Control and Optimization in Applied Mathematics (COAM)
Vol. 8, No. 1, Winter-Spring 2023 (55-67), ©2016 Payame Noor University, Iran

DOI. 10.30473/coam.2022.61960.1185 (Cited this article)

Research Article

Optimal Edges in Morphological Snakes

Rasool Hatamian Joghali*

*Department of Mathematics, Payame Noor University (PNU),
P.O. BOX 19395-4697, Tehran, Iran.

Received: December 03, 2021; **Accepted:** February 24, 2023.

Abstract. In 2010, Alvarez et al. proposed an algorithm for morphological snakes that could detect objects whose edges consist of convex sets and polygonal edges. However, the algorithm may not detect the boundary well if the edges of an object contain a convex set or if there are several separated objects in an image. In this paper, we present two optimal sub-algorithms that are modifications to the Alvarez et al. algorithm. Our algorithms provide optimal edge detection for images and we present examples to demonstrate their effectiveness.

Keywords. Morphological curves, Partial differential equation, Numerical solution, Optimization.

MSC. 15A18; 15A69; 74B99.

* Corresponding author
hatamianr@pnu.ac.ir
https://mathco.journals.pnu.ac.ir

1 Introduction

An important problem in image analysis is object segmentation. It involves the isolation of a single object from the rest of the image which may include other objects and a background. Here, we focus on boundary detection of one or several objects by a morphological approach to the solution of the PDE associated with snake model evolution [2]. Snakes are deformable models that are based on minimizing energy along a curve. The curve, or snake, deforms its shape to minimize "internal" and "external" energies along its boundary. The internal component keeps the curve smooth, while the external component attaches the curve to image structures, such as edges, lines and etc.

For object segmentation in images, such as tracking the moving objects in a sequence of images, an unconditionally stable numerical scheme has been introduced in [5] that implements a fast version of the geodesic active contour model. In [10], a new level set model is proposed for the segmentation of biomedical images. The image energy of the proposed model is derived from a robust image gradient feature. It gives the active contour as a global representation of the geometric configuration which makes it more robust in dealing with image noise, weak edges, and initial configurations. In [6] an algorithm is given for automatically detecting contours using the snake algorithm. A piece of prior knowledge is first used to locate the initial contours of the snakes. Also, a supervised active contour model, named the Self-Organizing active contour model, is introduced in [1] which combines a variational level set method with the weights of the neurons of two Self-Organizing Maps. Moreover, in [9] a local- and global-statistics-based active contour model for image segmentation has been given which applies the globally convex segmentation method. Another morphological approach has been introduced in [2]. By combining the morphological operators associated with the PDE components they achieve a new snakes evolution algorithm. In 2019, Wang et al. proposed a multi-atlas active contour segmentation method using template optimization algorithm [8]. In 2021, by minimizing the energy of the segmentation process, Fox et al. have introduced a Hybrid Morphological Active Contour for Natural Images [4].

The algorithm given in [2], which we abbreviate it by ABHM hereafter, is simple and detects an object whose edges consist of convex sets and polygonal edges. When the edges of an object are polygon or convex set, the ABHM algorithm detects well the edges. However, in some cases where the edges of an object involve a concave set, the ABHM algorithm is unable to detect the edges.

Figure 1(a) shows the letter S as an object and Figure 1(b) shows its edges obtained by the ABHM algorithm that is clearly a false boundary. Figure 1(c) sharps the false edges obtained in Figure 1(b).

In another case, when we are dealing with multiple objects in an image, sometimes the ABHM algorithm is unable to detect the correct boundary. Figure 2(a) shows three objects in an image. The detected edges with this algorithm are shown in Figure 2(b) and Figure 2(c) sharps the edges obtained in Figure 2(b). As it is seen, although the boundary of the objects in Figure 2(a) is a polygon, the detected edges are false.

