



Image Reconstruction by Minimizing Curvatures on Image Surface

Qiuxiang Zhong¹ · Ke Yin² · Yuping Duan¹

Received: 8 January 2020 / Accepted: 31 August 2020 / Published online: 15 October 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

The curvature regularities are well-known for providing strong priors in the continuity of edges, which have been applied to a wide range of applications in image processing and computer vision. However, these models are usually non-convex, non-smooth, and highly nonlinear, the first-order optimality condition of which are high-order partial differential equations. Thus, numerical computation is extremely challenging. In this paper, we estimate the discrete mean curvature and Gaussian curvature on the local 3×3 stencil, based on the fundamental forms in differential geometry. By minimizing certain functions of curvatures over the image surface, it yields a kind of weighted image surface minimization problem, which can be efficiently solved by the alternating direction method of multipliers. Numerical experiments on image restoration and inpainting are implemented to demonstrate the effectiveness and superiority of the proposed curvature-based model compared to state-of-the-art variational approaches.

Keywords Image reconstruction · Image surface · Differential geometry · Curvature regularization · Mean curvature · Gaussian curvature

1 Introduction

Curves and surfaces are important geometric elements in image processing and analysis, which can be ideally measured by quantities such as arc length, area and curvatures [12]. Let $u : \Omega \rightarrow \mathbb{R}$ be an image defined on an open-bounded subset $\Omega \subset \mathbb{R}^2$ with Lipschitz continuous boundary. For each gray level λ , by taking the length energy as the curve model, it yields the well-known total variation (TV) regularization [42]

$$\text{TV}(u) = \int_{-\infty}^{\infty} \text{Length}(\Gamma_{\lambda}) d\lambda = \int_{\Omega} |\nabla u| dx, \quad (1)$$

where $\Gamma_{\lambda} = \{x \in \Omega | u(x) = \lambda\}$. Curvatures can depict the amount of a curve from being straight as in the case of a line or a surface deviating from being a flat plane, which have also been applied to various image processing tasks [1,23,43]. Suppose we take the curvature curve model, then the image

model becomes

$$E(u) = \int_{\Omega} g(\kappa) |\nabla u| dx = \int_{\Omega} g \left(\left| \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) \right| \right) |\nabla u| dx, \quad (2)$$

where $g(\kappa) = 1 + \alpha \kappa^2$, $\alpha > 0$, yielding the Euler's elastica image model [36]. Nitzberg, Mumford and Shiota [39] observed that line energies such as Euler's elastica can be used as regularization for the completion of missing contours in images by providing strong priors for the continuity of edges. Since then, Euler's elastica has been successfully used for image denoising [47,56], inpainting [35,45,54], segmentation [3,18,61], segmentation with depth [22,59] and illusory contour [28].

However, the numerical minimization of Euler's elastica is highly challenging due to its non-smoothness, nonlinearity and non-convexity. The gradient flow was used to solve a set of coupled second-order partial differential equations in [4,45] for minimizing the Euler's elastica energy, which usually takes high computational cost in imaging applications. Schoenemann, Kahl and Cremers [43] solved the associated linear programming relaxation and thresholded the solution to approximate the original integer linear program regarding curvature regularization. Discrete algorithms based on graph cuts methods have been studied for Euler's elastica models in [2,21]. Thanks to the development

✉ Yuping Duan
yuping.duan@tju.edu.cn

¹ Center for Applied Mathematics, Tianjin University, Tianjin 300072, China

² Center for Mathematical Sciences, Huazhong Univeristy of Science and Technology, Wuhan, Hubei 430070, China

of operator splitting technique and augmented Lagrangian algorithm, fast solvers for Euler’s elastica models have been presented in [3,19,47,52,54]. Recently, Deng, Glowinski and Tai [15] proposed a Lie operator-splitting-based time discretization scheme, which is applied to the initial value problem associated with the optimality system. A convex, lower semi-continuous, coercive approximation of Euler’s elastica energy via functional lifting was discussed in [8]. Later, Chambolle and Pock [11] used a lifted convex representation of curvature depending variational energies in the roto-translational space and yielded a natural generalization of the total variation to the roto-translational space.

By considering the *image surface* or *graph* in a high-dimensional space, the image denoising problem turns to find a piecewise smooth surface to approximate the given image surface. Both *mean curvature* and *Gaussian curvature* can be used as the regularizer to preserve geometric features of the image surface for different image processing tasks. The mean curvature was first introduced for noise removal problems as the mean curvature-driven diffusion algorithms [20,55], which evolved the image surface at a speed proportional to the mean curvatures. Zhu and Chan [58] proposed to employ the L^1 -norm of mean curvature over the image surface as follows

$$H(u) = \int_{\Omega} \left| \nabla \cdot \left(\frac{\nabla u}{\sqrt{1 + |\nabla u|^2}} \right) \right| dx, \tag{3}$$

which was proven to be able to keep corners of objects and greyscale intensity contrasts of images and remove the staircase effect. Originally, the smoothed mean curvature regularized model was numerically solved by the gradient descent method, which involves high-order derivatives and converges slowly in practice. To deal with this difficulty, some effective and efficient numerical algorithms for mean curvature regularized models were proposed based on the augmented Lagrangian method [37,60]. However, there exist some inevitable problems in this kind of method, such as the choices of the algorithm parameters and the slow convergence rate.

Gaussian curvature-driven diffusion was studied in [30] for noise removal problems, which was shown superior in preserving image structures and details. Lu, Wang and Lin [32] proposed an energy functional based on Gaussian curvature for image smoothing, which was solved by a diffusion process. Gong and Sbalzarini [25] presented a variational model with local weighted Gaussian curvature as the regularizer, which was solved by the splitting techniques. In [9], the authors minimized the following L^1 -norm of Gaussian curvature of the image surface

$$K(u) = \int_{\Omega} \frac{|\det \nabla^2 u|}{(1 + |\nabla u|^2)^2} dx, \tag{4}$$

with $\nabla^2 u$ being the Hessian of function u and

$$\det \nabla^2 u = \frac{\partial^2 u}{\partial x^2} \frac{\partial^2 u}{\partial y^2} - \left| \frac{\partial^2 u}{\partial x \partial y} \right|^2.$$

Similarly, Gaussian curvature regularizer can also preserve image contrast, edges and corners very well. However, the minimization of (4) is even harder due to the determinant of Hessian, which was solved by a two-step method based on the vector field smoothing and gray-level interpolation.

Recently, efficient methods are proposed to directly estimate curvatures in the discrete setting. Ciomaga, Monasse and Morel [13] evaluated the curvatures on the level lines of the bilinearly interpolated images, where a smoothing step depending on affine erosion of the level lines is implemented on the images to remove pixelization artifacts. Gong and Sbalzarini [26] presented a filter-based approach to use the pixel-local analytical solutions to approximate the TV, mean curvature and Gaussian curvature by enumerating the constant, linear and developable surfaces in the discrete 3×3 neighborhood. Although the curvature filter avoids solving the high-order partial differential equations associated with the curvature-based variational models, it lacks rigorous definition and accurate estimation of the curvatures. Zhong *et al.* [57] introduced the total curvature regularization by minimizing all the normal curvatures over image surface, in which the normal curvatures are well-defined in a pixel-wise local neighborhood.

1.1 Our Contributions

Assume an image surface $(x, y, u(x, y))$ is defined on Ω . We define a level set function $\phi(x, y, z) = u(x, y) - z$, the zero level set of which corresponds to the surface $z = u(x, y)$. By integrating over Ω , it gives the surface area energy as follows

$$A(u) = \int_{\Omega} \sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1} dx = \int_{\Omega} \sqrt{1 + |\nabla u|^2} dx, \tag{5}$$

which has been used as the efficient regularization for image smoothing [46,55] and blind deblurring [31], etc. We formulate a new curvature regularized model by minimizing either the mean curvature or Gaussian curvature over image surfaces. Taking image denoising as an example, we define the curvature energy as follows

$$\min_u \int_{\Omega} g(\kappa) \sqrt{1 + |\nabla u|^2} dx + \frac{\lambda}{2} \int_{\Omega} (u - f)^2 dx, \tag{6}$$

where κ denotes either mean curvature or Gaussian curvature and $g(\cdot)$ is a certain function of the curvature. According to [11], the following three typical energies are considered and

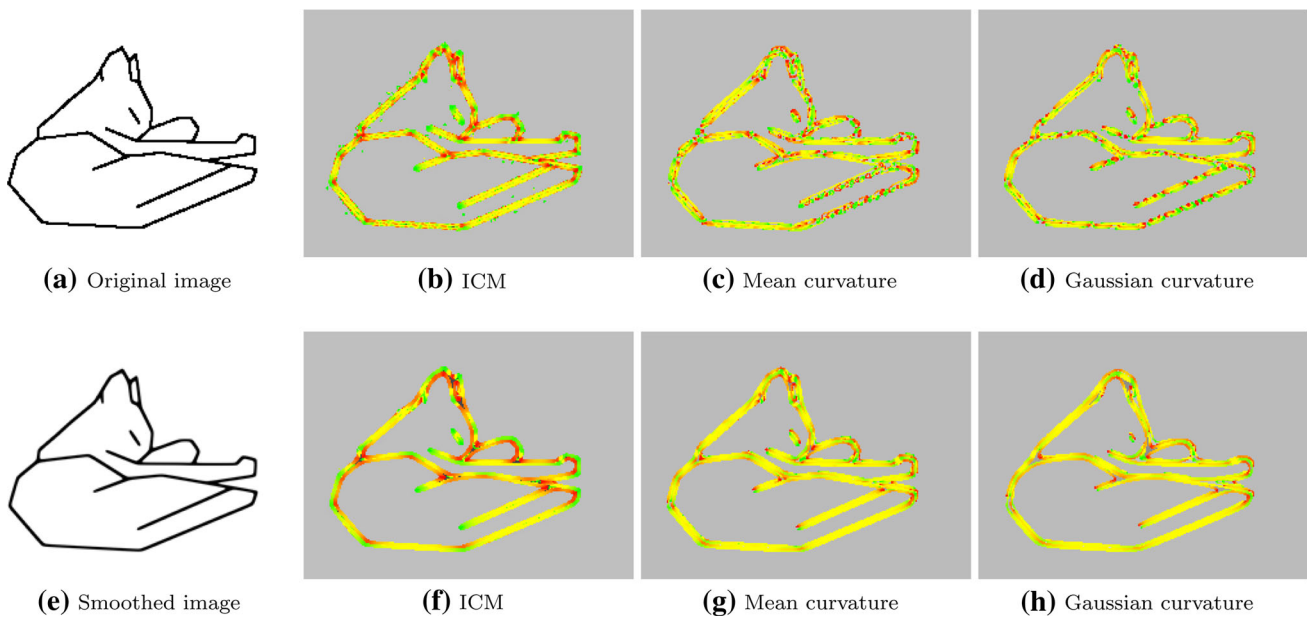


Fig. 1 The estimated curvature maps of a simple image ‘Cat’ estimated by image curvature microscope (ICM) [13] and our approach, where yellow denotes zero curvature, and green and red represent positive and negative curvatures, respectively (Color figure online)

extended to the graph, where α is a positive parameter to balance the curvature and surface area.

- (1) Total absolute curvature (TAC): measures the sum of the surface area and absolute curvature

$$g_1(\kappa(u)) = 1 + \alpha|\kappa(u)|. \quad (7)$$

- (2) Total square curvature (TSC): penalizes the surface area and the squared curvature

$$g_2(\kappa(u)) = 1 + \alpha|\kappa^2(u)|. \quad (8)$$

- (3) Total roto-translational variation (TRV): measures the surface area and curvature through an Euclidean metric

$$g_3(\kappa(u)) = \sqrt{1 + \alpha|\kappa^2(u)|}. \quad (9)$$

Both mean curvature and Gaussian curvature are estimated based on *normal curvatures*, which can be defined on a 3×3 neighborhood of each pixel followed our previous work [57]. As long as the curvatures being computed explicitly, we can use the operator splitting technique and ADMM-based algorithm to iteratively solve the curvature minimization problem (6). We prove the existence of a solution and discuss the convergence of the ADMM algorithm under certain assumptions. Numerical applications to image denoising and inpainting show the superiority of the proposed method, which can provide better restoration qualities with lower computational cost compared to several state-of-

the-art variational models. To sum up, our method presents the following advantages:

- (1) We develop a new discrete method to estimate both mean curvature and Gaussian curvature on image surfaces. Different from the discrete curvature in [13], which is designed for 2D level lines and strongly relies on the smoothness of the level lines, our approach makes use of 3D level sets information. As shown in Fig. 1, the curvature maps are different on image edges due to the intrinsic differences among the curvatures, and our method performs better on the original image by producing fewer outliers.
- (2) By computing the normal curvatures in the 3×3 neighborhood, we can estimate both mean curvature and Gaussian curvature in terms of principal/normal curvatures. Thus, our model can be regarded as a re-weighted minimal surface model, where the weights are automatically updated using curvature information. It can not only achieve good image restoration results but also preserve the geometric properties, such as edges, corners, etc., very well.
- (3) Because we only introduce one artificial variable, our ADMM has fewer parameters than other curvature-based models. More specifically, our algorithm has one parameter of the augmented term while the ADMMs for Euler’s elastica model in [47] and mean curvature model in [60] have three and four such kinds of parameters, respectively.

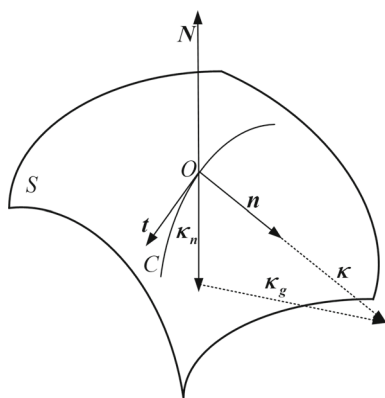


Fig. 2 Curvature vector

- (4) Our formulation is more flexible to adapt to the different combinations of the function-type and curvature-type without affecting the way of the operator-splitting and the associated ADMM-based algorithm. The proposed curvature regularization can be easily applied to other image processing tasks such as segmentation [18,33], reconstruction [52,56], etc.

1.2 Organizations

This paper is organized as follows. We introduce the estimation of curvatures based on differential geometry theory in Sect. 2. The ADMM-based algorithm and convergence analysis are discussed in Sect. 3. Section 4 is dedicated to numerical experiments on image reconstruction problem to demonstrate the efficiency and superiority of the proposed approach. Finally, we draw some conclusions in Sect. 5.

To summarize this section, we would like to mention that the proper functional frameworks to formulate the curvature-based variational models involving aforementioned curvature regularizations (2)–(4) have not been identified yet. Therefore, we do not know much about the function space of our model (6), which has to be a subspace of $L^2(\Omega)$. Obviously, the discrete problems largely ignore these functional analysis considerations. Thus, we discuss our model under the discrete setting in the followings.

2 Curvature Estimation

2.1 Notations and Definitions

Let $\mathbf{r} = \mathbf{r}(x, y) : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ be a regular parametric surface S and (x, y) be the coordinates on Ω . In order to quantify the curvature of a surface S , we consider a curve C on S passing through point O shown in Fig. 2. The *curvature vector* is used to measure the rate of change of the tangent

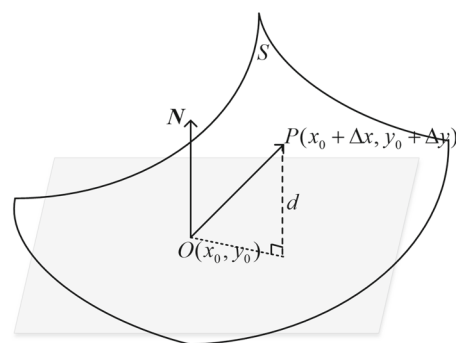


Fig. 3 The projection distance of a neighboring point to the tangent plane

along the curve, which can be defined using the *unit tangent* vector \mathbf{t} and the *unit normal* vector \mathbf{n} of the curve C at point O as

$$\boldsymbol{\kappa} = \frac{d\mathbf{t}}{ds} = \boldsymbol{\kappa}_n + \boldsymbol{\kappa}_g$$

with $\boldsymbol{\kappa}_n$ being the *normal curvature vector* and $\boldsymbol{\kappa}_g$ being the *geodesic curvature vector*. Let N be the *surface unit normal vector*, which is defined as

$$N = \frac{\mathbf{r}_x \times \mathbf{r}_y}{|\mathbf{r}_x \times \mathbf{r}_y|}$$

By differentiating $N \cdot \mathbf{t} = 0$ along the curve with respect to s , we obtain

$$\frac{d\mathbf{t}}{ds} \cdot N + \mathbf{t} \cdot \frac{dN}{ds} = 0.$$

Thus, the normal curvature of the surface at O in the direction \mathbf{t} can be expressed as

$$\begin{aligned} \kappa_n &= \frac{d\mathbf{t}}{ds} \cdot N = -\mathbf{t} \cdot \frac{dN}{ds} = -\frac{d\mathbf{r}}{ds} \cdot \frac{dN}{ds} \\ &= \frac{Ldx^2 + 2Mdx dy + Ndy^2}{Edx^2 + 2Fdx dy + Gdy^2}, \end{aligned} \tag{10}$$

which is the quotient of the second fundamental form and the first fundamental form. The denominator of (10) is the *first fundamental form*, which is the square of the arc length

$$I = ds^2 = d\mathbf{r} \cdot d\mathbf{r} = Edx^2 + 2Fdx dy + Gdy^2,$$

with the first fundamental form coefficients $E = \mathbf{r}_x \cdot \mathbf{r}_x$, $F = \mathbf{r}_x \cdot \mathbf{r}_y$, $G = \mathbf{r}_y \cdot \mathbf{r}_y$. On the other hand, the numerator of (10) is the *second fundamental form* such that

$$II = -d\mathbf{r} \cdot dN = d^2\mathbf{r} \cdot N = Ldx^2 + 2Mdx dy + Ndy^2,$$

Table 1 List of notations used in the this work

Notations	Description
\mathbf{r}	Regular parametric surface
$\boldsymbol{\kappa}$	Curvature vector
\mathbf{t}	Unit tangent vector
s	Arc length element
\mathbf{n}	Unit normal vector of curve C
\mathbf{N}	Surface unit normal vector
$\boldsymbol{\kappa}_n$	Normal curvature vector
$\boldsymbol{\kappa}_g$	Geodesic curvature vector
I	The first fundamental form
II	The second fundamental form
d	Projection distance of a point to the tangent plane
l	Length of the parametric curve
κ_n	Normal curvature
$\kappa_{\max}, \kappa_{\min}$	Principal curvatures
H	Mean curvature
K	Gaussian curvature

with the second fundamental form coefficients $L = \mathbf{r}_{xx} \cdot \mathbf{N}$, $M = \mathbf{r}_{xy} \cdot \mathbf{N}$, $N = \mathbf{r}_{yy} \cdot \mathbf{N}$.

