Introducing anisotropic tensor to high order variational model for image restoration

Jinming Duan[a], Wil O.C. Ward[a], Luke Sibbett[a], Zhenkuan Pan[b], Li Bai[a]

[a] *School of Computer Science, University of Nottingham, UK*
[b] *School of Computer Science and Technology, Qingdao University, China*

A B S T R A C T

Second order total variation (SOTV) models have advantages for image restoration over their first order counterparts including their ability to remove the staircase artefact in the restored image. However, such models tend to blur the reconstructed image when discretised for numerical solution [1–5]. To overcome this drawback, we introduce a new tensor weighted second order (TWSO) model for image restoration. Specifically, we develop a novel regulariser for the SOTV model that uses the Frobenius norm of the product of the isotropic SOTV Hessian matrix and an anisotropic tensor. We then adapt the alternating direction method of multipliers (ADMM) to solve the proposed model by breaking down the original problem into several subproblems. All the subproblems have closed-forms and can be solved efficiently. The proposed method is compared with state-of-the-art approaches such as tensor-based anisotropic diffusion, total generalised variation, and Euler's elastica. We validate the proposed TWSO model using extensive experimental results on a large number of images from the Berkeley BSDS500. We also demonstrate that our method effectively reduces both the staircase and blurring effects and outperforms existing approaches for image inpainting and denoising applications.

© 2017 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Anisotropic diffusion tensor can be used to describe the local geometry at an image pixel, thus making it appealing for various image processing tasks [6–13]. Variational methods allow easy integration of constraints and use of powerful modern optimisation techniques such as primal–dual [14–16], fast iterative shrinkage-thresholding algorithm [17,18], and alternating direction method of multipliers [2–4,19–24]. Recent advances on how to automatically select parameters for different optimisation algorithms [16,18, 25] dramatically boost performance of variational methods, leading to increased research interest in this field.

As such, the combination of diffusion tensor and variational methods has been investigated by researchers for image processing. Krajsek and Scharr [7] developed a linear anisotropic regularisation term that forms the basis of a tensor-valued energy functional for image denoising. Grasmair and Lenzen [8,9] penalised image variation by introducing a diffusion tensor that depends on the structure tensor of the image. Roussous and Maragos [10] developed a functional that utilises only eigenvalues of the structure tensor. Similar work by Lefkimmiatis et al. [13] used Schatten-

norm of the structure tensor eigenvalues. Freddie et al. proposed a tensor-based variational formulation for colour image denoising [11], which computes the structure tensor without Gaussian convolution to allow Euler-equation of the functional to be elegantly derived. They further introduced a tensor-based functional named the gradient energy total variation [12] that utilises both eigenvalues and eigenvectors of the gradient energy tensor.

However, these existing works mentioned above only consider the standard first order total variation (FOTV) energy. A drawback of the FOTV model is that it favours piecewise-constant solutions. Thus, it can create strong staircase artefacts in the smooth regions of the restored image. Another drawback of the FOTV model is its use of gradient magnitude to penalise image variations at pixel locations **x**, which uses less neighbouring pixel information as compared to high order derivatives in a discrete space. As such, the FOTV has difficulties inpainting images with large gaps. High order variational models thus can be applied to remedy these side effects. Among these is the second order total variation (SOTV) model [1,2,22,26,27]. Unlike the high order variational models, such as the Gaussian curvature [28], mean curvature [23, 29], Euler's elastica [21] etc., the SOTV is a convex high order extension of the FOTV, which guarantees a global solution. The SOTV is also more efficient to implement [4] than the convex total generalised variation (TGV) [30,31]. However, the inpainting results of

the model highly depend on the geometry of the inpainting region, and it also tends to blur the inpainted area [2,22]. Further, according to the numerical experimental results displayed in [1–5] the SOTV model tends to blur object edges for image denoising.

In this paper, we propose a tensor weighted second order (TWSO) variational model for image inpainting and denoising. A novel regulariser has been developed for the TWSO that uses the Frobenius norm of the product of the isotropic SOTV Hessian matrix and an anisotropic tensor, so that the TWSO is a nonlinear anisotropic high order model which effectively integrates orientation information. In numerous experiments, we found that the TWSO can reduce the staircase effect whilst improve the sharp edges/boundaries of objects in the denoised image. For image inpainting problems, the TWSO model is able to connect large gaps regardless of the geometry of the inpainting region and it would not introduce much blur to the inpainted image. As the proposed TWSO model is based on the variational framework, ADMM can be adapted to solve the model efficiently. Extensive numerical results show that the new TWSO model outperforms the state-of-the-art approaches for both image inpainting and denoising.

The contributions of the paper are twofold: 1) a novel anisotropic nonlinear second order variational model is proposed for image restoration. To the best of our knowledge, this is the first time the Frobenius norm of the product of the Hessian matrix and a tensor has been used as a regulariser for variational image denoising and inpainting; 2) A fast ADMM algorithm is developed for image restoration based on a forward-backward finite difference scheme.

The rest of the paper is organised as follows: Section 2 introduces the proposed TWSO model and the anisotropic tensor $\mathbf{T}$; Section 3 presents the discretisation of the differential operators used for ADMM based on a forward-backward finite difference scheme; Section 4 describes ADMM for solving the variational model efficiently; Section 5 gives details of the experiments using the proposed TWSO model and the state-of-the-art approaches for image inpainting and denoising. Section 6 concludes the paper.

2. The TWSO model for image restoration

2.1. The TWSO model

In [26], the authors considered the following SOTV model for image processing

$$\min_u \left\{ \frac{\eta}{2} \|u - f\|_2^2 + \|\nabla^2 u\|_1 \right\}, \tag{2.1}$$

where $\eta > 0$ is a regularisation parameter. $\nabla^2 u$ is the second order Hessian matrix of the form

$$\nabla^2 u = \begin{pmatrix} \partial_x \partial_x u & \partial_y \partial_x u \\ \partial_x \partial_y u & \partial_y \partial_y u \end{pmatrix}, \tag{2.2}$$

and $\|\nabla^2 u\|_1$ in (2.1) is the Frobenius norm of the Hessian matrix (2.2). By using such norm, (2.1) has several capabilities: 1) allowing discontinuity of gradients of u; 2) imposing smoothness on u; 3) satisfying the rotation-invariant property. Thought this high order model (2.1) is able to reduce the staircase artefact associated with the FOTV for image denoising, it can blur object edges in the image [1–4]. For inpainting, as investigated in [2,22], though the SOTV has the ability to connect large gaps in the image, such ability depends on the geometry of the inpainting region, and it can blur the inpainted image. In order to remedy these side effects in both image inpainting and denoising, we propose a more flexible and generalised variational model, i.e., the tensor weighted second order (TWSO) model that takes advantages of both the tensor and the second order derivative. Specifically, the TWSO model is defined as

$$\min_u \left\{ \frac{\eta}{p} \|1_\Gamma (u - f)\|_p^p + \|\mathbf{T}\nabla^2 u\|_1 \right\}, \tag{2.3}$$

where $p \in \{1, 2\}$ denotes the L^1 and L^2 data fidelity terms (i.e. $\|1_\Gamma (u - f)\|_1$ and $\|1_\Gamma (u - f)\|_2^2$), and Γ is a subset of Ω (i.e. $\Gamma \subset \Omega \subset R^2$). For image processing applications, the Ω is normally a rectangle domain. For image inpainting, f is the given image in $\Gamma = \Omega \backslash D$, where $D \subset \Omega$ is the inpainting region with missing or degraded information. The values of f on the boundaries of D need to be propagated into the inpainting region via minimisation of the weighted regularisation term of the TWSO model. For image denoising, f is the noisy image and $\Omega = \Gamma$. In this case, the choice of p depends on the type of noise found in f, e.g. $p = 2$ for Gaussian noise while $p = 1$ for impulsive noise.

In the regularisation term $\|\mathbf{T}\nabla^2 u\|_1$ of (2.3), $\mathbf{T}$ is a symmetric positive semi-definite 2×2 diffusion tensor whose four components are $\mathbf{T}_{11}$, $\mathbf{T}_{12}$, $\mathbf{T}_{21}$, and $\mathbf{T}_{22}$. They are computed from the input image f. It is worth pointing out that the regularisation term $\|\mathbf{T}\nabla^2 u\|_1$ is the Frobenius norm of a 2 by 2 tensor $\mathbf{T}$ multiplied by a 2 by 2 Hessian matrix $\nabla^2 u$, which has the form of

$$\begin{pmatrix} \mathbf{T}_{11}\partial_x\partial_x u + \mathbf{T}_{12}\partial_x\partial_y u & \mathbf{T}_{11}\partial_y\partial_x u + \mathbf{T}_{12}\partial_y\partial_y u \\ \mathbf{T}_{21}\partial_x\partial_x u + \mathbf{T}_{22}\partial_x\partial_y u & \mathbf{T}_{21}\partial_y\partial_x u + \mathbf{T}_{22}\partial_y\partial_y u \end{pmatrix}, \tag{2.4}$$

where the two orthogonal eigenvectors of $\mathbf{T}$ span the rotated coordinate system in which the gradient of the input image is computed. As such $\mathbf{T}$ can introduce orientations to the bounded Hessian regulariser $\|\nabla^2 u\|_1$. The eigenvalues of the tensor measure the degree of anisotropy in the regulariser and weight the four second order derivatives in $\nabla^2 u$ in the two directions given by the eigenvectors of the structure tensor introduced in [32]. It should be noting that the original SOTV (2.1) is an isotropic nonlinear model while the proposed TWSO (2.3) is an anisotropic nonlinear one. As a result, the new tensor weighted regulariser $\|\mathbf{T}\nabla^2 u\|_1$ is powerful for image inpainting and denoising, as illustrated in the experimental section. In the next section, we shall introduce the derivation of the tensor $\mathbf{T}$.

2.2. Tensor estimation

In [32], the author defined the structure tensor J_ρ of an image u

$$J_\rho (\nabla u_\sigma) = K_\rho * (\nabla u_\sigma \otimes \nabla u_\sigma), \tag{2.5}$$

where K_ρ is a Gaussian kernel whose standard deviation is ρ and ∇u_σ is the smoothed version of the gradient convolved by K_σ. The use of $(\nabla u_\sigma \otimes \nabla u_\sigma) := \nabla u_\sigma \nabla u_\sigma^T$ as a structure descriptor is to make J_ρ insensitive to noise but sensitive to change in orientation. The structure tensor J_ρ is positive semi-definite and has two orthonormal eigenvectors $v_1 \parallel \nabla u_\sigma$ (in the direction of gradient) and $v_2 \parallel \nabla u_\sigma$ (in the direction of the isolevel lines). The corresponding eigenvalues μ_1 and μ_2 can be calculated from

$$\mu_{1,2} = \frac{1}{2} \left(j_{11} + j_{22} \pm \sqrt{(j_{11} - j_{22})^2 + 4j_{12}^2} \right), \tag{2.6}$$

where j_{11}, j_{12} and j_{22} are the components of J_ρ. They are given as

$$j_{11} = K_\rho * (\partial_x u_\sigma)^2, \; j_{12} = j_{21} = K_\rho * (\partial_x u_\sigma \, \partial_y u_\sigma), \; j_{22}$$
$$= K_\rho * (\partial_y u_\sigma)^2. \tag{2.7}$$

The eigenvalues of J_ρ describe the ρ-averaged contrast in the eigendirections, meaning: if $\mu_1 = \mu_2 = 0$, the image is in homogeneous area; if $\mu_1 \gg \mu_2 = 0$, it is on a straight line; and if $\mu_1 > \mu_2 \gg 0$, it is at objects' corner. Based on the eigenvalues, we can define the following local structural coherence quantity

$$Coh = (\mu_1 - \mu_2)^2 = (j_{11} - j_{22})^2 + 4j_{12}^2. \tag{2.8}$$

This quantity is large for linear structures and small for homogeneous areas in the image. With the derived structure tensor (2.5) we define a new tensor $\mathbf{T} = \mathcal{D}\big(J_\rho(\nabla u_\sigma)\big)$ whose eigenvectors are parallel to the ones of $J_\rho(\nabla u_\sigma)$ and its eigenvalues λ_1 and λ_2 are chosen depending on different image processing applications. For denoising problems, we need to prohibit the diffusion across image edges and encourage strong diffusion along edges. We therefore consider the following two diffusion coefficients for the eigenvalues

$$\lambda_1 = \begin{cases} 1 & s \le 0 \\ 1 - e^{\frac{-3.31488}{(s/C)^8}} & s > 0 \end{cases}, \quad \lambda_2 = 1, \tag{2.9}$$

with $s := |\nabla u_\sigma|$ the gradient magnitude and C the contrast parameter. For inpainting problems, we want to preserve linear structures and thus regularisation along isophotes of the image is appropriate. We therefore consider the weights that Weickert [32] used for enhancing the coherence of linear structures. With μ_1, μ_2 being the eigenvalues of J_ρ as before, we define

$$\lambda_1 = \gamma, \quad \lambda_2 = \begin{cases} \gamma & \mu_1 = \mu_2 \\ \gamma + (1-\gamma)\, e^{-\frac{\mathcal{C}}{Coh}} & \text{else} \end{cases}, \tag{2.10}$$

where $\gamma \in (0,1)$, $\gamma \ll 1$. The constant γ determines how steep the exponential function is. The structure threshold $\mathcal{C}$ affects how the approach interprets local structures. The larger the parameter value is, the more coherent the method will be. With these eigenvalues, the regularisation is stronger in the neighbourhood of coherent structures (note that ρ determines the radius of neighbourhood) and weaker in homogeneous areas, at corners, and in incoherent areas of the image.

