

Segmentation and Shape Tracking of Whole Fluorescent Cells Based on the Chan–Vese Model

Martin Maška*, Ondřej Daněk, Saray Garasa, Ana Rouzaut, Arrate Muñoz-Barrutia, and Carlos Ortiz-de-Solorzano

Abstract—We present a fast and robust approach to tracking the evolving shape of whole fluorescent cells in time-lapse series. The proposed tracking scheme involves two steps. First, coherence-enhancing diffusion filtering is applied on each frame to reduce the amount of noise and enhance flow-like structures. Second, the cell boundaries are detected by minimizing the Chan–Vese model in the fast level set-like and graph cut frameworks. To allow simultaneous tracking of multiple cells over time, both frameworks have been integrated with a topological prior exploiting the object indication function. The potential of the proposed tracking scheme and the advantages and disadvantages of both frameworks are demonstrated on 2-D and 3-D time-lapse series of rat adipose-derived mesenchymal stem cells and human lung squamous cell carcinoma cells, respectively.

Index Terms—Cell tracking, Chan–Vese model, fluorescence microscopy, graph cut optimization, level set framework.

I. INTRODUCTION

UNDERSTANDING the mechanisms of cell motility and their regulation is an important challenge in biomedical research [1]. The ability of cells to exert forces on their environment and alter their shape as they move is essential to various biological processes such as the immune response, embryonic development, or tumorigenesis [2]. Recent technological advances in confocal fluorescence microscopy have given researchers the

opportunity to investigate these complex processes *in vivo* [3], [4]. However, they also lead to a tremendous increase in the amount of image data acquired during the studies. Therefore, the analysis of time-lapse experiments relies increasingly on automated image processing techniques. Namely, there is a high demand for fast and robust methods to help biologists to quantitatively analyze time-lapse image data. The crucial tasks are, in particular, segmenting, tracking, and evaluating movement tracks and morphological changes of cells, subcellular components, and other particles [5].

The typical fluorescence microscopy time-lapse series contain cells with significant spatio-temporal changes in intensity levels due to nonhomogeneous staining, uneven illumination, and photobleaching. Furthermore, these images are corrupted considerably by photon shot noise [4] and distorted by the spatially-variant point spread function of an optical system [6].

To cope with these difficulties, sophisticated tracking approaches have been proposed during the last decade. Although cell motility and intracellular flows can be analyzed using optic flow [7] or image registration [8] techniques, the scope of this paper aims at identifying the boundaries of individual cells in each frame and tracking their evolution over time. Approaches developed for this specific task can broadly be classified [9] as either *tracking by detection* [10]–[13] or *tracking by model evolution* [9], [14]–[21].

The idea of the former is to detect first all cells in the entire time-lapse series independently based on intensity [11], [12], gradient features [10], or wavelet decomposition [13], and then associate the detected cells between successive frames, typically by optimizing a probabilistic objective function. The tracking by detection approach is well suited to experiments with low cell density, for which it is easy to determine the exact number of interacting cells in the current frame. Nevertheless, with increasing cell density, the temporal association step often requires sophisticated strategies to deal with one-to-many and many-to-one matching problems effectively [12], [13], [22], [23]. Additionally, no information from the previous frames is usually considered during the detection step, which complicates proper separation of multiple touching cells [10].

The principle of the latter is to find cells in the first frame and update their position, shape, and orientation through the entire time-lapse series, taking the result of each frame as the initial condition for the following one. Each cell to be tracked is represented by a model that is evolved in time to fit the particular cell in the subsequent frames. When the precise cell boundaries are not required, the mean-shift processes can be applied to track cell centroids [14]. However, active contours, based on either an explicit [21], [24]–[27] or implicit [28]–[30] formulation,

Manuscript received October 22, 2012; revised January 09, 2013; accepted January 17, 2013. Date of publication January 29, 2013; date of current version May 29, 2013. This work was funded by the UTE Project CIMA, the Spanish Science and Innovation Ministry (MICINN), and the Spanish Ministry of Economy and Competitiveness under grants PSS-010000-2008-2, DPI2009-14115-C03-03, PI10/02131, and DPI2012-38090-C03-02. M. Maška was financed from the European Social Fund and the Czech Ministry of Education (Project No. CZ.1.07/2.3.00/30.0009). S. Garasa and A. Rouzaut were supported by the European Network for Cell Imaging and Tracking Expertise under grant EUS2008-03592. A. Muñoz-Barrutia and C. Ortiz-de-Solórzano hold a Ramón y Cajal Fellowship of the MICINN. Asterisk indicates corresponding author.

*M. Maška was with the Cancer Imaging Laboratory, Oncology Division, Center for Applied Medical Research, University of Navarra, 31008 Pamplona, Spain. He is now with the Centre for Biomedical Image Analysis, Faculty of Informatics, Masaryk University, 60200 Brno, Czech Republic (e-mail: xmaska@fi.muni.cz).

O. Daněk was with the Centre for Biomedical Image Analysis, Faculty of Informatics, Masaryk University, 60200 Brno, Czech Republic.

S. Garasa and A. Rouzaut are with the Laboratory of Tumor Microenvironment, Oncology Division, Center for Applied Medical Research, University of Navarra, 31008 Pamplona, Spain.

A. Muñoz-Barrutia and C. Ortiz-de-Solórzano are with the Cancer Imaging Laboratory, Oncology Division, Center for Applied Medical Research, University of Navarra, 31008 Pamplona, Spain.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TMI.2013.2243463

have become the first choice for the tracking by model evolution approach in recent years [15]–[21], [26]. The main reason is that they are especially appropriate to detect cells of complex boundaries. Furthermore, they can adapt easily to their temporal changes and incorporate prior information reflecting biological knowledge of observed specimens, such as size, shape, and topological constraints.

Among the tracking by model evolution approaches, the algorithm based on coupled implicit active contours governed by the modified Chan–Vese model, proposed originally by Dufour *et al.* [18] and improved by Dzyubachyk *et al.* [20], is the most elegant approach for simultaneous tracking of multiple cells currently available. However, the representation of each cell by one level set function evolved via a numerical solution of a partial differential equation (PDE) brings long execution times, even when the narrow band technique [31] is employed. Recently, Dufour *et al.* [21] reformulated this approach in the discrete active mesh framework to reduce its computational costs. Nonetheless, the line-to-mesh rasterization scheme, necessary to determine a mutual position between each image grid point and an active mesh (i.e., the inner-outer test), and the gradual collision detection procedure, evaluating a level of penetration of each vertex in the opposite mesh, still pose a high computational burden. Furthermore, the active mesh framework only works in 3-D, and cannot straightforwardly be used to analyze 2-D time-lapse series.

In this paper, we propose a different tracking by model evolution approach aimed at making the algorithm as fast as possible and directly applicable to 2-D as well as 3-D time-lapse series. In contrast to [18], [20], [21], we minimize the original Chan–Vese model [30] in the fast level set-like (FLS) [32] and graph cut (GC) [33] frameworks without solving any PDE, and replace coupling of multiple separate models with a topological prior that exploits the object indication function [34] to allow simultaneous tracking of multiple cells over time. Furthermore, to reduce the amount of noise in the acquired image data, increase the contrast along the cell boundaries, and enhance slender fiber-like structures often connecting individual regions of the cell cytoplasm, the tracking step is preceded by coherence-enhancing diffusion filtering (CED) [35].

A 2-D level-set-based tracking by model evolution approach integrated with a labeling function to allow simultaneous tracking of multiple targets was also proposed by Li *et al.* [9]. Comparing to that, our approach provides the following advantages. First, its application is not limited only to 2-D ellipse-like targets of similar radius. It can directly be applied on cells of various sizes and complex shapes in 2-D as well as 3-D. Second, label propagation does not require any topological numbers to be calculated, which speeds up the minimization process, especially in 3-D [34]. Finally, the regularization and data terms are minimized simultaneously, which prevents a data-term-driven evolution from being canceled or partially canceled by a separate regularization cycle [36], thus allowing more accurate detection of complex, rugged boundaries.

This work is a considerably extended version of the whole-cell tracking scheme previously presented at a conference [37]. Specifically, besides the FLS framework, the original approach is also integrated with graph cuts, another topologically flexible,

directly extendable into higher dimensions, and easy-to-implement framework used to minimize the Chan–Vese model. To the best of our knowledge, these frameworks have not been compared yet in the context of whole-cell segmentation and tracking of fluorescence microscopy time-lapse series. This motivated us to involve both of them in the proposed tracking scheme and compare their accuracy, execution time, and memory consumption. We also provide an extensive validation of the proposed tracking scheme on 2-D as well as 3-D time-lapse series. Although it can be applied to various cell types, the experimental evaluation, including a thorough comparison of the FLS and GC frameworks, is performed on 2-D and 3-D time-lapse series of rat adipose-derived mesenchymal stem cells (ADMSCs) and human lung squamous cell carcinoma cells (H157), respectively. We demonstrate that the proposed tracking scheme is more accurate and significantly faster than the other state-of-the-art tracking by model evolution approaches.

The organization of the paper is as follows. A brief review of the Chan–Vese model and its minimization is given in Section II. The time-lapse datasets analyzed in our experiments are introduced in Section III. Section IV describes in detail the proposed tracking scheme. Its experimental evaluation is given in Section V. Finally, the paper ends with conclusions and suggestions for future work in Section VI.