In this work, we improve the ABHM algorithm for

- detecting the edges of an object which consist of convex and concave sets,

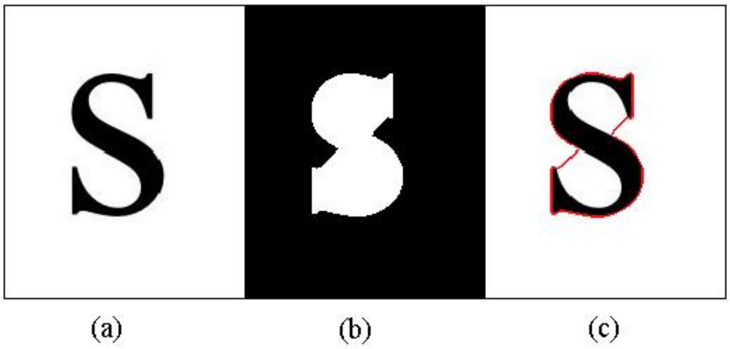

Figure 1: Finding the edges of a concave object by ABHM algorithm: (a) Original image, (b) boundary detected with $\theta_1 = 0.999$, $\theta_2 = 0.3$ and $\theta_3 = 0.1$ and $\nu = -1$, (c) sharpening the edges of (b).

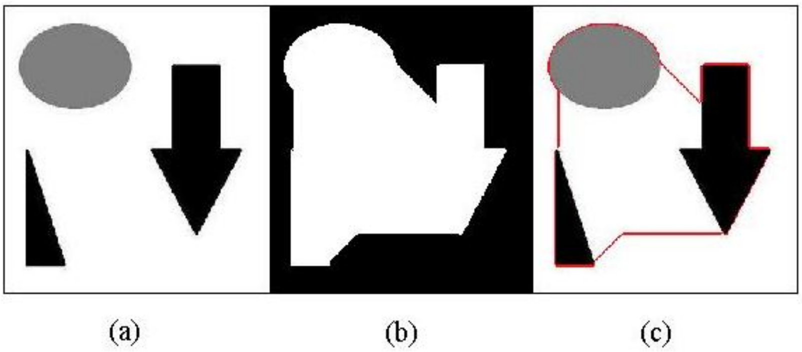

Figure 2: Finding the edges of multiple objects by ABHM algorithm: (a) Original image, (b) boundary detected with $\theta_1 = 0.999$, $\theta_2 = 0.3$ and $\theta_3 = 0.7$ and $\nu = -1$, (c) sharpening the edges of (b).

- detecting the edges of multiple objects in an image.

The structure of this paper is as follows. In Section 2, we present some preliminaries on active contours and level sets. In Section 3, we present the morphological evolution of geodesic active contours given in [2]. Finally, to generalize the ABHM algorithm for a wider range of images, our proposed method is given in Section 4.

2 Active Contours and Level Sets

For a given $C(0)$, the evolution of a closed simple planar curve in time along its normal direction $\vec{N}$ is defined by

$$C(p,t) = F.\vec{N},$$

where $C(p,t) : [0,1] \times [0,T) \to \mathbb{R}^2$ is the curve description, p denotes a mapping from a unit interval to the simple closed curve, t is the time of evolution that starts from an initially given curve and evolves by F, a smooth scalar velocity in the normal to the curve direction $\vec{N}$. Our purpose is to find some specific objects on an image by curve evolution. To this end, let $(\cdot,\cdot)$ denotes the usual inner product with Euclidean meter and $I:[0,M]\times[0,N]\longrightarrow\mathbb{R}$ be a gray-scale image. In the specific case of the active geodesic contours, the scalar F can be given below (see [2, 3]).

$$F = g(I)\kappa + g(I)\nu - (\nabla g(I),\vec{N}),$$

where κ is the curvature, ν is a constant value and $g(I)$ can be given in the following forms.

$$\begin{aligned} g(I) &= \frac{1}{\sqrt{1+\alpha|\nabla G_\sigma * I|}},\\ g(I) &= |G_\sigma * I|,\\ g(I) &= 1,\\ g(I) &= \frac{1}{1+|\nabla I|}. \end{aligned}$$

The Osher-Stethian [7] level set method represents the curve in an implicit form as the level set of an embedding function. Let $u:\mathbb{R}^+\times\mathbb{R}^2\longrightarrow\mathbb{R}$ be an implicit representation of C such that $C(t)=\{(x,y)|u(t,(x,y))=0\}$, i.e., C is the zero level set of u. The evolution of any function $u(x,y)$ which embeds the curve as one of its level sets is (see [3])

$$\frac{\partial u}{\partial t} = F.|\nabla u|.$$

By knowing that $\kappa = div(\nabla u/|\nabla u|)$, curve evolution for the geodesic active contours gets (see [2] for more details)

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|(div(\frac{\nabla u}{|u|})+\nu)+(\nabla g(I),\nabla u). \tag{1}$$

