Four-Color Theorem and Level Set Methods for Watershed Segmentation*

Erlend Hodneland† Xue-Cheng Tai‡ Hans-Herman Gerdes§

November 8, 2007

Abstract

A marker-controlled and regularized watershed segmentation is proposed for cell segmentation. 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

*The authors wish to thank Steffen Gurke and Nickolay Bukoreshtliev for providing the majority of pictures in this work. Erlend Hodneland and Hans-Hermann Gerdes would like to thank the Norwegian Cancer Society (project number A05103) for financial support. We also thank the national Molecular Imaging Center (MIC, Department of Biomedicine) for providing imaging facilities.

†Department of Biomedicine, University of Bergen Jonas Lies vei 91, N-5009 Bergen.

‡Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637616 and Department of Mathematics, University of Bergen, Johannes Brunsgate 12, 5007 Bergen, Norway. Emails: xctai@ntu.edu.sg and tai@mi.uib.no.

§Department of Biomedicine, University of Bergen Jonas Lies vei 91, N-5009 Bergen.

been proposed. Two important classes of segmentation approaches are the so-called energy-driven segmentation [4, 35, 11, 26, 27, 47, 13, 23] and watershed-based [33, 31, 48, 49]. Energy-driven segmentation normally uses an energy functional consisting of two parts, i.e. a data term and a regularizer. The data term assures a solution which is sufficiently close to the desired boundaries and the regularizer controls the smoothness of the obtained contours. A smoothing is often required due to noise and artifacts in real images. Watershed segmentation [33, 31, 48, 49] 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 segmentations that contain regularizers. In the following the energy-driven and the watershed based segmentation approaches are described 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 [23, 4]. Cremers [13] included statistical shape knowledge to the Mumford-Shah functional and Xu [52] 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 [32]. In Chan-Vese [47, 6], the Osher-Sethian level set idea [37] was combined with the Mumford-Shah model to solve the region-based segmentation. Recently, some variants of the Osher-Sethian level set idea was proposed by Tai et al. [28, 26, 27]. A good survey of variational segmentation methods can be found in [7].

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. The watershed transform algorithms can be divided into two groups, either based on the recursive flooding algorithm Vincent & Soille [49, 39, 10, 15, 33] or by different distance functions by Meyer [31, 34, 20, 38]. The former can be understood as a landscape which is flooded recursively and the watershed lines appear where the water from two different basins meet. The latter is computed from variants of the topographical distance, which can be implemented as a priority queue. Among these, 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. A review on different watershed approaches can be found in Roerdink & Meijster [41]. A common problem for the watershed transform is over-segmentation. However, watershed implemented by region growing based on a set of markers can avoid severe over-segmentation[49, 48, 15]. For the present work we use the topographical distance function as the method of choice, and we also create markers

to reduce over-segmentation.

The success of a watershed segmentation relies on the fact that the desired boundaries are ridges. Unfortunately, the standard watershed framework has a very limited flexibility on optimization parameters, for example, there exists no possibility to smooth the boundaries. However, recent progress allows a regularization of the watershed lines [34] 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 watershed lines. Moreover, the method is flexible with regard to several optimization parameters. It could allow optimization on the Euler numbers to avoid internal holes inside the segmented regions. Furthermore, it is possible to develop methods which could optimize the *number* of markers in addition to our implemented smoothing of the watersheds. This property is important since creating markers automatically often results in severe over-segmentation due to superfluous markers. Our method would permit an optimization on the number of markers.

It seems that level set methods have never been used for watershed segmentation. In this work, we shall combine the traditional level set methods [47] and the new variants [28, 26, 27] with watershed segmentation ideas. In §2, the foundation of our methods including the creation of the markers, the needed distance function and the four-color theorem are introduced. In §3, several level set methods are combined with the watershed segmentation idea using the topographical distance function. Details are given for the level set models. Combined with the Four-Color theorem, only one or two level set functions are needed to segment arbitrary numbers of regions. Implementation and numerical details are supplied in §5. Experiments with real data are given to demonstrate the performance of the proposed algorithms compared to the traditional watershed methods. It is shown that the methods can identify arbitrary number of regions just with one level set function.

2 Marker-controlled watershed segmentation by level set

2.1 Active contours and cell segmentation

The active contour models are able to perform segmentation of real cells when properly initialized [16, 3, 18]. In the active contour model, each seed point (or marker region) gives rise to a closed contour (snake) evolving until convergence. Then, the boundary of the cell should be captured. This process is repeated for all markers. However, boundary information between separate snakes from the different markers is mostly not connected to each other and therefore the resulting segmentation can result in regions of either vacuum or overlap. To deal with this problem, different conflict measures or restrictions of movement for the snakes to avoid overlap have to be implemented [16, 3]. Our level set watershed, which shall be introduced in §3, is implicitly defined in such way that vacuum or

overlap will never occur. This property is important in cell segmentation since cells are individual compartments, and the final segmentation should define every pixel either as part of one and only one cell, or as part of the background. Furthermore, a parametrical snake requires often one snake per object, or for the implicit snake, the number of level set functions is proportional to the number of objects as $\propto \log n$ when n is the number of objects or phases. Thus, the complexity increases swiftly for images containing a high number of cells. Due to the Four-Color theorem, c.f. §3, we are able to keep the number of level set functions at a very low level, independent of the number of objects to be segmented.

2.2 Creating Markers

Marker-controlled watershed segmentation is 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. Thus, each initial marker has a one-to-one relationship to the specific watershed region surrounded by the watershed lines. The final watershed segmentation is strictly depending on the markers, both the proposed watershed by level set and the standard watershed. This dependency is a consequence of both the one-to-one relationship as well as the size and position of the markers. Region-markers generally create results of higher quality than point-markers since their boundaries are closer to the desired boundaries and therefore there is a smaller probability of the flooding converging too early. The markers can be manually or automatically constructed, but high-throughput experiments often require automatically generated markers to save human time and resources. After segmentation, the boundaries of the watershed regions (watersheds) are arranged on the ridges, thus separating each object from its neighbors.

For the current project, the markers were automatically generated. The algorithm is sketched in the following and some detailed explanations are given afterward.

Flow scheme for creating markers.

1. u = ridge_enhancement(u)
2. ub = adaptive_thresholding(u)
3. ub = remove_small_objects(ub)
4. **for** i = 1 to (step = 1) 10
   a) se = get_structural_element(radius = i)
   b) closed = close(ub,se)
   c) filled = fill(closed)
   d) label = label_filled_regions(filled)
   e) **for** j = 1 to (step = 1) number_objects(label)
      if empty(intersection(object(j),markers))

markers(object(j)) = 1

5. markers = remove_small_objects(markers)

First, a Hessian ridge enhancement [17] was applied to enhance the ridges of the image f (#1 in flow scheme). The ridges are of special interest since the cell boundaries appear as ridges. The ridge enhancement is based on the eigenvalue decomposition of a Gaussian smoothed Hessian matrix

$$H = G_{\sigma,\delta} \begin{pmatrix} u_{xx} & u_{xy} & u_{xz} \\ u_{yx} & u_{yy} & u_{yz} \\ u_{zx} & u_{zy} & u_{zz} \end{pmatrix}$$

where the parameters σ and δ are the standard deviation and the height of the filter in the Gaussian. The eigenvector corresponding to the largest eigenvalue λ_1 points along the ridge, and the other points perpendicular to the ridge. A ridge is characterized by $\lambda_1 < 0$ and $\lambda_2 \approx 0$ [17], and is thus highlighted using a transfer function defined as $H(\lambda_1, \lambda_2) = -\lambda_1 - \lambda_2^2$, taking the highest values on the ridges. An example of a Hessian ridge enhancement is given in Fig 1 where the image (A) was used for Hessian ridge enhancement (B). For this example, $\sigma = 2$ and $\delta = 5$ were used. Clearly, the ridges are enhanced compared to other structures.

Adaptive thresholding [9, 19] was used to automatically create binary marker regions from the ridge enhanced image (#2 in flow scheme). The adaptive thresholding has a much higher resistance against noise and inhomogeneous illumination than global thresholding for labeling of high intensity objects, in our case the ridges. The adaptive thresholding image f_b was computed as

$$u_b(x,y) = \begin{cases} 1 & \text{if } f(x,y) > \mu \max(f) + A_\delta(f,x,y) \\ 0 & \text{else} \end{cases}$$

where μ is a user-defined scalar threshold with typical values of $[0.01 \rightarrow 0.2]$ and δ is the filter dimension of the average filter A_δ. $A_\delta(f,x,y)$ is the average values of f in a δ-neighborhood of (x,y). The image in Fig 1 (C) is an example of the adaptive thresholding with $\delta = 20$ and $\mu = 0.06$. Then, all small objects in u_b were removed since they were considered to be insignificant due to their size (D in Fig 1, #3 in the flow scheme). To be able to close minor gaps in the binary structures outlining the approximate boundaries, an iterative morphological closing was conducted (E in Fig 1, #4a-b in the flow scheme). For each iterative closing step, a larger structural element was applied to facilitate the closing of incrementally larger gaps. Directly after each closing step the function `imfill.m` in MATLAB was used to detect all holes in u_b that were not accessed from the image boundary (#4 in the flow scheme). All regions with a size within an interval and that had no intersection with earlier filled regions were then assigned to the marker image as a marker (F in Fig 1, #4e in the flow scheme). The closing was repeated iteratively with increasing radius r of the circular structural element (se), $r_i(\text{se}) < r_{i+1}(\text{se})$. This process is performed iteratively in order to obtain markers with boundaries as close as possible to the

desired cell boundaries, and it was repeated a predefined number of steps. Finally, removal of the smallest markers was necessary, using a threshold specified by the user (#5 in the flow scheme).

This marker-creation technique is important for the automated high-throughput experiments. It is fully automatic and requires a minimal use of human resources. The quality of the created marker is high in all the experiments we have performed.

2.3 Topographical distance function

We shall use the topographical distance function [31], closely related to the framework of minima paths [2], for our watershed segmentation. In addition, level set methods will be incorporated into this segmentation framework. The topographical distance function between two points $\mathbf{x}$ and $\mathbf{y}$ is defined as [34]:

Definition 1. For a smooth function $f(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ the topographical distance between two points $\mathbf{x}$ and $\mathbf{y}$ is the smallest integral of the gradient $|\nabla f|$ over any path γ from $\mathbf{x}$ to $\mathbf{y}$ in Ω, i.e.

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

In addition, we define $L_i(\mathbf{x}) = \inf_{\mathbf{y} \in M_i} L(\mathbf{x}, \mathbf{y})$. Therefore $\exists\ \mathbf{y}_i^* \in M_i$ such that $L_i(\mathbf{x}) = L(\mathbf{x}, \mathbf{y}_i^*)$. Furthermore, let K be the number of markers and let $\alpha_i = f(\mathbf{y}_i^*)$. Using $L_i(x)$, the definition of the *catchment basin* $CB(M_i)$ is given as below:

Definition 2. The catchment basin CB_i of a regional minimum M_i is defined by:

$$CB(M_i) = \{\mathbf{x} \in \Omega \mid \forall j \neq i, 1 \leq j \leq K : \alpha_i + L_i(\mathbf{x}) < \alpha_j + L_j(\mathbf{x})\}. \tag{2}$$

The parameter α_i works as a scaling of each minima to enable a comparison of the topographical distance functions between different minima. The watershed lines are closely related to the the catchment basins, and they are defined as the set of all points not belonging to any catchment basin.

