Elastica Models for Color Image Regularization

Hao Liu *, Xue-Cheng Tai †, Ron Kimmel ‡, Roland Glowinski §

In memory of Roland Glowinski—a dear friend, mentor, colleague and great leader.

Abstract

The choice of a proper regularization measure plays an important role in the field of image processing. One classical approach treats color images as two dimensional surfaces embedded in a five dimensional spatial-chromatic space. In this case, a natural regularization term arises as the image surface area. Choosing the chromatic coordinates as dominating over the spatial ones, the image spatial coordinates could be thought of as a parameterization of the image surface manifold in a three dimensional color space. Minimizing the area of the image manifold leads to the Beltrami flow or mean curvature flow of the image surface in the 3D color space, while minimizing the elastica of the image surface yields an additional interesting regularization. Recently, the authors proposed a color elastica model, which minimizes both the surface area and elastica of the image manifold. In this paper, we propose to modify the color elastica and introduce two new models for color image regularization. The revised measures are motivated by the relations between the color elastica model, Euler’s elastica model and the total variation model for gray level images. Compared to our previous color elastica model, the new models are direct extensions of Euler’s elastica model to color images. The proposed models are nonlinear and challenging to minimize. To overcome this difficulty, two operator-splitting methods are suggested. Specifically, nonlinearities are decoupled by introducing new vector- and matrix-valued variables. Then, the minimization problems are converted to solving initial value problems which are time-discretized by operator splitting. Each subproblem, after splitting, either has a closed-form solution or can be solved efficiently. The effectiveness and advantages of the proposed models are demonstrated by comprehensive experiments. The benefits of incorporating the elastica of the image surface as regularization terms compared to common alternatives are empirically validated.

*Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. Email: haoliu@hkbu.edu.hk.

†Hong Kong Center for Cerebro-Cardiovascular Health Engineering (COCHE), Hong Kong Science Park, Hong Kong. Email: xta@hkcoche.org, xuechengtai@gmail.com.

‡Computer Science Department, and Electrical and Computer Engineering Department, Technion, Haifa, Israel. Email: ron@cs.technion.ac.il.

§The author is deceased. Former address: Department of Mathematics, University of Houston, Houston, TX 77204, USA, and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong.
1 Introduction

Image regularization is a fundamental topic in image processing, and appears in almost every tasks in this field. In past decades, tremendous efforts have been devoted to looking for good image regularizers, while most of which focus on gray-scale images. As one important way to describe the magnificence of the world is through colors, good regularizers for color images is of high demand.

In the literature, image regularization for gray-scale images has been extensively studied. Given a gray-scale image $v$, a well-known regularizer is the total variation (TV) \[43\], given by

$$\int_{\Omega} |\nabla v| \, dx,$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain and $dx = dx_1dx_2$ with $x_1, x_2$ being the coordinates of a point $x$ in $\Omega$. TV is a first order regularizer since it only depends on the first order partial derivatives of $v$. It is known in preserving sharp changes in the gradient of the image. Fast algorithms for TV based image regularization models can be found in \[9, 8, 22\]. One drawback of the TV model is that it suffers from the staircase effects. To overcome this obstacle, high order regularizers were explored. One of the most popular regularizers is realized by minimizing Euler’s elastica energy

$$\int_{\Omega} \left( a + b \left( \nabla \cdot \frac{\nabla v}{|\nabla v|} \right)^2 \right) |\nabla v| \, dx,$$

(1)

where $a, b \geq 0$ are weight parameters. For gray-level images, Euler’s elastica treats the image as a function and minimizes the length and curvature of each of its level curves. Another perspective of (1) is that it penalizes both the TV and the variation of the TV. This observation provides a new perspective on defining the ‘Sobolev space’ for images. From this point of view, TV is a first order model and Euler’s elastica is a second order model, since the former and the latter one penalize the first and second variation of gray images, respectively. Due to the superior performance of (1) in various image processing models, designing efficient algorithms for Euler’s elastica based models has been a popular topic. Efforts in that direction include augmented Lagrangian multipliers based methods \[13, 48, 59, 33, 56, 12\], and split Bregmann method \[57\]. Recently, an operator splitting method was proposed in \[10\]. Unlike previous methods, this method is insensitive to the choice of parameters and is almost parameters free. We refer the readers to \[24\] for a survey on fast algorithms for Euler’s elastica based models in image inpainting.

Color images can be thought of as vector-valued signals with $m$ chromatic channels. One simple way to process a color image is to apply gray-scale image regularizers channel by channel. However, this way, the interactions between channels are ignored. In literature, regularizer models and fast algorithms that treat to color images are limited. As a generalization of the scalar TV, \[5\] proposed the color TV which is the square root of the sum of squared TV of each channel. The authors of \[49\] proposed a total curvature model in which the color TV in \[5\] was replaced by the sum of squared level set curvature of each channel. In the geometric point of view, inspired by the discussion of tensor gradient for vector-valued images in \[11\] in which a color image is considered as a two dimensional manifold in $\mathbb{R}^m$, \[45\] proposed an anisotropic diffusion framework and \[53\] suggested an edge-enhancing diffusion method. Based on the framework of \[11\], another generalization of the scalar TV, known as the vectorial TV (VTV), was proposed by \[21\]. Relating to the framework proposed in \[44\], VTV penalizes the largest singular value of the Jacobian of the color image (a $m \times 2$ matrix) on each pixel in its domain. Efficient algorithms for this family of regularizers are studied in \[6, 15\]. Algorithms dedicated to color image enhancement are studied in \[3, 4, 35, 34, 37, 50\].
Another geometry-based regularizer for color images is the Beltrami framework, which was proposed in [25, 46] and further investigated in [26, 47, 52, 42, 54]. In this framework, a color image is considered as a two-dimensional manifold embedded in the $m + 2$ dimensional space-feature space. The Beltrami framework minimizes the Polyakov action [38], which is a functional that measures the surface area of the surface. Its first variation gradient flow leads to a Beltrami flow. It was shown that at its limit, the Polyakov action reduces to TV model for gray-scale images. Fast algorithms for the Beltrami framework have been developed in [2, 39, 40, 41, 60].

Most of the aforementioned color image regularizers are first order, which may not be rich enough to capture image properties. Recently, based on the Beltrami framework, the authors have proposed a second order regularizer, the color elastica model [30]. The color elastica model is an extension of the Beltrami framework and penalizes both the Polyakov action and the Beltrami flow, the second term being the norm of the Laplace-Beltrami operator acting on the image coordinates. In the limit, it would be nothing but the square of mean curvature of the color image surface embedded in the $\mathbb{R}^3$ chromatic space. In our setting, the color elastica model involves a parameter $\alpha$, which controls the weight between spatial coordinates and feature (chromatic or color) coordinates. The color elastica model generalizes (1) to color images, as for gray-scale images it reduces to a weighted Euler’s elastica model by letting $\alpha$ go to zero. However, the weight in the reduced model is $1/|\nabla v|$, which may be challenging to handle when $\nabla v$ vanishes. The second drawback is that the algorithm proposed in [30] converges very slowly when $\alpha$ is small.

In this article, we propose two modified color elastica models, which are more natural extensions of (1) for color images. In the first model, we add a weight to the Laplace-Beltrami term so that the model exactly reduces to (1) as $\alpha \to 0$. The second model is based on the first one and the relation between TV and surface area of gray-scale images. The second model boils down directly to (1) for gray-scale images. The proposed models contain nonlinear functionals that are difficult to minimize. Two operator-splitting methods are designed which solve the proposed models efficiently. Operator-splitting methods are known for decomposing complicated problems into several easy-to-solve subproblems and have been applied in numerical PDEs [18, 16, 28], inverse problems [17], obstacle problem [31], fluid-structure interactions [7] and recently in image processing [10, 30, 23, 14]. In our proposed algorithms, we decouple the nonlinearity by introducing new vector- and matrix-valued variables. Then, minimizing the functionals is converted to solving initial-value problems until a steady state is reached. The initial-value problems are time-discretized by operator splitting methods such that each sub-problem either has a closed form solution or can be solved efficiently.

This article is structured as follows: We provide motivation and formulations of the proposed models in Section 2. In Section 3, we present our operator-splitting schemes and discuss the solution to each subproblem. The proposed operator-splitting methods are space discretized in Section 4. We empirically justify the proposed models in Section 5. We demonstrate the efficiency and performance of the proposed algorithms and models in Section 6 by comprehensive numerical experiments. This article is concluded in Section 7.
Problem formulation

2.1 Motivation towards the proposed models

In image processing, the image surface area [46, 58, 32] and total variation [43] are two popular regularizers for image restoration. We motivate our construction by reviewing the links between these models. Let $\Omega$ be a rectangular domain with coordinates $x_1, x_2$. Any grayscale image $f$ can be considered as a two-dimensional surface embedded in the three-dimensional space-feature space, $F(x_1, x_2) = (\sqrt{\alpha}x_1, \sqrt{\alpha}x_2, f(x_1, x_2))$, where $\alpha$ is a parameter controlling the weight of spatial coordinates. Under such a parameterization, the metric on the image manifold is $g = \det(G)$, where

$$G = \begin{pmatrix} \alpha + (\partial_1 f)^2 & \partial_1 f \partial_2 f \\ \partial_1 f \partial_2 f & \alpha + (\partial_2 f)^2 \end{pmatrix}$$

(2)

with $\partial_1 f = \partial f / \partial x_1$, $\partial_2 f = \partial f / \partial x_2$. From this metric, the surface area of $f$ can be computed as

$$S(f) = \int_{\Omega} \sqrt{g} \, dx = \int_{\Omega} \sqrt{\alpha^2 + \alpha[(\partial_1 f)^2 + (\partial_2 f)^2]} \, dx.$$  

(3)

At the other end, the total variation of $f$ is given by

$$TV(f) = \int_{\Omega} \sqrt{(\partial_1 f)^2 + (\partial_2 f)^2} \, dx.$$  

(4)

Comparing the right-hand side of (3) and (4), we observe that $TV(f)$ can be recovered by replacing $g$ in $S(f)$ by $g - \alpha^2$,

$$\int_{\Omega} \sqrt{g - \alpha^2} \, dx = \sqrt{\alpha} TV(f).$$  

(5)

This observation would guide us in our exploration of a modified color elastica model.

Recall that the color elastica model proposed in [30] takes an RGB image as a two dimensional surface embedded in the five dimensional space-feature space. The model is given as

$$\min_{v \in \mathcal{H}^2(\Omega)} \int_{\Omega} \left[1 + \beta \sum_{k=1}^{3} |\Delta_g v_k|^2\right] \sqrt{g} \, dx + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |v_k - f_k|^2 \, dx,$$

(6)

where $\beta > 0, \eta > 0$ are weight parameters, and $\mathcal{H}^2$ is the Sobolev space defined as

$$\mathcal{H}^2(\Omega) = \{ v | v \in \mathcal{L}^2(\Omega), \nabla v \in (\mathcal{L}^2(\Omega))^2, Dv \in (\mathcal{L}^2(\Omega))^{2 \times 2} \},$$

where $D$ denotes the Hessian and the derivatives being in the weak sense. In (6), $\Delta_g$ is the determinant of the manifold metric defined by, $g = \det G$, where $G = (g_{ij})_{1 \leq i,j \leq 2}$ and

$$g_{11} = \alpha + \sum_{k=1}^{3} \frac{\left| \frac{\partial v_k}{\partial x_1} \right|^2}{\partial x_1}, \quad g_{12} = g_{21} = \sum_{k=1}^{3} \frac{\partial v_k \partial v_k}{\partial x_1 \partial x_2}, \quad g_{22} = \alpha + \sum_{k=1}^{3} \frac{\left| \frac{\partial v_k}{\partial x_2} \right|^2}{\partial x_2},$$

with $\alpha > 0$. Whenever $\alpha > 0$, $G$ is positive definite. In (6), $\Delta_g$ is the Laplace-Beltrami operator associated with matrix $G$,

$$\Delta_g \phi = \frac{1}{\sqrt{g}} \nabla \cdot (\sqrt{g} G^{-1} \nabla \phi), \quad \forall \phi \in V.$$  

(7)
The second term in (6) is called the color elastica term, which captures the elastica of images in the chromatic space. The color elastica term is the variation of the Polyakov action [25, 46] given by
\[ \int_{\Omega} \sqrt{g} dx. \]

From the perspective of ‘Sobolev space’ of images as discussed in the introduction, the Polyakov action is a first order model and the color elastica model (6) is a second order model for color images.

We have shown in [30, Remark 3.2] that for the single-channel case, when \( \alpha \to 0 \), the color elastica model (6) reduces to a variant of Euler’s elastica model: the term \( \left( \nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \) is weighted by \( 1/|\nabla v|^2 \), see Appendix A for details. Since Euler’s elastica model have demonstrated impressive performance in processing single-channel images, we would like to derive multi-channel image models that are more direct extensions of Euler’s elastica model. As mentioned above, there are two drawbacks of (6) when connecting it with Euler’s elastica model: (i) we need \( \alpha \to 0 \), and (ii) the resulting model differs from Euler’s elastica model by a factor. Next, we introduce our new models by making two modifications that provide a remedy to each drawback.

The first modification targets drawback (ii). We weight the second term, the color elastica term, in (6) by \( g \) to get,
\[ \int_{\Omega} \left[ 1 + \beta \sum_{k=1}^{3} g |\Delta g v_k|^2 \right] \sqrt{g} dx + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |v_k - f_k|^2 dx, \]
(8)

This modification modulates the color elastica term by the metric \( g \). Due to this additional factor, the new regularizer maybe not coordinate invariant as \( g \) depends on the parametrization of the manifold. Compared to (6), the gradient flow of (8) has a preconditioning that amplifies the action along edges in the image. With this modification, model (8) reduces to Euler’s elastica model as \( \alpha \to 0 \) for one-channel images.

