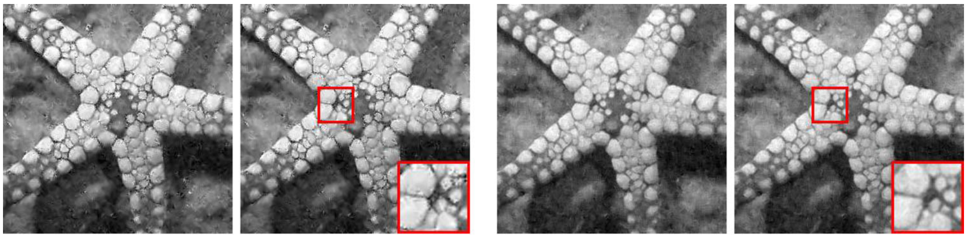
Model	ADMM		Fast ADMM with restart	
	CPU time(s)	Total iterations	CPU time(s)	Total iterations
DOTV	1.038061	35	0.780828	25
DOBH	0.677413	20	0.472239	10
DOTGV	0.843897	20	0.648775	15
DOTBH	1.032970	30	0.772573	15

anisotropic models can give a more pleasant visual result and better preserve edges than isotropic models. The contours produced by TV and TGV models are more realistic than BH model. As a better result, EADTV and DOTV models achieve similar results. ATV model is adept in suppressing big noise and keeping edges sharp especially for straight edges. For anisotropic higher-order models, it can be seen that edge preservation ability of DOBH model is better than BH model. But both of them are still not able to provided ideal performance. DOTBH and DOTGV models do the better than others in edge preservation. In addition, DOTBH model performs slightly better than DOTGV model in preserving the details while removing noise. It can be seen from the slice plots that all the models deal the denoised images with different degree of structure preservation. The denoised image curves of anisotropic models are closer to origin image. Figure 4 shows that DOTV and DOBH models have less geometry loss than TV and BH models, DOTBH and DOTGV models do the better than others in geometry preserving.

For the images which are not piecewise constant, five benchmark images are tested come from Set12 (Roth-Black2009) which are Lena, Butterfly, Parrot, Vegetables and Starfish. In those images showed in Fig. 1, Lena consists of smoothing and detailed regions, Butterfly and Vegetables include many approximated cartoon regions, Parrot and Starfish contain more texture components. Further, test images are corrupted by 10% and 15% Gaussian noise, respectively. Experiment results are compared with classical isotropic models and first-order anisotropic models first. In order to give a visual impression on the contrast between different models, results for these three images including Lena, Butterfly and Parrot are depicted, respectively. A part of denoised images and the close-up regions of them are shown in Figs. 5, 6 and 7. In order to show the edge preserve ability of different methods, contour of restored images for all the models of Butterfly corrupted by 15% Gaussian noise

Table 5 Comparison of PSNR of Starfish with 15% Gaussian noise

	NLTV PSNR	DOTV PSNR
Vegetables	26.4041	26.3572



(a) NLTV

(b) DOTV

Fig. 11 Denoised images and close-up region of Starfish with 15% Gaussian noise

is presented in Fig. 8. The contour of restored images for proposed models of parrot corrupted by 10% and 15% Gaussian noise is presented in Fig. 9. In order to show contour lines intuitively, three different colors are used to sign the values of different lines as yellow, cyan and purple, respectively. The values of the labeled colors are counted by Matlab toolbox for various images and methods. Table 2 and Table 3 contain two evaluation metrics for quantitative comparison of different methods on the clean image with 10% and 15% Gaussian noise added. Figure 10 presents hot maps of the value of PSNR with four images denoised in all experimental methods.

It can be seen that all of the methods can restore images in a different range of noise level. The anisotropic models such as DOTGV and DOTBH successfully preserve details while removing most noise with only scatter tiny noise spots left. BH model is weak in restoring sharp edges and smoothing flat areas, and DOBH model is superior to it. It is worth noting that the staircase effect of TV model derives false edges and this problem is avoided by anisotropic first-order models such as EADTV and DOTV models. According to Table 2 and Table 3, these methods can be quantitatively arranged in descending order as BH $<$ TV $<$ TGV $<$ DOBH $<$ EADTV $\approx$ ATV $\approx$ DOTV $<$ DOTGV $<$ DOTBH. For Lena with 15% Gaussian noise added, the performance of acceleration algorithm with process of convergence is shown in Table 4. It can be proved that acceleration method is effective for proposed models.

