Simultaneous Tracking of Multiple Objects Using Fast Level Set-Like Algorithm

Martin Maška, Pavel Matula, and Michal Kozubek

Centre for Biomedical Image Analysis, Faculty of Informatics
Masaryk University, Brno, Czech Republic
xmaska@fi.muni.cz

Abstract

A topological flexibility of implicit active contours is of great benefit, since it allows simultaneous detection of several objects without any a priori knowledge about their number and shapes. However, in tracking applications it is often required to keep desired objects mutually separated as well as allow each object to evolve itself, i.e., different objects cannot be merged together, but each object can split into several regions that can be merged again later in time. The former can be achieved by applying topology-preserving constraints exploiting either various repelling forces or the simple point concept from digital geometry, which brings, however, an indispensable increase in the execution time and also prevent the latter. In this paper, we propose more efficient and more flexible topology-preserving constraint based on a region indication function, that can be easily integrated into a fast level set-like algorithm [15] in order to obtain a fast and robust algorithm for simultaneous tracking of multiple objects. The potential of the modified algorithm is demonstrated on both synthetic and real image data.

Keywords and phrases level set framework, topology preservation, object tracking

Digital Object Identifier 10.4230/OASIcs.MEMICS.2010.69

1 Introduction

Detection and tracking of object boundaries is an important task in many computer vision applications such as video surveillance, monitoring, or robotics as well as in biomedical studies aimed at understanding the mechanics of cellular processes such as proliferation, differentiation, or migration. In general, desired objects can have arbitrary initial shapes that can, in addition, undergo changes in time. Therefore, an optimal tracking algorithm should be able to detect objects of complex boundaries and adapt easily to their changes. Furthermore, it should also achieve real-time or at least near real-time performance in order to be fruitfully applied in practice.

Implicit active contours [4, 5, 6, 23] have become popular namely due to their inherent topological flexibility and ability to detect objects of complex shapes. Their solution is usually carried out using the level set framework [19, 18], in which the contour is represented implicitly as the zero level set (also called *interface*) of a scalar higher-dimensional function. This representation has several advantages over the parametric one [10, 3]. In particular, it avoids parameterization problems, the topology of the contour is handled inherently, and the extension into higher dimensions is straightforward. On the other hand, a numerical solution of associated partial differential equations brings a significant computational burden limiting the use of this approach in real-time applications.

Many approximations, aimed at speeding up the basic level set framework, have been proposed in last two decades. In the family of gradient-based implicit active contours [4, 5], the narrow band [1], sparse-field [25], and fast marching method [20] have become popular. Later, other interesting approaches based on the additive operator splitting scheme [8]

Sixth Doctoral Workshop on Math. and Eng. Methods in Computer Science (MEMICS'10)—Selected Papers.
Editors: L. Matyska, M. Kozubek, T. Vojnar, P. Zemčík, D. Antoš; pp. 69–76
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

or a pointwise scheduled propagation of the implicit contour [7, 17] have emerged. Shi and Carl [22] proposed a fast algorithm that is able to track the gradient-based as well as region-based [6, 23] implicit active contours, provided the speed function can be decomposed into data-dependent and regularization terms. We also refer the reader to the work by Lie et al. [12], Wang et al. [24], and Maška et al. [15] introducing other fast algorithms that minimize popular Chan-Vese model [6].

The topological flexibility of implicit active contours is of great benefit, since it enables to detect several objects simultaneously without any a priori knowledge about their number or shapes. However, for tracking purposes such a flexibility is not always suitable. For instance, when two initially isolated objects touch later in time it is often required to keep them separated. This can be achieved by applying topology-preserving constraints based on either various repelling forces [2, 11] or the simple point concept from digital geometry [9, 14], which brings, however, an indispensable increase in the execution time caused by their evaluation in a local neighbourhood of the interface. Furthermore, they also prevent each object from being evolved at will, e.g., from splitting into several regions.