II. CHAN–VESE MODEL AND ITS MINIMIZATION

This section recalls shortly the theoretical background and formulation of the Chan–Vese model and the basic principles of the state-of-the-art frameworks developed for its minimization.

The Chan–Vese model [30] is a piecewise constant approximation to the functional formulation of image segmentation introduced by Mumford and Shah [38]. It has become popular in the image processing community mainly due to its ability to detect objects not necessarily defined by a gradient. Its aim is to partition an input scalar image $u_0 : \Omega \rightarrow \mathbb{R}$ defined over a d -dimensional image domain $\Omega \subset \mathbb{R}^d$ into two possibly disconnected regions Ω_1 (foreground) and Ω_2 (background) of low intra-region variance and separated by a smooth closed contour C ($\Omega = \Omega_1 \cup \Omega_2 \cup C$). It can be formulated as

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

where c_1 and c_2 denote the unknown average intensity levels inside Ω_1 and Ω_2 , respectively, and μ , λ_1 , and λ_2 are positive, user-defined weights. Hereinafter, we refer to the first term as the *regularization* term and to the others as *fidelity* terms. The optimal segmentation (C, c_1, c_2) corresponds to a global minimum of (1). The minimization is usually accomplished in three different frameworks: *explicit active contours* [24]–[26], *level sets* [39], [40], and *graph cuts* [41], [42].

Although the first two approaches represent the contour C differently, they are both based on a force-driven evolution of an initial contour. The explicit active contours are generally faster than level sets due to their light parametric representation of the contour C . However, their topological rigidity rules

out the detection of multiple isolated objects from a single initial contour unless special surgical routines are used [43], [44]. Since each explicit contour evolves independently of others, multiple contours need to be coupled [26] to share the global statistics involved in (1), and not to overlap with the others. On the contrary, the level set framework expresses the contour C implicitly as the zero level set of a higher-dimensional function. Such representation provides inherent topological flexibility and direct extension into higher dimensions, but is also significantly time-consuming. Therefore, fast approximations based on weighted two-means clustering [45] or threshold dynamics [46] as well as non-PDE-based contour evolution algorithms [32], [47], [48] have been proposed over the last decade to reduce that computational burden.

The minimization of the Chan–Vese model in the graph cut framework [33] is more straightforward than in the previous ones. It is based on the fact that for fixed c_1 and c_2 it is possible to construct a graph G in which the cost of every cut approximates the energy of the corresponding segmentation in (1). Therefore, a minimum cut of the graph G corresponds to a global minimum of (1) for fixed c_1 and c_2 . Subsequently, these values are updated according to the minimum cut and the computation is repeated until they reach a steady state. In contrast to the previous frameworks, graph cuts do not require any initial contour to be specified. They only need some initial estimates of c_1 and c_2 . However, discrete graph cuts suffer from metrication errors and produce angular boundaries. It was shown [49] that the denser the neighborhood system is when constructing the graph G , the smoother are the obtained boundaries. Unfortunately, increasing neighborhood density results in higher memory consumption, which could be critical, in particular, when processing large volumetric images.

For the sake of completeness, we refer the interested reader to the convex optimization framework [50], [51], in which the nonconvex Chan–Vese model is reformulated as a convex minimization problem and solved directly using fast iterative methods. However, its use for simultaneous tracking of multiple objects is limited since the topological prior exploiting the object indication function [34] is based on a local neighborhood of the contour C , and can barely be incorporated into the energy functional to be convexified and minimized.

III. INPUT IMAGE DATA

This section introduces two different time-lapse datasets of GFP-transfected cells analyzed in our experiments (Section V). All images were acquired using spinning-disk confocal microscopes. A summary of the cell types, acquisition setups, and image data properties is given in Table I. Sample frames from each of the datasets are shown in Fig. 1.

The first dataset consisted of five 2-D time-lapse series of rat adipose-derived mesenchymal stem cells suspended in Minimum Essential Medium α (Gibco, USA), which was supplemented with 10% Fetal Calf Serum (Biochrom AG, Berlin, Germany) and 1% Penicillin/Streptomycin (Lonza, Switzerland), and monitored in a mixture of 1 $\mu\text{g}/\mu\text{L}$ collagen and 0.25% agarose poured on confocal Glass Bottom Microwell Dishes (MatTek, Ashland, MA, USA). Each 2-D time-lapse series had

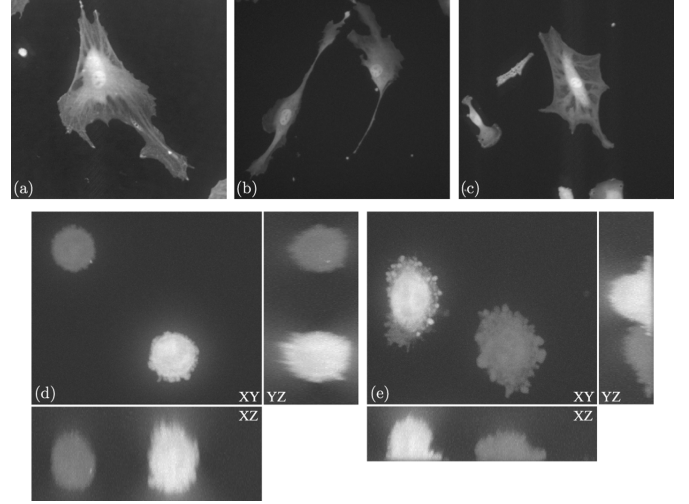


Fig. 1. Examples of analyzed input image data. (a)–(c) Three frames of different ADMSCs. (d) Maximum intensity projections along each axis of H157 cells poured on a confocal glass bottom microwell. (e) Maximum intensity projections along each axis of H157 cells injected in a growth medium.

TABLE I
CELL TYPE, ACQUISITION SETUP, AND PROPERTIES OF TWO TIME-LAPSE DATASETS ANALYZED IN OUR EXPERIMENTS

	Dataset 1	Dataset 2
Cell type	ADMSC	H157
Imaging system	Zeiss Cell Observer	PerkinElmer Ultraview ERS
Objective lens	Plan-Apo 20 \times /0.85	Plan-Apo 63 \times /1.20 Water
Frame size (vox)	512 \times 478	580 \times 540 \times 60 (40, 35)
Voxel size (nm)	650 \times 650	125.7 \times 125.7 \times 500.0
Time step (min)	5	1 (2)
No. of frames	19	20
No. of series	5	3 (4)

19 frames captured every 5 min. The acquired 16-bit image data was compensated for uneven illumination by applying a flat-field correction procedure. The time-lapse series contained from one up to five cells, 13 cells in total, having a bright nucleus within a lower intensity, heterogeneous cytoplasm of a complex shape, various sizes, and often containing elongated thin structures connecting its individual regions [Fig. 1(a)–(c)].

The second dataset consisted of seven 3-D time-lapse series of human lung squamous cell carcinoma cells suspended in a 1:1 mixture of Matrigel/Collagen (BD Biosciences, San Jose, CA, USA). Each 3-D time-lapse series had 20 frames that were captured every minute (three time-lapse series) or two (four time-lapse series). The acquired 16-bit image data contained one or two cells, 11 cells in total, having slightly heterogeneous cytoplasm of different intensity levels. The H157 cell cytoplasm shape varied depending on the environment where the H157 cells were placed. Whereas the H157 cells poured on confocal Glass Bottom Microwell Dishes (MatTek) previously covered with a thin layer of collagen were of almost ellipsoidal shapes [six cells; Fig. 1(d)], they exhibited more complex shapes [five cells; Fig. 1(e)] after being injected in a complete growth medium consisted of a human lymphatic endothelial cell monolayer (Promocell, Barcelona, Spain) seeded on collagen-coated confocal Glass Bottom Microwell Dishes (MatTek).

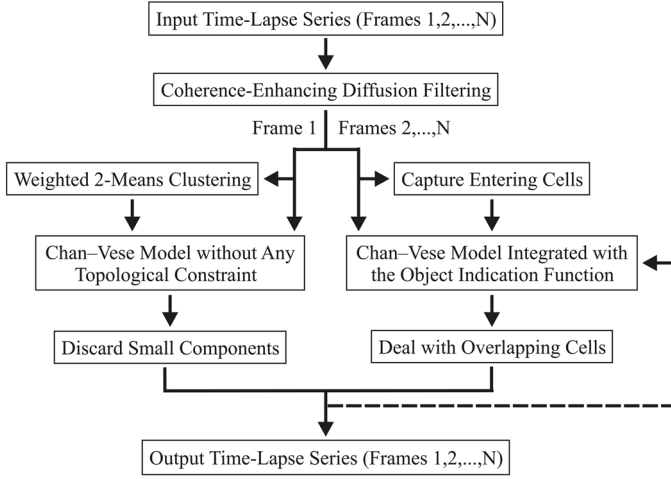


Fig. 2. Workflow of the proposed tracking scheme. The dashed line indicates using of a final result for the current frame as an additional information for the Chan-Vese model minimization routines when processing the next frame.

IV. PROPOSED TRACKING SCHEME

In this section, we describe in detail the proposed tracking scheme, which involves two steps. First, a coherence-enhancing diffusion filter [35] is applied on each frame to reduce the amount of noise and enhance flow-like structures. Second, the cell boundaries are detected by minimizing the Chan-Vese model in the FLS and GC frameworks. The initial models established for the cells detected in the first frame are evolved in time to fit the corresponding cells in the subsequent frames. Furthermore, to allow tracking of cells not appearing in the first frame, each cell entering the field of view is assigned a new model that is evolved simultaneously with the initial ones. To handle touching and dividing cells over time, both frameworks have been integrated with a topological prior that exploits the object indication function [34]. The workflow of the proposed tracking scheme is depicted in Fig. 2. In the rest of this section, we further describe its individual steps and explain how to address entering and slightly overlapping cells.