The directional information of directional Hessian is suited to shape preservation, while non-local operators can describe the characteristics of texture image features. Represented by the comparison of first-order regularizers, performance of directional Hessian can be evaluated by the results of DOTV and NLTV model [48]. The experimental results are presented in Fig. 11 and Table 5. It can be seen that two models are effective for texture preservation and obtain similar results after denoising process. The difference may prove that small textures in NLTV model handled slightly better, and DOTV model has a better edge denoising effect.

In the following comparison, DOTGV and DOTBH models which performed best in the previous experiments are compared with TDV and BM3D models. Only second-order directional regularizers are taken into consideration here so that the combination of first-order and second-order regularization in the experimental TDV model is adopted. Even a better result might be got in TDV model with a higher derivative order, but such efforts are expected to be too much computing ability consuming. Contours of denoised images for all the models of Vegetables and Starfish corrupted by 15% Gaussian noise are presented in Fig. 12. Table 6 shows the PSNR of the results. It can be concluded that all four models successfully completed the task of image restoration. DOTBH model is a competitor of the non-local denoising approach of BM3D model. For partial texture regions, BM3D may deal better with them. The results obtained by DOTGV and TDV models are similar.

Finally, the performance of proposed anisotropic variation models is evaluated on restoration of color textured images. The edge may be blurred due to the inconsistent diffusion intensity at the edge of each layer in color images. The difficulty in maintaining image recovery quality lies in description of texture components and coupling between layers in different colors. Non-local multi-channel total variation (M-NLTV) model [48] performs well in texture image restoration. Excellent coupling effects are achieved between layers based on the extension of multi-channel total variation (MTV) regularizer. Let $u(x) = (u^{(1)}(x); \ldots; u^{(m)}(x)) : \Omega \to \mathbb{R}^m$ be an m-channel image. It is natural to generalize (4.1) to multichannel images as follows:

$$MTV(u) = \int_{\Omega} \sqrt{\left\|\nabla u^{(1)}\right\|^2 + \cdots + \left\|\nabla u^{(m0)}\right\|^2} dx. \tag{4.1}$$

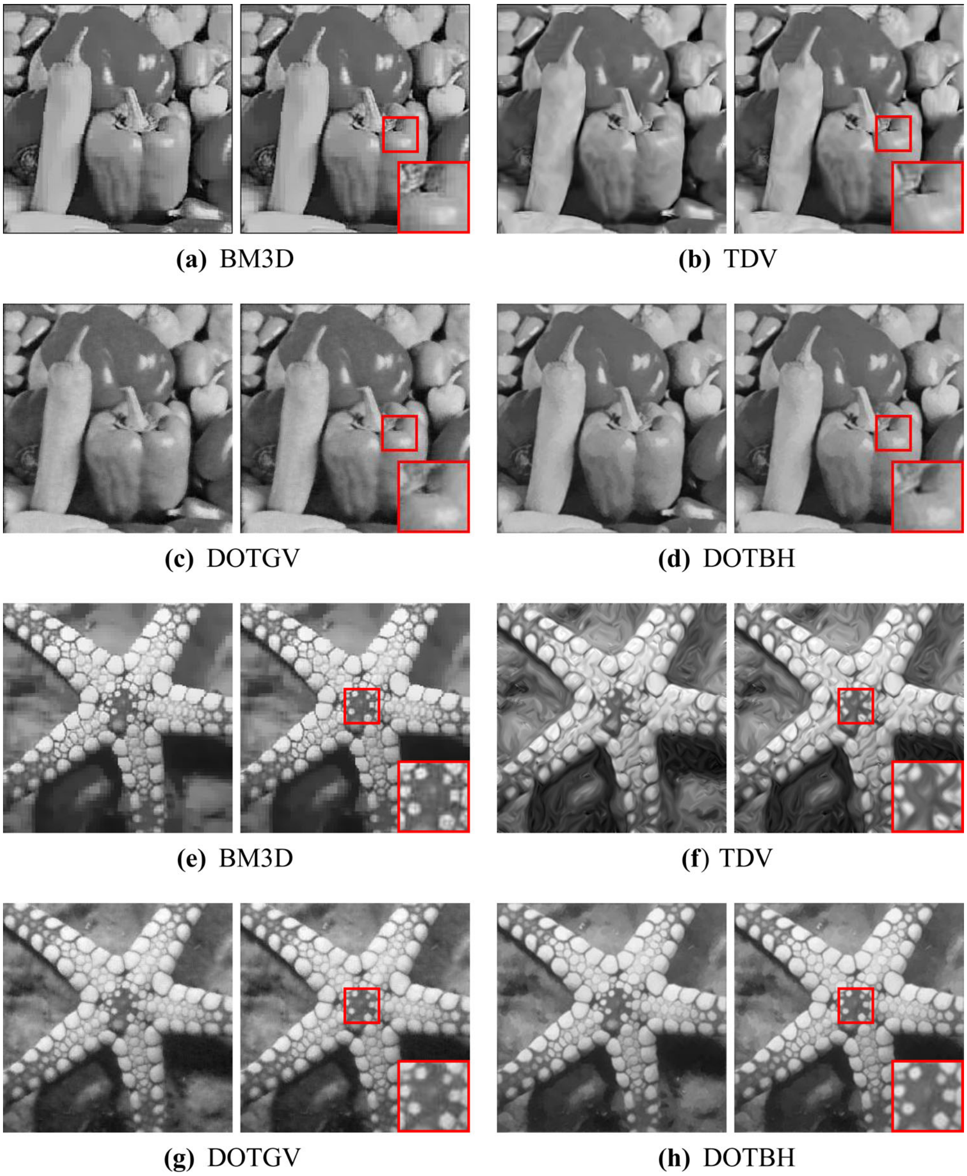
Therefore, same coupling scheme is adopted to design multi-channel denoising models based on proposed models mentioned above called M-DOTV model, M-DOBH model, M-DOTGV model, M-DOTBH model.

