Segmenting Cellular Retinal Images by Optimizing Super-pixels, Multi-level Modularity, and Cell Boundary Representation

Oscar Cuadros Linares*, Bernd Hamann, and João Batista Neto

***Abstract*—We introduce an interactive method for retina layer segmentation in gray-level and RGB images based on super-pixels, multi-level optimization of modularity, and boundary erosion. Our method produces highly accurate segmentation results and can segment very large images. We have evaluated our method with two datasets of 2D confocal microscopy (CM) images of a mammalian retina. We have obtained average Jaccard index values of 0.948 and 0.942 respectively, confirming the high-quality segmentation performance of our method relative to a known ground truth segmentation. Average processing time was two seconds.**

***Index Terms*—Image segmentation, Modularity optimization, Super-pixels, Confocal microscopy, Retina**

I. Introduction

.

Segmentation is one of the most important tasks in image processing and aims to identify objects of interest in images. In general, objects of interest are defined by some criteria of the underlying application. For example, in medical analysis, objects of interest are usually bones, organs, tissue, cellular structures, or retina layers [1].

High-quality image segmentation still is a challenge problem. Graph clustering methods developed for image segmentation have become increasingly important in recent years as they can be used to obtain high-accuracy segmentation results. However, graph- and network-based segmentation methods can be computationally expensive due to the data sizes involved. Community detection algorithms represent one approach to address the computational complexity issue.

There are many strategies to address the problem of image segmentation, and the choice also depends on the application. For instance, there exist methods for segmenting images automatically where no human intervention nor labeled data are necessary. Such strategies are usually applied on natural-scene image segmentation [2]. Some other methods rely on labeled datasets to guide the segmentation process and normally exhibit good accuracy. Nevertheless, reliable labeled datasets may be hard to obtain [3]. A third group of methods, semi-supervised, employs user interaction to label pixels during execution time to guide the segmentation process, bringing user knowledge into it [4].

Oscar Cuadros Linares, Institute of Mathematics and Computer Sciences (ICMC), University of Sao Paulo (USP) Brazil, SP 13566-590, Phone: +55 (16) 3373-9700. ocuadros@icmc.usp.br

Bernd Hamann, Department of Computer Science, University of California, Davis, CA 95616, U.S.A. hamann@cs.ucdavis.edu

João Batista Neto, ICMC-USP. jbatista@icmc.usp.br

More recently, graph clustering algorithms have successfully been used for segmenting images [5]. Nevertheless, the time complexity of such methods are highly related to the graph cardinality [6], which turns them suitable for segmenting short to medium size images only. However, along with the complex networks theory, a number of new graph clustering (so-called community detection algorithms) have been proposed. The main advantage of these methods is their ability to handle huge graphs (million of vertices) in shorter processing time [7].

Segmentation of complex structures, more specifically, retina layer segmentation, is often done via a combination of different methods. Nhat Vu et al. (2007) [8] proposed a method based on the parametric active contour model of Lobregt and Viergever [9]. Due to the complexity of retina layer contours, this method is based on a modified external force formulation to make the algorithm more insensitive to initialization. Prior knowledge, when available, can also be used to segment complex shapes. Bertelli et al. (2007) [10] introduced a method that uses prior knowledge and a dissimilarity function for pixels to separate objects (retina layers) and background. This method uses labeled reference image as prior information. Another work for retina segmentation based on non-rigid registration using Thin Plate Splines was also proposed by Bertelli et al. [11]. This method uses prior information as atlases of retina layers and then a non-rigid registration is applied to segment a retina layer.

A reliable segmentation of retina layers is the first step to understanding structural and cellular changes in the retina. However, visual irregularities introduced by staining, considerable variation of layer shape, and statistically heterogeneous image characteristics make the segmentation process a challenging task [10]. Due to that, is important to include prior information in the process to produce accurate results. Our proposal uses "prior" information as user-labels, which allow specialist not only include their knowledge but also guide the entire segmentation process.

We introduce an interactive and highly efficient retina layer segmentation method that allows a specialist to guide the segmentation process through manual placement of seeds on the object of interest. Segmentation of very large images can be done in short processing times. Our approach consists of three steps:

1) Super-pixels to reduce not only redundancy in the image, but also processing time.

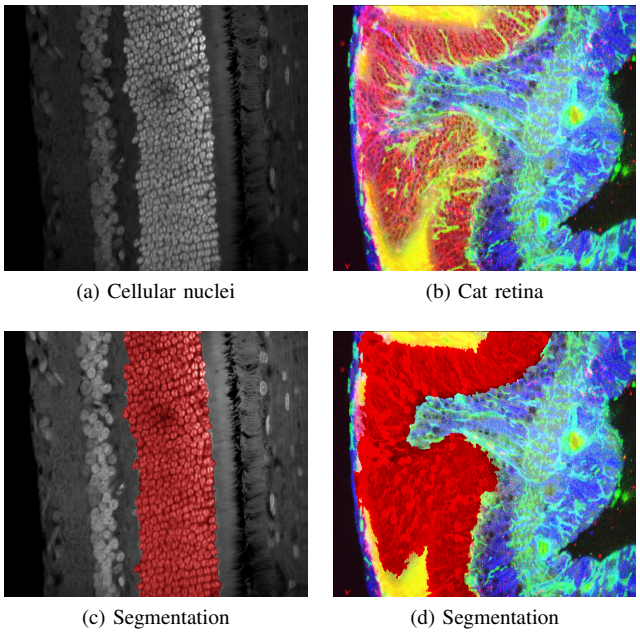
2) Interactive segmentation based-on optimization of modularity in graphs.
3) Morphological sharpening operator to improve the segmentation accuracy in complex contours.

