

Segmentation by Elastica Energy with L^1 and L^2 Curvatures: a Performance Comparison

Xuan He^{1,*}, Wei Zhu¹ and Xue-Cheng Tai²

¹ *Department of Mathematics, University of Alabama, Box 870350, Tuscaloosa, AL 35487, USA*

² *Department of Mathematics, Hong Kong Baptist University, Kowloon Tong Kowloon, Hong Kong*

Received 12 April 2017; Accepted (in revised version) 7 February 2018

Abstract. In this paper, we propose an algorithm based on augmented Lagrangian method and give a performance comparison for two segmentation models that use the L^1 - and L^2 -Euler's elastica energy respectively as the regularization for image segmentation. To capture contour curvature more reliably, we develop novel augmented Lagrangian functionals that ensure the segmentation level set function to be signed distance functions, which avoids the reinitialization of segmentation function during the iterative process. With the proposed algorithm and with the same initial contours, we compare the performance of these two high-order segmentation models and numerically verify the different properties of the two models.

AMS subject classifications: 68U10, 65K10, 49M05

Key words: Augmented Lagrangian method, Euler's elastica, image segmentation.

1. Introduction

Image segmentation is a typical problem in image processing, with a broad range of applications in medical image analysis, object detection, recognition, etc. It aims to partition a given image domain into several disjoint regions, each of which describes either a meaningful object or background. During the last few decades, numerous variational models have been developed for this problem. These include the snake and active contour model by Kass, Witkin, and Terzopoulos [17], the Mumford-Shah model [23], the geodesic active contour model by Caselles, Kimmel, and Sapiro [7], and the Chan-Vese model [11], to name a few. The Chan-Vese model can be regarded as a special case of the Mumford-Shah model by confining the approximation functions to be binary functions, and an attractive feature of the Chan-Vese model is its treatment of segmentation contours using level set functions [25].

*Corresponding author. *Email addresses:* xhe12@crimson.ua.edu (X. He), wzhu7@ua.edu (W. Zhu), xuechengtai@hkbu.edu.hk (X. C. Tai)

Recently, in [35], we considered a modification of the Chan-Vese (CV) model by employing the Euler’s elastica energy as the regularization of segmentation contour. Euler’s elastica energy was first seriously studied for visual perception by Mumford [22], and it has been widely utilized as a regularizer in image inpainting [1, 2, 10], capturing illusory contours [24, 33], and image denoising [29]. Benefited from the attributes of this high-order regularizer, the modified CV model (*ECV-L²* model) presents several new features when compared with the original CV model for image segmentation (cf. [35]): 1) automatically connecting broken parts of objects; 2) capturing objects of large size while omitting small ones; 3) being more suited than the CV model for keeping elongated structures.

Later on, another variant of the Chan-Vese model was discussed in [4], where the L^1 variant of Euler’s elastica was taken as the new regularization term of segmentation contours. The most remarkable feature of this new segmentation model lies in the fact that it privileges convex contours once a strong weight is imposed on the curvature term, which is supported by the theorem [21] in differential geometry that the integral of the magnitude of curvature along any closed piecewise smooth curve is greater than or equal to 2π , and the minimum value is attained only when the closed curve is convex.

To present these two variants of the Chan-Vese model, we recall the standard Euler’s elastica that refers to a curve Γ that minimizes the elasticity energy

$$\mathbf{E}(\Gamma) = \int_{\Gamma} (a + b\kappa^2) ds \tag{1.1}$$

among all curves satisfying some boundary conditions, where κ represents the curvature of curves and $a, b > 0$ are two parameters. The L^1 -variant of Euler’s elastica energy can be expressed as

$$\mathbf{E}(\Gamma) = \int_{\Gamma} (a + b|\kappa|) ds. \tag{1.2}$$

This elastica energy linearly depends on the magnitude of curvature, which helps maintain corners during the segmentation process, as discussed in the well-known segmentation with depth model by Nitzberg, Mumford, Shiota [24].

By incorporating Euler’s elastica energy, those two modified Chan-Vese models can be written in the level set setting as follows:

$$\begin{aligned} \min_{\{\phi, c_1, c_2\}} E(\phi, c_1, c_2) = & \min_{\{\phi, c_1, c_2\}} \int_{\Omega} (f - c_1)^2 H(\phi) + (f - c_2)^2 (1 - H(\phi)) \\ & + \int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right|^2 \right) |\nabla H(\phi)|, \end{aligned} \tag{1.3a}$$

$$\begin{aligned} \min_{\{\phi, c_1, c_2\}} E(\phi, c_1, c_2) = & \min_{\{\phi, c_1, c_2\}} \int_{\Omega} (f - c_1)^2 H(\phi) + (f - c_2)^2 (1 - H(\phi)) \\ & + \int_{\Omega} \left(a + b \left| \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} \right| \right) |\nabla H(\phi)|, \end{aligned} \tag{1.3b}$$

where $f : \Omega \rightarrow \mathbb{R}^1$, ϕ is a level set function whose zero contour locates the segmentation contours, $H(\cdot)$ is the Heaviside function, and c_1, c_2 are constants. Moreover, $\nabla \cdot (\nabla \phi / |\nabla \phi|)$ represents the mean curvature of level curves of ϕ [25]. The positive parameters a, b are able to balance the fitting term and the Euler's elastica energy. Just as the $ECV-L^2$ model (1.3a), the second one is called as the $ECV-L^1$ model in the following contexts.

As both models incorporate Euler's elastica energy as the regularization, they share similar features, such as connecting broken parts of objects and putting small objects into background. However, the difference in their exponents of curvature leads to dramatical discrepancies in their performance:

- The $ECV-L^1$ model is able to keep object corners, while the $ECV-L^2$ smears them unavoidably because of its quadratic dependence of curvature of segmentation contours;
- The $ECV-L^1$ model privileges convex contours when the curvature term is large while the $ECV-L^2$ model presents no such a property;
- Again due to the quadratic dependence of curvature, the $ECV-L^2$ model is more inclined to remove small objects than the $ECV-L^1$ model.

Even though these segmentation models possess such interesting features, it is nontrivial to develop reliable numerical schemes for those models. The main difficulties originate from their high-order and non-convex attributes and also a precise way of capturing curvature during the minimization processes of these models numerically.

Recently, lots of research works have focused on the development of fast and reliable numerical methods for minimizing curvature based functionals, such as the multigrid algorithm [6], the homotopy method [31], graph cut based algorithms [3, 13], linear programming method [28], and augmented Lagrangian method (ALM) based algorithms [15, 27].

Specifically, augmented Lagrangian method based algorithms have been proposed to deal with non-differentiable and/or high-order functionals in image processing [29, 30], see also [16] for a more detailed survey on this. The merit of using ALMs lies in the fact that the original minimization problem of non-differentiable and/or high-order functionals can be converted to be the seek of minimizers of several lower-order functionals or subproblems, which can be solved either using closed-form solutions or using Fast solver like the Fast Fourier transformation (FFT). Therefore, by utilizing ALMs, the original minimization problem of high-order functionals becomes more tractable numerically. However, as a trade-off, as shown in [34], the pursuit of simple form of those subproblems inevitably leads to too many subproblems that need to be solved, which naturally raises the issue of finding appropriate penalty parameter for each dual variable. Therefore, the difficulty of the original problem is transferred to the choice of penalty parameters that ensure the convergence to saddle points of the corresponding augmented Lagrangian functional. Based on this argument, in [4], we proposed making use of fewer dual variables at the expense of solving more complicated subproblems, which balances the two difficulties of dealing with the minimization of high-order functionals and finding suitable penalty parameters of Lagrange multipliers.

In this work, we intend to study and compare the performance of these two segmentation models numerically by designing novel ALM based algorithms. As discussed above, one of the most important concerns of treating these elastica based models is how to reliably express the curvature term. As mentioned in [4], once the coefficient of curvature is large, it is advisable to apply the reinitialization technique [26] to restore the segmentation function to be a signed distance function. However, an issue is often accompanied with the application of this procedure, that is, how often it should be taken during the iteration. It will be too expensive to be carried out for each iteration, and an excessive reinitialization also prevents the propagation of segmentation contour. To avoid such an issue, we add an extra constraint requiring the segmentation function to be signed distance functions. The idea is very similar to but slightly different from Chunming Li et al.'s work in [19]. In [19], they avoid the reinitialization by adding a penalization term in the energy functional. In our work, we impose the signed distance requirement as a constraint in the optimization problem, which will be illustrated in detail in Section 3.

To see how signed distance functions help capture curvature numerically, we consider two different functions ϕ_1 and ϕ_2 but with the same zero level curve as follows:

1. $\phi_1(x, y) = 1 - (\frac{x}{a})^2 - (\frac{y}{b})^2$. ϕ_1 is defined on a 600×400 grid lattice with its center being the origin point, and $a = 200$ and $b = 100$.
2. ϕ_2 is a signed distance function that shares the same positive and negative region as ϕ_1 and is constructed using the method in [12].

It is easy to see that ϕ_1 is not a signed distance function. Numerical calculation shows that the term $\int |\kappa_{\phi_1}| |\nabla H_\epsilon(\phi_1)| dx dy = 2.03$ for $\epsilon = 1.5$, where H_ϵ is a regularized version of the Heaviside function that will be discussed later. However, similar calculation yields $\int |\kappa_{\phi_2}| |\nabla H_\epsilon(\phi_2)| dx dy = 6.22$, which is very close to the exact value 2π , since the zero level curve is closed and convex [21]. This simple comparison indicates that it helps represent curvature reliably by requiring ϕ to be a signed distance function.

The outline of the paper is as follows. In Section 2, we provide a simple but heuristic example to show the difference between the two elastica regularizers on the preservation of corners. In Section 3, we propose novel augmented Lagrangian method based algorithms for the $ECV-L^1$ and the $ECV-L^2$ segmentation models and also present the details of solving those corresponding subproblems. In Section 4, we discuss the numerical implementation, and then present the numerical results and the comparison for the two models by using the proposed algorithms in Section 5. Section 6 is devoted to our conclusions.

2. A geometric difference between L^1 and L^2 -Euler's elastica

In this section, we use a simple example to demonstrate the fact that the L^1 -Euler's elastica is able to preserve corners while the L^2 -Euler's elastica will erode them inevitably. As shown in Fig. 1, we consider a red wedge with a sharp corner, and try to calculate the two elastica energies along the red lines, each of which has the length L . Since the curvature is not well-defined at the corner, we just do the integrals along its smoothed

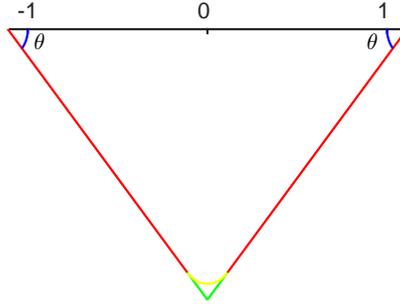


Figure 1: A red wedge and its corner-smoothed version.

version denoted as Γ_r , that is, the corner part is replaced by a small circular arc that is tangent to both red edges and has a radius r , and then let the radius r go to zero.

Note that the magnitude of curvature is zero along the red edges and is $1/r$ along the yellow arc, direct calculation leads to the following:

$$\int_{\Gamma_r} (a + b|\kappa|) ds = 2a(L - r \tan \theta + r\theta) + 2b\theta, \quad (2.1a)$$

$$\int_{\Gamma_r} (a + b\kappa^2) ds = 2a(L - r \tan \theta + r\theta) + 2b\theta/r. \quad (2.1b)$$

As $r \rightarrow 0$, the L^2 -Euler's elastica goes to ∞ , which indicates that the corner may not be preserved, while for L^1 -Euler's elastica, the limit is just $2aL + 2b\theta$, which means that a small value of r is allowed. From this simple calculation, another interesting phenomenon is that with $a > 0$, the L^1 -Euler's elastica energy gives a smaller value for a larger $r > 0$. This is because that $\theta < \tan \theta$ for $\theta \in (0, \pi/2)$. However, as shown in Fig. 2, for the $ECV-L^1$ segmentation model, the L^1 -Euler's elastica energy also needs to compete with the fitting term, and therefore, how well the corner is preserved depends on how small the parameter a is. Both facts explain the results in Fig. 2: with a small value of a , the $ECV-L^1$ model almost keeps the corner completely, while the $ECV-L^2$ model unavoidably removes it.