Proposition 2.1 Suppose $\mathcal{S}: \mathbf{r} = \mathbf{r}(x, y)$ is a regular parametric surface, $O(x_0, y_0)$ being an arbitrary point on \mathcal{S} , then the second fundamental form at point O can be estimated by

$$II \approx 2d, \tag{11}$$

where d denotes the projection distance of its neighboring point $P(x_0 + \Delta x, y_0 + \Delta y)$ to the tangent plane of O .

Proof As shown in Fig. 3, the projection distance of the neighboring point $P(x_0 + \Delta x, y_0 + \Delta y)$ to the tangent plane is obtained as follows

$$d(\Delta x, \Delta y) = (\mathbf{r}(x_0 + \Delta x, y_0 + \Delta y) - \mathbf{r}(x_0, y_0)) \cdot \mathbf{N}.$$

By Taylor’s expansion, we have

$$\begin{aligned} &\mathbf{r}(x_0 + \Delta x, y_0 + \Delta y) - \mathbf{r}(x_0, y_0) \\ &= (\mathbf{r}_x \Delta x + \mathbf{r}_y \Delta y) + \frac{1}{2}(\mathbf{r}_{xx}(\Delta x)^2 + 2\mathbf{r}_{xy} \Delta x \Delta y \\ &\quad + \mathbf{r}_{yy}(\Delta y)^2) + \mathcal{o}((\Delta x)^2 + (\Delta y)^2), \end{aligned}$$

and

$$\lim_{(\Delta x)^2 + (\Delta y)^2 \rightarrow 0} \frac{\mathcal{o}((\Delta x)^2 + (\Delta y)^2)}{(\Delta x)^2 + (\Delta y)^2} = 0.$$

Owing to $\mathbf{r}_x \cdot \mathbf{N} = \mathbf{r}_y \cdot \mathbf{N} = 0$, it follows that

$$\begin{aligned} d(\Delta x, \Delta y) &= \frac{1}{2}[L(\Delta x)^2 + 2M\Delta x \Delta y + N(\Delta y)^2] \\ &\quad + \mathcal{o}((\Delta x)^2 + (\Delta y)^2), \end{aligned}$$

the first term on the right-hand side of which gives the *second fundamental form*. Thus, as long as $\sqrt{(\Delta x)^2 + (\Delta y)^2} \rightarrow 0$, there is

$$II \approx 2d(\Delta x, \Delta y),$$

which completes the proof. ■

With the normal curvatures, we can further define the two *principal curvatures* of point O to measure how the surface bends by different amounts in different directions at that point, which are

$$\begin{aligned} \kappa_{\max} &:= \text{the maximum normal curvature at } O, \\ \kappa_{\min} &:= \text{the minimum normal curvature at } O. \end{aligned}$$

Now we are ready to define the *mean curvature* and *Gaussian curvature* based on the principal curvatures as follows.

Definition 2.2 The mean curvature and Gaussian curvature of \mathcal{S} at point O are defined as

$$H = \frac{1}{2}(\kappa_{\max} + \kappa_{\min}), \text{ and } K = \kappa_{\max} \kappa_{\min}. \tag{12}$$

The Gaussian curvature is also known as the *curvature* of a surface, which is an intrinsic measure of the curvature, depending only on distances that measured on the surface, not on the way that isometrically embedded in Euclidean space. Although the mean curvature is not intrinsic, a surface with zero mean curvature at all points is called the *minimal surface*, which is widely used in shape analysis, high-dimensional data processing, etc. For convenience, the important notations used in this paper are summarized in Table 1.

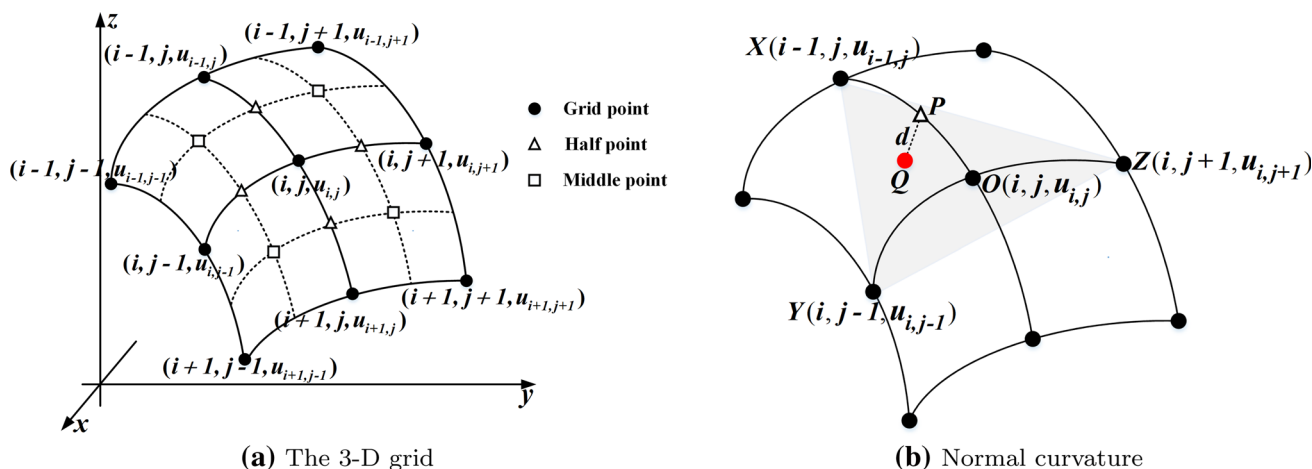


Fig. 4 Calculate the normal curvature on image surface, the gray triangle in which represents a tangent plane of center point O

2.2 Calculation of Normal Curvatures

Without loss of generality, we represent a gray image as a $m \times n$ matrix and the grid $\Omega = \{(ih, jh) : 1 \leq i \leq m, 1 \leq j \leq n\}$ with h being the grid step size fixed as the same value in the x -axis and the y -axis. In the following, we use the notation $u(i, j) = u(ih, jh)$ and (i, j) denotes a discrete point. Similarly to Euler’s elastica energies [11,45,47], we use the staggered grid in the $x - y$ plane, where an example with 3×3 pixels is displayed in Fig. 4a. The Δ -nodes express half grids and \square -nodes represent middle points, which are used as the neighboring points in the calculation of normal curvatures. The intensity values on Δ -nodes are estimated as the mean of its two neighboring \bullet -nodes, while on \square -nodes are estimated as the mean of the four surrounding \bullet -nodes.

Now we discuss how to estimate the discrete normal curvatures locally. As shown in Fig. 4b, we use the triangular plane T_{XYZ} to approximate tangent plane of point O . Since the 3D coordinates of point X, Y and Z are known, the normal vector N of T_{XYZ} can be decided by the cross-product of the vector \vec{XY} and \vec{XZ} as follows

$$N = \vec{XY} \times \vec{XZ} = (2u_{i,j} - u_{i,j-1} - u_{i,j+1}, u_{i,j-1} - u_{i,j+1}, 2).$$

In order to implement Proposition 2.1 to approximate the second fundamental form, we use the half point $P(i - \frac{1}{2}, j, u_{i-\frac{1}{2},j})$ in Fig. 4b for estimating the projection distance d of P to its tangent plane T_{XYZ} , which gives

$$d = \text{dist}(P, T_{XYZ}) = \vec{QP} \cdot N = \frac{2u_{i,j} - u_{i,j-1} - u_{i,j+1}}{2\sqrt{(2u_{i-1,j} - u_{i,j-1} - u_{i,j+1})^2 + (u_{i,j-1} - u_{i,j+1})^2 + 4}} \tag{13}$$

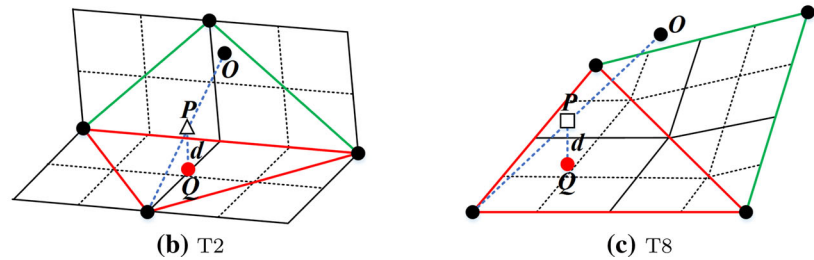
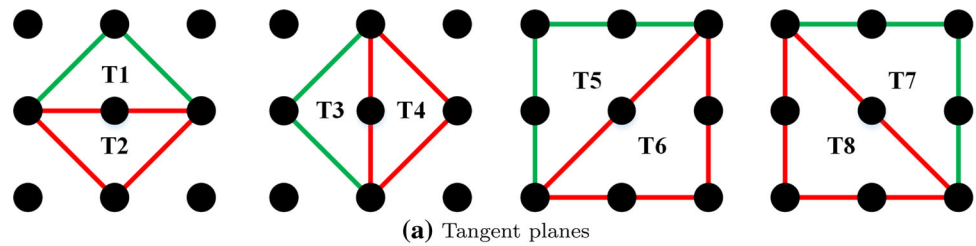
On the other hand, the first fundamental form can be estimated using the arc length \widehat{OP} , which is approximated by the length of the line segment, i.e., $|OP|$. Based on the Pythagorean theorem, we have

$$ds = \widehat{OP} \approx \sqrt{(u_{i-\frac{1}{2},j} - u_{i,j})^2 + h^2}, \tag{14}$$

where $u_{i-\frac{1}{2},j}$ can be estimated as the mean of its two neighboring points $u_{i-1,j}$ and $u_{i,j}$. Therefore, according to formula (10), the normal curvature of point O in direction \vec{OP} can be expressed as follows

$$\kappa_n \approx \frac{2d}{ds^2} \approx \frac{2u_{i,j} - u_{i,j-1} - u_{i,j+1}}{((u_{i-\frac{1}{2},j} - u_{i,j})^2 + h^2)\sqrt{(2u_{i-1,j} - u_{i,j-1} - u_{i,j+1})^2 + (u_{i,j-1} - u_{i,j+1})^2 + 4}} \tag{15}$$

Fig. 5 The eight tangent planes of the center point in a 3×3 local window and the representatives of the two different kinds of projections



Proposition 2.3 *The approximation of the normal curvature by (15) is $o(h^2)$.*

Proof As shown in Fig. 4b, $P(i - \frac{1}{2}, j, \frac{u_{i-1,j} + u_{i,j}}{2})$ is a neighboring point of $O(i, j, u_{i,j})$ used to calculate the normal curvature. Obviously, when $h \rightarrow 0$, we obtain $\lim_{h \rightarrow 0} \frac{OP}{|OP|} = 1$. As a result, the approximate error of first fundamental form using the square of $|OP|$ is $o(h^2)$. On the other hand, according to Proposition 2.1, the approximate error of second fundamental form using the projection distance d is also $o(h^2)$. Thus, the estimation of the normal curvature by (15) is a second-order approximation. ■

2.3 Mean Curvature and Gaussian Curvature

In order to estimate the mean and Gaussian curvatures at each point of the image surface, we compute the normal curvatures along eight directions in the local 3×3 window. As shown in Fig. 5a, eight triangular planes (i.e., T1-T8) are defined as the approximate tangent planes, which locate physically nearest to the center point O and pairwise centrosymmetric to avoid the grid bias. We can simply divide these tangent planes into two categories such that T1-T4 use the half points and T5-T8 use the middle points to compute the projections. The examples of each type (i.e., T2 and T8) are also presented in Fig. 5b, c, which illustrate the tangent planes and the corresponding projections.

Given the eight neighboring points being either the half points or the middle points, we can estimate the second fundamental form by calculating the distance from the neighboring point P as depicted in Fig. 5b, c to the corresponding tangent plane. More specifically, we calculate $d_\ell = \text{dist}(P_\ell, T_\ell)$, $\ell = 1, \dots, 8$, which are defined as

$$\begin{aligned}
 d_1 &= \frac{2u_{i,j} - u_{i,j-1} - u_{i,j+1}}{2\sqrt{(2u_{i-1,j} - u_{i,j-1} - u_{i,j+1})^2 + (u_{i,j-1} - u_{i,j+1})^2 + 4}}; \\
 d_2 &= \frac{u_{i,j-1} + u_{i,j+1} - 2u_{i,j}}{2\sqrt{(2u_{i+1,j} - u_{i,j-1} - u_{i,j+1})^2 + (u_{i,j+1} - u_{i,j-1})^2 + 4}}; \\
 d_3 &= \frac{u_{i-1,j} + u_{i+1,j} - 2u_{i,j}}{2\sqrt{(u_{i+1,j} - u_{i-1,j})^2 + (u_{i-1,j} + u_{i+1,j} - 2u_{i,j-1})^2 + 4}}; \\
 d_4 &= \frac{2u_{i,j} - u_{i-1,j} - u_{i+1,j}}{2\sqrt{(u_{i-1,j} - u_{i+1,j})^2 + (u_{i-1,j} + u_{i+1,j} - 2u_{i,j+1})^2 + 4}}; \\
 d_5 &= \frac{u_{i-1,j-1} + u_{i-1,j+1} + u_{i+1,j-1} - u_{i,j} - u_{i-1,j} - u_{i,j-1}}{2\sqrt{(u_{i+1,j-1} - u_{i-1,j-1})^2 + (u_{i-1,j+1} - u_{i-1,j-1})^2 + 4}}; \\
 d_6 &= \frac{u_{i,j} + u_{i+1,j} + u_{i,j+1} - u_{i+1,j+1} - u_{i-1,j+1} - u_{i+1,j-1}}{2\sqrt{(u_{i-1,j+1} - u_{i+1,j+1})^2 + (u_{i+1,j-1} - u_{i+1,j+1})^2 + 4}}; \\
 d_7 &= \frac{u_{i,j} + u_{i-1,j} + u_{i,j+1} - u_{i-1,j-1} - u_{i+1,j+1} - u_{i-1,j+1}}{2\sqrt{(u_{i-1,j+1} - u_{i+1,j+1})^2 + (u_{i-1,j-1} - u_{i-1,j+1})^2 + 4}}; \\
 d_8 &= \frac{u_{i-1,j-1} + u_{i+1,j+1} + u_{i+1,j-1} - u_{i,j} - u_{i+1,j} - u_{i,j-1}}{2\sqrt{(u_{i+1,j-1} - u_{i-1,j-1})^2 + (u_{i+1,j+1} - u_{i+1,j-1})^2 + 4}}.
 \end{aligned}$$

On the other hand, we estimate the arc length of the central point $O(i, j, u_{i,j})$ to the neighboring points P_ℓ , $\ell = 1, \dots, 8$, in the local 3×3 neighborhood, which is computed by the square root of the quadratic sum of the intensity differences and the grid size between two points according to (14). As a result, eight normal curvatures can be calculated using (15) as

$$\kappa_\ell \approx \begin{cases} \frac{2d_\ell}{(u_\ell^\Delta - u_{i,j})^2 + h^2}, & \ell = 1, 2, 3, 4, \\ \frac{2d_\ell}{(u_\ell^\square - u_{i,j})^2 + 2h^2}, & \ell = 5, 6, 7, 8, \end{cases} \quad (16)$$

where u_ℓ^Δ and u_ℓ^\square denote the intensities of the two types of neighboring points as shown in Fig. 4a. The normal curvatures are well-defined and applicable to every point on the image surface.

With all normal curvatures, we are ready to obtain the two principal curvatures κ_{\max} and κ_{\min} from

$$\kappa_{\max} = \max\{\kappa_\ell\}, \quad \kappa_{\min} = \min\{\kappa_\ell\}, \quad \text{for } \ell = 1, 2, \dots, 8. \tag{17}$$

According to Definition 2.2, both mean and Gaussian curvatures are evaluated on each point of the image surface, which only involve simple arithmetic operations.

3 The Curvature-Based Variational Model and Numerical Algorithm

3.1 ADMM-Based Numerical Algorithm

Now we discuss the numerical algorithm for solving the minimization problem (6) by firstly rewriting it into the following discrete form

$$\min_u \sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |\nabla u_{i,j}|^2} + \frac{\lambda}{2}\|u - f\|^2, \tag{18}$$

where $\kappa(u_{i,j})$ denotes either the mean curvature or Gaussian curvature, $|\cdot|$ is the usual Euclidean norm in \mathbb{R}^2 and $\|\cdot\|$ is the l^2 norm. Note that all matrix multiplication and divisions in this paper are element-wise. The discrete gradient operator $\nabla : \mathbb{R}^{mn} \rightarrow \mathbb{R}^{mn \times mn}$ is defined by

$$(\nabla u)_{i,j} = ((\nabla u)_{i,j}^x, (\nabla u)_{i,j}^y) \quad \text{for } i=1, \dots, m, \quad j=1, \dots, n$$

with

$$(\nabla u_{i,j})^x = \begin{cases} u_{i+1,j} - u_{i,j}, & \text{if } 1 \leq i < m, \\ u_{1,j} - u_{i,j}, & \text{if } i = m \end{cases}, \quad \text{and}$$

$$(\nabla u_{i,j})^y = \begin{cases} u_{i,j+1} - u_{i,j}, & \text{if } 1 \leq j < n, \\ u_{i,1} - u_{i,j}, & \text{if } j = n \end{cases}.$$

As can be seen, when the mean and Gaussian curvatures are given, the curvature regularized model (18) becomes a weighted minimal surface model, for which fast algorithms can be used as the numerical solver, for instance the split Bregman method [24], primal-dual splitting method [10] and augmented Lagrangian method [50]. Therefore, we solve the weighted minimal surface model based on the ADMM algorithm and then evaluate the curvatures separately using the most recent update of u in an iterative way. To be specific, we

introduce an auxiliary variable v and rewrite the minimization problem (6) into a constrained minimization as follows

$$\min_{u,v} \sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |v_{i,j}|^2} + \frac{\lambda}{2}\|u - f\|^2 \tag{19}$$

s.t., $v_{i,j} = \nabla u_{i,j}$.