Five standard testing images in Berkeley segmentation database (BSDS500) presented in Fig. 13 are used to evaluate the proposed models. In order to compare the texture, edge and smoothness preserving, Fig. 14 shows denoised images of Zebra and close-up regions with 15% Gaussian noise. Table 7 gives PSNR values for the results of images denoising in Fig. 13.

As the denoised results shown in Fig. 14, there are no obvious blurred edges between layers as all models adopt the same coupling design. Due to the complex texture features of image local parts, some image details processed by TV and TGV models are missing. For first-order anisotropic models, M-NLTV model has a slight advantage in feature preservation of local small texture regions, while DOTV model can achieve approximate results in local large-scale texture image feature preservation. By comparing all models, M-DOTGV and M-DOTBH models achieve best image restorational qualities as the texture feature can be better preserved. Especially, it can be obvious displayed in large texture regions. Table 7 further proves that proposed models achieve perfect performance for the directional and textural color images in large database.

5 Conclusion

Based on the consideration of shape preservation, three high-order anisotropic variational models are introduced. The applications of the models in image denoising are proved. Discretization processes based on the discrete finite difference scheme and numerical realization of fast ADMM with restart algorithm to solve the equations are given. In the last

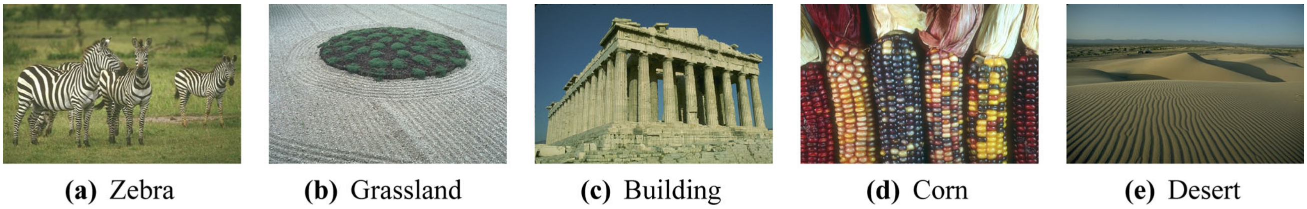


(a) BM3D (b) TDV (c) DOTGV (d) DOTBH (e) BM3D (f) TDV (g) DOTGV (h) DOTBH

Fig. 12 Denoised images and close-up region of Vegetables and Starfish with 15% Gaussian noise