3 Morphological Evolution of Geodesic Active Contours

In this section, we present an abstract of the morphological evolution approach that has been given in [2]. Of course, in Subsection 3.4 we add some comments on the balloon force term and the selection of suitable parameters for the attraction force term. First, we consider the implicit representation $u(0,\mathbf{x})$ of the closed simple curve $C(0)=\{\mathbf{x}:u(0,\mathbf{x})=0\}$ as a binary piecewise constant function so that

$$u(0,\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is an interior point of the curve } C(0),\\ 0, & \text{if } \mathbf{x} \text{ is an exterior point of the curve } C(0). \end{cases}$$

3.1 The balloon force term

The balloon force operator term is given by the equation

$$\frac{\partial u}{\partial t} = g(I)\nu|\nabla u|.$$

According to the sign of ν, for a constant value θ_1 such that $0 \le \theta_1 < 1$, the numerical solution of the above equation can be given by

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} D_d u^n(\mathbf{x}_i), & \text{if } \ g(I)(\mathbf{x}_i) \ge \theta_1 \text{ and } \nu > 0, \\ E_d u^n(\mathbf{x}_i), & \text{if } \ g(I)(\mathbf{x}_i) \ge \theta_1 \text{ and } \nu < 0, \\ u^n(\mathbf{x}_i), & \text{otherwise,} \end{cases} \tag{2}$$

where D_d and E_d are the discrete version of dilation and erosion that are defined as

$$(D_d u)(\mathbf{x}) = \max_{\mathbf{y}\in B} \ u(\mathbf{x}-\mathbf{y}), \tag{3}$$

$$(E_d u)(\mathbf{x}) = \min_{\mathbf{y}\in B} \ u(\mathbf{x}-\mathbf{y}), \tag{4}$$

in which B is the neighborhood 3×3 of pixels.

3.2 The attraction force term

The attraction force PDE term is given by the equation

$$\frac{\partial u}{\partial t} = \nabla g(I)\nabla u.$$

The morphological discretization of this PDE term is given by

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} 1, & \text{if } (\nabla u^n(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) > 0 \text{ and } g(I)(\mathbf{x}_i) > \theta_2, \\ 0, & \text{if } (\nabla u^n(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0 \text{ and } g(I)(\mathbf{x}_i) > \theta_2, \\ u^n(\mathbf{x}_i), & \text{otherwise,} \end{cases} \tag{5}$$

where $0 \le \theta_2 < 1$ is a constant value.

3.3 The smoothing term

The smoothing operator term can be thought of as the following equation:

$$\frac{\partial u}{\partial t} = g(I)|\nabla u|(div(\frac{\nabla u}{|\nabla u|})). \tag{6}$$

The solution of (6) can be approached using line morphological operators as follows. Let

$$\beta = \left\{ \begin{array}{c} \{(0,0),(1,0),(-1,0)\} \\ \{(0,0),(0,1),(0,-1)\} \\ \{(0,0),(1,1),(-1,-1)\} \\ \{(0,0),(1,-1),(-1,1)\} \end{array} \right\},$$

be the set of the line segments of length 2 centered at the origin. The morphological discrete line operators are defined as

$$(SI_d u)(\mathbf{x}) = \max_{B\in\beta} \min_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}), \tag{7}$$

$$(IS_d u)(\mathbf{x}) = \min_{B\in\beta} \max_{\mathbf{y}\in\mathbf{x}+B} u(\mathbf{y}). \tag{8}$$

The morphological evolution of the PDE (6) for a known value u^n can be given by

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} (SI_d \circ IS_d u^n)(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) \geq \theta_3, \\ u^n(\mathbf{x}_i), & \text{otherwise ,} \end{cases} \tag{9}$$

where $0 \leq \theta_3 < 1$ is a constant value. The instruction of $SI_d \circ IS_d$ operator, is illustrated in Figures 3, 4 and 5.