Given some $(u^k, v^k) \in \mathbb{R}^{mn} \times \mathbb{R}^{mn \times mn}$, the proximal augmented Lagrangian of above-constrained minimization problem can be defined as

$$\mathcal{L}(u, v; \Lambda) = \sum_{i,j} g(\kappa(u_{i,j}^k))\sqrt{1 + |v_{i,j}|^2} + \frac{\lambda}{2}\|u - f\|^2$$

$$+ \langle \Lambda, v - \nabla u \rangle + \frac{\mu}{2}\|v - \nabla u\|^2$$

$$+ \frac{\tau}{2}\|u - u^k\|^2 + \frac{\sigma}{2}\|v - v^k\|^2,$$

where Λ represents the Lagrangian multiplier, and μ, τ, σ are positive parameters. Then, we iteratively and alternatively solve the u -subproblem and v -subproblem until reaching the terminating condition; see Algorithm 3.1.

Algorithm 3.1: ADMM-based algorithm for the curvature regularized model (18)

- 1: **Input:** Degraded image f , parameters $\alpha, \lambda, \mu, \tau, \sigma$, maximum iteration number I_{\max} and stopping threshold ϵ .
- 2: **Initialize:** $u^0 = f, v^0 = 0, \Lambda^0 = 0$.
- 3: **while** (not converged and $k \leq I_{\max}$) do

- (i) Compute v^{k+1} from:

$$v^{k+1} = \arg \min_v \left\{ \sum_{i,j} g(\kappa(u_{i,j}^k))\sqrt{1 + |v_{i,j}|^2} \right. \tag{20}$$

$$\left. + \frac{\mu}{2}\|v - \nabla u^k + \frac{\Lambda^k}{\mu}\|^2 + \frac{\sigma}{2}\|v - v^k\|^2 \right\};$$

- (ii) Compute u^{k+1} from:

$$u^{k+1} = \arg \min_u \left\{ \frac{\lambda}{2}\|u - f\|^2 + \frac{\mu}{2}\|\nabla u - v^{k+1} \right. \tag{21}$$

$$\left. - \frac{\Lambda^k}{\mu}\|^2 + \frac{\tau}{2}\|u - u^k\|^2 \right\};$$

- (iii) Update Λ^{k+1} from:

$$\Lambda^{k+1} = \Lambda^k + \mu(v^{k+1} - \nabla u^{k+1}); \tag{22}$$

- (iv) Evaluate κ (H or K) according to (12) using the latest update u^{k+1} ;

(v) Check convergence condition by the relative error defined using the l^1 norm:

$$\|u^{k+1} - u^k\|_1 \leq \epsilon \|u^k\|_1;$$

4: **end while**

5: **output:** Restored image u .

3.1.1 The v -Subproblem

The sub-minimization problem w.r.t. v is a typical nonlinear minimization problem, which can be solved by Newton’s method. Assume $v^{k+1,0} = v^k$, we compute

$$v^{k+1,\ell+1} = v^{k+1,\ell} - \frac{g(\kappa(u^k))v^{k+1,\ell}(1 + |v^{k+1,\ell}|^2)^{-\frac{1}{2}} + \mu(v^{k+1,\ell} - \nabla u^k) + \Lambda^k + \sigma(v^{k+1,\ell} - v^{k,\ell})}{g(\kappa(u^k))(1 + |v^{k+1,\ell}|^2)^{-\frac{3}{2}} + \mu + \sigma},$$

for $\ell = 0, 1, \dots, L - 1$. Then, the update of variable v is given as $v^{k+1} = v^{k+1,L}$.

3.1.2 The u -Subproblem

The first-order optimality condition of (21) gives us a linear equation as follows

$$((\lambda + \tau)\mathcal{I} - \mu\Delta)u^{k+1} = \lambda f + \tau u^k - \nabla \cdot (\mu v^{k+1} + \Lambda^k)$$

with \mathcal{I} being the identity matrix. Under the periodic boundary condition, we can solve the above equation by the fast Fourier Transform (FFT), i.e.,

$$u^{k+1} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\lambda f - \nabla \cdot (\mu v^{k+1} + \Lambda^k) + \tau u^k)}{(\lambda + \tau)\mathcal{I} - \mu\mathcal{F}(\Delta)}\right), \quad (23)$$

where \mathcal{F} and \mathcal{F}^{-1} denote the Fourier transform and inverse Fourier transform, respectively.

3.2 Convergence Analysis

In this subsection, we discuss the convergence of Algorithm 3.1. Firstly, we prove that a solution of our curvature-based regularization model (18) exists.

Lemma 3.1 *There exists a minimizer $u^* \in \mathbb{R}^{mn}$ for the discrete minimization problem (18).*

Proof By the definitions of g in (7), (8) and (9), $g(\kappa(u)) \geq 1$. According to Lemma 3.8 of [27], we have $\sum_{i,j} |\nabla u_{i,j}| + \frac{\lambda}{2} \|u -$

$f\|^2$ is coercive. Then, there is

$$\begin{aligned} & \sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |\nabla u_{i,j}|^2} + \frac{\lambda}{2} \|u - f\|^2 \\ & > \sum_{i,j} |\nabla u_{i,j}| + \frac{\lambda}{2} \|u - f\|^2 \end{aligned} \quad (24)$$

is also coercive. By definition of $\kappa = \{H, K\}$ as defined in (12) and continuity of the min/max functions, $\kappa = \kappa(\{\kappa_{\min}, \kappa_{\max}\})$ is continuous on $\{\kappa_\ell : \ell = 1, \dots, 8\}$. More-

over by (16), κ_ℓ ($\ell = 1, \dots, 8$) are continuous functions on u . Therefore, $\sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |\nabla u_{i,j}|^2} + \frac{\lambda}{2} \|u - f\|^2$ is continuous on u . Together with coercivity and continuity, we have that the discrete minimization problem (18) has a minimizer $u^* \in \mathbb{R}^{mn}$. ■

The convergence analysis of the proximal ADMM scheme has been widely studied [6,14,16,44]. We follow the work [54] to give some theoretical analysis of Algorithm 3.1. We first prove the monotonic decrease of the error sequence and the convergence of the residuals under certain assumption. Then, we show the iterative sequences generated by Algorithm 3.1 converges to a limit point that satisfies the first-order optimality conditions.

Theorem 3.2 *Let $\{(u^k, v^k; \Lambda^k)\}_{k \in \mathbb{N}}$ be the sequence generated by proposed ADMM-based Algorithm 3.1 and $(\bar{u}, \bar{v}; \bar{\Lambda})$ be a point satisfying the first-order optimality conditions*

$$\begin{cases} g(\kappa(u)) \frac{v}{\sqrt{1+|v|^2}} + \Lambda = 0, \\ \lambda(u - f) + \nabla \cdot \Lambda = 0, \\ v - \nabla u = 0. \end{cases} \quad (25)$$

Suppose $u_e^k = u^k - \bar{u}$, $v_e^k = v^k - \bar{v}$ and $\Lambda_e^k = \Lambda^k - \bar{\Lambda}$ denote the error sequence. If the quantity

$$\begin{aligned} \Delta^k & := \left\langle g(\kappa(u^{k-1})) \frac{v^k}{\sqrt{1+|v^k|^2}} - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}}, \right. \\ & \left. v^k - \bar{v} \right\rangle \geq 0, \quad \forall k \in \mathbb{N}, \end{aligned} \quad (26)$$

then the followings hold: (1) The sequence $\{e^k\}_{k \in \mathbb{N}}$ defined by

$$e^k := \frac{\tau}{2} \|u_e^k\|^2 + \frac{\mu}{2} \|\nabla u_e^k\|^2 + \frac{\sigma}{2} \|v_e^k\|^2 + \frac{1}{2\mu} \|\Lambda_e^k\|^2 \quad (27)$$

is monotonically decreasing.

(2) The sequence $\{(u^k, v^k; \Lambda^k)\}_{k \in \mathbb{N}}$ converges to a limit point $(u^*, v^*; \Lambda^*)$ that satisfies the first-order optimality conditions (25).

Proof (1) The optimality conditions of subproblems in proposed ADMM-based algorithm are

$$\begin{cases} g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} + \mu(v^{k+1} - \nabla u^k) \\ + \Lambda^k + \sigma(v^{k+1} - v^k) = 0, \\ \lambda(u^{k+1} - f) - \mu \nabla \cdot (\nabla u^{k+1} - v^{k+1}) \\ + \nabla \cdot \Lambda^k + \tau(u^{k+1} - u^k) = 0, \\ \Lambda^{k+1} = \Lambda^k + \mu(v^{k+1} - \nabla u^{k+1}). \end{cases} \quad (28)$$

We express (28) in the form of the error differences u_e^k, v_e^k and Λ_e^k and obtain

$$\begin{cases} g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}} \\ + \mu(v_e^{k+1} - \nabla u_e^k) + \Lambda_e^k + \sigma(v_e^{k+1} - v_e^k) = 0, \\ \lambda u_e^{k+1} - \mu \nabla \cdot (\nabla u_e^{k+1} - v_e^{k+1}) + \nabla \cdot \Lambda_e^k \\ + \tau(u_e^{k+1} - u_e^k) = 0, \\ \Lambda_e^{k+1} = \Lambda_e^k + \mu(v_e^{k+1} - \nabla u_e^{k+1}). \end{cases}$$

Taking the inner product of each of the above equations, respectively, with v_e^{k+1}, u_e^{k+1} and Λ_e^k , we obtain

$$\begin{cases} \langle g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}}, v_e^{k+1} \rangle \\ + \mu \langle v_e^{k+1} - \nabla u_e^k, v_e^{k+1} \rangle + \sigma \langle v_e^{k+1} - v_e^k, v_e^{k+1} \rangle \\ = - \langle \Lambda_e^k, v_e^{k+1} \rangle, \\ \lambda \|u_e^{k+1}\|^2 + \mu \langle \nabla u_e^{k+1} - v_e^{k+1}, \nabla u_e^{k+1} \rangle \\ + \tau \langle u_e^{k+1} - u_e^k, u_e^{k+1} \rangle = \langle \Lambda_e^k, \nabla u_e^{k+1} \rangle, \\ \langle \Lambda_e^{k+1} - \Lambda_e^k, \Lambda_e^k \rangle = \mu \langle v_e^{k+1} - \nabla u_e^{k+1}, \Lambda_e^k \rangle. \end{cases}$$

Then, applying $\langle x - y, x \rangle = \frac{1}{2}(\|x\|^2 + \|x - y\|^2 - \|y\|^2)$ to the above equations yields

$$\begin{cases} \langle g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}}, v_e^{k+1} \rangle \\ + (\frac{\sigma+\mu}{2}) \|v_e^{k+1}\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^k\|^2 \\ + \frac{\sigma}{2} \|v^{k+1} - v^k\|^2 = \frac{\mu}{2} \|\nabla u_e^k\|^2 + \frac{\sigma}{2} \|v_e^k\|^2 - \langle \Lambda_e^k, v_e^{k+1} \rangle, \\ \lambda \|u_e^{k+1}\|^2 + \frac{\mu}{2} \|\nabla u_e^{k+1}\|^2 + \frac{\mu}{2} \|\nabla u_e^{k+1} - v_e^{k+1}\|^2 \\ + \frac{\tau}{2} \|u_e^{k+1}\|^2 + \frac{\tau}{2} \|u^{k+1} - u^k\|^2 \\ = \frac{\tau}{2} \|u_e^k\|^2 + \frac{\mu}{2} \|v_e^{k+1}\|^2 + \langle \Lambda_e^k, \nabla u_e^{k+1} \rangle, \\ \frac{1}{2\mu} \|\Lambda_e^{k+1}\|^2 = \frac{1}{2\mu} \|\Lambda_e^k\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^{k+1}\|^2 \\ + \langle v_e^{k+1} - \nabla u_e^{k+1}, \Lambda_e^k \rangle. \end{cases} \quad (29)$$

Moreover, summing of the equations in (29), it follows that

$$\begin{aligned} & \frac{\tau}{2} \|u_e^{k+1}\|^2 + \frac{\mu}{2} \|\nabla u_e^{k+1}\|^2 + \frac{\sigma}{2} \|v_e^{k+1}\|^2 + \frac{1}{2\mu} \|\Lambda_e^{k+1}\|^2 \\ & + \frac{\tau}{2} \|u^{k+1} - u^k\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^k\|^2 \\ & + \frac{\sigma}{2} \|v^{k+1} - v^k\|^2 + \lambda \|u_e^{k+1}\|^2 + \langle g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} \\ & - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}}, v_e^{k+1} \rangle = \frac{\tau}{2} \|u_e^k\|^2 + \frac{\mu}{2} \|\nabla u_e^k\|^2 \\ & + \frac{\sigma}{2} \|v_e^k\|^2 + \frac{1}{2\mu} \|\Lambda_e^k\|^2. \end{aligned}$$

By the assumption of the theorem, we have $\langle g(\kappa(u^k)) \frac{v^{k+1}}{\sqrt{1+|v^{k+1}|^2}} - g(\kappa(\bar{u})) \frac{\bar{v}}{\sqrt{1+|\bar{v}|^2}}, v_e^{k+1} \rangle = \Delta^{k+1} \geq 0$. By dropping the nonnegative terms $\lambda \|u_e^{k+1}\|^2$ and Δ^{k+1} , there is

$$\begin{aligned} e^{k+1} + \frac{\tau}{2} \|u^{k+1} - u^k\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^k\|^2 \\ + \frac{\sigma}{2} \|v^{k+1} - v^k\|^2 \leq e^k. \end{aligned} \quad (30)$$

Since $\frac{\tau}{2} \|u^{k+1} - u^k\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^k\|^2 + \frac{\sigma}{2} \|v^{k+1} - v^k\|^2 \geq 0$, then $\{e^k\}_{k \in \mathbb{N}}$ is a monotonically decreasing sequence.

(2) By the sequence $\{e^k\}_{k \in \mathbb{N}}$ being monotonically decreasing, we add the inequality (30) from $k = 0$ to ∞ to obtain

$$\begin{aligned} \sum_{k=0}^{\infty} \left(\frac{\tau}{2} \|u^{k+1} - u^k\|^2 + \frac{\mu}{2} \|v_e^{k+1} - \nabla u_e^k\|^2 \right. \\ \left. + \frac{\sigma}{2} \|v^{k+1} - v^k\|^2 \right) \leq \sum_{k=0}^{\infty} e^k - e^{k+1} < \infty. \end{aligned}$$

This indicates

$$\begin{aligned} \lim_{k \rightarrow \infty} \|u^{k+1} - u^k\| &= \lim_{k \rightarrow \infty} \|v_e^{k+1} - \nabla u_e^k\| \\ &= \lim_{k \rightarrow \infty} \|v^{k+1} - v^k\| = 0. \end{aligned} \quad (31)$$

In addition, using (31) and the Minkowski's inequality, we also obtain

$$\begin{aligned} \|\Lambda^{k+1} - \Lambda^k\| &\leq (\mu \|v_e^{k+2} - v_e^{k+1}\| \\ &+ \mu \|v_e^{k+2} - \nabla u_e^{k+1}\|) = 0, \end{aligned}$$

thus

$$\lim_{k \rightarrow \infty} \|\Lambda^{k+1} - \Lambda^k\| = \lim_{k \rightarrow \infty} \|v^{k+1} - \nabla u^{k+1}\| = 0.$$

Based on part (1), the error term $\{e^k\}_{k \in \mathbb{N}}$ is a monotone decreasing sequence for any point $(\bar{u}, \bar{v}; \bar{\Lambda})$ that satisfies

(25) in \mathbb{R}^+ , this shows the sequence $\{(u^k, v^k; \mathbf{A}^k)\}_{k \in \mathbb{N}}$ generated by Algorithm 3.1 is a Cauchy sequence and has a weakly convergent subsequence. We denote the subsequence as $\{(u^{k_\ell}, v^{k_\ell}; \mathbf{A}^{k_\ell})\}_{\ell \in \mathbb{N}}$ and the limit point as $(u^*, v^*; \mathbf{A}^*)$.