3. Discretisation of differential operators

In order to implement ADMM for the proposed TWSO model, it is necessary to discretise the derivatives involved. We note that different discretisation may lead to different numerical experimental results. In this paper, we use the forward–backward finite difference scheme. Let Ω denote the two dimensional grey scale image of size MN, and x and y denote the coordinates along image column and row directions respectively. The discrete second order derivatives of u at point (i,j) along x and y directions can be then written as

$$\partial_x^+ \partial_x^- u_{i,j} = \partial_x^- \partial_x^+ u_{i,j} =$$
$$\begin{cases} u_{i,N} - 2u_{i,j} + u_{i,j+1} & \text{if } 1 \le i \le M,\ j=1 \\ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} & \text{if } 1 \le i \le M,\ 1 < j < N \\ u_{i,j-1} - 2u_{i,j} + u_{i,1} & \text{if } 1 \le i \le M,\ j=N \end{cases}, \tag{3.1}$$

$$\partial_y^+ \partial_y^- u_{i,j} = \partial_y^- \partial_y^+ u_{i,j} =$$
$$\begin{cases} u_{M,j} - 2u_{i,j} + u_{i+1,j} & \text{if } i=1,\ 1 \le j \le N \\ u_{i-1,j} - 2u_{i,j} + u_{i+1,j} & \text{if } 1 < i < M,\ 1 \le j \le N \\ u_{i-1,j} - 2u_{i,j} + u_{1,j} & \text{if } i=M,\ 1 \le j \le N \end{cases}, \tag{3.2}$$

$$\partial_x^+ \partial_y^+ u_{i,j} = \partial_y^+ \partial_x^+ u_{i,j} =$$
$$\begin{cases} u_{i,j} - u_{i+1,j} - u_{i,j+1} + u_{i+1,j+1} & \text{if } 1 \le i < M,\ 1 \le j < N \\ u_{i,j} - u_{1,j} - u_{i,j+1} + u_{1,j+1} & \text{if } i=M,\ 1 \le j < N \\ u_{i,j} - u_{i+1,j} - u_{i,1} + u_{i+1,1} & \text{if } 1 \le i < M,\ j=N \\ u_{i,j} - u_{1,j} - u_{i,1} + u_{1,1} & \text{if } i=M,\ j=N \end{cases}, \tag{3.3}$$
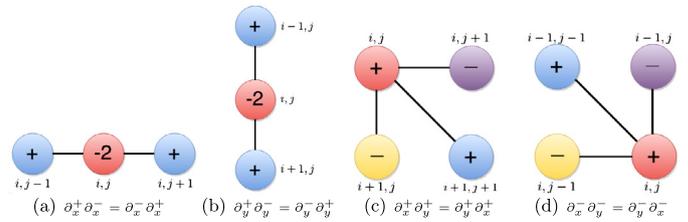

Fig. 1. Discrete second order derivatives.

$$\partial_x^- \partial_y^- u_{i,j} = \partial_y^- \partial_x^- u_{i,j} =$$
$$\begin{cases} u_{i,j} - u_{i,N} - u_{M,j} + u_{M,N} & \text{if } i=1,\ j=1 \\ u_{i,j} - u_{i,j-1} - u_{M,j} + u_{M,j-1} & \text{if } i=1,\ 1 < j \le N \\ u_{i,j} - u_{i,N} - u_{i-1,j} + u_{i-1,N} & \text{if } 1 < i \le M,\ j=1 \\ u_{i,j} - u_{i,j-1} - u_{i-1,j} + u_{i-1,j-1} & \text{if } 1 < i \le M,\ 1 < j \le N \end{cases}. \tag{3.4}$$

Fig. 1 summarises these discrete differential operators. Based on the above forward–backward finite difference scheme, the second order Hessian matrix $\nabla^2 u$ in (2.2) can be discretised as

$$\nabla^2 u = \begin{pmatrix} \partial_x^- \partial_x^+ u & \partial_y^+ \partial_x^+ u \\ \partial_x^+ \partial_y^+ u & \partial_y^- \partial_y^+ u \end{pmatrix}. \tag{3.5}$$

In (3.1)–(3.5), we assume that u satisfies the periodic boundary condition so that the fast Fourier transform (FFT) solver can be applied to solve (2.3) analytically. The numerical approximation of the second order divergence operator div^2 is based on the following expansion

$$div^2(\mathbf{P}) = \partial_x^+ \partial_x^-(\mathbf{P}_1) + \partial_y^- \partial_x^-(\mathbf{P}_2) + \partial_x^- \partial_y^-(\mathbf{P}_3) + \partial_y^+ \partial_y^-(\mathbf{P}_4), \tag{3.6}$$

where $\mathbf{P}$ is a 2×2 matrix whose four components are $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$ and $\mathbf{P}_4$, respectively. For more detailed description of the discretisation of other high order differential operators, we refer the reader to [3,4]. More advanced discretisation on a staggered grid can be found in [21,33,34].

Finally, we address the implementation problem of the first order derivatives of u_σ in (2.7). Since the differentiation and convolution are commutative, we can take the derivative and smooth the image in either order. In this sense, we have

$$\partial_x u_\sigma = \partial_x K_\sigma * u, \quad \partial_y u_\sigma = \partial_y K_\sigma * u. \tag{3.7}$$

Alternatively, the central finite difference scheme can be used to approximate $\partial_x u_\sigma$ and $\partial_y u_\sigma$ to satisfy the rotation-invariant property. Once all necessary discretisation is done, the numerical computation can be implemented.

4. Numerical optimisation algorithm

It is nontrivial to directly solve the TWSO model due to the facts: 1) it is nonsmooth; 2) it couples the tensor $\mathbf{T}$ and Hessian matrix $\nabla^2 u$ in $\|\mathbf{T}\nabla^2 u\|_1$ as shown in (2.4), making the resulting high order Euler–Lagrange equation drastically difficult to discretise to solve computationally. To address these two difficulties, we present an efficient numerical algorithm based on ADMM to minimise the variational model (2.3).

4.1. Alternating Direction Method of Multipliers (ADMM)

ADMM combines the decomposability of the dual ascent with superior convergence properties of the method of multipliers. Recent research [20] unveils that ADMM is also closely related to Douglas–Rachford splitting, Spingarn's method of partial inverses, Dykstra's alternating projections, split Bregman iterative algorithm etc. Given a constrained optimisation problem

$$\min_{x,z} \{ f(x) + g(z) \} \text{ s.t. } Ax + Bz = c, \quad (4.1)$$

where $x \in R^{d_1}$, $z \in R^{d_2}$, $A \in R^{m \times d_1}$, $B \in R^{m \times d_2}$, $c \in R^m$, and both $f(\cdot)$ and $g(\cdot)$ are assumed to be convex. The augmented Lagrange function of problem (4.1) can be written as

$$\mathcal{L}_{\mathcal{A}}(x, z; \rho) = f(x) + g(z) + \frac{\beta}{2} \|Ax + Bz - c - \rho\|_2^2, \quad (4.2)$$

where ρ is an augmented Lagrangian multiplier and $\beta > 0$ is an augmented penalty parameter. At the kth iteration, ADMM attempts to solve problem (4.2) by iteratively minimising $\mathcal{L}_{\mathcal{A}}$ with respect to x and z, and updating ρ accordingly. The resulting optimisation procedure is summarised in Algorithm 1.

Algorithm 1 ADMM.

1: Initialization: Set $\beta > 0$, z^0 and ρ^0.
2: **while** *a stopping criterion is not satisfied* **do**
3: $x^{k+1} = \arg\min_x \left\{ f(x) + \frac{\beta}{2} \|Ax + Bz^k - c - \rho^k\|_2^2 \right\}$.
4: $z^{k+1} = \arg\min_z \left\{ g(z) + \frac{\beta}{2} \|Ax^{k+1} + Bz - c - \rho^k\|_2^2 \right\}$.
5: $\rho^{k+1} = \rho^k - (Ax^{k+1} + Bz^{k+1} - c)$.
6: **end while**

4.2. Application of ADMM to solve the TWSO model

We now use ADMM to solve the minimisation problem of the proposed TWSO model (2.3). The basic idea of ADMM is to first split the original nonsmooth minimisation problem into several subproblems by introducing some auxiliary variables, and then solve each subproblem separately. This numerical algorithm benefits from both solution stability and fast convergence.

In order to implement ADMM, one scalar auxiliary variable $\tilde{u}$ and two 2×2 matrix-valued auxiliary variables $\mathbf{W}$ and $\mathbf{V}$ are introduced to reformulate (2.3) into the following constraint optimisation problem

$$\min_{\tilde{u}, u, \mathbf{W}, \mathbf{V}} \left\{ \frac{\eta}{p} \|1_\Gamma (\tilde{u} - f)\|_p^p + \|\mathbf{W}\|_1 \right\} \text{ s.t. } \tilde{u} = u, \mathbf{V} = \nabla^2 u, \mathbf{W} = \mathbf{TV}, \quad (4.3)$$

where $\mathbf{W} = \begin{pmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{pmatrix}$, $\mathbf{V} = \begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} \\ \mathbf{V}_{21} & \mathbf{V}_{22} \end{pmatrix}$. The constraints $\tilde{u} = u$ and $\mathbf{W} = \mathbf{TV}$ are respectively applied to handle the non-smoothness of the data fidelity term ($p = 1$) and the regularisation term, whilst $\mathbf{V} = \nabla^2 u$ decouples $\mathbf{T}$ and $\nabla^2 u$ in the TWSO. The three constraints together make the calculation for each subproblem point-wise and thus no huge matrix multiplication or inversion is required. To guarantee an optimal solution, the above constrained problem (4.3) can be solved through ADMM summarised in Algorithm 1. Let $\mathcal{L}_{\mathcal{A}}(\tilde{u}, u, \mathbf{W}, \mathbf{V}; s, \mathbf{d}, \mathbf{b})$ be the augmented Lagrange functional of (4.3), which is defined as follows

$$\mathcal{L}_{\mathcal{A}}(\tilde{u}, u, \mathbf{W}, \mathbf{V}; s, \mathbf{d}, \mathbf{b})$$
$$= \frac{\eta}{p} \|1_\Gamma (\tilde{u} - f)\|_p^p + \|\mathbf{W}\|_1 + \frac{\theta_1}{2} \|\tilde{u} - u - s\|_2^2$$
$$+ \frac{\theta_2}{2} \|\mathbf{V} - \nabla^2 u - \mathbf{d}\|_2^2 + \frac{\theta_3}{2} \|\mathbf{W} - \mathbf{TV} - \mathbf{b}\|_2^2, \quad (4.4)$$

where s, $\mathbf{d} = \begin{pmatrix} \mathbf{d}_{11} & \mathbf{d}_{12} \\ \mathbf{d}_{21} & \mathbf{d}_{22} \end{pmatrix}$ and $\mathbf{b} = \begin{pmatrix} \mathbf{b}_{11} & \mathbf{b}_{12} \\ \mathbf{b}_{21} & \mathbf{b}_{22} \end{pmatrix}$ are the augmented Lagrangian multipliers, and θ_1, θ_2 and θ_3 are positive penalty constants controlling the weights of the penalty terms.