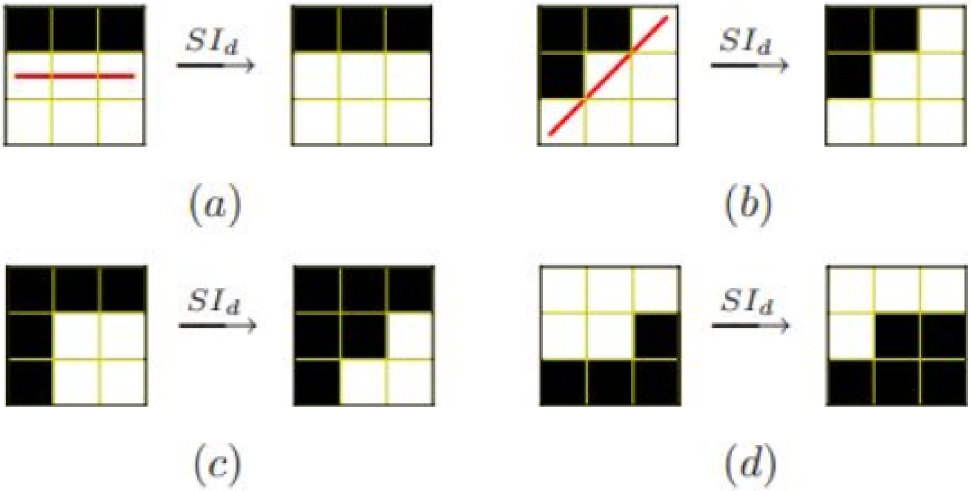

Figure 3: Some examples of the effect of the SI_d operator on individual pixels of binary images. In those cases where a straight line is found (marked in red), the central pixel remains active ((a) and (b)). When the central pixel does not belong to a straight line of active pixels, it is made inactive ((c) and (d)). For exemplification purposes, we assume the pixels on the borders are not affected by the operator.

3.4 Complete numerical solution of evolution equation

The active contour equation (1) is made up of three different components: a smoothing force, a balloon force, and an attraction force. Let $u^n : \mathbb{R}^2 \to \{0,1\}$ be the snake evolution at n-th iteration. By (2), (9) and (5), the morphological evolution scheme at iteration $n+1$, can be given by the following three steps.

$$u^{n+\frac{1}{3}}(\mathbf{x}_i) = \begin{cases} (D_d u^n)(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) = 1 \text{ and } \nu > 0, \\ (E_d u^n)(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) = 1 \text{ and } \nu < 0, \\ u^n(\mathbf{x}_i), & \text{otherwise.} \end{cases} \tag{10}$$

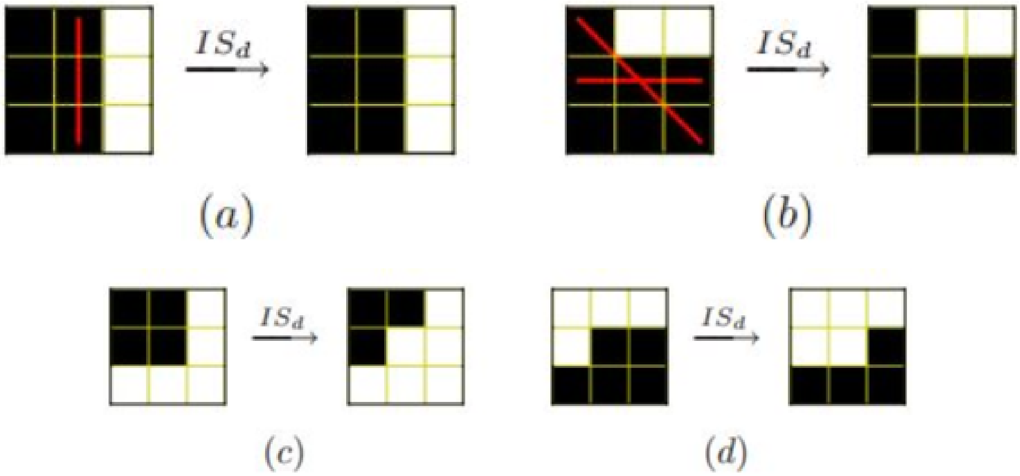

Figure 4: Some examples of the IS_d operator.

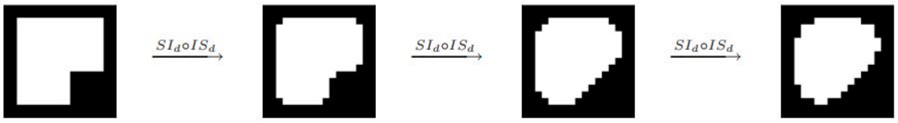

Figure 5: Some examples of the $SI_d \circ IS_d$ operator iterated until convergence.