A. Coherence-Enhancing Diffusion Filtering

Simple median and Gaussian filters are not very suitable for preprocessing our image data because they smooth out small and thin structures, thus precluding the detection of fiber-like structures connecting individual regions of the cell cytoplasm (Fig. 3). A more robust filtering approach is the coherence-enhancing diffusion filter [35], a mathematically well-founded method based on the solution of a nonlinear anisotropic diffusion equation. This spatial filter can enhance flow-like structures while reducing the amount of noise. From the implementation point of view, the continuous diffusion process is discretized using finite difference schemes and solved iteratively. Our implementation is based on the semi-implicit scheme stabilized by an additive operator splitting [35]. This scheme can be easily extended into higher dimensions and allows the use of longer time steps compared to explicit schemes, which results in a lower number of iterations and faster computation.

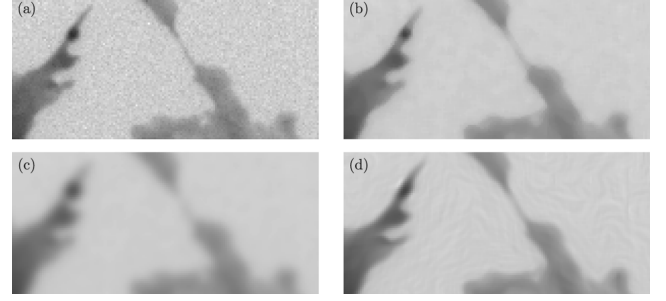


Fig. 3. Filtering effect on a thin fiber-like structure. (a) Inverted original image data. (b) Median filter for a square window ($r = 2$). (c) Gaussian filter for $\sigma = 2$. (d) CED filter for $\sigma = 1$, $\rho = 3$, and $\Delta t = 1.25$ after four iterations.

B. First Frame Segmentation

Since the number of cells in the first frame is not known in advance, the cell boundaries are detected using the Chan-Vese model without any topological constraint. Both frameworks are initialized using a fully automated weighted two-means clustering [45] that corresponds to the minimization of (1) without the regularization term. The FLS framework takes the final contour provided by the clustering, whereas graph cuts start with the corresponding foreground and background statistics. Once a steady state is reached, small components enclosing foreign particles such as dust are discarded from the final binary mask. The remaining ones are established as the cells to be tracked. The criterion for discarding a component or not compares its size in grid points with a user-defined threshold s_r that reflects *a priori* rough estimate of the size of tracked cells.

C. Object Indication Function

One of the main aims of every tracking algorithm is to keep the identity of each tracked target over time. Since the FLS and GC frameworks are both topologically flexible, a simple binary separation of the image domain might yield undesired results. For instance, when two isolated cells in one frame form a cluster in the next one, it is necessary to prevent their contours from merging, thus preserving the cell identities. In addition, to construct a cell lineage tree, the information about relations between mother and daughter cells needs to be maintained.

To fulfill these requirements, the FLS and GC frameworks have been integrated with a topological prior exploiting the object indication function $\psi : \Omega \rightarrow \{0, 1, 2, \dots\}$ [34] that assigns the zero label to the background grid points and a positive label to the foreground ones. It assumes that the image domain Ω is divided into a possibly disconnected background region Ψ_0 and M possibly disconnected disjoint foreground objects $\Psi_1, \Psi_2, \dots, \Psi_M$ ($\Omega = \bigcup_{0 \leq i \leq M} \Psi_i$). Each foreground object Ψ_i can split into several regions that can merge again later in time, whereas the regions of different object indicators cannot merge together. By assigning different positive labels to individual cells, two previously isolated cells are distinguished when they touch. Furthermore, the object indication function allows each foreground object to adapt freely to topological changes of the corresponding cell [34].

The topological prior exploiting the object indication function is rather simple. The object indication function is evolved

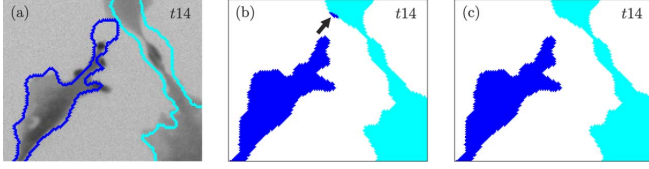


Fig. 4. Tracking of two overlapping cells in the FLS framework. (a) Inverted original image data overlaid with the thick final contours from the previous frame. (b), (c) Segmentation results for $w_o = 0$ and $w_o = 10$.

simultaneously with deforming contours. When a contour deflates at a grid point $p \in \Omega$, $\psi(p) > 0$, the object indicator $\psi(p)$ is set to zero. On the contrary, the contour is propagated, including its object indicator, only to those neighboring grid points that are assigned the zero object indicator.

This concept can be easily incorporated into the local propagation of the interface points in the case of the fast level set-like algorithm [34] or into the local construction of search trees in the case of graph cuts. In particular, in the latter, the object indication function is integrated into the static maximum flow algorithm [41] by maintaining four search trees instead of the original two. They are constructed in the same way as those in our previous work [52], in which the Boykov–Kolmogorov algorithm was integrated with the simple point concept from digital geometry [53], with two simple modifications. First, each graph node stores an object indicator that is propagated to its descendants in the tree growth stage. Second, all conditions verifying whether a particular grid point is simple or not are omitted since the object indication function allows each object to change freely its topology. For further technical details, we refer the interested reader to the original papers [34], [52].

Note that the Boykov–Kolmogorov algorithm is not the most efficient algorithm for finding a minimum cut. It is outperformed by the dynamic maximum flow algorithm by Kohli and Torr [54] allowing fast and efficient updating of residual graphs between two successive iterations. However, this algorithm is not well suited to incorporate the object indication function since the dynamic update of the residual graphs may change completely the order of processed grid points in the search tree construction needed for a balanced propagation of object indicators [52]. Therefore, we apply the Kohli–Torr algorithm to analyze the first frame for which no topological prior is involved, whereas the modified Boykov–Kolmogorov algorithm is used for the others.

D. Overlapping Cells

By *overlapping cells*, we mean situations when the same spatial position is occupied by different cells in two successive frames. In those cases, the initial seed of one cell might expand within the overlapped cell, thus yielding undesired results [Fig. 4(b)]. Let w_o denote the diameter of the overlap. Clearly, the overlap size is bounded by w_o^d , where d is the image domain dimension. If w_o is only a few grid points, the initial seed of one cell can be forced out of the overlapped cell by strengthening the regularization term in (1). Unfortunately, stronger regularization also breaks thin fiber-like structures of the cell cytoplasm. We propose, therefore, an alternative strategy that involves expanding the initial seed in the overlapped cell and

changing the object indicators upon convergence. Given a component m_1 of the object indicator l_1 and of the size smaller than w_o^d , which touches a component m_2 of a different object indicator l_2 [Fig. 4(b)], its object indicator is changed from l_1 to l_2 [Fig. 4(c)]. Note that this solution is applicable only to small overlaps, when cell displacements between two successive frames are not too large.

E. Capturing Entering Cells

New cells entering the field of view must be paid special attention. They have not been detected in the previous frames and are, therefore, considered part of the image background. This can affect significantly the background statistics and lead to incorrect segmentation of a particular frame.

To deal with entering cells, Dzyubachyk *et al.* [20] proposed the following strategy. After the current frame has been segmented without taking entering cells into consideration, a non-PDE-based segmentation approach [47] is applied to capture them. They are selected as border components disjoint with the existing cells. For each entering cell, a new level set function is created. Next, all the new level set functions are evolved. However, including the entering cells in the minimization process can change the background statistics. If the already converged contours of existing cells had also evolved, they would have attained slightly different positions. To obtain more accurate results, we propose instead to add the entering cells before segmenting the current frame, choosing them from a binary mask obtained using the weighted two-means clustering [45].

F. Stopping Criterion

So far, we have assumed that the minimization of the Chan–Vese model stops when a steady state is reached. Specifying a suitable stopping criterion is an essential component of every iterative algorithm since determining the exact moment when the algorithm converges avoids unnecessary computations. Although a fixed number of iterations is usually the first choice, in the case of the Chan–Vese model it is profitable to establish a stopping rule based on the values of c_1 and c_2 .

Specifically, the stopping criterion for the FLS framework is based on the detection of an oscillatory cycle. Let Δ^i denote the sum $|c_1^{i+1} - c_1^i| + |c_2^{i+1} - c_2^i|$ for two successive iterations i and $i + 1$, $i \geq 0$. Let the contour C approach a cell boundary (i.e., a local minimum) in iteration $k \geq 0$. The contour C oscillates iff

$$|\Delta^{k+1} - \Delta^k| = 0. \quad (2)$$

To demonstrate this, let $f : \Omega \rightarrow \{I_1, I_2\}$, $I_2 < I_1$, be, without loss of generality, a one-dimensional bilevel image of an ideal edge. For simplicity, let $\mu = 0$ and $\lambda_1 = \lambda_2 = 1$, thus an interface point $x \in \Omega$ is locally propagated inside or outside the contour C depending on the sign of the speed function $F : \Omega \rightarrow \mathbb{R}$ [32], given as

$$F(x) = (f(x) - c_2)^2 - (f(x) - c_1)^2. \quad (3)$$

Let $a, b \in \Omega$, $f(a) = I_1$, $f(b) = I_2$, $b = a + \Delta x$, be adjacent grid points reached by C in two successive iterations (Fig. 5). Let $c_{i,x}$, $i \in \{1, 2\}$, $x \in \Omega$, denote the foreground and background statistics when the contour C reaches the grid point x .