3. Novel augmented Lagrangian method

Recently, ALMs have proved to be very successful for dealing with non-differentiable and/or high-order functionals in image processing [16, 29, 30]. In this section, inspired by our previous works [4, 35], we propose novel ALM based algorithms for the two Euler's elastica based segmentation models in order to compare their performance.

3.1. New augmented Lagrangian functionals

When compared with other standard methods, such as solving the associated Euler-Lagrange equation, it becomes relatively easier to get minimizers of the original high-order

and/or non-differential functionals by using ALMs. In fact, based on optimization theory, finding the minimizer of the original optimization problem can be solved by finding saddle points of the augmented Lagrangian functional. For this, one just needs to solve several associated sub-problems alternatingly and repeatedly until all those variables converge to steady states.

It might be tempting to establish a binary formulation of the model, but as is pointed out in [4], when standard central difference is used for segmentation problems, it is more accurate to measure the curvature by formulating $H(\phi)$ as a continuous function with respect to ϕ than formulating $H(\phi)$ as a relaxed binary function. Moreover, the continuous formulation requires one less auxiliary variable in the augmented functional. To this end, as in [11], we approximate the function $H(\phi)$ by its regularized version H_ϵ as follows:

$$H_\epsilon(\phi) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{\phi}{\epsilon}\right),$$

$$\delta_\epsilon(\phi) = \frac{\epsilon}{\pi(\epsilon^2 + \phi^2)},$$

where δ_ϵ represents the derivative of H_ϵ . In this paper, we choose $\epsilon = 1.5$.

ALM is shown to be able to handle non-differential TV regularity term [30] and mean curvature regularity term through the idea of splitting variables [29]. In this work, we use the same split technique for the minimization of functionals (1.3a) and (1.3b) by converting them to be the following constrained optimization problems:

$$\min_{\phi, \mathbf{p}, \mathbf{n}, \mathbf{q}, \mathbf{m}, c_1, c_2} \int_{\Omega} (f - c_1)^2 H_\epsilon(\phi) + (f - c_2)^2 (1 - H_\epsilon(\phi)) + (a + b|q|) \delta_\epsilon |\mathbf{p}|,$$

$$s.t. \mathbf{p} = \nabla \phi, \mathbf{n} = \frac{\mathbf{p}}{|\mathbf{p}|}, \mathbf{q} = \nabla \cdot \mathbf{n}, \mathbf{p} = \mathbf{m}, |\mathbf{m}| = 1, \quad (3.1a)$$

$$\min_{\phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, c_1, c_2} \int_{\Omega} (f - c_1)^2 H_\epsilon(\phi) + (f - c_2)^2 (1 - H_\epsilon(\phi)) + (a + b(\nabla \cdot \mathbf{n})^2) \delta_\epsilon |\mathbf{p}|,$$

$$s.t. \mathbf{p} = \nabla \phi, \mathbf{n} = \frac{\mathbf{p}}{|\mathbf{p}|}, \mathbf{p} = \mathbf{m}, |\mathbf{m}| = 1. \quad (3.1b)$$

The auxiliary variables $\mathbf{p}, \mathbf{n}, \mathbf{m}, \mathbf{q}$ are introduced in order to convert the original nonlinear and non-convex minimization problem to be a simpler constrained optimization problem. The main difference between the above augmented Lagrangian methods and those in the previous works [4, 35] is the incorporation of the new constraint $|\mathbf{p}| = 1$, which takes effect through a new variable \mathbf{m} and a new constraint $\mathbf{p} = \mathbf{m}$. The reason of adding such a constraint is to help represent curvature more faithfully than using general level set functions, which is supported by the ‘‘ellipse’’ example discussed in the introduction. In this example, by checking the value of the integral $\int_{\{\phi=0\}} |\kappa| ds$, we found that it is far from 2π even though the closed curve $\{\phi = 0\}$ is convex. However, once we require ϕ to be a signed distance function, this quantity is very close to 2π , which indicates that the curvature can be captured reliably by using signed distance functions.

We then set up the following augmented Lagrangian dual functionals for the $ECV-L^1$ and $ECV-L^2$ models respectively.

$$\begin{aligned}
& \mathcal{L}(\phi, \mathbf{p}, \mathbf{n}, q, \mathbf{m}, c_1, c_2, \lambda_1, \lambda_2, \lambda_3, \lambda_4) \\
&= \int (f - c_1)^2 H_\epsilon(\phi) + (f - c_2)^2 (1 - H_\epsilon(\phi)) + \int (a + b|q|) \delta_\epsilon(\phi) |\mathbf{p}| \\
&\quad + \frac{r_1}{2} \int |\mathbf{p} - \nabla \phi|^2 + \int \lambda_1 \cdot (\mathbf{p} - \nabla \phi) + \frac{r_2}{2} \int \left| |\mathbf{p}| \mathbf{n} - \mathbf{p} \right|^2 \\
&\quad + \int \lambda_2 \cdot (|\mathbf{p}| \mathbf{n} - \mathbf{p}) + \frac{r_3}{2} \int |\mathbf{p} - \mathbf{m}|^2 + \int \lambda_3 \cdot (\mathbf{p} - \mathbf{m}) \\
&\quad + \frac{r_4}{2} \int (q - \nabla \cdot \mathbf{n})^2 + \int \lambda_4 (q - \nabla \cdot \mathbf{n}) + \delta_{\mathcal{R}}(\mathbf{m}), \tag{3.2a}
\end{aligned}$$

$$\begin{aligned}
& \mathcal{L}(\phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, c_1, c_2, \lambda_1, \lambda_2, \lambda_3) \\
&= \int (f - c_1)^2 H_\epsilon(\phi) + (f - c_2)^2 (1 - H_\epsilon(\phi)) + \int (a + b(\nabla \cdot \mathbf{n})^2) \delta_\epsilon(\phi) |\mathbf{p}| \\
&\quad + \frac{r_1}{2} \int |\mathbf{p} - \nabla \phi|^2 + \int \lambda_1 \cdot (\mathbf{p} - \nabla \phi) + \frac{r_2}{2} \int \left| |\mathbf{p}| \mathbf{n} - \mathbf{p} \right|^2 \\
&\quad + \int \lambda_2 \cdot (|\mathbf{p}| \mathbf{n} - \mathbf{p}) + \frac{r_3}{2} \int |\mathbf{p} - \mathbf{m}|^2 + \int \lambda_3 \cdot (\mathbf{p} - \mathbf{m}) + \delta_{\mathcal{R}}(\mathbf{m}), \tag{3.2b}
\end{aligned}$$

where

$$\mathcal{R} = \{\mathbf{x} \in \mathbb{R}^2 : |\mathbf{x}| = 1\}, \quad \delta_{\mathcal{R}}(\mathbf{m}) = \begin{cases} 0, & \mathbf{m} \in \mathcal{R}, \\ +\infty, & \mathbf{m} \notin \mathcal{R}. \end{cases}$$

Lagrangians λ_i are introduced to enforce the constrains.

Every term in the dual functionals above has derivatives at most of the first order. It significantly reduces the complexity of solving the subproblems which will be presented in the next subsection. But this does come at the cost of increasing the number of parameters. Notice that $ECV-L^1$ model involves with 6 parameters (including a and b for Euler's elastica) while $ECV-L^2$ model has 5 parameters. The fact that the $ECV-L^1$ model has one more parameter is from the non-differentiation of the $ECV-L^1$ model.

Finding the minimizer of the original optimization problem for $ECV-L^1$ and $ECV-L^2$ can be solved by finding saddle points of the augmented Lagrangian functionals. We use the following two algorithms to find the saddle points for these two Lagrangian functionals:

In both algorithms, we simply initialize λ 's to be 0. We initialize the signed distance function ϕ by the contours from the Chan-Vese model. If we choose $b = 0$ in the $ECV-L^1$ and $ECV-L^2$ models, then it reduces to the Chan-Vese model which can be turned into a convex minimization problems as shown in [9, 32] and can be solved by fast binary relaxation methods with Augmented Lagrangian method as in [8, 30]. Also we let $\mathbf{p}, \mathbf{n}, \mathbf{m}$ initially be 0. This helps the final contours to not be trapped near the initial contour.

Algorithm 3.1 ALM Algorithm for L^1 Model.

1. Initialize $\phi^0, \mathbf{p}^0, \mathbf{n}^0, \mathbf{m}^0, q^0, \lambda_1^0, \lambda_2^0, \lambda_3^0, \lambda_4^0$, and calculate c_1^0, c_2^0 .
2. For $k \geq 1$, find $\phi^k, \mathbf{p}^k, \mathbf{n}^k, \mathbf{m}^k, q^k$, such that

$$(\phi^k, \mathbf{p}^k, \mathbf{n}^k, \mathbf{m}^k, q^k) = \operatorname{argmin} \mathcal{L} (c_1^{k-1}, c_2^{k-1}, \phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, q, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}).$$

3. Update the λ 's:

$$\begin{aligned} \lambda_1^k &= \lambda_1^{k-1} + r_1(\mathbf{p} - \nabla \phi), \\ \lambda_2^k &= \lambda_2^{k-1} + r_2(|\mathbf{p}| \mathbf{n} - \mathbf{p}), \\ \lambda_3^k &= \lambda_3^{k-1} + r_3(\mathbf{p} - \mathbf{m}), \\ \lambda_4^k &= \lambda_4^{k-1} + r_4(q - \nabla \cdot \mathbf{n}). \end{aligned}$$

4. Update c_1^k, c_2^k .

Algorithm 3.2 ALM Algorithm for L^2 Model.

1. Initialize $c_1^0, c_2^0, \phi^0, \mathbf{p}^0, \mathbf{m}^0, \mathbf{n}^0, \lambda_1^0, \lambda_2^0, \lambda_3^0$, and calculate c_1^0, c_2^0 .
2. For $k \geq 1$, find $c_1^k, c_2^k, \phi^k, \mathbf{p}^k, \mathbf{n}^k, \mathbf{m}^k$, such that

$$(\phi^k, \mathbf{p}^k, \mathbf{n}^k, \mathbf{m}^k) = \operatorname{argmin} \mathcal{L} (c_1^{k-1}, c_2^{k-1}, \phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}).$$

3. Update the λ 's:

$$\begin{aligned} \lambda_1^k &= \lambda_1^{k-1} + r_1(\mathbf{p} - \nabla \phi), \\ \lambda_2^k &= \lambda_2^{k-1} + r_2(|\mathbf{p}| \mathbf{n} - \mathbf{p}), \\ \lambda_3^k &= \lambda_3^{k-1} + r_3(\mathbf{p} - \mathbf{m}). \end{aligned}$$

4. Update c_1^k, c_2^k .

3.2. Minimization with the subproblems

It is difficult to solve for the optimal minimizer $(\phi^k, \mathbf{p}^k, \mathbf{n}^k, \mathbf{m}^k, q^k)$ of the functional $L(\phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, q, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1})$ for the $ECV-L^1$ model and $(\phi^k, \mathbf{p}^k, \mathbf{m}^k, \mathbf{n}^k)$ of the functional $L(\phi, \mathbf{p}, \mathbf{n}, \mathbf{m}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1})$ for the $ECV-L^2$ model. Instead we alternately search the minimizers of sub-problems as follows to find approximate minimizers.

Throughout our experiments, to simplify the process, we let $L = 1$.