To be satisfied the value $u^{n+\frac{1}{3}}(\mathbf{x}_i)$ in the first or second condition of (10) for $g(I)(\mathbf{x}_i)$ so many close to 1, the above step can be modified by

$$u^{n+\frac{1}{3}}(\mathbf{x}_i) = \begin{cases} (D_d u^n)(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) > \theta_1 \text{ and } \nu > 0, \\ (E_d u^n)(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) > \theta_1 \text{ and } \nu < 0, \\ u^n(\mathbf{x}_i), & \text{otherwise,} \end{cases} \tag{11}$$

where $\theta_1 = 1 - \epsilon > 0$ is a constant value and ϵ is an enough small real number. Also, the factor ν in the above step implies we take a different threshold level for the balloon operator. Indeed, the dilation (erosion) occurs if $\nu > 0$ $(\nu < 0)$.

By (5), the second step is given by

$$u^{n+\frac{2}{3}}(\mathbf{x}_i) = \begin{cases} 1, & \text{if } (\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) > 0 \text{ and } g(I)(\mathbf{x}_i) > \theta_2, \\ 0, & \text{if } (\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0 \text{ and } g(I)(\mathbf{x}_i) > \theta_2, \\ u^{n+\frac{1}{3}}(\mathbf{x}_i), & \text{if } (\nabla u^{n+\frac{1}{3}}, \nabla g(I)(\mathbf{x}_i)) = 0. \end{cases} \tag{12}$$

A suitable selection of the constant value θ_2 can be given by

$$\theta_2 = \max_{\mathbf{x_i} \in \partial \boldsymbol{\Omega}} \{g(I)(\mathbf{x_i})\}, \tag{13}$$

where $\partial\Omega$ is the boundary of the domain Ω. Finally, the numerical solution of the smoothing term can be stated by

$$u^{n+1}(\mathbf{x}_i) = \begin{cases} (SI_d \circ IS_d u^{n+\frac{2}{3}})(\mathbf{x}_i), & \text{if } g(I)(\mathbf{x}_i) > \theta_3, \\ u^{n+\frac{2}{3}}(\mathbf{x}_i), & \text{otherwise.} \end{cases} \tag{14}$$

4 Modification on the ABHM Algorithm

Although the ABHM algorithm works well in many practical cases, however, there are some images in which the ABHM algorithm fails to find out the edges of the object(s) in the images. In this section, we present some modifications to the ABHM algorithm to generalize the method given in [2] for a wider class of images. To this end, we present some sub-algorithms to modify the ABHM algorithm. In all of the sub-algorithms we use:

$$g(I) := \frac{1}{1 + |\nabla I|}.$$

4.1 Concave edge recognition

When we are dealing with an object whose edges form a convex set or a polygon, the ABHM algorithm works well. However, when the edges of an object form a concave set including nonlinear segment(s), or the object is made up of two or more separate sets, the ABHM algorithm does not usually work well. Figure 6(a) shows the number 5 as an object and Figure 6(b) shows its edges obtained by the ABHM algorithm that is clearly a false boundary. Also, Figure 6(c) sharps the false edges obtained in Figure 6(b). This defect can be described as follows.

Suppose that $\mathbf{x}_i$ is a false boundary pixel shown in Figure 6(b) The pixel $\mathbf{x}_i$ is close to the real boundary that yields $\nabla g(I)(\mathbf{x}_i) \neq 0$. Moreover, the angle between the vectors $\nabla u(\mathbf{x}_i)$ and $\nabla g(I)(\mathbf{x}_i)$ is more than $\frac{\pi}{2}$, i.e., $(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0$. Furthermore, since $\mathbf{x}_i$ is not a boundary pixel, hence, by (13), we clearly observe that the inequality $g(I)(\mathbf{x}_i) > \theta_2$ holds. Thus, by (9) we have $u^{n+\frac{2}{3}}(\mathbf{x}_i) = 0$. On the other hand, in different directions passing $\mathbf{x}_i$, we have a change of intensities, so by (10) we have $u^{n+1}(\mathbf{x}_i) = 1$ (see Figure 7). To modify the ABHM algorithm, we enter the following sub-algorithm.