The second modification targets drawback (i) and utilizes relation (5). First note that an alternative expression of \( \Delta_g \phi \) is
\[ \Delta_g \phi = \frac{1}{\sqrt{g}} \nabla \cdot \left( \frac{1}{\sqrt{g}} \text{cof}(G) \nabla \phi \right), \forall \phi \in V, \]
(9)

where \( \text{cof}(G) \) is the cofactor matrix of \( G \). Taking advantage of (5), we replace \( g \) in (8) and (9) by \( g - \alpha^2 \) to get the modified model:
\[ \int_{\Omega} \left[ 1 + \beta \sum_{k=1}^{3} (g - \alpha^2) |\Delta_g v_k|^2 \right] \sqrt{g - \alpha^2} dx + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |v_k - f_k|^2 dx, \]
(10)

where
\[ \tilde{\Delta}_g \phi = \frac{1}{\sqrt{g - \alpha^2}} \nabla \cdot \left( \frac{1}{\sqrt{g - \alpha^2}} \text{cof}(G) \nabla \phi \right), \forall \phi \in V. \]
(11)

In the single-channel case, the functional (10) reduces to
\[ \int_{\Omega} \sqrt{\alpha} \left[ 1 + \beta \left| \nabla \cdot \frac{\nabla v}{|\nabla v|} \right| \right] |\nabla v| dx + \frac{1}{2\eta} \int_{\Omega} |v - f|^2 dx \]
(12)
Figure 1: Comparison of the relative energy of the regularizers in the modified color elastica models (8) and (10), and the original color elastica model (6). The noisy images are generated by adding Gaussian noise with SD varying from 0.01 to 0.1. This test takes the image ’Pens’ in Figure 2 as an example. The considered regularizers are $\mathcal{F}_0$ (in model (6)), $\mathcal{F}_1$ (in model (8)) and $\mathcal{F}_2$ (in model (10)). Each relative energy is averaged over 10 experiments.

which exactly is the Euler’s elastica model without any condition.

In Figure 1, we use a simple example to demonstrate the improvement of the regularizers in model (8) and (10) over that in the original color elastica model (6). We use $\mathcal{F}_0, \mathcal{F}_1$ and $\mathcal{F}_2$ to denote the regularizers in model (6), (8) and (10), respectively. For any noisy image $f$ and its clean version $f_0$, we define the relative energy of regularizer $\mathcal{F}$ by $RF(f) = \mathcal{F}(f)/\mathcal{F}(f_0)$. As the noise level of $f$ increases, the regularizer is more effective if its relative energy increases faster. In Figure 1, we take the clean image ’Pens’ in Figure 11 as an example and compare the plots of relative energies of all three regularizers versus noise level. The new regularizers in model (8) and (10) have faster increasing rates than that in model (6), demonstrating that they are more effective and can better characterize natural images than the regularizer in model (6). We refer readers to Section 5 for more details and comprehensive comparisons with other models.

In the rest of this article, we focus on efficient algorithms to minimize (8) and (10).

**Remark 2.1.** Given a color image $v$, one can show that the corresponding $g$ has the following expression

$$
g = \alpha^2 + \alpha (|\nabla v_1|^2 + |\nabla v_2|^2 + |\nabla v_3|^2)
+ (\det(\nabla v_1; \nabla v_2))^2 + (\det(\nabla v_1; \nabla v_3))^2 + (\det(\nabla v_2; \nabla v_3))^2.
$$

(13)

In (13), the first two terms have $\alpha$ as a factor and the first term is a lower bound of $g$. The second term is the sum of squared total variation of each channel, which is called color TV in [5]. The third term is independent of $\alpha$ and describes the interactions between channels. Therefore, larger $\alpha$ gives more weights to the color TV term and less weight to the inter-channel interaction term. Compared to (8), the measure used in (10) shifts $g$ so that its lower bound is zero.

### 2.2 Reformulation of (8) and (10)

The modified models (8) and (10) are nonlinear functionals and are difficult to minimize. To develop efficient algorithms to find their minimizers, we decouple the nonlinearities
in the Laplace-Beltrami operator by introducing several vector-valued and matrix-valued variables. For \( k = 1, 2, 3 \) and \( r = 1, 2 \), let us denote by \( q_{kr} \) the real valued function \( \frac{\partial q_r}{\partial x_k} \) and by \( q \) the \( 3 \times 2 \) matrix

\[
q = \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \\ q_{31} & q_{32} \end{pmatrix} = \nabla v, \text{ with } v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.
\]

Denote \( q_k = (q_{k1} \quad q_{k2}) \), \( k = 1, 2, 3 \), we introduce the \( 3 \times 2 \) matrix \( \mu = \sqrt{g}qG^{-1} \) with

\[
\mu_k = \sqrt{g}q_k G^{-1}
\]

and \( \nu = \frac{1}{\sqrt{g - \alpha^2}}q \text{cof}(G) \) with

\[

\nu_k = \frac{1}{\sqrt{g - \alpha^2}}q_k \text{cof}(G).
\]

Here \( \mu \) and \( \nu \) are proxies of the parts inside the divergence in (9) and (11), respectively. Expressions (14) and (15) imply

\[
q_k = \frac{1}{\sqrt{g}}\mu_k G \quad \text{and} \quad \sqrt{g - \alpha^2} \nu_k = q_k \text{cof}(G).
\]

We denote by \( M(q) \) the matrix-valued function defined by

\[
\begin{pmatrix} \alpha + q_{11}^2 + q_{21}^2 + q_{31}^2 & q_{11}q_{12} + q_{21}q_{22} + q_{31}q_{32} \\ q_{11}q_{12} + q_{21}q_{22} + q_{31}q_{32} & \alpha + q_{12}^2 + q_{22}^2 + q_{32}^2 \end{pmatrix}
\]

and denote \( \det M(q) \) by \( m(q) \). Define the sets \( \Sigma_f, S_G, \tilde{S}_G \) as

\[
\Sigma_f = \left\{ q \in (L^2(\Omega))^{3 \times 2}, \exists v \in H^1(\Omega) \text{ such that } q = \nabla v \text{ and } \int_\Omega v_k - f_k dx = 0 \text{ for } k = 1, 2, 3 \right\},
\]

\[
S = \left\{ (q, \mu) \in ((L^2(\Omega))^{3 \times 2}, (L^2(\Omega))^{3 \times 2}), \mu_k = \sqrt{\det M(q)q_k (M(q))^{-1}}, k = 1, 2, 3 \right\},
\]

\[
\tilde{S} = \left\{ (q, \nu) \in ((L^2(\Omega))^{3 \times 2}, (L^2(\Omega))^{3 \times 2}), \sqrt{\det M(q) - \alpha^2} \nu_k = q_k \text{cof}(M(q)), k = 1, 2, 3 \right\},
\]

and their indicator functions as

\[
I_{\Sigma_f}(q) = \begin{cases} 0 & \text{if } q \in \Sigma_f, \\ +\infty & \text{otherwise,} \end{cases}
\]

\[
I_S(q, \mu) = \begin{cases} 0 & \text{if } (q, \mu) \in S, \\ +\infty & \text{otherwise,} \end{cases}
\]

\[
I_{\tilde{S}}(q, \nu) = \begin{cases} 0 & \text{if } (q, \nu) \in \tilde{S}, \\ +\infty & \text{otherwise.} \end{cases}
\]

Let \( (p, \lambda) \) be the minimizer of

\[
\min_{q \in (H^1(\Omega))^{3 \times 2}, \mu \in (H^1(\Omega))^{3 \times 2}} \int_\Omega \left( \frac{1}{2} \sum_{k=1}^3 |\nabla \cdot \mu_k|^2 \right) \sqrt{m(q)}dx + \frac{1}{2\eta} \sum_{k=1}^3 \int \left( |v_k - f_k|^2 \right) dx + I_{\Sigma_f}(q) + I_S(q, \mu),
\]

\[
(17)
\]
where $v_q = \{(v_q)_k\}_{k=1}^3$ is the solution of

$$
\begin{cases}
\nabla^2(v_q)_k = \nabla \cdot q_k, \\
(\nabla(v_q)_k - q_k) \cdot n = 0, \\
\int_\Omega (v_q)_k dx = \int_\Omega f_k dx, \\
k = 1, 2, 3.
\end{cases}
$$ (18)

Then, $u_p$ solving (18) is the minimizer of (8). In (17), $v_q$ can be uniquely determined by $q$. The complicated Laplace-Beltrami operator is represented using the divergence of $\mu$. The resulting formulation is an unconstrained optimization problem of $q$ and $\mu$ only.

Similarly, if $(p, \lambda)$ be the minimizer of

$$
\min_{q \in (H^1(\Omega))^3 \times 2} \int_\Omega \left( 1 + \sum_{k=1}^3 |\nabla \cdot \nu_k|^2 \right) \sqrt{m(q)} - \alpha^2 dx + \frac{1}{2\eta} \sum_{k=1}^3 \int_\Omega |(v_q)_k - f_k|^2 dx + I_{\Sigma,f}(q) + I_{S}(q, \nu).
$$ (19)

Then $u_p$ solving (18) is the minimizer of (10).

## 3 Operator splitting methods

In this section, we derive operator-splitting schemes to solve (17) and (20). We will first derive the Euler-Lagrange equations of both functional and associate them with initial-value problems, which are suitable to be solved by operator-splitting methods.

### 3.1 Optimal conditions of (17) and (20)

Define

$$
J_1(q, \mu) = \int_\Omega \left( 1 + \beta \sum_{k=1}^3 |\nabla \cdot \mu_k|^2 \right) \sqrt{m(q)} dx,
$$ (21)

$$
\tilde{J}_1(q, \nu) = \int_\Omega \left( 1 + \beta \sum_{k=1}^3 |\nabla \cdot \nu_k|^2 \right) \sqrt{m(q)} - \alpha^2 dx,
$$ (22)

$$
J_2(q) = \frac{1}{2\eta} \sum_{k=1}^3 \int_\Omega |(v_q)_k - f_k|^2 dx.
$$ (23)

If $(p, \lambda)$ is the minimizer of (17), it satisfies

$$
\begin{cases}
\partial_q J_1(p, \lambda) + D_q J_2(p) + \partial_q I_{\Sigma,f}(p) + \partial_q I_S(p, \lambda) \ni 0, \\
D_\mu J_1(p, \lambda) + \partial_\mu I_S(p, \lambda) \ni 0,
\end{cases}
$$ (24)

where $D_q$ (resp. $\partial_q$) denotes the partial derivative (resp. subdifferential) of a differentiable function (resp. non-differentiable function) with respect to $q$. 

8
Remark 3.1. Although the set $S$ may be nonconvex, which makes $I_S$ nonconvex, $I_S$ is an indicator function with special properties. Actually, the subdifferential of $I_S(q, \mu)$ exists for any $(q, \mu) \in S$, and $(0, 0)$ is an element of it. Let $(q^*, \mu^*)$ be any element of $S$, we have $I_S(q^*, \mu^*) = 0$ and

$$I_S(q, \mu) \geq 0 = I_S(q^*, \mu^*) + \langle 0, q - q^* \rangle + \langle 0, \mu - \mu^* \rangle,$$

for any $(q, \mu)$ (not necessarily in $S$). By definition, the subdifferential of $I_S$ at $(q^*, \mu^*)$ exists, and $(0, 0)$ is an element of it.

In (24), $(p, \lambda)$ is a minimizer of (17). We have $(p, \lambda) \in S$ and $(0, 0)$ an element of the subdifferential of $I_S$ at $(p, \lambda)$, which proves (24).

We then introduce an artificial time and associate (24) with the following initial-value problem

$$
\begin{align*}
\frac{\partial p}{\partial t} + \partial_q J_1(p, \lambda) + D_0 J_2(p) + \partial_q I_S(p, \lambda) &\geq 0, \\
\gamma_1 \frac{\partial \lambda}{\partial t} + D_\mu J_1(p, \lambda) + \partial_\mu I_S(p, \lambda) &\geq 0, \\
(p(0), \lambda(0)) &\equiv (p^0, \lambda^0),
\end{align*}
$$

where $\gamma_1$ is a positive constant controlling the evolution speed of $\lambda$, respectively. In (25), $(p^0, \lambda^0)$ is the initial condition whose choice will be discussed in Section 3.8. Note that the steady state solution of (25) solves (17).

Similarly, for problem (20), we solve for the steady state solution of the following initial-value problem

$$
\begin{align*}
\frac{\partial p}{\partial t} + \partial_q \tilde{J}_1(p, \lambda) + D_0 J_2(p) + \partial_q I_{\tilde{S}}(p, \lambda) &\geq 0, \\
\gamma_1 \frac{\partial \lambda}{\partial t} + D_\mu \tilde{J}_1(p, \lambda) + \partial_\mu I_{\tilde{S}}(p, \lambda) &\geq 0, \\
(p(t^n), \lambda(t^n)) &\equiv (p^n, \lambda^n),
\end{align*}
$$

The argument in Remark 3.1 also applies to $I_{\tilde{S}}$, which validates (26).

### 3.2 Operator-splitting schemes

The expression of (25) and (26) are well suited to be time discretized by operator-splitting methods, as what has been done in [10, 30, 29]. We refer the readers to [19] for a complete discussion of operator-splitting methods. A simple choice is the Lie scheme [20]. In the following, we use $\tau$ to denote the time step. For (25), we update $p, \lambda$ as follows:

**Initialization**

Initialize $p^0, \lambda^0$. (27)

**Fractional step 1**

Solve

$$
\begin{align*}
\frac{\partial p}{\partial t} + \partial_q J_1(p, \lambda) &\geq 0, \\
\gamma_1 \frac{\partial \lambda}{\partial t} + D_\mu J_1(p, \lambda) &\geq 0, \\
(p(t^n), \lambda(t^n)) &\equiv (p^n, \lambda^n),
\end{align*}
$$

and set $p^{n+1/3} = p(t^{n+1}), \lambda^{n+1/3} = \lambda(t^{n+1})$.

**Fractional step 2**

Solve

$$
\begin{align*}
\frac{\partial p}{\partial t} + \partial_q I_S(p, \lambda) &\geq 0, \\
\gamma_1 \frac{\partial \lambda}{\partial t} + D_\mu I_S(p, \lambda) &\geq 0, \\
(p(t^n), \lambda(t^n)) &\equiv (p^{n+1/3}, \lambda^{n+1/3}),
\end{align*}
$$

(29)
and set \( p^{n+2/3} = p(t^{n+1}), \lambda^{n+2/3} = \lambda(t^{n+1}) \).

