Level set methods for watershed image segmentation

Erlend Hodneland[1], Xue-Cheng Tai[2*], Joachim Weickert[3], Nickolay V. Bukoreshtliev[1], Arvid Lundervold[1], and Hans-Hermann Gerdes[1]

[1] Department of Biomedicine, University of Bergen, Jonas Lies vei 91, 5009 Bergen, Norway
[2] Department of Mathematics, University of Bergen, Johannes Brunsgate 12, 5007, Bergen, Norway
[3] Faculty of Mathematics and Computer Science, Bldg. E11, Saarland University, 66041 Saarbrücken, Germany
*Corresponding author: tai@mi.uib.no

Abstract. In this paper a marker-controlled and regularized watershed segmentation is proposed. Only a few previous studies address the task of regularizing the obtained watershed lines from the traditional marker-controlled watershed segmentation. In the present formulation, the topographical distance function is applied in a level set formulation to perform the segmentation, and the regularization is easily accomplished by regularizing the level set functions. Based on the well-known Four-Color theorem, a mathematical model is developed for the proposed ideas. With this model, it is possible to segment any 2D image with arbitrary number of phases with as few as one or two level set functions. The algorithm has been tested on real 2D fluorescence microscopy images displaying rat cancer cells, and the algorithm has also been compared to a standard watershed segmentation as it is implemented in MATLAB. For a fixed set of markers and a fixed set of challenging images, the comparison of these two methods shows that the present level set formulation performs better than a standard watershed segmentation.

1 Introduction

Segmentation is a major challenge in image analysis, referring to the task of detecting boundaries of objects of interest in an image. Several approaches have been proposed, and they are mainly divided into two categories: energy-driven segmentation [1–7] and watershed-based [8–11]. Energy-driven segmentation normally consists of two parts, the data term and the regularizer. The data term assures a solution which is sufficiently close to the desired boundaries and the regularizer controls the smoothness of the boundaries. A smoothing is often required due to noise and artifacts in real images. Watershed segmentation [8–11] is a region growing technique belonging to the class of morphological operations. Traditionally, the watershed techniques have been conducted without a smoothing term, but recent progress has resulted in energy-based watershed segmen-

tations that contain regularizers. In the following the two main approaches for segmentation are treated more carefully.

The energy-driven segmentation methods are mainly divided into two classes, contour-based (snakes) and region-based. The contour based methods rely on strong edges or ridges as a stopping term in a curve evolution which is balanced between a data term and a smoothness term. The snake approach has been studied in [1, 7]. Cremers [6] included statistical shape knowledge to the Mumford-Shah functional and Xu [12] introduced the gradient vector flow (GVF) incorporating a global and external force which improved the capture range of their parametrical snake. One of the most well-known region-based method is the Mumford and Shah model [13]. In Chan-Vese [5, 14], the Osher-Sethian level set idea [15] was combined with the Mumford-Shal model to solve the region-based segmentation. Recently, some variants of the Osher-Sethian level set idea was proposed by Tai et al. [4, 16]. In this work, we shall extend these ideas to watershed segmentation.

The watershed segmentation has proven to be a powerful and fast technique for both contour detection and region-based segmentation. In principal, watershed segmentation depends on ridges to perform a proper segmentation, a property which is often fulfilled in contour detection where the boundaries of the objects are expressed as ridges. For region-based segmentation it is possible to convert the edges of the objects into ridges by calculating an edge map of the image. Watershed is normally implemented by region growing based on a set of markers to avoid severe over-segmentation [10,11,17]. Different watershed methods use slightly different distance measures, but they all share the property that the watershed lines appear as the points of equidistance between two adjacent minima. Meyer [9] use the topographical distance function for segmenting images using watershed segmentation, while Najman and Schmitt [8] present the watershed differences with classical edge detectors. Felkel et al. [17] use the shortest path cost between two nodes which is defined as the smallest lexiographic cost of all paths between two points, which reflects the flooding process when the water reaches a plateau.