In order to realize the alternating algorithm above, it amounts to solving the following subproblems alternately and iteratively. As stated in the alternating algorithm above, we find optimal solutions of the subproblems with respect to one variable by fixing the other variables. We use ε_1 through ε_7 to denote the minimization energy functionals of the subproblems for the $ECV-L^1$ model, and $\bar{\varepsilon}_1$ through $\bar{\varepsilon}_6$ to denote the minimization energy

Algorithm 3.3 Alternating Algorithm for L^1 Model.

1. At the k th step, let $\tilde{\phi}^0 = \phi^{k-1}, \tilde{\mathbf{p}}^0 = \mathbf{p}^{k-1}, \tilde{\mathbf{n}}^0 = \mathbf{n}^{k-1}, \tilde{\mathbf{m}}^0 = \mathbf{m}^{k-1}, \tilde{q}^0 = q^{k-1}$.
2. for $l = 1, 2, \dots, L$,

$$\begin{aligned}\tilde{\phi}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}, \tilde{\mathbf{p}}^{l-1}, \tilde{\mathbf{n}}^{l-1}, \tilde{\mathbf{m}}^{l-1}, \tilde{q}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}), \\ \tilde{\mathbf{p}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}, \tilde{\mathbf{n}}^{l-1}, \tilde{\mathbf{m}}^{l-1}, \tilde{q}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}), \\ \tilde{\mathbf{n}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}^l, \tilde{\mathbf{n}}, \tilde{\mathbf{m}}^{l-1}, \tilde{q}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}), \\ \tilde{\mathbf{m}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}^l, \tilde{\mathbf{n}}^l, \tilde{\mathbf{m}}, \tilde{q}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}), \\ \tilde{q}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}^l, \tilde{\mathbf{n}}^l, \tilde{\mathbf{m}}^l, \tilde{q}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}, \lambda_4^{k-1}),\end{aligned}$$

3. $\phi^k = \tilde{\phi}^L, \mathbf{p}^k = \tilde{\mathbf{p}}^L, \mathbf{n}^k = \tilde{\mathbf{n}}^L, \mathbf{m}^k = \tilde{\mathbf{m}}^L, q^k = \tilde{q}^L$.

Algorithm 3.4 Alternating Algorithm for L^2 Model.

1. At the k th step, let $\tilde{\phi}^0 = \phi^{k-1}, \tilde{\mathbf{p}}^0 = \mathbf{p}^{k-1}, \tilde{\mathbf{n}}^0 = \mathbf{n}^{k-1}, \tilde{\mathbf{m}}^0 = \mathbf{m}^{k-1}$.
2. for $l = 1, 2, \dots, L$,

$$\begin{aligned}\tilde{\phi}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}, \tilde{\mathbf{p}}^{l-1}, \tilde{\mathbf{n}}^{l-1}, \tilde{\mathbf{m}}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}), \\ \tilde{\mathbf{p}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}, \tilde{\mathbf{n}}^{l-1}, \tilde{\mathbf{m}}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}), \\ \tilde{\mathbf{n}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}^l, \tilde{\mathbf{n}}, \tilde{\mathbf{m}}^{l-1}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}), \\ \tilde{\mathbf{m}}^l &= \operatorname{argmin} \mathcal{L}(\tilde{\phi}^l, \tilde{\mathbf{p}}^l, \tilde{\mathbf{n}}^l, \tilde{\mathbf{m}}, \lambda_1^{k-1}, \lambda_2^{k-1}, \lambda_3^{k-1}).\end{aligned}$$

3. $\phi^k = \tilde{\phi}^L, \mathbf{p}^k = \tilde{\mathbf{p}}^L, \mathbf{n}^k = \tilde{\mathbf{n}}^L, \mathbf{m}^k = \tilde{\mathbf{m}}^L$.

functionals of subproblems for the $ECV-L^2$ model.

The minimization energy functionals for the subproblems of the $ECV-L^1$ model are:

$$\begin{aligned}\varepsilon_1(\phi) &= \int \left((f - c_1)^2 - (f - c_2)^2 \right) H_\epsilon(\phi) + (a + b|q|)|\mathbf{p}| \delta_\epsilon(\phi) \\ &\quad + \frac{r_1}{2} |\mathbf{p} - \nabla \phi|^2 + \lambda_1 \cdot (\mathbf{p} - \nabla \phi), \\ \varepsilon_2(\mathbf{p}) &= \int (a + b|q|) \delta_\epsilon(\phi) |\mathbf{p}| + \frac{r_1}{2} \int |\mathbf{p} - \nabla \phi|^2 + \int \lambda_1 \cdot (\mathbf{p} - \nabla \phi) \\ &\quad + \frac{r_2}{2} \int \|\mathbf{p} \mathbf{n} - \mathbf{p}\|^2 + \int \lambda_2 \cdot (|\mathbf{p} \mathbf{n} - \mathbf{p}|) + \frac{r_3}{2} \int |\mathbf{p} - \mathbf{m}|^2 + \int \lambda_3 (\mathbf{p} - \mathbf{m}), \\ \varepsilon_3(\mathbf{n}) &= \int \frac{r_4}{2} (q - \nabla \cdot \mathbf{n})^2 + \lambda_4 (q - \nabla \cdot \mathbf{n}) + \frac{r_2}{2} \|\mathbf{p} \mathbf{n} - \mathbf{p}\|^2 + |\mathbf{p}| \lambda_2 \cdot \mathbf{n}, \\ \varepsilon_4(q) &= \int (a + b|q|) \delta_\epsilon(\phi) |\mathbf{p}| + \frac{r_4}{2} (q - \nabla \cdot \mathbf{n})^2 + \lambda_4 (q - \nabla \cdot \mathbf{n}),\end{aligned}$$

$$\varepsilon_5(\mathbf{m}) = \int \frac{r_3}{2} |\mathbf{p} - \mathbf{m}|^2 + \boldsymbol{\lambda}_3 \cdot (\mathbf{p} - \mathbf{m}) + \delta_{\mathcal{B}}(\mathbf{m}),$$

$$\varepsilon_6(c_1) = \int (f - c_1)^2 H_\epsilon(\phi),$$

$$\varepsilon_7(c_2) = \int (f - c_2)^2 (1 - H_\epsilon(\phi)).$$

The minimization energy functionals for the subproblems of the $ECV-L^2$ model are:

$$\begin{aligned} \bar{\varepsilon}_1(\phi) = & \int \left((f - c_1)^2 - (f - c_2)^2 \right) H_\epsilon(\phi) + (a + b(\nabla \cdot \mathbf{n})^2) |\mathbf{p}| \delta_\epsilon(\phi) \\ & + \frac{r_1}{2} |\mathbf{p} - \nabla \phi|^2 + \boldsymbol{\lambda}_1 \cdot (\mathbf{p} - \nabla \phi), \end{aligned}$$

$$\begin{aligned} \bar{\varepsilon}_2(\mathbf{p}) = & \int (a + b(\nabla \cdot \mathbf{n})^2) \delta_\epsilon(\phi) |\mathbf{p}| + \frac{r_1}{2} \int |\mathbf{p} - \nabla \phi|^2 + \int \boldsymbol{\lambda}_1 \cdot (\mathbf{p} - \nabla \phi) \\ & + \frac{r_2}{2} \int ||\mathbf{p}| \mathbf{n} - \mathbf{p}|^2 + \int \boldsymbol{\lambda}_2 \cdot (|\mathbf{p}| \mathbf{n} - \mathbf{p}) + \frac{r_3}{2} \int (\mathbf{p} - \mathbf{m})^2 + \int \boldsymbol{\lambda}_3 \cdot (\mathbf{p} - \mathbf{m}), \end{aligned}$$

$$\bar{\varepsilon}_3(\mathbf{n}) = \int b \delta_\epsilon(\phi) |\mathbf{p}| (\nabla \cdot \mathbf{n})^2 + \frac{r_2}{2} ||\mathbf{p}| \mathbf{n} - \mathbf{p}|^2 + |\mathbf{p}| \boldsymbol{\lambda}_2 \cdot \mathbf{n},$$

$$\bar{\varepsilon}_4(\mathbf{m}) = \int \frac{r_3}{2} |\mathbf{p} - \mathbf{m}|^2 + \boldsymbol{\lambda}_3 \cdot (\mathbf{p} - \mathbf{m}) + \delta_{\mathcal{B}}(\mathbf{m}),$$

$$\bar{\varepsilon}_5(c_1) = \int (f - c_1)^2 H_\epsilon(\phi),$$

$$\bar{\varepsilon}_6(c_2) = \int (f - c_2)^2 (1 - H_\epsilon(\phi)).$$

We want to remark that each of these minimization functionals only depends on one variable. This means that we keep all the other unknown functions fixed and only minimize with this variables for each of the subproblems. In the following, we supply the details on solving these energy functional for each of these subproblems.

3.2.1. Minimizing subproblems for the $ECV-L^1$ Model

We tackle the minimization subproblems ε_1 through ε_7 in this subsection. Minimizing $\varepsilon_1, \varepsilon_3$ can be achieved by solving the corresponding Euler-Lagrange equations. We give closed form formulas for the minimizers of ε_4 , to ε_7 . For ε_2 , we quote a theorem from [4] which categorizes the minimizer into several cases and addresses the minimizer in each circumstance.

The stationary point for $\varepsilon_1, \varepsilon_3$ are given by the Euler-Lagrange Equation as follows:

$$-r_1 \Delta \phi = -\frac{\left((f - c_1)^2 - (f - c_2)^2 \right) \epsilon}{\pi(\epsilon^2 + \phi^2)} + \frac{2\epsilon \phi (a + b|q|) |\mathbf{p}|}{\pi(\epsilon^2 + \phi^2)^2} - r_1 \nabla \cdot \mathbf{p} - \nabla \cdot \boldsymbol{\lambda}_1, \quad (3.3a)$$

$$-\nabla(\nabla \cdot \mathbf{n}) = -\frac{r_2}{r_4} (|\mathbf{p}| \mathbf{n} - \mathbf{p}) |\mathbf{p}| - \frac{\lambda_2}{r_4} |\mathbf{p}| - \nabla q - \frac{\nabla \lambda_4}{r_4}. \quad (3.3b)$$

Details on discretization of these equations and their fast numerical solutions will be supplied later.

The shrinkage formula offers a closed-form solution for the minimizer of ϵ_4 at every $(x, y) \in \Omega$:

$$q(x, y) = \max \left\{ \frac{|\mathbf{v}(x, y)| - w(x, y)}{|\mathbf{v}(x, y)|}, 0 \right\} \frac{\mathbf{v}(x, y)}{r_4}, \quad (3.4)$$

where $\mathbf{v} = r_4 \nabla \cdot \mathbf{n} - \lambda_4$, $w = b \tilde{\delta}(\phi) |\mathbf{p}|$.