**Fractional step 3**

Solve

\[
\begin{cases}
\frac{\partial p}{\partial t} + D q J_2(p) + \partial_q I_{\Sigma}(p) \geq 0, \\
\gamma_1 \frac{\partial p}{\partial t} = 0, \\
(p(t^n), \lambda(t^n)) = (p^{n+2/3}, \lambda^{n+2/3}),
\end{cases}
\]

in \( \Omega \times (t^n, t^{n+1}) \),

(30)

and set \( p^{n+1} = p(t^{n+1}), \lambda^{n+1} = \lambda(t^{n+1}) \).

For \( p \) and \( \lambda \) in (28)-(30), we update them by implicit schemes. Specifically, we use the Marchuk-Yanenko type scheme to time discretize (28)-(30): Initialize \((p^0, \lambda^0)\).

For \( n \geq 0 \), we update \((p^n, \lambda^n) \rightarrow (p^{n+1/3}, \lambda^{n+1/3}) \rightarrow (p^{n+2/3}, \lambda^{n+2/3}) \rightarrow (p^{n+1}, \lambda^{n+1}) \) as:

\[
\begin{align*}
\left\{ \frac{p^{n+1/3} - p^n}{\tau} + \partial_q J_1(p^{n+1/3}, \lambda^n) \geq 0, \\
\frac{\gamma_1}{\tau} \lambda^{n+1/3} - \lambda^n + D \mu J_1(p^{n+1/3}, \lambda^{n+1/3}) \geq 0, \\
\frac{p^{n+2/3} - p^{n+1/3}}{\tau} + \partial_q J_5(p^{n+2/3}, \lambda^{n+2/3}) \geq 0, \\
\frac{\gamma_1}{\tau} \lambda^{n+2/3} - \lambda^{n+1/3} + D \mu J_5(p^{n+2/3}, \lambda^{n+2/3}) \geq 0, \\
\frac{p^{n+1} - p^{n+2/3}}{\tau} + D q J_2(p^{n+1}) + \partial q I_{\Sigma}(p^{n+1}) \geq 0, \\
\lambda^{n+1} = \lambda^{n+2/3},
\end{align*}
\]

(31)

Analagously, the operator splitting scheme for (26) is

\[
\begin{align*}
\left\{ \frac{\tilde{p}^{n+1/3} - p^n}{\tau} + \partial_q \tilde{J}_1(p^{n+1/3}, \lambda^n) \geq 0, \\
\frac{\gamma_1}{\tau} \tilde{\lambda}^{n+1/3} - \lambda^n + D \mu \tilde{J}_1(p^{n+1/3}, \lambda^{n+1/3}) \geq 0, \\
\frac{\tilde{p}^{n+2/3} - \tilde{p}^{n+1/3}}{\tau} + \partial_q \tilde{J}_5(p^{n+2/3}, \lambda^{n+2/3}) \geq 0, \\
\frac{\gamma_1}{\tau} \tilde{\lambda}^{n+2/3} - \tilde{\lambda}^{n+1/3} + D \mu \tilde{J}_5(p^{n+2/3}, \lambda^{n+2/3}) \geq 0, \\
\frac{\tilde{p}^{n+1} - \tilde{p}^{n+2/3}}{\tau} + D q \tilde{J}_2(p^{n+1}) + \partial q I_{\Sigma}(p^{n+1}) \geq 0, \\
\tilde{\lambda}^{n+1} = \tilde{\lambda}^{n+2/3},
\end{align*}
\]

(32)

(33)

(34)

(35)

(36)

In the rest of this section, we discuss solutions \((p, \lambda)\) to each of the subproblem in scheme (31)-(33) and (34)-(36).

### 3.3 On the solution of (31)

In (31), \( p^{n+1/3} \) is the minimizer of

\[
p^{n+1/3} = \arg \min_{q \in (L^2(\Omega))^3 \times 2} \left[ \frac{1}{2\tau} \int_{\Omega} |q - p^n|^2 dx + \int_{\Omega} \left( 1 + \beta \sum_{k=1}^{3} |\nabla \cdot \lambda^{n_k}|^2 \right) \sqrt{m(q)} dx \right].
\]

(37)

We suggest to use the fixed point method to solve it. The functional in (37) is in the form of

\[
E_1(q) = \frac{1}{2\tau} \int_{\Omega} |q - p|^2 dx + \int_{\Omega} s \sqrt{m(q)} dx
\]

(38)

with some \( s > 0 \) and \( p \in (L^2(\Omega))^3 \times 2 \).
The first variation of $E_1$ with respect to $q_{kr}, k = 1, 2, 3, r = 1, 2$, is

$$\frac{\partial E_1}{\partial q_{kr}} = \frac{1}{\tau} (q_{kr} - p_{kr}) + \frac{s}{2\sqrt{m(q)}} \frac{\partial m(q)}{\partial q_{kr}}, \quad (39)$$

with

$$\frac{\partial m(q)}{\partial q_{k1}} = 2g_{22}g_{k1} - 2g_{12}g_{k2}, \quad \frac{\partial m(q)}{\partial q_{k2}} = 2g_{11}g_{k2} - 2g_{12}g_{k1} \quad (40)$$

for $k = 1, 2, 3$. In (40), the notation $M(q) = \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix}$ is used. Given an initial guess $q^0$, in the $(\omega + 1)$-th iteration, we freeze the denominator and update $q_k$ by solving for $\frac{\partial E_1}{\partial q_{kr}} = 0$:

$$q_{k1}^{\omega+1} = \frac{\sqrt{m(q^\omega)p_{k1} + s\tau g_{12}^\omega g_{k2}^\omega}}{\sqrt{m(q^\omega) + s\tau g_{22}^\omega}}, \quad q_{k2}^{\omega+1} = \frac{\sqrt{m(q^\omega)p_{k2} + s\tau g_{12}^\omega g_{k1}^\omega}}{\sqrt{m(q^\omega) + s\tau g_{11}^\omega}} \quad (41)$$

We continue updating until $\|q^{\omega+1} - q^*\|_\infty < \xi_1$ for some small $\xi_1 > 0$, where we define $\|q\|_\infty = \max_{k,r} |q_{kr}|$. Then we set $p^{n+1/3} = q^*$ where $q^*$ is the converged variable.

For $\lambda^{n+1/3}$, it is the unique solution to

$$\left\{ \begin{array}{l} \lambda^{n+1/3}_k = (\lambda^{n+1/3}_k)_{k=1}^3 \in (H^1(\Omega))^{3\times 2}, \\
\gamma_1 \int_{\Omega} \lambda^{n+1/3}_k \cdot \mu_k \, dx + 2\beta \tau \int_{\Omega} \sqrt{m(p^{n+1/3})} (\nabla \cdot \lambda^{n+1/3}_k) (\nabla \cdot \mu_k) \, dx = \gamma_1 \int_{\Omega} \lambda^n \cdot \mu_k \, dx, \\
\forall \mu_k \in (H^1(\Omega))^2, k = 1, 2, 3. \end{array} \right. \quad (42)$$

Here $\lambda^{n+1/3}$ is also the weak solution to the linear elliptic Neumann problem

$$\left\{ \begin{array}{l} \gamma_1 \lambda^{n+1/3}_k - 2\beta \tau \nabla \left( \sqrt{m(p^{n+1/3})} (\nabla \cdot \lambda^{n+1/3}_k) \right) = \gamma_1 \lambda^n_k \quad \text{in} \ \Omega, \\
\lambda^{n+1/3}_k \cdot n = 0 \quad \text{on} \ \partial \Omega, \\
k = 1, 2, 3, \end{array} \right. \quad (43)$$

where $n$ denotes the outward normal direction.

In Section 3.5, we introduce a new auxiliary variable $G$ approximating $M(p)$. See Remark 3.3 for another choice to update $\lambda^{n+1/3}$ in (31).

### 3.4 On the solution of (34)

In (34), we can compute $p^{n+1/3}$ in the same way as that in (31), except replacing $m(q)$ in Section 3.3 by $m(q) - \alpha^2$. The updating formula analogous to (41) is

$$q_{k1}^{\omega+1} = \frac{p_{k1} + \frac{s\tau}{\sqrt{m(q^\omega) - \alpha^2 + \varepsilon}} g_{12}^\omega g_{k2}^\omega}{1 + \frac{s\tau}{\sqrt{m(q^\omega) - \alpha^2 + \varepsilon}} g_{22}^\omega}, \quad q_{k2}^{\omega+1} = \frac{p_{k2} + \frac{s\tau}{\sqrt{m(q^\omega) - \alpha^2 + \varepsilon}} g_{12}^\omega g_{k1}^\omega}{1 + \frac{s\tau}{\sqrt{m(q^\omega) - \alpha^2 + \varepsilon}} g_{11}^\omega} \quad (44)$$

In (44), $\varepsilon > 0$ is a small number to avoid division by 0. In our experiments, $\varepsilon = 10^{-3}$ gives fast convergence rate while providing good results.
For $\lambda^{n+1/3}$, follow the derivation Section 3.3, it is the weak solution of

$$
\begin{cases}
\gamma_1 \lambda_k^{n+1/3} - 2\beta \nabla \left( \sqrt{m(p^{n+1/3}) - \alpha^2 (\nabla \cdot \lambda_k^{n+1/3})} \right) = \gamma_1 \lambda_k^n & \text{in } \Omega, \\
\lambda_k^{n+1/3} \cdot n = 0 & \text{on } \partial \Omega,
\end{cases}
$$

(45)

See Remark 3.4 for another option to update $\lambda_k^{n+1/3}$ in (34).

### 3.5 On the solution of (32)

The solution $(p^{n+2/3}, \lambda^{n+2/3})$ in (32) is the minimizer of

$$
(p^{n+2/3}, \lambda^{n+2/3}) = \arg \min_{(q, \mu) \in S} \frac{1}{2} \int_\Omega \left( |q - p^{n+1/3}|^2 + \gamma_1 |\mu - \lambda^{n+1/3}|^2 \right) dx.
$$

(46)

The constraint in the set $S$ is nonlinear in $q$, making (46) difficult to solve. In this paper, instead of directly solving (46), we borrow the idea of sequential quadratic programming (SQP) [36] and replace $S$ by $S_G$ defined by

$$
S_G = \left\{ (q, \mu) \in \left( (L^2(\Omega))^{3x2}, (L^2(\Omega))^{3x2} \right), \mu_k = \sqrt{\det G} q_k G^{-1}, k = 1, 2, 3 \right\}
$$

(47)

for some $G$ close to $M(q^{n+2/3})$. The constraint in $S_G$ is a ‘linearization’ of the constraint in $S$ (in the flavor of the fixed point method). Such a strategy is a variant of the first step of the first order SQP studied in [1] with $x_0 = [p^{n+1/3}, \lambda^{n+1/3}]$.

For a small $\tau$, we expect that $p^{n+1/3}$ is close to $p^{n+2/3}$. Thus $G = M(p^{n+1/3})$ is a natural choice. However, such a choice makes our algorithm unstable. To improve the robustness, we take $G$ as a variable and update it with $p$ during iterations (with damping) while keeping it being close to $M(p)$. Specifically, given an initial value $G^0$, for $k = 1, 2, 3$, every time $p^{n+k/3}$ is computed, we update $G^{n+k/3}$ with damping as

$$
G^{n+k/3} = e^{-\gamma_2 \tau} G^{n+(k-1)/3} + (1 - e^{-\gamma_2 \tau}) M(p^{n+k/3})
$$

(48)

for some $\gamma_2 > 0$. The updating formula (48) is the solution $G(t^{n+1})$ to the differential equation

$$
\frac{\partial G}{\partial t} + \gamma_2 (G - M(p^{n+k/3})) = 0
$$

(49)

given $G(t^n) = G^{n+(k-1)/3}$. In (47), we set $G = G^{n+1/3}$, i.e., the latest $G$ computed using $p^{n+1/3}$. Consequently, (46) is approximated by

$$
(p^{n+2/3}, \lambda^{n+2/3}) = \arg \min_{(q, \mu) \in S_{G^{n+1/3}}} \frac{1}{2} \int_\Omega \left( |q - p^{n+1/3}|^2 + \gamma_1 |\mu - \lambda^{n+1/3}|^2 \right) dx.
$$

(50)

We then focus on the solution of (50). For $(q, \mu) \in S_{G^{n+1/3}}$, we have $\mu_i = \sqrt{g^{n+1/3}} q_i G^{n+1/3}$ with $g^{n+1/3} = \det G^{n+1/3}$. While when $\alpha$ is very close to 0, the matrix $G^{n+1/3}$ may be singular and $g^{n+1/3}$ may equal to 0 at some locations. To avoid computing $(G^{n+1/3})^{-1}$ and $1/\sqrt{g^{n+1/3}}$, we rewrite the relation as

$$
\sqrt{g^{n+1/3}} q = \mu G^{n+1/3},
$$

(51)
and consider the following constrained optimization problem

$$(p^{n+2/3}, \lambda^{n+2/3}) = \arg\min_{(q, \mu)} \int_{\Omega} \left( |q - p^{n+1/3}|^2 + \gamma_1 |\mu - \lambda^{n+1/3}|^2 \right) dx. \quad (52)$$

For simplicity, we temporarily use $g, g_{ij}, p_{kr}, \lambda_{kr}$ to denote $g^{n+1/3}, g_{ij}^{n+1/3}, p_{kr}^{n+1/3}, \lambda_{kr}^{n+1/3}$ in this subsection. The explicit solution to (52) is given in the following theorem

**Theorem 3.1.** Let

$$a_{k1} = -\frac{g_{11}^2}{\gamma_1} - \frac{g_{21}^2}{\gamma_1} - g, \quad a_{k2} = -\frac{g_{11}g_{12}}{\gamma_1} - \frac{g_{21}g_{22}}{\gamma_1}, \quad a_{k3} = \lambda_{k1}g_{11} + \lambda_{k2}g_{21} = \sqrt{g}p_{k1},$$

$$b_{k1} = -\frac{g_{11}g_{12}}{\gamma_1} - \frac{g_{21}g_{22}}{\gamma_1}, \quad b_{k2} = \frac{g_{22}^2}{\gamma_1} - g, \quad b_{k3} = \lambda_{k1}g_{12} + \lambda_{k2}g_{22} - \sqrt{g}p_{k2}. $$

The solution to (52) is given as

$$\lambda_{k1}^{n+2/3} = \lambda_{k1} - \frac{g_{11}a_{k1}b_{k3} - a_{k3}b_{k1}}{\gamma_1 a_{k1}b_{k2} - a_{k2}b_{k1}} - \frac{g_{12}a_{k1}b_{k3} - a_{k3}b_{k1}}{\gamma_1 a_{k2}b_{k1} - a_{k1}b_{k2}},$$

$$\lambda_{k2}^{n+2/3} = \lambda_{k2} - \frac{g_{21}a_{k2}b_{k3} - a_{k3}b_{k2}}{\gamma_1 a_{k1}b_{k2} - a_{k2}b_{k1}} - \frac{g_{22}a_{k1}b_{k3} - a_{k3}b_{k2}}{\gamma_1 a_{k2}b_{k1} - a_{k1}b_{k2}},$$

$$p_{k1}^{n+2/3} = p_{k1} + \sqrt{g} \frac{a_{k2}b_{k3} - a_{k3}b_{k2}}{a_{k1}b_{k2} - a_{k2}b_{k1}}, \quad p_{k2}^{n+2/3} = p_{k2} + \sqrt{g} \frac{a_{k1}b_{k3} - a_{k3}b_{k1}}{a_{k2}b_{k1} - a_{k1}b_{k2}}.$$

for $k = 1, 2, 3$.