The success of a watershed segmentation relies on a situation where the desired boundaries are ridges. Unfortunately, the standard watershed framework has a very limited flexibility on optimization parameters. As an example, there exists no possibility to smooth the boundaries. However, recent progress allows a regularization of the watershed lines [18] with an energy-based watershed algorithm (watersnakes). In contrast to the standard watershed and the watersnakes, our work is based on partial differential equations which easily allow a regularization of the watersheds. Moreover, the method is flexible with regard to several optimization parameters. As an example, it could allow optimization on the Euler number to avoid internal holes inside the phases. Furthermore, it would be desireable to develop methods which could optimize the *number* of markers in addition to the already implemented location of the watersheds. This property is important since creating markers automatically often results in se-

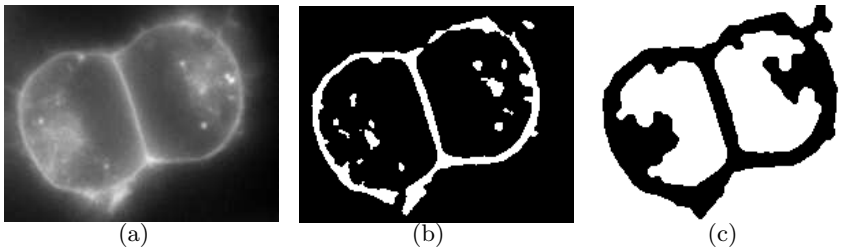
vere over-segmentation due to superfluous markers. Our method would permit an optimization on the number of markers.

2 Marker-controlled watershed segmentation by level set

2.1 Creating markers

The marker-controlled watershed segmentation has been shown to be a robust and flexible method for segmentation of objects with closed contours where the boundaries are expressed as ridges. The marker image used for watershed segmentation is a binary image consisting of either single marker points or larger marker regions where each connected marker is placed inside an object of interest. Each initial marker has a one-to-one relationship to a specific watershed region, thus the number of markers will equal the final number of watershed regions. After segmentation, the boundaries of the watershed regions are arranged on the desired ridges, thus separating each object from its neighbors. The markers can be manually or automatically selected, but high throughput experiments often employ automatically generated markers to save human time and resources.

In the present work, an adaptive thresholding [19] and filling of closed objects were used to automatically create markers. A similar combination of these operators was used in [20]. First, an adaptive thresholding was performed on the image f to label locally high intensity-valued pixels. In contrast to global thresholding, an adaptive thresholding demonstrates a much higher ability to deal with uneven scene illumination. A binary image f_b was thus constructed, where high intensity pixels are given the value 1 and the others 0. Then, all small objects in f_b were removed since they were considered to be insignificant due to their size. To be able to close minor gaps in the binary structures outlining the approximate boundaries, an iterative morphological closing was conducted. For each iterative closing step, a larger structural element was applied to facilitate the closing of incrementally larger gaps. Directly after each closing step a morphological filling was performed to fill all holes in f_b that were not accessed from the image boundary. All filled regions that had no intersection with earlier filled regions were then assigned to the marker image as a marker. The closing was performed iteratively in order to obtain markers that were as close as possible to the desired boundaries, and it was repeated a predefined number of steps which applied for all cell images in our experiments. Figure 1 demonstrates the process of creating markers from adaptive thresholding and filling, where the image in (a) was used to create a binary image (b) from adaptive thresholding. The smallest objects were removed and iterative closing and filling were applied to (b) to obtain the final binary marker image (c) where the marker regions are labeled white (value 1) and the background is black (value 0).



(a) (b) (c)

Fig. 1: Creating markers for watershed segmentation with level sets. Adaptive thresholding was applied to the image in (a) to construct a binary image (b). Thereafter, morphological closing and filling was applied to achieve the final marker image (c) which was used in the watershed segmentation.