We will now decompose the optimisation problem (4.4) into four subproblems with respect to $\tilde{u}$, u, $\mathbf{W}$ and $\mathbf{V}$, and then update the Lagrangian multipliers s, $\mathbf{d}$ and $\mathbf{b}$ until the optimal solution is found and the process converges.

1) $\tilde{u}$-subproblem: This problem $\tilde{u}^{k+1} \leftarrow \min_{\tilde{u}} \mathcal{L}_{\mathcal{A}}(\tilde{u}, u^k, \mathbf{W}^k, \mathbf{V}^k; s^k, \mathbf{d}^k, \mathbf{b}^k)$ can be solved by considering the following minimisation problem

$$\tilde{u}^{k+1} = \arg\min_{\tilde{u}} \left\{ \frac{\eta}{p} \|1_\Gamma (\tilde{u} - f)\|_p^p + \frac{\theta_1}{2} \|\tilde{u} - u^k - s^k\|_2^2 \right\}. \quad (4.5)$$

The solution of (4.5) depends on the choice of p. Given the domain Γ for image denoising or inpainting, the closed-form formulae for the minimisers $\tilde{u}^{k+1}$ under different conditions are

$$\begin{cases} \tilde{u}^{k+1} = (1_\Gamma \eta f + \theta_1 (u^k + s^k)) / (1_\Gamma \eta + \theta_1) & \text{if } p = 2 \\ \tilde{u}^{k+1} = f + \max \left(|\psi^k| - \frac{1_\Gamma \eta}{\theta_1}, 0 \right) \circ sign(\psi^k) & \text{if } p = 1 \end{cases}, \quad (4.6)$$

where $\psi^k = u^k + s^k - f$. $\circ$ and $sign$ symbols denote the componentwise multiplication and signum function, respectively.

2) u-subproblem: We then solve u-subproblem $u^{k+1} \leftarrow \min_u \mathcal{L}_{\mathcal{A}}(\tilde{u}^{k+1}, u, \mathbf{W}^k, \mathbf{V}^k; s^k, \mathbf{d}^k, \mathbf{b}^k)$ by minimising the following problem.

$$u^{k+1} = \arg\min_u \left\{ \frac{\theta_1}{2} \|\tilde{u}^{k+1} - u - s^k\|_2^2 + \frac{\theta_2}{2} \|\mathbf{V}^k - \nabla^2 u - \mathbf{d}^k\|_2^2 \right\}, \quad (4.7)$$

whose closed-form can be obtained using the following FFT under the assumption of the circulant boundary condition (Note that to benefit from the fast FFT solver for image inpainting problems, the introduction of $\tilde{u}$ is compulsory due to the fact that $\mathcal{F}(1_\Gamma u) \neq 1_\Gamma \mathcal{F}(u)$)

$$u^{k+1} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\theta_1 (\tilde{u}^{k+1} - s^k) + \theta_2 div^2 (\mathbf{V}^k - \mathbf{d}^k))}{\theta_1 + \theta_2 \mathcal{F}(div^2 \nabla^2)} \right), \quad (4.8)$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ respectively denote the discrete Fourier transform and inverse Fourier transform; div^2 is a second order divergence operator whose discrete form is defined in (3.6); "−" stands for the pointwise division of matrices. The values of the coefficient matrix $\mathcal{F}(div^2 \nabla^2)$ equal $4(\cos \frac{2\pi q}{N} + \cos \frac{2\pi r}{M} - 2)$, where M and N respectively stand for the image width and height, and $r \in [0, M)$ and $q \in [0, N)$ are the frequencies in the frequency domain. Note that in addition to FFT, AOS and Gauss–Seidel iteration can be applied to minimise the problem (4.7) with very low cost.

3) $\mathbf{W}$-subproblem: We now solve the $\mathbf{W}$-subproblem $\mathbf{W}^{k+1} \leftarrow \min_{\mathbf{W}} \mathcal{L}_{\mathcal{A}}(\tilde{u}^{k+1}, u^{k+1}, \mathbf{W}, \mathbf{V}^k; s^k, \mathbf{d}^k, \mathbf{b}^k)$. Note that the unknown matrix-valued variable $\mathbf{W}$ is componentwise separable, which can be effectively solved through the analytical shrinkage operation, also known as the soft generalised thresholding equation

$$\mathbf{W}^{k+1} = \arg\min_{\mathbf{W}} \left\{ \|\mathbf{W}\|_1 + \frac{\theta_3}{2} \|\mathbf{W} - \mathbf{TV}^k - \mathbf{b}^k\|_2^2 \right\}, \quad (4.9)$$

whose solution $\mathbf{W}^{k+1}$ is given by

$$\mathbf{W}^{k+1} = \max \left(\left| \mathbf{TV}^k + \mathbf{b}^k \right| - \frac{1}{\theta_3}, 0 \right) \circ \frac{\mathbf{TV}^k + \mathbf{b}^k}{|\mathbf{TV}^k + \mathbf{b}^k|}, \quad (4.9)$$

with the convention that $0 \cdot (0/0) = 0$.

4) The $\mathbf{V}$-subproblem: Given fixed u^{k+1}, $\mathbf{W}^{k+1}$, $\mathbf{d}^k$, $\mathbf{b}^k$, the solution $\mathbf{V}^{k+1}$ of the $\mathbf{V}$-subproblem $\mathbf{V}^{k+1} \leftarrow \min_{\mathbf{V}} \mathcal{L}_{\mathcal{A}}(\tilde{u}^{k+1}, u^{k+1}, \mathbf{W}^{k+1}, \mathbf{V}; s^k, \mathbf{d}^k, \mathbf{b}^k)$ is equivalent to solving the following least-square optimisation problem

$$\mathbf{V}^{k+1} = \arg\min_{\mathbf{V}} \left\{ \frac{\theta_2}{2} \|\mathbf{V} - \nabla^2 u^{k+1} - \mathbf{d}^k\|_2^2 + \frac{\theta_3}{2} \|\mathbf{W}^{k+1} - \mathbf{TV} - \mathbf{b}^k\|_2^2 \right\},$$

which results in the following linear system with respect to each component in the variable $\mathbf{V}^{k+1}$

$$
\begin{cases}
\mathbf{R}_{11}\mathbf{V}_{11}^{k+1} + \mathbf{R}_{21}\mathbf{V}_{21}^{k+1} = \theta_2(\partial_x^- \partial_x^+ u^{k+1} + \mathbf{d}_{11}^k) \\
\qquad\qquad\qquad - \theta_3(\mathbf{T}_{11}\mathbf{Q}_{11} + \mathbf{T}_{21}\mathbf{Q}_{21}) \\
\mathbf{R}_{12}\mathbf{V}_{11}^{k+1} + \mathbf{R}_{22}\mathbf{V}_{21}^{k+1} = \theta_2(\partial_x^+ \partial_y^+ u^{k+1} + \mathbf{d}_{21}^k) \\
\qquad\qquad\qquad - \theta_3(\mathbf{T}_{12}\mathbf{Q}_{11} + \mathbf{T}_{22}\mathbf{Q}_{21}) \\
\mathbf{R}_{11}\mathbf{V}_{12}^{k+1} + \mathbf{R}_{21}\mathbf{V}_{22}^{k+1} = \theta_2(\partial_y^+ \partial_x^+ u^{k+1} + \mathbf{d}_{12}^k) \\
\qquad\qquad\qquad - \theta_3(\mathbf{T}_{11}\mathbf{Q}_{12} + \mathbf{T}_{21}\mathbf{Q}_{22}) \\
\mathbf{R}_{12}\mathbf{V}_{12}^{k+1} + \mathbf{R}_{22}\mathbf{V}_{22}^{k+1} = \theta_2(\partial_y^- \partial_y^+ u^{k+1} + \mathbf{d}_{22}^k) \\
\qquad\qquad\qquad - \theta_3(\mathbf{T}_{12}\mathbf{Q}_{12} + \mathbf{T}_{22}\mathbf{Q}_{22})
\end{cases}, \qquad (4.10)
$$

where we define $\mathbf{R}_{11} = \theta_3(\mathbf{T}_{11}^2 + \mathbf{T}_{21}^2) + \theta_2$, $\mathbf{R}_{12} = \mathbf{R}_{21} = \theta_3(\mathbf{T}_{11}\mathbf{T}_{12} + \mathbf{T}_{21}\mathbf{T}_{22})$, $\mathbf{R}_{22} = \theta_3(\mathbf{T}_{12}^2 + \mathbf{T}_{22}^2) + \theta_2$; $\mathbf{Q}_{11} = \mathbf{b}_{11}^k - \mathbf{W}_{11}^{k+1}$, $\mathbf{Q}_{12} = \mathbf{b}_{12}^k - \mathbf{W}_{12}^{k+1}$, $\mathbf{Q}_{21} = \mathbf{b}_{21}^k - \mathbf{W}_{21}^{k+1}$ and $\mathbf{Q}_{22} = \mathbf{b}_{22}^k - \mathbf{W}_{22}^{k+1}$. Note that the closed-forms of $\mathbf{V}_{11}^{k+1}$ and $\mathbf{V}_{21}^{k+1}$ can be calculated from the first two equations in (4.10), whilst the last two equations lead to the analytical solutions of $\mathbf{V}_{12}^{k+1}$ and $\mathbf{V}_{22}^{k+1}$.

5) s, $\mathbf{d}$ *and* $\mathbf{b}$ *update*: At each iteration, we update the augmented Lagrangian multiplies s, $\mathbf{d}$ and $\mathbf{b}$ as shown from Step 09 to 11 in Algorithm 2.

Algorithm 2 ADMM for the proposed TWSO model for image restoration.

01: Input: f, 1_Γ, p, ρ, σ, γ, $\mathcal{C}$, C, η and $(\theta_1, \theta_2, \theta_3)$.
02: Initialise: $u = f$, $\mathbf{W}^0 = 0$, $\mathbf{V}^0 = 0$, $\mathbf{d}^0 = 0$, $\mathbf{b}^0 = 0$.
03: **while** *some stopping criterion is not satisfied* **do**
04:    Compute $\tilde{u}^{k+1}$ according to (4.6).
05:    Compute u^{k+1} according to (4.8).
06:    Compute $\mathbf{W}^{k+1}$ according to (4.9).
07:    Compute $\mathbf{V}^{k+1}$ according to (4.10).
08:    Compute $\mathbf{T}^{k+1} = \mathcal{D}\left(J_\rho\left(\nabla u_\sigma^{k+1}\right)\right)$ according to (2.9) or (2.10).
09:    Update Lagrangian multiplier $s^{k+1} = s^k + u^{k+1} - \tilde{u}^{k+1}$.
10:    Update Lagrangian multiplier $\mathbf{d}^{k+1} = \mathbf{d}^k + \nabla^2 u^{k+1} - \mathbf{V}^{k+1}$.
11:    Update Lagrangian multiplier $\mathbf{b}^{k+1} = \mathbf{b}^k + (\mathbf{T}\mathbf{V})^{k+1} - \mathbf{W}^{k+1}$.
12: **end while**

In summary, an ADMM-based iterative algorithm was developed to decompose the original nonsmooth minimisation problem into four simple subproblems, each of which has a closed-form solution that can be point-wisely solved using the efficient numerical methods (i.e. FFT and shrinkage). The overall numerical optimisation algorithm of our proposed method for image restoration can be summarised in Algorithm 2. We note that in order to obtain better denoising and inpainting results, we iteratively refine the diffusion tensor $\mathbf{T}$ using the recovered image (i.e. u^{k+1} as shown in Step 8 in Algorithm 2). This refinement is crucial as it provides more accurate tensor information for the next round iteration, thus leading to more pleasant restoration results. Due to the reweighed process the convergence of the algorithm is not guaranteed. However, the algorithm is still fast and stable, as evident in our experiments.