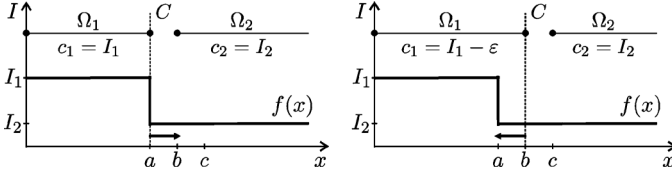


Fig. 5. The behavior of the contour C oscillating between two adjacent grid points a and b within a one-dimensional bilevel image $f(x)$. The thin lines with balls delimit the regions Ω_1 and Ω_2 . The arrows show the direction of the contour evolution. Note that $0 < \varepsilon < I_1 - I_2$.

Obviously, $c_{1,a} = I_1$, $c_{2,a} = I_2$, and $F(a) = (I_1 - I_2)^2 > 0$ hold when C is at the position a . Analogously, $c_{1,b} = I_1 - \varepsilon$, $0 < \varepsilon < I_1 - I_2$, $c_{2,b} = I_2$, and $F(b) = -(I_2 - I_1 + \varepsilon)^2 < 0$ are valid at the position b . Since the contour C attains the position a in iteration k and since $F(a)$ and $F(b)$ have opposite signs, C starts oscillating between the positions a and b . Certainly, $c_i^{k+j} = c_{i,a}$, $c_i^{k+j+1} = c_{i,b}$, $i \in \{1, 2\}$, hold for $j = 0, 2, \dots$, which yields $\Delta^{k+m} = \varepsilon$ for $m = 0, 1, \dots$, and gives (2).

In the GC framework, the minimization of (1) iterates until the condition $\Delta^i \leq K$, $i \geq 0$, is satisfied. The tolerance K can be, in fact, set to zero and the computation stops when a steady state is reached, assuming that a deterministic maximum flow algorithm is employed. Nevertheless, K may be increased to speed up the computation as the final segmentation often does not change much during the last iterations. In practice, the choice $K = 0.001$ results in only a few iterations needed to minimize the Chan–Vese model.

V. EXPERIMENTAL RESULTS

This section gives the experimental evaluation of the proposed tracking scheme conducted on a common workstation (Intel Core i7-880 3.07 GHz, 3 GB RAM, Windows XP Professional). We focus first on the optimal parameter settings for each framework. Then, we compare thoroughly their accuracy, execution time, and memory consumption. Finally, we compare their performance with that of the tracking algorithm by Dzyubachyk *et al.* (DZY) [20].

A. Parameter Settings

The proposed tracking scheme has nine parameters that influence its overall performance. They can be divided into three groups according to their meaning and purpose.

First, the parameters s_r and w_o reflect quantitative properties of tracked cells. They can be easily derived from the visual inspection of analyzed time-lapse series. The parameter s_r was fixed at 400 and 24 000 grid points in 2-D and 3-D, respectively. These values corresponded roughly to 20% of the minimum average size of a complete cell (i.e., a cell not touching the image border in the first frame) and allowed us to detect and track cells that appear only partly in the field of view. Furthermore, the occurrence of overlapping cells was very sporadic in our time-lapse datasets. When some appeared, their overlap was only in the order of a few grid points in diameter. Therefore, the parameter w_o was fixed at 10 grid points.

The next group of parameters characterizes the coherence-enhancing diffusion filter used in the first step of the proposed

tracking scheme. Since its main task is to reduce the amount of noise and preserve thin fiber-like structures, only a few iterations are needed. We performed four iterations with fixed time step $\Delta t = 1.25$, noise scale $\sigma = 1$, and integration scale $\rho = 3$ in all experiments. The choice of σ and ρ reflects the width of the thin fiber-like structures to be preserved.

The typical behavior of a CED filter is illustrated in Fig. 6(a) and (b). The former shows the profiles of normalized differences of the standard deviations of intensity levels within five cells from different time-lapse series in two successive iterations. The later shows the contrast of a cell boundary measured as the average gradient magnitude along that boundary. Note that a fall in the contrast measure after the first iteration and its temporal rise in the subsequent ones [Fig. 6(b)] are induced by the combination of smoothing and enhancing effects of the CED filtering. As shown in Fig. 6(a), the dominating photon shot noise is primarily reduced during the first few iterations. Simultaneously, the coherence-oriented diffusion process enhances flow-like patterns, including the cell boundaries. Consequently, the contrast measure reaches its local maximum for a number of iterations between 3 and 7, not considering the measure for the original noisy image data. The maximum contrast averaged over all five cells was obtained after four iterations. With the increasing number of iterations, the diffusion effect abates and is outstripped again by a permanent smoothing that guides the diffusion process to its equilibrium. It corresponds to a constant image with the intensity level being equal to the average one of the original image data.

Finally, the most important parameters are the weights of the regularization and fidelity terms of the Chan–Vese model. In order to find the optimal values of μ , λ_1 , and λ_2 , for which the proposed tracking scheme achieves the highest accuracy, two studies were performed. Since the analyzed datasets were acquired using different imaging systems, having completely different intensity distributions, it was necessary to find the optimal parameters separately for each of them. Without loss of generality, one can set $\lambda_1 = 1$ since the minimizer of (1) remains unchanged if E_{CV} is replaced with E_{CV}/λ_1 .

In 2-D, we examined 3050 different parameter configurations: $\mu = 0.01, 0.02, \dots, 0.50$ and $\lambda_2 = 1, 5, 10, \dots, 300$. All 2-D time-lapse series were manually segmented by an expert biologist and taken as a ground truth. The Jaccard coefficient JC given, for a ground truth X and segmentation result Y , as $JC = |X \cap Y|/|X \cup Y|$ was used as the accuracy measure. It was evaluated separately for each of eight complete cells and averaged over all 2-D time-lapse series (Fig. 7).

Since the analysis of 3-D time-lapse series is much more time-consuming than the analysis of 2-D ones, we exploited the knowledge of the accuracy map profiles in 2-D to minimize as much as possible the total number of evaluated parameter configurations in 3-D. We examined 250 different configurations: $\mu = 0.01, 0.02, \dots, 0.25$ and $\lambda_2 = 1, 2, \dots, 10$. Whereas all 3-D time-lapse series were analyzed completely, their manual segmentation would be an extremely laborious task. Therefore, it was restricted only to one randomly selected slice per frame. In total, 140 slices were manually segmented by an expert biologist and taken as a ground truth. The Jaccard coefficient was

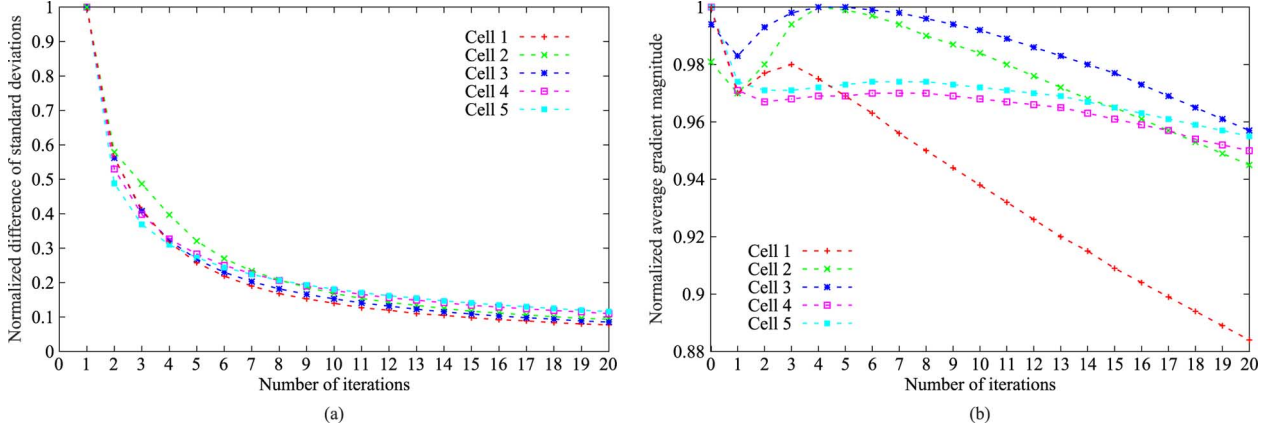


Fig. 6. Smoothing and contrast-enhancing effects of the CED filter ($\sigma = 1$, $\rho = 3$, $\Delta t = 1.25$) within five cells from different time-lapse series. (a) Normalized difference of the standard deviations of intensity levels in two successive iterations. (b) Normalized average gradient magnitude along the cell boundary.