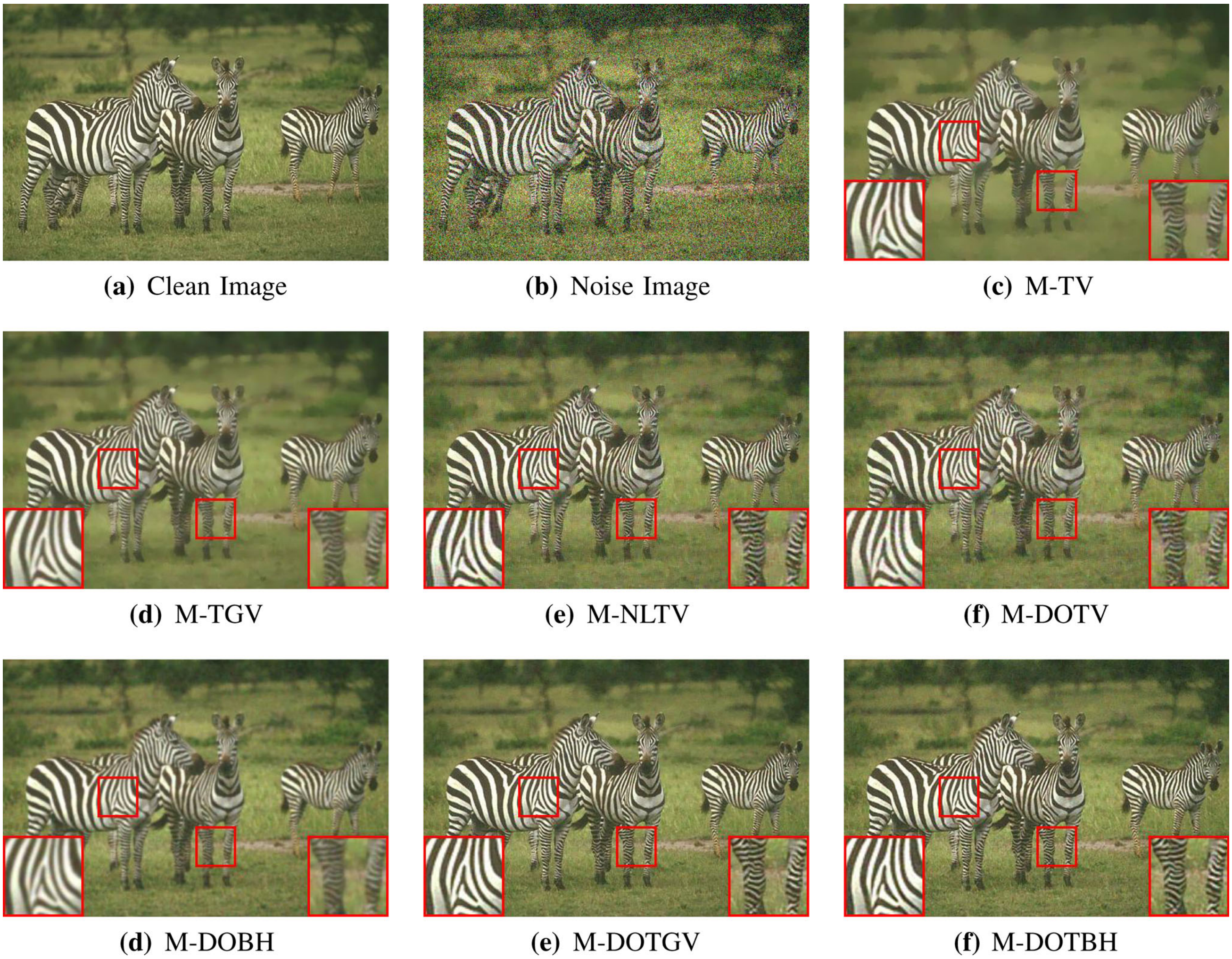
Table 6 Comparison of PSNR of Vegetables and Starfish with 15% Gaussian noise

	BM3D	TDV	DOTGB	DOTBG
	PSNR	PSNR	PSNR	PSNR
Vegetables	27.1089	26.9023	26.9146	27.1015
Starfish	27.5467	27.4177	27.2171	27.4275