2.2 Topographical distance function

Following [9], we apply the topographical distance function to obtain a watershed segmentation which is formulated within the level set theory. The topographical distance function between two points $\mathbf{x}$ and $\mathbf{y}$ is according to [18] defined as:

Definition 1. *For a smooth function* $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ *the topographical distance between two points* $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as the geodesic distance weighted by the gradient* $|\nabla f|$*, i.e.*

$$L(\mathbf{x}, \mathbf{y}) = \inf_{\gamma \in [\mathbf{x} \rightarrow \mathbf{y}]} \int_{\gamma} |\nabla f(\gamma(s))| ds \tag{1}$$

where $[\mathbf{x} \rightarrow \mathbf{y}]$ *denotes all possible paths from* $\mathbf{x}$ *to* $\mathbf{y}$ *.*

In 1D the topographical distance function is straightforward to compute since there is only one possible path between any two points $\mathbf{x}$ and $\mathbf{y}$. For 2D and 3D, the topographical distance function can be calculated using the iterative forest transform (IFT) [17] which computes the shortest path forest from given markers within a priority queue of pixels, where always the lowest intensity-valued pixel has the highest priority. The algorithm has a low time complexity of $O(m + n \log n)$ where $n = n_1 n_2$ is the number of pixels in the image and $m = n_1(n_2 - 1) + (n_1 - 1)n_2$, defined by a 6-connectivity neighborhood. For the present study, we have used the IFT algorithm to calculate the topographical distance function. To exemplify, see the synthetic image in Figure 2 (a) resembling a real cell image. Let L_1 be the topographical distance function from a point marker inside the cell (shown in b) and let L_2 be the corresponding topographical distance function from a point marker outside the cell (shown in c). Note the property of the topographical distance function that regions outside a ridge from a given marker are projected onto high-intensity regions. The desired ridge is obtained as the locations where $L_1 = L_2$. For more complex images, the ridge is given as locations where $L_i = L_j$ for the topographical distance functions L_i and L_j associated with adjacent markers. The plot in (d) shows an intensity profile of the image along the dashed line in (a) together with the corresponding

intensity profile of the topographical distance functions L_1 (dashed line) and L_2 (dotted line). Clearly, the ridge in (a) is obtained where $L_1 = L_2$.

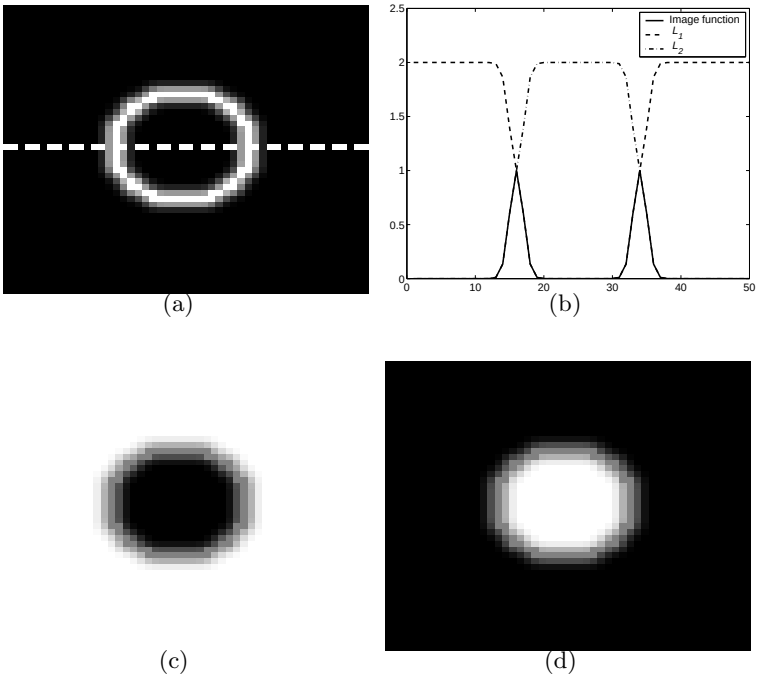


Fig. 2: Topographical distance function. The synthetic cell image (a) was used for calculating the topographical distance functions L_1 (shown in c) and L_2 (shown in d) from a point marker inside the cell and from a point marker outside the cell, respectively. The image profile along the dashed line in (a) is plotted in (b) (solid line) along with the corresponding profile of the topographical distance functions L_1 (dashed line) and L_2 (dotted line). It is evident that the desired ridge from the original image is at the spatial locations where the two topographical distance functions are equal.