Definition 3. The watershed line of the function f is the set of points not belonging to any catchment basin:

$$W(f) = \Omega \backslash \bigcup_i CB(M_i) \tag{3}$$

As $L_i(\mathbf{x})$ is continuous, $W(f)$ is equivalent to the points where $\alpha_i + L_i(\mathbf{x}) = \alpha_j + L_j(\mathbf{x}),\ i \neq j,\ \ i, j = \{1, \ldots K\}$. Thus, the watershed lines are given as the points where $\alpha_i + L_i(\mathbf{x})$ are equal over two adjacent regions.

As an illustration, Figure 2 shows an example of two adjacent marker-points M_1 and M_2 and their corresponding biased topographical distance functions

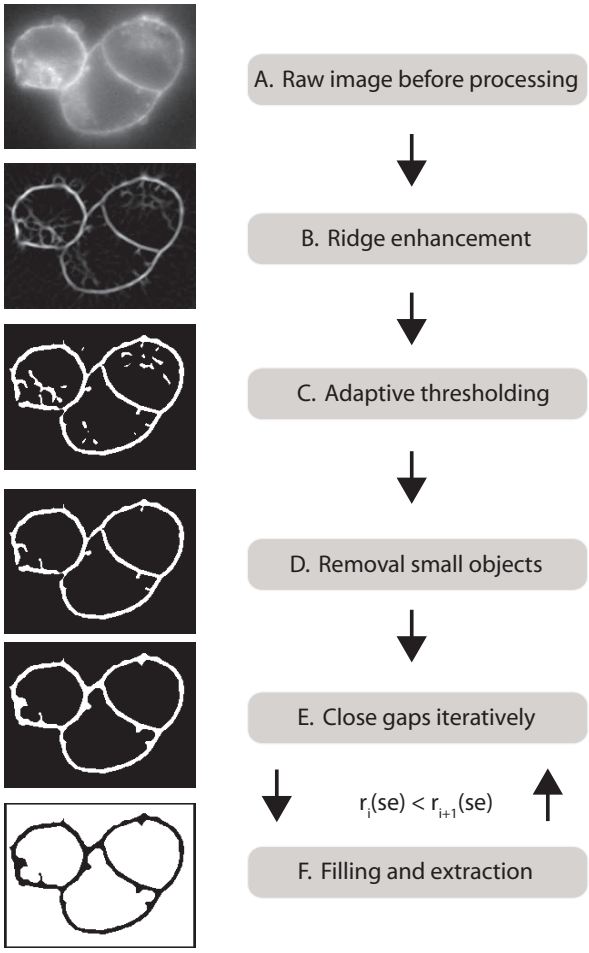


Figure 1: Automated construction of markers. The image (A) was used for ridge enhancement (B) to improve the signal intensity of the ridges. An adaptive thresholding was applied to detect the ridges and convert them into binary structures (C). The smallest objects were removed due their size (D) and a morphological closing was performed to close gaps in the binary structure (E). The closing was repeated iteratively with increasing radius r of the circular structural element (se), $r_i(\text{se}) < r_{i+1}(\text{se})$. A binary filling was computed after the closing at each iterative step (F) and all binary objects of the size within a user-defined interval were selected and used as marker regions. However, the objects were only selected if they had no intersection to previously selected objects. This enables larger marker regions with their boundaries closer to the true boundaries of the desired objects.

$\alpha_1 + L_1(x)$ and $\alpha_2 + L_2(x)$. In the figure, the biased topographical distance functions are indistinguishable in the convex regions around the markers since they are aligned with the function value (solid line). Note that $\alpha_1 + L_1(x) = \alpha_2 + L_2(x)$ on the ridge between the markers, $\alpha_1 + L_1(x) < \alpha_2 + L_2(x)$ for $x \in CB(M_1)$ and $\alpha_2 + L_2(x) < \alpha_1 + L_1(x)$ for $x \in CB(M_2)$. This reflects that among all markers M_i and their associated biased topographical distance functions $\alpha_i + L_i$, the biased topographical distance function $\alpha_i + L_i(x) \leq \alpha_j + L_j(x)$ for $x \in CB(M_i), i, j = 1 \ldots K$.

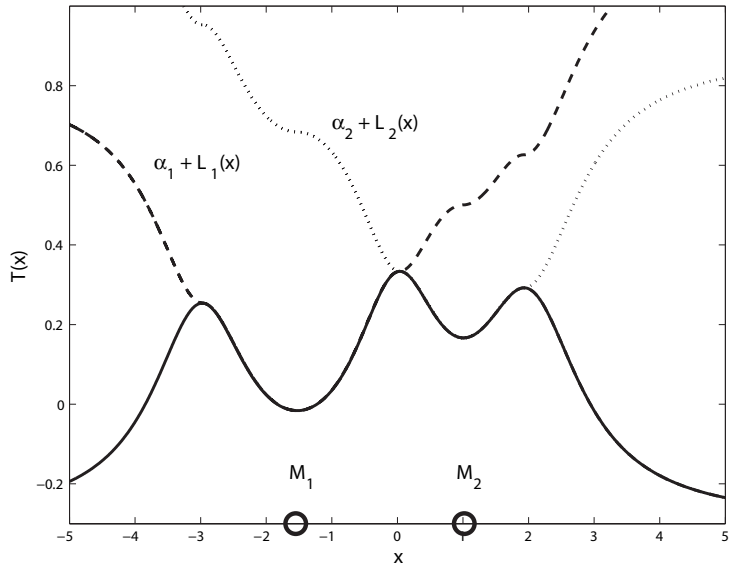


Figure 2: The biased topographical distance functions $\alpha_1 + L_1$ (dashed) and $\alpha_2 + L_2$ (dotted) around two adjacent marker-points M_1 and M_2, respectively. The underlying function is given by the solid line. Note that among all markers M_1 and M_2, the biased topographical distance function $\alpha_i + L_i(x)$ possesses the smallest value inside the catchment basin of M_i. On the ridge between two adjacent markers M_1 and M_2, $\alpha_1 + L_1(x) = \alpha_2 + L_2(x)$.

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) [15] which computes the shortest path energy between any two points. The algorithm has a low cost with a 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 4-connectivity neighborhood. For the present study, we have used the IFT algorithm to calculate the topographical distance function $L_i(\mathbf{x})$ and the flow scheme is given below:

```
1. Initialization
    a) for all nodes p
         flag(p) = TEMP;
    b) for all non–marker nodes p do
         C(p) = infinity;
    c) for all marker nodes p do
         C(p) = f(p);
    d) for all boundary marker nodes p do
         EnQueue(p,0);
2. Propagation
   while QueueNotEmpty() do
         a) v = DeQueueMin;
         b) flag(v) = DONE;
         c) for each p neighbor of v with
              flag(p) == TEMP do
                   if C(v) + w(v,p) < C(p) then
                        A) C(p) = C(v) + w(v,p);
                        B) if IsInQueue(p) then
                             DeQueue(p);
                        C) EnQueue(p,C(p));
```

The IFT algorithm has two steps. In the initialization phase (1), every pixel in an image f is assigned a flag as temporary TEMP (1a) and all non-marker pixels are assigned the cost infinity (1b). All marker pixels are assigned the cost of the image f (1c) and all the boundary pixels of the markers are added to the priority queue (1d) with a cost equal to zero. The loop in the propagation phase (2) runs as long as the priority queue is not empty. For each iteration, the pixel v with the lowest cost is removed from the queue (2a) and the pixel is assigned as DONE (2b). The neighbors around pixel v are detected, but only those labeled with TEMP are considered as candidates for the priority queue (2c) since the pixels labeled DONE are finished and already assigned the final cost value. If the cost of reaching pixel p from the pixel v is lower than previously assigned costs $C(p)$, pixel p is given the new and lower cost $C(p) = C(v) + w(v,p)$ (2cA) where $w(v,p)$ is the absolute value of the gradient, $w(v,p) = \frac{|f(v)-f(p)|}{d(v,p)}$, $d(v,p)$ is the Euclidean distance between v and p. If the neighbor p already was in the priority queue, it is removed from the queue (2cB) and returned with the new and lower cost $C(p)$ (2cC). If p was not in the queue, it is queued for the first time with the corresponding cost C(p) (2cC). The algorithm is designed such that only the front pixels of FLAG are inside the queue at any time, which ensures connectivity in the path. The fact that the pixel at each iteration with the lowest cost in the queue is removed, ensures that the path with the minimum cost is found.

2.4 Four-Color theorem

We shall use the Four-Color theorem in our watershed segmentation combined with level set methods. The Four-Color theorem, c.f. [1], states that it is possible to label any 2D planar graph with as few as four colors such that no neighbors have the same color. The Four-Color theorem was proven first by Appel and Haken in 1976 [1], and it has been validated again by different approaches in recent years [40]. Consider a set of regions (or countries) and select an arbitrary point inside each region (a capital). Join the points of every pair of neighboring regions with a line. Then, one arrives at the definition of a planar *graph* G. It consists of a finite number of *vertices* $V(G)$ and a finite set of *edges* $E(G)$.

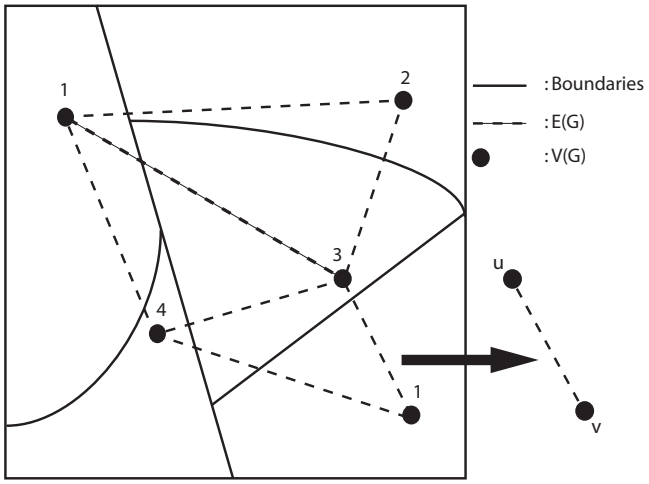


Figure 3: The Four-Color theorem. A planar graph consists of one arbitrarily selected *vertice* $V(G)$ (black dots) per region (solid line), and *edges* $E(G)$ (dashed lines) connecting the vertices of neighboring regions. Each edge has two ends, u and v. The Four-Color theorem states it is possible to label every u and v with different colors, using at most four colors. The number above the vertices indicate one possible Four-Color labeling.

Figure 3 shows the boundaries of the regions (solid lines) and the arbitrary points $V(G)$ inside each region (black spots, vertices) which are connected with the edges $E(G)$. Every edge must have two distinct *ends* u and v (Fig 3) which coincide with two distinct vertices. Thus, loops are not permitted. The Four-Color theorem states:

Definition 4. 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 [40].