(a) Zebra (b) Grassland (c) Building (d) Corn (e) Desert

Fig. 13 The ground truth images from Berkeley segmentation database



(a) Clean Image **(b)** Noise Image **(c)** M-TV

(d) M-TGV **(e)** M-NLTV **(f)** M-DOTV

(d) M-DOBH **(e)** M-DOTGV **(f)** M-DOTBH

Fig. 14 The denoised images and close-up region of Zebra with 15% Gaussian noise

Table 7 Comparison of PSNR of images in Fig. 13 and Fig. 14 with 15% Gaussian noise

	M-TV PSNR	M-TGV PSNR	M-NLTV PSNR	M-DOTV PSNR	M-DOBH PSNR	M-DOTGB PSNR	M-DOTBG PSNR
Zebra	25.7701	26.4791	27.1089	27.0087	26.0701	27.5109	27.8305
Grassland	25.1848	26.8826	27.1077	26.7980	25.7088	27.2023	27.3002
Building	26.2829	26.8929	27.4512	27.2419	26.6096	27.6378	27.9177
Corn	25.7992	26.6679	27.1321	27.0131	25.9986	27.4996	27.6867
Desert	25.5088	26.4623	27.3309	27.3990	26.4280	27.9089	28.1061

section, through the comparative experiments, characteristics of proposed models are summarized as follows.

The directional information of directional Hessian involved in the restorational process is particularly suitable to shape preservation in the presence of heavy noise. Compared with other direction estimational methods, double-orientational neighborhoods in bivariate based on structural tensor are appropriated to oriented patterns which occlude to each other. The relative orientation of connected nodes changes less appearance and other characteristics. It may be easier to mark the edges in terms of its orientations. A well-orientational estimation approach has been applied in proposed anisotropic regularizers which makes directionality particularly useful. The proposed models which benefit from the orientational estimation are adept at processing large-scale texture image restoration. In addition, they possess certain capabilities for processing small damaged areas.

TGV model has the best denoising effect for the processed image of isotropic models mentioned before. The result shows the proposed first-order anisotropic model DOTV can achieve similar image recovery effect and better edges preservation. High-order anisotropic models such as DOTGV and DOTBH can improve the quality of the resulting image to get a smoother boundary. DOTBH model performs a little better than DOTGV model in geometry preserving. The combination of first-order and second-order regularization of TDV model has the same effect with DOTGV model. However, they rely on massive number of parameters which is far more than expected. In light of this, it is believed that BM3D has more advantages in terms of parameters and handles most cases well.

The proposed models can achieve better performance on texture images. Texture preservation performance of directional Hessian can be evaluated by the compare results of DOTV and NLTV models which applied first-order directional regularizer. In addition, the proposed models still perform well in color images of large dataset such as Berkeley segmentation database. The coupling design effectively avoids the blurring between the color layers. Texture feature can be better preserved by directional estimation; especially, it can be displayed in large-scale texture regions.

In future work, estimation for double orientations from directional Hessian in signals of arbitrary signal dimensionality is intended to study and the proposed methods are extended to deal with optical flow, registration and other challenges.