5. Numerical experiments

We conduct numerical experiments to compare the TWSO model with state-of-the-art approaches for image inpainting and denoising. The images used in this study are normalised to [0,255] before inpainting and denoising. The metrics for quantitative evaluation of different methods are the peak signal-to-noise ratio (PSNR) and structure similarity index map (SSIM). The higher PSNR and SSIM a method obtains the better the method will be. In order to maximise the performance of all the compared methods, we carefully adjust their built-in parameters such that the resulting PSNR and SSIM by these methods are maximised. In the following, we shall introduce how to select these parameters for the TWSO model in detail.

5.1. Parameters

For image inpainting, there are 8 parameters in the proposed model: the standard deviations ρ and σ in (2.5), the constant γ and structure threshold $\mathcal{C}$ in (2.10), the regularisation parameter η in (4.4), and the penalty parameters θ_1, θ_2 and θ_3 in (4.4).

The parameter ρ averages orientation information and helps to stabilise the directional behaviour of the diffusion. Large interrupted lines can be connected if ρ is equal or larger than the gap size. The selection of ρ for each example has been listed in Table 1. σ is used to decrease the influence of noise and guarantee numerical stabilisation and thereby gives a more robust estimation of the structure tensor J_ρ in (2.5). γ in (2.10) determines how steep the exponential function is. Weickert [32] suggested $\gamma \in (0, 1)$ and $\gamma \ll 1$. Therefore γ is fixed at 0.01. The structure threshold $\mathcal{C}$ affects how the method understands local image structures. The larger the parameter value is, the more sensitive the method will be. However, if $\mathcal{C}$ goes to infinity, our TWSO model reduces to the isotropic SOTV model. $\mathcal{C}$'s selection for different examples is presented in Table 1. The regularisation parameter η controls the smoothness of the restored image. Smaller η leads to smoother restoration, and it should be relatively large to make the inpainted image close to the original image. The value of η is chosen as 100 for all the inpainting experiments.

We now illustrate how to choose the penalty parameters θ_1, θ_2 and θ_3. Due to the augmented Lagrangian multipliers used, different combinations of the three parameters will provide similar inpainting results. However, the convergence speed will be affected. In order to guarantee the convergence speed, we use the rule introduced in [35] to tune θ_1, θ_2 and θ_3. Namely, the residuals R_i^k ($i = 1, 2, 3$) defined in (5.1), the relative errors of the Lagrangian multipliers L_i^k ($i = 1, 2, 3$) defined in (5.2) and the relative errors of u defined in (5.3) should reduce to zero with nearly the same speed as the iteration proceeds. For example, in all the inpainting experiments, setting $\theta_1 = 0.01$, $\theta_2 = 0.01$ and $\theta_3 = 0.001$ gives a good convergence speed as each pair of curves in Fig. 2 (a) and (b) decrease to zero with very similar speed. In addition, Fig. 2 (d) shows that the numerical energy of the TWSO model (2.3) has reached to a steady state after only a few iterations. Fig. 2 shows the plots for a real inpainting example in Fig. 7.

The residuals R_i^k ($i = 1, 2, 3$) are defined as

$$
\left(R_1^k, R_2^k, R_3^k\right)
$$
$$
= \left(\frac{1}{|\Omega|}\|\tilde{u}^k - \tilde{u}^k\|_{L^1}, \ \frac{1}{|\Omega|}\|\mathbf{V}^k - \nabla^2 u^k\|_{L^1}, \ \frac{1}{|\Omega|}\|\mathbf{W}^k - \mathbf{T}\mathbf{V}^k\|_{L^1} \right), \qquad (5.1)
$$

where $\|\cdot\|_{L^1}$ denotes the L^1-norm on the image domain Ω. The relative errors of the augmented Lagrangian multipliers L_i^k ($i = 1, 2, 3$) are defined as

$$
\left(L_1^k, L_2^k, L_3^k\right)
$$
$$
= \left(\frac{||s^k - s^{k-1}||_{L^1}}{||s^{k-1}||_{L^1}}, \ \frac{||\mathbf{d}^k - \mathbf{d}^{k-1}||_{L^1}}{||\mathbf{d}^{k-1}||_{L^1}}, \ \frac{||\mathbf{b}^k - \mathbf{b}^{k-1}||_{L^1}}{||\mathbf{b}^{k-1}||_{L^1}} \right). \quad (5.2)
$$

The relative errors of u are defined as

$$
R_u^k = \frac{||u^k - u^{k-1}||_{L^1}}{||u^{k-1}||_{L^1}}. \qquad (5.3)
$$

In summary, there are only two built-in parameters ρ and $\mathcal{C}$ that need to be adjusted for inpainting different images (see Table 1 for their selections). Consequently, the proposed TWSO is a very robust model in terms of parameter tuning.
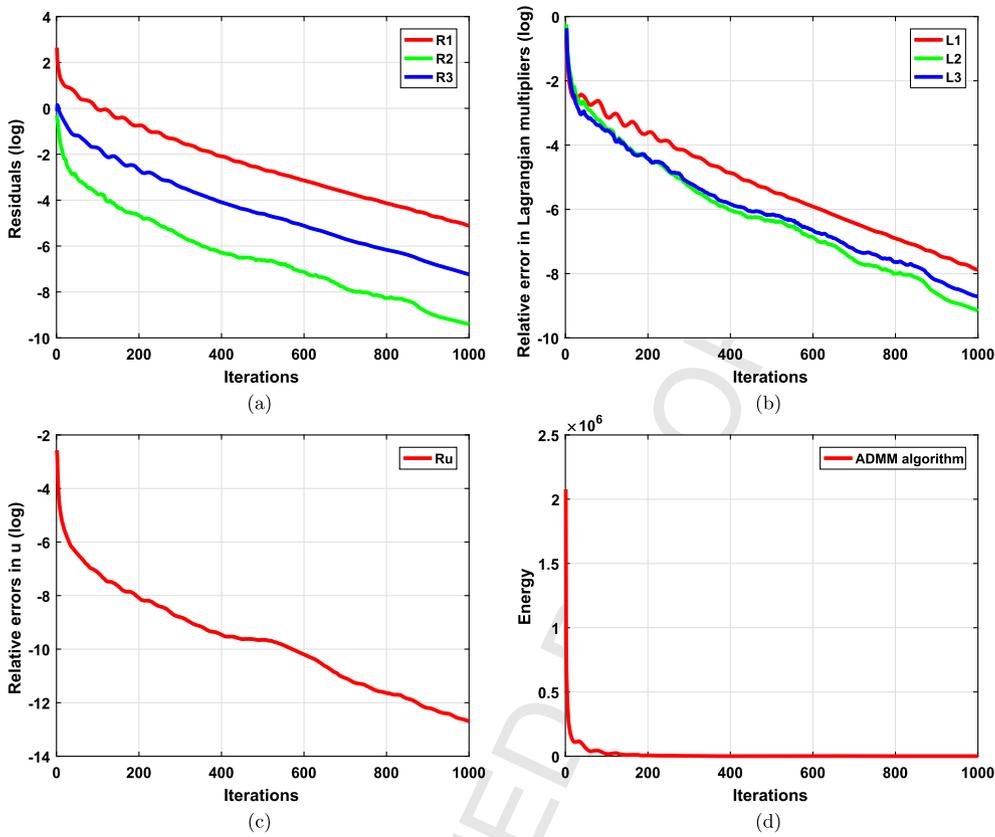
Fig. 2. Plots of the residuals (5.1), the relative errors of the Lagrangian multipliers (5.2), the relative errors of the function u in (5.3), and the energy of the proposed model (2.3) against number of iterations for the real inpainting example in Fig. 7.
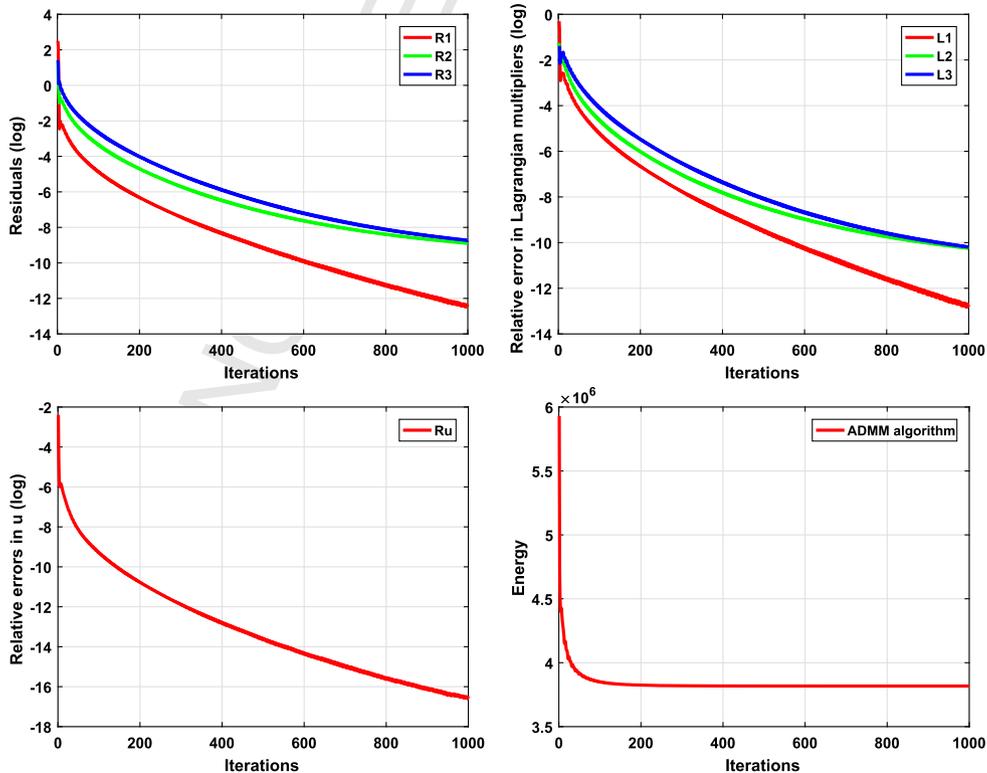

Fig. 3. Plots of the residuals (5.1), relative errors of the Lagrangian multipliers (5.2), relative errors of the function u in (5.3), and the energy of the TWSO model (2.3) against the number of iterations for the synthetic image in Fig. 9.

Table 1

Parameters used in the TWSO model for the examples in Figs. 4–7 and Figs. 9–10.