Algorithm 4 Sub-Algorithm 1

```
• The structure of Sub-Algorithm 1
Step 1.
$g(I)(\mathbf{x}_i) > \theta_1 \quad and \quad \nu > 0 \ u^{n+\frac{1}{3}}(\mathbf{x}_i) = (D_d u^n)(\mathbf{x}_i)$
$g(I)(\mathbf{x}_i) > \theta_1 \quad and \quad \nu < 0 \ u^{n+\frac{1}{3}}(\mathbf{x}_i) = (E_d u^n)(\mathbf{x}_i)$ Step 2.
$(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) > 0 \quad and \quad g(I)(\mathbf{x}_i) > \theta_2 \ u^{n+\frac{2}{3}}(\mathbf{x}_i) = 1$
$(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0 \quad and \quad g(I)(\mathbf{x}_i) > \theta_2 \ u^{n+\frac{2}{3}}(\mathbf{x}_i) = 0$ Step 3.
$u^{n+\frac{2}{3}}(\mathbf{x}_i) = u^{n+\frac{1}{3}}(\mathbf{x}_i) \ g(I)(\mathbf{x}_i) > \theta_3 u^{n+1}(\mathbf{x}_i) = (SI_d \circ IS_d) u^{n+\frac{2}{3}}(\mathbf{x}_i)$
```

We note that based on the Sub-Algorithm 4, when $(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) > 0$ or $(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0$, Step 3 would not be run. In other words, Step 3 would be run, if $(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) = 0$.

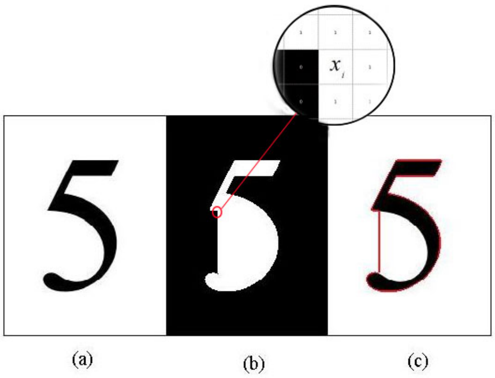

Figure 6: Finding the edges of a concave object by ABHM algorithm: (a) original image, (b) boundary detected with $\theta_1 = 0.1$, $\theta_2 = 0$, $\theta_3 = 0.18$ and $\nu = -1$, (c) shaped b.

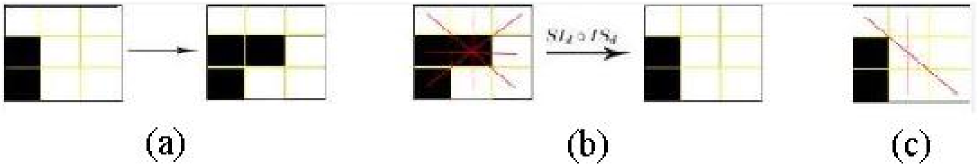

Figure 7: (a) point $\mathbf{x}_i$ would be an active pixel in Step 1, (b) point $\mathbf{x}_i$ would be an inactive pixel by smoothing in Step 3, (c) the pixel $\mathbf{x}_i$ don't change in the directions $\{(-1,-1),(0,0),(1,1)\}$ and $\{(-1,0),(0,0),(1,0)\}$ in Step 3.

4.2 Edge recognition of separated objects

When we are dealing with several separated objects in an image, the ABHM algorithm can not detect the true boundary (see Figure 8). We describe this occurrence for Figure

8(a). Suppose that $\mathbf{x}_i$ is a false boundary pixel. Because $g(I)(\mathbf{x}_i) = 1$ in (10), for a given $\nu < 0$, we have $u^{n+\frac{1}{3}}(\mathbf{x}_i) = (E_d u^n)(\mathbf{x}_i) = 0$ in (11). In other words, the pixel $\mathbf{x}_i$ would be inactive. In Step 2, $\nabla g(I)(\mathbf{x}_i) = 0$ that implies $u^{n+\frac{2}{3}}(\mathbf{x}_i) = u^{n+\frac{1}{3}}(\mathbf{x}_i)$. Finally, in a similar comment, it is seen that $u^{n+1}(\mathbf{x}_i) = (SI_d \circ IS_d u^{n+\frac{2}{3}})(\mathbf{x}_i)$ in Step 3, i.e., the pixel $\mathbf{x}_i$ would be active, that implies the curve evolution is stopped (see Figure 11). A modification of the ABHM algorithm in the presence of several separated objects in an image can be given by the sub-algorithm 5:

Algorithm 5 Sub-Algorithm 2

```
• The structure of Sub-Algorithm 2
Step 1.
$g(I)(\mathbf{x}_i) > \theta_1$   and   $\nu > 0$ $u^{n+\frac{1}{3}}(\mathbf{x}_i) = (D_d u^n)(\mathbf{x}_i)$
$g(I)(\mathbf{x}_i) > \theta_1$   and   $\nu < 0$ $u^{n+\frac{1}{3}}(\mathbf{x}_i) = (E_d u^n)(\mathbf{x}_i)$ Step 2.
$(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) > 0$   and   $g(I)(\mathbf{x}_i) > \theta_2$ $u^{n+\frac{2}{3}}(\mathbf{x}_i) = 1$
$(\nabla u^{n+\frac{1}{3}}(\mathbf{x}_i), \nabla g(I)(\mathbf{x}_i)) < 0$   and   $g(I)(\mathbf{x}_i) > \theta_2$ $u^{n+\frac{2}{3}}(\mathbf{x}_i) = 0$
```

Indeed, the Sub-Algorithm 5 is run when the Sub-Algorithm 4 stops. Also, we note that in Sub-Algorithm 5, Step 3 is removed.

In the end, we present some examples for each sub-algorithm and compare them with the ABHM algorithm. These examples show that some modifications to the ABHM algorithm are needed to detect the true edges of the object(s) in images.

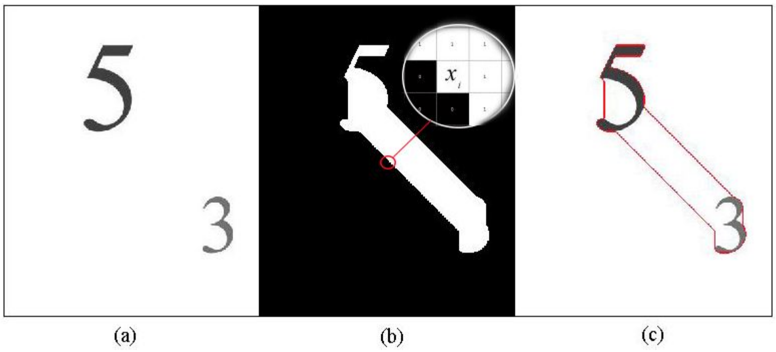

Figure 8: Finding the edges of a concave object by ABHM algorithm: (a) original image, (b) boundary detected with $\theta_1 = 0.1$, $\theta_2 = 0$, $\theta_3 = 0.8$ and $\nu = -1$, (c) sharped b.

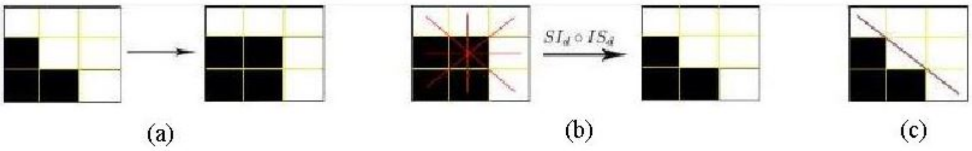

Figure 9: (a) Point $\mathbf{x}_i$ would be an active pixel in Step 1, (b) point $\mathbf{x}_i$ would be an inactive pixel by smoothing in Step 3, (c) the pixel $\mathbf{x}_i$ do not change in the direction $\{(-1,-1),(0,0),(1,1)\}$ in Step 3.

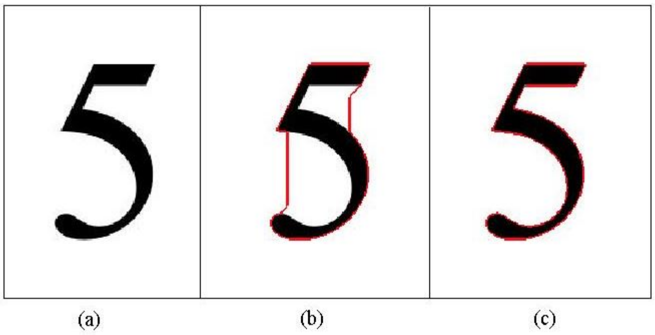

Figure 10: (a) Original image, (b) finding the edges by ABHM algorithm with $\theta_1 = 0.999$, $\theta_2 = 0.3$, $\theta_3 = 0.1$ and $\nu = -1$, (c) finding the edges by our algorithm with $\theta_1 = 0.999$, $\theta_2 = 0.3$, $\theta_3 = 0.1$ and $\nu = -1$.