Reformulate ϵ_5 as

$$\epsilon_5(\mathbf{m}) = \int_{\Omega} \frac{r_3}{2} |\mathbf{m} - (\mathbf{p} + \frac{\lambda_3}{r_3})|^2 + \delta_{\mathcal{R}}(\mathbf{m}) + C,$$

where C is a constant that is independent of \mathbf{m} and

$$\delta_{\mathcal{R}}(\mathbf{m}) = \begin{cases} 0, & |\mathbf{m}| = 1, \\ +\infty, & |\mathbf{m}| \neq 1. \end{cases}$$

By some elementary calculations, one can show that the minimizer \mathbf{m}^* of the integral of the form $\int_{\Omega} |\mathbf{m} - \mathbf{m}_0|^2 + \delta(\mathbf{m})$ is $\mathbf{m}^* = \frac{\mathbf{m}_0}{|\mathbf{m}_0|}$. Thus, the minimizer \mathbf{m}^* of $\epsilon_5(\mathbf{m})$ is

$$\mathbf{m}^* = \frac{r_3 \mathbf{p} + \lambda_3}{|r_3 \mathbf{p} + \lambda_3|}.$$

The minimizers of ϵ_6 and ϵ_7 are given by

$$c_1 = \frac{\int f H_{\epsilon}(\phi)}{\int H_{\epsilon}(\phi)}, \quad c_2 = \frac{\int f (H_{\epsilon}(\phi))}{\int 1 - H_{\epsilon}(\phi)}. \quad (3.5)$$

As for minimizing ϵ_2 , due to the non-differentiable term of \mathbf{p} , the Euler-Lagrange equation can only be given for its regularized version. So, instead of solving the Euler-Lagrange equation, we would like to find its minimizer directly. For this, we rewrite ϵ_2 as in the following form:

$$\epsilon_2 = \int \lambda |\mathbf{p}| + \frac{\mu}{2} |\mathbf{p} - \mathbf{a}|^2 + (\mathbf{v} \cdot \mathbf{p}) |\mathbf{p}|, \quad (3.6)$$

where

$$\begin{aligned} \lambda &= (a + b|q|) \delta(\phi) + \lambda_2 \cdot \mathbf{n}, & \mu &= r_1 + r_2 |\mathbf{n}|^2 + r_3, \\ \mathbf{a} &= \frac{r_1 \nabla \phi - \lambda_1 + \lambda_2 - \lambda_3 + r_3 \mathbf{m}}{r_1 + r_2 |\mathbf{n}|^2 + r_3}, & \mathbf{v} &= -r_2 \mathbf{n}. \end{aligned}$$

Let $g(\mathbf{p}) = \lambda |\mathbf{p}| + \frac{\mu}{2} |\mathbf{p} - \mathbf{a}|^2 + (\mathbf{v} \cdot \mathbf{p}) |\mathbf{p}|$. If at each point $(x, y) \in \Omega$, $\mathbf{p}(x, y)$ minimizes the functional g , then \mathbf{p} minimizes the the integral $\epsilon_2(\mathbf{p})$. We refer readers to Theorem 2 in [4] for the details of finding the minimizer of $g(\mathbf{p})$ at each point (x, y) .

3.2.2. Minimizing subproblems of the $ECV-L^2$ Model

The subproblems of the $ECV-L^2$ model are similar to subproblems of the $ECV-L^1$ model. The stationary point for $\bar{\varepsilon}_1, \bar{\varepsilon}_3$ are given by the Euler-Lagrange equation as follows:

$$\begin{aligned} -r_1 \Delta \phi = & -\frac{((f - c_1)^2 - (f - c_2)^2) \epsilon}{\pi(\epsilon^2 + \phi^2)} + \frac{2\epsilon \phi (a + b(\nabla \cdot \mathbf{n})^2) |\mathbf{p}|}{\pi(\epsilon^2 + \phi^2)^2} \\ & - r_1 \nabla \cdot \mathbf{p} - \nabla \cdot \boldsymbol{\lambda}_1, \end{aligned} \quad (3.7a)$$

$$-\nabla (2b\delta(\phi)|\mathbf{p}|(\nabla \cdot \mathbf{n})) = -|\mathbf{p}|\boldsymbol{\lambda}_2 - r_2 (|\mathbf{p}|^2 \mathbf{n} - |\mathbf{p}|\mathbf{p}). \quad (3.7b)$$

Details on discretization of these equations and their fast numerical solutions will be supplied later.

The minimization for subproblem $\bar{\varepsilon}_4$ turns out to be exactly the same as ε_5 in the $ECV-L^1$ model. So the minimizer is

$$\mathbf{m}^* = \frac{r_3 \mathbf{p} + \boldsymbol{\lambda}_3}{|r_3 \mathbf{p} + \boldsymbol{\lambda}_3|}.$$

The minimizer for c_1, c_2 in subproblem $\bar{\varepsilon}_5$ and $\bar{\varepsilon}_6$ are given in (3.5) as in Section 3.2.1. The minimization functional for subproblem $\bar{\varepsilon}_2$ can be rewritten as

$$\bar{\varepsilon}_2 = \int \lambda |\mathbf{p}| + \frac{\mu}{2} |\mathbf{p} - \mathbf{a}|^2 + (\mathbf{v} \cdot \mathbf{p}) |\mathbf{p}|,$$

where

$$\begin{aligned} \lambda &= (a + b(\nabla \cdot \mathbf{n})^2) \delta_\epsilon(\phi) + \boldsymbol{\lambda}_2 \cdot \mathbf{n}, & \mu &= r_1 + r_2 |\mathbf{n}|^2 + r_3, \\ \mathbf{a} &= \frac{r_1 \nabla \phi - \boldsymbol{\lambda}_1 + \boldsymbol{\lambda}_2 - \boldsymbol{\lambda}_3 + r_3 \mathbf{m}}{r_1 + r_2 |\mathbf{n}|^2 + r_3}, & \mathbf{v} &= -r_2 \mathbf{n}. \end{aligned}$$

Then by applying Theorem 2 in [4], the minimizer \mathbf{p} can be obtained.

3.3. Initialization of ϕ

Since the ECV models are non-convex, different initial condition of ϕ could lead to different segmentation results. To be fair for the comparison, we take the same initial condition for both segmentation models. Specifically, we first apply the CV model to the test images and then use the resulting contour to construct a signed distance function, which will be the common initial ϕ for the two models. For this, we reformulate the CV model using a binary function as in [20] and then relaxed it as in [9, 32]. More precisely, we let $u = H(\phi)$ and $\mathbf{p} = \nabla \mathbf{u}$ and solve the minimizer of the following problem.

$$\min_{u \in [0,1], c_1, c_2} \int_{\Omega} (f - c_1)^2 u + (f - c_2)^2 (1 - u) + \mu |\mathbf{p}|, \quad s.t. \mathbf{p} = \nabla u. \quad (3.8)$$

In this way, the CV model given above can be easily minimized by using ALM as in [8, 30]. In our experiment, we use the same codes as were used in [4, 35] and set $b = 0$. We do 20 iterations to get the initial segmentation contour, from which the initial ϕ can be constructed by letting $\phi = u - 0.5$. Based on this initial segmentation function ϕ , we can calculate the initial values of c_1 and c_2 as follows:

$$c_1 = \frac{\int fH(\phi)}{\int H(\phi)}, \quad c_2 = \frac{\int f(1-H(\phi))}{\int 1-H(\phi)}.$$

4. Numerical implementation

In this section, we discuss the detail of numerical implementation of the models. Suppose $u : \Omega \rightarrow \mathbb{R}$ is a given image. The image domain Ω is discretized as a $M \times N$ grid. Denote the forward partial derivatives in x and y direction of u as $\partial_1^+ u$, $\partial_2^+ u$, and the backward partial derivatives in x and y direction of u as $\partial_1^- u$, $\partial_2^- u$, respectively. Imposing periodic boundary conditions, the partial derivatives are defined as:

$$\begin{aligned} \partial_1^+ u(i, j) &= \begin{cases} u(i+1, j) - u(i, j), & 1 \leq i \leq M-1, \\ u(1, j) - u(M, j), & i = M, \end{cases} \\ \partial_2^+ u(i, j) &= \begin{cases} u(i, j+1) - u(i, j), & 1 \leq j \leq N-1, \\ u(i, 1) - u(i, N), & j = N, \end{cases} \\ \partial_1^- u(i, j) &= \begin{cases} u(i, j) - u(i-1, j), & 2 \leq i \leq M, \\ u(1, j) - u(M, j), & i = 1, \end{cases} \\ \partial_2^- u(i, j) &= \begin{cases} u(i, j) - u(i, j-1), & 2 \leq j \leq N, \\ u(i, 1) - u(i, N), & j = 1. \end{cases} \end{aligned}$$

The discretization of the gradient and divergence follows naturally:

$$\begin{aligned} \nabla^\pm u(i, j) &= (\partial_1^\pm u(i, j), \partial_2^\pm u(i, j)), \\ \nabla^\pm \cdot \mathbf{v}(i, j) &= \partial_1^\pm v_1(i, j) + \partial_2^\pm v_2(i, j), \end{aligned}$$

where vector $\mathbf{v} = (v_1, v_2)$.

First of all, we discuss the solutions of Eq. (3.3a) and Eq. (3.7a) since these two equations are very similar. Because we impose periodic boundary conditions, we can use fast Fourier transformation (FFT) to solve the equations iteratively.

In order to avoid singularity, we add a small number δ (we let $\delta = 0.01$). Then discretize Eq. (3.3a) and Eq. (3.7a) as:

$$\begin{aligned} & -r_1 \nabla^- \cdot (\nabla^+ \phi^{(s+1)}) + \delta \\ = & \delta - \frac{((f - c_1)^2 - (f - c_2)^2) \epsilon}{\pi (\epsilon^2 + (\phi^{(s)})^2)} + \frac{2\epsilon \phi^{(s)} (a + b|q|) |\mathbf{p}|}{\pi (\epsilon^2 + (\phi^{(s)})^2)^2}, \end{aligned} \quad (4.1a)$$

$$\begin{aligned}
 & -r_1 \nabla^- \cdot (\nabla^+ \phi^{(s+1)}) + \delta \\
 = & \delta - \frac{\left((f - c_1)^2 - (f - c_2)^2 \right) \epsilon}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)} + \frac{2\epsilon \phi^{(s)} \left(a + b(\nabla^- \cdot \mathbf{n})^2 \right) |\mathbf{p}|}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)^2} \\
 & - r_1 \nabla^- \cdot \mathbf{p} - \nabla^- \cdot \boldsymbol{\lambda}_1,
 \end{aligned} \tag{4.1b}$$

where $\phi^{(s=0)}$ is initialized as the solution in the previous loop. Both Eq. (4.1a) and Eq. (4.1b) can be simplified as

$$-r_1 \nabla^- \cdot (\nabla^+ \phi^{(s+1)}) + \delta = g(\phi^{(s)}),$$

where in Eq. (4.1a)

$$g(\phi^{(s)}) = \delta - \frac{\left((f - c_1)^2 - (f - c_2)^2 \right) \epsilon}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)} + \frac{2\epsilon \phi^{(s)} \left(a + b|q| \right) |\mathbf{p}|}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)^2} - r_1 \nabla^- \cdot \mathbf{p} - \nabla^- \cdot \boldsymbol{\lambda}_1$$

in Eq. (4.1b)

$$g(\phi^{(s)}) = \delta - \frac{\left((f - c_1)^2 - (f - c_2)^2 \right) \epsilon}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)} + \frac{2\epsilon \phi^{(s)} \left(a + b(\nabla^- \cdot \mathbf{n})^2 \right) |\mathbf{p}|}{\pi \left(\epsilon^2 + (\phi^{(s)})^2 \right)^2}.$$

Apply FFT on both sides, then

$$\left(-r_1 \mathcal{F}(\nabla^- \cdot \nabla^+) + \delta \right) \mathcal{F}(\phi^{(s+1)}) = \mathcal{F} \left(g(\phi^{(s)}) \right). \tag{4.2}$$

Notice that

$$\begin{aligned}
 \mathcal{F} \left(\partial_1^\pm \phi(i, j) \right) &= \pm \left(e^{\pm \sqrt{-1} z_i^1} - 1 \right) \mathcal{F} \left(\phi(i, j) \right), \\
 \mathcal{F} \left(\partial_2^\pm \phi(i, j) \right) &= \pm \left(e^{\pm \sqrt{-1} z_j^2} - 1 \right) \mathcal{F} \left(\phi(i, j) \right),
 \end{aligned}$$

where $z_i^1 = 2\pi(i - 1)/M$, $i = 1, \dots, M$, $z_j^2 = 2\pi(j - 1)/N$, $i = 1, \dots, N$.

Also notice that

$$\nabla^- \cdot \left(\nabla^+ \phi(i, j) \right) = \partial_1^+ \phi(i, j) - \partial_1^- \phi(i, j) + \partial_2^+ \phi(i, j) - \partial_2^- \phi(i, j).$$

Therefore Eq. (4.2) can be simplified to

$$\left(-2r_1 (\cos(z_i^1) + \cos(z_j^2)) - 2 \right) + \delta \mathcal{F} \left(\phi^{(s+1)}(i, j) \right) = \mathcal{F} \left(g(\phi^{(s)}(i, j)) \right).$$

Applying the inverse Fourier transformation gives $\phi^{(s+1)}$.