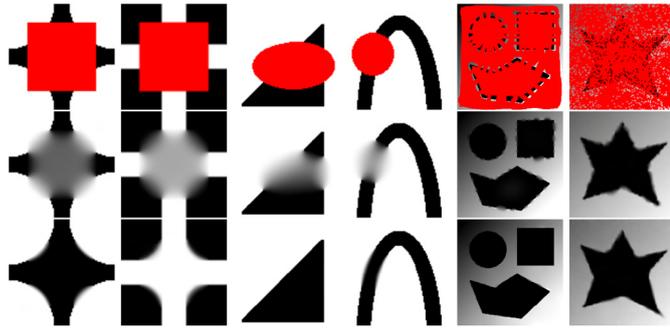| Parameter	Fig. 4			Fig. 5	Fig. 6	Fig. 7	Fig. 9	Fig. 10
	1st, 2nd columns	3rd, 4th columns	5th, 6th columns	All rows				
ρ	10	10	10	10	10	5	1	1
σ	1	1	1	1	1	1	1	1
γ	0.01	0.01	0.01	0.01	0.01	0.01	–	–
$\mathcal{C}/C$	10	1	5	1	1	1	5	5
η	100	100	100	100	100	100	0.04	0.05
θ_1	0.01	0.01	0.01	0.01	0.01	0.01	5	5
θ_2	0.01	0.01	0.01	0.01	0.01	0.01	5	5
θ_3	0.001	0.001	0.001	0.001	0.001	0.001	10	10

Fig. 4. Performance comparison of the SOTV and TWSO models on some degraded images. The red regions in the first row are inpainting domains. The second and third rows show the corresponding inpainting results by the SOTV and the TWSO models, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
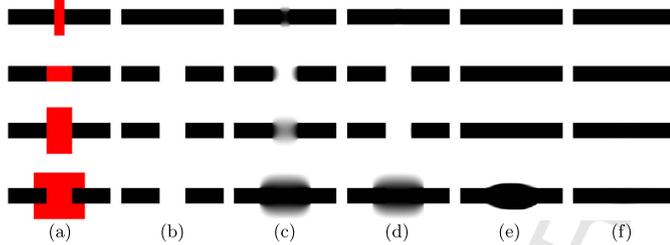

Fig. 5. Test on a black stripe with different geometries. (a): Images with red inpainting regions. The last three inpainting gaps have the same width but with different geometry; (b–f): results of TV, SOTV, TGV, Euler's elastica, and TWSO, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For image denoising, there are 7 parameters in the TWSO model: ρ and σ in (2.5), the contrast parameter C in (2.9), and η, θ_1, θ_2 and θ_3 in (4.4). Each parameter plays a similar role as its counterpart in the inpainting model. However, in contrast to inpainting, ρ here should be small because we do not need to connect gaps for image denoising. We have also found that $\sigma = 1$ is sufficient for the cases considered in this paper. C determines how much anisotropy the TWSO model will bring to the resulting image. We illustrate the effect of this parameter using the example in Fig. 10. The regularisation parameter η should be small when image noise level is high. The three penalty parameters are chosen based on the quantities defined in (5.1), (5.2) and (5.3). Fig. 3 shows that using $\theta_1 = 5$, $\theta_2 = 5$ and $\theta_3 = 10$ leads to fast convergence speed for the synthetic image in Fig. 9 and these values are used for the rest of image denoising examples. Finally, the parameters selected are given in the last two columns in Table 1.

5.2. Image inpainting

In this section we test the capability of the TWSO model for image inpainting and compare it with several state-of-the-art inpainting methods such as the TV [36], TGV [37], SOTV [2,22], and Euler's elastica [21,38] models. We denote the inpainting domain as D in the following experiments. The region Γ in equation (2.3) is $\Omega \backslash D$. We use $p = 2$ for all examples in image inpainting.

Fig. 4 illustrates the importance of the tensor **T** in the proposed TWSO model. The damaged images overlaid with the red inpainting domains are shown in the first row. Second row and third row show that the TWSO performs much better than SOTV. SOTV introduces blurring to the inpainted regions, whilst TWSO recovers these shapes very well without causing much blurring. In addition to the capability of inpainting large gaps, the TWSO interpolates smoothly along the level curves of the image in the inpainting domain, indicating the effectiveness of tensor in TWSO for inpainting.

In Fig. 5, we show how different geometries of the inpainting region affect the results by different methods. Column (b) illustrates that TV performs satisfactorily if the inpainting area is thin. However, it fails in images that contain large gaps. Column (c) indicates that the inpainting result by SOTV depends on the geometry of the inpainting region. It is able to inpaint large gaps but at the
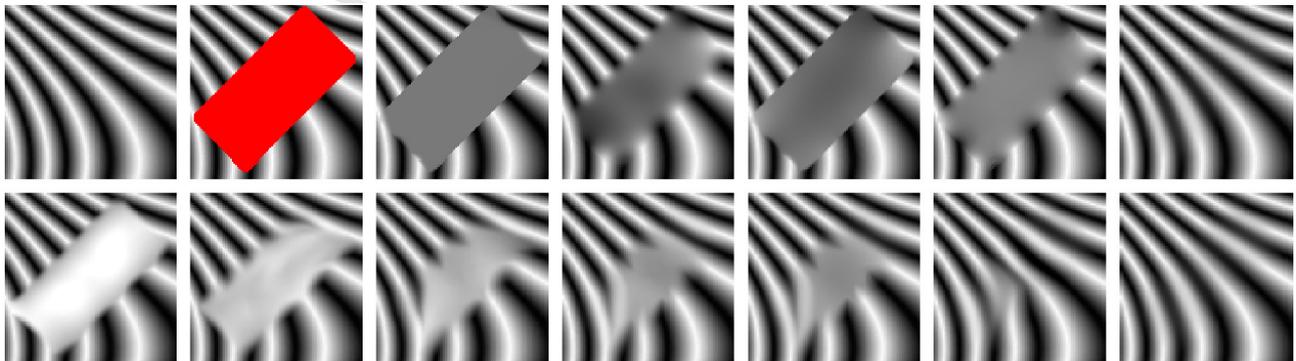

Fig. 6. Inpainting a smooth image with large gaps. First row from left to right are the original image, degraded image, inpainting results of TV, SOTV, TGV, Euler's elastica, and TWSO, respectively; Second row shows the intermediate results obtained by the TWSO model. Image is from [33].

Fig. 7. Real example inpainting. "Cornelia, mother of the Gracchi" by J. Suvee (Louvre). From left to right are degraded image, inpainting results of TV, SOTV, TGV, Euler's elastica, and TWSO, respectively; Second row shows the local magnification of the corresponding results in the first row.

Table 2
Comparison of PSNR and SSIM of different methods in Figs. 5–7.

| Figure # | PSNR test | | | | | | SSIM value | | | | | |
| | Fig. 5 | | | | Fig. 6 | Fig. 7 | Fig. 5 | | | | Fig. 6 | Fig. 7 |
	1st row	2nd row	3rd row	4th row			1st row	2nd row	3rd row	4th row		
Degraded	18.5316	12.1754	11.9285	11.6201	8.1022	14.9359	0.9418	0.9220	0.8778	0.8085	0.5541	0.8589
TV	**Inf**	12.7107	12.7107	12.7107	14.4360	35.7886	**1.0000**	0.9230	0.9230	0.9230	0.6028	0.9936
SOTV	28.7742	14.1931	18.5122	12.8831	14.9846	35.6840	0.9739	0.9289	0.9165	0.8118	0.6911	0.9951
TGV	43.9326	12.7446	12.7551	13.4604	14.0600	36.0854	0.9995	0.9232	0.9233	0.8167	0.6014	0.9941
Euler's elastica	70.2979	54.0630	51.0182	13.0504	14.8741	34.3064	**1.0000**	0.9946	0.9982	0.8871	0.6594	0.9878
TWSO	**Inf**	**Inf**	**Inf**	**43.3365**	**27.7241**	**41.1400**	**1.0000**	**1.0000**	**1.0000**	**0.9992**	**0.9689**	**0.9976**

Table 3
Mean and standard deviation (SD) of PSNR and SSIM calculated using 5 different methods for image inpainting over 100 images from the Berkeley database BSDS500 with 4 different random masks.

| Missing | PSNR value (Mean±SD) | | | | SSIM value (Mean±SD) | | | |
	40%	60%	80%	90%	40%	60%	80%	90%
Degraded	9.020 ± 0.4208	7.250 ± 0.4200	6.010 ± 0.4248	5.490 ± 0.4245	0.05 ± 0.0302	0.03 ± 0.0158	0.01 ± 0.0065	0.007 ± 0.003
TV	31.99 ± 3.5473	28.68 ± 3.3217	24.61 ± 2.7248	20.94 ± 2.2431	0.92 ± 0.0251	0.84 ± 0.0412	0.67 ± 0.0638	0.48 ± 0.0771
TGV	31.84 ± 3.9221	28.70 ± 3.8181	25.52 ± 3.6559	23.32 ± 3.5221	0.93 ± 0.0310	0.87 ± 0.0580	0.75 ± 0.1023	0.64 ± 0.1372
SOTV	32.43 ± 4.9447	29.39 ± 4.5756	26.38 ± 4.2877	24.41 ± 4.0720	0.94 ± 0.0308	0.89 ± 0.0531	0.80 ± 0.0902	0.71 ± 0.1204
Euler's elastica	32.12 ± 4.0617	29.21 ± 3.9857	26.30 ± 3.8194	24.34 ± 3.6208	0.94 ± 0.0288	0.88 ± 0.0526	0.78 ± 0.0901	0.69 ± 0.1208
TWSO	34.33 ± 3.4049	31.12 ± 3.0650	27.66 ± 3.6540	25.26 ± 3.2437	0.95 ± 0.0204	0.90 ± 0.0456	0.82 ± 0.0853	0.73 ± 0.1061

cost of introducing blurring. Mathematically, TGV is a combination of TV and SOTV, so its results, as shown in Column (d), are similar to those of TV and SOTV. TGV results also seem to depend on the geometry of the inpainting area. The last column shows that TWSO inpaints the images perfectly without any blurring and regardless local geometry. TWSO is also slightly better than Euler's elastica, which has proven to be very effective in dealing with larger gaps.

We now show an example of inpainting a smooth image with large gaps. Fig. 6 shows that all the methods, except the TWSO model, fail in inpainting the large gaps in the image. This is due to the fact that TWSO integrates the tensor **T** with its eigenvalues defined in (2.10), which makes it suitable for restoring linear structures, as shown in the second row of Fig. 6.

Fig. 7 compares the inpainting of a real image by all the methods. The inpainting result of the TV model seems to be more satisfactory than those of SOTV and TGV, though it produces piecewise constant restoration. From the second row, neither TV nor TGV is able to connect the gaps on the cloth. SOTV reduces some gaps but blurs the connecting region. Euler's elastica performs better than TV, SOTV and TGV, but no better than the proposed TWSO model. Table 2 shows that the TWSO is the most accurate among all the methods compared for the examples in Figs. 5–7.

Finally, we evaluate TV, SOTV, TGV, Euler's elastica and TWSO on the Berkeley database BSDS500 for image inpainting. We use 4 random masks (i.e. 40%, 60%, 80% and 90% pixels are missing) for each of the 100 images randomly selected from the database. The performance of each method for each mask is measured in terms of mean and standard derivation of PSNR and SSIM over all 100 images. The results are demonstrated in the following Table 3 and Fig. 8. The accuracy of the inpainting results obtained by different methods decreases as the percentage of missing pixels region becomes larger. The highest averaged PSNR values are again achieved by the TWSO, demonstrating its effectiveness for image inpainting.

5.3. Image denoising

For denoising images corrupted by Gaussian noise, we use $p = 2$ in (2.3) and the Γ in (2.3) is the same as the image domain Ω. We compare the proposed TWSO model with the Perona–Malik (PM) [39], coherent enhancing diffusion (CED) [32], total variation (TV) [40], second order total variation (SOTV) [1,22,26], total generalised variation (TGV) [30], and extended anisotropic diffusion model[4] (EAD) [11] on both synthetic and real images.

Fig. 9 shows the denoised results on a synthetic image (a) by different methods. The results by the PM and TV models, as shown in (c) and (e), have a jagged appearance (i.e. staircase artefact).