Therefore, we could associate each watershed region with a vertex and use no more than four colors to mark the watershed regions. By doing so, adjacent

objects can be labeled among four colors and they are thus uniquely distinguishable since any two neighboring watershed regions will be labeled with different colors. Using the Four-Color theorem it is therefore possible to segment an arbitrary number of objects with as few as four phases. Chan and Vese [47] noted 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.

3 The algorithm

3.1 Euclidean influence zones

The four-color coding must be applied to a planar graph with no vacuum between the regions. Therefore, it can not be applied directly to the markers which are objects with gaps in between. To overcome this problem, an approximation of the final object partition based on the markers is computed. Ideally, this partition should capture information about the final boundaries of the objects, reflecting the assumed neighborship between the regions. If this approximation is accomplished, a coding can be achieved which contains a true four-color coding to distinguish the adjacent objects. To obtain the approximation of the final boundaries, the Euclidean distance transform from each marker is computed. To improve the chances of success in determining the neighborship between the regions, larger marker regions are preferred instead of small. Using large marker regions, it is a higher chance of obtaining a good approximation of the boundaries since the outer periphery of the markers is closer to the true object boundaries. Every pixel in the image is assigned to one influence zone of a marker such that each point from this zone has the smallest distance to the marker.

Mathematically, consider K markers and label all markers $\{M_i\}_{i=1}^K$. The Euclidean distance function $d_i(\mathbf{x}) = dist(\mathbf{x}, M_i)$ is calculated around each marker M_i using `bwdist.m` in MATLAB. Thus, K distance functions are obtained, $\{d_i(\mathbf{x})\}_{i=1}^K$. The Euclidean influence zone image f_{IZ} is a function defined as:

$$f_{IZ}(\mathbf{x}) = \{i \mid dist(\mathbf{x}, M_i) \leq dist(\mathbf{x}, M_j),\ \forall j\}. \tag{4}$$

Thus, $f_{IZ} = i$ if x has the shortest distance to make M_i. So, f_{IZ} is a piecewise constant function taking values from 1 to K. The region where $f_{IZ}(x) = i$ is called the Euclidean influence zone of marker M_i. This representation has a high computational speed and divides the image into K disjoint regions suitable for a further labeling within the Four-Color theorem. Figure 4 is an example of the Euclidean influence zones f_{IZ}. Using the method described in §2.2, the image in Fig. 4(a) was used to automatically construct the markers in (b). From these markers the Euclidean influence zones f_{IZ} were obtained using Eq. 4. Thus, a piecewise constant image f_{IZ} is constructed where each region is uniquely labeled by an integer from $\{1, 2, \ldots, K\}$, see (c).

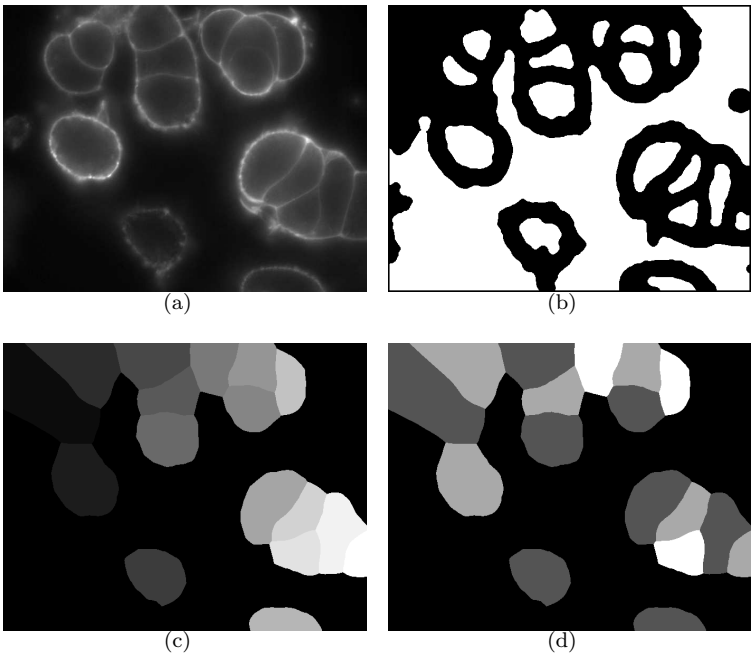


Figure 4: Euclidean influence zones and the Four-Color theorem. The image in (a) was used to automatically create markers (b) using the method described in §2.2. The Euclidean influence zones f_{IZ} were computed around the markers, shown in (c) where each region has a unique integer. The Four-Color theorem was applied to f_{IZ}, resulting in f_c (d) where no neighboring regions have the same color (black : 1, dark grey : 2, light grey : 3, white : 4).

Associated with the K markers, we have obtained K Euclidean influence zones. These zones are suitable for a color coding using the Four-Color theorem. Figure 4(d) shows one possible Four-Color coding of (c), where each zone has been labeled by an integer value from $\{1, 2, 3, 4\}$ and no region has a neighbor with the same integer value.

4 Level set methods for watershed segmentation

After applying the Four-Color theorem to the Euclidean influence zones $f_{IZ} = i$, we get an approximation to the final boundaries for the segmentation. For the images in this paper, the painting of the regions was done by an automated code where various combinations were tried and faulty ones eliminated to achieve a true partition according to the Four-Color theorem. Thus, a final coloring $f_c(x) \in \{1, 2, 3, 4\}$ was obtained where adjacent influence zones and their corre-

sponding markers are always assigned different colors, see an example of this in Fig 4(d). Empty colors will not influence the performance of the algorithm.

Once each marker has been painted with one of the four colors, the markers can be grouped into four groups, i.e., we define the group of markers $C_i = \cup_{f_c(M_j)=i} M_j$ and the boundaries of these, $\partial C_i, \quad i = 1, 2, 3, 4$. The method of [15] is then used to compute up to four topographical distance functions from each of the marker groups C_i, i.e., $L_i(\mathbf{x}) = \inf_{\mathbf{y} \in C_i} L(\mathbf{x}, \mathbf{y}),\ i = 1, 2, 3, 4$. It is true that there exists a $\mathbf{y}_i^* \in \partial C_i$ such that $L_i(x) = L(x, \mathbf{y}_i^*)$. Correspondingly, we also take $\alpha_i = f(\mathbf{y}_i^*)$. As was proven in [44], a partition $\{\Omega_i\}_{i=1}^4$ minimizes the functional

$$E(\Omega_1, \ldots, \Omega_4) = \sum_{i=1}^{4} \int_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x} \tag{5}$$

if and if it is a watershed segmentation around the group markers defined by the four colors. The proof is essentially the same as given in [34].

In the following, we propose to use level set methods to solve the above watershed segmentation problem. We shall use three different variants of the level set idea to accomplish the watershed segmentation based on the function $L_i, i = 1, 2, 3, 4$. These are the Chan-Vese level set [6], the Binary level set [35] and a variant of the Piecewise constant level set (PCLS) [28, 27].

4.1 Watersheds and the Chan-Vese model

First, we propose to use the level set idea [37] as in Chan-Vese [6] for the segmentation. Let $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ be two continuous level set functions defined on the domain Ω. Normally, ϕ_1, ϕ_2 are required to be distance functions to some curves, but this is not needed for our method here. We just need them to be continuous functions. 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, i.e., $H_\epsilon(x) = \frac{1}{2}\left(1 + \frac{2}{\pi} \arctan(\frac{x}{\epsilon})\right)$ where $\epsilon > 0$ is small, see [6]. 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.

Assume that the sub-regions $\Omega_i, i = 1, 2, 3, 4$ are related to $\phi_i, i = 1, 2$ as above, then we see that

$$\sum_{i=1}^{4} \int_{\Omega_i} \{\alpha_i + L_i(\mathbf{x})\} d\mathbf{x} = \int_{\Omega} \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\} \psi_i(\phi_1, \phi_2) d\mathbf{x}. \tag{6}$$

Based on this observation, we can try to solve the following minimization problem:

$$\min_{\phi_1,\phi_2} \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i d\mathbf{x}. \tag{7}$$

If $\{\phi_i\}_{i=1}^2$ is a minimizer of the above problem, then the corresponding subregions $\{\Omega_i\}_{i=1}^4$ is a watershed segmentation. In order to regularize the boundaries of the watershed regions, we shall add a regularization term into the minimization functional. Instead of solving (7), we try to minimize

$$\min_{\phi_1,\phi_2} F(\phi_1,\phi_2), \quad F = \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i d\mathbf{x} + \lambda \int_\Omega \sum_{i=1}^{4} |\nabla\psi_i| d\mathbf{x}. \tag{8}$$

The first term is the data term providing the watershed segmentation and the second term is the regularization to ensure a sufficiently smooth boundary for the watershed segmentation. The regularization is performed on $\{\psi_i\}_{i=1}^4$ which is different from [6] where the regularization is performed directly on $\{\phi_i\}_{i=1}^2$. The difference between these two approaches is discussed in more details in §5.7. A minimization of (8) with regard to ϕ_1 and ϕ_2 produces the following Euler-Lagrange equations

$$\begin{aligned}
\sum_{i=1}^{4} (\alpha_i + L_i(\mathbf{x})) \frac{\partial\psi_i}{\partial\phi_1} - \lambda \sum_{i=1}^{4} \nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_1} &= 0 \\
\sum_{i=1}^{4} (\alpha_i + L_i(\mathbf{x})) \frac{\partial\psi_i}{\partial\phi_2} - \lambda \sum_{i=1}^{4} \nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_2} &= 0.
\end{aligned}$$

In order to get the above results, we need to use the chain-rule as in [43, p.29] and [8, p.45]. The terms $\frac{\partial\psi_i}{\partial\phi_1}$ and $\frac{\partial\psi_i}{\partial\phi_2}$ are calculated as

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

In numerical simulations, δ and H are replaced by their smoothed counter parts δ_ϵ and H_ϵ respectively. As usual, we can use the gradient descent method to solve these equations. With some initial conditions for $\{\phi_i\}_{i=1}^2$, the gradient

flow equations are:

$$\begin{aligned}
\frac{\partial \phi_1}{\partial t} &= -\sum_{i=1}^{4} (\alpha_i + L_i(\mathbf{x})) \frac{\partial \psi_i}{\partial \phi_1} + \lambda \sum_{i=1}^{4} \nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \frac{\partial \psi_i}{\partial \phi_1} \\
\frac{\partial \phi_2}{\partial t} &= -\sum_{i=1}^{4} (\alpha_i + L_i(\mathbf{x})) \frac{\partial \psi_i}{\partial \phi_2} + \lambda \sum_{i=1}^{4} \nabla \cdot \left(\frac{\nabla \psi_i}{|\nabla \psi_i|} \right) \frac{\partial \psi_i}{\partial \phi_2}.
\end{aligned} \tag{9}$$

The following explicit scheme will be used in our numerical experiments. Note that faster methods can be used to solve these equations. For example, the AOS scheme proposed in [29, 30] and re-discoverted in [51] can be used for the equations (9) similarly as in [25]; The dual algorithm of [5] can also be used to get fast algorithms. In addition, the graph-cut techniques [14] could also be a powerful tool for these equations. For simplicity, we have only tested on the following explicit scheme so far:

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

where $\psi_i^n = \psi_i(\phi_1^n, \phi_2^n)$. This iteration is not the fastest algorithm. However, it often converges in less than 200 iterations for our experiments.