**Proof.** We derive the formulas using the method of Lagrange multipliers. The formula can be derived component-wisely. Let $k \in \{1, 2, 3\}$. The constraint (51) implies

$$\begin{cases}
F_1(\mu_k, q_k) = 0, \\
F_2(\mu_k, q_k) = 0,
\end{cases} \quad \text{with} \quad \begin{cases}
F_1(\mu_k, q_k) = g_{11}\mu_{k1} + g_{21}\mu_{k2} - \sqrt{g}q_{k1}, \\
F_2(\mu_k, q_k) = g_{12}\mu_{k1} + g_{22}\mu_{k2} - \sqrt{g}q_{k2}. 
\end{cases} \quad (53)$$

To derive the formula for $\mu_k, q_k$, consider the following Lagrangian functional

$$E_{k2} = \int_{\Omega} \left( |q_k - p_k|^2 + \gamma_1 |\mu_k - \lambda_k|^2 \right) dx + s_{k1}F_1(\mu_k, q_k) + s_{k2}F_2(\mu_k, q_k), \quad (54)$$

where $s_{k1}, s_{k2}$ are Lagrange multipliers. Computing partial derivatives of $E_{k2}$ with respect to $\mu_k, q_k$ and setting them to 0 gives rise to

$$\begin{cases}
\gamma_1 (\mu_{k1} - \lambda_{k1}) + s_{k1}g_{11} + s_{k2}g_{12} = 0, \\
\gamma_1 (\mu_{k2} - \lambda_{k2}) + s_{k1}g_{21} + s_{k2}g_{22} = 0, \\
q_{k1} - p_{k1} - s_{k1}\sqrt{g} = 0, \\
q_{k2} - p_{k2} - s_{k2}\sqrt{g} = 0.
\end{cases} \quad (55)$$

Reorganizing (55), we have

$$\begin{cases}
\mu_{k1} = \lambda_{k1} - \frac{g_{11}}{\gamma_1} s_{k1} - \frac{g_{12}}{\gamma_1} s_{k2}, \\
\mu_{k2} = \lambda_{k2} - \frac{g_{21}}{\gamma_1} s_{k1} - \frac{g_{22}}{\gamma_1} s_{k2}, \\
q_{k1} = p_{k1} + s_{k1}\sqrt{g}, \\
q_{k2} = p_{k2} + s_{k2}\sqrt{g}.
\end{cases} \quad (56)$$
Substituting (56) into (53) gives rise to
\[
\begin{align*}
  a_{k1}s_{k1} + a_{k2}s_{k2} + a_{k3} &= 0, \\
  b_{k1}s_{k1} + b_{k2}s_{k2} + b_{k3} &= 0
\end{align*}
\] (57)

for \(a_{k1}, a_{k2}, a_{k3}, b_{k1}, b_{k2}, b_{k3}\) defined in Theorem 3.1. Solving (57) for \(s_{k1}, s_{k2}\) gives
\[
s_{k1} = \frac{a_{k2}b_{k3} - b_{k2}a_{k3}}{b_{k2}a_{k1} - a_{k2}b_{k1}}, \quad s_{k2} = \frac{a_{k1}b_{k3} - a_{k3}b_{k1}}{b_{k1}a_{k2} - b_{k2}a_{k1}}.
\] (58)

Substituting (58) into (56) finishes the proof.

**Remark 3.2.** We remark that the approximation (50) is not strictly the first step of the first order SQP [1] to solve (46), as we did not use the Jacobian of the constraint in \(S\) when constructing \(S_G\). Nevertheless, numerical experiments suggest that our proposed algorithm converges with this numerical approximation.

**Remark 3.3.** Using \(G^{n+1/3}\) and \(g^{n+1/3}\) defined above, another option to update \(\lambda^{n+1/3}\) in (31) is to replace \(m(p^{n+1/3})\) in (42) and (43) by \(g^{n+1/3}\) to get
\[
\begin{align*}
  \lambda^{n+1/3} &= (\lambda_k^{n+1/3})^3_{k=1} \in (H^1(\Omega))^{3 \times 2}, \\
  \gamma_1 \int_{\Omega} \lambda_k^{n+1/3} \cdot \mu_k dx + 2\beta \tau \int_{\Omega} \sqrt{g^{n+1/3}} (\nabla \cdot \lambda_k^{n+1/3}) (\nabla \cdot \mu_i) dx = \gamma_1 \int_{\Omega} \lambda_k^{n+1/3} \cdot \mu_k dx, \\
  \forall \mu_k \in (H^1(\Omega))^2, & \quad k = 1, 2, 3.
\end{align*}
\] (59)

and
\[
\begin{align*}
  \gamma_1 \lambda_k^{n+1/3} - 2\beta \tau \nabla \left( \sqrt{g^{n+1/3}} (\nabla \cdot \lambda_k^{n+1/3}) \right) = \gamma_1 \lambda_k^n & \quad \text{in } \Omega, \\
  \lambda_k^{n+1/3} \cdot n = 0 & \quad \text{on } \partial \Omega, \quad k = 1, 2, 3.
\end{align*}
\] (60)

Here \(g^{n+1/3}\) is a relaxed version of \(m(p^{n+1/3})\). Such a treatment enhances the coupling between \((p, \lambda)\) and \(G\). In our experiments, (59)-(60) give similar results as (42)-(43). While our algorithm is more stable with (59)-(60). In the rest of the paper, we stick with (59)-(60).

**Remark 3.4.** Similar to Remark 3.3, another option to update \(\lambda^{n+1/3}\) in (34) is to replace \(m(p^{n+1/3})\) in (45) by \(g^{n+1/3}\) to get
\[
\begin{align*}
  \gamma_1 \lambda_k^{n+1/3} - 2\beta \tau \nabla \left( \sqrt{g^{n+1/3}} - \alpha^2 (\nabla \cdot \lambda_k^{n+1/3}) \right) = \gamma_1 \lambda_k^n & \quad \text{in } \Omega, \\
  \lambda_k^{n+1/3} \cdot n = 0 & \quad \text{on } \partial \Omega, \quad k = 1, 2, 3.
\end{align*}
\] (61)

In the rest of the paper, we stick with (61).

### 3.6 On the solution of (35)

The solution \((p^{n+2/3}, \lambda^{n+2/3})\) is the minimizer of
\[
(p^{n+2/3}, \lambda^{n+2/3}) = \arg \min_{(q, \mu) \in S} \int_{\Omega} \left( |q - p^{n+1/3}|^2 + \gamma_1 |\mu - \lambda^{n+1/3}|^2 \right) dx.
\] (62)
Similar to what has been done in Section 3.5, we approximate (62) by
\[
(p^{n+2/3}, \lambda^{n+2/3}) = \arg \min_{(q, \nu) \in S_{G^{n+1/3}}} \int_{\Omega} \left( |q - p^{n+1/3}|^2 + \gamma_1 |\mu - \lambda^{n+1/3}|^2 \right) dx
\] (63)

with
\[
\tilde{S}_G = \left\{ (q, \nu) \in ((L^2(\Omega))^{3 \times 2}, (L^2(\Omega))^{3 \times 2}) \mid \sqrt{\det G - \alpha^2 \nu_k} = q_k \cof(G), k = 1, 2, 3 \right\}.
\] (64)

Denote \(g^{n+1/3} = \det G^{n+1/3}\). In \(\tilde{S}_{G^{n+1/3}}\), we have \(\sqrt{g^{n+1/3} - \alpha^2 \nu} = q \cof(G^{n+1/3})\). We consider the following constrained optimization problem
\[
(p^{n+2/3}, \lambda^{n+2/3}) = \arg \min_{(q, \nu) : \sqrt{g^{n+1/3} - \alpha^2 \nu} = q \cof(G^{n+1/3})} \int_{\Omega} \left( |q - p^{n+1/3}|^2 + \gamma_1 |\nu - \lambda^{n+1/3}|^2 \right) dx.
\] (65)

For simplicity, we temporarily use \(g, g_{ij}, p_{kr}, \lambda_{kr}\) to denote \(g^{n+1/3}, g_{ij}^{n+1/3}, p_{kr}^{n+1/3}, \lambda_{kr}^{n+1/3}\) in this subsection. The explicit solution for (65) is given in the following theorem

**Theorem 3.2.** Let
\[
a_{k1} = -g_{22}^2 - g_{12}^2 - \frac{g - \alpha^2}{\gamma_1}, \quad a_{k2} = g_{12}g_{22} + g_{11}g_{12}, \quad a_{k3} = g_{22}p_{k1} - g_{12}p_{k2} - \sqrt{g - \alpha^2 \lambda_{k1}},
\]
\[
b_{k1} = -g_{11}g_{12} + g_{12}g_{22}, \quad b_{k2} = g_{11}^2 - g_{12}^2 - \frac{g - \alpha^2}{\gamma_1}, \quad b_{k3} = g_{11}p_{k1} - g_{12}p_{k2} - \sqrt{g - \alpha^2 \lambda_{k2}}.
\]

The solution to (65) is given as
\[
\lambda_{k1}^{n+2/3} = \lambda_{k1} + \frac{\sqrt{g - \alpha^2 a_{k2}b_{k3} - b_{k2}a_{k3}}}{b_{k2}a_{k1} - a_{k2}b_{k1}},
\]
\[
\lambda_{k2}^{n+2/3} = \lambda_{k2} + \frac{\sqrt{g - \alpha^2 a_{k1}b_{k3} - b_{k1}a_{k3}}}{b_{k1}a_{k2} - a_{k1}b_{k2}},
\]
\[
p_{k1}^{n+2/3} = p_{k1} - g_{22} \frac{a_{k2}b_{k3} - a_{k3}b_{k2}}{b_{k2}a_{k1} - a_{k2}b_{k1}} + g_{22} \frac{a_{k1}b_{k3} - a_{k3}b_{k1}}{a_{k2}b_{k1} - a_{k1}b_{k2}},
\]
\[
p_{k2}^{n+2/3} = p_{k2} + g_{12} \frac{a_{k2}b_{k3} - a_{k3}b_{k2}}{a_{k1}b_{k2} - a_{k2}b_{k1}} - g_{11} \frac{a_{k1}b_{k3} - a_{k3}b_{k1}}{a_{k2}b_{k1} - a_{k1}b_{k2}},
\]
for \(i = 1, 2, 3\).

Theorem 3.2 can be proved similarly to Theorem 3.1. The proof is omitted here.

### 3.7 On the solution of (33) and (36)

Problems (33) and (36) are the same, in which \(p^{n+1}\) solves
\[
\arg \min_{q \in (H^1(\Omega))^{3 \times 2}} \left[ \frac{1}{2\gamma} \int_{\Omega} |q - p^{n+2/3}|^2 dx + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |(v_q)_k - f_k|^2 dx + I_{\Sigma}(q) \right].
\] (66)
Since \( p^{n+1} \in \Sigma_f \), there exists some \( u^{n+1} \in (H^1(\Omega))^3 \) such that \( p^{n+1} = \nabla u^{n+1} \). Furthermore, \( u^{n+1} \) solves

\[
\arg \min_{v \in (H^1(\Omega))^3} \left[ \frac{1}{2\tau} \int_{\Omega} |\nabla v - p^{n+2/3}|^2 dx + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |v_k - f_k|^2 dx \right].
\]

(67)

Here \( u^{n+1} \) is also the unique weak solution of the following linear elliptic problem

\[
\begin{cases}
-\eta \nabla^2 u^{n+1} + \frac{\tau}{\eta} u^{n+1} = -\eta \nabla \cdot p^{n+2/3} + \tau f_k \text{ in } \Omega, \\
\nabla u^{n+1} \cdot n = 0 \text{ on } \partial \Omega, \\
k = 1, 2, 3.
\end{cases}
\]

(68)

After \( u^{n+1} \) is solved, we set \( p^{n+1} = \nabla u^{n+1} \).

### 3.8 Initial condition

Both schemes (31)–(33) and (34)–(36) require an initial condition \((p^0, \lambda^0)\). We first initialize

\( u^0 = f \) or \( u^0 = 0 \)

and compute \( p^0 = \nabla u^0 \). Then we set \( \lambda^0 = \sqrt{\det p^0 p^0 (M(p^0))^{-1}} \) for problem (17) or \( \lambda^0 = \frac{1}{\sqrt{\det p^0 - \alpha^2}} p^0 \text{cof}(M(p^0)) \) for problem (20).