Therefore, the sequence $\{(u^{k_\ell}, v^{k_\ell}; \mathbf{A}^{k_\ell})\}_{\ell \in \mathbb{N}}$ satisfies the optimality conditions (28) such that

$$\begin{cases} g(\kappa(u^{k_\ell})) \frac{v^{k_\ell+1}}{\sqrt{1+|v^{k_\ell+1}|^2}} + \mu(v^{k_\ell+1} - \nabla u^{k_\ell}) + \mathbf{A}^{k_\ell} \\ \quad + \sigma(v^{k_\ell+1} - v^{k_\ell}) = 0, \\ \lambda(u^{k_\ell+1} - f) - \mu \nabla \cdot (\nabla u^{k_\ell+1} - v^{k_\ell+1}) + \nabla \cdot \mathbf{A}^{k_\ell} \\ \quad + \tau(u^{k_\ell+1} - u^{k_\ell}) = 0, \\ \mathbf{A}^{k_\ell+1} = \mathbf{A}^{k_\ell} + \mu(v^{k_\ell+1} - \nabla u^{k_\ell+1}). \end{cases} \quad (32)$$

By taking the limit of the convergent sequence, it yields

$$\begin{cases} g(\kappa(u^*)) \frac{v^*}{\sqrt{1+|v^*|^2}} + \mathbf{A}^* = 0, \\ \lambda(u^* - f) + \nabla \cdot \mathbf{A}^* = 0, \\ v^* - \nabla u^* = 0, \end{cases}$$

for almost every point in Ω . This implies that the limit point $(u^*, v^*; \mathbf{A}^*)$ of the sequence $\{(u^k, v^k; \mathbf{A}^k)\}_{k \in \mathbb{N}}$ satisfies the first-order optimality conditions (25). \blacksquare

Remark 3.3 Similarly to the κ TV method in [54], we introduce the proximal terms to gain the theoretical convergence of Algorithm 3.1 and simply set $\sigma = 0$ and $\tau = 0$ in the numerical experiments. Although it is difficult to validate the lower bound of Δ_k , we observe that $\Delta_k \geq 0$ holds for $k \geq 0$ numerically as shown in Fig. 6.

4 Experiments

In this section, comprehensive experiments on both synthetic and real image reconstruction problems are implemented to verify the effectiveness and efficiency of our curvature-based model. All numerical experiments are performed utilizing MATLAB R2016a on a machine with 3.40GHz Intel(R) Core(TM) i7-6700 CPU and 32GB RAM.

In our experiments, we use both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [48] to quantitatively evaluate the qualities of the reconstruction results. The PSNR is defined explicitly as

$$\text{PSNR}(u_0, u) = 10 \log \frac{255^2}{\text{MSE}},$$

where u_0 denotes the clean image, u denotes the recovery image and MSE denotes the mean squared error between u_0 and u . We further define the SSIM as follows

$$\text{SSIM}(u_0, u) = \frac{(2\mu_{u_0}\mu_u + c_1)(2\sigma_{u_0u} + c_2)}{(\mu_{u_0}^2 + \mu_u^2 + c_1)(\sigma_{u_0}^2 + \sigma_u^2 + c_2)},$$

where μ_{u_0} and μ_u express the local mean values of images u_0 and u , σ_{u_0} and σ_u signify the respective standard deviations, σ_{u_0u} is the covariance value between the clean image u_0 and recovered image u , and c_1 and c_2 are two constants to avoid instability for near-zero denominator values. Theoretically, higher PSNR and SSIM values indicate better performance of the restoration methods.

4.1 Computational Complexity

In this subsection, we analyze the computational complexity of Algorithm 3.1. The main computationally expensive components include the calculation of H or K, the FFT, inverse FFT and Newton iterations. Generally speaking, calculating the H or K on image surface costs $\mathcal{O}(mn)$. The computational complexity of FFT, inverse FFT in u -subproblem is well-known as $\mathcal{O}(mn \log(mn))$ at each iteration. The v -subproblem with two components can be computed at the cost $\mathcal{O}(2mn)$ using the Newton iterations. Therefore, the total computational complexity of Algorithm 3.1 is $\mathcal{O}(2mn \log(mn) + 3mn)$. On the other hand, the augmented Lagrangian method (ALM) of the Euler elastica model in [47] has four subproblems, which are solved by the FFT, inverse FFT and shrinkage operators. Its total computational complexity can be expressed as $\mathcal{O}(6mn \log(mn) + 4mn)$ per iteration. Besides, the augmented Lagrangian method for mean curvature regularization model in [60] has five subproblems, whose total computational complexity can be denoted as $\mathcal{O}(6mn \log(mn) + 8mn)$ per iteration. Therefore, our algorithm has lower computational complexity per iteration compared to the other two curvature-based models.

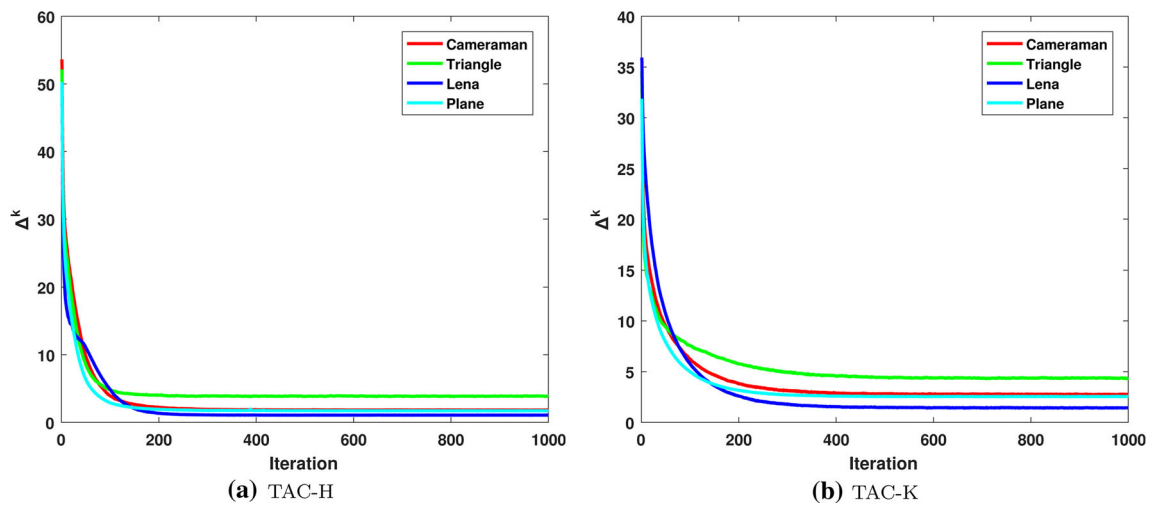


Fig. 6 The behavior of Δ^k with iteration numbers in TAC method on different test images with $\tau = 0$ and $\sigma = 0$. Note that $\Delta^k \geq 0$ for all iterations

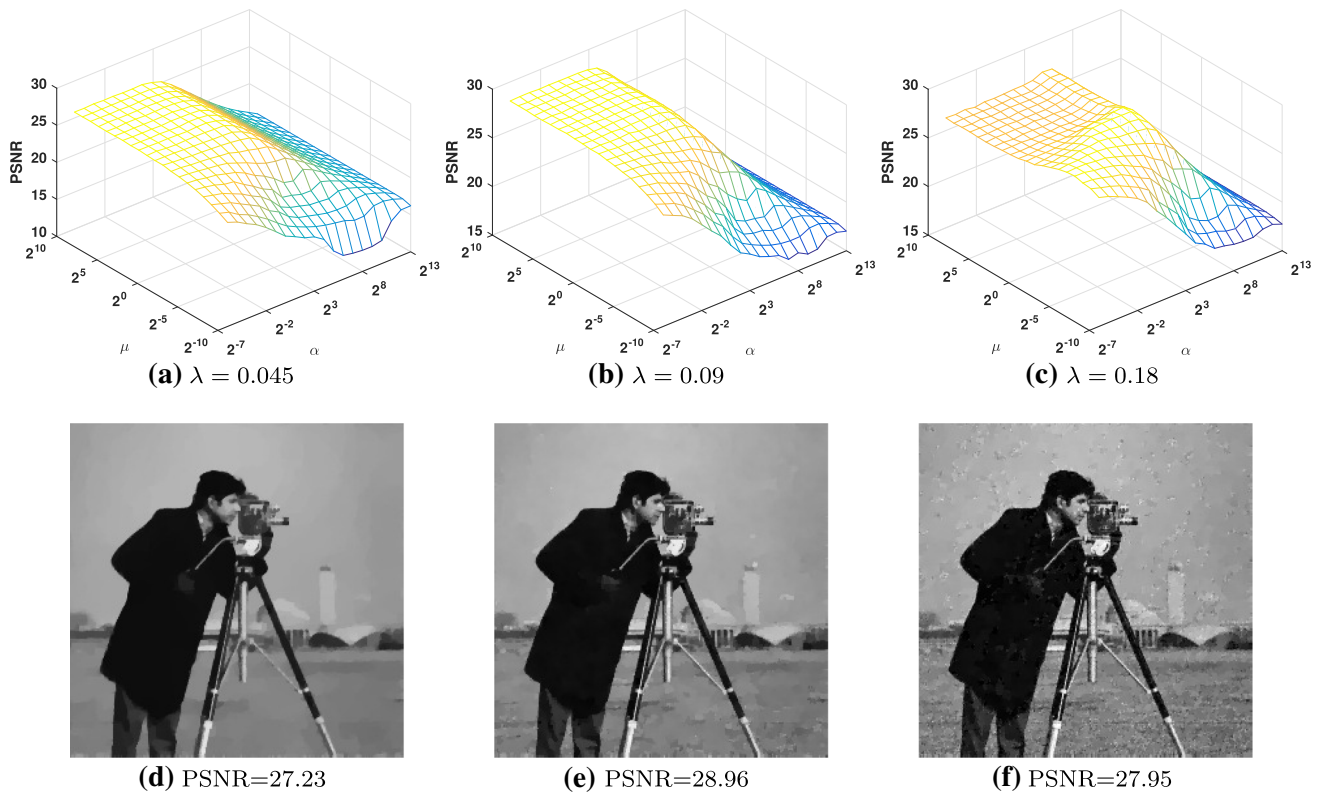


Fig. 7 The PSNR evolutions of ‘Cameraman’ obtained by different combinations of the parameter μ and α with fixed regularized parameters λ in TAC-K method

4.2 Parameters Discussion

There are a total of three parameters in the proposed algorithm, i.e., λ , α and μ . Both λ and μ are model parameters, such that λ balances the contributions between the data fidelity and the regularization term, and α tradeoffs the influences between the curvature regularization and area min-

imization. The parameter μ is the penalty parameter, which affects the convergent speed and stability of the proposed algorithm. In the paragraphs below, we use the total absolute Gaussian curvature (TAC-K) as the regularization term to discuss the impacts of the parameters in more details.

We first evaluate the impact of the parameter λ with regard to α and μ on image ‘Cameraman’, where λ is chosen from

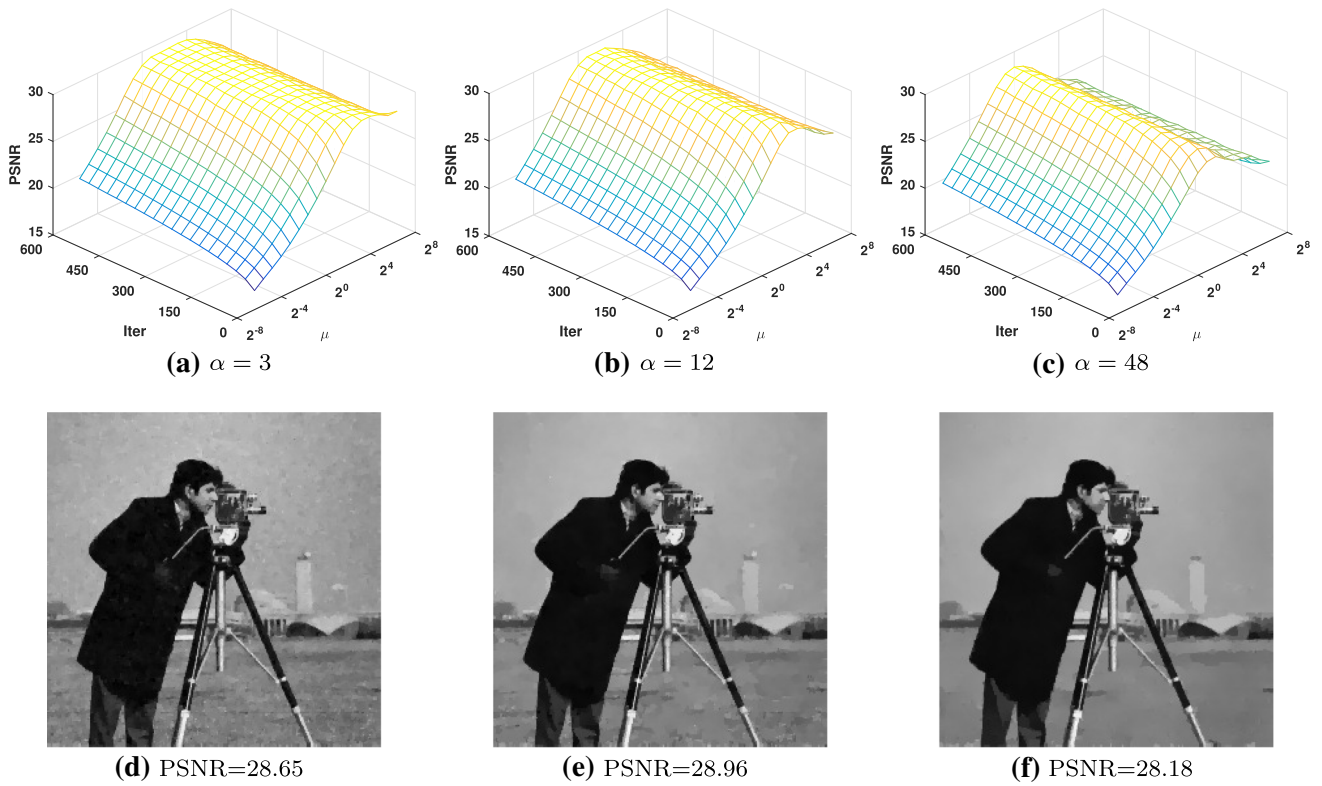


Fig. 8 The PSNR evolutions of ‘Camerman’ obtained by different combinations of the parameter μ and Iter with fixed regularized parameters α in TAC-K method

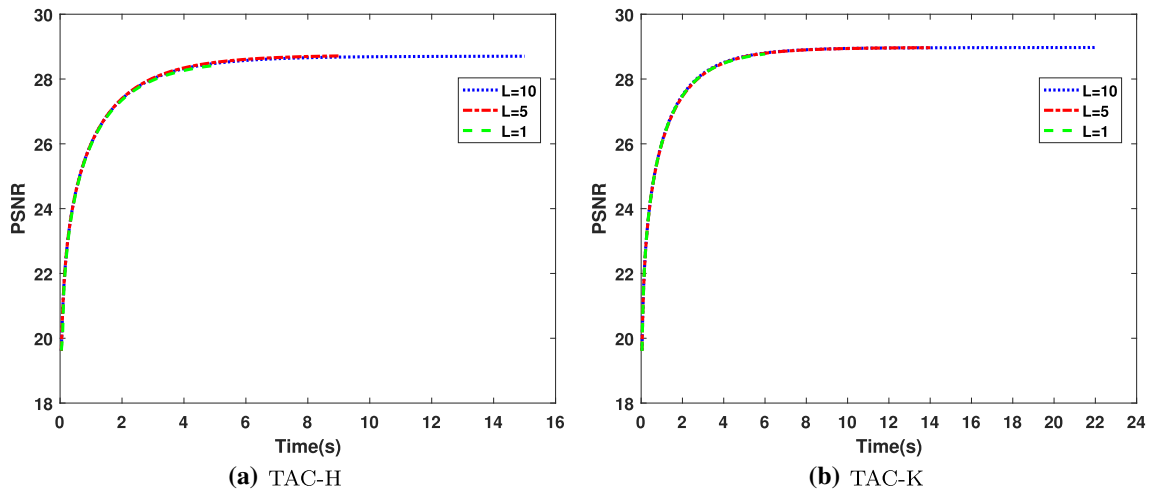


Fig. 9 The PSNR evolutions of ‘Camerman’ obtained by different inner Newton iterations (L) in TAC-H and TAC-K methods

$\lambda \in \{0.045, 0.09, 0.18\}$. For each λ , we vary the parameters $(\alpha, \mu) \in \{\alpha^0 \times 2^{-\delta}, \alpha^0 \times 2^{-\delta+1}, \dots, \alpha^0 \times 2^{\delta-1}, \alpha^0 \times 2^{\delta}\} \times \{\mu^0 \times 2^{-\delta}, \mu^0 \times 2^{-\delta+1}, \dots, \mu^0 \times 2^{\delta-1}, \mu^0 \times 2^{\delta}\}$ with $\alpha^0 = 12, \mu^0 = 2$ and $\delta = 9$. In Fig. 7, we plot the values of PSNR versus different combinations of λ and (α, μ) , together with the best recovery results for $\lambda = \{0.045, 0.09, 0.18\}$, respectively. For a fixed λ , the value of PSNR drops as α and μ become too large, but there are relatively large intervals

for both α and μ to produce satisfactory results. In addition, too small λ results in over smooth recovery result with some details missing, while too large λ leads to non-smooth recovery results with some noises remaining. Thus, λ needs to be manually tuned according to the noise levels and image structures in practice.