In this paper, we propose more flexible topology-preserving constraint that brings only negligible increase in the execution time. It exploits a region indication function, has constant time complexity, and can be easily integrated into our fast level set-like algorithm [15] in order to obtain a fast and robust algorithm for simultaneous tracking of multiple objects based on the minimization of the Chan-Vese model [6]. In comparison to the tracking algorithm by Shi and Carl [21] that exploits the region indication function as well, the proposed algorithm does not require the contours to be initially separated by the background nor evaluate relaxed topological numbers. It also allows two different object contours to touch inherently, without any additional tests.

The organization of the paper is as follows. In Section 2, the theoretical background of the Chan-Vese model and the basic principle of our fast level set-like algorithm [15] intended for its minimization are reviewed. Section 3 is devoted to the topology-preserving modification of the original algorithm. Experimental results are demonstrated in Section 4. We conclude the paper with a discussion and suggestions for future work in Section 5 and 6, respectively.

2 Fast Algorithm Minimizing the Chan-Vese Model

In order to obtain a mathematically easier minimization problem, Chan and Vese [6] introduced a piecewise constant approximation to the well-known functional formulation of image segmentation by Mumford and Shah [16]. Let Ω be an image domain and $u_0 : \Omega \to \mathbb{R}$ be an input image defined over this domain. The basic idea of the Chan-Vese model is to find a piecewise constant approximation of u_0 being constant in two possibly disconnected regions Ω_1 and Ω_2 of constant levels c_1 and c_2, respectively, separated by a closed segmenting contour C ($\Omega = \Omega_1 \cup \Omega_2 \cup C$) of minimal length. The Chan-Vese model can be formulated as

$$E_{CV}(C, c_1, c_2) = \mu|C| + \lambda_1 \int_{\Omega_1} (u_0(x) - c_1)^2 \, dx + \lambda_2 \int_{\Omega_2} (u_0(x) - c_2)^2 \, dx \quad , \tag{1}$$

where μ is nonnegative and λ_1 and λ_2 are positive constants. Embedding the contour C in a scalar higher-dimensional function ϕ with C as its zero level set, the functional can be minimized using the level set framework. The associated Euler-Lagrange equation has the following form:

$$\frac{\partial \phi}{\partial t} + \delta_\varepsilon(\phi) \left[\mu \cdot \mathrm{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \lambda_1 (u_0 - c_1)^2 + \lambda_2 (u_0 - c_2)^2 \right] = 0 \quad , \tag{2}$$

where

$$c_1 = \frac{\int_\Omega u_0(x)(1 - H_\varepsilon(\phi(x)))\,dx}{\int_\Omega (1 - H_\varepsilon(\phi(x)))\,dx} \quad \text{and} \quad c_2 = \frac{\int_\Omega u_0(x) H_\varepsilon(\phi(x))\,dx}{\int_\Omega H_\varepsilon(\phi(x))\,dx} \ . \tag{3}$$

The symbols H_ε and δ_ε denote regularized versions of the Heaviside and Dirac delta functions. Careful attention has to be paid to the regularization of these functions, since it affects the model performance. Provided δ_ε is nonzero in the whole domain, the Chan-Vese model has the tendency to compute a global minimizer. On the contrary, the choice of δ_ε with a compact support results only in a local minimizer and, therefore, the dependence on the initialization.

In our previous work [15], we introduced a fast level set-like algorithm that locally minimizes the Chan-Vese model (a suitable choice of initial model, however, often leads to finding a global minimum) and avoids a nontrivial and time-consuming numerical solution of the associated Euler-Lagrange equation. Instead of evolving the whole implicit function in a small time step, only the interface points stored in a list data structure are moved to the exterior or interior depending on the sign of the speed function F in the normal direction given as

$$F = \mu\kappa - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2 \ , \tag{4}$$

where κ denotes the curvature of the interface. Simultaneously, their local neighbourhoods (4-neighbourhoods in 2D and 6-neighbourhoods in 3D, respectively) are updated accordingly. The local propagation of each interface point allows the values c_1 and c_2 to be updated incrementally, since we know exactly which points move to the exterior and interior. Furthermore, considering the level set function ϕ as a mapping of the set membership of each point (i.e. the points of the interface are represented by the value 0, interior points by -1, and exterior ones by 1), the curvature of the interface can be roughly approximated in an incremental manner. These ideas result in a fast algorithm for tracking implicit contours driven by the Chan-Vese model. We refer the reader to the original paper [15] for further details.