We also take \( G \) as a variable which is updated during iterations, we initialize \( G^0 = M(p^0) \).

Our algorithms for problems (17) and problem (20) are summarized in Algorithm 1 and 2, respectively.

**Remark 3.5.** The functionals in (8) and (10) are complicated and nonconvex. All we can expect is that our algorithms converge to a local minimizer. However, in our experiments, both algorithms are robust to initial conditions and noise. With either initial condition discussed above, every time we generate noisy images with random Gaussian noise, our algorithms always provide good results.

### 3.9 On the periodic boundary condition

The solutions to each subproblem discussed so far use Neumann boundary conditions. In image processing, the periodic boundary condition is a popular condition which allows one to use FFT. In this subsection, we discuss the minimal efforts necessary to modify the algorithms and solvers above to accommodate periodic boundary conditions.

Assume our computational domain is \( \Omega = [0, L_1] \times [0, L_2] \). We first replace the function space \( H^1(\Omega) \) by \( H^1_P(\Omega) \) defined as

\[
H^1_P(\Omega) = \left\{ v \in H^1(\Omega) : v(0,:) = v(L_1,:), \ v(:,0) = v(:,L_2) \right\}.
\]

The set \( \Sigma_f \) is replaced by

\[
\Sigma_f = \left\{ q \in (L^2(\Omega))^{3 \times 2} \exists v \in H^1_P(\Omega) \text{ such that } q = \nabla v \text{ and } \int_{\Omega} v_k - f_k dx = 0 \text{ for } k = 1, 2, 3 \right\}.
\]
Problems (17), (18) and (20) are replaced by

\begin{align*}
\min_{\mathbf{q} \in (\mathcal{H}^1(\Omega))^3 \times 2, \mathbf{\mu} \in (\mathcal{H}^1(\Omega))^3 \times 2} & \int_{\Omega} \left( 1 + \sum_{k=1}^{3} |\nabla \cdot \mathbf{\mu}_k|^2 \right) \sqrt{m(\mathbf{q})} d\mathbf{x} \\
& + \frac{1}{2\eta} \sum_{k=1}^{3} \int_{\Omega} |(v_\mathbf{q})_k - f_k|^2 d\mathbf{x} + I_{\Sigma_f}(\mathbf{q}) + I_S(\mathbf{q}, \mathbf{\mu}),
\end{align*}

(69)

\begin{align*}
\nabla^2 (v_\mathbf{q})_k &= \nabla \cdot \mathbf{q}_k, \\
(v_\mathbf{q})_k(x_1, 0) &= (v_\mathbf{q})_{\mu}(x_1, L_2), \quad 0 \leq x_1 \leq L_1, \\
(v_\mathbf{q})_k(0, x_2) &= (v_\mathbf{q})_{\mu}(L_1, x_2), \quad 0 \leq x_2 \leq L_2, \\
\frac{\partial (v_\mathbf{q})_k}{\partial x_2} - q_k^2 (x_1, 0) &= \frac{\partial (v_\mathbf{q})_k}{\partial x_2} - q_k^2 (x_1, L_2), \quad 0 \leq x_1 \leq L_1, \\
\frac{\partial (v_\mathbf{q})_k}{\partial x_1} - q_k^1 (0, x_2) &= \frac{\partial (v_\mathbf{q})_k}{\partial x_1} - q_k^1 (L_1, x_2), \quad 0 \leq x_2 \leq L_2,
\end{align*}

(70)

Integration of (69) gives

\begin{align*}
\int_{\Omega} (v_\mathbf{q})_k d\mathbf{x} &= \int_{\Omega} f_k d\mathbf{x},
\end{align*}

17
and
\[
\min_{q \in (H^1_0(\Omega))^3, \nu \in (H^1_0(\Omega))^3} \int_\Omega \left( 1 + \sum_{k=1}^3 |\nabla \cdot \nu_k|^2 \right) \sqrt{m(q) - \alpha^2} dx + \frac{1}{2\eta} \sum_{k=1}^3 \int_\Omega |(v_q)_k - f_k|^2 dx + I_{\Sigma_f(q)} + I_{\bar{\Sigma}}(q, \nu),
\]
respectively. We modify subproblem solvers as follows. Replace (59), (60) and (61) by
\[
\begin{align*}
\gamma_1 \int \lambda_k^{n+1/3} \cdot \nu_k dx + 2\beta \tau \int \sqrt{g^{n+1/3} \nabla \cdot \lambda_k^{n+1/3}} \nabla \cdot \nu_k dx &= \gamma_1 \int \lambda_k^n \cdot \nu_k dx, \\
\forall \nu_k \in (H^1_P(\Omega))^3, k = 1, 2, 3,
\end{align*}
\]
(72)
and
\[
\begin{align*}
\gamma_1 \lambda_k^{n+1/3} - 2\beta \tau \nabla \left( \sqrt{g^{n+1/3} \nabla \cdot \lambda_k^{n+1/3}} \right) &= \gamma_1 \lambda_k^n \quad \text{in } \Omega, \\
\lambda_k^{n+1/3}(x_1, 0) &= \lambda_k^{n+1/3}(x_1, L_2), \quad 0 \leq x_1 \leq L_1, \\
\lambda_k^{n+1/3}(0, x_2) &= \lambda_k^{n+1/3}(L_1, x_2), \quad 0 \leq x_2 \leq L_2, \\
k &= 1, 2, 3,
\end{align*}
\]
(73)
and
\[
\begin{align*}
\gamma_1 \lambda_k^{n+1/3} - 2\beta \tau \nabla \left( \sqrt{g^{n+1/3} \nabla \cdot \lambda_k^{n+1/3}} - \alpha^2 \nabla \cdot \lambda_k^{n+1/3} \right) &= \gamma_1 \lambda_k^n \quad \text{in } \Omega, \\
\lambda_k^{n+1/3}(x_1, 0) &= \lambda_k^{n+1/3}(x_1, L_2), \quad 0 \leq x_1 \leq L_1, \\
\lambda_k^{n+1/3}(0, x_2) &= \lambda_k^{n+1/3}(L_1, x_2), \quad 0 \leq x_2 \leq L_2, \\
k &= 1, 2, 3.
\end{align*}
\]
(74)
Equations (43) and (45) can be modified analogously.

Finally, we replace (67) and (68) by
\[
\arg \min_{\nu \in (H^1_0(\Omega))^3} \left[ \frac{1}{2\tau} \int_\Omega |\nabla \nu - \nu_k^{n+2/3}|^2 dx + \frac{1}{2\eta} \sum_{k=1}^3 \int_\Omega |v_k - f_k|^2 dx \right],
\]
and
\[
\begin{align*}
\frac{\partial u_k^{n+1}}{\partial x_1}(x_1, 0) &= \frac{\partial u_k^{n+1}}{\partial x_1}(x_1, L_2), \quad 0 \leq x_1 \leq L_1, \\
\frac{\partial u_k^{n+1}}{\partial x_2}(0, x_2) &= \frac{\partial u_k^{n+1}}{\partial x_2}(L_1, x_2), \quad 0 \leq x_2 \leq L_2, \\
k &= 1, 2, 3,
\end{align*}
\]
(76)
respectively.

Problems (70), (74) and (76) are linear elliptic problems which can be solved efficiently by FFT. In the rest of this article, periodic boundary conditions are used.
4 Numerical discretization

In this section, we numerically discretize the scheme (31)-(33) and (34)-(36) with periodic boundary conditions. Let our computational domain be \( \Omega = [0, L_1] \times [0, L_2] \). We discretize \( \Omega \) by \( M \times N \) grids with step \( h = L_1/M = L_2/N \). For any function \( v \) defined on \( \Omega \), we denote \( v(ih,jh) \) by \( v(i,j) \) for \( 1 \leq i \leq M, 1 \leq j \leq N \). We assume all functions satisfy the periodic boundary condition.

We first define several difference operators. For \( 1 \leq i \leq M, 1 \leq j \leq N \) and a scalar-valued function \( v \), define the forward (+) and backward (−) difference by

\[
\begin{align*}
\partial_1^+ v(i,j) &= (v(i+1,j) - v(i,j))/h, \\
\partial_1^- v(i,j) &= (v(i,j) - v(i-1,j))/h, \\
\partial_2^+ v(i,j) &= (v(i,j+1) - v(i,j))/h, \\
\partial_2^- v(i,j) &= (v(i,j) - v(i,j-1))/h,
\end{align*}
\]

where \( v(M+1,j) = v(1,j), v(0,j) = v(M,j), v(i,N+1) = v(i,1), v(i,0) = v(i,N) \) are used. With the notations above, the forward (+) and backward (−) gradient and divergence for scalar-valued function \( v \) and vector-valued function \( \mathbf{q} \) are defined as

\[
\begin{align*}
\nabla^+ v(i,j) &= (\partial_1^+ v(i,j), \partial_2^+ v(i,j)), \\
\nabla^- v(i,j) &= (\partial_1^- v(i,j), \partial_2^- v(i,j)), \\
\text{div}^+ \mathbf{q}(i,j) &= \partial_1^+ q_1(i,j) + \partial_2^+ q_2(i,j), \\
\text{div}^- \mathbf{q}(i,j) &= \partial_1^- q_1(i,j) + \partial_2^- q_2(i,j).
\end{align*}
\]

Define the shifting operator and identity operator by

\[
S_1^+ v(i,j) = v(i \pm 1,j), \quad S_2^+ v(i,j) = v(i,j \pm 1), \quad \mathcal{I} v(i,j) = v(i,j).
\]

Let \( \mathcal{F} \) and \( \mathcal{F}^{-1} \) denote the discrete Fourier transform and the inverse transform. We have

\[
\begin{align*}
\mathcal{F}(S_1^+ v)(i,j) &= (\cos z_i \pm \sqrt{-1} \sin z_i) \mathcal{F}(v)(i,j), \\
\mathcal{F}(S_2^+ v)(i,j) &= (\cos z_j \pm \sqrt{-1} \sin z_j) \mathcal{F}(v)(i,j),
\end{align*}
\]

where

\[
z_i = \frac{2\pi}{M}(i-1), \quad z_j = \frac{2\pi}{N}(j-1)
\]

for \( i = 1, \ldots, M \) and \( j = 1, \ldots, N \).

4.1 Computing the discrete analogue of \( \mathbf{p}^{n+1/3}, \mathbf{\lambda}^{n+1/3} \) in (31) and (34)

In (31), \( \mathbf{p}^{n+1/3} \) can be computed pixel-wisely using (41). For \( \mathbf{\lambda}^{n+1/3} \), we solve (73) whose discrete analogue is

\[
\gamma_1 \mathbf{\lambda}_k^{n+1/3} - 2\beta \tau \nabla^+ \left( \sqrt{g^{n+1/3}} (\text{div}^- \mathbf{\lambda}_k^{n+1/3}) \right) = \gamma_1 \mathbf{\lambda}_k^n
\]

for \( k = 1, 2, 3 \). Instead of solving (78), we use the frozen coefficient approach [48, 23] to solve

\[
\gamma_1 \mathbf{\lambda}_k^{n+1/3} - c_1 \nabla^+ (\text{div}^- \mathbf{\lambda}_k^{n+1/3}) = \gamma_1 \mathbf{\lambda}_k^n + \nabla^+ \left( 2\beta \tau \left( \sqrt{g^{n+1/3}} - c_1 \right) (\text{div}^- \mathbf{\lambda}_k^n) \right),
\]

where

\[
\mathbf{\lambda}_k^{n+1/3} = \mathbf{\lambda}_k^n + \frac{2\tau}{\gamma_1} \left( \sqrt{g^{n+1/3}} - c_1 \right) \nabla^- \mathbf{\lambda}_k^n,
\]
where \( c_1 > 0 \) is some constant. Problem (79) can be solved efficiently by FFT. We first rewrite (79) in matrix form as

\[
\begin{pmatrix}
\gamma_1 - c_1 \partial_1^+ \partial_1^- \\
- c_1 \partial_2^+ \partial_2^-
\end{pmatrix}
\begin{pmatrix}
\lambda_{k1}^{n+1/3} \\
\lambda_{k2}^{n+1/3}
\end{pmatrix}
= \begin{pmatrix} w_1 \\ w_2 \end{pmatrix}
\]

(80)

with

\[
w_1 = \gamma_1 \lambda_{k1}^n + \partial_1^+ \left( 2\beta \tau \left( \sqrt{g^{n+1/3}} - c_1 \right) \left( \partial_1^- \lambda_{k1}^n + \partial_2^- \lambda_{k2}^n \right) \right),
\]

\[
w_2 = \gamma_1 \lambda_{k2}^n + \partial_2^+ \left( 2\beta \tau \left( \sqrt{g^{n+1/3}} - c_1 \right) \left( \partial_1^- \lambda_{k1}^n + \partial_2^- \lambda_{k2}^n \right) \right).
\]

Problem (80) is equivalent to

\[
\begin{pmatrix}
\gamma_1 - c_1 (S_1^+ - I)(I - S_1^-)/h^2 \\
- c_1 (S_2^+ - I)(I - S_2^-)/h^2
\end{pmatrix}
\begin{pmatrix}
\lambda_{k1}^{n+1/3} \\
\lambda_{k2}^{n+1/3}
\end{pmatrix}
= \begin{pmatrix} \mathcal{F}(\lambda_{k1}^{n+1/3}) \\ \mathcal{F}(\lambda_{k2}^{n+1/3}) \end{pmatrix}.
\]

(81)

Taking discrete Fourier transform on both sides of (81) gives rise to

\[
\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}
\begin{pmatrix} \mathcal{F}(\lambda_{k1}^{n+1/3}) \\ \mathcal{F}(\lambda_{k2}^{n+1/3}) \end{pmatrix}
= \begin{pmatrix} \mathcal{F}(w_1) \\ \mathcal{F}(w_2) \end{pmatrix}
\]