Now we discuss the solutions of \mathbf{n} in both the $ECV-L^1$ and L^2 models. We use frozen coefficient method to solve for \mathbf{n} . The way of solving Eq. (3.3b) and Eq. (3.7b) are similar but with slight differences. The discretization of Eq. (3.3b) and Eq. (3.7b) leads to:

$$-r_4 \nabla^+ (\nabla^- \cdot \mathbf{n}) = -r_2 \left(|\mathbf{p}| \mathbf{n} - \mathbf{p} \right) |\mathbf{p}| - \boldsymbol{\lambda}_2 |\mathbf{p}| - \nabla^+ q - \nabla^+ \boldsymbol{\lambda}_4,$$

$$-\nabla^+ \left(2b\delta_\epsilon(\phi)|\mathbf{p}|(\nabla^- \cdot \mathbf{n}) \right) = -|\mathbf{p}|\lambda_2 - r_2 \left(|\mathbf{p}|^2 \mathbf{n} - |\mathbf{p}|\mathbf{p} \right).$$

Let $D = \max_{(i,j)}(r_2|\mathbf{p}|^2)$ in both Eq. (3.3b) and Eq. (3.7b), and in Eq. (3.7b) let $c = \max_{(i,j)}(2b\delta_\epsilon(\phi)|\mathbf{p}|)$. Then the Eq. (3.3b) and Eq. (3.7b) can be solved iteratively as following respectively:

$$\begin{aligned} & -r_4\nabla^+ \left(\nabla^- \cdot \mathbf{n}^{(s+1)} \right) + D\mathbf{n}^{(s+1)} \\ = & D\mathbf{n}^{(s)} - r_2|\mathbf{p}|^2\mathbf{n}^{(s)} + r_2|\mathbf{p}|\mathbf{p} - \lambda_2|\mathbf{p}| - \nabla^+q - \nabla^+\lambda_4, \end{aligned} \quad (4.3a)$$

$$\begin{aligned} & -c\nabla^+ \left(\nabla^- \cdot \mathbf{n}^{(s+1)} \right) + D\mathbf{n}^{(s+1)} \\ = & D\mathbf{n}^{(s)} - \nabla^+ \left((c - 2b\delta_\epsilon(\phi)|\mathbf{p}|) \left(\nabla^- \cdot \mathbf{n}^{(s)} \right) \right) - |\mathbf{p}|\lambda_2 - r_2 \left(|\mathbf{p}|^2\mathbf{n}^{(s)} - |\mathbf{p}|\mathbf{p} \right). \end{aligned} \quad (4.3b)$$

$\mathbf{n}^{(s=0)}$ is initialized as the solution in the previous loop. For Eq. (4.3a), we have the following denotation:

$$\begin{aligned} \omega &= r_4, \\ \mathbf{h} &= D\mathbf{n}^{(s)} - r_2|\mathbf{p}|^2\mathbf{n}^{(s)} + r_2|\mathbf{p}|\mathbf{p} - \lambda_2|\mathbf{p}| - \nabla^+q - \nabla^+\lambda_4. \end{aligned}$$

For Eq.(4.3b), we have the following denotation:

$$\begin{aligned} \omega &= c, \\ \mathbf{h} &= D\mathbf{n}^{(s)} - |\mathbf{p}|\lambda_2 - r_2(|\mathbf{p}|^2\mathbf{n}^{(s)} - |\mathbf{p}|\mathbf{p}) - \nabla^+ \left((c - 2b\delta_\epsilon(\phi)|\mathbf{p}|)(\nabla^- \cdot \mathbf{n}^{(s)}) \right). \end{aligned}$$

Then each of Eq. (4.3a) and Eq. (4.3b) can be rewritten as a system of equations:

$$-\omega \left(\partial_1^+ \partial_1^- n_1^{(s+1)} + \partial_1^+ \partial_2^- n_2^{(s+1)} \right) + Dn_1^{(s+1)} = h_1, \quad (4.4a)$$

$$-\omega \left(\partial_2^+ \partial_1^- n_1^{(s+1)} + \partial_2^+ \partial_2^- n_2^{(s+1)} \right) + Dn_2^{(s+1)} = h_2, \quad (4.4b)$$

where vector $\mathbf{n}^{(s)} = \langle n_1^{(s)}, n_2^{(s)} \rangle$, $\mathbf{h} = \langle h_1, h_2 \rangle$.

Apply Fourier transformation to each equation of (4.4), and at every pixel (i, j) , we can get a 2 x 2 system of equations:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \mathcal{F}(n_1)(i, j) \\ \mathcal{F}(n_2)(i, j) \end{bmatrix} = \begin{bmatrix} \mathcal{F}(h_1)(i, j) \\ \mathcal{F}(h_2)(i, j) \end{bmatrix},$$

where

$$\begin{aligned} a_{11} &= -\omega(2\cos(z_i^1) - 2) + D, \\ a_{12} &= -\omega(1 - \cos(z_i^1) - \sqrt{-1}\sin(z_i^1))(-1 + \cos(z_j^2) - \sqrt{-1}\sin(z_j^2)), \\ a_{21} &= -\omega(1 - \cos(z_i^1) + \sqrt{-1}\sin(z_i^1))(-1 + \cos(z_j^2) + \sqrt{-1}\sin(z_j^2)), \\ a_{22} &= -\omega(2\cos(z_j^2) - 2) + D \end{aligned}$$

with $z_i^1 = 2\pi(i-1)/M, i = 1, \dots, M, z_j^2 = 2\pi(j-1)/N, j = 1, \dots, N$. Then we can solve for \mathbf{n}^s by means of inverse Fourier transformation.

Now we discretize q in the $ECV-L^1$ model:

$$q(i, j) = \max \left\{ \frac{|v(i, j)| - b\delta_\epsilon(\phi(i, j))|\mathbf{p}(i, j)|}{|v(i, j)|}, 0 \right\} \frac{v(i, j)}{r_4},$$

where

$$v(i, j) = r_4(\partial_1^- n_1(i, j) + \partial_2^- n_2(i, j)) - \lambda_4.$$

Furthermore, if we suppose vectors $\boldsymbol{\lambda}_1 = (\lambda_{11}, \lambda_{12}), \boldsymbol{\lambda}_2 = (\lambda_{21}, \lambda_{22}), \boldsymbol{\lambda}_3 = (\lambda_{31}, \lambda_{32}), \mathbf{p} = (p_1, p_2), \mathbf{n} = (n_1, n_2), \mathbf{m} = (m_1, m_2)$, then we update the Lagrangian multipliers for $ECV-L^1$ model at each (i, j) as follows:

$$\begin{aligned} \lambda_{11}^k &= \lambda_{11}^{k-1} + r_1(p_1 - \partial_1^+ \phi), & \lambda_{12}^k &= \lambda_{12}^{k-1} + r_1(p_2 - \partial_2^+ \phi), \\ \lambda_{21}^k &= \lambda_{21}^{k-1} + r_2(|\mathbf{p}|n_1 - p_1), & \lambda_{22}^k &= \lambda_{22}^{k-1} + r_2(|\mathbf{p}|n_2 - p_2), \\ \lambda_{31}^k &= \lambda_{31}^{k-1} + r_3(p_1 - m_1), & \lambda_{32}^k &= \lambda_{32}^{k-1} + r_3(p_2 - m_2), \\ \lambda_4^k &= \lambda_4^{k-1} + r_4(q - (\partial_1^- n_1 + \partial_2^- n_2)). \end{aligned}$$

We update the Lagrangian multipliers for $ECV-L^2$ model at (i, j) as follows:

$$\begin{aligned} \lambda_{11}^k &= \lambda_{11}^{k-1} + r_1(p_1 - \partial_1^+ \phi), & \lambda_{12}^k &= \lambda_{12}^{k-1} + r_1(p_2 - \partial_2^+ \phi), \\ \lambda_{21}^k &= \lambda_{21}^{k-1} + r_2(|\mathbf{p}|n_1 - p_1), & \lambda_{22}^k &= \lambda_{22}^{k-1} + r_2(|\mathbf{p}|n_2 - p_2), \\ \lambda_{31}^k &= \lambda_{31}^{k-1} + r_3(p_1 - m_1), & \lambda_{32}^k &= \lambda_{32}^{k-1} + r_3(p_2 - m_2). \end{aligned}$$

5. Numerical experiments

In this section, we present the numerical results by applying the $ECV-L^1$ and L^2 models to synthetic and real images. In each experiment, we record the initial condition of the contour ϕ , the final segmentation result, residues, relative error of $\boldsymbol{\lambda}$, relative error of ϕ and the value of the energy function. The last four numerical records which are calculated as in [4, 29, 35], are for the purpose of tracking the convergence of ALM algorithm.

Let's denote the residues of $\boldsymbol{\lambda}$'s of the $ECV-L^1$ model as R_i and the ones of the $ECV-L^2$ model as \bar{R}_i . At the k th iteration, the residues are calculated as follows:

$$\begin{aligned} R_1^k &= \frac{1}{|\Omega|} \|\mathbf{p}^k - \nabla \phi^k\|_{L^1}, & R_2^k &= \frac{1}{|\Omega|} \|\mathbf{p}^k \|\mathbf{n}^k - \mathbf{p}^k\|_{L^1}, \\ R_3^k &= \frac{1}{|\Omega|} \|\mathbf{p}^k - \mathbf{m}^k\|_{L^1}, & R_4^k &= \frac{1}{|\Omega|} \|q^k - \nabla \cdot \mathbf{n}^k\|_{L^1}, \\ \bar{R}_1^k &= \frac{1}{|\Omega|} \|\mathbf{p}^k - \nabla \phi^k\|_{L^1}, & \bar{R}_2^k &= \frac{1}{|\Omega|} \|\mathbf{p}^k \|\mathbf{n}^k - \mathbf{p}^k\|_{L^1}, \end{aligned}$$

$$\bar{R}_3^k = \frac{1}{|\Omega|} |\mathbf{p}^k - \mathbf{m}^k|_{L^1},$$

where $|\cdot|_{L^1}$ denotes the L^1 -norm, and $|\Omega|$ is the area of the domain, with $|\Omega| = M \times N$.

We calculate the relative error of λ 's in both models as

$$L_i^k = \frac{|\lambda_i^k - \lambda_i^{k-1}|_{L^1}}{|\lambda_i^{k-1}|_{L^1}};$$

the relative error of ϕ of both models as

$$\frac{|\phi^k - \phi^{k-1}|_{L^1}}{|\phi^{k-1}|_{L^1}};$$

the energy of the $ECV-L^1$ model as

$$\int_{\Omega} (f - c_1^k)^2 H_{\epsilon}(\phi^k) + (f - c_2^k)^2 (1 - H_{\epsilon}(\phi^k)) + \left(a + b \left| \nabla \cdot \frac{\nabla \phi^k}{|\nabla \phi^k|} \right| \right) \delta_{\epsilon}(\phi^k) |\mathbf{p}^k|;$$

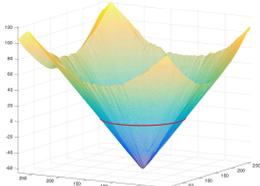
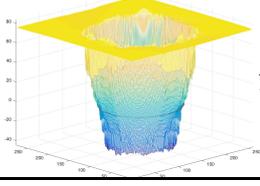
the energy of the $ECV-L^2$ model as

$$\int_{\Omega} (f - c_1^k)^2 H_{\epsilon}(\phi^k) + (f - c_2^k)^2 (1 - H_{\epsilon}(\phi^k)) + \left(a + b \left(\nabla \cdot \frac{\nabla \phi^k}{|\nabla \phi^k|} \right)^2 \right) \delta_{\epsilon}(\phi^k) |\mathbf{p}^k|,$$

where $H_{\epsilon}(\phi)$ and $\delta_{\epsilon}(\phi)$ are the approximation functions of the Heaviside function $H(\phi)$ and its derivative function $\delta(\phi)$.