---

[4] Code: http://liu.diva-portal.org/smash/get/diva2:543914/SOFTWARE01.zip
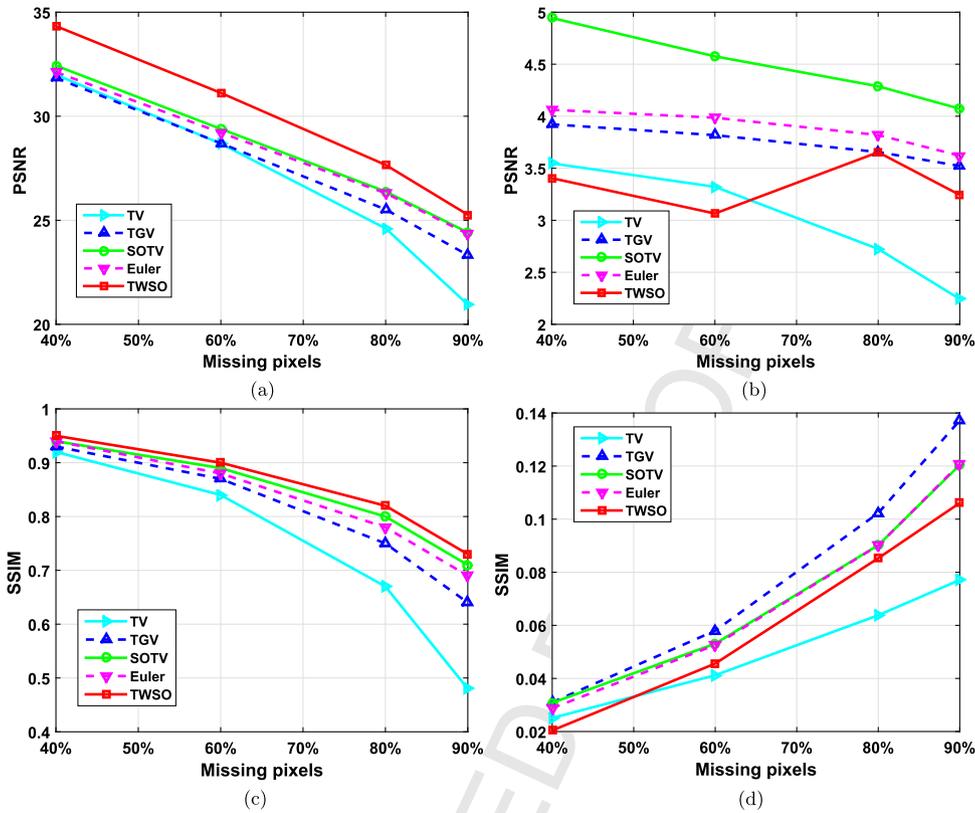
Fig. 8. Plots of mean and standard derivation in Table 3 obtained by 5 different methods over 100 images from the Berkeley database BSDS500 with 4 different random masks. (a) and (b): mean and standard derivation plots of PSNR; (c) and (d): mean and standard derivation plots of SSIM.
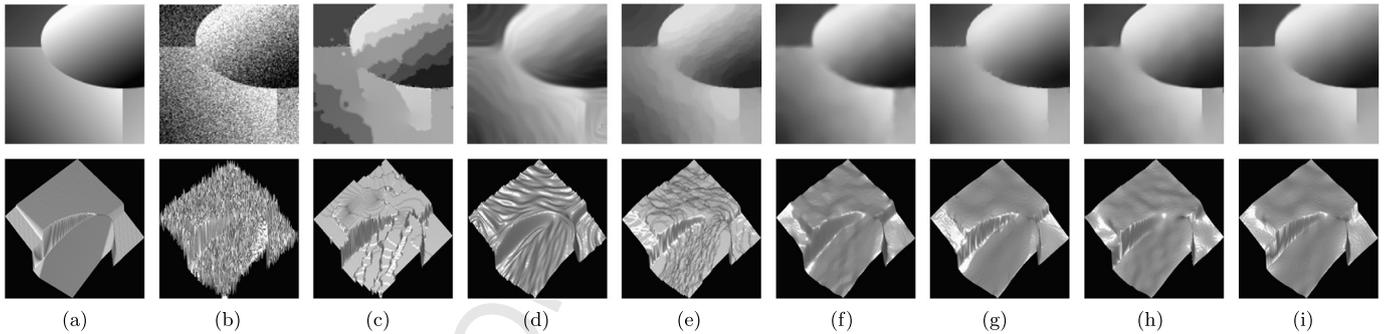

Fig. 9. Denoising results of a synthetic test image. (a) Clean data; (b) Image corrupted by 0.02 variance Gaussian noise; (c–i): Results of PM, CED, TV, SOTV, TGV, EAD, and TWSO, respectively. Second row shows the isosurface rendering of the corresponding results above.

However, the scale-space-based PM model shows much stronger staircase effect than the energy-based variational TV model for denoising piecewise smooth images. Due to the anisotropy, the result (d) by the CED method displays strong directional characteristics. Due to the high order derivatives involved, the SOTV, TGV and TWSO models can reduce the staircase artefact. However, because the SOTV imposes too much regularity on the image, it smears the sharp edges of the objects in (f). Better results are obtained by the TGV, EAD and TWSO models, which show no staircase and blurring effects, though TGV leaves some noise near the discontinuities and EAD over-smooths image edges, as shown in (g) and (h).

Fig. 10 presents the denoised results on a real image (a) by different methods. Both the CED and the proposed TWSO models use the anisotropic diffusion tensor $\mathbf{T}$. CED distorts the image while TWSO does not. The reason for this is that TWSO uses the eigenvalues of $\mathbf{T}$ defined in (2.9), which has two advantages: i) it allows us to control the degree of the anisotropy of TWSO. If the contrast parameter C in (2.9) goes to infinity, the TWSO model degenerates to the isotropic SOTV model. The larger C is, the less anisotropy TWSO has. The eigenvalues (2.10) used in CED however are not able to adjust the anisotropy; ii) it can determine if there exists diffusion in TWSO along the direction parallel to the image gradient. The diffusion halts along the image gradient direction if λ_1 in (2.9) is small and encouraged if λ_1 is large. By choosing a suitable C, (2.9) allows TWSO to diffuse the noise along object edges and prohibit the diffusion across edges. The eigenvalue λ_1 used in (2.10) however remains small (i.e. $\lambda_1 \ll 1$) all the time, meaning that the diffusion along image gradient in CED is always prohibited. CED thus only prefers the orientation that is perpendicular to the image gradient, which explains why CED distorts the image.

In addition to qualitative evaluation of different methods in Fig. 9 and 10, we also calculate the PSNR and SSIM values for these methods and show them in Table 4 and Fig. 11. These metrics show that the PDE-based methods (i.e. PM and CED) perform worse than the variational methods (i.e. TV, SOTV, TGV, EAD and
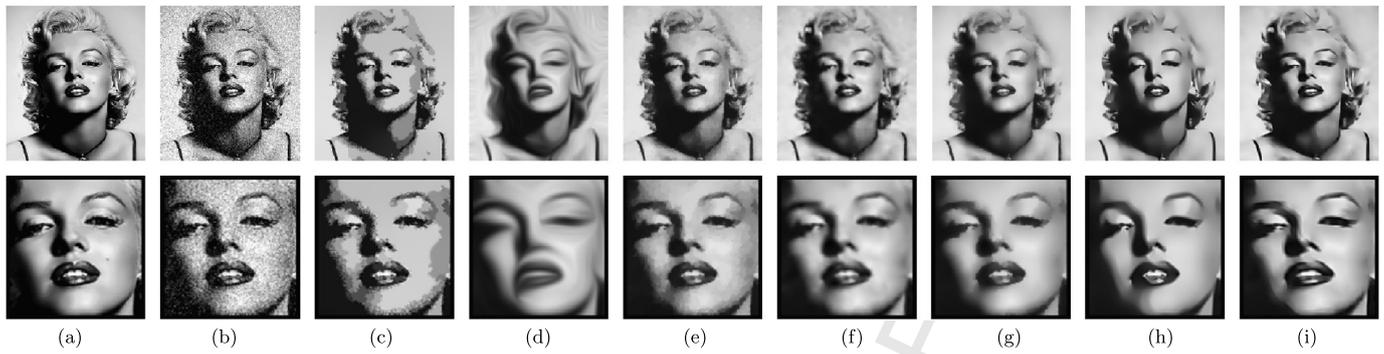
Fig. 10. Noise reduction results of a real test image. (a) Clean data; (b) Noisy data corrupted by 0.015 variance Gaussian noise; (c–i): Results of PM, CED, TV, SOTV, TGV, EAD, and TWSO, respectively. The second row shows local magnification of the corresponding results in the first row.

Table 4
Comparison of PSNR and SSIM using different methods on Fig. 9 and 10 with different noise variances.

| | Fig. 9 | | | | | | | | | | Fig. 10 | | | | | | | | | |
| | PSNR value | | | | | SSIM value | | | | | PSNR value | | | | | SSIM value | | | | |
Noise variance	0.005	0.01	0.015	0.02	0.025	0.005	0.01	0.015	0.02	0.025	0.005	0.01	0.015	0.02	0.025	0.005	0.01	0.015	0.02	0.025
Degraded	23.1328	20.2140	18.4937	17.3647	16.4172	0.2776	0.1827	0.1419	0.1212	0.1038	23.5080	20.7039	19.0771	17.9694	17.1328	0.5125	0.4167	0.3625	0.3261	0.3009
PM	26.7680	23.3180	20.0096	19.4573	18.9757	0.8060	0.7907	0.7901	0.7850	0.7822	25.8799	24.0247	22.4998	22.0613	21.1334	0.7819	0.7273	0.6991	0.6760	0.6579
CED	29.4501	29.2004	28.4025	28.2601	27.7806	0.9414	0.9264	0.9055	0.8949	0.8858	23.8593	21.3432	20.2557	20.1251	20.0243	0.6728	0.6588	0.6425	0.6375	0.6233
TV	32.6027	29.9910	28.9663	28.3594	27.1481	0.9387	0.9233	0.9114	0.9017	0.8918	27.3892	25.5942	24.4370	23.5511	22.8590	0.8266	0.7826	0.7529	0.7331	0.7149
SOTV	32.7415	31.1342	30.1882	29.4404	28.3784	0.9653	0.9552	0.9474	0.9444	0.9332	26.1554	24.6041	23.5738	23.2721	22.5986	0.8252	0.7841	0.7546	0.7367	0.7201
TGV	36.1762	34.6838	33.1557	32.0843	30.6564	0.9821	0.9745	0.9688	0.9644	0.9571	27.5006	25.6966	24.5451	23.6456	22.9540	0.8375	0.7937	0.7659	0.7456	0.7386
EAD	34.8756	33.7161	32.6091	31.6987	30.6599	0.9764	0.9719	0.9678	0.9610	0.9548	27.0495	25.0156	24.1903	23.4999	23.2524	0.8232	0.7791	0.7606	0.7424	0.7382
TWSO	**36.3192**	**34.7220**	**33.4009**	**32.7334**	**31.5537**	**0.9855**	**0.9771**	**0.9726**	**0.9704**	**0.9658**	**27.5997**	**25.8511**	**24.9060**	**24.1328**	**23.4983**	**0.8437**	**0.8063**	**0.7811**	**0.7603**	**0.7467**
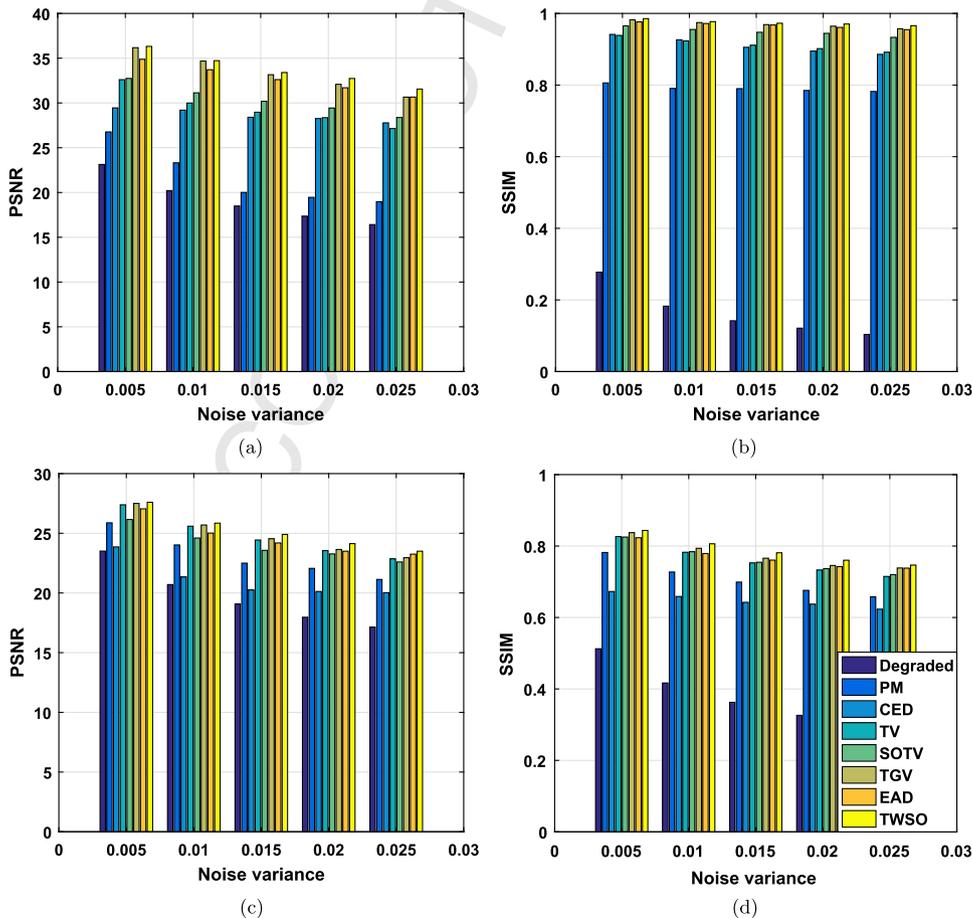
Fig. 11. Quantitative image quality evaluation for Gaussian noise reduction in Table 4. (a) and (b): PSNR and SSIM values of various denoised methods for Fig. 9 (a) corrupted by different noise levels; (c) and (d): PSNR and SSIM values of various denoised methods for Fig. 10 (a) corrupted by different noise levels.