(82)

with

\[
a_{11} = \gamma_1 - 2c_1 (\cos z_i - 1)/h^2, \quad a_{21} = \gamma_1 - 2c_1 (\cos z_j - 1)/h^2,
\]

\[
a_{12} = -c_1 (1 - \cos z_i - \sqrt{-1} \sin z_i)(1 - \cos z_j + \sqrt{-1} \sin z_j)/h^2,
\]

\[
a_{22} = -c_1 (1 - \cos z_j - \sqrt{-1} \sin z_j)(1 - \cos z_i + \sqrt{-1} \sin z_i)/h^2,
\]

and \( z_i, z_j \) defined in (77). We have

\[
\begin{pmatrix}
\lambda_{k1}^{n+1/3} \\
\lambda_{k2}^{n+1/3}
\end{pmatrix}
= \text{Real} \left( \mathcal{F}^{-1} \left[ \frac{1}{a_{11}a_{22} - a_{12}a_{21}} \left( a_{22} \mathcal{F}(w_1) - a_{12} \mathcal{F}(w_2) \right) \right] \right).
\]

(83)

In (34), \( p^{n+1/3} \) can be computed pixel-wisely using (44). The computation of the discrete analogue of \( \lambda^{n+1/3} \) can be conducted similarly to that of \( \lambda^{n+1/3} \) in (31) as is discussed above. One can use (83) except replacing \( w_1, w_2 \) by

\[
w_1 = \gamma_1 \lambda_{k1}^n + \partial_1^+ \left( 2\beta \tau \left( \sqrt{g^{n+1/3}} - \alpha^2 - c_1 \right) \left( \partial_1^- \lambda_{k1}^n + \partial_2^- \lambda_{k2}^n \right) \right),
\]

\[
w_2 = \gamma_1 \lambda_{k2}^n + \partial_2^+ \left( 2\beta \tau \left( \sqrt{g^{n+1/3}} - \alpha^2 - c_1 \right) \left( \partial_1^- \lambda_{k1}^n + \partial_2^- \lambda_{k2}^n \right) \right).
\]

### 4.2 Computing the discrete analogue of \( p^{n+2/3}, \lambda^{n+2/3} \) in (32) and (35)

The computation of the discrete analogues of \( p^{n+2/3}, \lambda^{n+2/3} \) in (32) and (35) can be conducted pixel-wisely using Theorem 3.1 and 3.2, respectively.
Figure 2: Empirical justification of proposed models on three images sampled from [55]. From left to right, ‘Pens’, ‘Plant’ and ‘Toy’.

4.3 Computing the discrete analogue of $p_{n+1}^n$ in (33) and (36)

For $p_{n+1}^n$ in (33) and (36), we solve (76) whose discrete analogue is

$$\begin{cases} -\eta \text{div}(\nabla^+ u_k^{n+1}) + \tau u_k^{n+1} = -\eta \text{div} p_k^{n+2/3} + \tau f_k \text{ in } \Omega, \\ k = 1, 2, 3. \end{cases} \quad (84)$$

Problem (84) is equivalent to

$$\begin{cases} (\tau h^2 I - \eta (I - S^-_1)(S^+_1 - I) - \eta (I - S^-_2)(S^+_2 - I)) u_k^{n+1} = g_k, \\ k = 1, 2, 3, \end{cases} \quad (85)$$

where $g_k = -\eta h^2 \text{div} p_k^{n+2/3} + \tau h^2 f_k$. Taking the discrete Fourier transform on both sides, we get

$$b\mathcal{F}(u_k^{n+1}) = \mathcal{F}(g_k)$$

with $b = \tau h^2 + 4\eta - 2\eta \cos z_i - 2\eta \cos z_j$, where $z_i, z_j$ are defined in (77). We compute

$$u_k^{n+1} = \text{Real} \left( \mathcal{F}^{-1} \left( \frac{\mathcal{F}(g)}{b} \right) \right) \quad (86)$$

and $p_{n+1} = \nabla^+ u$.

5 Empirical justifications of the new models

In this section, we empirically justify the new terms we introduced to the modified color elastica models. In model (8) and (10), three new terms are considered. A new surface area measure $A_1$ and two new color elastica terms $E_1, E_2$, which are defined by

$$\begin{align*}
A_1(f) &= \int_{\Omega} \sqrt{g - \alpha^2} d\mathbf{x}, \\
E_1(f) &= \int_{\Omega} \left( \sum_{k=1}^{3} g |\Delta_g f_k|^2 \right) \sqrt{g} d\mathbf{x}, \\
E_2(f) &= \int_{\Omega} \left( \sum_{k=1}^{3} (g - \alpha^2) |\tilde{\Delta}_g f_k|^2 \right) \sqrt{g - \alpha^2} d\mathbf{x}.
\end{align*}$$
Figure 3: Comparison of the relative energy of the surface area (top) and the elastica (bottom) as functions of the noise standard deviation (SD). Noisy images are generated by adding Gaussian noise with SD varying from 0 to 0.4. (a)–(c) correspond to the images ‘Pens’, ‘Plant’ and ‘Toy’ in Figure 2, respectively. Each relative energy is averaged over 10 experiments.

Figure 4: Comparison of the relative energy of different regularizers. The noisy images are generated by adding Gaussian noise with SD varying from 0 to 0.4. (a)–(c) correspond to ‘Pens’, ‘Plant’ and ‘Toy’ images in Figure 2. The tested regularizers are $F_0$ (in CE), $F_1$ (in model (8)), $F_2$ (in model (10)), $F_{PA}$ (in PA), $F_{CTV}$ (in CTV) and $F_{VTV}$ (in VTV). Each relative energy is averaged over 10 experiments.

The term $E_1$ is used in model (8). The terms $A_1$ and $E_2$ are used in model (10). We denote the original surface area and elastica term in the color elastica model (6) by

$$A_0(f) = \int_{\Omega} \sqrt{g} dx, \quad E_0(f) = \int_{\Omega} \left( \sum_{k=1}^{3} |\Delta_g f_k|^2 \right) \sqrt{g} dx.$$  \hspace{1cm} (87)

Given a clean image $f_0$ and its noisy version $f$, we measure the effectiveness of these terms by computing the ratio between their values at $f$ and $f_0$. The quantity is considered to be more sensitive if the ratio increases more rapidly as the noise level increases. Specifically,
we define the relative energy for surface areas and elastica terms as

\[
\mathcal{R}\mathcal{A}_i(f) = \frac{\mathcal{A}_i(f)}{\mathcal{A}_i(f_0)},\\
\mathcal{R}\mathcal{E}_j(f) = \frac{\mathcal{E}_j(f)}{\mathcal{E}_j(f_0)}
\]

for \(i = 0, 1\) and \(j = 0, 1, 2\). For each term, we monitor the relation between its relative energy as a function of the noise level injected to the image.

In this set of experiments, we set \(\alpha = 10^{-3}\) and consider three images in Figure 2: ‘Pens’, ‘Plant’ and ‘Toy’. The noisy images are generated by adding Gaussian noise with standard deviation (variance) varying from 0 to 0.4. The comparison of the two relative energies of all measures are shown in Figure 3. Each relative energy is averaged over 10 experiments. The first row shows the comparison for variations on surface areas. The newly introduced term \(\mathcal{A}_1\) appears to be more sensitive than \(\mathcal{A}_0\) as its relative energy increases faster as a function of noise level for all three images. The second row shows comparisons for the elastica terms. The newly introduced measures \(\mathcal{E}_1\) and \(\mathcal{E}_2\) are similar. Both are more sensitive than \(\mathcal{E}_0\).

We then compare the effectiveness of the regularizers in the proposed models and other models for color image regularization, including the color elastica (CE) model [30], the Polyakov action (PA) model [27], the color total variation (CTV) model [5] and the vectorial total variation (VTV) model [21]. We denote the regularizers in CE, Model (8), Model (10), PA, CTV and VTV by

\[
\mathcal{F}_0(f) = \mathcal{A}_0(f) + \beta\mathcal{E}_0(f),\\
\mathcal{F}_1(f) = \mathcal{A}_0(f) + \beta\mathcal{E}_1(f),\\
\mathcal{F}_2(f) = \mathcal{A}_1(f) + \beta\mathcal{E}_2(f),\\
\mathcal{F}_{PA}(f) = \mathcal{A}_0(f),\\
\mathcal{F}_{CTV}(f) = \int_\Omega \sqrt{\sum_{k=1}^{3} |\nabla f_k|^2} dx,\\
\mathcal{F}_{VTV}(f) = \int_\Omega \sigma_1(\nabla f) dx,
\]

respectively, where in \(\mathcal{F}_{VTV}(f)\), \(\sigma_1(\nabla f)\) denotes the largest singular value of the Jacobian.

Figure 5: Comparison of the relative energy of variants of model (8) and (10). The noisy images are generated by adding Gaussian noise with SD varying from 0 to 0.4. (a)–(c) correspond to images ‘Pens’, ‘Plant’ and ‘Toy’ in Figure 2. Each relative energy is averaged over 10 experiments.
matrix \( \nabla f \). For any energy \( F \) defined above, we define its relative energy as

\[
RF(f) = \frac{F(f)}{F(f_0)},
\]

where \( f_0 \) is the clean image and \( f \) is its noisy version.

Figure 4 displays the relative energies as a function of the noise levels for all regularizers. In this set of experiments, the three images presented in Figure 2, ‘Pens’, ‘Plant’ and ‘Toy’, are used. The noisy images are generated by adding Gaussian noise with SD varying from 0 to 0.4. For \( F_0, F_1, F_2 \), we set \( \alpha = 10^{-3} \). We use \( \beta = 10^{-2} \) in \( F_0 \) and \( \beta = 30 \) in \( F_1 \) and \( F_2 \). Each relative energy is averaged over 10 experiments. In Figure 4, the relative energy of CTV and VTV are close to each other. That is because both regularizers only depend on the Jacobian matrix \( \nabla f \). In this comparison, the relative energy of model (8) and (10), that is, \( F_1 \) and \( F_2 \), are the most effective as they increase fastest with the noise level. This comparison empirically justifies the fact that the new terms in model (8) and (10) are better suited for modeling and denoising natural images.

In model (8) and (10), the elastica term is weighted by \( g \) or \( g - \alpha^2 \). In the next test, we compare the relative energy of variants of model (8) and (10) in which the elastica term is weighted by different powers of \( g \) or \( g - \alpha^2 \). Specifically, denote

\[
F_{1,m}(f) = A_0(f) + \beta \int_\Omega \left( \sum_{k=1}^{3} g^m |\Delta_g f_k|^2 \right) \sqrt{g} \, dx,
\]

\[
F_{2,m}(f) = A_1(f) + \beta \int_\Omega \left( \sum_{k=1}^{3} (g - \alpha^2)^m |\Delta_g f_k|^2 \right) \sqrt{g - \alpha^2} \, dx.
\]

We compared in Figure 5 the relative energies \( RF_{1,m} \) and \( RF_{2,m} \) for \( m = 1, 2, 3, 4 \). In this experiment, we set \( \alpha = 10^{-3} \) and \( \beta = 30 \). We observe that model (8) and (10), corresponding to \( m = 1 \), are most effective for all three test images when SD \leq 0.3. While model (8) and (10) are not the best when SD > 0.3, they are still most effective for practical consideration, since usually very large noise is rare in practice.

6 Numerical experiments

We demonstrate the effectiveness of the propose models and the efficiency of the proposed algorithms in this section. All experiments are implemented by MATLAB (R2020b) on a Windows desktop with 16GB RAM and Intel(R) Core(TM) i7-10700 CPU: 2.90GHz. In our experiments, images with pixel values in [0, 1] are used. We set \( h = 1, \gamma_1 = 1, \gamma_2 = 3, \xi_1 = 10^{-5} \) and \( \epsilon = 10^{-3} \), where \( \xi_1 \) is the stopping criterion of the iterative method for \( p^{n+1/3} \) and \( \epsilon \) is the small constant in (44). For both algorithms, we use the stopping criterion

\[
\frac{\|u^{n+1} - u^n\|}{\|u^n\|} \leq \zeta,
\]

for some small \( \zeta > 0 \), where \( \| \cdot \| \) denotes the Frobenius norm.

There are three model parameters: \( \alpha, \beta \) and \( \eta \). The parameter \( \beta \) controls the weight of the elastica term and \( \eta \) controls the weight of fidelity term. One can get a smoother result if larger \( \beta \) and \( \eta \) are used. The parameter \( \alpha \) controls the weight of the spatial coordinates in the manifold parametrization, see (2). As information from chromatic coordinates (feature coordinates) is usually more important than that from spatial coordinates, a
small $\alpha$ should be used. Without specification, we set $\alpha = 5 \times 10^{-4}$ for model (8), $\alpha = 3 \times 10^{-2}$ for model (10) and $\zeta = 10^{-5}$ for both algorithms.

Remark 6.1. There are three algorithm parameters: $\gamma_1$, $\gamma_2$ and $\tau$, which control the evolution speed of all auxiliary variables. The performance of our algorithms is not sensitive to these parameters. In our experiments, our algorithms converge as long as these parameters are small enough. Actually, we do not need to set these parameters too small. Setting $\gamma_1 = 1$, $\gamma_2 = 1$ and $\tau = 0.05$ already provides good results.

6.1 Performance on general examples

We apply the propose algorithm on three images to test the performance and efficiency. The clean images are presented in the first row of Figure 6. Then Gaussian noise with standard deviation (SD) 0.06 is added to generate the noisy images, as shown in the second row of Figure 6. The denoised images by model (8) and (10) are shown in the third and fourth row of Figure 6, respectively. Both models smooth the flat region of the images while keeping sharp edges. For the second column, the image 'Peppers', the clean image has some small oscillations. These oscillations are removed in the denoised images by both models. Meanwhile, the textures and shadows are kept.
Figure 7: Performance of the proposed models on general images. Surface plot of the zoomed regions of images in Figure 6. (a) Clean images. (b) Noisy images with Gaussian noise and SD=0.06. (c) Denoised images by model (8) with $\alpha = 5 \times 10^{-4}, \beta = 50$ and $\eta = 3$. (d) Denoised images by model (10) with $\alpha = 3 \times 10^{-2}, \beta = 30$ and $\eta = 0.2$. From Row 1 to Row 3: 'Portrait', 'Peppers', 'Fruits'.