We further analyze the impact of α relating to μ and the convergence with a fixed $\lambda = 0.09$. Different values

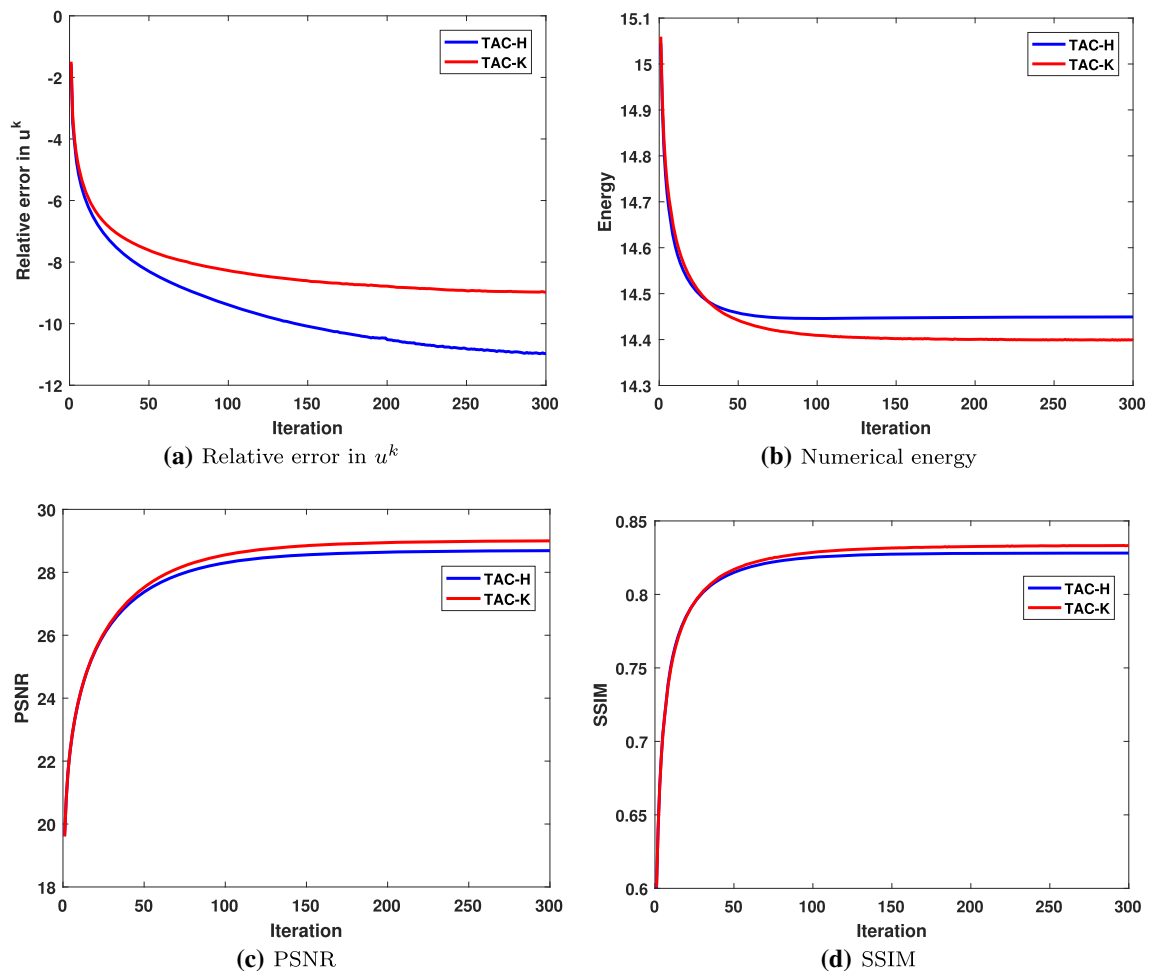


Fig. 10 Evaluations of ‘Cameraman’ obtained by the proposed methods. From left to right: Relative error in u^k , numerical energy, PSNR and SSIM, respectively (Color figure online)

Table 2 The PSNR and SSIM of Gaussian noise removal for the Euler’s elastica and our curvature-based models

Images	Noisy image	Euler’s elastica	TAC-H	TAC-K	TSC-H	TSC-K	TRV-H	TRV-K
A1(100 × 100)	28.24	36.52	38.21	38.08	38.05	37.94	37.98	37.87
PSNR/SSIM	0.5925	0.9515	0.9660	0.9647	0.9650	0.9638	0.9625	0.9622
A2(60 × 60)	28.29	35.02	36.27	36.16	36.14	36.05	36.08	36.02
PSNR/SSIM	0.6458	0.9484	0.9570	0.9555	0.9542	0.9535	0.9533	0.9525
A3(128 × 128)	28.25	38.85	39.65	39.50	39.54	39.45	39.51	39.40
PSNR/SSIM	0.5164	0.9690	0.9769	0.9760	0.9763	0.9750	0.9752	0.9748

The bold values denote the best results

of $\alpha \in \{3, 12, 48\}$ and $(\mu, \text{Iter}) \in \{2^{-8}, 2^{-7}, \dots, 2^7, 2^8\} \times \{0, 30, \dots, 570, 600\}$ are used to evaluate the denoising performance. As shown in Fig. 8, for a fixed α , we notice that the value of μ affects the numerical convergence such that a relatively large μ yields faster convergence and a too small μ usually results in using more iterations to reach convergence. Therefore, we use $\mu = 2$ as its default value unless otherwise specified. With the default μ , the plots reveal that the number

of iterations $\text{Iter} = 300$ is enough for different α . We also observe that a too small α (e.g., $\alpha = 3$ for ‘Cameraman’) leads to non-smooth results and a too large α (e.g., $\alpha = 48$ for ‘Cameraman’) results in over-smooth results. Thus, α should be properly chosen according to image structures and noise levels to smooth out the homogenous regions as well as preserve the image details.

Moreover, the iteration number of inner Newton's method also affects the efficiency of the proposed algorithm. Under the same stopping condition $\epsilon = 2 \times 10^{-5}$, Fig. 9 displays the values of PSNR versus CPU time with three different options of the iteration number L , i.e., $L = 1, 5$ and 10 . It can be seen that a too large L results in higher computational cost and a too small L leads to a lower PSNR. Therefore, we fix $L = 5$ for all applications. Lastly, we fix the spatial step size h as $h = 0.5$ in all experiments for the best balance between the smoothness and fine details.

4.3 Numerical Convergence

The decays of the relative error and numerical energy are provided to illustrate the convergence of the ADMM algorithm versus the iterations

$$\text{ReErr}(u^k) = \frac{\|u^k - u^{k-1}\|_1}{\|u^{k-1}\|_1},$$

and

$$E(u^k) = \sum_{i,j} g(\kappa(u_{i,j}^{k-1})) \sqrt{1 + |\nabla u_{i,j}^k|^2} + \frac{\lambda}{2} \|u^k - f\|^2.$$

With the optimal parameters $\lambda = 0.09, \mu = 2, \alpha = \{0.3, 12\}$ for TAC-H and TAC-K models, respectively, we track the decays of the relative errors and the numerical energies, as well as the values of PSNRs and SSIMs for 'Cameraman', which are displayed using log-scale in Fig. 10. These plots demonstrate the convergence of the iterative process and the stability of the proposed methods. Note that the TAC-K model usually converges to a comparable larger relative error due to a relatively larger α is used in the Gaussian curvature-based model to achieve the best restoration results.

4.4 Gaussian Denoising

In the first place, we compare the proposed models relying on TAC, TSC and TRV, with the Euler's elastica model on image denoising examples. Three smooth images (i.e., A1, A2 and A3) corrupted by Gaussian noise with zero mean and the standard deviation 10 are used in the evaluation. We fix the parameters $\lambda = 0.11$ and $\mu = 2$ for our model, and set $\alpha = 0.3$ for the mean curvature-based variational models (i.e., TAC-H, TSC-H and TRV-H) and $\alpha = 12$ for the Gaussian curvature-based models (i.e., TAC-K, TSC-K and TRV-K). In particular, we implement the ALM algorithm in [47] with the same parameters as the ones used in the original paper such that $\alpha = 10, \lambda = 10^2, r_1 = 1, r_2 = 2 \cdot 10^2, r_4 = 5 \cdot 10^2$ and $\epsilon = 10^{-2}, 1.3 \cdot 10^{-3}, 8 \cdot 10^{-3}$ for image A1 to A3, respectively.

In Table 2, we detail the comparison results in terms of PSNR and SSIM. It can be observed that our curvature-based model always achieves higher PSNR and SSIM than the Euler's elastica method for all combinations of curvature functions and curvature types. Moreover, the TAC-H model gives the best recovery results for all three images among the combinations. In Fig. 11, we display the restoration results obtained by the Euler's elastica model and our TAC-H model, which clearly shows our model can ideally preserve image structures such as edges and corners. The numerical mean curvatures of the image A1 and A2 are exhibited in Fig. 12, which are computed using Eqs. (12)–(17) on the clean images, denoising images obtained by the Euler's elastica model and our TAC-H model. For a fair comparison, we min-max scale all images into $[0, 1]$ in advance of the calculation. It can be observed the numerical mean curvature is relatively small in the homogeneous regions, and jumps across the edges, which demonstrates that mean curvature regularization can well preserve sharp jumps in images. The image surfaces of the clean images and restored images of Euler's elastica and TAC-H models in Fig. 13 also accord with our conclusion such that our mean curvature regularization has an advantage in preserving the edges and sharp corners over the Euler's elastica.

To further demonstrate the effectiveness and efficiency of our curvature-based model, we evaluate the performance on more synthetic and natural images by comparing with several state-of-the-art variational denoising methods including the total variation (TV) model in [53], Euler's elastica (Euler) model in [47], the second-order total generalized variation (TGV) model in [7] and mean curvature regularizer (MEC) model in [60]. Four test images (i.e., 'Cameraman', 'Triangle', 'Lena' and 'Plane') are explored and degraded by the Gaussian noise with zero mean and the standard deviation 20. To set up the experimental comparison as fair as possible, the parameters of the comparative methods are selected as suggested in the corresponding papers, which are given as (a) TV: $r_1 = 10$ and $\lambda = 15$; (b) Euler's elastica: $\alpha = 10, r_1 = 1, r_2 = 2 \cdot 10^2, r_4 = 5 \cdot 10^2$ and $\lambda = 2 \cdot 10^2$; (c) TGV: $\alpha_0 = 1.5, \alpha_1 = 1.0, r_1 = 10, r_2 = 50$ and $\lambda = 10$; (d) MEC: $r_1 = 40, r_2 = 40, r_3 = 10^5, r_4 = 1.5 \cdot 10^5$ and $\lambda = 10^2$. The experience-dependent parameters are set as $\lambda = 0.09, \mu = 2, \alpha = 0.3$ for TAC-H model and $\alpha = 12$ for TAC-K model.

We compare the restoration results both quantitatively and qualitatively. The recovery results and the residual images $f - u$ of 'Cameraman' and 'Triangle' are visually exhibited in Figs. 14 and 15, while the denoising images and the selected local magnified views of 'Lena' and 'Plane' are shown in Figs. 16 and 17. In general, all methods can remove the noises and recover the major structures and features quite well. However, the TV model suffers from obvious staircase-like artifacts and lots of image details and textures

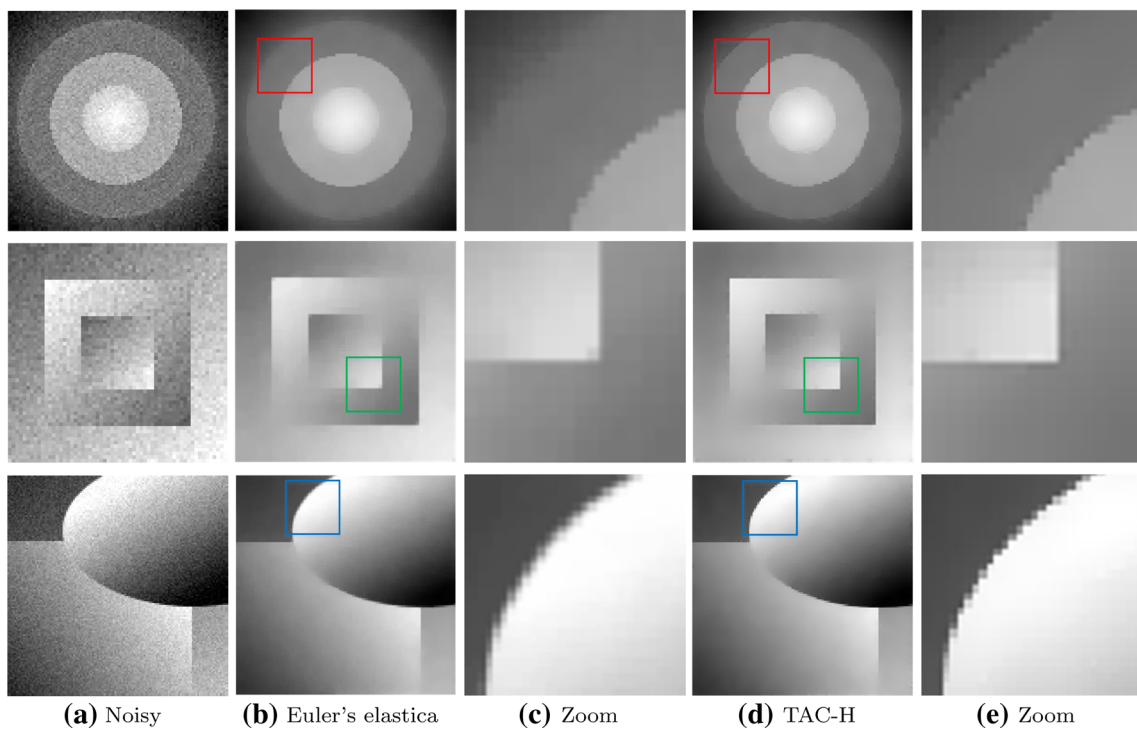


Fig. 11 The denoising results of the smooth images A1, A2 and A3 (from top to bottom) obtained by the Euler's elastica and our TAC-H model

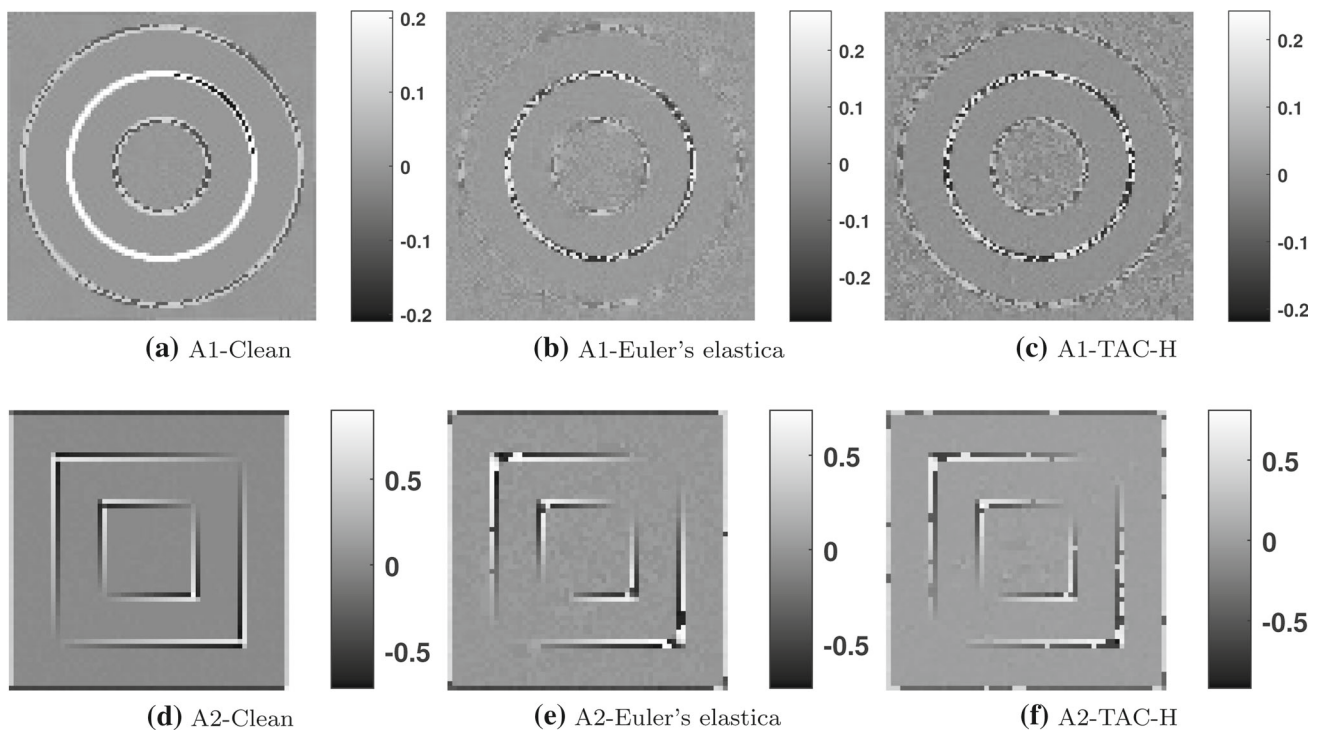


Fig. 12 The numerical mean curvature of the clean images and denoising images obtained by the Euler's elastica and TAC-H model

can be observed in the residual images. As representatives of high-order regularities, the Euler's elastica, TGV and MEC models can overcome the staircase effects and preserve image

details to a certain extent. And our TAC-H and TAC-K models always give the best recovery results, which can not only keep the smoothness in the homogeneous regions but also