3 Topology-Preserving Modification

To ensure that different objects are kept mutually separated as well as allow each object to evolve itself, we integrate our fast algorithm described in the previous section with a region indication function $\psi : \Omega \to \{0, 1, 2, \dots\}$ that is evolved simultaneously with the simplified level set function ϕ. Remind that in each iteration the original algorithm propagates each interface point locally depending on the sign of the speed function F. Therefore, a modification of the local propagation of each interface point will result in a modification of the original algorithm itself.

Let ϕ be determined by a possibly disconnected background region Ψ_0 and M possibly disconnected disjoint objects $\Psi_1, \Psi_2, \dots, \Psi_M$ $(\Omega = \bigcup_{0 \le i \le M} \Psi_i)$. Let $p \in \Omega$ be a point of the interface of the object Ψ_i, $0 < i \le M$, that is being propagated. The behaviour of the modified algorithm can be divided into two cases depending on the sign of $F(p)$. First, assume that $F(p) < 0$ (Fig. 1a). The original algorithm transfers p to the exterior and adds all its interior neighbours to the interface. The modified algorithm behaves in the same way as the original one. Clearly, only p is switched from the foreground to the background. It is therefore sufficient to reset its region indicator to 0.

The second case, when $F(p) > 0$ (Fig. 1b), is more complicated than the first one. The original algorithm transfers p to the interior and adds all its exterior neighbours (denote them

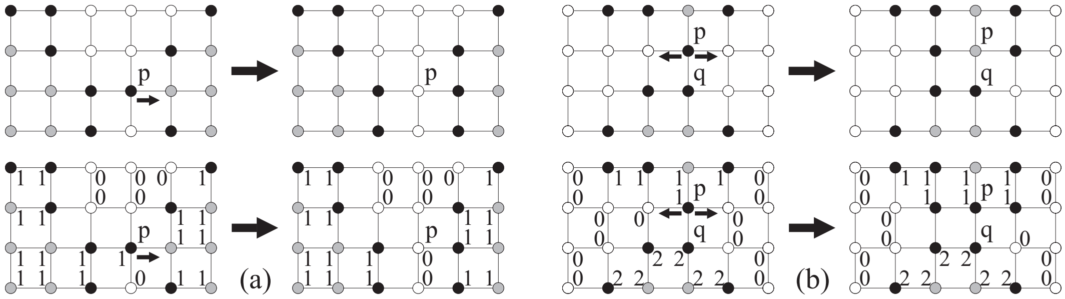

Figure 1 Comparison of one iteration of the original algorithm (top row) and the modified one (bottom row) in case of (a) $F(p) < 0$ and (b) $F(p) > 0$. The black points correspond to the interface, the white ones to the exterior, and the gray ones to the interior. The arrows from p correspond to the directions of possible propagations of the interface in this iteration. The numbers correspond to the region indication function ψ.

by $E(p)$) to the interface. In this case, each point in $E(p)$ is switched from the background to the foreground. Therefore, the modified algorithm changes their region indicators to i. It is important to note that one more test has to be performed in the modified algorithm in order to preserve the interface connectedness of each object. Let $N(p)$ be a set of neighbours of p of different region indicators. Clearly, if $|E(p)| < |N(p)|$, p must be put back to the interface, $\phi(p) = 0$, in order to preserve the interface connectedness of the object Ψ_i, since p has a neighbour q of the region indicator j, $0 < j \leq M$, $j \neq i$, that belongs to the interface of the object Ψ_j.

4 Experimental Results

In this section, we present several results and comparisons on both synthetic and real image data to demonstrate the potential of the proposed algorithm. The experiments have been performed on a common workstation (Intel Core2 Duo 2.0 GHz, 2 GB RAM, Windows XP Professional). For comparison purposes, we integrated the original algorithm [15] with the simple point concept from digital geometry to obtain a fast topology-preserving alternative to the modified algorithm described in the previous section. We denote these algorithms as SP (simple point) and RI (region indicator), respectively, depending on the concept used for preserving the contour topology.