2.3 Four-Color theorem

The Four-Color theorem will be applied in the watershed segmentation by level set to distinguish between objects inside the same phase. Using the Four-Color theorem, adjacent objects can be labeled among four colors and they are thus uniquely distinguishable. Chan and Vese [5] note that the Four-Color theorem can be used in image segmentation in the piecewise smooth case to distinguish between any number of objects with as few as four phases. The Four-Color theorem was proven first by Appel and Haken in 1976 [21], and it has been

validated again by different approaches in recent years [21]. The Four-Color theorem states the following: Define a graph G consisting of a finite set $V(G)$ of vertices and a finite set $E(G)$ of edges. Every edge is connected to two vertices, called the *ends*, whereas an edge with equal ends is called a loop. Each vertex is an arbitrary point associated to one region, such that each region has one vertex.

Theorem 1. *Every loopless plane graph G can be divided into 4 colors, that is a mapping $c : V(G) \rightarrow \{1, 2, 3, 4\}$ such that $c(u) \neq c(v)$ for every edge of G with ends u and v [21].*

3 The algorithm

Given a set of k markers, it is possible to divide the image into k regions according to the four-color theorem. This partition must be completed such that no objects belonging to the same color are adjacent after segmentation. If the markers are merely points or small regions this can be demanding since the boundaries of the objects are unknown prior to segmentation. Therefore, markers are preferrably automatically constructed as large marker regions with a certain extent, to allow an approximation of the influence zone of each marker which in turn enables a reliable partitioning of the markers according to the four-color theorem. In principle, this partitioning may fall short of success and turn out to be inconsistent, but in practical applications it is possible to achieve a reliable four-color partitioning prior to segmentation.

3.1 Euclidean influence zones

Given k markers, an influence zone is calculated around each marker which for all markers will divide the image into k influence zones. Label all markers $\{K_i\}_{i=1}^{k}$, then calculate the Euclidean distance function $d_i(\mathbf{x}) = dist(\mathbf{x}, K_i)$ around each marker K_i. Thus, k distance functions are obtained, $\{d_i(\mathbf{x})\}_{i=1}^{k}$. The Euclidean influence zone image f_{IZ} is a function such that

$$f_{IZ}(\mathbf{x}) = argmin_i\{d_i(\mathbf{x})\}_{i=1}^{k} = \{i|\ dist(\mathbf{x}, K_i) \leq dist(\mathbf{x}, K_j),\ \forall j\}. \qquad (2)$$

This representation is fast to compute and it divides the image into k disjoint regions suitable for a further labeling within the four-color theorem. See Figure 3 as an example where the given markers were automatically generated by the method described in Section 2.1, and the Euclidean influence zones f_{IZ} were obtained from Equation 2. Thus, the piecewise constant image f_{IZ} is constructed where each region i is uniquely labeled by an integer from $\{1, 2, \cdots, k\}$.

3.2 The Four-Color theorem and the topographical distance functions

The four-color theorem was applied to the Euclidean influence zones f_{IZ} function which contains an approximation to the final boundaries after segmentation.

Fig. 3: The Euclidean influence zones f_{IZ}. The image in Figure 1(a) was used to create the marker image in Figure 1(c) where the three markers are uniquely labeled. Based on the distance transform, the Euclidean influence zones f_{IZ} were calculated to obtain an approximation on the boundaries of the finally segmented image. The influence zones are then later painted with at most four colors.