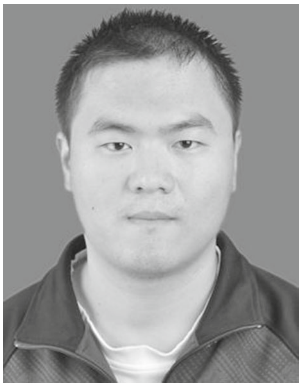
References

1. Scherzer, Otmar: Handbook of Mathematical Methods in Imaging. Springer Verlag New York (2015)
2. Aubert, G., Kornprobst, P.: Mathematical Problems in Image Processing. Springer, New York (2002)
3. Vogel Curtis, R.: Computational Methods for Inverse Problems. Tsinghua University Press, Beijing, China (2011)
4. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D Nonlinear Phenomena. **60**(1–4), 259–268 (1992)
5. You, Y.L., Kaveh, M.: Fourth-order partial differential equations for noise removal. IEEE Trans. Image Proc. A Publ. IEEE Signal Proc. Soc. **9**(10), 1723–1730 (2000)
6. Hinterberger, W., Scherzer, O.: Variational methods on the space of functions of bounded Hessian for convexification and denoising. Computing. **76**(1), 109–133 (2006)
7. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. Siam J. Imaging Sci. **3**(3), 492–526 (2010)
8. Zhu, W., Tai, X.C., Chan, T.: Image segmentation using Euler's Elastica as the regularization. J. Sci. Comput. **57**(2), 414–438 (2013)
9. Witkin, A.P.: Scale space filtering. Read. Comput. Vis. **42**(3), 329–332 (1987)
10. Perona, P., Malik, J., Intelligence, M.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. Pattern Anal. **12**(7), 629–639 (2002)
11. Weickert, J.: Anisotropic Diffusion In Image Processing. Teubner Stuttgart (1998)
12. Brox, T., Weickert, J., Burgeth, B., Mrzek, P.: Nonlinear structure tensors. Image Vis. Comput. **24**(1), 41–55 (2004)
13. Grasmair, M., Lenzen, F.: Anisotropic total variation filtering. Appl. Math. Optim. **62**(3), 323–339 (2010)
14. Bayram, I., Kamasak, M.E.: A Directional Total Variation. In: Signal Processing Conference (2012)
15. Zhang, H., Wang, Y.: Edge adaptive directional total variation. J. Eng. **1**(1), 1–2 (2013)
16. Lefkimmiatis, S., Roussos, A., Maragos, P., Unser, M.: Structure tensor total variation. Siam J. Imag. Sci. **8**(2), 1090–1122 (2015)
17. Kongskov, R.D., Dong, Y., Knudsen, K.: Directional total generalized variation regularization. BIT Numer. Math. **59**, 903–928 (2019)
18. Parisotto, S., Masnou, S., Schonlieb, C.B.: Higher-order total directional variation: analysis. SIAM J. Imag. Sci. **13**(1), 474–496 (2020)
19. Goldstein, T., Osher, S.: The split Bregman method for L1-regularized problems. SIAM J. Imag. Sci. **2**, 323–343 (2009)
20. Marius, L., Arvid, L., Tai, X.C.: Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. IEEE Trans. Image Proc. **12**(12), 1579–1590 (2003)
21. Ying, W., Sun, J.B., Guo, Z.C.: A New Anisotropic Fourth-Order Diffusion Equation Model Based on Image Feature for Image Denoising. UCLA Computational and Applied Mathematics Report, August 2020, 20-32 (2020)
22. Pang, Z.F., Zhou, Y.M., Wu, T., Li, D.J.: Image denoising via A new anisotropic total variation based model. Signal Proc.: Image Commun. **13**, 140–152 (2019)
23. Abderrahim, Elmoataz: Olivier, Lezoray, Sbastien, Bougleux: nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. IEEE Trans. Image Proc. **17**(7), 1047–1060 (2008)
24. Chierchia, G., Pustelnik, N., Pesquet-Popescu, B., Pesquet, J.C.: A non-local structure tensor based approach for multicomponent image recovery problems. IEEE Trans. Image Proc. **23**(12), 5531–5544 (2014)
25. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Proc. **16**(8), 2080–2095 (2007)
26. Fu, H., Liu, W., Chen, H., Wang, Z.: An anisotropic gaussian filtering model for image De-Hazing. IEEE Access. **8**, 175140–175149 (2020)
27. Raad, L., Oliver, M., Ballester, C., Haro, G., Meinhardt, E.: On anisotropic optical flow inpainting algorithms. Image Proc. Line. **10**, 78–104 (2020)
28. Wang, G., Baets, B.D.: Superpixel segmentation based on anisotropic edge strength. J. Imag. **5**(6), 57 (2019)
29. Allain, P., Guillo, L., Guillemot, C.: Light Field Denoising Using 4D Anisotropic Diffusion. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 1692-1696 (2019)
30. Wei, W., Fang, L., Michael, K.N.: Structural similarity based non-local variational models for image restoration. IEEE Trans. Image Proc. **28**(9), 4260–4272 (2019)
31. Wei, W., Michael, K.: A nonlocal total variation model for image decomposition: illumination and reflectance. Numer. Math.: Theory, Methods Appl. **7**, 334–355 (2014)
32. Pang, Z.F., Zhang, H.L., Luo, S., Zeng, T.: Image denoising based on the adaptive weighted TV^P regularization. Signal Proc. **167**, 1–21 (2020)
33. Li, Z., Malgouyres, F., Zeng, T.: Regularized non-local total variation and application in image restoration. J. Math. Imag. Vis. **59**(2), 296–317 (2017)
34. Jia, F., Wong, W.H., Zeng, T.: DDUNet: Dense Dense U-Net with Applications in Image Denoising. In Proceedings of the IEEE/CVF