We start with a synthetic binary image of size 200 × 200 pixels containing two circles (Fig. 2). In case of the SP algorithm, the contour cannot change its topology and, therefore, only one 8-connected component is obtained as a result. On the other hand, the RI algorithm allows the contour to split into several parts and each circle is detected separately. The execution time was less than 0.01 seconds in both cases.

The second experiment is aimed at separation of two touching objects in a noisy synthetic image of size 350 × 170 pixels (Fig. 3). Both algorithms output two 8-connected components. However, in case of the SP algorithm they are separated by often undesired 4-connected background path. The computation took 0.014 and 0.013 seconds, respectively.

We conclude this section with an application of the SP and RI algorithms for tracking of AIF-transfected living cells of the MCF-7 cell line (Fig. 4 and 5, respectively). The time-lapse series acquired using a fluorescence microscope has 25 frames of size 648 × 515 pixels. The execution time was about 0.111 and 0.107 seconds, respectively, in average per frame.

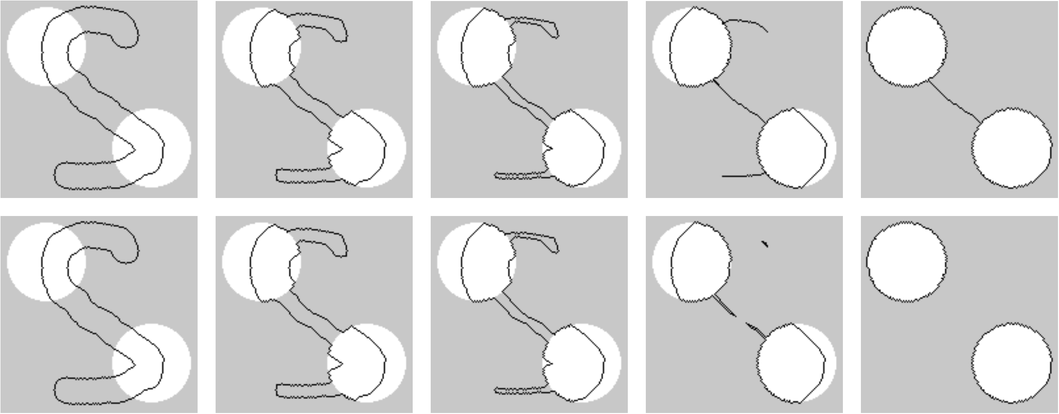

Figure 2 Segmentation of a synthetic image with two circles ($\mu = 0.5$, $\lambda_1 = \lambda_2 = 1$). Top row: Evolution of the SP contour. Bottom row: Evolution of the RI contour.

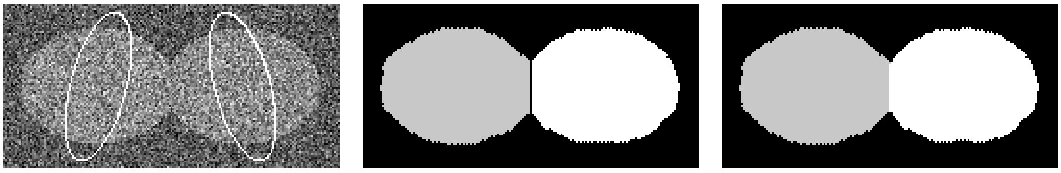

Figure 3 Segmentation of touching objects ($\mu = 0.5$, $\lambda_1 = \lambda_2 = 1$). Left: Input image overlaid with two initial contours. Centre: Segmentation result of the SP algorithm. Right: Segmentation result of the RI algorithm.

5 Discussion

The final evaluation of the modified algorithm is introduced in this section. We discuss, namely, the experimental results presented in Sect. 4 in detail.