For the images in this paper the painting of the regions were done by hand. Thus, a final partitioning f_c was obtained where adjacent zones in f_{IZ} and their corresponding markers are always assigned different colors. Empty colors will not influence the performance of the algorithm.

Once the markers have been painted with one of the four colors, we can group the marker into four groups, i.e. we define

$$C_i = \cup_{f_c(K_j)=i} K_j, \quad i = 1, 2, 3, 4.$$

We then use the method of [17] to compute up to four topographical distance functions from the marker groups C_i

$$L_i(\mathbf{x}) = \inf_{y \in C_i} L(\mathbf{x}, \mathbf{y}) \quad i = 1, 2, 3, 4.$$

As was proven in [18], a partition $\{\Omega_i\}_{i=1}^k$ minimizes the following functional

$$E(\Omega_1, \ldots, \Omega_k) = \sum_{i=1}^{k} \int_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x} \tag{3}$$

if and if it is a watershed segmentation. In our calculations, α_i is the minimum value of f on the boundary of marker i. In the following, we propose a level set method to solve the above watershed segmentation problem.

3.3 Level set formulation

We shall use three different variants of the level set idea to do the watershed segmentation based on the function $L_i, i = 1, 2, 3, 4$.

First, we propose to use the Osher-Sethian level set idea [15] as in Chan-Vese [14] for the segmentation. Let $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ be two level set functions defined on the domain Ω. These functions will partition the domain into four (possibly disconnected) sub-regions. The characteristic functions for these sub-regions are $\psi_i, i = \{1, 2, 3, 4\}$ given as

$$\begin{aligned} \psi_1(\phi_1, \phi_2) &= H(\phi_1)H(\phi_2), & \psi_2(\phi_1, \phi_2) &= (1 - H(\phi_1))H(\phi_2), \\ \psi_3(\phi_1, \phi_2) &= H(\phi_1)(1 - H(\phi_2)), & \psi_4(\phi_1, \phi_2) &= (1 - H(\phi_1))(1 - H(\phi_2)). \end{aligned}$$

The sub-regions are $\Omega_i = \{\mathbf{x}|\ \psi_i(\mathbf{x}) = 1\}, i = 1, 2, 3, 4$. This partition of the domain has no vacuum and no overlaps. In the above, $H(\cdot)$ denotes the Heaviside function, i.e. $H(x) = 1$ if $x \geq 0$, $H(x) = 0$ if $x < 0$. For the numerical experiments, a regularized Heaviside was used, $H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan(\frac{x}{\epsilon})\right)$ where $\epsilon > 0$ is small, see [14]. Using Eq. 3 within the level seg formulation, the desired watershed lines can be obtained by minimizing the following functional

$$F = \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i d\mathbf{x} + \beta \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i| d\mathbf{x}. \tag{4}$$

The first term is the data term providing the watershed segmentation and the second term is the regularization.

The second level set method we propose to use is the so-called Binary level set method [2, 4]. For this method, we need to find two functions $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ satisfying $\phi_i(\mathbf{x})^2 = 1,\ i = 1, 2$. The characteristic functions for the sub-regions partitioned by ϕ_i are given by

$$\psi_{i+1+2*j} = \frac{1}{4}\left(1 + (-1)^i \frac{\phi_1}{|\phi_1|}\right)\left(1 + (-1)^j \frac{\phi_2}{|\phi_2|}\right), \quad i, j = 0, 1.$$

This will give us the characteristic functions $\psi_k, k = 1, 2, 3, 4$ for the four sub-regions. In our experiments, $\phi/|\phi|$ are replaced by $\phi/\sqrt{|\phi|^2 + \epsilon}$ with a small $\epsilon > 0$. The watershed segmentation for this level set method is obtained by minimizing:

$$F = \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i d\mathbf{x} + \beta \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i| d\mathbf{x} + \frac{1}{\sigma}\sum_{i=1}^{2} \int_\Omega (\phi_i^2 - 1)^2 d\mathbf{x}. \tag{5}$$

The constant $\sigma > 0$ is a penalization constant to enforce $\phi_i^2 = 1$. Due to the special constructions of the characteristic functions ψ_i, we can choose any $\sigma > 0$ in the above minimization functional.