4.2 Watersheds and the Binary level set

The second level set method we propose to use is the so-called Binary level set [26, 42, 27]. This method has been used for image segmentation and inverse problems in [36, 46]. For this method, we need to find two functions $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}$ satisfying $\phi_i(\mathbf{x})^2 = 1,\ i = 1, 2$. These functions can also partition Ω into four sub-regions with the characteristic functions 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 sub-regions associated with the characteristic functions $\psi_i, i = 1, 2, 3, 4$ have no overlaps and vacuum. This method is closely related to the Chan-Vese model using two level set functions creating four sub-regions. However, the signum function is used instead of the Heaviside function. In the numerical experiments, $\phi/|\phi|$ are replaced by $\phi/\sqrt{|\phi|^2 + \epsilon}$ with a small $\epsilon > 0$.

Similar to (8), we solve the following minimization problem:

$$\min_{\phi_1, \phi_2} F(\phi_1, \phi_2) \tag{10}$$

$$F = \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\} \psi_i d\mathbf{x} + \lambda \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i| d\mathbf{x} + \sigma \sum_{i=1}^{2} \int_\Omega (\phi_i^2 - 1)^2 d\mathbf{x}. \tag{11}$$

Once the minimizer is obtained, the corresponding characteristic functions will give us the watershed sub-regions. In the above, the constant $\sigma > 0$ is a penalization constant to enforce $\phi_i^2 = 1$ and λ is the regularization parameter which influences the smoothness of the watershed lines. Due to the special constructions of the characteristic functions ψ_i, we can choose any $\sigma > 0$ in the above minimization functional. We need to use a large σ if the characteristic functions ψ_i are replaced by $\psi_{i+1+2*j} = \frac{1}{4}(1+(-1)^i\phi_1)(1+(-1)^j\phi_2)$. The Euler-Lagrange equations for minimization problem(11) with respect to ϕ_1 and ϕ_2 are:

$$\begin{aligned}
\sum_{i=1}^{4}(\alpha_i + L_i(\mathbf{x}))\frac{\partial\psi_i}{\partial\phi_1} - \lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_1} + 4\sigma\phi_1(\phi_1^2-1) &= 0\\
\sum_{i=1}^{4}(\alpha_i + L_i(\mathbf{x}))\frac{\partial\psi_i}{\partial\phi_2} - \lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi_2} + 4\sigma\phi_2(\phi_2^2-1) &= 0
\end{aligned}$$

Again, we use the following explicit scheme to solve the corresponding gradient flow equations:

$$\begin{aligned}
\frac{\phi_1^{n+1}-\phi_1^n}{\tau} = &-\sum_{i=1}^{4}(\alpha_i + L_i(\mathbf{x}))\frac{\partial\psi_i^n}{\partial\phi_1^n} + \\
&+\lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i^n}{|\nabla\psi_i^n|}\right)\frac{\partial\psi_i^n}{\partial\phi_1^n} - 4\sigma\phi_1^n((\phi_1^n)^2-1),\\
\frac{\phi_1^{n+1}-\phi_1^n}{\tau} = &-\sum_{i=1}^{4}(\alpha_i + L_i(\mathbf{x}))\frac{\partial\psi_i^n}{\partial\phi_2^n} + \\
&+\lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i^n}{|\nabla\psi_i^n|}\right)\frac{\partial\psi_i^n}{\partial\phi_2^n} - 4\sigma\phi_2^n((\phi_2^n)^2-1).
\end{aligned}$$

This algorithm is not sensitive to the values of σ and ϵ. We have always used $\sigma = 1$. Some discussions about the value of ϵ will be given later. The algorithm often converges in less than 200 iterations. It is also expected that some other discrete minimization method will accelerate the convergence [14].

4.3 Watersheds and the piecewise constant level set (PCLS)

The third level set method we propose to use is a variant of the "Piecewise Constant Level Set (PCLS)" method [28, 27]. This method has been used for image segmentation [46], inverse problems [45, 24] and optimal shape design problems [50, 25]. For this method, only one level set function $\phi : \mathbb{R}^2 \to \mathbb{R}$ is needed, satisfying

$$\kappa(\phi) = (\phi-1)(\phi-2)(\phi-3)(\phi-4) = 0 \quad \text{in } \Omega, \tag{12}$$

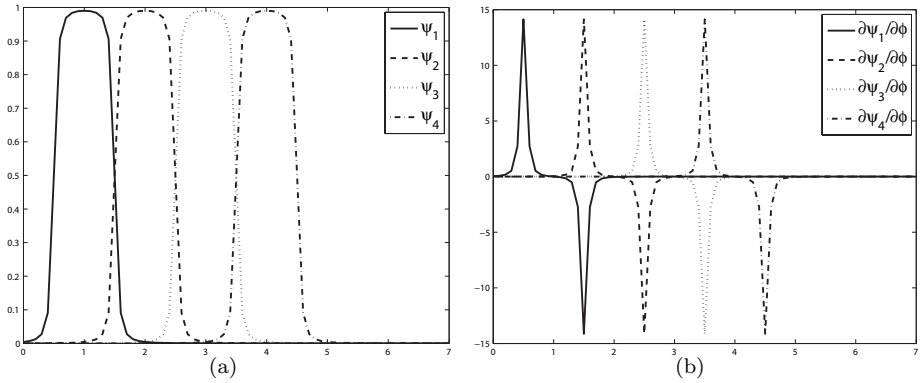
which ensures that ϕ takes piecewise constant values $\phi = \{1, 2, 3, 4\}$. In [45, 24, 25, 46, 28, 11, 27], penalization or Augmented Lagrangian methods were

used to deal with the constraint (12). In [21], it was shown that penalization methods can be connected with Modica-Mortola phase transition model to get even a rigorous convergence analysis.

In this work, we shall use the PCLS method more like a multi-layer level set method of [12]. Associated with ϕ, we define the characteristic functions for the sub-regions by

$$\psi_i(\phi) = \frac{1}{2}\left(\frac{\phi - i + 0.5}{\sqrt{(\phi - i + 0.5)^2 + \epsilon}} - \frac{\phi - i - 0.5}{\sqrt{(\phi - i - 0.5)^2 + \epsilon}}\right) \tag{13}$$

The first term is an approximation of a step function around $i - 0.5$ and the second term approximates a step function around $i + 0.5$. Fig. 5 illustrates the characteristic functions ψ_i for $i = \{1, 2, 3, 4\}$ (a) and their derivatives (b).



(a) (b)

Figure 5: The characteristic functions ψ_i (a) for $\epsilon = 0.005$, $i = \{1, 2, 3, 4\}$ and the corresponding derivatives (b). Note how the derivatives are overlapping with their neighbors.

As noted, the characteristic functions are constructed to ensure that there are overlaps between the support of the derivatives. In order to use this method for the watershed segmentation, we need to solve the following minimization problem:

$$\min_{\phi} \int_\Omega \sum_{i=1}^{4} \{\alpha_i + L_i(\mathbf{x})\}\psi_i(\phi) d\mathbf{x} + \lambda \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i(\phi)| d\mathbf{x} + \gamma \int_\Omega \kappa(\phi)^2 d\mathbf{x}. \tag{14}$$

As in [45, 24, 25, 46, 28, 11, 27], augmented Lagrangian can be used to solve the above minimization problem. Here, we have just used the penalization method for the constraint $\kappa(\phi) = 0$. Due to the special construction for the characteristic functions ψ_i used in (13), it is not necessary to use large values for the penalization constant γ.

For minimization problem (14), the Euler-Lagrange equation is, c.f. [43]

$$\sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial\psi_i}{\partial\phi} - \lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i}{|\nabla\psi_i|}\right)\frac{\partial\psi_i}{\partial\phi} + 2\gamma\kappa\frac{\partial\kappa}{\partial\phi} = 0, \tag{15}$$

where the term $\partial\psi_i/\partial\phi$ is given as

$$\frac{\partial\psi_i}{\partial\phi} = \frac{1}{2}\left(\frac{\epsilon}{((\phi - i + 0.5)^2 + \epsilon)^{\frac{3}{2}}} - \frac{\epsilon}{((\phi - i - 0.5)^2 + \epsilon)^{\frac{3}{2}}}\right). \tag{16}$$

The following explicit gradient flow problem must be solved to steady state:

$$\frac{\phi^{n+1} - \phi^n}{\tau} = -\sum_{i=1}^{4}\{\alpha_i + L_i(\mathbf{x})\}\frac{\partial\psi_i^n}{\partial\phi^n} + \lambda\sum_{i=1}^{4}\nabla\cdot\left(\frac{\nabla\psi_i^n}{|\nabla\psi_i^n|}\right)\frac{\partial\psi_i^n}{\partial\phi^n} - 2\gamma\kappa^n\frac{\partial\kappa^n}{\partial\phi^n} \tag{17}$$

where $\psi_i^n = \psi_i(\phi^n)$ and $\kappa^n = \kappa(\phi^n)$.

4.4 Differences between the level set methods

We have proposed three different level set methods with different advantages and weak points, producing slightly different results which are complementary to each other. All three methods produce good results. The Binary level set has several similarities to the Chan-Vese model in the way it produces a fixed number of 2^n phases where n is the number of level set functions. The Piecewise constant level set differs from the Chan-Vese and the Binary level set in several aspects. It only requires a single level set function. Furthermore, it is possible to define the desired number of phases without being limited to a fixed number which is the case for the Chan-Vese model and the Binary level set.

From our numerical experiences, it seems that the binary level set method is fast and stable for some examples, while the piecewise constant level set method is faster and stable for some other examples. Generally, when the structure of the object is complicated and the number of objects is large, the piecewise constant level set method seems to be advantageous. With three different methods at our disposal, it is really an advantage that we can use them to confirm that the obtained results are correct.