The topology-preserving constraint exploiting the region indication function is very simple and has constant time complexity. There is no need to evaluate any complex condition in a local neighbourhood of a considered point. In comparison to the original algorithm, the increase in the execution time of the modified algorithm is negligible, from about 2 up to 4 percent in both 2D as well as 3D. Compared with the SP algorithm, it is about 4 percent faster in 2D and even about 9 percent faster in 3D, where the breadth-first search algorithm [13] has been used for the simple point detection. On the other hand, the RI algorithm consumes slightly more memory than the others, since it requires additional space for storing region indicators. However, the increase is less than 5 percent.

The experiments illustrated in Fig. 2–5 showed the main advantages of the RI algorithm over the SP one for simultaneous tracking of multiple objects. Considering the simplest tracking scheme in which the final contour from the previous frame is used as a seed in the next one, the RI algorithm adapts easily to splitting of a connected object in one frame into several regions in the next one. Furthermore, it also allows us to find boundaries of touching objects without any background gap between them. It is important to note that considered tracking scheme might have problems in situations involving large movements of the objects or when the final contour of one object from the previous frame overlaps with another object in the next frame. This will be addressed in future work.

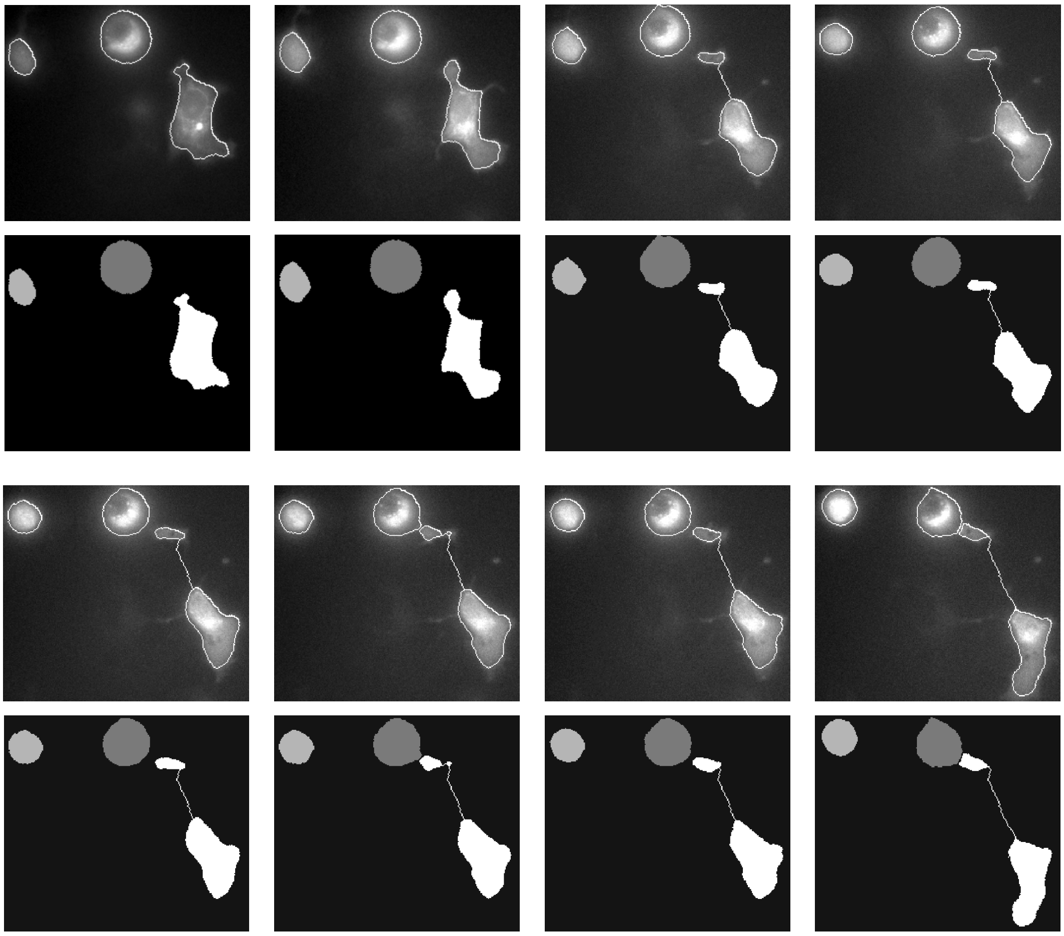

Figure 4 Tracking of AIF-transfected living cells using the SP algorithm ($\mu = 0.3$, $\lambda_1 = 1$, $\lambda_2 = 2$). The frames number 1, 3, 7, 10, 13, 14, 15, and 22 are shown. Top rows: Original image data overlaid with final contours. Bottom rows: Segmentation results of the SP algorithm.