We first compare the performances of the algorithms proposed in this paper and the one in [4] when they are applied for a simple synthetic image of the size 276×262 with a convex contour, respectively. We only consider the $ECV-L^1$ model since the energy $\int |\kappa| ds$ is equal to 2π for any closed convex contour, which gives us a benchmark for comparison. To make a fair comparison, we fix the iteration of both the algorithms to be 200 and calculate the energy $\int |\kappa| ds$ as well as the average time spent by each of the algorithm. For the algorithm in [4], the procedure of reinitialization [26] is conducted after each iteration. The comparison results are shown in Table 1. From this Table, we can see that a) our algorithm with signed-distance constraint is twice as fast as the algorithm involving reinitialization; b) our algorithm generates ϕ that is close to be a signed-distance function, while the algorithm in [4] fails to do so; c) our algorithm captures the curvature much more accurately than the algorithm in [4]. The following parameters are used for these experiments: for our algorithm, $a = .001, b = 1.5, r_1 = .3, r_2 = .3, r_3 = .3, r_4 = .3$; for the algorithm in [4], $a = .001, b = 1.5, r_1 = .3, r_2 = .3, r_3 = .5$, and moreover, for the reinitialization, we follow the procedure discussed in [26] by solving a Hamilton-Jacobi type equation, and in this comparison, the equation is solved by using an explicit scheme with a step size 0.01 and 30 iterations. Note that from the plot, with the algorithm in [4], more iterations in the reinitialization are needed to get a signed-distance function, which

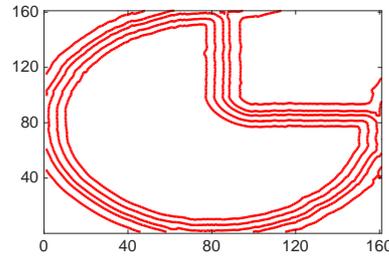
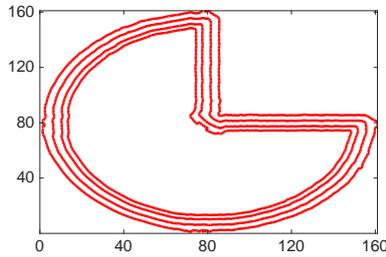
Table 1: The comparison between the proposed algorithm and the one in [4].

	segmentation	function ϕ	energy $\int \kappa ds$	speed(second)
our algorithm			$8.13 \approx 2.58\pi$	15.76
algorithm in [4]			$112.78 \approx 35.92\pi$	35.32



(a) L^1 Segmentation

(b) L^2 Segmentation



(c) level sets using L^1 model

(d) level sets using L^2 model

Figure 2: The results for the $ECV-L^1$ and $ECV-L^2$ models and the level curves around the zero level set of the corresponding segmentation function ϕ .

becomes more expensive numerically. Besides helping represent curvature more faithfully, this comparison also shows that the proposed algorithm is more efficient than the one in [4].

In Fig. 2, we apply the $ECV-L^1$ and $ECV-L^2$ models for a synthetic image of an incomplete disk with a quarter part missing by using the proposed algorithms. This example clearly shows one major difference between the two models: the L^1 model preserves almost

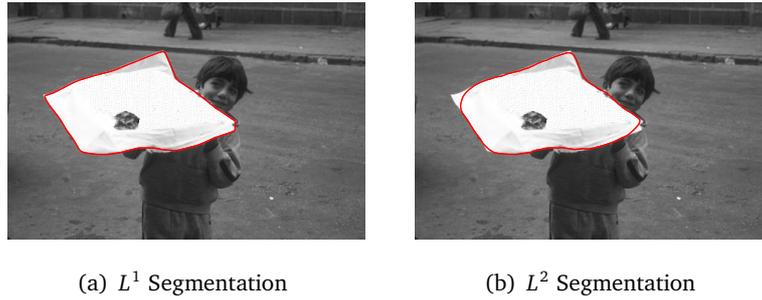


Figure 3: ECV- L^1 model parameters: $a = 0.001, b = 1, r_1 = 0.5, r_2 = 0.4, r_3 = 0.1, r_4 = 8$. ECV- L^2 model parameters: $a = 0.001, b = 1, r_1 = 0.5, r_2 = 0.3, r_3 = 0.5$.

completely the corner near the center, while the L^2 model smears it, which is consistent to the discussion in Section 2. Moreover, to check whether the constraint of keeping $|\mathbf{p}| = 1$ is effective, we present the level curves around the zero level set of the segmentation function ϕ for both cases. The plots show that near the segmentation contour the function ϕ presents equally distributed level curves, which help capture curvatures reliably.

We then test real images for the two models. These numerical examples give us insights on the similarities and discrepancies of their performances.

We begin with an image of a kid holding a tray shown in Fig. 3. The white tray has a sharp contrast compared with the rest of the image. ECV- L^1 and L^2 models both select the tray as the foreground while the rest as the background. However the details differ significantly. The corners of the tray are mostly preserved by ECV- L^1 model, while none of the corners is kept by ECV- L^2 model.

We then perform our algorithms on a brain blood vessel image as shown in Fig. 4. The image is severely contaminated by the noise, and if the CV model is applied with a relatively small length parameter, it will lead to fuzzy segmentation boundaries. To avoid this, we here chose a relatively large length parameter and thus obtained smooth segmentation contours. Due to the low grey intensity contrast, blood vessels cannot be connected by the CV model and several isolated tiny curves present as parts of the segmentation contour. As a contrast, both the ECV- L^1 and L^2 models are able to smooth out the edges and automatically join those broken parts of vessels. This example indicates that both models are capable of connecting boundaries to restore meaningful blood vessels, and also prohibit those tiny closed curves as segmentation contours.

The segmentation results in Fig. 4 for both models look somewhat similar. However there exists some subtle difference. As shown in Fig. 5(a) and Fig. 5(b), we see that corners are preserved by the L^1 model but not L^2 model, which is suggested by our analysis in Section 2. Moreover, in Fig. 5(c) and Fig. 5(d), the segmentation of the L^1 model covers a few more pixels than that of the L^2 model. Notice the tip of the vessel is very narrow. If L^2 model were to cover the tip, it would need an arc of a circle with small radius and lead to a large value of energy. However for the ECV- L^1 model, covering the tip gives the same value for the elastica term, but brings down the fitting term value. Therefore, this

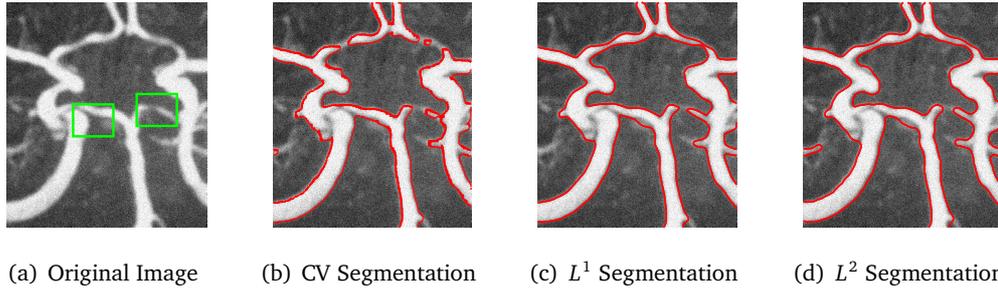


Figure 4: $ECV-L^1$ model parameters: $a = 0.001, b = 0.8, r_1 = 0.1, r_2 = 0.3, r_3 = 0.1, r_4 = 0.5$. $ECV-L^2$ model with parameters $a = 0.001, b = 5, r_1 = 0.1, r_2 = 0.3, r_3 = 1.2$.

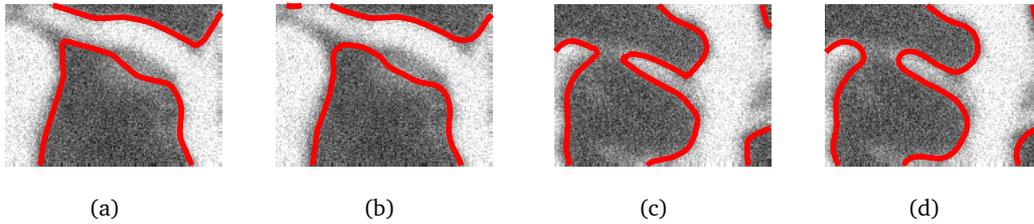


Figure 5: Zoomed in segmentations: (a)(c) are the results from the L^1 model, (b)(d) are the results from the L^2 model.

discrepancy demonstrates that the L^1 model is more suitable for keeping sharp corners of contour than the L^2 model.

To check whether the iteration of the proposed algorithms for the two models converges to saddle points of their augmented Lagrangian functionals, in Fig. 6, we list the plots of the relative residuals, relative errors of the Lagrange multipliers, relative error of ϕ^k , and the energy versus iteration for this example. These plots show that there is a tendency that saddle points can be approached.

In Fig. 7, we perform the algorithms on an image of a kid wearing a hat. In the sense of size the main object in this image is the hat. But the hat is partially occluded by the kid. With only the length regularity, the CV model separates the hat and several small objects from the rest of the background, and in the mean time the shape of the hat is undermined from the occlusion. Our results from the $ECV-L^1$ and L^2 models give up the smaller objects and treat them as the background. Both models offer “guesses” of the occluded part of the hat. The result by the L^1 model is closer to the disocclusion of human perception, and it has considerably less wavy or less inward at the missing part of the hat when compared with the result obtained by the L^2 model. The geometric property of the integration $\int |\kappa| ds$ determines that the L^1 model prefers convex contour [4, 14, 21], and the ultimate segmentation is close to being convex for a large enough curvature parameter b . We may understand the curvy disocclusion of the L^2 model under a slight different setting. If we clamp the two ends of a curve with free length and fixed tangent directions at the end points, then the curve that minimizes the energy $\int \kappa^2 ds$ is part of the so called