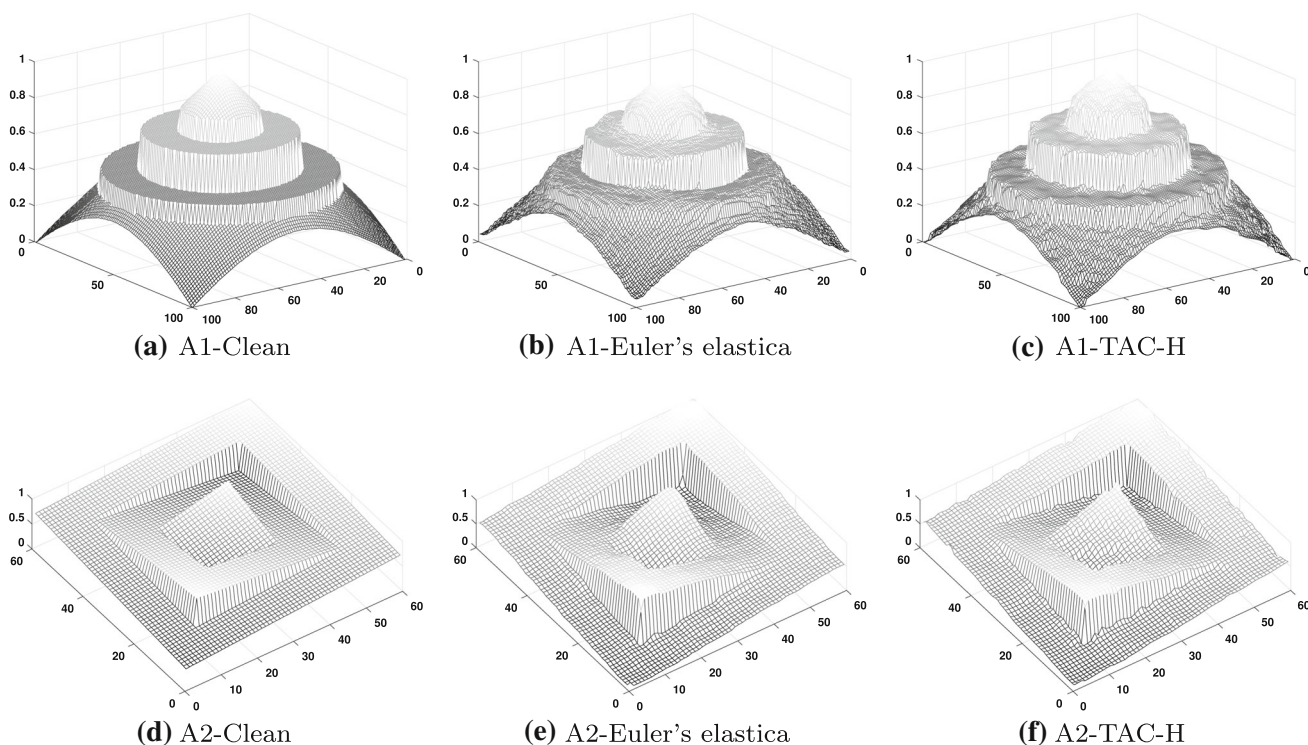


Fig. 13 The image surfaces of the clean images and reconstructed images by the Euler's elastica and our TAC-H model

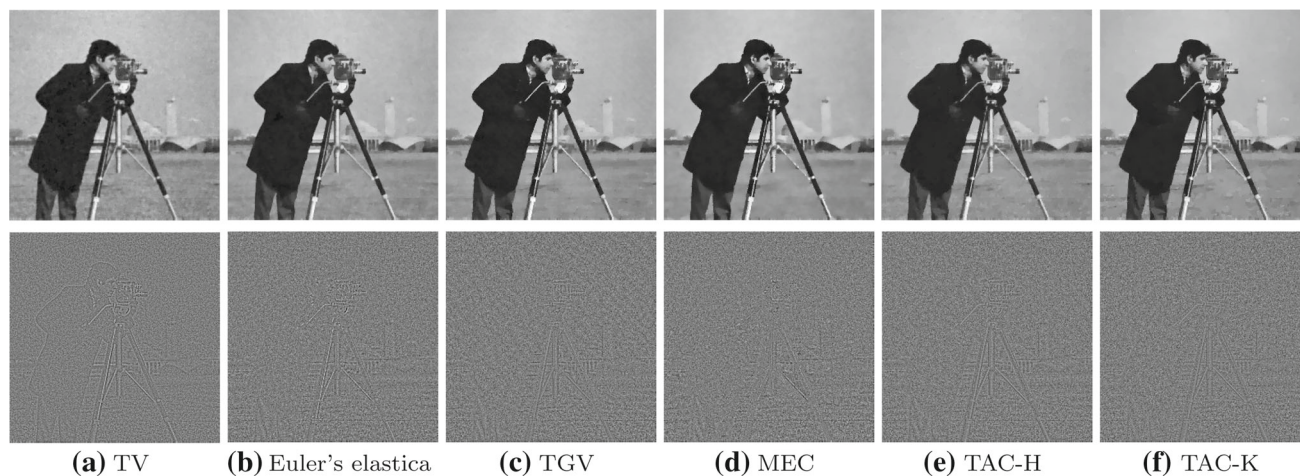


Fig. 14 The denoising results of 'Cameraman' (top) and the corresponding residual images (bottom) generated by the comparative methods

preserve fine details in the texture regions. Table 3 records the values of PSNR and SSIM for all the algorithms, which also convinces our TAC-K model and gives the overall best recovery results.

The CPU time recorded in Table 4 illustrates that our TAC-H and TAC-K models are more efficient than other high-order methods, which are significantly faster than the Euler's elastica and mean curvature models. To better visualize the convergence of the comparative methods, we plot the relative errors in u^k of these methods for 'Lena' and 'Plane' in Fig. 18. Although the relative error of the Euler's elastica model drops

faster at the beginning, our curvature-based models can attain smaller relative errors as the number of iterations increases. Thus, our proposal usually converges faster than others when a stringent relative error is used as the stopping criterion.

The visual illustrations of the numerical mean and Gaussian curvatures of 'Cameraman' and 'Lena' estimated on the noisy images, restoration images, and the clean images are presented in Fig. 19. Significant noises can be observed in the curvature maps of noisy images, while the curvature maps of the recovery images are noiseless and only jump on edges. Indeed, the curvature maps of the restorations are much alike

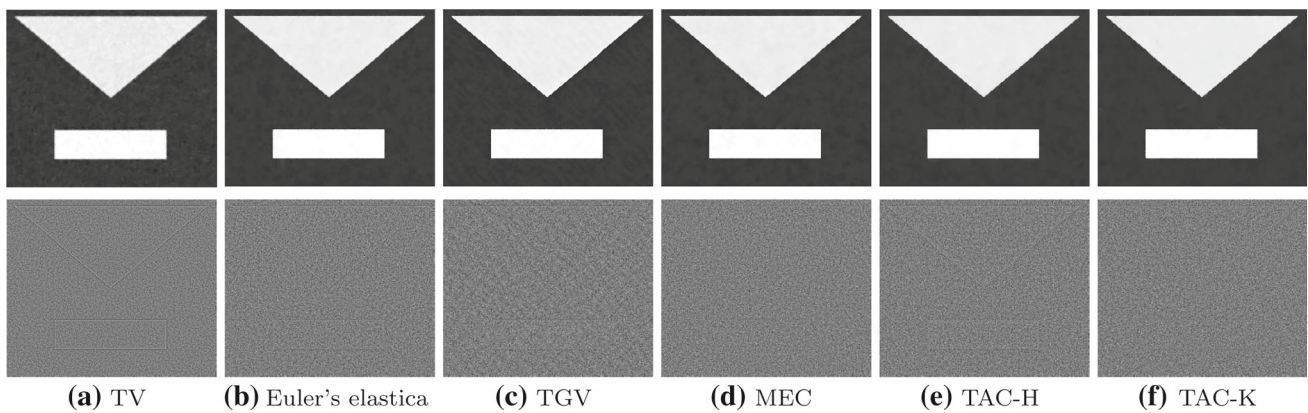


Fig. 15 The denoising results of ‘Triangle’ (top) and the corresponding residual images (bottom) generated by the comparative methods

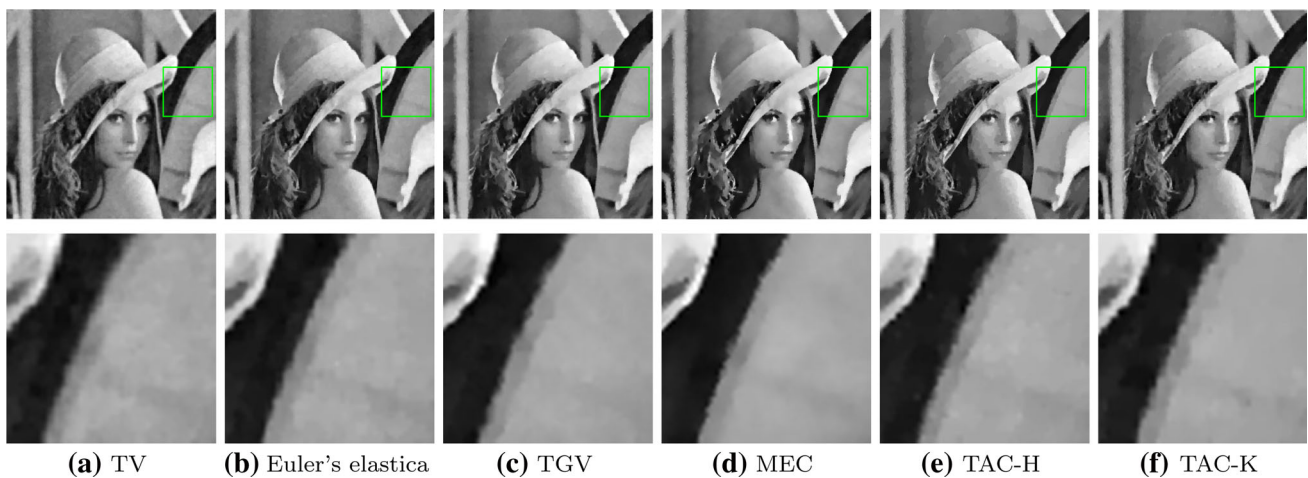


Fig. 16 The denoising results of ‘Lena’ (top) and the corresponding local magnification views (bottom) generated by the comparative methods

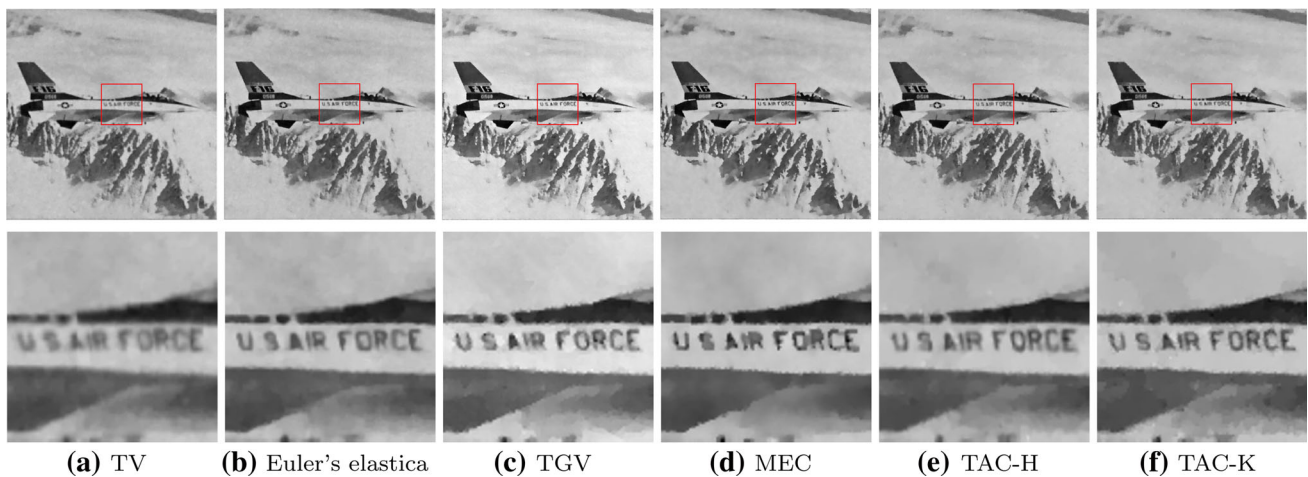


Fig. 17 The denoising results of ‘Plane’ (top) and the corresponding local magnification views (bottom) generated by the comparative methods

to the ones obtained on the clean images in visual perceptions. It reveals that the TAC-H and TAC-K models successfully reduce the noises contained in the mean curvature and Gaussian curvature maps, which indicates the reasonability and

effectiveness of our proposed models. Through an in-depth comparison between the curvature maps of the recovery images and clean images, we have the following two observations:

Table 3 The PSNR and SSIM of Gaussian noise removal for comparative methods

Images	Noisy images	TV	Euler’s elatica	TGV	MEC	TAC-H	TAC-K
Cameraman	22.45	27.29	27.93	28.22	28.38	28.70	28.96
PSNR/SSIM	0.4087	0.7905	0.8187	0.8161	0.8203	0.8264	0.8340
Triangle	22.71	32.04	34.85	35.52	36.65	35.92	37.01
PSNR/SSIM	0.2666	0.9247	0.9588	0.9504	0.9654	0.9596	0.9762
Lena	22.34	27.25	28.10	28.04	28.18	28.32	28.56
PSNR/SSIM	0.4855	0.8139	0.8335	0.8307	0.8352	0.8388	0.8405
Plane	22.12	29.48	30.22	30.16	30.35	30.63	30.92
PSNR/SSIM	0.3555	0.8505	0.8681	0.8548	0.8719	0.8731	0.8769

The bold values denote the best results

Table 4 The CPU time(s) comparison of Gaussian noise removal for comparative methods, where underline represents the lowest computational cost, bold and italics denote the computational cost of our TAC-H and TAC-K models, respectively

Images Methods	Cameraman (256 × 256)		Triangle (214 × 254)		Lena (256 × 256)		Plane (512 × 512)	
	Time	Iterations	Time	Iterations	Time	Iterations	Time	Iterations
TV	<u>6.17</u>	300	<u>5.61</u>	300	<u>6.06</u>	300	<u>33.98</u>	300
Euler’s elatica	21.83	300	19.38	300	21.65	300	137.72	300
TGV	22.89	300	21.18	300	22.85	300	115.95	300
MEC	28.01	300	26.23	300	27.68	300	170.63	300
TAC-H	8.91	200	9.76	255	8.60	195	43.62	141
TAC-K	<i>13.32</i>	<i>300</i>	<i>11.27</i>	<i>300</i>	<i>13.24</i>	<i>300</i>	<i>90.60</i>	<i>300</i>

- Only main edges are presented in the Gaussian curvature maps. The Gaussian curvature has a small magnitude, as long as one principal curvature is small. It well explains why the Gaussian curvature-based model gives lower numerical energy; see Fig. 10b. Thus, minimizing Gaussian curvature allows for fine details and structures, which is more suitable for natural images such as ‘Lena’ and ‘Cameraman’ etc.
- More details and small edges exist in the mean curvature maps. By minimizing the total mean curvature of the noisy image, some tiny structures in the mean curvature maps will be smoothed out. Thus, the mean curvature regularization works better for images containing large homogeneous or slowly varying regions e.g., the smooth images in Fig. 11.

4.5 Salt & Pepper and Poisson Denoising

In this subsection, both the salt & pepper and Poisson denoising examples are operated to further illustrate the advantages of our curvature regularized model. According to the statistical properties of the salt & pepper noise, we adopt the l^1 -norm data fidelity term instead of the l^2 -norm one [17,24,38], which gives

$$\min_u \sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |\nabla u_{i,j}|^2} + \lambda \|u - f\|_1. \quad (33)$$

To deal with the above minimization problem, two auxiliary variables are introduced to rewrite it into the following constrained minimization problem

$$\begin{aligned} \min_{u,v,w} \quad & \sum_{i,j} g(\kappa(u_{i,j}))\sqrt{1 + |v_{i,j}|^2} + \lambda \|w\|_1 \\ \text{s.t.} \quad & v_{i,j} = \nabla u_{i,j}, \quad w = u - f. \end{aligned}$$

More details of the numerical algorithm for solving such constrained minimization problem can be referred to [24,53].

We use two grayscale test images ‘House’ (256 × 256) and ‘Realtest’ (400 × 420), both of which are corrupted by 30% salt & pepper noise. The parameters are set as $\lambda = 1.8$, $\alpha = 2$, and $\mu_1 = 2$, $\mu_2 = 0.1$. Figure 20 shows the recovery results and the selected magnified views obtained by the Euler’s elatica, TGV and our TAC-K methods. It can be observed that all methods can avoid the staircase-like artifacts in smooth regions. Table 5 illustrates that our TAC-K model gives slightly higher PSNR and SSIM than the Euler’s elatica and TGV models. Simultaneously, our TAC-K model can save much computational cost compared to the other two high-order methods on both images for meeting the same stopping criteria.