5 Numerical experiments

5.1 Numerical implementation

All three models have in common the biased topographical distance function, $\alpha_i + L_i(\mathbf{x})$. The values of the topographical distance function inside the markers were approximated by the image values, and α_i was approximated as $\alpha_i \approx \min f(\partial C_i)$, where ∂C_i is the boundary of marker group of color i. The second term $L_i(\mathbf{x})$ was computed as described in §2.3, one topographical distance function for each group of markers that are inside the same color i. The functions

$L_i(\mathbf{x})$ only need to be computed once for the whole computation. The regularization term $\nabla \cdot \left(\frac{\nabla \psi}{|\nabla \psi|}\right)$ was computed according to the following scheme. Consider forward, backward and central differences for x and y,

$$\begin{aligned} D_x^+(\psi_{i,j}^n) &= \frac{\psi_{i+1,j}^n - \psi_{i,j}^n}{h}, \quad D_y^+(\psi_{i,j}^n) = \frac{\psi_{i,j+1}^n - \psi_{i,j}^n}{h} \\ D_x^-(\psi_{i,j}^n) &= \frac{\psi_{i,j}^n - \psi_{i-1,j}^n}{h}, \quad D_y^-(\psi_{i,j}^n) = \frac{\psi_{i,j}^n - \psi_{i,j-1}^n}{h} \\ D_x(\psi_{i,j}^n) &= \frac{\psi_{i+1,j}^n - \psi_{i-1,j}^n}{2h}, \quad D_y(\psi_{i,j}^n) = \frac{\psi_{i,j+1}^n - \psi_{i,j-1}^n}{2h} \end{aligned}$$

where h is the stepsize. Using forward differences for the gradient and backward differences for the divergence,

$$\nabla \cdot \left(\frac{\nabla \psi_{i,j}^n}{|\nabla \psi_{i,j}^n|}\right) =$$

$$D_x^- \left(\frac{D_x^+(\psi_{i,j}^n)}{\sqrt{D_x(\psi_{i,j}^n)^2 + D_y(\psi_{i,j}^n)^2} + \epsilon}\right) + D_y^- \left(\frac{D_y^+(\psi_{i,j}^n)}{\sqrt{D_x(\psi_{i,j}^n)^2 + D_y(\psi_{i,j}^n)^2} + \epsilon}\right),$$

where ϵ is a small parameter to avoid singularities in regions where the gradient is zero. For the Chan-Vese model and the Binary level set, ϕ_1 and ϕ_2 were initialized as zero everywhere which enabled a fast convergence to the correct solution. For the PCLS, it was necessary to initialize the level set function ϕ in a special way, i.e.

$$\phi(\mathbf{x}, t=0) = \begin{cases} 2 & f_{IZ}(\mathbf{x}) \leq 2 \\ 3 & f_{IZ}(\mathbf{x}) \geq 3. \end{cases}$$

where f_{IZ} is the image for the Euclidean influence zones (§3.1). This initialization increased the computational speed.

The watershed lines (watersheds) appear at convergence as the interface between the regions of different colors. To obtain the watersheds, the characteristic functions ψ_i were computed with high accuracy after convergence. For the Chan-Vese model, the exact Heaviside was used in these calculations. For the Binary level set and the PCLS a very small $\epsilon = 10^{-10}$ was used. Then, the characteristic functions ψ_i for all three models were converted into binary functions ψ_i^b by thresholding,

$$\psi_i^b(\mathbf{x}) = \begin{cases} 1 & \psi_i(\mathbf{x}) \geq 0.5 \\ 0 & \psi_i(\mathbf{x}) < 0.5. \end{cases}$$

The outer boundary of the four binary functions $\psi_i^b(\mathbf{x})$ were used as the final watersheds of the image f,

$$WS(f) = \cup_{i=1}^4 \partial \psi_i^b(\mathbf{x}). \tag{18}$$

In all experiments values of $\lambda = [0.01, 0.1]$ were chosen for the regularization parameter, depending on the amount of endocytoced particles in the image.

A higher amount of noise requires larger λ. The largest possible time-step supporting a stable solution was $\Delta t = 0.1$. A smaller time-step does not change the minimizer in the Chan-Vese model or the Binary level set, but a large number of iterations is required to reach a steady state. For the PCLS, it is critical to use suitable associated settings of Δt, γ and ϵ. Empirically, values of $\Delta t = 0.05$, $\epsilon = 0.1$ and $\gamma = 0.05$ were appropriate as a global setting to converge toward a global minimizer. The same setting for ϵ also applied to the Binary level set. The number of iterations at convergence was between 200 and 400. All numerical code was written in MATLAB®, and the selected images used as examples were processed using one of the three proposed methods in §4.1-4.3. Additionally a standard marker-controlled watershed segmentation as implemented in MATLAB® [49] was calculated for comparison of performance between the two methods. Equal and automatically generated markers were applied to the compared methods in each example.

5.2 Synthetic data

The level set watershed method was applied to a synthetic image with multiplicative Gaussian noise and a linear gradient to test the robustness of the method. The original image free from noise is shown in Fig 6(a), and the same image after addition of Gaussian multiplicative noise and a linear gradient is displayed in (b). The obtained noisy image has a $SNR = 3.4$. The standard watershed segmentation and the PCLS level set watershed segmentation are shown in (c) and (d), respectively. Apparently, the level set watershed has a better ability to deal with the noise.

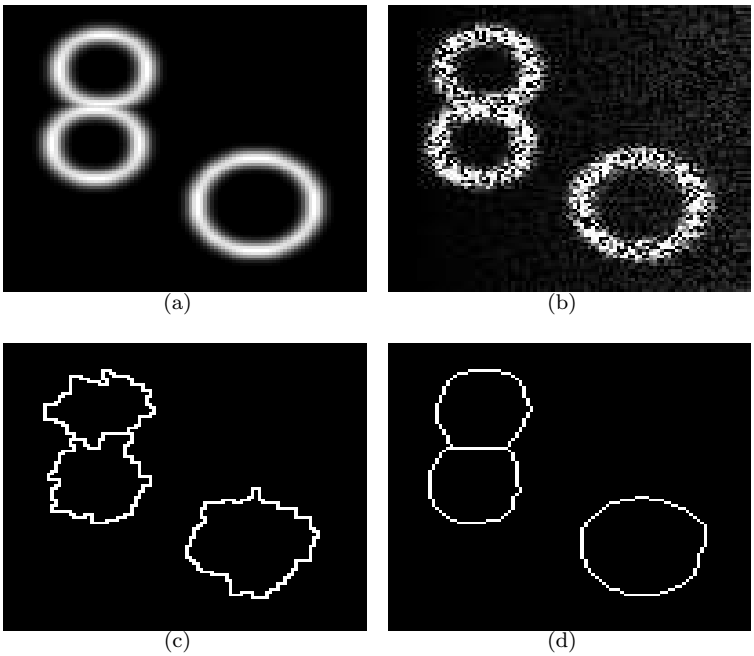
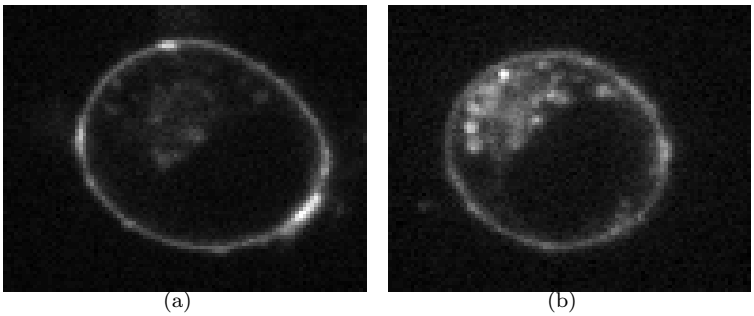


Figure 6: Segmentation of a synthetic, noisy image. Gaussian multiplicative noise and a linear gradient was add to the original image in (a) and the obtained noisy image (b) was segmented using the standard watershed (c) and the PCLS level set watershed (d). Note that the level set watershed has a higher capacity to deal with the noise and the linear gradient than the standard watershed.

5.3 Real data

This section contains experiments involving real cell images taken by fluorescence microscopy showing rat pheochromocytoma PC12 cells [22]. The images are optical planes extracted from 3D stacks. The cells are stained with Wheat Germ Agglutinine (WGA-Alexa Fluor®) which is a lectin that binds N-glycosylated proteins, thus high-lighting the cell membrane. WGA-Alexa Fluor® creates a strong signal from the cell membrane but appears shortly after administration inside the cells due to constitutive endocytosis of the plasma membrane. This causes a significant decrease in the desired signal from the cell membrane and creates correspondingly an increase in undesired signal from inside the cell. Figure 7 shows an example of a PC12 cell shortly (a) after administration of WGA-Alexa Fluor® and the same cell one hour later (b). Clearly, the signal from the cell border decreases and simultaneously a brighter signal from internalized vesicles emerges. Endocytosis is the underlying reason for a vast majority of all mis-segmentations in our images, in contrast to the

Gaussian noise which represents a minor problem. In this work, we shall use these real images to test the ability of our proposed algorithms.

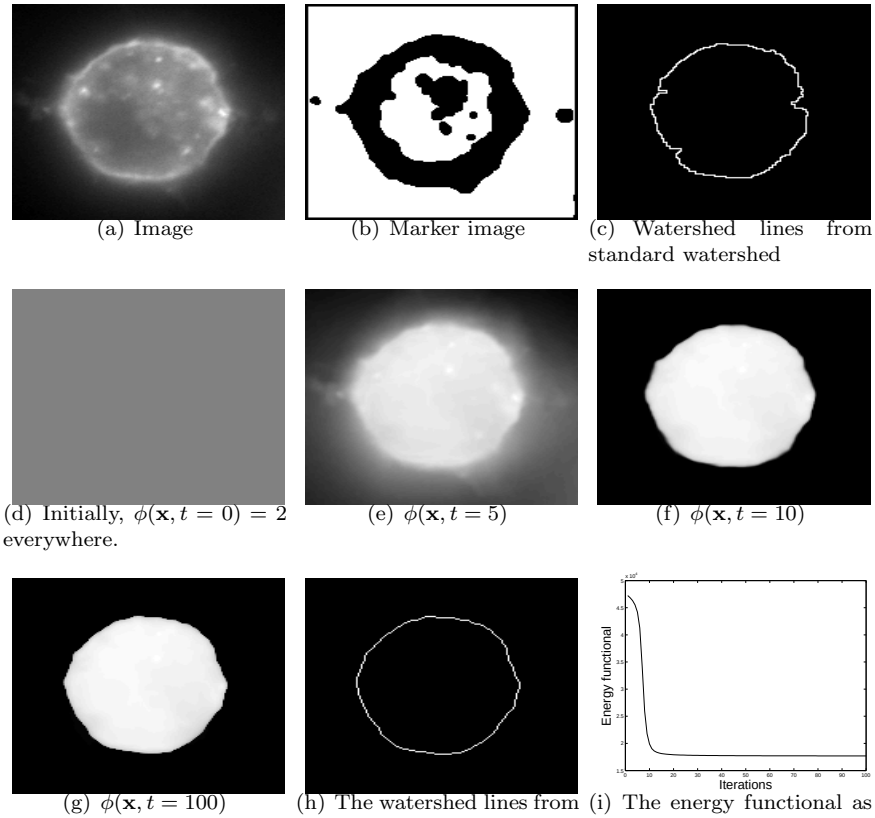


(a) (b)

Figure 7: Endocytosis of WGA. Shortly after administration of WGA-Alexa Fluor® there is a sharp and clear signal from the cell border (a). One hour later, WGA-Alexa Fluor®-stained intracellular vesicles, deriving from endocytosis of small portions of the plasma membrane, create a strong signal from inside the cell and at the same time depleting the signal strength from the desired cell border (b). These processes of endocytosis are responsible for demanding tasks of segmentation.

5.4 Convergence in time, $t \to \infty$

This example contains one cell in addition to background, and it demonstrates the influence of endocytosis on the segmentation. The image in Fig 8(a) was used for segmentation. The method described §2.2 was used to obtain the marker image (b). The white regions are the markers. Based on the marker image (b), the segmentation using the standard watershed (c) has oscillating boundaries, particularly where the endocytosed particles in the original image are close to the boundary. A segmentation was also performed using the PCLS method. The evolution of the level set function is shown for t = 0 (d), t = 10 (e) t = 50 (f) and t = 100 (g). Note that the level set function approaches two piecewise constant regions $\phi \to \{1 \text{ (dark)}, 2 \text{ (bright)}\}$ at convergence. The interface labels the watershed lines (h). The watershed lines in the level set watershed approach are smoother than in (c). Evidently this method is more resistant to the influence of endocytoced high intensity particles than the standard watershed. The panel in (i) shows the convergence of the energy functional versus the number of iterations (time t).