Table 5

Mean and standard deviation (SD) of PSNR and SSIM calculated using 7 different methods for image denoising over 100 images from the Berkeley database BSDS500 with 5 different noise variances.

|	PSNR value (Mean±SD)					SSIM value (Mean±SD)				
Noise variance	0.005	0.01	0.015	0.02	0.025	0.005	0.01	0.015	0.02	0.025
Degraded	23.19 ± 0.2173	20.28 ± 0.2831	18.61 ± 0.3203	17.44 ± 0.3492	16.55 ± 0.3683	0.49 ± 0.1229	0.37 ± 0.1164	0.31 ± 0.1078	0.27 ± 0.1004	0.24 ± 0.0941
PM	26.39 ± 1.8442	24.70 ± 1.9644	23.30 ± 1.6893	22.73 ± 2.0353	21.92 ± 2.0287	0.73 ± 0.0656	0.67 ± 0.0798	0.62 ± 0.0787	0.59 ± 0.1091	0.56 ± 0.1221
CED	25.06 ± 3.2078	24.71 ± 2.9300	24.39 ± 2.7133	24.09 ± 2.5376	23.80 ± 2.3884	0.66 ± 0.1060	0.64 ± 0.0976	0.62 ± 0.0907	0.60 ± 0.0856	0.58 ± 0.0811
TV	27.86 ± 2.8359	27.20 ± 2.5462	26.40 ± 2.1771	25.57 ± 1.8715	24.82 ± 1.6308	0.76 ± 0.0859	0.75 ± 0.0735	0.71 ± 0.0591	0.66 ± 0.0508	0.62 ± 0.0510
SOTV	26.50 ± 3.1114	26.13 ± 2.9269	25.75 ± 2.7474	25.37 ± 2.5885	25.00 ± 2.4484	0.72 ± 0.1055	0.71 ± 0.1024	0.69 ± 0.0985	0.68 ± 0.0947	0.66 ± 0.0909
TGV	27.84 ± 2.8632	27.23 ± 2.5818	26.42 ± 2.1958	25.59 ± 1.8814	24.84 ± 1.6360	0.76 ± 0.0875	0.75 ± 0.0757	0.71 ± 0.0603	0.66 ± 0.0513	0.62 ± 0.0512
EAD	28.90 ± 3.6852	28.17 ± 3.7699	27.14 ± 2.5467	26.14 ± 3.2178	25.00 ± 2.2179	0.79 ± 0.1027	0.77 ± 0.1033	0.72 ± 0.2609	0.67 ± 0.2571	0.63 ± 0.0571
TWSO	29.65 ± 2.6622	28.24 ± 2.2530	27.19 ± 2.0262	26.40 ± 1.8969	25.95 ± 1.9856	0.81 ± 0.0669	0.78 ± 0.0609	0.73 ± 0.0581	0.70 ± 0.0580	0.69 ± 0.0676
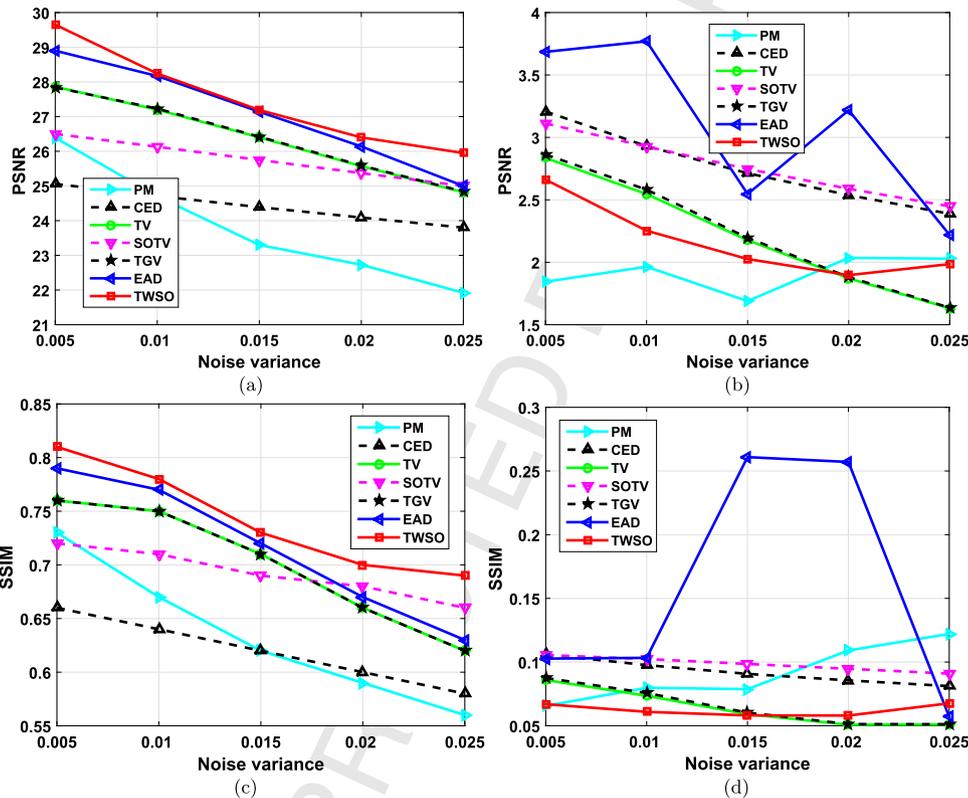
Fig. 12. Plots of mean and standard derivation in Table 5 obtained by 7 different methods over 100 images from the Berkeley database BSDS500 with 5 different noise variances. (a) and (b): mean and standard derivation plots of PSNR; (c) and (d): mean and standard derivation plots of SSIM.

TWSO). The TV model introduces staircase effect, SOTV blurs the image edges, and EAD tends to over-smooth the image structure. The proposed TWSO model performs well in all the cases, showing its effectiveness for image denoising.

We now evaluate PM, CED, TV, SOTV, TGV, EAD and TWSO on the Berkeley database BSDS500 for image denoising. 100 images are randomly selected from the database and each of the chosen images is corrupted by Gaussian noise with zero-mean and 5 variances ranging from 0.005 to 0.025 at 0.005 interval. The performance of each method for each noise variance is measured in terms of mean and standard derivation of PSNR and SSIM over all the 100 images. The final quantitative results are shown in Table 5 and Fig. 12. The mean values of PSNR and SSIM obtained by the TWSO remain the largest in all the cases. The standard derivation of PSNR and SSIM are smaller and relatively stable, indicating that the proposed TWSO is robust against the increasing level of noise and performs the best among all the 7 methods compared for image denoising.

In addition to demonstrating TWSO for Gaussian noise removal, Fig. 13 and Table 6 show the denoising results of images with salt-and-pepper noise by different methods. Here, we use $p = 1$ in (2.3)

and Γ in (2.3) is the same as the domain of the noise in the image. We compare our TWSO method with another two state-of-the-art methods (i.e. media filter and TVL1 [41]) that have been widely employed in salt-and-pepper noise removal. From the first and second row of Table 6, it is clear that all three methods perform very well when the noise level is low. As the noise level increases, the media filter and TVL1 become increasingly worse, introducing more and more artefacts to the resulting images, and both fail in restoring the image when the noise level reaches 90%. In contrast, TWSO has produced visually more pleasing results against increasing noise and its PSNR and SSIM remain the highest in all the cases, as the introduction of tensor in TWSO offers richer neighbourhood information.

6. Conclusion

In this paper, we present the TWSO model for image processing which introduces an anisotropic diffusion tensor to the SOTV model. Specifically, the model integrates a novel regulariser to the SOTV model that uses the Frobenius norm of the product of the SOTV Hessian matrix and the anisotropic tensor. The advantage of the TWSO model includes its ability to reduce both the staircase
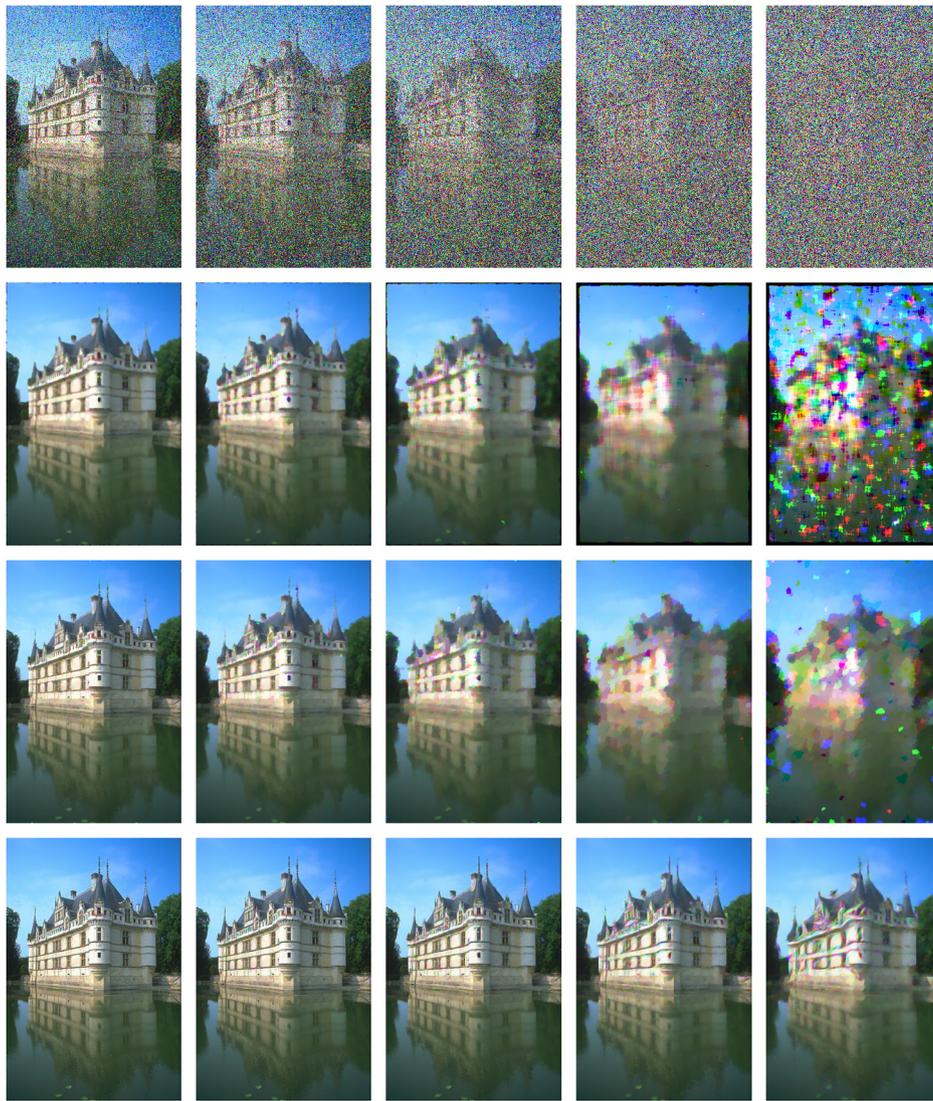
Fig. 13. Denoising results on a real colour image from Berkeley database BSDS500. First row: Images (from top to bottom) corrupted by salt-and-pepper noise of 20%, 40%, 60%, 80% and 90%; Second row: Media filter results; Third row: TVL1 results; Last row: the new TWSO results.