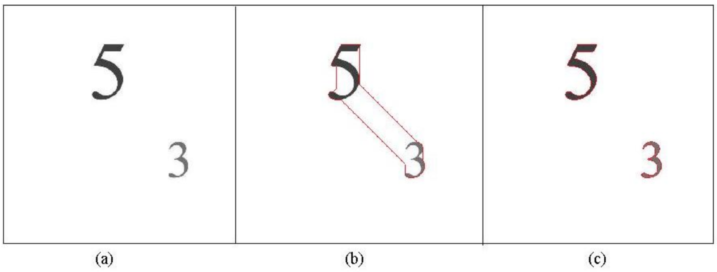

Figure 11: (a) Original image, (b) finding the edges by ABHM algorithm with $\theta_1 = 0.999$, $\theta_2 = 0.7$, $\theta_3 = 0.7$ and $\nu = -1$, (c) finding the edges by our algorithm with $\theta_1 = 0.999$, $\theta_2 = 0.7$, $\theta_3 = 0.7$ and $\nu = -1$.

Table 1 shows the CPU time, the number of iterations, and the assigned quantities $\theta_1, \theta_2, \theta_3$ and ν in Algorithm ABHM.

Table 1: The CPU time, the number of iterations, and the assigned quantities $\theta_1, \theta_2, \theta_3$ and ν in Algorithm ABHM

	Figure 6	Figure 8	Figure 10	Figure 11
CPU time(s)	15	18	22	25
Iteration	57	64	68	70
Quantities	$\theta_1 = 0.1$	0.1	0.999	0.999
	$\theta_2 = 0$	0	0.3	0.7
	$\theta_3 = 0.18$	0.8	0.1	0.7

5 Conclusion

In this paper, we presented two optimal sub-algorithms that were modifications to the Alvarez et al. algorithm. Our algorithms provided optimal edge detection for images and we presented examples to demonstrate their effectiveness. Here, we showed that the modified version of the algorithm finds better the edges of an image.

Declarations

Availability of supporting data

All data generated or analyzed during this study are included in this published paper.

Funding

This study received no funds, grants, or other financial support.

Competing interests

The authors declare no competing interests that are relevant to the content of this

paper.

Authors' contributions

The main manuscript text is collectively written by all authors.

References

[1] Abdelsamea, M.M., Gnecco G., Gaber, M.M. (2014). "An efficient self-organizing active contour model for image segmentation", Neurocomputing, 149, 820-835.

[2] Alvarez, L., Baumela, L., Marquez-Neila, P., Henriquez, P. (2010). "Morphological snakes", Computer Vision and Pattern Recognition, 2197-2202.

[3] Caselles, V., Kimmel, R., Sapiro, G. (1997). "Geodesic active contours", International Journal of Computer Vision, 22, 61-79.

[4] Fox, V.L., Milanova, M., Al-Ali, S. "A hybrid morphological active contour for natural images", Available at SSRN 3775362, 2021.

[5] Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M. (2001). "Fast geodesic active contours", IEEE Transactions Image Process, 10, 1467-1475.

[6] Kussener, F. (2011). "A parallel genetic algorithm approach", International Conference on Swarm Intelligence, 1-9.

[7] Osher, S., Sethian, J.A. (1988). "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations", Journal of Computational Physics, 79, 12-49.

[8] Wang, M., Li, P., Liu, F. (2019). "Multi-atlas active contour segmentation method using template optimization algorithm", BMC Medical Imaging, 19(1), 1-13.

[9] Wu, B., Yang, Y. (2012)., "local- and global-statistics-based active contour model for image segmentation", Mathematical Problems in Engineering, 1-16.

[10] Yeo, S. Y., Xie, X., Sazonov, I., Nithiarasu, P. (2014). “Segmentation of biomedical images using active contour model with robust image feature and shape prior”, International Journal for Numerical Methods in Biomedical Engineering, 30, 232-248.

How to Cite this Article:
Hatamian Joghali, R. (2023). “Optimal edges in morphological snakes”, Control and Optimization in Applied Mathematics, 8(1): 55-67. doi: 10.30473/coam.2022.61960.1185.