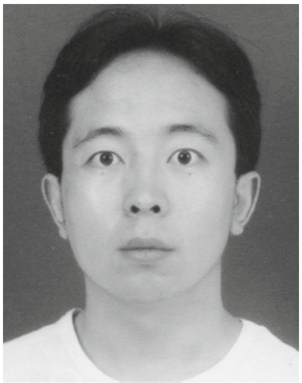
International Conference on Computer Vision. October 11-17, 354-364 (2021)

35. Fang, F., Li, J., Yuan, Y., Zeng, T., Zhang, G.: Multilevel edge features guided network for image denoising. IEEE Trans. Neural Netw. Learn. Syst. **32**(9), 3956–3970 (2020)
36. Tschumperl, D.: Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. Int. J. Comput. Vis. **68**(1), 65–82 (2006)
37. Chambolle, A., Pock, T.: Vision: a first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imag. **40**(1), 120–145 (2011)
38. Tschumperl, D., Deriche, R.: Vector-valued image regularization with PDEs: a common framework for different applications. IEEE Trans. Pattern Anal. Mach. Intell. **27**(4), 506–517 (2005)
39. Yin, X.H., Chen, S.L., Wang, L.P., Zhou, B.: Fractional-order difference curvature-driven fractional anisotropic diffusion equation for image super-resolution. Int. J. Mod., Simul., Sci. Comput. **10**(1), 1–13 (2019)
40. Mhlich, M., Aach, T.: A Theory of Multiple Orientation Estimation. Computer Vision - ECCV 2006, 9th European Conference on Computer Vision. 3952, 69-82 (2006)
41. Shizawa, M., Mase, K.: Simultaneous Multiple Optical Flow Estimation. In: 10th International Conference on Pattern Recognition. 1, 274-278 (1990)
42. Mota, C., Stuke, I., Aach, T., Barth, E.: Estimation of Multiple Orientations at Corners and Junctions. In, Berlin, Heidelberg 2004. Pattern Recognition. 3175, 163-170 (2004)
43. Steidl, G., Teuber, T.: Anisotropic Smoothing Using Double Orientations. In: Scale Space and Variational Methods in Computer Vision, Second International Conference, SSVM 2009. 5567, 477-489 (2009)
44. Aach, T., Mota, C., Stuke, I., Muhlich, M., Barth, E.: Analysis of superimposed oriented patterns. IEEE Trans. Image Proc. A Publ. IEEE Signal Proc. Soc. **15**(12), 3690–3700 (2006)
45. Beck, A., Teboulle, M.: Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Trans. Image Proc. **18**(11), 2419–2434 (2009)
46. Bioucas-Dias, J.M., Figueiredo, M.: A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration. IEEE Trans. Image Proc. **16**(12), 2992–3004 (2008)
47. Goldstein, T., O'Donoghue, B., Setzer, S., Baraniuk, R.: Fast alternating direction optimization methods. Siam J. Imag. Sci. **7**(3), 1588–1623 (2014)
48. Duan, J.M., Pan, Z.K., Tai, X.C.: Non-local TV models for restoration of color texture images. J. Image Gr. **18**(7), 753–760 (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Guangyu Yang was born in 1992. He is a PhD student of College of Computer Science and Technology, Qingdao University, Qingdao, China. His current research interests include variational image restoration and computer vision problems. Email: lyyangguangyu @126.com.



Weibo Wei was born in 1981. He received his PhD from Nanjing University of Science and Technology, Nanjing, China in 2006. He is an associate professor with College of Computer Science and Technology, Qingdao University, Qingdao, China. His current research interests include image processing, computer vision, target recognition and tracking. Email: njustwwb@163.com.

Zhenkuan Pan was born in 1966. He received his PhD in Engineering Mechanics from Shanghai Jiao Tong University, Shanghai, China in 1992. He is a Professor with College of Computer Science and Technology, Qingdao University, Qingdao, China. His current research interests include variational image processing, optimization algorithms computer vision and patternrecognition.