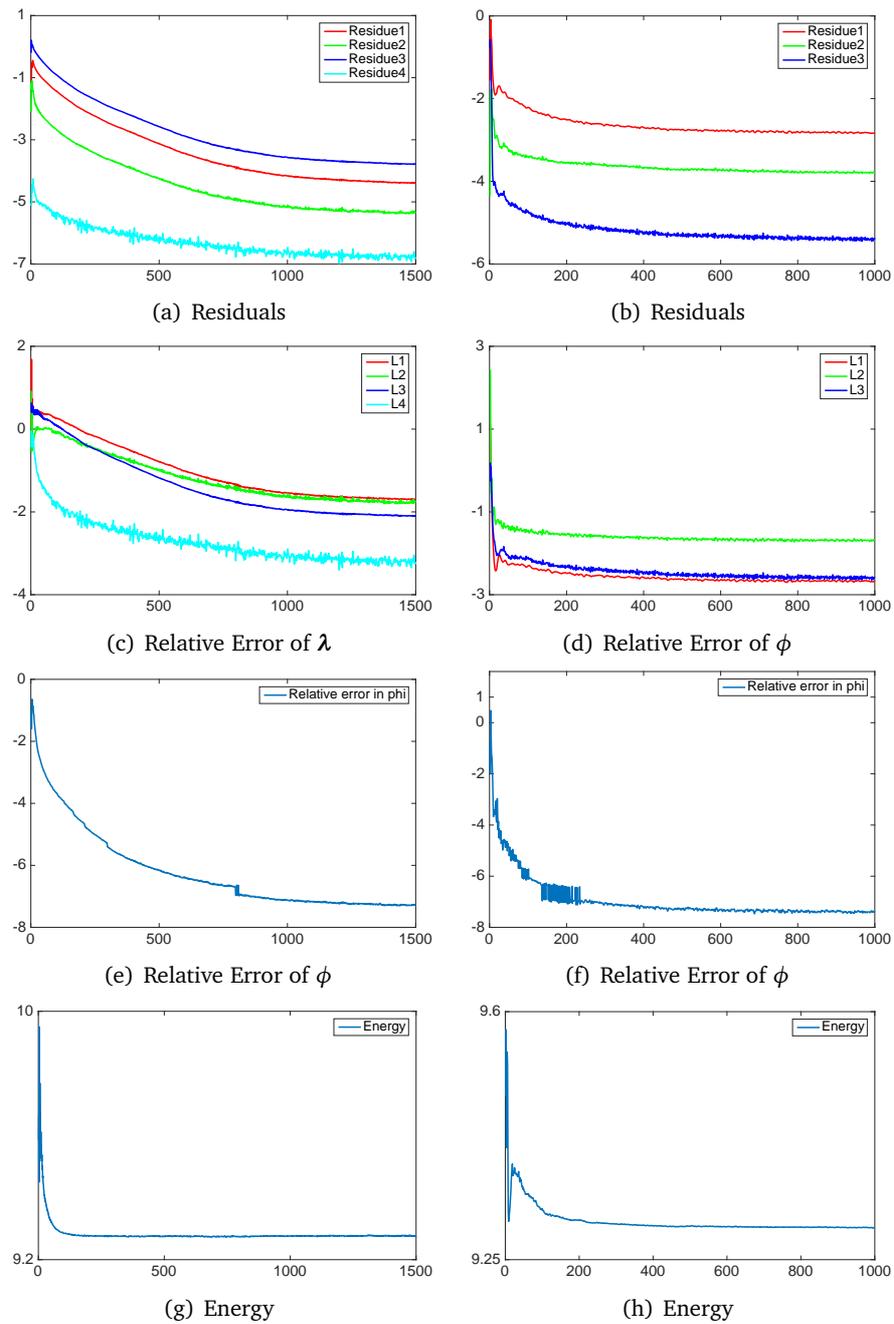


Figure 6: The plots of residuals, relative errors of λ and ϕ , as well as the energy for the blood vessel example by applying the $ECV-L^1$ model (left column) and the $ECV-L^2$ (right column). The vertical axes are all in the base-e logarithmic scales.

rectangular elastica [5, 18].

In our final example as shown in Fig. 8, the models are applied to the image with a

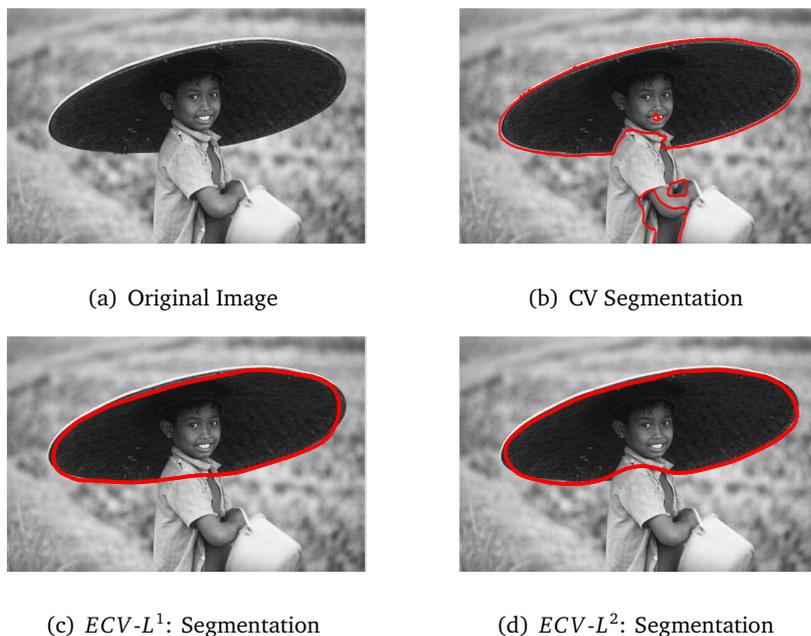


Figure 7: $ECV-L^1$ model parameters: $a = 0.001, b = 90, r_1 = 3, r_2 = 43, r_3 = 9, r_4 = 24$. $ECV-L^2$ model parameters: $a = 0.001, b = 1000, r_1 = 5, r_2 = 1, r_3 = 1$.

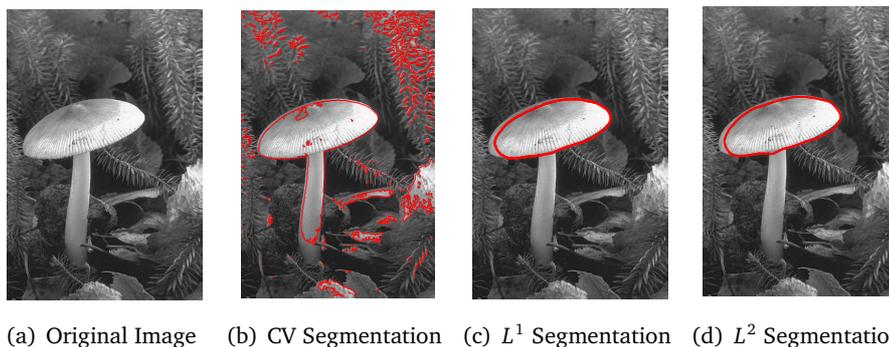


Figure 8: $ECV-L^1$ parameters: $a = 0.001, b = 40, r_1 = 1, r_2 = 8, r_3 = 1, r_4 = 2$. $ECV-L^2$ parameters: $a = 0.001, b = 40, r_1 = 0.2, r_2 = 6.5, r_3 = 3$.

standing mushroom. With a relatively large curvature parameter b , both models extract the mushroom cap from the mushroom stem and the background. Notice that the stem is relatively slim and a sharp turn presents along the contour if the stem is preserved. These two facts lead to a large energy for both models, and therefore only the cap is kept in order to achieve a low energy. However, by carefully studying the two resulting contours, one can see that the one by the L^1 model is more closer to a convex contour than that of the L^2 model, which shows that the L^1 model is more sensitive to the convexity of contour. To see how the curvature term affects the smoothness and convexity of segmentation contours,

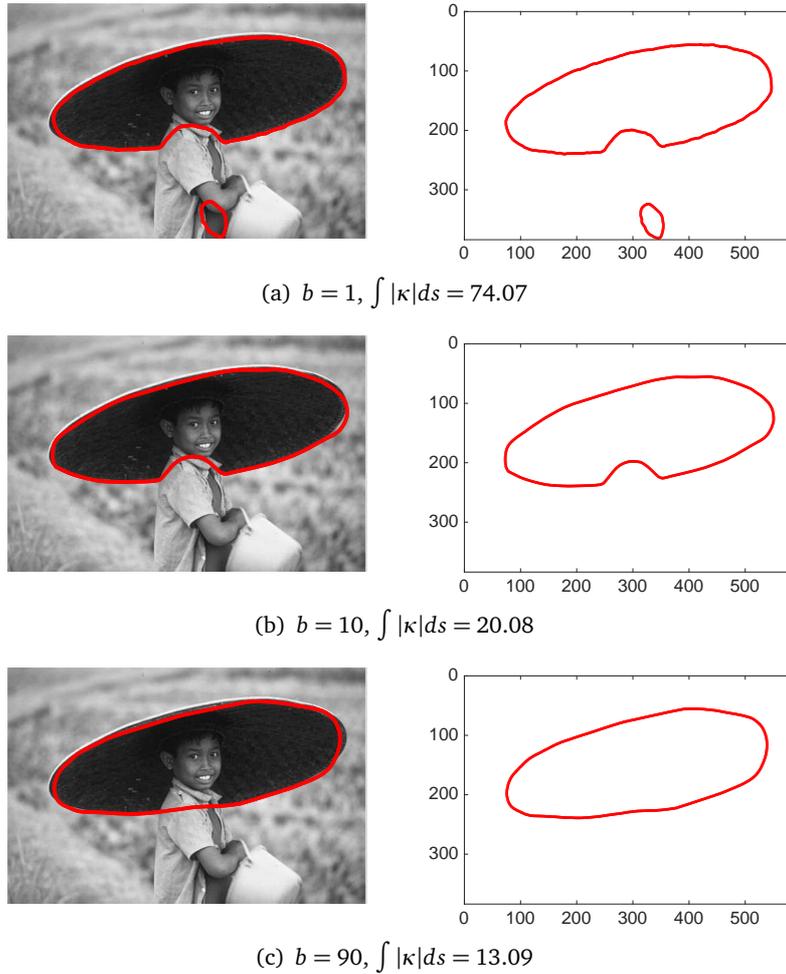


Figure 9: The left column presents segmentations using the $ECV-L^1$ model for the “hat” image with different values of b , while the right one lists the corresponding separate zero-contours. Note that the integral $\int |\kappa| ds$ is calculated along the contour of “hat”.

we present numerical experiments for the “hat” image with different values of b for both models. We also measure the curvature term $\int |\kappa| ds$ for the $ECV-L^1$ model by using the resulting ϕ . As shown in Fig. 9, we set the curvature coefficient $b = 1, 10, 90$ for the $ECV-L^1$ model, and the corresponding curvature term equals 74.07, 20.08, 13.09, respectively. These results justify that once the parameter b is larger, the obtained segmentation contour becomes more closer to be a convex closed curve and the quantity $\int |\kappa| ds$ also drops. Similarly, in Fig. 10, when the $ECV-L^2$ model is applied, the larger the parameter b is, the more the notch near the boy’s neck will be restored. However, even with a relatively large parameter $b = 1000$, the segmentation still presents concave part near the neck, which demonstrates the major features of the two models. In fact, we may see more if we zoom in these contours. As shown in Fig. 11, the same part of Fig. 9(a) and Fig. 9(b) is

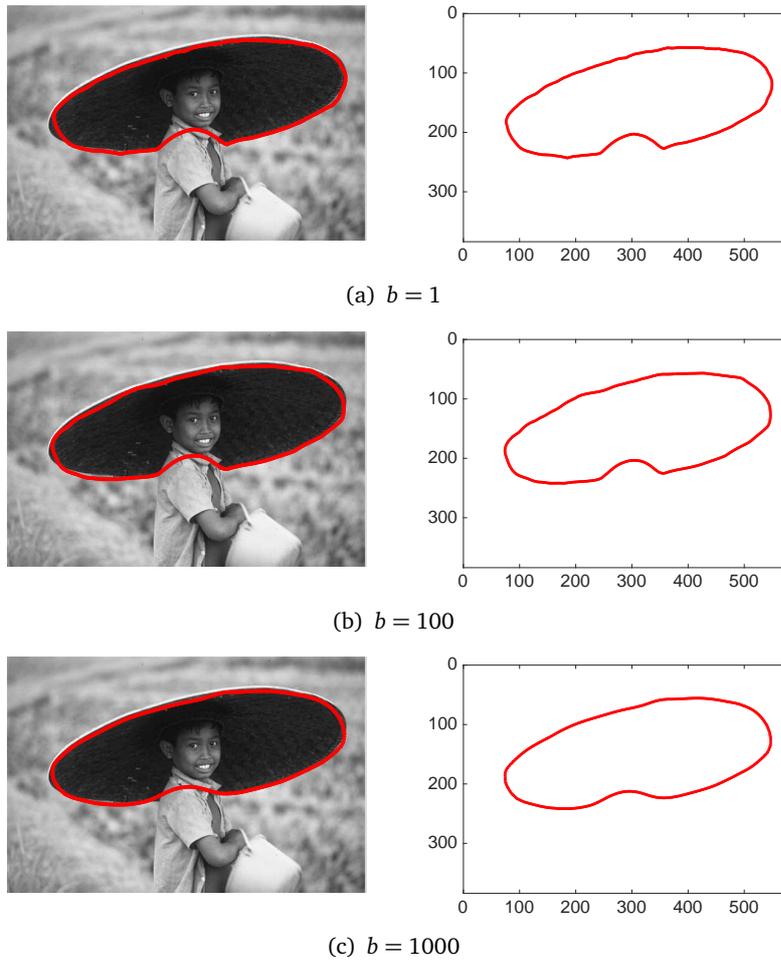
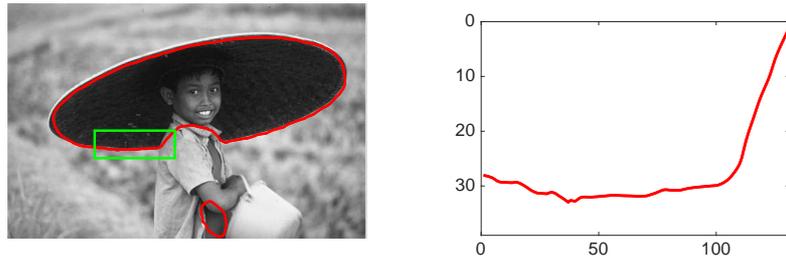
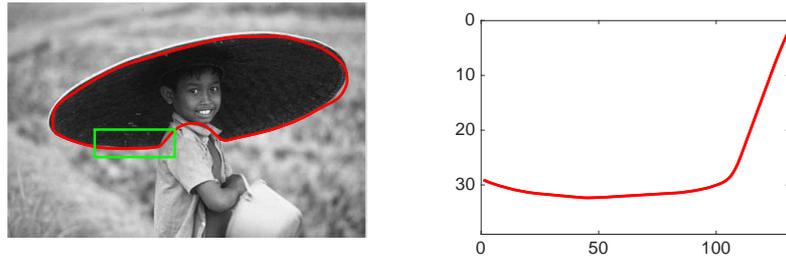


Figure 10: The left column presents segmentations using the $ECV-L^2$ model for the “hat” image with different values of b , while the right one lists the corresponding separate zero-contours.