The third level set method we propose to use is the "Piecewise Constant Level Set Methods (PCLSM)" [16]. For this method, we just need to use one level set function $\phi : \mathbb{R}^2 \to \mathbb{R}$ satisfying

$$\kappa(\phi) = (\phi - 1)(\phi - 2)(\phi - 3)(\phi - 4) = 0, \quad \text{in } \Omega.$$

Associated with this ϕ, the characteristic functions for the sub-regions are given by

$$\psi_i = \frac{1}{\lambda_i} \prod_{\substack{j=1 \\ j \neq i}}^{4} (\phi - j) \ \text{ and } \ \lambda_i = \prod_{\substack{k=1 \\ k \neq i}}^{4} (i - k).$$

In order to use this method for the watershed segmentation, we need to solve the following constrained minimization problem:

$$\min_{\phi,\ \kappa(\phi)=0} \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i d\mathbf{x} + \beta \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i| d\mathbf{x}. \tag{6}$$

As in [3, 16], Augmented Lagrangian or penalization methods can be used to solve the above constrained minimization problem.

For minimization problem (4), the Euler-Lagrange equations for ϕ_1 and ϕ_2 are:

$$\sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial \psi_i}{\partial \phi_1} + \beta\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_1} = 0,$$
$$\sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial \psi_i}{\partial \phi_2} + \beta\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_2} = 0.$$

As usual, the explicit gradient flow problem must be solved to steady state:

$$\frac{\phi_1^{n+1}-\phi_1^n}{\tau} = \sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial \psi_i(\phi_1^n,\phi_2^n)}{\partial \phi_1^n} + \beta\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i(\phi_1^n,\phi_2^n)}{|\nabla\psi_i(\phi_1^n,\phi_2^n)|}\right)\frac{\partial\psi_i(\phi_1^n,\phi_2^n)}{\partial\phi_1^n}$$
$$\frac{\phi_2^{n+1}-\phi_2^n}{\tau} = \sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial \psi_i(\phi_1^n,\phi_2^n)}{\partial \phi_2^n} + \beta\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i(\phi_1^n,\phi_2^n)}{|\nabla\psi_i(\phi_1^n,\phi_2^n)|}\right)\frac{\partial\psi_i(\phi_1^n,\phi_2^n)}{\partial\phi_2^n},$$

where the differentiation of $|\nabla\cdot|$ with respect to ϕ is in detail explained in [14]. The relation $H'(x) = \delta(x)$ was used for differentiation of ψ_i, and a smooth $\delta_\epsilon(x)$ was used in the numerical experiments by calculating the derivative of the smooth Heaviside. For minimization problems (5) and (6), the Euler-Lagrange equations equation can be obtained in a similar way. We shall omit the details.

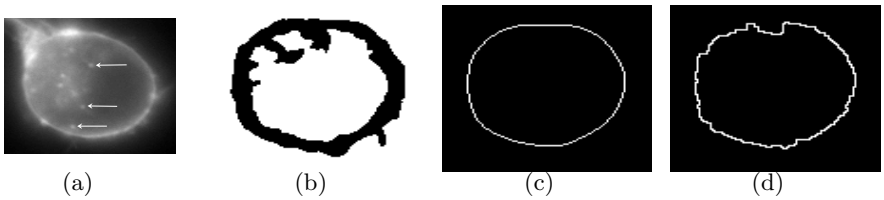
Here, we have proposed three different level set methods with different advantages and weak points, producing slightly different results which are complementary to each other.