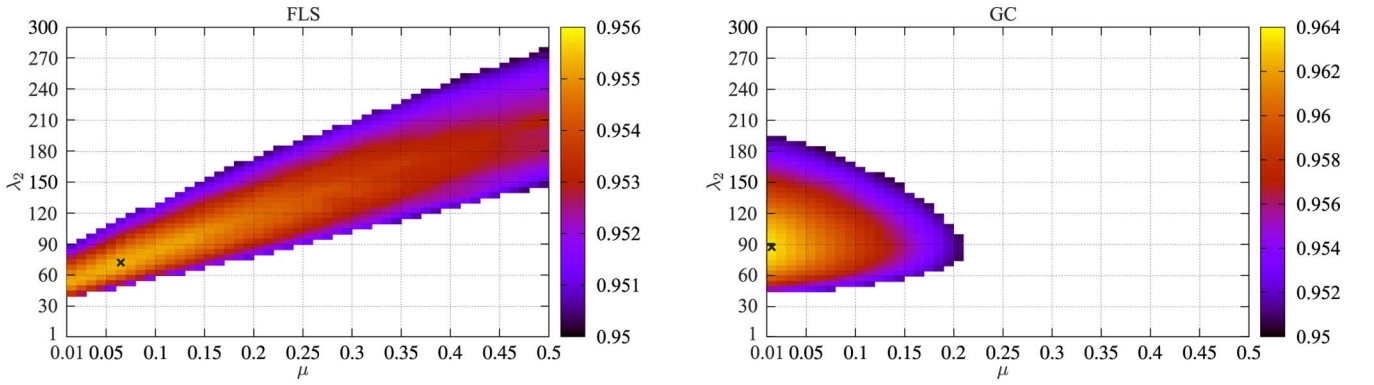


Fig. 7. Thresholded accuracy maps of the Jaccard coefficient at the level $JC_{avg} = 0.95$ for the FLS and GC frameworks in 2-D. The maxima over different parameter configurations (μ , $\lambda_1 = 1$, λ_2) of the Chan–Vese model are marked using black crosses.

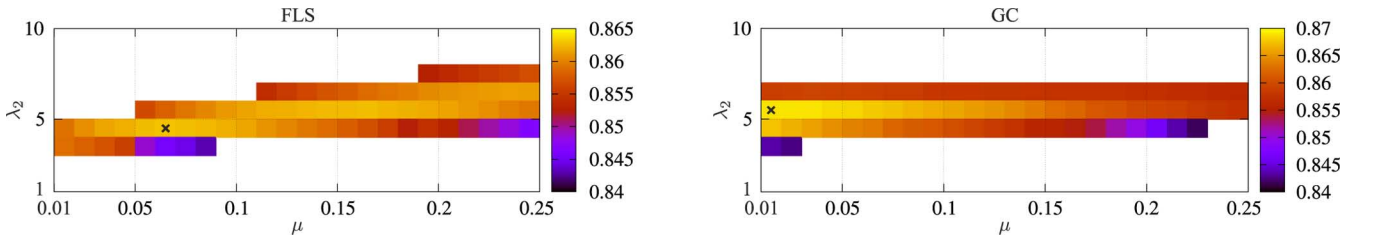


Fig. 8. Thresholded accuracy maps of the Jaccard coefficient at the level $JC_{avg} = 0.84$ for the FLS and GC frameworks in 3-D. The maxima over different parameter configurations (μ , $\lambda_1 = 1$, λ_2) of the Chan–Vese model are marked using black crosses.

evaluated separately for every cell in each of the selected slices and averaged over all of them (Fig. 8).

The parameters of DZY, used to compare with the proposed tracking scheme, were optimized to achieve the best possible results: weight of the regularization term $\alpha = 0.9$ in 2-D and $\alpha = 13$ in 3-D, number of iterations $n_i = 10$ after which the foreground and background statistics are recomputed, narrow band width $k = 3$, and time step $\Delta t = 0.1$. To obtain comparable results, DZY was applied on the preprocessed image data after coherence-enhancing diffusion filtering.

The highest average accuracy achieved by each of the compared methods and the corresponding parameter configurations are listed in Table II. Examples of 2-D and 3-D tracking results are depicted in Figs. 9 and 10, respectively.

B. Accuracy

In order to quantify the difference between the results of the tested methods, the Hausdorff distance H [53], given as

$$H(X, Y) = \max \left\{ \max_{x \in X} \min_{y \in Y} d_e(x, y), \max_{y \in Y} \min_{x \in X} d_e(y, x) \right\} \quad (4)$$

generalizing the Euclidean distance d_e on two sets of grid points (segmentation results) X and Y was measured for each pair of them. Note that the measurements were no longer restricted only to 140 randomly chosen slices in 3-D, but performed on complete 3-D stacks. The measured median and maximum Hausdorff distances are listed in Table III. It also contains p -values of a paired Student t -test performed for each

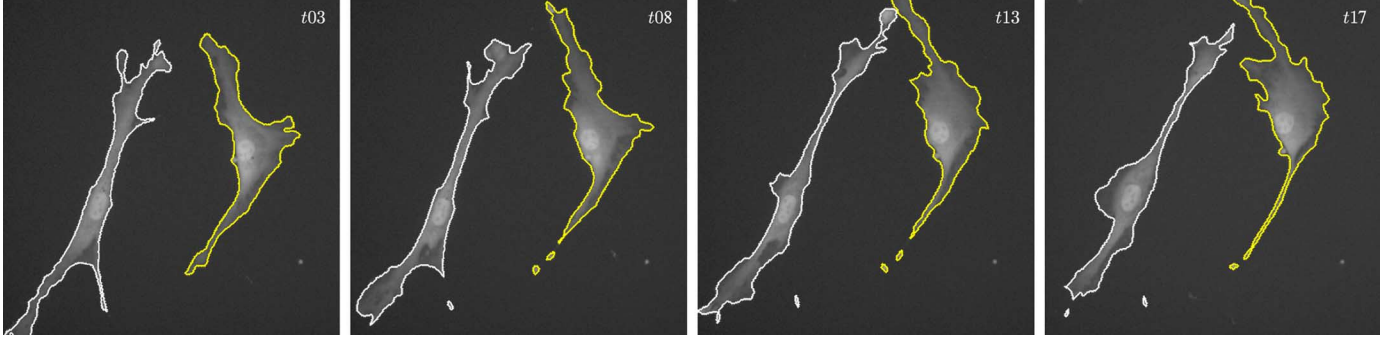


Fig. 9. Example of 2-D tracking results of the FLS framework. Original image data overlaid with thick final contours. The tracking results of the entire time-lapse series are available at <http://ieeexplore.ieee.org> as a supplementary animated GIF file of size 976 kB.

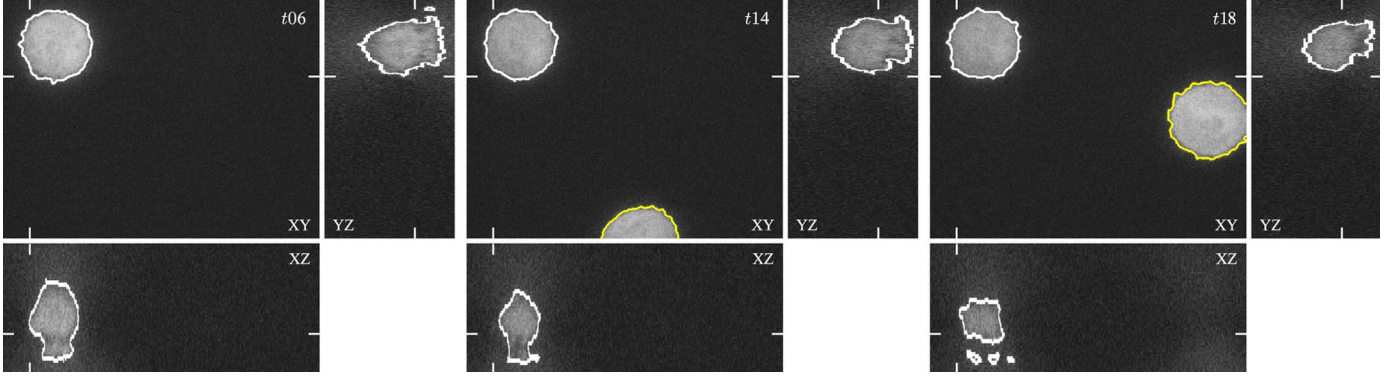


Fig. 10. Example of 3-D tracking results of the GC framework. Three orthogonal cross-sections of original image data at the position $(x, y, z) = (48, 242, 42)$ overlaid with thick final contours. Ticks show the positions of the other two cross sections. New cell entering the field of view captured in the middle frame. The tracking results of the entire time-lapse series are available at <http://ieeexplore.ieee.org> as a supplementary animated GIF file of size 2.8 MB.

TABLE II
HIGHEST AVERAGE ACCURACY JC_{avg} OF THE COMPARED METHODS
AND THE CORRESPONDING PARAMETER CONFIGURATIONS

Method	JC_{avg}		Configuration $(\mu, \lambda_1, \lambda_2)$	
	2D	3D	2D	3D
FLS	0.9555	0.8628	(0.06, 1, 70)	(0.06, 1, 4)
GC	0.9636	0.8689	(0.01, 1, 85)	(0.01, 1, 5)
DZY	0.8730	0.8193	NA	NA

TABLE III
MEDIAN (H_{med}) AND MAXIMUM (H_{max}) HAUSDORFF DISTANCE AND
 p -VALUE OF A PAIRED STUDENT t -TEST ($p < 0.05$ DENOTED BY *)
FOR EACH PAIR OF THE COMPARED METHODS

Pair	H_{med} (μm)		H_{max} (μm)		p -Value	
	2D	3D	2D	3D	2D	3D
(GC, FLS)	0.92	1.06	12.15	2.94	*	0.08
(GC, DZY)	6.90	4.84	48.85	10.79	*	*
(FLS, DZY)	6.96	4.98	49.43	10.49	*	*

pair of the tested methods in order to determine the statistical significance of the accuracy improvement. Statistically significant results ($p < 0.05$) are denoted by an asterisk. We have also performed a leave-one-out cross validation of the FLS and GC frameworks to measure an expected average accuracy for new time-lapse series from the same source. The measured values for both frameworks were about 0.6% lower than those obtained for the optimal parameter configurations (Table II).