(a) Image (b) Marker image (c) Watershed lines from standard watershed

(d) Initially, $\phi(\mathbf{x}, t = 0) = 2$ everywhere. (e) $\phi(\mathbf{x}, t = 5)$ (f) $\phi(\mathbf{x}, t = 10)$

(g) $\phi(\mathbf{x}, t = 100)$ (h) The watershed lines from level set watershed, 100 iterations (i) The energy functional as a function of iterations

Figure 8: Watershed segmentation of one cell and a background region. The image in (a) was used to create the marker image (b) which was used for a standard watershed segmentation (c) and a level set watershed segmentation using the Piecewise constant level set (PCLS). The level set function ϕ is shown at different times in (d) $t = 0$) (e) $t = 5$, (f) $t = 20$ and (g) $t = 100$, showing that the level set function approaches piecewise constant values. In this case, $\phi \rightarrow \{1 \text{ (dark)} , 2 \text{ (bright)}\}$. The obtained watershed lines from the PCLS are displayed in (h), apparently smoother than the watershed lines obtained for the standard watershed (c). The panel (i) shows the energy functional (Eq. 14) of the PCLS versus the iterations. The parameter settings for the level set watershed are $\lambda = 0.05,\ \Delta t = 0.1,\ \gamma = 0.1,\ \epsilon = 0.1$.

5.5 The regularization parameter λ

The image in Fig 9(a) represents a challenging task of segmentation where the signal from the cell membrane partly disappears or becomes blurry. It shows two attached cells with an inhomogeneously distributed membrane marker, which is the reason for the inhomogeneous signal. The automatically generated marker image in (b) was used for segmentation by the standard watershed (c) and

the Piecewise constant level set watershed (PCLS) (d-f). The regularization parameter λ in the level set watershed was given different values (d) $\lambda = 0$, (e) $\lambda = 0.01$ and (f) $\lambda = 0.1$, to demonstrate how λ affects the smoothness of the final watershed lines. Apparently, higher values of λ produce smoother watershed lines, which is to be expected. For a suitably chosen value of λ in (f), the level set watershed segmentation produces smoother boundaries than the standard watershed (c). Generally, a smooth solution is closer to the true boundaries of the cells than an oscillating solution.

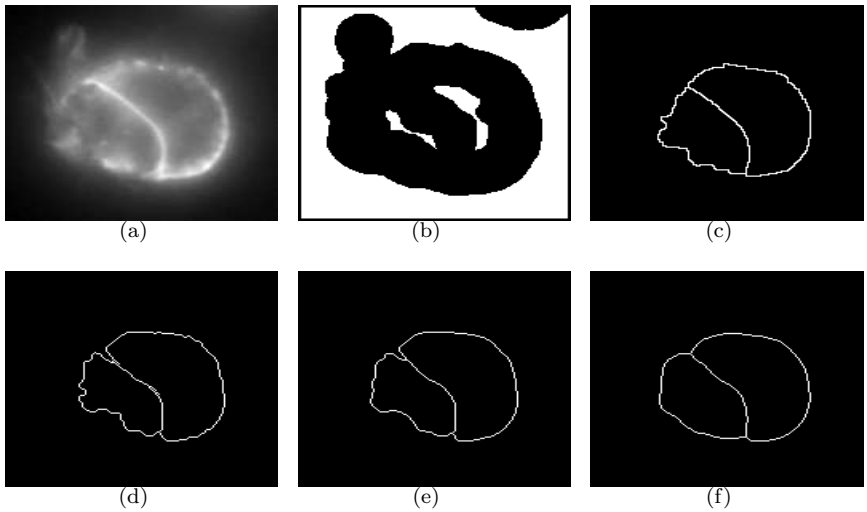


Figure 9: Watershed segmentation of two cells and a background region. The image in (a) was used to automatically obtain the markers in (b) using the method of §2.2. The marker was used for a standard watershed segmentation (c) and a level set watershed segmentation (d-f). The PCLS watershed segmentation was performed with increasing values of the regularization parameter λ in (d) $\lambda = 0$, (e) $\lambda = 0.01$ and (f) $\lambda = 0.1$ to demonstrate the effect of λ. Note how increasing values of λ create smoother watershed lines. It is also clear that the level set watershed in (f) creates smoother watershed lines than the standard watershed in (c). This example was created using parameter settings of $\Delta t = 0.1,\ \gamma = 0.1,\ \epsilon = 0.1, t = 300$.

5.6 Comparing the three level set models

The three level set approaches and the standard watershed segmentation produce similar results applied to images with weak noise. This is shown in Fig 10, where the high-quality image in (a) was used to automatically create the marker image (b). Based on the marker image, a standard watershed segmentation was performed (c) and also the three level set approaches, the Chan-Vese model (d), the Binary level set (e) and the PCLS (f). Evidently, all four models produce very similar segmentation results.

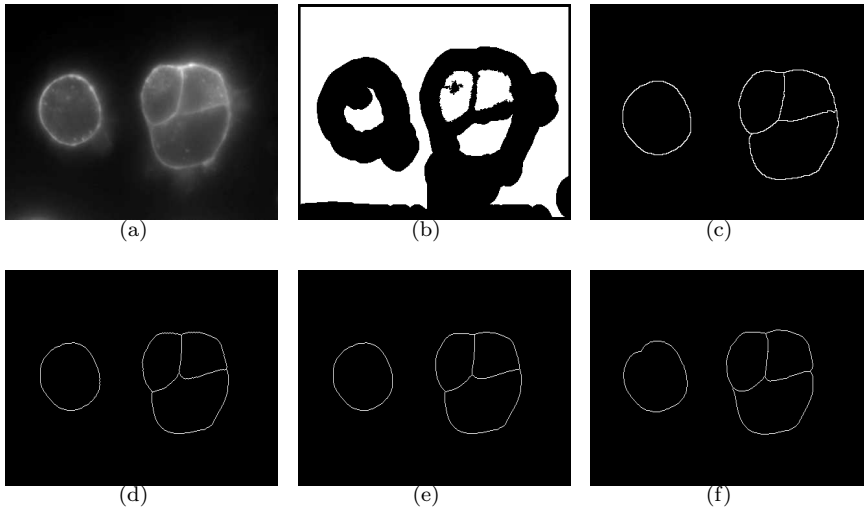


Figure 10: Watershed segmentation of four cells and a background region. The high-quality image in (a) with only weak noise was used to create the marker image (b). The marker image was used as input for a standard watershed segmentation (c) and for level set watershed segmentation (d-f) using the Chan-Vese model (d), the Binary level set (e) and the PCLS (f). Apparently, applied to images of high quality containing weak noise signals, all four models behave very similar. The examples were run with parameter settings of (d) $\Delta t = 0.05,\ \lambda = 0.1, t = 200$, (e) $\Delta t = 0.1,\ \lambda = 0.05,\ \sigma = 0.1, t = 200$ and (f) $\Delta t = 0.1,\ \lambda = 0.1, \gamma = 0.02, \epsilon = 0.5, t = 300$.

5.7 Regularization applied to the level set function(s) or the characteristic functions

The regularization term enables a smoothing of the watershed lines. This is normally useful for segmentation of real images. However, a smoothing of the watershed lines could also smooth junctions between cells, which is not always appropriate.

In (8), we have used

$$R_1(\phi_1, \phi_2) = \int_\Omega \sum_{i=1}^{4} |\nabla \psi_i| d\mathbf{x}. \tag{19}$$

as the regularization functional. At convergence, $\int_\Omega |\nabla \psi_i| d\mathbf{x}$ is the length of the boundary of region Ω_i. In the original paper [47, 6], the regularization functional was different, i.e.

$$R_2(\phi_1, \phi_2) = \int_\Omega \sum_{i=1}^{2} |\nabla \phi_i| d\mathbf{x}. \tag{20}$$

From our numerical experiments, we have observed some interesting phenomena with these two different regularizations. When we increase the regularization

parameter λ, we would expect that the watershed lines are getting smoother. This is observed in the experiments, especially see Fig. 11 which is segmented in Fig. 12 for different values of λ and using two different regularization functionals R_1 and R_2. However, the junctions behave differently. For R_2, the corner of the junction in the upper part of the watershed line is getting sharper when the values of λ is getting bigger. While for R_1, the junction is getting smoother when the values of λ is getting bigger. For the PCLS, we can also replace R_1 by

$$R_2(\phi) = \int_\Omega |\nabla \phi| d\mathbf{x}. \tag{21}$$

These two functionals produce different results. In [25], it was observed that R_1 can treat triple-junctions in a proper manner. It is known that R_2 given in (20) and (21) are not able to get symmetric triple junctions. Historically, regularization functional R_2 was first proposed in [6], while R_1 was first proposed in [28, 26, 27].

Oscillating watershed lines are not preferred for segmentation. However, sharp corner may be preferred in some situations. These experiments show that R_1 should be used for these applications.

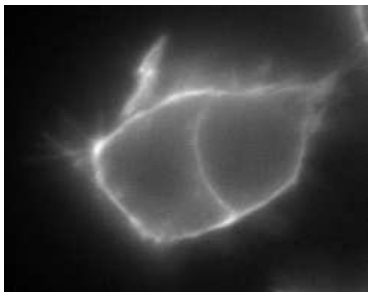
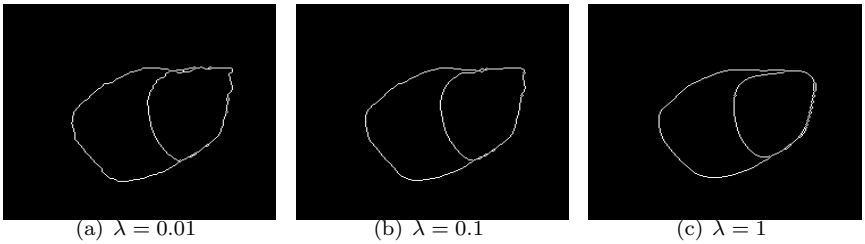


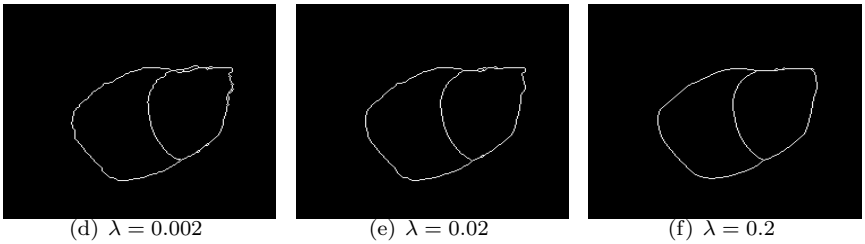
Figure 11: The image used in Fig. 12 to demonstrate the difference between regularization functionals R_1 and R_2.