4 Experimental results of segmentation

This section contains experiments involving real cell images taken by fluorescence microscopy showing rat pheochromocytoma PC12 cells. The images are optical planes extracted from 3D stacks, demonstrating the ability of the proposed method. In all experiments values of $\beta = 0.1$ and a time-step of $\Delta t = 0.01$ were chosen for the steepest descent method, and a standard watershed segmentation as implemented in MATLAB [11] was also calculated for comparison of performance between the two methods which used the same automatically generated markers. The images used for experimental testing point out important limitations in the image quality, in the sense that the traditional additive noise is not a serious question of matter, but rather internalized particles that appear as bright spots inside the cells (Figure 4(a), arrows). These internalized particles affect the segmentation negatively by causing more oscillatory watersheds. Additionally, such high intensity spots can be responsible for poorly segmented objects with boundaries passing through the middle of the objects (data not shown).

4.1 Two objects

For this example, the image in Figure 4(a) was chosen for segmentation. It shows one PC12 cell in addition to a background region. Based on the obtained marker image in (b), the watersheds for the watershed by level set (c) and for the standard watershed segmentation (d) were obtained. Note the smoothness of the watersheds in (c) compared to (d).

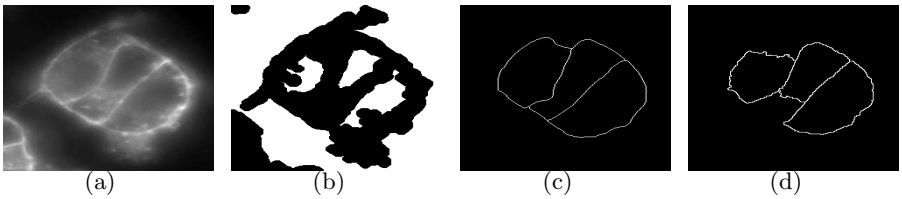


(a) (b) (c) (d)

Fig. 4: Watershed segmentation of one cell and a background region. The image in (a) was used to automatically obtain markers (b) which were used for watershed segmentation by level sets (c) and for a standard watershed segmentation (d). Note how the watershed by level set creates smoother boundaries (c) than the standard watershed (d). The arrows in (a) point to internalized particles appearing as bright spots inside fluorescently labeled cells. These can easily interfere with the standard watershed segmentation such that the watershed lines become oscillatory.

4.2 Four objects

The image in Figure 5 (a) shows three cells in addition to background. This image was used to automatically obtain a marker image (b). Based on the marker image, the watersheds for the watershed by level set (c) and the standard watershed segmentation (d) were obtained. Note that the watershed by level set captures more of the cells than the standard watershed segmentation. The cells in the lower left corner of (a) were omitted from the analysis since they are partially outside the image frame.



(a) (b) (c) (d)

Fig. 5: Watershed segmentation of three cells and a background region. The image in (a) was used to automatically obtain markers (b) which were used for watershed segmentation by level sets (c) and for a standard watershed segmentation (d). Note that the watershed by level set captures more of the cells than the standard watershed segmentation.

4.3 Multiple objects

The image in Figure 6 (a) shows six PC12 cells and background. Based upon this image, a marker image was automatically created and the watersheds for the watershed by level set (c) and standard watershed segmentation (d) were obtained. Compared to the standard watershed, the watershed by level set has a higher capacity of detecting the weak boundaries of the cell.

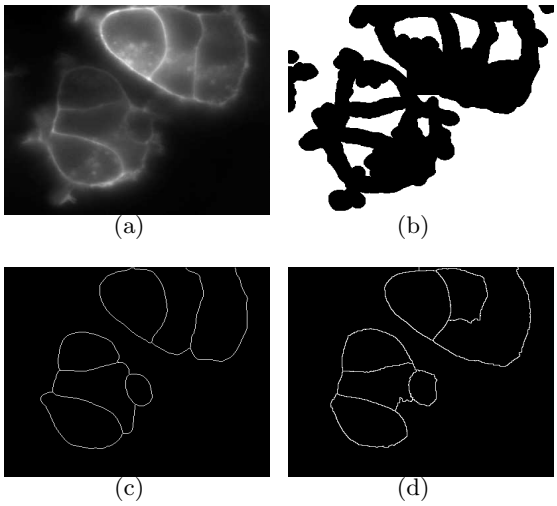


Fig. 6: Watershed segmentation of multiple cells and a background region. The image in (a) was used to automatically obtain markers (b) which were used for watershed segmentation by level sets (c) and for a standard watershed segmentation (d). Compared to the standard watershed, the watershed by level set has a higher capacity of detecting the weak boundaries of the cell.