C. Execution Time and Memory Consumption

The last experiment compared the performance of the tested methods. The proposed tracking scheme was implemented purely in C++. Nevertheless, since we had available only a precompiled Matlab interface of DZY (provided by the first author of [20]), we also developed a Matlab interface to our C++ code to allow a fair comparison of execution times. We observed no significant difference in the execution times when calling it from Matlab or directly in C++. The total execution time needed to analyze each of the tested time-lapse series is listed in Table IV. In order to avoid the memory consumption of the proposed tracking scheme being biased by the Matlab environment itself, it was measured only for the C++ implementation, performing all calculations in single (32-bit) precision, to investigate its dependence on the number of tracked cells. The measured values, decomposed into a constant memory consumed by each analyzed frame itself and a variable peak memory consumed by each framework when analyzing the tested time-lapse series, are listed in Table V.

D. Discussion

In this section, we discuss the obtained experimental results and highlight the differences between the tested methods.

In comparison to the FLS framework, graph cuts were up to about 0.8% more accurate (Table II) and had a completely different accuracy map profile (Figs. 7 and 8). The difference arose from a distinct approximation of the regularization term.

TABLE IV
TOTAL EXECUTION TIME (IN SECONDS) OF CED AND THE COMPARED METHODS NEEDED TO ANALYZE EACH OF THE TESTED TIME-LAPSE SERIES

Method	Dataset 1					Dataset 2						
	512 × 478 pixels					580 × 540 × (N = 60) voxels					N = 40	N = 35
	Series 1 1 cell	Series 2 2 cells	Series 3 5 cells	Series 4 4 cells	Series 5 1 cell	Series 1 2 cells	Series 2 2 cells	Series 3 2 cells	Series 4 1 cell	Series 5 1 cell	Series 6 1 cell	Series 7 2 cells
CED	4.59	4.62	4.62	4.63	4.61	353.95	353.16	353.66	353.72	353.53	235.67	206.58
FLS	0.74	0.85	0.83	0.79	0.73	76.13	70.02	90.73	55.67	61.28	40.59	79.06
GC	2.01	2.14	2.29	2.21	2.01	161.27	162.92	160.65	314.47	168.15	102.87	104.18
DZY	32.56	55.74	35.94	31.88	26.73	3484.97	3392.14	3456.48	2287.55	2975.37	2369.33	2481.17

TABLE V
MEMORY CONSUMPTION (IN MEGABYTES) OF THE FLS AND GC FRAMEWORKS DECOMPOSED INTO THE MEMORY CONSUMED BY IMAGE DATA PER FRAME AND THE PEAK MEMORY CONSUMED BY EACH FRAMEWORK DURING THE ANALYSIS OF EACH OF THE TESTED TIME-LAPSE SERIES

Image Data	Dataset 1					Dataset 2						
	512 × 478 pixels					580 × 540 × (N = 60) voxels					N = 40	N = 35
	Series 1 1 cell	Series 2 2 cells	Series 3 5 cells	Series 4 4 cells	Series 5 1 cell	Series 1 2 cells	Series 2 2 cells	Series 3 2 cells	Series 4 1 cell	Series 5 1 cell	Series 6 1 cell	Series 7 2 cells
FLS	17					53					41	38
GC	4	4	4	4	4	274	279	286	268	265	180	169
	19	19	19	19	19	1326	1326	1319	1321	1327	885	774

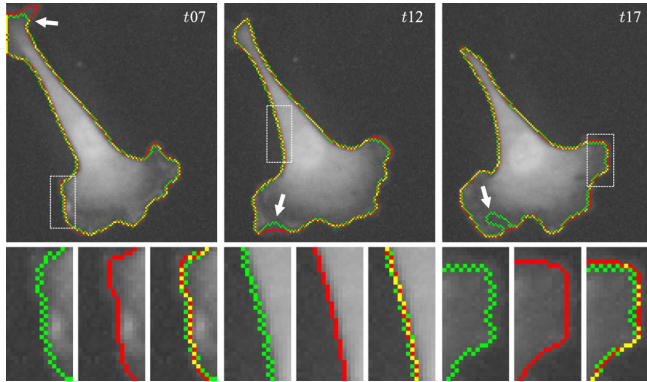


Fig. 11. Comparison of the final contours of the FLS (green) and GC (red) frameworks. The contour intersection is in yellow. The zoomed white rectangular areas illustrate a different boundary smoothness of each framework. The white arrows show foreground areas of low intensity levels in the close vicinity of the cell boundary, which are undersegmented by the FLS framework.

The GC framework, estimating $|C|$ using an Euclidean metric approximation derived from the Cauchy–Crofton formula [55], produces smoother boundaries than the FLS one (Fig. 11) in which the mean curvature motion affecting the boundary smoothness is only roughly approximated [32]. In addition, graph cuts had the tendency to detect more accurately foreground areas of low intensity levels (Fig. 11, white arrows), which appear occasionally in the close vicinity of the cell boundaries. However, the results of both frameworks were very similar. The median Hausdorff distance was about $1 \mu\text{m}$ (Table III), which roughly corresponded to two grid points with respect to given spatial image resolution (Table I).

Comparing to the proposed tracking scheme, DZY achieved down to about 9% lower accuracy (Table II). This significant deterioration is likely a consequence of the low signal-to-noise ratio, from about 0.8 to 1.2, of the tested time-lapse series, with

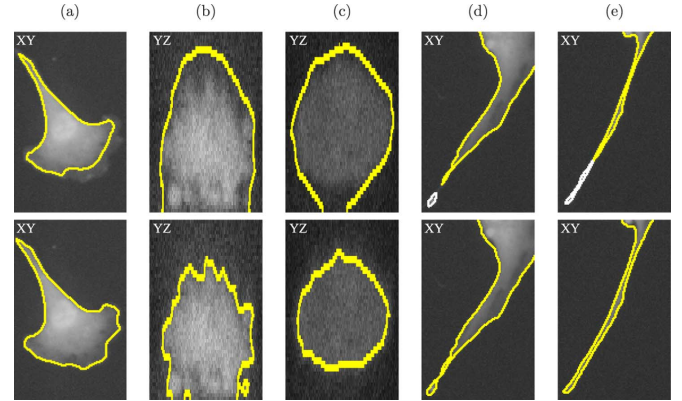


Fig. 12. Typical errors produced by DZY (top row) and improved performance by the GC framework (bottom row). (a)–(c) Incorrect segmentation of a cell cytoplasm of highly varying intensity levels. (d), (e) Oversegmentation of a slender cell extension is accumulated in a subsequent frame.

which the performance of DZY strongly correlates [20]. We also observed that DZY often had problems when segmenting cell cytoplasms of highly varying intensity levels leading to segmentation errors [Fig. 12(a)–(c)]. A false division event [Fig. 12(d)] led to treating the single cell as two separate ones, thus resulting in incorrect segmentation of a subsequent frame [Fig. 12(e)] and also in significant increase in the total execution time needed to analyze the corresponding time-lapse series (Table IV, Series 2 from Dataset 1). This increase is due to a higher number of handled level set functions as well as repeated application of the Radon transform used to find the optimal separating line between each pair of touching contours within the oversegmented cell. However, it is also important to note that DZY was mainly developed to analyze time-lapse series of cell nuclei. Therefore, it is not directly designed for application under such very specific conditions, under which the proposed tracking scheme can profit from the *a priori* knowledge incorporated into the model.



Fig. 13. Tracking of an artificial cell phantom with two pseudopods that touch over time in the GC framework.

Note that the reduction, up to almost 10%, in accuracy in 3-D (Table II) stems from the fact that we restricted the evaluation to only one randomly selected slice per frame, thus decreasing the absolute accuracy due to a generally worse segmentation of the slices close to the top and bottom of the cells.

The behavior when an elongated pseudopod splits into several regions, some of which merge together later, as shown in Fig. 9, is rather atypical for the state-of-the-art cell tracking algorithms. They assume that cells can split, but not merge again. Note that this behavior reflects limitations of 2-D imaging (i.e., the invisible parts of the nonflat pseudopod are in out-of-focus planes), rather than real separation and fusion of the pseudopod. Allowing this behavior can be useful, for instance, in studies focused on viral surfing [56], in which the accurate detection of individual cell extensions is the most crucial task.

Note that there are also rare situations which are misinterpreted by the object indication function. For instance, if two pseudopods of a moving cell touched, the tracked cell would contain a hole (Fig. 13). In this specific situation, it would be more appropriate to apply a different topological prior, based on the simple point concept from digital geometry [53]. This prior would keep both pseudopods separated, thus preserving the tracked cell to be topologically equivalent to a circle. Nevertheless, the simple point concept preserves strictly the initial topology of tracked objects. Therefore, splitting a cell into several isolated regions would result in artificial foreground paths cohering these regions together [34].