5.8 Challenging situations in real images

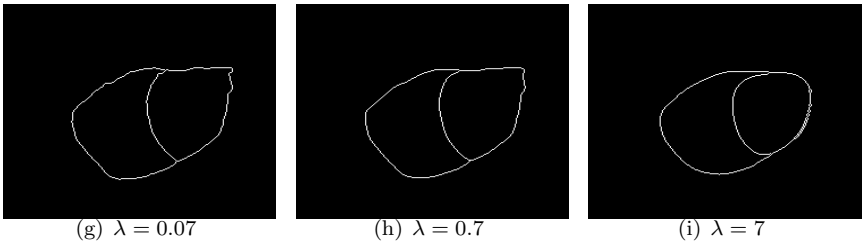
This example shows three PC12 cells with partly inhomogeneously labeled cell membrane. Therefore, it represents a challenging segmentation, approaching the limits of both the standard watershed and the level set watershed segmentation. The image in Fig 13(a) was used to create markers automatically (b). The marker image was then applied as input for the standard watershed (c) and also the Binary watershed level set (d-f). The level set functions ϕ_1 and ϕ_2 are displayed in (d) and (e), respectively. The sign of the level set functions is responsible for creating the four phases. The boundaries of the four phases are drawn in (f), showing the final segmentation. Clearly, the watershed level set (f) produces a better segmentation than the standard watershed (c), although none of them is perfect.



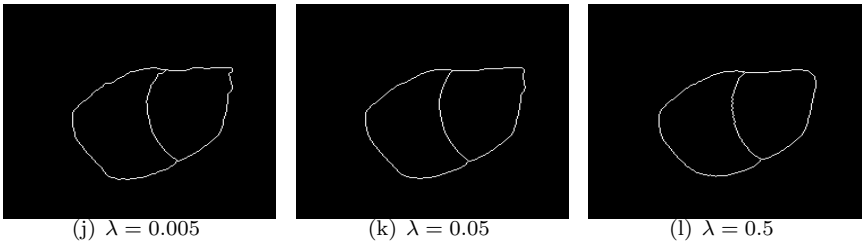
(a) $\lambda = 0.01$ (b) $\lambda = 0.1$ (c) $\lambda = 1$



(d) $\lambda = 0.002$ (e) $\lambda = 0.02$ (f) $\lambda = 0.2$



(g) $\lambda = 0.07$ (h) $\lambda = 0.7$ (i) $\lambda = 7$



(j) $\lambda = 0.005$ (k) $\lambda = 0.05$ (l) $\lambda = 0.5$

Figure 12: The difference between regularization functionals R_1 and R_2, demonstrated on the image in Fig. 11. From left to right, the regularization parameter λ was multiplied by 10. Pictures (a-c) were created using the Chan-Vese method (see §4.1) with the regularization functional $R_2(\phi_1, \phi_2)$, and (d-f) were created using regularization functional $R_1(\phi_1, \phi_2)$. Pictures (g-i) were computed using the Binary level set (see §4.2) with the regularization functional $R_2(\phi_1, \phi_2)$ and and (j-l) were created using regularization functional $R_1(\phi_1, \phi_2)$. The right column (largest λ) clearly shows the difference between the regularization functionals. $R_2(\phi_1, \phi_2)$ creates sharper corners with the junctions than the regularization functional $R_1(\phi_1, \phi_2)$. In the test, $\Delta t = 0.005$ for all examples (a-l).

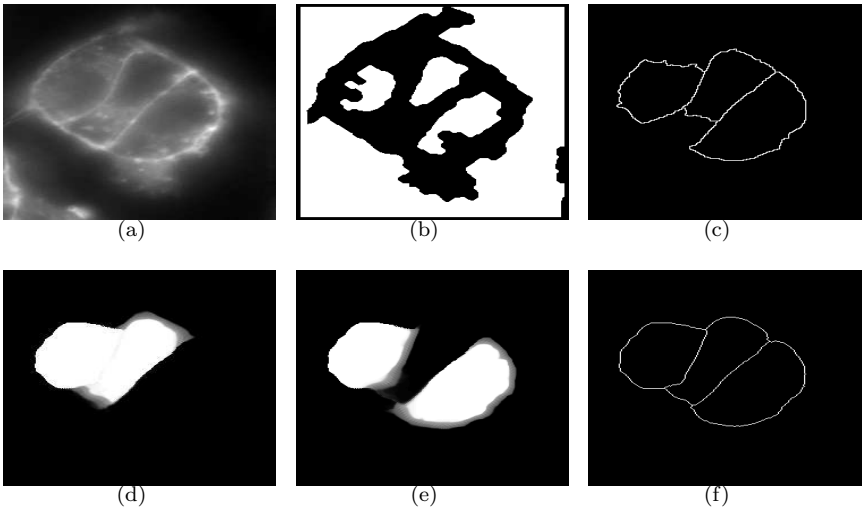


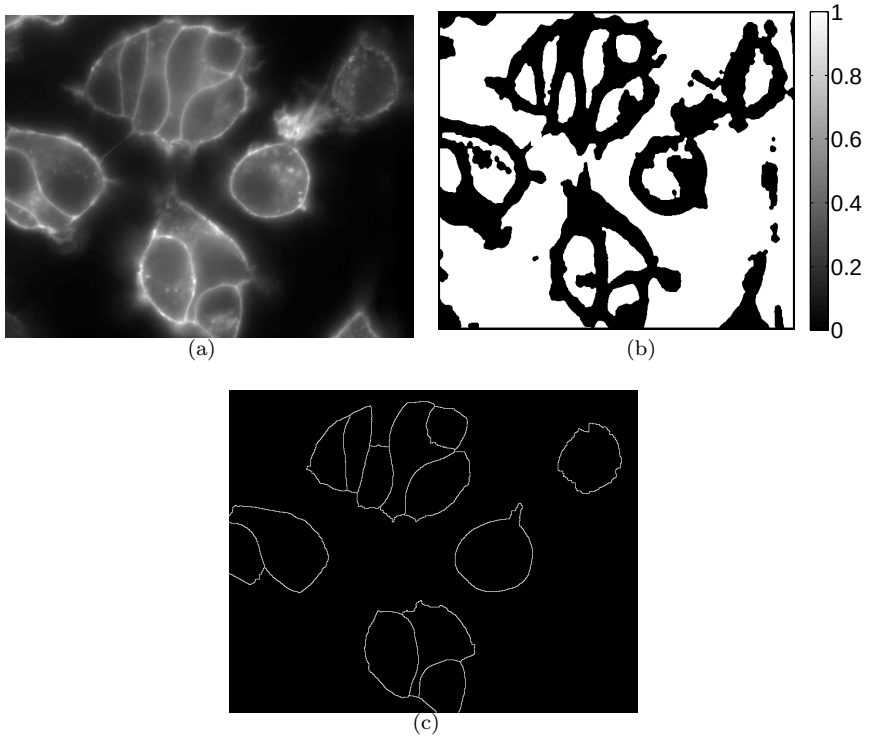
Figure 13: The image in (a) was applied to create markers (b) for a standard watershed segmentation (c) and also a watershed level set segmentation (d-f). The four combinations of the sign of the level set functions ϕ_1 (d) and ϕ_2 (e) settle the boundaries (f) of the characteristic functions ψ_i. Note that the level set watershed (f) obtains a better segmentation result than the standard watershed (c). These examples were executed with parameter settings of $\Delta t = 0.1, t = 200$ and $\lambda = 0.1$.

5.9 Multiple objects and the Four-Color theorem

Selecting high-quality images in high-throughput experiments may involve a too high degree of effort. Consequently, segmentation of lower quality images is often a necessity. Therefore it is important to validate the segmentation protocol for such images as well, and not solely for high-quality images. The image in Fig 14 (a) shows one optical plane from an image stack. It represents a severe challenge for segmentation since the boundaries are partly blurred, broken and there exists a significant amount of endocytosed particles. This image also demonstrates the use of the Four-Color coding since this example requires the use of all four colors. The image in Fig 14(a) was used to automatically construct the marker image Fig 14(b) which was used as input for the standard watershed segmentation. The obtained watershed lines are shown in Fig 14(c).

To compute the level set watershed segmentation, the Euclidean influence zones f_{IZ} (§3.1) were constructed based on the marker image in Fig 14(b). The Euclidean influence zones were computed with the purpose of grouping the markers in at most four groups according to the Four-Color theorem. The result of this grouping is shown in Fig 15(a), where all markers are assigned an integer from $1 \rightarrow 4$. The markers possessing the same integer value belong to the same group. Note that all markers within each group are non-adjacent, which is the crucial point. If they were adjacent, it would not be possible to separate all markers within the same color. The topographical distance function

was calculated around the set of all markers inside each color, and the level set watershed was computed using the Binary level set. The obtained level set functions ϕ_1 and ϕ_2 are shown in Fig 15(b) and Fig 15(c), respectively. The two level set functions approach binary values of 0 (black) and 1 (white). The watershed lines from the level set watershed are shown in Fig 15(d). Note how the level set watershed (Fig 15(d)) produces smoother watershed lines than the standard watershed (Fig 14(c)). The level set watershed also captures more of the cells than the standard watershed, selected cells are indicated with asterisks.



(a) (b) (c)

Figure 14: An optical plane from an image stack (a) and the automatically generated binary marker image (b) which was used for a standard watershed segmentation (c).

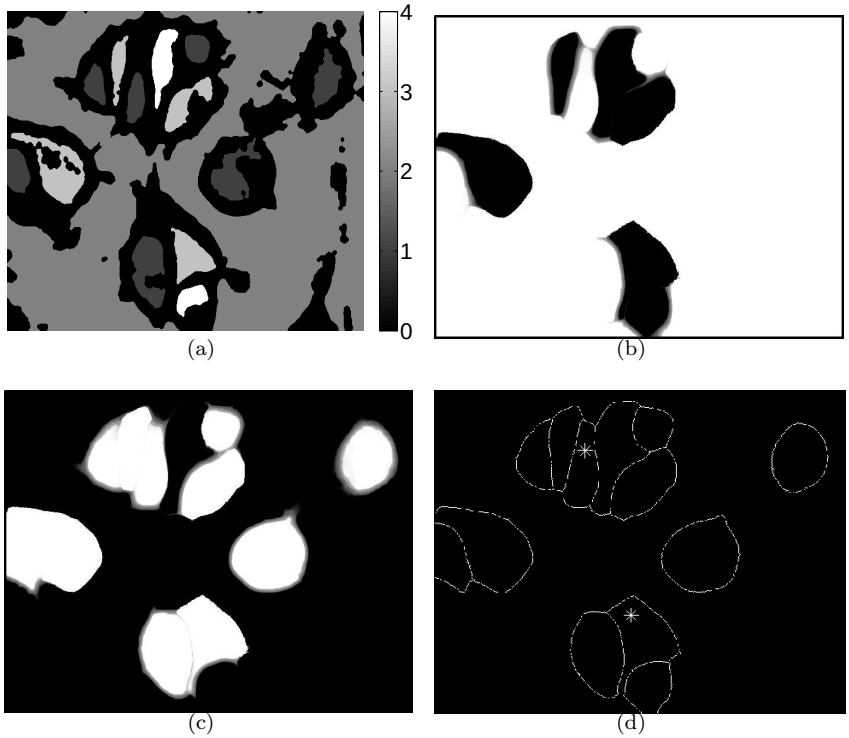


Figure 15: Segmentation of a high number of cells and demonstration of the Four-Color coding. To compute the level set watershed, the Euclidean influence zones f_{IZ} were calculated around the marker regions in Fig 14 (b), where each region in f_{IZ} represents a specific marker region. The Four-Color theorem was applied to f_{IZ}, grouping the corresponding markers into four groups $\{1, 2, 3, 4\}$, shown in (a) where the background has the value zero and all markers belonging to the same group have the same integer value. The Binary level set watershed segmentation was calculated, and the level set functions ϕ_1 and $\phi_2 \rightarrow \{0, 1\}$ at convergence are displayed in (b) and (c). The watershed lines from the level set watershed are shown in (d). Note how the level set watershed (d) produces a smoother result than the standard watershed in Fig 14(c). The asterisks indicate selected cells where the level set watershed captures more of the true cell than the standard watershed. The level set watershed in this example was executed with parameter settings $\Delta t = 0.1, t = 200, \lambda = 0.2, \sigma = 0.01$, and the regularization was applied to the characteristic functions ψ_i.