6 Conclusion

We have addressed the problem of imposing topology-preserving constraints on evolving implicit contours. We have proposed a topology-preserving constraint exploiting a region indication function, that is more flexible than the ones based on either various repelling forces or the simple point concept from digital geometry, has constant time complexity, and can be easily integrated into our fast level set-like algorithm minimizing the Chan-Vese model. The experiments verified topology-preserving properties of the modified algorithm and showed its speed and better usability for simultaneous tracking of multiple objects in comparison to the one exploiting the simple point concept.

Acknowledgments. This work has been supported by the Ministry of Education of the Czech Republic (Projects No. MSM-0021622419, No. LC535 and No. 2B06052). The authors would also like to thank Dr. Miroslav Vařecha for providing the time-lapse series of MCF-7 cell line.

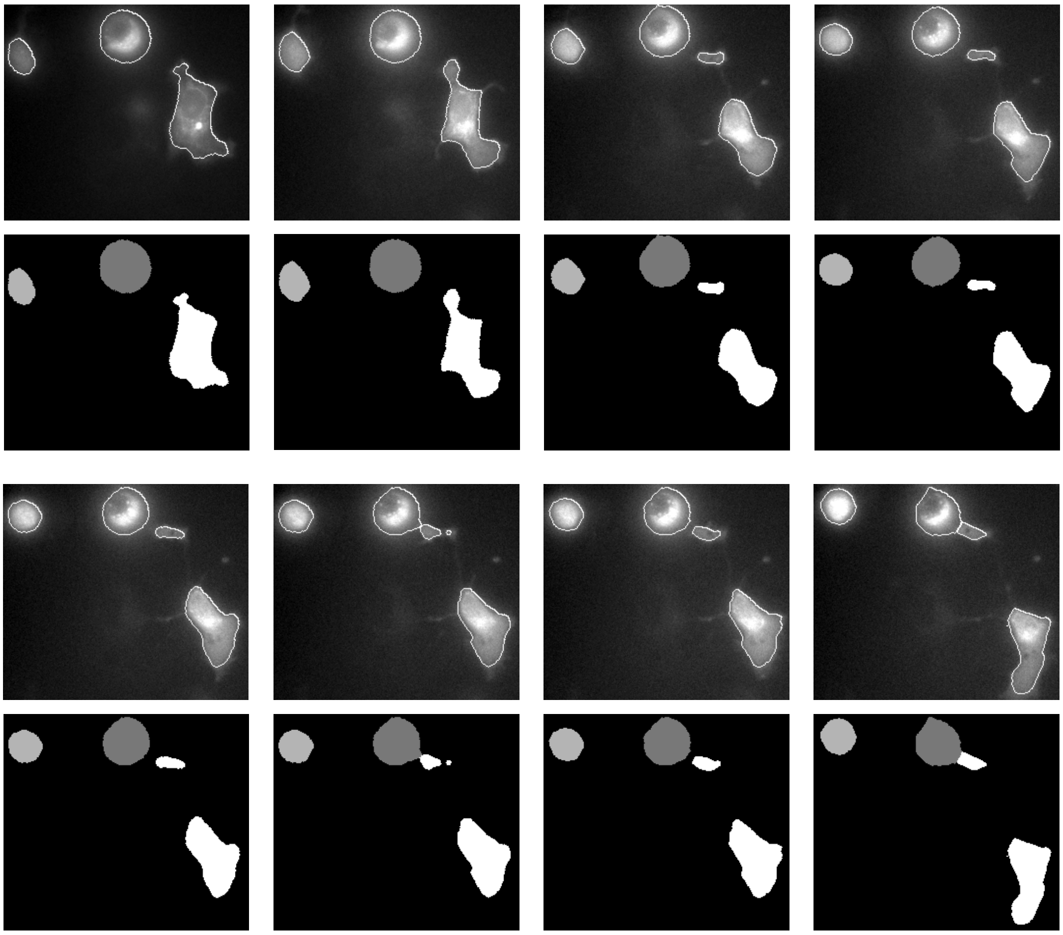

Figure 5 Tracking of AIF-transfected living cells using the RI algorithm ($\mu = 0.3$, $\lambda_1 = 1$, $\lambda_2 = 2$). The frames number 1, 3, 7, 10, 13, 14, 15, and 22 are shown. Top rows: Original image data overlaid with final contours. Bottom rows: Segmentation results of the RI algorithm.

References

1 D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

2 O. Alexandrov and F. Santosa. A topology-preserving level set method for shape optimization. *Journal of Computational Physics*, 204(1):121–130, 2005.

3 P. Brigger, J. Hoeg, and M. Unser. B-spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, 2000.

4 V. Caselles, F. Catté, T. Coll, and F. Dibos. A geometric model for active contours in image processing. *Numerische Mathematik*, 66(1):1–31, 1993.

5 V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision*, 22(1):61–79, 1997.

6 T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

7 J. Deng and H. T. Tsui. A fast level set method for segmentation of low contrast noisy biomedical images. *Pattern Recognition Letters*, 23(1-3):161–169, 2002.

8 R. Goldenberg, R. Kimmel, E. Rivlin, and M. Rudzsky. Fast geodesic active contours. *IEEE Transactions on Image Processing*, 10(10):1467–1475, 2001.