Although the GC framework outperformed the other two methods in terms of accuracy, its execution time and memory consumption were not superior (Tables IV and V). Graph cuts were about one order of magnitude faster than DZY, but they were almost three times slower in comparison to the FLS framework. The main reason for this is that the FLS framework is more suitable to incorporate the topological prior that exploits the object indication function than graph cuts due to a direct access to the evolving contour. Nevertheless, graph cuts could achieve comparable execution times in certain scenarios. This would especially hold for single-cell or low-cell-density studies in which no topological prior is needed. In those cases, the Chan–Vese model could be minimized using the more efficient Kohli–Torr algorithm, instead of the modified Boykov–Kolmogorov one, the speed of which is comparable to that of the FLS algorithm [32]. However, the roughly five-times higher memory consumption of the GC framework, in comparison to the FLS one, would remain since it depends mostly on the size of a graph constructed for a particular frame rather than the routine for computing a minimum cut.

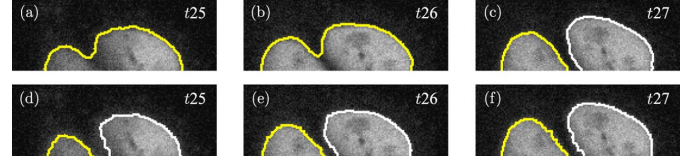


Fig. 14. Tracking a new cell nucleus entering the field of view closely to an existing one. (a)–(c) DZY does not capture the new nucleus, which leads to a false division event. (d)–(f) The GC framework does not suffer from that since the new nucleus is captured before segmenting the current frame.

In summary, the key factors that influence, approximately linearly, the execution time of the proposed tracking scheme are the frame size and the total number of iterations needed to minimize the Chan–Vese model. In the case of the FLS framework, this number is mostly given by the size of cell displacements between two successive frames. The farther away the initial seeds are from the cell boundaries, the more iterations need to be performed to obtain the final contours. In the case of graph cuts, the number of iterations depends mainly on the difference between the initial and final background and foreground statistics as well as the difference between these two statistics themselves. The memory consumption of the proposed tracking scheme is fairly linear with respect to the frame size, nearly independent of the number of tracked cells.

The most important parameters of the proposed tracking scheme are the user-defined weights of the Chan–Vese model (i.e., μ , λ_1 , and λ_2). The performed experiments on our 2-D and 3-D time-lapse datasets, having completely different intensity distributions (Section III), revealed that only the parameter λ_2 needs to be optimized. The other two parameters can be fixed, depending on the used minimization framework (Table II). Furthermore, given fixed μ and λ_1 , the range of λ_2 for which the proposed tracking scheme achieves almost the best possible accuracy is relatively wide (Fig. 7).

Although the main purpose of the proposed tracking scheme is to analyze whole fluorescent cells, its application is not limited only to this particular task. We have also tested that it can be used to segment and track fluorescently labeled cell nuclei, and that the object indication function is also appropriate to handle nuclei of mitotic cells. This ability was verified on the PCNA-GFP dataset from [20] (courtesy of J. Essers, Erasmus Medical Center, The Netherlands), which was acquired using a Zeiss LSM-510 laser-scanning confocal microscope. It had 92 frames of size $512 \times 443 \times 5$ voxels and contained 1270 cell nuclei in total, taken with a wide field of view. The images were not preprocessed in any way in order to obtain exactly the same tracking results for DZY as those presented in [20]. Their accuracy was measured again using the Jaccard coefficient JC . For this purpose, 92 randomly selected slices, one per frame, were manually segmented by an expert biologist and taken as a ground truth. Although DZY was targeted at this type of image data, the FLS and GC frameworks were slightly more accurate and about one order of magnitude faster than DZY (Table VI). Furthermore, they detected correctly all mitotic events and did not suffer from false ones owing to uncaptured new nuclei entering the field of view closely to the existing ones (Fig. 14). An example of the tracking results is depicted in Fig. 15.

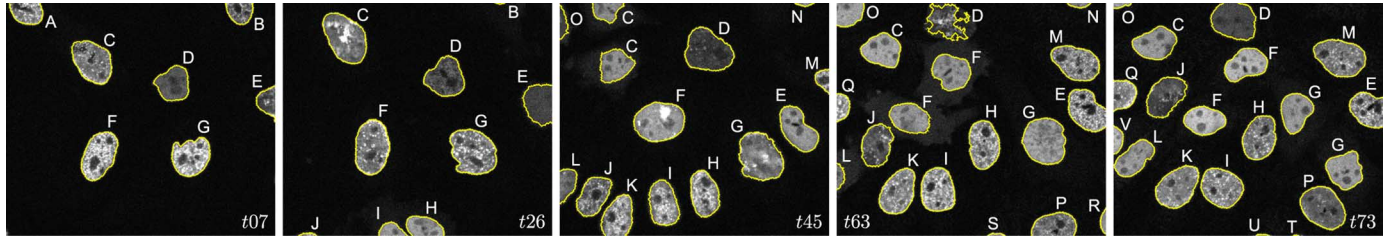


Fig. 15. Example of cell nucleus tracking results of the FLS framework. Original image data (the first slice of a frame) overlaid with thick final contours. The white letters, from A to V, encode object indicators of individual contours.

TABLE VI
AVERAGE ACCURACY JC_{avg} , TOTAL EXECUTION TIME (IN SECONDS), AND THE NUMBER OF TRUE AND FALSE DIVISION EVENTS OF THE COMPARED METHODS ON A TIME-LAPSE SERIES OF CELL NUCLEI

Method	JC_{avg}	Time	True Divisions	False Divisions
FLS	0.8602	124.19	4 / 4	0
GC	0.8635	188.72	4 / 4	0
DZY	0.8598	2988.29	3 / 4	1

VI. CONCLUDING REMARKS

In this paper, we have presented a fast and robust approach to tracking whole fluorescent cells in time-lapse series. The proposed tracking scheme combines CED filtering with the FLS and GC frameworks that minimize the Chan–Vese model. It allows simultaneous tracking of multiple cells over time by applying a topological prior that exploits the object indication function. The experimental evaluation was performed on 2-D and 3-D time-lapse series of rat adipose-derived mesenchymal stem cells and human squamous cell carcinoma cells, respectively. It clearly verified the improved accuracy up to about 9% and speed about one order of magnitude compared to DZY. The thorough comparison of the FLS and GC frameworks showed that graph cuts were slightly more accurate and produced smoother boundaries, but they were also nearly three times slower and consumed about five times more memory than the FLS framework. Thus, they could be preferred in studies focused on local morphological changes in the cell shape, in which as-accurate-as-possible cell boundaries are the most crucial task. On the contrary, the experiments aiming at global characteristics of cell migratory processes, such as speed or trajectory, might favor the FLS framework thanks to its low computational demands, but yet sufficient accuracy.

There are several limitations that we intend to address in future work to improve the overall performance of the proposed tracking scheme. First, a manual separation of cells clustered in the first frame is required to track each of them correctly over time. This complicates the use of the proposed tracking scheme in experiments with high density of tightly packed cells. Furthermore, coherence-enhancing diffusion filtering takes up to about 85% of the total execution time. Therefore, a different choice of the filtering technique would make the proposed tracking scheme significantly faster and more suitable for high-throughput applications. Last but not least, it might be profitable to integrate the FLS framework with a different approximation of the mean curvature motion to obtain smoother

boundaries, or incorporate the object indication function directly to the Kohli–Torr algorithm to improve the overall speed of the GC framework.

The C++ source codes as well as the Matlab interface of the proposed tracking scheme are made publicly available at <http://www.codesolorzano.com/software/CellFrost>, free of charge for noncommercial and research purposes.

ACKNOWLEDGMENT

The authors would like to thank A. Sancho (Centre of Studies and Technical Research of Gipuzkoa, Spain), and B. Pelacho and C. Ederra (Center for Applied Medical Research, Spain) for providing the ADMSC time-lapse series, J. Essers (Erasmus Medical Center, The Netherlands) for providing the PCNA-GFP time-lapse series, and O. Dzyubachyk (Leiden University Medical Center, The Netherlands) for providing the Matlab interface of the reference tracking algorithm [20].