Figure 8: Performance of the proposed models. Histories of energies of results in Figure 6. Row 1: Results by model (8). Row 2: Results by model (10). (a)-(c) correspond to 'Portrait', 'Peppers' and 'Fruits', respectively.

To better demonstrate the power of the proposed models, in Figure 7, we present the surface plot of the zoomed region of images in Figure 6. Column 1–4 correspond to the
Figure 9: Performance of the proposed models. Histories of relative errors of results in Figure 6. Row 1: Results by model (8). Row 2: Results by model (10). (a)-(c) correspond to 'Portrait', 'Peppers' and 'Fruits', respectively.

clean images, noisy images, denoised images by (8) and denoised images by model (10), respectively. The three rows correspond to the three test images. The red, green and blue surfaces are surface plot of the RGB channels of the images. The denoised images by both models provide smooth image surfaces while keeping the contrast.

We then demonstrate the efficiency of the proposed algorithms in Figure 8 and 9, which present the histories of energy and relative error of the results in Figure 6 during iterations, respectively. Here the energy refers to the value of the functionals in (8) and (10). In both figures, the first row shows results by Algorithm 1. The second row shows the results by Algorithm 2. Column (a)–(c) correspond to 'Portrait', 'Peppers' and 'Fruits', respectively. In all experiments, Algorithm 1 needs about 100 iterations for the energy to achieve its minimum, while Algorithm 2 is more efficient and only uses 50 iterations. In terms of the relative error, sublinear converges is observed for both algorithms.

6.2 Comparison with other models

We next demonstrate the advantages of the proposed models by comparing them with CE, PA, CTV and VTV. In this experiment with noisy images and SD=0.06, we set $\alpha = 5 \times 10^{-4}$, $\beta = 50$, $\eta = 3$ for model (8), $\alpha = 3 \times 10^{-2}$, $\beta = 30$, $\eta = 0.2$ for model (10). For other models, we use $\alpha = 1 \times 10^{-2}$, $\beta = 5 \times 10^{-3}$, $\eta = 1$ for CE, $\lambda = 6$ for CTV and $\lambda = 0.1$ for VTV. The results on the natural images, 'Cat', is shown in Figure 10, in which (a) and (b) show the clean and noisy images, (c)–(h) show denoised images by model (8), model (10), CE, PA, CTV and VTV, respectively. To better demonstrate the advantage of the propose models, for each image, the zoomed image of some selected region is presented under it. The zoomed images in (c)–(d) have a uniform theme: the pixel colors of the same object change smoothly. In (f)–(h), many pixels have abrupt and
artificial colors. In particular, the zoomed image in (h) looks like a color palette. In this comparison, the proposed models give more natural recoveries of images with less abrupt and artificial colors.

To quantify the improvements of the proposed models over others, we present the PSNR and SSIM [51] values of the denoised images of all experiments and models in Table 1. For each experiment, the largest value is marked in bold. The proposed models provide the largest PSNR and SSIM values in almost all experiments.

We present in Table 2 the number of iterations and CPU time (in seconds) used to satisfy the stopping criteria of each experiment in Table 1. Due to the model complexity, such as nonlinearities and high order derivatives, algorithms for the three color elastica related models (model (8), model (10) and CE) need more time than those for the other models. While among these three models, our algorithm for model (10) is the most efficient one. It only uses about half of the CPU time of that used by the other two
We then conduct a comprehensive comparison of the proposed models with CE, PA, CTV and VTV. We consider eight benchmark images [55] shown in Figure 11. The noisy images are generated by adding Gaussian noise with SD=0.06. The parameters of all models are the same as those mentioned at the beginning of the this subsection. The PSNR and SSIM values of all results are summarized in Table 3. On average, the results by model (8) have an increment of 0.71 in PSNR and 0.0018 in SSIM over the best existing models. Compared to that, Model (10) provides results with a larger PSNR but slightly smaller SSIM. The comparison of these models on images with larger noise, SD=0.2, is summarized in Table 4. In this experiment, we use $\alpha = 5 \times 10^{-4}, \beta = 50, \eta = 10$ for model (8), $\alpha = 5 \times 10^{-3}, \beta = 30, \eta = 3.5$ for model (10), $\alpha = 1 \times 10^{-2}, \beta = 5 \times 10^{-3}, \eta = 4$

|               | Noisy | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------------|-------|-----------|------------|-----|-----|-----|-----|
| Portrait      | 24.89 | 33.91     | **34.00**  | 33.10| 30.56| 31.74| 32.94|
| Peppers       | 24.43 | **31.48** | 31.27      | 30.91| 29.57| 30.45| 31.03|
| Fruits        | 24.44 | 32.63     | **32.67**  | 31.71| 29.58| 30.94| 31.94|
| Cat           | 24.44 | **28.69** | 28.37      | 27.34| 27.28| 27.41| 27.68|

|               | Noisy | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------------|-------|-----------|------------|-----|-----|-----|-----|
| Portrait      | 0.6300| 0.9514    | **0.9508** | 0.9379| 0.9132| 0.9022| **0.9623**|
| Peppers       | 0.9174| **0.9822**| 0.9817     | 0.9802| 0.9755| 0.9781| 0.9803|
| Fruits        | 0.8149| **0.9676**| 0.9673     | 0.9611| 0.9486| 0.9518| 0.9649|
| Cat           | 0.6919| **0.8839**| 0.8778     | 0.8465| 0.8435| 0.8493| 0.8576|

Table 1: Comparison of the proposed models with existing models on images with Gaussian noise and SD=0.06. (a) PSNR values of the noisy image and denoised images by different methods. (b) SSIM values of the noisy image and denoised images by different methods. The largest value for each experiment (row) is marked in bold.

|               | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------------|-----------|------------|-----|-----|-----|-----|
| Portrait (295 × 263) | 385       | 182        | 196 | 118 | 100 | 492 |
| Peppers (512 × 512)   | 532       | 245        | 262 | 113 | 119 | 461 |
| Fruits (512 × 512)    | 455       | 211        | 244 | 114 | 113 | 427 |
| Cat (204 × 213)       | 654       | 261        | 374 | 65  | 116 | 342 |

|               | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------------|-----------|------------|-----|-----|-----|-----|
| Portrait (295 × 263) | 61.92     | 25.63      | 64.44| 2.07| 1.28| 8.86|
| Peppers (512 × 512)   | 489.72    | 190.02     | 346.27| 8.75| 5.20| 34.05|
| Fruits (512 × 512)    | 475.37    | 164.47     | 331.49| 8.68| 4.60| 31.43|
| Cat (204 × 213)       | 40.48     | 15.77      | 41.69| 0.49| 0.55| 2.48|

Table 2: Comparison of the proposed models with existing models on images with Gaussian noise and SD=0.06. (a) Number of iterations used to satisfy stopping criterion. (b) CPU time (in seconds) used to satisfy stopping criterion.
Figure 11: Benchmark images from [55]. Images in the first row: 'Wall', 'Flower', 'Classroom' and 'Pens'. Images in the second row: 'Plant', 'Toy', 'Wires' and 'Stairs'.

|                | Noisy | Model (8) | Model (10) | CE   | PA   | CTV   | VTV   |
|----------------|-------|-----------|------------|------|------|-------|-------|
| Wall           | 24.91 | 33.91     | 33.84      | 33.02| 31.63| 32.10 | 33.57 |
| Flower         | 24.50 | 35.14     | 35.08      | 34.73| 33.39| 32.93 | 34.02 |
| Classroom      | 25.35 | 34.41     | 34.49      | 33.64| 32.39| 32.59 | 34.05 |
| Pens           | 24.74 | 33.38     | 33.50      | 32.41| 29.78| 31.05 | 31.96 |
| Plant          | 24.60 | 33.30     | 33.46      | 32.24| 30.20| 31.13 | 31.91 |
| Toy            | 24.53 | 35.56     | 35.40      | 34.92| 33.64| 33.15 | 35.11 |
| Wires          | 24.74 | 34.06     | 34.15      | 33.56| 31.38| 31.19 | 32.78 |
| Stairs         | 24.47 | 36.65     | 36.59      | 36.17| 36.22| 33.86 | 36.22 |
| Average        | 24.73 | 34.55     | 34.56      | 33.84| 32.33| 32.38 | 33.70 |

|                | Noisy | Model (8) | Model (10) | CE   | PA   | CTV   | VTV   |
|----------------|-------|-----------|------------|------|------|-------|-------|
| Wall           | 0.8724| 0.9836    | 0.9860     | 0.9816| 0.9766| 0.9744| 0.9817|
| Flower         | 0.6866| 0.9600    | 0.9583     | 0.9564| 0.9469| 0.9348| 0.9506|
| Classroom      | 0.8350| 0.9651    | 0.9655     | 0.9624| 0.9587| 0.9540| 0.9637|
| Pens           | 0.7070| 0.9589    | 0.9581     | 0.9502| 0.9230| 0.9235| 0.9585|
| Plant          | 0.6326| 0.9503    | 0.9491     | 0.9388| 0.9118| 0.9072| 0.9506|
| Toy            | 0.7872| 0.9767    | 0.9757     | 0.9740| 0.9711| 0.9627| 0.9743|
| Wires          | 0.6424| 0.9511    | 0.9485     | 0.9434| 0.9228| 0.9069| 0.9519|
| Stairs         | 0.6995| 0.9731    | 0.9723     | 0.9702| 0.9711| 0.9461| 0.9730|
| Average        | 0.7328| 0.9648    | 0.9639     | 0.9596| 0.9477| 0.9387| 0.9630|

Table 3: Comparison of the proposed models with existing models on benchmark images in Figure 11. The noisy images are generated by adding Gaussian noise with SD=0.06. (a) PSNR values of the noisy image and denoised images by different methods. (b) SSIM values of the noisy image and denoised images by different methods. The largest value for each experiment (row) is marked in bold.
Table 4: Comparison of the proposed models with existing models on benchmark images in Figure 11 with large noise. The noisy images are generated by adding Gaussian noise with SD=0.2. (a) PSNR values of the noisy image and denoised images by different methods. (b) SSIM values of the noisy image and denoised images by different methods. The largest value for each experiment (row) is marked in bold.

|         | Noisy | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------|-------|-----------|------------|-----|-----|-----|-----|
| Flower  | 14.42 | 26.64     | **28.62**  | 28.22 | 27.41 | 26.36 | 28.28 |
| Pens    | 15.06 | **25.37**  | 25.28      | 24.32 | 23.83 | 24.15 | 24.67 |
| Plant   | 15.01 | **25.95**  | 25.85      | 25.02 | 24.74 | 24.76 | 25.33 |
| Wires   | 15.20 | 26.01     | **26.03**  | 25.54 | 24.96 | 24.96 | 25.47 |
| Average | 14.92 | **26.49**  | 26.45      | 25.76 | 25.24 | 25.06 | 25.94 |

Table 5: Comparison of the proposed models with existing models on images with Gaussian noise and SD=0.15. (a) PSNR values of the noisy image and denoised images by different methods. (b) SSIM values of the noisy image and denoised images by different methods. The largest value for each experiment (row) is marked in bold.

|         | Noisy | Model (8) | Model (10) | CE  | PA  | CTV | VTV |
|---------|-------|-----------|------------|-----|-----|-----|-----|
| Flower  | 0.2018 | 0.8746    | **0.8767** | 0.8643 | 0.8241 | 0.7797 | 0.8688 |
| Pens    | 0.3131 | 0.8521    | **0.8553** | 0.8246 | 0.7774 | 0.7610 | 0.8521 |
| Plant   | 0.2165 | 0.8292    | 0.8330     | 0.7972 | 0.7379 | 0.7085 | **0.8415** |
| Wires   | 0.2900 | 0.8542    | **0.8619** | 0.8368 | 0.7832 | 0.7441 | 0.8605 |
| Average | 0.2554 | 0.8525    | **0.8567** | 0.8307 | 0.7807 | 0.7483 | 0.8557 |

for CE, $\lambda = 2.5$ for CTV and $\lambda = 0.3$ for VTV. In this comparison, both model (8) and (10) give results with larger PSNR than other models. The increment is about 0.5 on average. For the SSIM value, on average, model (10) gives the best results.

We further compare all models on three images with large noise in Figure 12–13. The clean images and noisy images containing Gaussian noise with SD=0.15 are shown in (a) and (b), respectively. Denoised images by model (8), model (10), CE, PA, CTV and VTV are shown in (c)–(h), respectively. In this set of experiments, we set $\alpha = 5 \times 10^{-4}$, $\beta = 50$, $\eta = 7$ for model (8), $\alpha = 5 \times 10^{-3}$, $\beta = 30$, $\eta = 2.4$ for model (10). For other models, we use $\alpha = 1 \times 10^{-2}$, $\beta = 5 \times 10^{-3}$, $\eta = 2.5$ for CE, $\lambda = 2.5$ for CTV and $\lambda = 0.25$ for VTV. The proposed models provide the best results which recover the features with uniform color themes. For results by other models, either they have strong smoothing
Figure 12: Comparison of the proposed models and existing models on an image with large noise. (a) Clean image. (b) Noisy image with SD=0.15. (c) Denoised image by model (8). (d) Denoised image by model (10). (e) Denoised image by CE. (f) Denoised image by PA. (g) Denoised image by CTV. (h) Denoised image by VTV.

effects or contain pixels with abrupt colors. To quantify the differences, the PSNR and SSIM values of the denoised images are shown in Table 5. Again, results by the proposed models have the largest values.