We also conduct examples on Poisson noises removal, the variational model of which can be formalized by integrating

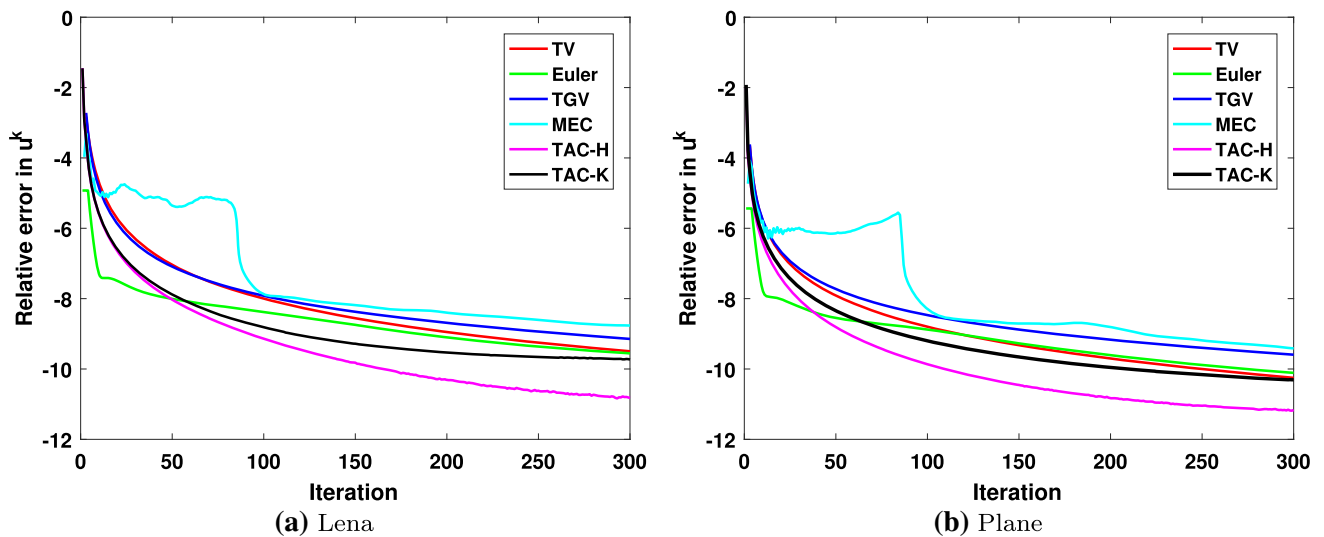


Fig. 18 Relative errors in u^k of ‘Lena’ and ‘Plane’ obtained by the comparative methods

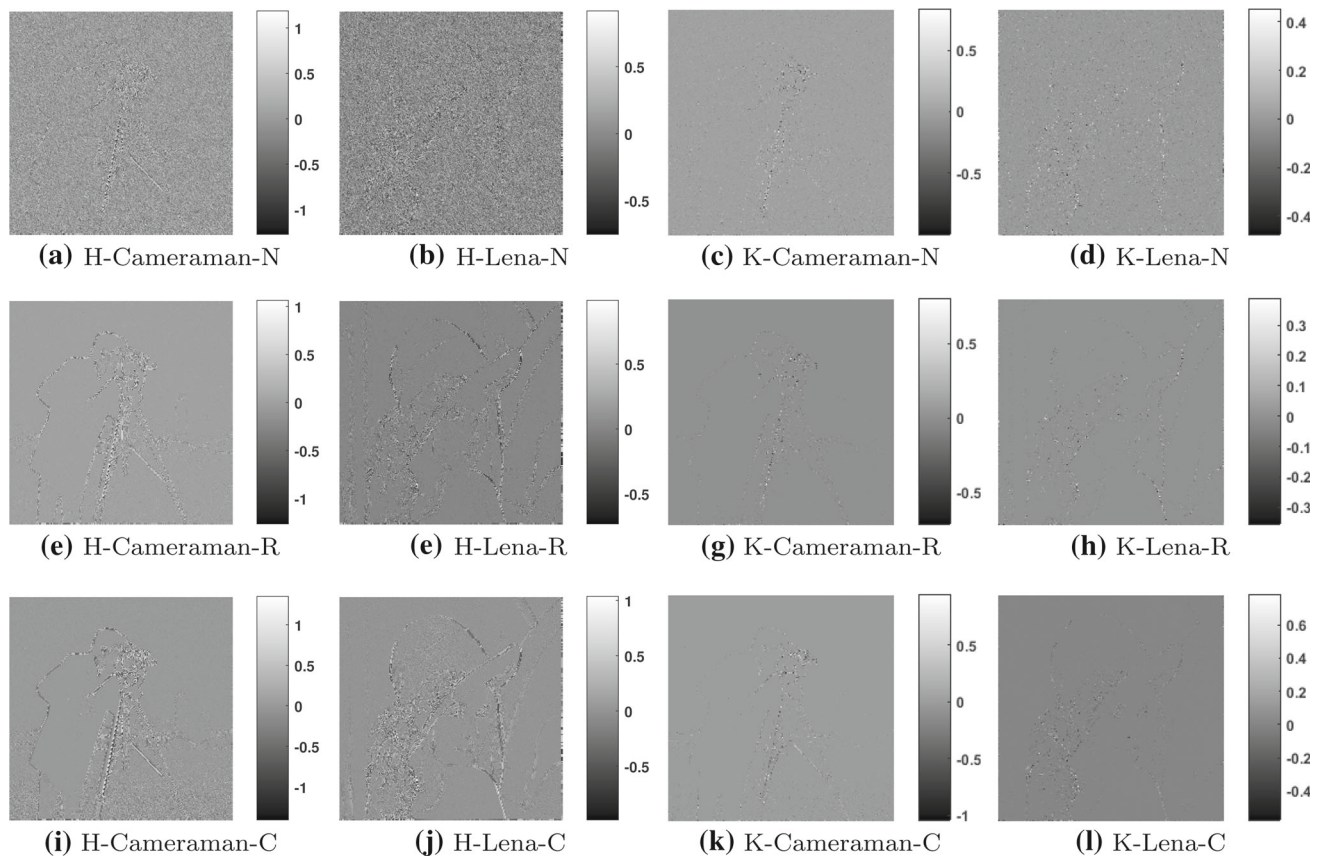


Fig. 19 The numerical mean curvature and Gaussian curvature maps of the noisy images (N), the restoration results (R) obtained by our proposals and clean images (C) on the first, second and third row, respectively

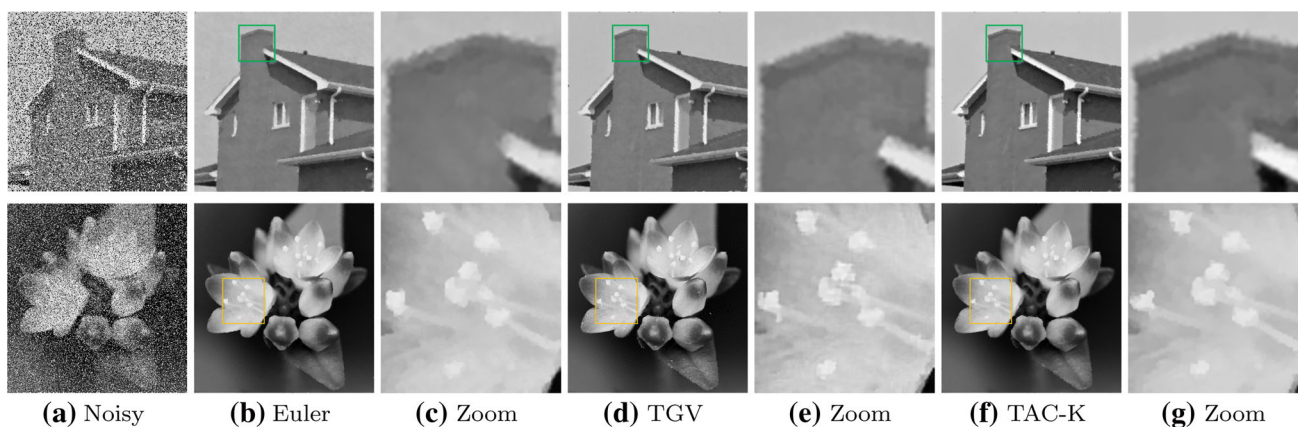


Fig. 20 The salt & pepper denoising results of ‘House’ and ‘Realtest’ obtained by the Euler’s elastica, TGV and TAC-K methods

Table 5 The evaluations of salt & pepper noise removal for the Euler’s elastica, TGV and TAC-K methods

Methods Images	Euler’s elastica			TGV			TAC-K		
	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
House (256 × 256)	30.69	0.8719	20.09	30.75	0.8726	19.25	30.94	0.8744	12.73
Realtest (420 × 400)	33.05	0.9104	76.85	32.95	0.9095	74.81	33.16	0.9112	54.65

The bold values denote the best results

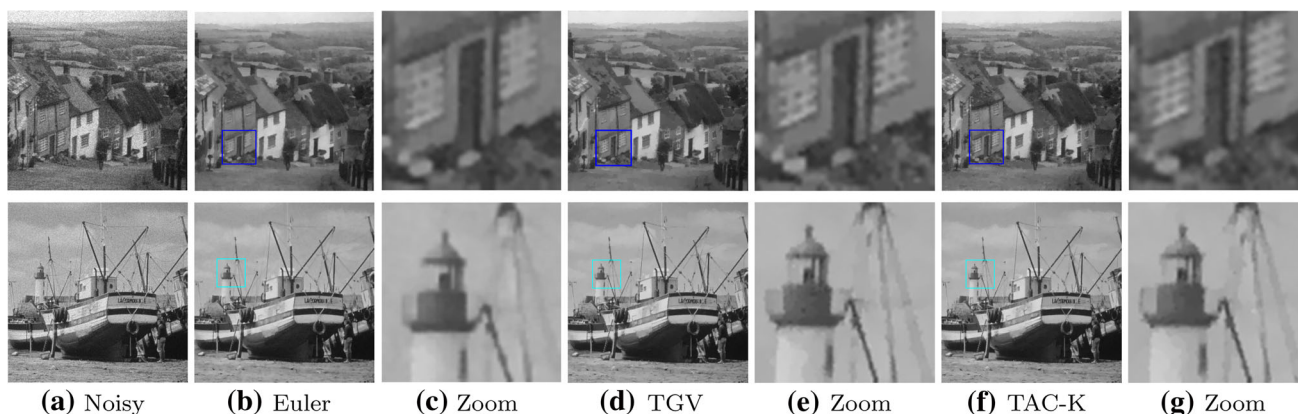


Fig. 21 The Poisson denoising results of ‘Hill’ and ‘Boats’ obtained by the Euler’s elastica, TGV and TAC-K methods

with the Kullback-Leibler (KL) distance as follows

$$\begin{aligned}
 \min_u \sum_{i,j} g(\kappa(u_{i,j})) \sqrt{1 + |\nabla u_{i,j}|^2} \\
 + \lambda \sum_{i,j} (u_{i,j} - f_{i,j} \log u_{i,j}). \tag{34}
 \end{aligned}$$

More detailed implementation of the operator splitting and ADMM-based algorithm for the above minimization problem (34) can be found in [29,51].

In Fig. 21, Poisson noises are introduced into two clean images, i.e., ‘Hill’ (256 × 256) and ‘Boats’ (512 × 512). We set the parameters in our model as $\lambda = 25$, $\alpha = 15$, $\mu_1 = 2$ and $\mu_2 = 4$. As shown in Fig. 21, the TAC-K model can

preserve more image details and features than both Euler’s elastica and TGV models, e.g., the window area in ‘Hill’ and the mast area in ‘Boats’, which are further verified by the PSNR and SSIM in Table 6. Similarly to the previous experiment, our TAC-K method outperforms the other two methods by significantly reducing the calculation time.

4.6 Color Images Denoising

In this subsection, we extend our TAC-K model to color image restoration [34,49]. Without loss of generality, we consider a vectorial function $\mathbf{u} = (u^r, u^g, u^b) : \Omega \rightarrow \mathbb{R}^3$. For the sake of simplicity, we propose to independently recover each RGB channel of color images (channel by channel),

Table 6 The evaluations of Poisson noise removal for the Euler’s elastica, TGV and TAC-K methods

Methods Images	Euler’s elastica			TGV			TAC-K		
	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
Hill (256 × 256)	31.53	0.8689	21.47	31.85	0.8739	20.04	32.15	0.8792	13.20
Boats (512 × 512)	31.90	0.8772	138.10	32.34	0.8854	126.18	32.68	0.8894	89.24

The bold values denote the best results

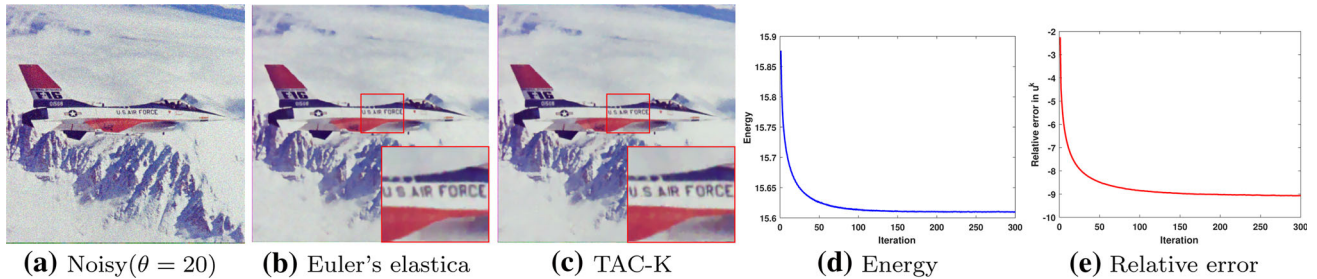


Fig. 22 The denoising results of ‘Airplane’ generated by the Euler’s elastica and TAC-K methods

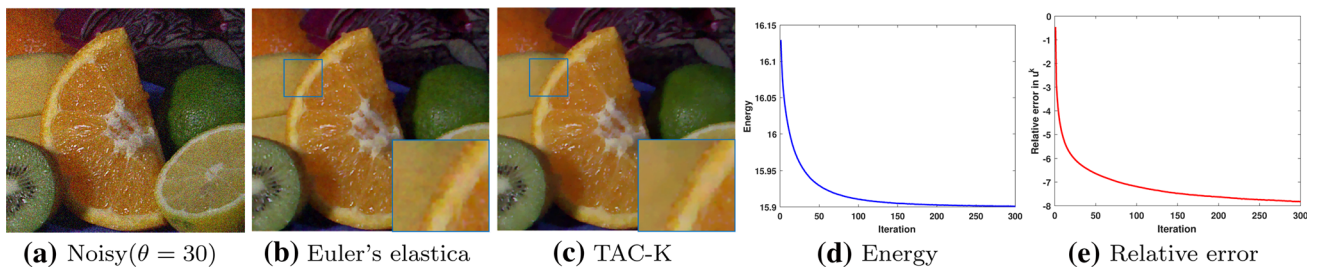


Fig. 23 The denoising results of ‘Fruits’ generated by the Euler’s elastica and TAC-K methods

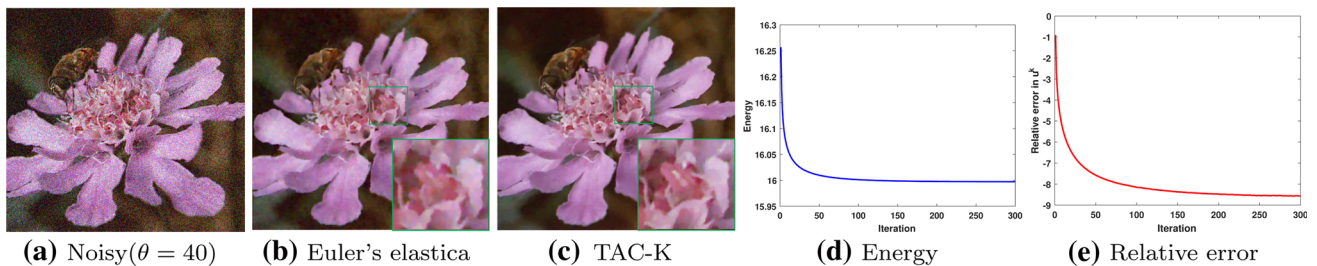


Fig. 24 The denoising results of ‘Flower’ generated by the Euler’s elastica and TAC-K methods

Table 7 The evaluations of color image noise removal for the Euler’s elastica and TAC-K methods

Images	Methods	PSNR	SSIM	Time
Airplane($\theta = 20$)	Euler’s elastica	30.55	0.8955	403.98
512 × 512	TAC-K	31.17	0.9086	258.94
Fruits($\theta = 30$)	Euler’s elastica	28.34	0.9156	335.99
480 × 512	TAC-K	28.90	0.9226	241.82
Flower($\theta = 40$)	Euler’s elastica	28.32	0.9282	336.78
480 × 512	TAC-K	28.95	0.9372	246.30

The bold values denote the best results

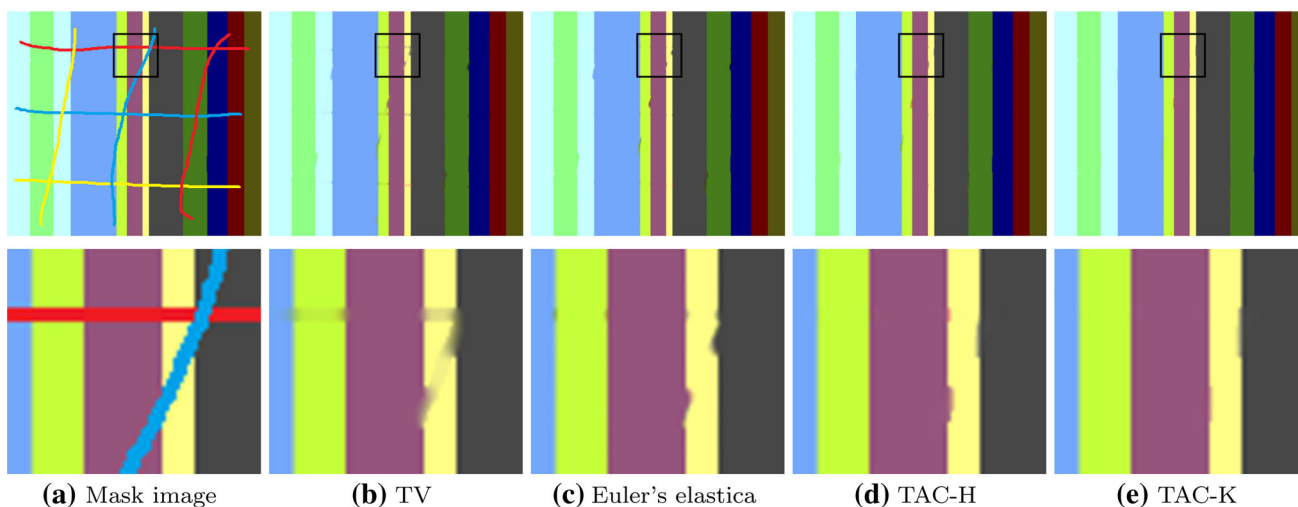


Fig. 25 The inpainting results (top) and local magnification views (bottom) of synthetic image obtained by TV, Euler's elastica, TAC-H and TAC-K methods, where the parameters are adopted as $\lambda = 5, \alpha = \{0.5, 10\}, \mu_1 = 2, \mu_2 = 0.2$