Table 6
Comparison of PSNR and SSIM using different methods on Fig. 13 with different noise levels.

Noise density	PSNR value					SSIM value				
	20%	40%	60%	80%	90%	20%	40%	60%	80%	90%
Degraded	11.9259	8.92250	7.16820	5.92240	5.41430	0.2490	0.1304	0.0714	0.0317	0.0164
Media filter	24.1623	22.6650	20.1166	16.8871	12.0975	0.8582	0.8146	0.7620	0.6639	0.3265
TVL1	27.9226	25.0812	23.1954	20.8760	17.7546	0.9405	0.8850	0.8328	0.7544	0.6427
TWSO	**59.6130**	**29.8168**	**27.7261**	**25.1963**	**23.3120**	**0.9985**	**0.9614**	**0.9373**	**0.8911**	**0.8427**

and blurring artefacts in the restored image. To avoid numerically solving the high order PDEs associated with the TWSO model, we develop a fast alternating direction method of multipliers based on a discrete finite difference scheme. Extensive numerical experiments demonstrate that the proposed TWSO model outperforms several state-of-the-art methods for image restoration for different applications. Future work will combine the capabilities of the TWSO model for image denoising and inpainting for medical imaging, such as optical coherence tomography [42].

References

[1] Maïtine Bergounioux, Loic Piffet, A second-order model for image denoising, Set-Valued Var. Anal. 18 (3–4) (2010) 277–306.

[2] Konstantinos Papafitsoros, Carola-Bibiane Schönlieb, A combined first and second order variational approach for image reconstruction, J. Math. Imaging Vis. 48 (2) (2014) 308–338.

[3] Wenqi Lu, Jinming Duan, Zhaowen Qiu, Zhenkuan Pan, Ryan Wen Liu, Li Bai, Implementation of high-order variational models made easy for image processing, Math. Methods Appl. Sci. 39 (14) (2016) 4208–4233.

[4] Jinming Duan, Zhaowen Qiu, Wenqi Lu, Guodong Wang, Zhenkuan Pan, Li Bai, An edge-weighted second order variational model for image decomposition, Digit. Signal Process. 49 (2016) 162–181.

[5] Jinming Duan, Wenqi Lu, Zhenkuan Pan, Li Bai, New second order Mumford–Shah model based on γ-convergence approximation for image processing, Infrared Phys. Technol. 76 (2016) 641–647.

[6] Jooyoung Hahn, Chang-Ock Lee, A nonlinear structure tensor with the diffusivity matrix composed of the image gradient, J. Math. Imaging Vis. 34 (2) (2009) 137–151.

[7] Kai Krajsek, Hanno Scharr, Diffusion filtering without parameter tuning: models and inference tools, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2536–2543.

[8] Markus Grasmair, Frank Lenzen, Anisotropic total variation filtering, Appl. Math. Optim. 62 (3) (2010) 323–339.

[9] Frank Lenzen, Florian Becker, Jan Lellmann, Stefania Petra, Christoph Schnörr, A class of quasi-variational inequalities for adaptive image denoising and decomposition, Comput. Optim. Appl. 54 (2) (2013) 371–398.

[10] Anastasios Roussos, Petros Maragos, Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals, in: 2010 17th IEEE International Conference on Image Processing (ICIP), IEEE, 2010, pp. 4141–4144.

[11] Freddie Åström, George Baravdish, Michael Felsberg, On tensor-based PDEs and their corresponding variational formulations with application to color image denoising, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 215–228.

[12] Freddie Åström, George Baravdish, Michael Felsberg, A tensor variational formulation of gradient energy total variation, in: Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, 2015, pp. 307–320.

[13] Stamatios Lefkimmiatis, Anastasios Roussos, Petros Maragos, Michael Unser, Structure tensor total variation, SIAM J. Imaging Sci. 8 (2) (2015) 1090–1122.

[14] Antonin Chambolle, Thomas Pock, A first-order primal–dual algorithm for convex problems with applications to imaging, J. Math. Imaging Vis. 40 (1) (2011) 120–145.

[15] Antonin Chambolle, Stacey E. Levine, Bradley J. Lucier, An upwind finite-difference method for total variation–based image smoothing, SIAM J. Imaging Sci. 4 (1) (2011) 277–299.

[16] Tom Goldstein, Min Li, Xiaoming Yuan, Ernie Esser, Richard Baraniuk, Adaptive primal–dual hybrid gradient methods for saddle-point problems, arXiv preprint, arXiv:1305.0546, 2013.

[17] Amir Beck, Marc Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM J. Imaging Sci. 2 (1) (2009) 183–202.

[18] Tom Goldstein, Christoph Studer, Richard Baraniuk, A field guide to forward–backward splitting with a FASTA implementation, arXiv preprint, arXiv:1411.3406, 2014.

[19] Chunlin Wu, Xue-Cheng Tai, Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models, SIAM J. Imaging Sci. 3 (3) (2010) 300–339.

[20] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.

[21] Xue-Cheng Tai, Jooyoung Hahn, Ginmo Jason Chung, A fast algorithm for Euler's elastica model using augmented Lagrangian method, SIAM J. Imaging Sci. 4 (1) (2011) 313–344.

[22] Konstantinos Papafitsoros, Carola Bibiane Schoenlieb, Bati Sengul, Combined first and second order total variation inpainting using split Bregman, Image Process. On Line 3 (2013) 112–136.

[23] Mirko Myllykoski, Roland Glowinski, T. Karkkainen, Tuomo Rossi, A new augmented Lagrangian approach for L^1-mean curvature image denoising, SIAM J. Imaging Sci. 8 (1) (2015) 95–125.

[24] Jinming Duan, Zhenkuan Pan, Baochang Zhang, Wanquan Liu, Xue-Cheng Tai, Fast algorithm for color texture image inpainting using the non-local CTV model, J. Glob. Optim. 62 (4) (2015) 853.

[25] Zheng Xu, Mário AT Figueiredo, Tom Goldstein, Adaptive ADMM with spectral penalty parameter selection, arXiv preprint, arXiv:1605.07246, 2016.

[26] Marius Lysaker, Arvid Lundervold, Xue-Cheng 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.

[27] Stamatios Lefkimmiatis, John Paul Ward, Michael Unser, Hessian Schatten-norm regularization for linear inverse problems, IEEE Trans. Image Process. 22 (5) (2013) 1873–1888.

[28] Carlos Brito-Loeza, Ke Chen, Victor Uc-Cetina, Image denoising using the Gaussian curvature of the image surface, Numer. Methods Partial Differ. Equ. 32 (3) (2015) 1066–1089.

[29] W. Zhu, X-C. Tai, T. Chan, Augmented Lagrangian method for a mean curvature based image denoising model, Inverse Probl. Imaging 7 (4) (2013) 1409.

[30] Kristian Bredies, Karl Kunisch, Thomas Pock, Total generalized variation, SIAM J. Imaging Sci. 3 (3) (2010) 492–526.

[31] Jinming Duan, Wenqi Lu, Christopher Tench, Irene Gottlob, Frank Proudlock, Niraj Nilesh Samani, Li Bai, Denoising optical coherence tomography using second order total generalized variation decomposition, Biomed. Signal Process. Control 24 (2016) 120–127.

[32] Joachim Weickert, Anisotropic Diffusion in Image Processing, vol. 1, Teubner, Stuttgart, 1998.

[33] Jooyoung Hahn, Chunlin Wu, Xue-Cheng Tai, Augmented Lagrangian method for generalized TV-Stokes model, J. Sci. Comput. 50 (2) (2012) 235–264.

[34] Michael Hintermüller, Carlos N. Rautenberg, Jooyoung Hahn, Functional-analytic and numerical issues in splitting methods for total variation-based image reconstruction, Inverse Probl. 30 (5) (2014) 055014.

[35] Roland Glowinski, Tsorng-Whay Pan, Xue-Cheng Tai, Some facts about operator-splitting and alternating direction methods, in: Splitting Methods in Communication, Imaging, Science, and Engineering, Springer, 2016, pp. 19–94.

[36] Jianhong Shen, Tony F. Chan, Mathematical models for local nontexture inpaintings, SIAM J. Appl. Math. 62 (3) (2002) 1019–1043.

[37] Carola-Bibiane Schönlieb, Partial Differential Equation Methods for Image Inpainting, vol. 29, Cambridge University Press, 2015.

[38] Jianhong Shen, Sung Ha Kang, Tony F. Chan, Euler's elastica and curvature-based inpainting, SIAM J. Appl. Math. 63 (2) (2003) 564–592.

[39] Pietro Perona, Jitendra Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Mach. Intell. 12 (7) (1990) 629–639.

[40] Leonid I. Rudin, Stanley Osher, Emad Fatemi, Nonlinear total variation based noise removal algorithms, Phys. D, Nonlinear Phenom. 60 (1) (1992) 259–268.

[41] Mila Nikolova, A variational approach to remove outliers and impulse noise, J. Math. Imaging Vis. 20 (1–2) (2004) 99–120.

[42] Jinming Duan, Christopher Tench, Irene Gottlob, Frank Proudlock, Li Bai, New variational image decomposition model for simultaneously denoising and segmenting optical coherence tomography images, Phys. Med. Biol. 60 (22) (2015) 8901.

Jinming Duan received his BSc degree from the Nanjing University of Information Science and Technology, PR China, in 2011, and the MSc degree from the Qingdao University, PR China, in 2014. He is currently a PhD student with the School of Computer Science at the University of Nottingham, UK. His research interests include variational image restoration, multiphase image segmentation, implicit surface reconstruction and medical image analysis.

Wil O.C. Ward received his MSci in Mathematics and Computer Science at the University of Nottingham, UK, and is currently finishing his PhD with the School of Computer Science there, in collaboration with the British Geological Survey. His research topics cover recursive Bayesian filtering and uncertainty analysis with applications to near-surface geophysics and parameter estimation in hydrogeological modelling. Other research interests include medical imaging and clustering methods.

Luke Sibbett received his MSci degree in Physics from the University of Nottingham in 2015. He is currently working towards an interdisciplinary PhD with at the School of Computer Science at the University of Nottingham and the British Geological Survey. His research interests include the regularisation of the nonlinear geophysical inverse problems and the use of automated analysis techniques for geophysical monitoring.

Zhenkuan Pan received his BSc degree in Engineering Mechanics from the Northwestern Polytechnical University, PR China, in 1987, and his PhD degree in Engineering Mechanics from Shanghai Jiao Tong University, PR China, in 1992. He is currently a professor of the College of Computer Science and Technology at the Qingdao University, China. From 2005 to 2006, he was a senior visiting scholar with the Department of Mathematics, the University of California, Los Angeles. His research interests include dynamics and optimisation of multibody systems, medical simulation and image processing.

Li Bai is with the School of Computer Science, University of Nottingham. She has a BSc and MSc in Mathematics, and a PhD in Computer Science from the University of Nottingham, UK. Her main research interests are in the area of computer vision, pattern recognition and medical image analysis.