9 X. Han, C. Xu, and J. L. Prince. A topology preserving level set method for geometric deformable models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):755–768, 2003.

10 M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

11 C. Le Guyader and L. A. Vese. Self-repelling snakes for topology-preserving segmentation models. *IEEE Transactions on Image Processing*, 17(5):767–779, 2008.

12 J. Lie, M. Lysaker, and X. C. Tai. A binary level set model and some applications to Mumford-Shah image segmentation. *IEEE Transactions on Image Processing*, 15(5):1171–1181, 2006.

13 G. Malandain and G. Bertrand. Fast characterization of 3d simple points. In *Proceedings of 11th International Conference on Pattern Recognition*, pages 232–235, 1992.

14 M. Maška and P. Matula. A fast level set-like algorithm with topology preserving constraint. In *Proceedings of the 13th International Conference on Computer Analysis of Images and Patterns*, LNCS 5702, pages 930–938. Springer-Verlag, 2009.

15 M. Maška, P. Matula, O. Daněk, and M. Kozubek. A fast level set-like algorithm for region-based active contours. In *Proceedings of the 6th International Symposium on Visual Computing*, LNCS 6455, pages 387–396. Springer-Verlag, 2010.

16 D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.

17 B. Nilsson and A. Heyden. A fast algorithm for level set-like active contours. *Pattern Recognition Letters*, 24(9-10):1331–1337, 2003.

18 S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer-Verlag New York, Inc., Secaucus, USA, 2003.

19 S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulation. *Journal of Computational Physics*, 79(1):12–49, 1988.

20 J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

21 Y. Shi and W. C. Carl. Real-time tracking using level sets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 34–41, 2005.

22 Y. Shi and W. C. Karl. A real-time algorithm for the approximation of level-set-based curve evolution. *IEEE Transactions on Image Processing*, 17(5):645–656, 2008.

23 A. Tsai, A. Yezzi, and A. S. Willsky. Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. *IEEE Transactions on Image Processing*, 10(8):1169–1186, 2001.

24 X. F. Wang, D. S. Huang, and H. Xu. An efficient local chan-vese model for image segmentation. *Pattern Recognition*, 43(3):603–618, 2010.

25 R. T. Whitaker. A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.