6 Conclusion

In this work we have combined the level set method [37, 47, 6, 28, 26, 27] and the marker-controlled watershed segmentation [33, 31, 48, 49] to develop a method for segmentation of real cells and other structures of similar nature. A set of markers, also called initialization regions, were automatically created

by adaptive thresholding and iterative filling (§2.2). Such automated methods are of high value in high-throughput experiments and other experimental setups producing large amount of data. Based on the markers, the watershed distance transform was computed from the topographical distance function [31]. We then used the Four-Color theorem to group the markers into a maximum of four groups, thus reducing the complexity of the problem. Inspired by [34], we propose to compute the watershed lines around each group of markers using three different level set approaches, §4.1 - §4.3. These level set methods were tested for real and synthetic images of different complexity, and containing a large number of cells. The experimental results show that all three level set methods are able to perform a good segmentation of the given images.

References

[1] K. I. Appel and W. Haken. Every planar map is four colorable. *Illinois J. Math.*, 21:429–567, 1977.

[2] Pablo Andrés Arbeléz and Laurent D. Cohen. Energy partitions and image segmentation. *J. Math. Imaging Vis.*, 20(1-2):43–57, 2004.

[3] Pascal Bamford and Brian Lovell. Unsupervised cell nucleus segmentation with active contours. *Signal Process.*, 71(2):203–213, 1998.

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

[5] Antonin Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20(1-2):89–97, 2004. Special issue on mathematics and image analysis.

[6] T. Chan and L. Vese. Active contours without edges. *IEEE Trans on Image Processing*, 10:266–277, 2001.

[7] Tony F. Chan, Mark Moelich, and Berta Sandberg. Some recent developments in variational image segmentation. In X.-C. Tai, K. A. Lie, T. Chan, and S. Osher, editors, *Image Processing based on partial differential equations*, pages 175–201. Springer, Heidelberg, 2006.

[8] Tony F. Chan and Xue-Cheng Tai. Level set and total variation regularization for elliptic inverse problems with discontinuous coefficients. *J. Comput. Phys.*, 193(1):40–66, 2004.

[9] S. Grace Chang, Bin Yu, and Martin Vetterli. Spatially adaptive wavelet thresholding with context modeling for image denoising. *IEEE Transactions on Image Processing*, 9(9):1522–1531, 2000.

[10] S.Y. Chien, Y.W. Huang, and L.G. Chen. Predictive watershed: a fast watershed algorithm for video segmentation. *CirSysVideo*, 13(5):453–461, May 2003.

[11] Oddvar Christiansen and Xue-Cheng Tai. Fast implementation of piecewise constant level set methods. In X.-C. Tai, K. A. Lie, T. Chan, and S. Osher, editors, *Image Processing based on partial differential equations*, pages 289–308. Springer, Heidelberg, 2006.

[12] Ginmo Chung and Luminita A. Vese. Energy minimization based segmentation and denoising using a multilayer level set approach. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757, pages 439–455. Springer Berlin, Heidelberg, 2005.

[13] Daniel Cremers, Florian Tischhäuser, Joachim Weickert, and Christoph Schnörr. Diffusion snakes: introducing statistical shape knowledge into the Mumford–Shah functional. *International Journal of Computer Vision*, 50:295 – 313, 2002.

[14] Jérôme Darbon and Marc Sigelle. Image restoration with discrete constrained total variation. I. Fast and exact optimization. *J. Math. Imaging Vision*, 26(3):261–276, 2006.

[15] Petr Felkel, Mario Bruckschwaiger, and Rainer Wegenkittl. Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest. *Computer Graphics Forum*, 20:2001, 2002.

[16] Ying-Lun Fok, Joseph C. K. Chan, and Roland T. Chin. Automated analysis of nerve-cell images using active contour models. *IEEE Transactions on Medical Imaging*, 15(3), 1996.

[17] S. Gautama, W. Goeman, and J. D'Haeyer. Robust detection of road junctions in vhr images using an improved ridge detector. *The International Archives of Photogrammetry, Remote sensing and Spatial Information Sciences*, 34, 2004.

[18] Matthias Gebhard, Julian Mattes, and Roland Eils. An active contour model for segmentation based on cubic b-splines and gradient vector flow. In *MICCAI '01: Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 1373–1375, London, UK, 2001. Springer-Verlag.

[19] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[20] Vicente Grau, Andrea J. U. Mewes, Mariano Alcaiz Raya, Ron Kikinis, and Simon K. Warfield. Improved watershed transform for medical image segmentation using prior information. *IEEE Trans. Med. Imaging*, 23(4):447–458, 2004.

[21] Y. M. Jung S. H., Kang, and J. Shen. Multiphase image segmentation via modica-mortola phase transition. *SIAM JOURNAL ON APPLIED MATHEMATICS*, 67(5):1213–1232, 2006.

[22] Erlend Hodneland, Arvid Lundervold, Steffen Gurke, Xue-Cheng Tai, Amin Rustom, and Hans-Hermann Gerdes. Automated detection of tunneling nanotubes in 3d images. *Cytometry Part A*, 69A:961–972, 2006.

[23] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, V1(4):321–331, January 1988.

[24] Hongwei Li and Xue-Cheng Tai. Piecewise constant level set method for interface problems. In *Free boundary problems*, volume 154 of *Internat. Ser. Numer. Math.*, pages 307–316. Birkhäuser, Basel, 2007.

[25] Hongwei Li and Xue-Cheng Tai. Piecewise constant level set method for multiphase motion. *Int. J. Numer. Anal. Model.*, 4(2):291–305, 2007.

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

[27] Johan Lie, Marius Lysaker, and Xue-Cheng Tai. A piecewise constant level set framework. *Int. J. Numer. Anal. Model.*, 2(4):422–438, 2005.

[28] Johan Lie, Marius Lysaker, and Xue-Cheng Tai. A variant of the level set method and applications to image segmentation. *Mathematics of computation*, 75(255):1155–1174, 2006.

[29] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting up method and its application to Navier-Stokes equations. *Appl. Math. Lett.*, 4(2):25–29, 1991.

[30] T. Lu, P. Neittaanmäki, and X.-C. Tai. A parallel splitting-up method for partial differential equations and its applications to Navier-Stokes equations. *RAIRO Modél. Math. Anal. Numér.*, 26(6):673–708, 1992.

[31] Fernand Meyer. Topographic distance and watershed lines. *Signal Processing*, 38(1):113–125, 1994.

[32] David Mumford and Jyant Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure Applied Mathematics*, 42:577–685, 1989.

[33] Laurent Najman and Michel Schmitt. Watershed of a continuous function. *Signal Process.*, 38(1):99–112, 1994.

[34] H.T. Nguyen, M. Worring, and R. van den Boomgaard. Watersnakes: Energy-driven watershed segmentation. *IEEE Trans. on PAMI*, 25(3):330–342, 2003.

[35] Lars Kristian Nielsen, Xue-Cheng Tai, Sigurd I. Aanonsen, and Magne Espedal. Reservoir description using a binary level set model. In X.-C. Tai, K. A. Lie, T. Chan, and S. Osher, editors, *Image Processing based on partial differential equations*, pages 403–426. Springer, Heidelberg, 2006.

[36] Lars Kristian Nielsen, Xue-Cheng Tai, Sigurd Ivar Aanonsen, and Magne Espedal. A binary level set model for elliptic inverse problems with discontinuous coefficients. *Int. J. Numer. Anal. Model.*, 4(1):74–99, 2007.

[37] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[38] Víctor Osma-Ruiz, Juan I. Godino-Llorente, Nicolás Sáenz-Lechón, and Pedro Gómez-Vilda. An improved watershed algorithm based on efficient computation of shortest paths. *Pattern Recogn.*, 40(3):1078–1090, 2007.

[39] C. Rambabu and I. Chakrabarti. An efficient immersion-based watershed transform method and its prototype architecture. *J. Syst. Archit.*, 53(4):210–226, 2007.

[40] N. Robertson, D. Sanders, P. Seymour, and R. Thomas. A new proof of the four colour theorem. *Electronic Research Announcements of the American Mathematical Society*, 2(1), 1996.

[41] J. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization techniques. *Institute for Mathematics and Computer Science, University of Groningen, Groningen, The Netherlands, IWI 99–9-06*, 1999.

[42] B. Song and T. Chan. Fast algorithm for level set based optimization. *UCLA CAM Report, CAM-02-68*, 2002.

[43] Xue-Cheng Tai and Tony F. Chan. A survey on multiple level set methods with applications for identifying piecewise constant functions. *Int. J. Numer. Anal. Model.*, 1(1):25–47, 2004.

[44] Xue-Cheng Tai, Erlend Hodneland, Joachim Weickert, Nickolay V. Buroresthliev, Arvid Lundervold, and Hans-Hermann Gerdes. Level set methods for watershed image segmentation. In *Scale Space and Variational Methods in Computer Vision, LNCS 4485*, pages 178–190. Springer, 2007.

[45] Xue-Cheng Tai and Hongwei Li. A piecewise constant level set method for elliptic inverse problems. *Appl. Numer. Math.*, 57(5-7):686–696, 2007.

[46] Xue-cheng Tai and Chang-hui Yao. Image segmentation by piecewise constant Mumford-Shah model without estimating the constants. *J. Comput. Math.*, 24(3):435–443, 2006.

[47] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, 2002.

[48] Luc Vincent and Edward R Dougherty. *Digital Image Processing Methods*, chapter Morphological Segmentation for Textures and Particles, pages 43–102. E. Dougherty, Editor, Marcel-Dekker, New York, 1994.

[49] Luc Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, 1991.

[50] P. Wei and M.Y. Wang. A piecewise constant level set method for structural shape and topology optimization. In *7th World Congress of Structural and Multidisciplinary Optimization, Seoul, Korea*, 2007.

[51] J. Weickert, B. Romeny, and M. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, March 1998.

[52] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, pages 359–369, 1998.