5 Conclusion

In this study we have shown that the proposed watershed by level set is able to segment real cell images containing severe irregularites in a better way than the standard watershed segmentation. Our formulation is based on level set theory which easily allows a regularization of the watersheds, and which is a flexible approach for further optimization parameters.

References

1. Caselles, V., Catté, F., Coll, T., Dibos, F.: A geometric model for active contours in image processing. Numer. Math. **66**(1) (1993) 1–31
2. Nielsen, L.K., Tai, X.C., Aanonsen, S.I., Espedal, M.: Reservoir Description using a Binary Level Set Model. In: Image Processing based on partial differential equations. Springer, Heidelberg, Editor, X.-C. Tai, K. A. Lie, T. Chan and S. Osher (2006)

3. Christiansen, O., Tai, X.C.: Fast Implementation of Piecewise constant level set methods. In: Image Processing based on partial differential equations. Springer, Heidelberg, Editor, X.-C. Tai, K. A. Lie, T. Chan and S. Osher (2006)
4. Lie, J., Lysaker, M., Tai, X.C.: A binary level set model and some applications to Mumford-Shah image segmentation. IEEE Transactions on Image Processing **15**(5) (2006) 1171–1181
5. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the Mumford and Shah model. International Journal of Computer Vision **50**(3) (2002) 271–293
6. Cremers, D., Tischhäuser, F., Weickert, J., Schnörr, C.: Diffusion snakes: introducing statistical shape knowledge into the Mumford–Shah functional. International Journal of Computer Vision **50** (2002) 295 – 313
7. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision **V1**(4) (January 1988) 321–331
8. Najman, L., Schmitt, M.: Watershed of a continuous function. Signal Process. **38**(1) (1994) 99–112
9. Meyer, F.: Topographic distance and watershed lines. Signal Processing **38**(1) (1994) 113–125
10. Vincent, L., Dougherty, E.R.: Morphological Segmentation for Textures and Particles. In: Digital Image Processing Methods. E. Dougherty, Editor, Marcel-Dekker, New York (1994) 43–102
11. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6) (1991) 583–598
12. Xu, C., Prince, J.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing (1998) 359–369
13. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. Communications on Pure Applied Mathematics **42** (1989) 577–685
14. Chan, T., Vese, L.: Active contours without edges. IEEE Trans on Image Processing **10** (2001) 266–277
15. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics **79** (1988) 12–49
16. Lie, J., Lysaker, M., Tai, X.C.: A variant of the level set method and applications to image segmentation. Mathematics of computation **75**(255) (2006) 1155–1174
17. Felkel, P., Bruckschwaiger, M., Wegenkittl, R.: Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest. Computer Graphics Forum **20** (2002) 2001
18. Nguyen, H., Worring, M., van den Boomgaard, R.: Watersnakes: Energy-driven watershed segmentation. IEEE Trans. on PAMI **25**(3) (2003) 330–342
19. Chang, S.G., Yu, B., Vetterli, M.: Spatially adaptive wavelet thresholding with context modeling for image denoising. IEEE Transactions on Image Processing **9**(9) (2000) 1522–1531
20. Hodneland, E., Lundervold, A., Gurke, S., Tai, X.C., Rustom, A., Gerdes, H.H.: Automated detection of tunneling nanotubes in 3d images. Cytometry Part A **69A** (2006) 961–972
21. Robertson, N., Sanders, D., Seymour, P., Thomas, R.: A new proof of the four colour theorem. Electronic Research Announcements of the American Mathematical Society **2**(1) (1996)