In the last example, we compared all algorithms on images with very large noise, SD=0.3 in Figure 14 and SD=0.5 in Figure 15. For the experiments in Figure 14, we set $\alpha = 5 \times 10^{-4}, \beta = 50, \eta = 12$ for model (8), $\alpha = 5 \times 10^{-3}, \beta = 30, \eta = 3$ for model (10). For other models, we use $\alpha = 1 \times 10^{-2}, \beta = 5 \times 10^{-3}, \eta = 3.5$ for CE, $\lambda = 1$ for CTV and $\lambda = 0.6$ for VTV. Our results are presented in Figure 14. For the experiments in Figure 15, we set $\alpha = 5 \times 10^{-4}, \beta = 50, \eta = 18$ for model (8), $\alpha = 5 \times 10^{-3}, \beta = 30, \eta = 5$ for model (10). For other models, we use $\alpha = 1 \times 10^{-2}, \beta = 5 \times 10^{-3}, \eta = 7$ for CE, $\lambda = 0.8$ for CTV and $\lambda = 0.8$ for VTV. Again, the proposed models (8) and (10) give the best results which recover the features best.
7 Conclusion

We propose in this article two modified color elastica models for vector-valued image regularization. Compared to the original color elastica model, model (8) multiplies the Laplace-Beltrami term by the image metric $g$, and model (10) utilizes the relation between the surface area regularizer and total variation regularizer. Both models reduces to Euler’s elastica model for gray-scale images. For each proposed model, we introduced an operator-splitting method to find the minimizer. The nonlinearity is decoupled by introducing matrix- and vector-valued variables. Then, finding the minimizer is converted to solving an associated initial value problem, which is time-discretized by an operator-splitting method. Each subproblem after splitting either has a closed-form solution or can be solved efficiently. The advantages of the proposed models are demonstrated by systematic numerical experiments. Compared to existing models, the proposed models give more natural recoveries of images with less abrupt or artificial colors, and better PSNR and
Figure 14: Comparison of the proposed models and existing models on an image with large noise. (a) Clean image. (b) Noisy image with SD=0.3. (c) Denoised image by model (8). (d) Denoised image by model (10). (e) Denoised image by CE. (f) Denoised image by PA. (g) Denoised image by CTV. (h) Denoised image by VTV.

Acknowledgment

The authors would like to thank the anonymous reviewers of this article for most helpful comments and suggestions.
Figure 15: Comparison of the proposed models and existing models on an image with large noise. (a) Clean image. (b) Noisy image with SD=0.5. (c) Denoised image by model (8). (d) Denoised image by model (10). (e) Denoised image by CE. (f) Denoised image by PA. (g) Denoised image by CTV. (h) Denoised image by VTV.

Appendix

A  Relation between the color elastica model (6) and Euler’s elastica model

We discuss the color elastica model (6) for one-channel images and its relation with Euler’s elastica model. This discussion follows from [30, Remark 3.2]. For a one-channel image $v$, model (6) becomes

$$
\min_{v \in H^2(\Omega)} \sqrt{\alpha} \int_{\Omega} \left( 1 + \frac{\beta}{\alpha + |\nabla v|^2} \left( \nabla \cdot \frac{\nabla v}{\sqrt{\alpha + |\nabla v|^2}} \right)^2 \right) \sqrt{\alpha + |\nabla v|^2} dx
+ \frac{1}{2\eta} \int_{\Omega} |v - f|^2 dx. \tag{88}
$$
Divide the first integral of (88) by $\sqrt{\alpha}$ and let $\alpha \to 0$, one gets

$$\min_{v \in H^2(\Omega)} \int_{\Omega} \left( 1 + \beta \frac{1}{|\nabla v|^2} \left( \nabla \cdot \frac{\nabla v}{|\nabla v|} \right)^2 \right) |\nabla v| \, dx + \frac{1}{2\eta} \int_{\Omega} |v - f|^2 \, dx. \quad (89)$$

The first term in (89) is a variant of Euler’s elastica model, except the term $\left( \nabla \cdot \frac{\nabla v}{|\nabla v|} \right)^2$ is weighted by $1/|\nabla v|^2$.

References

[1] Y. Bai and S. Mei. Analysis of sequential quadratic programming through the lens of riemannian optimization. arXiv preprint arXiv:1805.08756, 2018.

[2] L. Bar, A. Brook, N. Sochen, and N. Kiryati. Deblurring of color images corrupted by impulsive noise. IEEE Transactions on Image Processing, 16(4):1101–1111, 2007.

[3] T. Batard and M. Bertalmio. A geometric model of brightness perception and its application to color images correction. Journal of Mathematical Imaging and Vision, 60(6):849–881, 2018.

[4] T. Batard, J. Hertrich, and G. Steidl. Variational models for color image correction inspired by visual perception and neuroscience. Journal of Mathematical Imaging and Vision, 62(9):1173–1194, 2020.

[5] P. Blomgren and T. F. Chan. Color TV: total variation methods for restoration of vector-valued images. IEEE transactions on image processing, 7(3):304–309, 1998.

[6] X. Bresson and T. F. Chan. Fast dual minimization of the vectorial total variation norm and applications to color image processing. Inverse Problems & Imaging, 2(4):455, 2008.

[7] M. Bukač, S. Čanić, R. Glowinski, J. Tambača, and A. Quaini. Fluid–structure interaction in blood flow capturing non-zero longitudinal structure displacement. Journal of Computational Physics, 235:515–541, 2013.

[8] A. Chambolle. An algorithm for total variation minimization and applications. Journal of Mathematical imaging and vision, 20(1):89–97, 2004.

[9] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of mathematical imaging and vision, 40(1):120–145, 2011.

[10] L.-J. Deng, R. Glowinski, and X.-C. Tai. A new operator splitting method for the Euler elastica model for image smoothing. SIAM Journal on Imaging Sciences, 12(2):1190–1230, 2019.

[11] S. Di Zenzo. A note on the gradient of a multi-image. Computer vision, graphics, and image processing, 33(1):116–125, 1986.
[12] Y. Duan, W. Huang, J. Zhou, H. Chang, and T. Zeng. A two-stage image segmentation method using Euler’s elastica regularized Mumford-Shah model. In 2014 22nd International Conference on Pattern Recognition, pages 118–123. IEEE, 2014.

[13] Y. Duan, Y. Wang, and J. Hahn. A fast augmented Lagrangian method for Euler’s elastica models. *Numerical Mathematics: Theory, Methods and Applications*, 6(1):47–71, 2013.

[14] Y. Duan, Q. Zhong, X.-C. Tai, and R. Glowinski. A fast operator-splitting method for Beltrami color image denoising. *Journal of Scientific Computing*, 92(3):1–28, 2022.

[15] V. Duval, J.-F. Aujol, and L. Vese. Projected gradient based color image decomposition. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 295–306. Springer, 2009.

[16] R. Glowinski, S. Leung, H. Liu, and J. Qian. On the numerical solution of nonlinear eigenvalue problems for the Monge-Ampère operator. *ESAIM: Control, Optimisation & Calculus of Variations*, 26, 2020.

[17] R. Glowinski, S. Leung, and J. Qian. A penalization-regularization-operator splitting method for eikonal based traveltime tomography. *SIAM Journal on Imaging Sciences*, 8(2):1263–1292, 2015.

[18] R. Glowinski, H. Liu, S. Leung, and J. Qian. A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge–Ampère equation. *Journal of Scientific Computing*, 79(1):1–47, 2019.

[19] R. Glowinski, S. J. Osher, and W. Yin. *Splitting methods in communication, imaging, science, and engineering*. Springer, 2017.

[20] R. Glowinski, T.-W. Pan, and X.-C. Tai. Some facts about operator-splitting and alternating direction methods. In *Splitting Methods in Communication, Imaging, Science, and Engineering*, pages 19–94. Springer, 2016.

[21] B. Goldluecke, E. Strekalovskiy, and D. Cremers. The natural vectorial total variation which arises from geometric measure theory. *SIAM Journal on Imaging Sciences*, 5(2):537–563, 2012.

[22] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM journal on imaging sciences*, 2(2):323–343, 2009.

[23] Y. He, S. H. Kang, and H. Liu. Curvature regularized surface reconstruction from point clouds. *SIAM Journal on Imaging Sciences*, 13(4):1834–1859, 2020.

[24] S. H. Kang, X.-C. Tai, and W. Zhu. Survey of fast algorithms for Euler’s elastica-based image segmentation. In *Handbook of Numerical Analysis*, volume 20, pages 533–552. Elsevier, 2019.

[25] R. Kimmel. A natural norm for color processing. In *Asian Conference on Computer Vision*, pages 88–95. Springer, 1998.
[26] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. *International Journal of Computer Vision*, 39(2):111–129, 2000.

[27] R. Kimmel, N. Sochen, and R. Malladi. From high energy physics to low level vision. In *International Conference on Scale-Space Theories in Computer Vision*, pages 236–247. Springer, 1997.

[28] H. Liu, R. Glowinski, S. Leung, and J. Qian. A finite element/operator-splitting method for the numerical solution of the three dimensional Monge–Ampère equation. *Journal of Scientific Computing*, 81(3):2271–2302, 2019.

[29] H. Liu, X.-C. Tai, and R. Glowinski. An operator-splitting method for the gaussian curvature regularization model with applications in surface smoothing and imaging. *arXiv preprint arXiv:2108.01914*, 2021.

[30] H. Liu, X.-C. Tai, R. Kimmel, and R. Glowinski. A color elastaic model for vector-valued image regularization. *SIAM Journal on Imaging Sciences*, 14(2):717–748, 2021.

[31] H. Liu and D. Wang. Fast operator splitting methods for obstacle problems. *arXiv preprint arXiv:2203.08380*, 2022.

[32] J. Liu, M. Yan, and T. Zeng. Surface-aware blind image deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[33] Z. Liu, S. Wali, Y. Duan, H. Chang, C. Wu, and X.-C. Tai. Proximal ADMM for Euler’s elastica based image decomposition model. *Numer. Math. Theory Methods Appl.*, 12(2):370–402, 2018.

[34] S. K. Naik and C. Murthy. Hue-preserving color image enhancement without gamut problem. *IEEE Transactions on image processing*, 12(12):1591–1598, 2003.

[35] M. Nikolova and G. Steidl. Fast hue and range preserving histogram specification: Theory and new algorithms for color image enhancement. *IEEE transactions on image processing*, 23(9):4087–4100, 2014.

[36] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 1999.

[37] F. Pierre, J.-F. Aujol, A. Bugeau, G. Steidl, and V.-T. Ta. Variational contrast enhancement of gray-scale and RGB images. *Journal of Mathematical Imaging and Vision*, 57(1):99–116, 2017.

[38] A. M. Polyakov. Quantum geometry of bosonic strings. *Physics Letters B*, 103(3):207–210, 1981.

[39] G. Rosman, L. Dascal, A. Sidi, and R. Kimmel. Efficient Beltrami image filtering via vector extrapolation methods. *SIAM Journal on Imaging Sciences*, 2(3):858–878, 2009.

[40] G. Rosman, L. Dascal, X.-C. Tai, and R. Kimmel. On semi-implicit splitting schemes for the Beltrami color image filtering. *Journal of Mathematical Imaging and Vision*, 40(2):199–213, 2011.
[41] G. Rosman, X.-C. Tai, L. Dascal, and R. Kimmel. Polyakov action minimization for efficient color image processing. In European Conference on Computer Vision, pages 50–61. Springer, 2010.

[42] A. Roussos and P. Maragos. Tensor-based image diffusions derived from generalizations of the total variation and Beltrami functionals. In 2010 IEEE International Conference on Image Processing, pages 4141–4144. IEEE, 2010.

[43] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena, 60(1-4):259–268, 1992.

[44] G. Sapiro. Vector-valued active contours. In Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 680–685. IEEE, 1996.

[45] G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. IEEE transactions on image processing, 5(11):1582–1586, 1996.

[46] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. IEEE transactions on image processing, 7(3):310–318, 1998.

[47] A. Spira, R. Kimmel, and N. Sochen. A short-time Beltrami kernel for smoothing images and manifolds. IEEE Transactions on Image Processing, 16(6):1628–1636, 2007.

[48] X.-C. Tai, J. Hahn, and G. J. Chung. A fast algorithm for Euler’s elastica model using augmented Lagrangian method. SIAM Journal on Imaging Sciences, 4(1):313–344, 2011.

[49] L. Tan, W. Liu, and Z. Pan. Color image restoration and inpainting via multi-channel total curvature. Applied Mathematical Modelling, 61:280–299, 2018.

[50] P. E. Trahanias and A. N. Venetsanopoulos. Color image enhancement through 3-D histogram equalization. In 11th IAPR International Conference on Pattern Recognition. Vol. III. Conference C: Image, Speech and Signal Analysis., volume 1, pages 545–548. IEEE Computer Society, 1992.

[51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4):600–612, 2004.

[52] Z. Wang, J. Zhu, F. Yan, and M. Xie. Fidelity-Beltrami-sparsity model for inverse problems in multichannel image processing. SIAM Journal on Imaging Sciences, 6(4):2685–2713, 2013.

[53] J. Weickert. Coherence-enhancing diffusion of colour images. Image and Vision Computing, 17(3-4):201–212, 1999.

[54] A. Wetzler and R. Kimmel. Efficient Beltrami flow in patch-space. In International Conference on Scale Space and Variational Methods in Computer Vision, pages 134–143. Springer, 2011.
[55] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018.

[56] M. Yashtini and S. H. Kang. Alternating direction method of multiplier for Euler’s elastica-based denoising. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 690–701. Springer, 2015.

[57] M. Yashtini and S. H. Kang. A fast relaxed normal two split method and an effective weighted TV approach for Euler’s elastica image inpainting. *SIAM Journal on Imaging Sciences*, 9(4):1552–1581, 2016.

[58] A. Yezzi. Modified curvature motion for image smoothing and enhancement. *IEEE Transactions on Image Processing*, 7(3):345–352, 1998.

[59] J. Zhang, R. Chen, C. Deng, and S. Wang. Fast linearized augmented Lagrangian method for Euler’s elastica model. *Numerical Mathematics: Theory, Methods and Applications*, 10(1):98–115, 2017.

[60] D. Zosso and A. Bustin. A primal-dual projected gradient algorithm for efficient Beltrami regularization. *Computer Vision and Image Understanding*, pages 14–52, 2014.