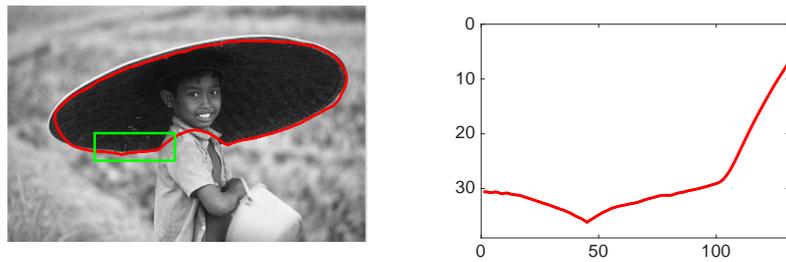
taken and enlarged. One can find that for the $ECV-L^1$ model the chopped segmentation contours are oscillating and shaky when $b = 1$, and while b increases to 10, those tiny wavy parts are completely suppressed. This is mainly due to the term $\int |\kappa| ds$, since the curvature $\kappa = d\theta/ds$, where θ describes the angle between the tangent line of contour and the positive horizontal axis [21], and therefore the oscillation of contour will lead to large value of $\int |\kappa| ds$. However, as shown in Fig. 11 for the $ECV-L^2$ model, even for $b = 100$, the chopped contours still present oscillations. This comparison again illustrates the discrepancy of the two models in their sensitiveness of convexity of segmentation contour. Moreover, from Fig. 11, one can see that when $b = 1$, besides the contour of “hat”, the $ECV-L^1$ model also keeps a small ellipse-like curve around the boy’s elbow, however, with the same value of b , this segmentation contour is missing completely for the $ECV-L^2$ model, which indicates another important difference between the two models: the $ECV-L^2$ model is more prone to removing small objects than the $ECV-L^1$ model.



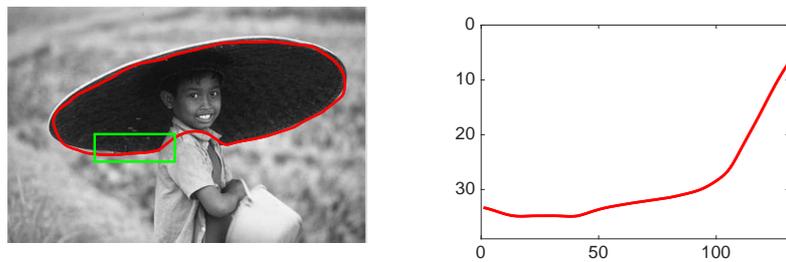
(a) $b = 1$ for the $ECV-L^1$ model



(b) $b = 10$ for the $ECV-L^1$ model



(c) $b = 1$ for the $ECV-L^2$ model



(d) $b = 100$ for the $ECV-L^2$ model

Figure 11: The comparison of zoomed-in details for the $ECV-L^1$ and $ECV-L^2$ models with different values of b . The left column lists the segmentation results for both models, and the right one shows the zoomed-in contours prescribed in those green windows. For the $ECV-L^1$ model, when b changes from $b = 1$ to $b = 10$, the contour is stretched out and oscillations are well suppressed, while for the $ECV-L^2$ model, even for $b = 100$, it still allows those oscillations of contours. This difference is mainly due to the special feature of the L^1 -Euler's elastica, which penalizes those oscillations.

6. Conclusions

In this paper, we propose novel ALM based algorithms for the $ECV-L^1$ and L^2 segmentation models in order to compare their performance. When compared with our previous work [4], we introduced a new constraint that requires the segmentation function to be a signed distance function, which helps avoid the utilization of reinitialization procedure during the iteration. By applying the proposed algorithms, the analytical differences, that is, the preference of convex contour and preservation of corners, are justified. Moreover, the zoomed in results also demonstrate that once the curvature parameter is chosen large, the $ECV-L^1$ is more inclined to remove small oscillations than the $ECV-L^2$ model. Besides these difference, as shown in [4, 35], both models are able to connect broken parts automatically and are more suitable to keep elongated structures than the CV model.

Acknowledgments X. C. Tai was supported by the startup grant at Hong Kong Baptist University, grant RG(R)-RC/17-18/02-MATH and FRG2/17-18/033.

References

- [1] L. AMBROSIO AND S. MASNOU, *A direct variational approach to a problem arising in image reconstruction*, Interface. Free Bound., 5(1) (2003), pp. 63–81.
- [2] L. AMBROSIO AND S. MASNOU, *On a variational problem arising in image reconstruction*, Free Boundary Problems, Birkhäuser Basel, 147 (2003), pp. 17–26.
- [3] E. BAE, J. SHI AND X. C. TAI, *Graph cuts for curvature based image denoising*, IEEE T. Image Proces., 20 (2011), pp. 1199–1210.
- [4] E. BAE, X. C. TAI AND W. ZHU, *Augmented lagrangian method for an euler’s elastica based segmentation model that promotes convex contours*, Inverse Probl. Imag., 11(1) (2017), pp. 1–23.
- [5] G. BIRKHOFF AND C. R. DE BOOR, *Piecewise polynomial interpolation and approximation*, Proc. General Motors Symp. of, 1965, pp. 164–190.
- [6] C. BRITO-LOEZA AND K. CHEN, *Multigrid algorithm for high order denoising*, SIAM J. Imaging Sci., 3 (2010), pp. 363–389.
- [7] V. CASELLES, R. KIMMEL AND G. SAPIRO, *Geodesic active contours*, Int. J. Comput. Vision, 22(1) (1997), pp. 61–79.
- [8] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vis., 40(1) (2011), pp. 120–145.
- [9] T. F. CHAN, S. ESEDOGLU AND M. NIKOLOVA, *Algorithms for finding global minimizers of image segmentation and denoising models*, SIAM J. Appl. Math., 66(5) (2006), pp. 1632–1648.
- [10] T. F. CHAN, S. H. KANG AND J. H. SHEN, *Euler’s elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63(2) (2003), pp. 564–592.
- [11] T. F. CHAN AND L. A. VESE, *Active contours without edges*, IEEE T. Image Proces., 10(2) (2001), pp. 266–277, 2001.
- [12] D. EBERLY, *Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid*, Geometric Tools, LLC, 2011.
- [13] N. Y. EL-ZEHIRY AND L. GRADY, *Fast global optimization of curvature*, Computer Vision & Pattern Recognition, 23(3) (2010), pp. 3257–3264.
- [14] M. ELSEY AND S. ESEDOGLU, *Analogue of the total variation denoising model in the context of geometry processing*, Multiscale Model. Sim., 7(4) (2009), pp. 1549–1573.

- [15] R. GLOWINSKI AND P. L. TALLEC, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM, 1989.
- [16] R. GLOWINSKI, T. W. PAN AND X. C. TAI, *Some Facts about Operator-Splitting and Alternating Direction Methods*, Springer International Publishing, 2016.
- [17] M. KASS, A. WITKIN AND D. TERZOPOULOS, *Snakes: Active contour models*, *Int. J. Comput. Vision*, 1(4) (1988), pp. 321–331.
- [18] R. L. LEVIEN, *From spiral to spline: Optimal techniques in interactive curve design*, PhD Thesis, University of California, Berkeley, 2009.
- [19] C. M. LI, C. Y. XU, C. F. GUI AND M. D. FOX, *Level set evolution without re-initialization: a new variational formulation*, *Computer Vision and Pattern Recognition, 2005, CVPR 2005, IEEE Computer Society Conference*, 1 (2005), pp. 430–436.
- [20] J. LIE, M. LYSAKER AND X. C. TAI, *A binary level set model and some applications to mumford-shah image segmentation*, *IEEE T. Image Proces.*, 15(5) (2006), pp. 1171–1181.
- [21] M. P. D. CARMO, *Differential Geometry of Curves and Surfaces*, Prentice Hall, 1976.
- [22] D. MUMFORD, *Elastica and Computer Vision*, Springer-Verlag, New York, 1994.
- [23] D. MUMFORD AND J. SHAH, *Optimal approximations by piecewise smooth functions and associated variational problems*, *Commun. Pure Appl. Math.*, 42(5) (1989), pp. 577–685.
- [24] M. NITZBERG, D. MUMFORD AND T. SHIOTA, *Filtering, Degmentation and Depth*, 1993.
- [25] S. OSHER AND J. A. SETHIAN, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, *J. Comput. Phys.*, 79(1) (1988), pp. 12–49.
- [26] D. P. PENG, B. MERRIMAN, S. OSHER, H. K. ZHAO AND M. KANG, *A PDE-based fast local level set method*, *J. Comput. Phys.*, 155(2) (1999), pp. 410–438.
- [27] R. T. ROCKAFELLAR, *Augmented lagrangians and applications of the proximal point algorithm in convex programming*, *Math. Oper. Res.*, 1(2) (1976), pp. 97–116.
- [28] T. SCHOENEMANN, F. KAHL AND D. CREMERS, *Curvature regularity for region-based image segmentation and inpainting: a linear programming relaxation*, *IEEE Int. Conf. Comput. Vis. (ICCV)*, 2009.
- [29] X. C. TAI, J. HAHN AND G. J. CHUNG, *A fast algorithm for euler’s elastica model using augmented lagrangian method*, *SIAM J. Imaging Sci.*, 4(1) (2011), pp. 313–344.
- [30] C. L. WU AND X. C. TAI, *Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models*, *SIAM Journal on Imaging Sciences*, 3(3) (2010), pp. 300–339.
- [31] F. L. YANG, K. CHEN AND B. YU, *Homotopy method for a mean curvature-based denoising model*, *Appl. Numer. Math.*, 62 (2012), pp. 185–200.
- [32] J. YUAN, E. BAE, AND X. C. TAI, *A study on continuous max-flow and min-cut approaches*, *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010, pp. 2217–2224.
- [33] W. ZHU AND T. CHAN, *A variational model for capturing illusory contours using curvature*, *J. Math. Imaging Vis.*, 27(1) (2007), pp. 29–40.
- [34] W. ZHU, X. C. TAI, AND T. CHAN, *Augmented lagrangian method for a mean curvature based image denoising model*, *Inverse Prob. Imag.*, 7(4) (2013), pp. 1409–1432.
- [35] W. ZHU, X. C. TAI AND T. CHAN, *Image segmentation using Euler’s elastica as the regularization*, *J. Sci. Comput.*, 57(2) (2013), pp. 414–438.