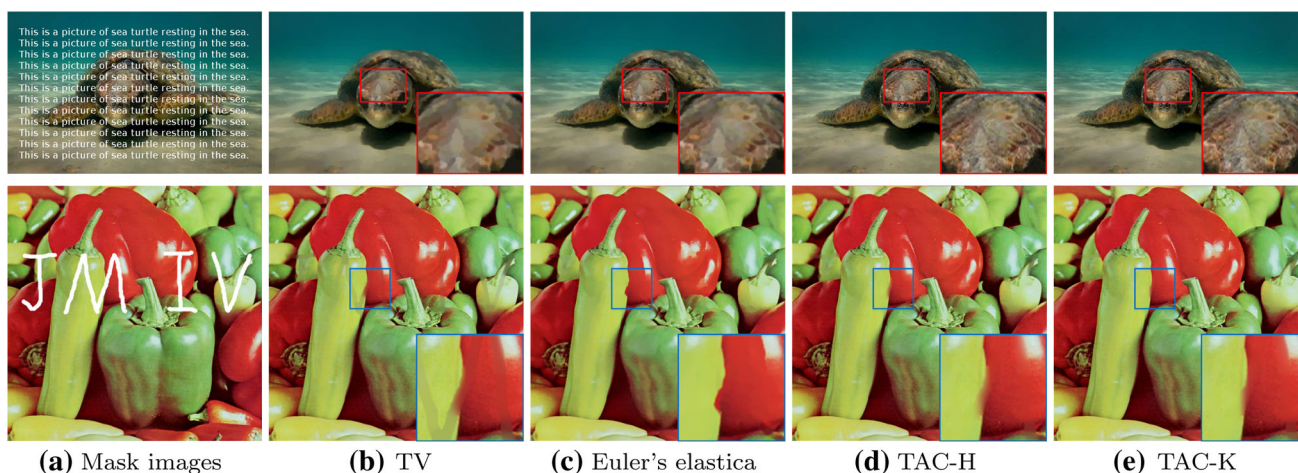


Fig. 26 The inpainting results of real images obtained by TV, Euler's elastica, TAC-H and TAC-K methods

Table 8 The evaluations of image inpainting for TV, Euler's elastica, TAC-H and TAC-K methods

Images Methods	Harmonic (315 × 357)			Turtle (318 × 500)			Pepper (512 × 512)		
	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM	Time
TV	35.92	0.9823	12.49	37.09	0.9724	40.38	34.28	0.9810	65.46
Euler's elastica	37.65	0.9895	91.66	38.57	0.9818	248.10	35.30	0.9908	410.24
TAC-H	38.87	0.9965	60.89	39.70	0.9925	165.11	36.37	0.9956	264.71
TAC-K	39.21	0.9982	62.24	40.01	0.9937	166.20	36.71	0.9970	266.56

The bold values denote the best results

and then generate the final restored image by combining the RGB channels together. Thus, the corresponding color image denoising model for Gaussian noise removal can be described as

$$\min_{\mathbf{u}} \sum_{c \in \{r, g, b\}} \sum_{i, j} g(\kappa(u_{i, j}^c)) \sqrt{1 + |\nabla u_{i, j}^c|^2}$$

$$+ \frac{\lambda}{2} \sum_{c \in \{r, g, b\}} \|u^c - f^c\|^2. \tag{35}$$

We plan to extend our curvature regularized models to the color TV model [5] and Beltrami color image model [40,41] in the future.

Three different color images are selected as examples to demonstrate the efficiency and superiority of our TAC-K model, which are ‘Airplane’, ‘Fruits’ and ‘Flower’ degraded by the Gaussian noise with zero mean and the standard deviation $\theta = \{20, 30, 40\}$, respectively. The parameters are set as $\lambda = \{0.09, 0.06, 0.03\}$, $\alpha = 10$ and $\mu = 2$ for different noise levels accordingly to guarantee satisfactory restoration results can be achieved. On the other hand, the parameters of the Euler’s elastica model are set as $\lambda = \{2 \cdot 10^2, 1.5 \cdot 10^2, 10^2\}$, $\alpha = 10$, $r_1 = 1$, $r_2 = 2 \cdot 10^2$ and $r_4 = 5 \cdot 10^2$ for the three images, respectively. As shown in Figs. 22, 23 and 24, the proposed TAC-K model can provide sharper image edges and smoother homogenous regions. The energy curves demonstrate that our algorithm converges well for the vectorial image problems. To further evaluate the denoising performance, quantitative results with different degradations are summarized in Table 7, which shows the TAC-K model outperforms the Euler’s elastica model in both recovery quality and computational efficiency.

4.7 Image Inpainting

Last but not least, we demonstrate some examples of our curvature regularized model on the applications of color image inpainting [45]. In general, the task of image inpainting is to reconstruct missing parts of an image using information from the given regions, where the missing parts are called the inpainting domain and denoted by $D \subset \Omega$. For a given image $\mathbf{f} = (f^r, f^g, f^b)$, we formulate the curvature-based inpainting model as follows

$$\min_{\mathbf{u}} \sum_{c \in \{r, g, b\}} \sum_{i, j} g(\kappa(u_{i, j}^c)) \sqrt{1 + |\nabla u_{i, j}^c|^2} + \frac{\lambda}{2} \sum_{c \in \{r, g, b\}} \|u^c - f^c\|_{\Omega \setminus D}^2, \quad (36)$$

which is also implemented channel by channel. More details of the numerical implementation of the above model can be found in [54].

In Figs. 25 and 26, we present several convincing examples obtained by the TV, Euler’s elastica, and our TAC-H, TAC-K models on both color synthetic and real image inpainting applications. It is obviously shown that the results of the TV model tend to fill in the inpainting domain with the piecewise constant values, which are unnatural for large missing domains. The Euler’s elastica, TAC-H and TAC-K models have the capability of connecting the large gap as well as protecting image structures owing to the minimization of the curvatures. Moreover, it can be observed that the TAC-K model gives the visually best inpainting results, which can well recover the inpainting domain and preserve fine details. Table 8 records the evaluations of the inpainting results in

terms of PSNR, SSIM and CPU time, which further verifies the efficiency and superiority of our curvature regularized models in providing significantly better PSNR and SSIM using only 2/3 of the CPU time of the Euler’s elastica model.

5 Conclusions

In this work, we introduced novel curvature regularizers for image reconstruction problems, where both mean curvature and Gaussian curvature were derived and investigated using the normal curvatures according to differential geometry theory. The proposed curvature regularized model was regarded as a re-weighted image surface minimization model and efficiently solved by the ADMM-based algorithm. We discussed the existence of the minimizer and provided the convergence analysis for the proposed ADMM-based algorithm. Numerical experiments on both gray and color images have illustrated the efficacious and superior performance of our proposed method in terms of quantitative and qualitative evaluations. The proposed method can be used for other practical applications in image processing and computer vision, for instance, image segmentation, image registration, image super-resolution, etc.

Acknowledgements The authors would like to thank Dr. Gong and Prof. Sbalzarini for sharing the MATLAB code of curvature filter and the referees for providing us numerous valuable suggestions to revise the paper. The work was partially supported by National Natural Science Foundation of China (NSFC 12071345, 11701418), Major Science and Technology Project of Tianjin 18ZXRHSY00160 and Recruitment Program of Global Young Expert. The second author was supported by NSFC 11801200 and a startup Grant from HUST.

References

1. Ambrosio, L., Masnou, S.: A direct variational approach to a problem arising in image reconstruction. *Interfaces Free Boundaries* **5**(1), 63–81 (2003)
2. Bae, E., Shi, J., Tai, X.C.: Graph cuts for curvature based image denoising. *IEEE Trans. Image Process.* **20**(5), 1199–1210 (2011)
3. Bae, E., Tai, X.C., Zhu, W.: Augmented Lagrangian method for an Euler’s elastica based segmentation model that promotes convex contours. *Inverse Probl. Imaging* **11**(1), 1–23 (2017)
4. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Process.* **10**(8), 1200–1211 (2001)
5. Blomgren, P., Chan, T.: Color TV: total variation methods for restoration of vector-valued images. *IEEE Trans. Image Process.* **7**(3), 304–309 (1998)
6. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **146**(1–2), 459–494 (2013)
7. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. *SIAM J. Imaging Sci.* **3**(3), 492–526 (2010)
8. Bredies, K., Pock, T., Wirth, B.: A convex, lower semicontinuous approximation of Euler’s elastica energy. *SIAM J. Math. Anal.* **47**(1), 566–613 (2015)

9. Brito-Loeza, C., Chen, K., Uc-Cetina, V.: Image denoising using the Gaussian curvature of the image surface. *Numer. Methods Partial Differ. Equ.* **32**(3), 1066–1089 (2016)
10. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
11. Chambolle, A., Pock, T.: Total roto-translational variation. *Numerische Mathematik* **142**(3), 611–666 (2019)
12. Chan, T.F., Jianhong, S.: *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic*. Society for Industrial and Applied Mathematics, Philadelphia (2005)
13. Ciomaga, A., Monasse, P., Morel, J.M.: The image curvature microscope: accurate curvature computation at subpixel resolution. *Image Process. Line* **7**, 197–217 (2017)
14. Cui, Y., Li, X., Sun, D., Toh, K.C.: On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions. *J. Optim. Theory Appl.* **169**(3), 1013–1041 (2016)
15. Deng, L.J., Glowinski, R., Tai, X.C.: A new operator splitting method for the Euler elastica model for image smoothing. *SIAM J. Imaging Sci.* **12**(2), 1190–1230 (2019)
16. Deng, W., Yin, W.: On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.* **66**(3), 889–916 (2015)
17. Dong, Y., Hintermüller, M., Neri, M.: An efficient primal-dual method for L_1 TV image restoration. *SIAM J. Imaging Sci.* **2**(4), 1168–1189 (2009)
18. Duan, Y., Huang, W., Zhou, J., Chang, H., Zeng, T.: A two-stage image segmentation method using Euler’s elastica regularized Mumford-Shah model. In: *Proceedings of the IEEE International Conference on Pattern Recognition*, pp. 118–123 (2014)
19. Duan, Y., Wang, Y., Hahn, J.: A fast augmented Lagrangian method for Euler’s elastica models. *Numer. Math. Theory Methods Appl.* **6**(1), 47–71 (2013)
20. El-Fallah, A., Ford, G.: Mean curvature evolution and surface area scaling in image filtering. *IEEE Trans. Image Process.* **6**(5), 750–753 (1997)
21. El-Zehiry, N.Y., Grady, L.: Fast global optimization of curvature. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3257–3264 (2010)
22. Esedoglu, S., March, R.: Segmentation with depth but without detecting junctions. *J. Math. Imaging Vis.* **18**(1), 7–15 (2003)
23. Goldluecke, B., Cremers, D.: Introducing total curvature for image processing. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1267–1274 (2011)
24. Goldstein, T., Osher, S.: The split Bregman method for L_1 -regularized problems. *SIAM J. Imaging Sci.* **2**(2), 323–343 (2009)
25. Gong, Y., Sbalzarini, I.F.: Local weighted Gaussian curvature for image processing. In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 534–538 (2013)
26. Gong, Y., Sbalzarini, I.F.: Curvature filters efficiently reduce certain variational energies. *IEEE Trans. Image Process.* **26**(4), 1786–1798 (2017)
27. Huang, Y., Ng, M.K., Wen, Y.: A new total variation method for multiplicative noise removal. *SIAM J. Imaging Sci.* **2**(1), 20–40 (2009)
28. Kang, S.H., Zhu, W., Jianhong, J.: Illusory shapes via corner fusion. *SIAM J. Imaging Sci.* **7**(4), 1907–1936 (2014)
29. Le, T., Chartrand, R., Asaki, T.J.: A variational approach to reconstructing images corrupted by Poisson noise. *J. Math. Imaging Vis.* **27**(3), 257–263 (2007)
30. Lee, S.H., Seo, J.K.: Noise removal with Gauss curvature-driven diffusion. *IEEE Trans. Image Process.* **14**(7), 904–909 (2005)
31. Liu, J., Yan, M., Zeng, T.: Surface-aware blind image deblurring. *IEEE Trans. Pattern Anal. Mach. Intell.* **99**, 1–1 (2019)
32. Lu, B., Wang, H., Lin, Z.: High order Gaussian curvature flow for image smoothing. In: *Proceedings of the IEEE International Conference on Multimedia Technology*, pp. 5888–5891 (2011)
33. Ma, Q., Peng, J., Kong, D.: Image segmentation via mean curvature regularized Mumford–Shah model and thresholding. *Neural Process. Lett.* **48**(2), 1227–1241 (2017)
34. Mairal, J., Elad, M., Sapiro, G.: Sparse representation for color image restoration. *IEEE Trans. Image Process.* **17**(1), 53–69 (2007)
35. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: *Proceedings of the IEEE International Conference on Image Processing*, pp. 259–263 (1998)
36. Mumford, D.: *Elastica and Computer Vision*, pp. 491–506. Springer, Berlin (1994)
37. Myllykoski, M., Glowinski, R., Karkkainen, T., Rossi, T.: A new augmented Lagrangian approach for l_1 -mean curvature image denoising. *SIAM J. Imaging Sci.* **8**(1), 95–125 (2015)
38. Nikolova, M.: A variational approach to remove outliers and impulse noise. *J. Math. Imaging Vis.* **20**(1–2), 99–120 (2004)
39. Nitzberg, M., Mumford, D., Shiota, T.: *Filtering, Segmentation and Depth*. Springer, Berlin (1993)
40. Rosman, G., Dascal, L., Tai, X.C., Kimmel, R.: On semi-implicit splitting schemes for the Beltrami color image filtering. *J. Math. Imaging Vis.* **40**(2), 199–213 (2011)
41. Rosman, G., Tai, X.C., Dascal, L., Kimmel, R.: Polyakov action minimization for efficient color image processing. In: *Proceedings of the Springer European Conference on Computer Vision*, pp. 50–61 (2010)
42. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D Nonlinear Phenom.* **60**(1–4), 259–268 (1992)
43. Schoenemann, T., Kahl, F., Cremers, D.: Curvature regularity for region-based image segmentation and inpainting: a linear programming relaxation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 17–23 (2009)
44. Shefi, R., Teboulle, M.: Rate of convergence analysis of decomposition methods based on the proximal method of multipliers for convex minimization. *SIAM J. Optim.* **24**(1), 269–297 (2014)
45. Shen, J., Kang, S.H., Chan, T.F.: Euler’s elastica and curvature-based inpainting. *SIAM J. Appl. Math.* **63**(2), 564–592 (2003)
46. Sochen, N., Kimmel, R., Malladi, R.: A general framework for low level vision. *IEEE Trans. Image Process.* **7**(3), 310–318 (1998)
47. Tai, X.C., Hahn, J., Chung, G.J.: A fast algorithm for Euler’s elastica model using augmented Lagrangian method. *SIAM J. Imaging Sci.* **4**(1), 313–344 (2011)
48. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
49. Wen, Y.W., Ng, M.K., Huang, Y.M.: Efficient total variation minimization methods for color image restoration. *IEEE Trans. Image Process.* **17**(11), 2081–2088 (2008)
50. Wu, C., Tai, X.C.: Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models. *SIAM J. Imaging Sci.* **3**(3), 300–339 (2010)
51. Wu, C., Zhang, J., Tai, X.C.: Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Probl. Imaging* **5**(1), 237–261 (2011)
52. Yan, M., Duan, Y.: Nonlocal elastica model for sparse reconstruction. *J. Math. Imaging Vis.* **62**(4), 532–548 (2020)
53. Yang, J., Zhang, Y., Yin, W.: An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM J. Sci. Comput.* **31**(4), 2842–2865 (2009)
54. Yashtini, M., Kang, S.H.: A fast relaxed normal two split method and an effective weighted TV approach for Euler’s elastica image inpainting. *SIAM J. Imaging Sci.* **9**(4), 1552–1581 (2016)
55. Yezzi, A.: Modified curvature motion for image smoothing and enhancement. *IEEE Trans. Image Process.* **7**(3), 345–352 (1998)

56. Zhang, H., Wang, L., Duan, Y., Li, L., Hu, G., Yan, B.: Euler's elastica strategy for limited-angle computed tomography image reconstruction. *IEEE Trans. Nuclear Sci.* **64**(8), 2395–2405 (2017)
57. Zhong, Q., Li, Y., Yang, Y., Duan, Y.: Minimizing discrete total curvature for image processing. In: *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9474–9482 (2020)
58. Zhu, W., Chan, T.: Image denoising using mean curvature of image surface. *SIAM J. Imaging Sci.* **5**(1), 1–32 (2012)
59. Zhu, W., Chan, T., Esedolu, S.: Segmentation with depth: a level set approach. *SIAM J. Sci. Comput.* **28**(5), 1957–1973 (2006)
60. Zhu, W., Tai, X.C., Chan, T.: Augmented Lagrangian method for a mean curvature based image denoising model. *Inverse Probl. Imaging* **7**(4), 1409–1432 (2013)
61. Zhu, W., Tai, X.C., Chan, T.: Image segmentation using Euler's elastica as the regularization. *J. Sci. Comput.* **57**(2), 414–438 (2013)

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

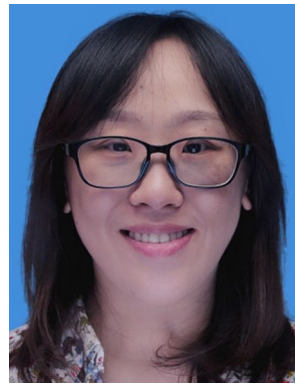


Qiuxiang Zhong received the M.D. degree in mathematics from the Wuhan University of Technology in 2018. He is currently pursuing the Ph.D. degree in Center for applied Mathematics at Tianjin University. His research interests include variation image processing, mathematical imaging, computer vision, and intelligent computing.



ing, optimization and machine learning.

Ke Yin received Bachelor's degree from University of Science and Technology of China (USTC) in 2008 and Ph.D. degree in Mathematics from Georgia Institute of Technology in 2013. Then he was an adjunct assistant professor in the Department of Mathematics at UCLA from 2013–2016. He joined Center for Mathematical Sciences at Huazhong University of Science and Technology in China as associate professor in 2016. His research interests include inverse problems, image processing,



chine learning.

Yuping Duan is a professor at Center for Applied Mathematics, Tianjin University, China. She received her Ph.D. degree from the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore in 2012. From January 2012 to September 2015, she worked as a research scientist at Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore. Her research interests include mathematical image processing, inverse problems and machine learning.