REFERENCES

- [1] C. Zimmer, B. Zhang, A. Dufour, A. Thébaud, S. Berlemont, V. Meas-Yedig, and J.-C. Olivo-Marin, “On the digital trail of mobile cells,” *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 54–62, May 2006.
- [2] R. Ananthakrishnan and A. Ehrlicher, “The forces behind cell movement,” *Int. J. Biol. Sci.*, vol. 3, no. 5, pp. 303–317, 2007.
- [3] R. Fernández-González, A. Muñoz-Barrutia, M. H. Barcellos-Hof, and C. Ortiz-de-Solórzano, “Quantitative in vivo microscopy: The return from the ‘omics’,” *Curr. Opin. Biotechnol.*, vol. 17, no. 5, pp. 501–510, 2006.
- [4] C. Vonesch, F. Aguet, J.-L. Vonesch, and M. Unser, “The colored revolution of bioimaging,” *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 20–31, May 2006.
- [5] E. Meijering, O. Dzyubachyk, I. Smal, and W. A. Cappellen, “Tracking in cell and developmental biology,” *Seminars Cell Develop. Biol.*, vol. 20, no. 8, pp. 894–902, 2009.
- [6] P. Sarder and A. Nehorai, “Deconvolution methods for 3-D fluorescence microscopy images,” *IEEE Signal Process. Mag.*, vol. 23, no. 3, pp. 32–45, May 2006.
- [7] J. Weickert, A. Bruhn, T. Brox, and N. Papenberg, “A survey on variational optic flow methods for small displacements,” in *Mathematical Models for Registration and Applications to Medical Imaging*. New York: Springer-Verlag, 2006, pp. 103–136.
- [8] P. Matula, P. Matula, M. Kozubek, and V. Dvořák, “Fast point-based 3-D alignment of live cells,” *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2388–2396, Aug. 2006.
- [9] K. Li, M. Chen, T. Kanade, E. D. Miller, L. E. Weiss, and P. G. Campbell, “Cell population tracking and lineage construction with spatiotemporal context,” *Med. Image Anal.*, vol. 12, no. 5, pp. 546–566, 2008.
- [10] O. Al-Kofahi, R. J. Radke, S. K. Goderie, Q. Shen, S. Temple, and B. Roysam, “Automated cell lineage construction: A rapid method to analyze clonal development established with murine neural progenitor cells,” *Cell Cycle*, vol. 5, no. 3, pp. 327–335, 2006.
- [11] N. Harder, F. Mora-Bermúdez, W. J. Godínez, J. Ellenberg, R. Eils, and K. Rohr, “Automated analysis of the mitotic phases of human cells in 3-D fluorescence microscopy,” in *Medical Image Computing and Computer-Assisted Intervention*. New York: Springer, 2006, pp. 840–848.

- [12] F. Li, X. Zhou, J. Ma, and S. T. C. Wong, "Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis," *IEEE Trans. Med. Imag.*, vol. 29, no. 1, pp. 96–105, Jan. 2010.
- [13] D. Padfield, J. Rittscher, and B. Roysam, "Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis," *Med. Image Anal.*, vol. 15, no. 1, pp. 650–668, 2011.
- [14] O. Debeir, P. V. Ham, R. Kiss, and C. Decaestecker, "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes," *IEEE Trans. Med. Imag.*, vol. 24, no. 6, pp. 697–711, Jun. 2005.
- [15] R. Nilanjan, S. T. Acton, and K. Ley, "Tracking leukocytes in vivo with shape and size constrained active contours," *IEEE Trans. Med. Imag.*, vol. 21, no. 10, pp. 1222–1235, Oct. 2002.
- [16] C. Zimmer, E. Labruière, V. Meas-Yedid, N. Guillén, and J.-C. Olivo-Marin, "Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing," *IEEE Trans. Med. Imag.*, vol. 21, no. 10, pp. 1212–1221, Oct. 2002.
- [17] D. Mukherjee, N. Ray, and S. Acton, "Level set analysis for leukocyte detection and tracking," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 562–572, Apr. 2004.
- [18] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J.-C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Trans. Image Process.*, vol. 14, no. 9, pp. 1396–1410, Sep. 2005.
- [19] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam, "Spatio-temporal cell cycle phase analysis using level sets and fast marching methods," *Med. Image Anal.*, vol. 13, no. 1, pp. 143–155, 2009.
- [20] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering, "Advanced level-set-based cell tracking in time-lapse fluorescence microscopy," *IEEE Trans. Med. Imag.*, vol. 29, no. 3, pp. 852–867, Mar. 2010.
- [21] A. Dufour, R. Thibaux, E. Labruière, N. Guillén, and J.-C. Olivo-Marin, "3-D active meshes: Fast discrete deformable models for cell tracking in 3-D time-lapse microscopy," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1925–1937, Jul. 2011.
- [22] N. Chenouard, I. Bloch, and J.-C. Olivo-Marin, "Multiple-hypothesis tracking in microscopy images," in *Proc. 6th IEEE Int. Symp. Biomed. Imag.*, 2009, pp. 1346–1349.
- [23] I. Smal, N. Carranza-Herrezuelo, S. Klein, W. Niessen, and E. Meijering, "Quantitative comparison of tracking methods for motion analysis in tagged MRI," in *Proc. 8th IEEE Int. Symp. Biomed. Imag.*, 2011, pp. 345–348.
- [24] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vis.*, vol. 1, no. 4, pp. 321–331, 1987.
- [25] M. Jacob, T. Blu, and M. Unser, "Efficient energies and algorithms for parametric snakes," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1231–1244, Sep. 2004.
- [26] C. Zimmer and J.-C. Olivo-Marin, "Coupled parametric active contours," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 11, pp. 1838–1842, Nov. 2005.
- [27] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, "Fast parametric snakes for 3-D microscopy," in *Proc. 9th IEEE Int. Symp. Biomed. Imag.*, 2012, pp. 852–855.
- [28] V. Caselles, F. Catté, T. Coll, and F. Dibos, "A geometric model for active contours in image processing," *Numerische Mathematik*, vol. 66, no. 1, pp. 1–31, 1993.
- [29] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *Int. J. Comput. Vis.*, vol. 22, no. 1, pp. 61–79, 1997.
- [30] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 266–277, Feb. 2001.
- [31] D. Chopp, "Computing minimal surfaces via level set curvature flow," *J. Computat. Phys.*, vol. 106, no. 1, pp. 77–91, 1993.
- [32] M. Maška, P. Matula, O. Daněk, and M. Kozubek, "A fast level set-like algorithm for region-based active contours," in *Proc. 6th Int. Symp. Vis. Comput.*, 2010, pp. 387–396.
- [33] N. El-Zehiry, S. Xu, P. Sahoo, and A. Elmaghraby, "Graph cut optimization for the Mumford-Shah model," in *Proc. 7th IASTED Int. Conf. Visualizat., Imag. Image Process.*, 2007, pp. 182–187.
- [34] M. Maška, P. Matula, and M. Kozubek, "Simultaneous tracking of multiple objects using fast level set-like algorithm," in *Proc. 6th Doctoral Workshop Math. Eng. Methods Comput. Sci. (Sel. Papers)*, 2011, pp. 69–76.
- [35] J. Weickert, "Coherence-enhancing diffusion filtering," *Int. J. Comput. Vis.*, vol. 31, no. 2/3, pp. 111–127, 1999.
- [36] J. Kybic and J. Krátý, "Discrete curvature calculation for fast level set segmentation," in *Proc. 16th IEEE Int. Conf. Image Process.*, 2009, pp. 3017–3020.
- [37] M. Maška, A. Muñoz-Barrutia, and C. Ortiz-de-Solórzano, "Fast tracking of fluorescent cells based on the Chan-Vese model," in *Proc. 9th IEEE Int. Symp. Biomed. Imag.*, 2012, pp. 1316–1319.
- [38] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, 1989.
- [39] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation," *J. Computat. Phys.*, vol. 79, no. 1, pp. 12–49, 1988.
- [40] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. New York: Springer-Verlag, 2003.
- [41] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
- [42] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *Int. J. Comput. Vis.*, vol. 70, no. 2, pp. 109–131, 2006.
- [43] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Med. Image Anal.*, vol. 4, no. 2, pp. 73–91, 2000.
- [44] H. Delingette and J. Montagnat, "Shape and topology constraints on parametric active contours," *Comput. Vis. Image Understand.*, vol. 83, no. 2, pp. 140–171, 2001.
- [45] F. Gibou and R. Fedkiw, "A fast hybrid k-means level set algorithm for segmentation," in *Proc. 4th Annu. Hawaii Int. Conf. Stat. Math.*, 2005, pp. 281–291.
- [46] S. Esedoglu and Y.-H. R. Tsai, "Threshold dynamics for the piecewise constant Mumford-Shah functional," *J. Computat. Phys.*, vol. 211, no. 1, pp. 367–384, 2006.
- [47] B. Song and T. F. Chan, A fast algorithm for level set based optimization Computat. Appl. Math. Group, Univ. California, Tech. Rep., 2002.
- [48] Y. Shi and W. C. Karl, "A real-time algorithm for the approximation of level-set-based curve evolution," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 645–656, May 2008.
- [49] Y. Boykov and V. Kolmogorov, "Computing geodesics and minimal surfaces via graph cuts," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, 2003, pp. 26–33.
- [50] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2006.
- [51] E. S. Brown, T. F. Chan, and X. Bresson, "Completely convex formulation of the Chan-Vese image segmentation model," *Int. J. Comput. Vis.*, vol. 98, no. 1, pp. 103–121, 2012.
- [52] O. Daněk and M. Maška, "A simple topology preserving max-flow algorithm for graph cut based image segmentation," in *Proc. 6th Doctoral Workshop n Math. Eng. Methods Comput. Sci. (Sel. Papers)*, 2011, pp. 19–25.
- [53] R. Klette and A. Rosenfeld, *Digital Geometry: Geometric Methods for Digital Picture Analysis*. San Francisco, CA: Morgan Kaufmann, 2004.
- [54] P. Kohli and P. H. S. Torr, "Dynamic graph cuts for efficient inference in Markov random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2079–2088, Dec. 2007.
- [55] O. Daněk and P. Matula, "Graph cuts and approximation of the Euclidean metric on anisotropic grids," in *Int. Conf. Comput. Vis. Theory Appl.*, 2010, pp. 68–73.
- [56] V. Tsimashchuk, W. J. Godinez, M. Lehmann, and K. Rohr, "Analysis of viral surfing based on fluorescence microscopy imaging and automatic tracking," in *Proc. 9th IEEE Int. Symp. Biomed. Imag.*, 2012, pp. 378–381.