We validate our approach on two datasets of Confocal Microscopy (CM) images [12]. The first is composed of gray-scale images of a vertical section of mammalian retina, which contains many layers with different structures (Figure 1.a). The second dataset is composed of retinal color-channel images of the cellular nuclei (Figure 1.b).

In the image processing context, cell layers in both datasets poses challenging issues [8], [10]:

- Heterogeneous illumination.
- Variation in object shape, size, and orientation.
- Very irregular contours.
- Cell-like texture.

Each image of both datasets has a corresponding segmentation ground truth (GT) data.



(a) Cellular nuclei (b) Cat retina

(c) Segmentation (d) Segmentation

Fig. 1. Example images of the Confocal microscopy of the mammalian retina. In (a) a cellular nuclei, and in (b) vertical section through of the cat retina. Respective segmentation using our proposal, red region in (c) and (d).

The rest of the paper is organized as follows: Section II details important concepts that underpin our proposal. In Section III we preset the method proposed in this paper. Experimental results and validation are discussed in Section IV. Finally the conclusion is presented in Section V.

II. Related Concepts

We define an image Ω as an image consisting of n pixels, and we define the value (gray-scale or color channels) of the i^{th} pixel as Ω_i. The spatial location of pixel i is $\mathbf{x}_i$, where $\mathbf{x}$ is the vector of coordinates. The graph $G = (V, E, W_E)$ is an undirected and weighted graph, where V is the set of vertices, and E the set of edges with associated weights W_E. The weight assigned to edge (i, j) is $w_{ij} \in W_E$.

A. Super-pixels

Super-pixels (called super-voxels in 3D image processing) is a region-based technique that segments an image into groups of pixels that share a similar color and neighboring position [6]. The goal is to capture image redundancy and reduce the number of pixels in the image, thereby reducing the complexity of subsequent processing tasks [13]. Several approaches for super-pixels have been proposed [14], but not all of them can be adapted easily to the 3D super-voxel scenario. We use a color-based version [6] of the super-pixel algorithm, called speeded-up turbo pixels (SUTP) [15], which is efficient and simple, producing high-quality super-pixels. After applying this algorithm to an image, it is possible to eliminate up to 90% of the pixels [6].

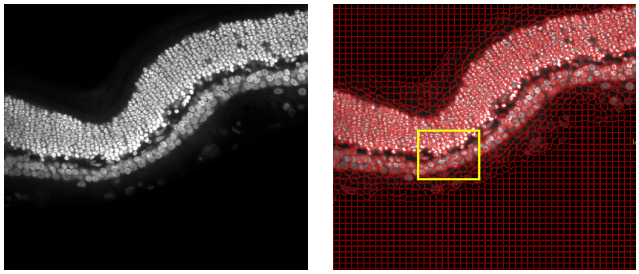
The SUTP method is based on the k-means clustering algorithm. First, a given image is divided into N regular super-pixels Ω_I ($I \in N$), with edge length l. Initially, all super-pixels have the same area and equidistant centers. For simplicity, we assume that the minimal bounding box of the entire image is itself a square. Pixels along a certain super-pixel boundary are iteratively analyzed and possibly shifted across the border between neighboring super-pixels. This pixel-swapping process is based on this cost function to be minimized:

$$C_{i,I} = p_1 \|\Omega_i - \Omega_I\|^2 + p_2 \|\mathbf{x}_i - \mathbf{x}_I\|^2, \quad (1)$$

where Ω_I is the mean intensity (RGB vector for multichannel images) value of the I^{th} super-pixel, and $\mathbf{x}_I$ is the spatial center of the I^{th} super-pixel. The parameters p_1 and p_2 are weights for color similarity and super-pixel border rigidity, respectively. "More convex" super-pixels are generated for high values of p_2, although super-pixels may not accurately fit to object boundaries. Determining near-optimal values for p_1 and p_2 depends on the application. In our method, super-pixels adjusted well to boundaries are preferable to convex ones. Hence, we use the values $p_1 = 1$, $p_2 = 0.5$, and the number of iterations is chosen as $it = 5$, as suggested in [2]. Defining the initial size of the super-pixels l also depends on the application, in Section IV-A, we provide an intensive evaluation of this parameter. Figure 2 shows the super-pixels after convergence.

B. Community Detection in Graphs

The "community detection" approach has its origin in complex network theory and is closely related to the well-known problem of graph clustering [7]. The goal of graph clustering methods is to find the "best" division of a graph into clusters of vertices. Finding such a division is an NP-hard problem. Nevertheless, there exist methods to approximate the optimal solution, including Laplacian partitioning, Max-flow, and community detection [7]. Unfortunately, Laplacian partitioning and max-flow methods are highly dependent on the graph cardinality (number of vertices and edges), and their application may be restricted to small graphs [7]. Community detection algorithms can deal with much larger graphs (millions of vertices). Most strategies employed are greedy, based



(a) Input (b) Super-pixels

(c) Close-up view of yellow frame in (b)

Fig. 2. Super-pixel convergence after five iterations.

on optimization of modularity, e.g., the Newman algorithm, the fast greedy method, and multi-level optimization [16]–[18].

Modularity: Newman et al. (2004) [16] proposed a metric aimed at measuring the clustering "quality" inherent in graphs. Basically, the modularity evaluates the trade-off between the number of edges within clusters (internal edges) and the expected number of edges. This idea is mathematically described as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - P_{ij} \right] \delta(C_i, C_j), \tag{2}$$

where A_{ij} is an element of the adjacency matrix $\mathbf{A}$ of the graph G, C_i is the cluster that vertex i belongs to, and the function $\delta(C_i, C_j)$ returns 1 if $C_i = C_j$ and zero otherwise. The term P_{ij} defines the probability of an edge to exist between vertices i and j, defined as:

$$P_{ij} = \frac{k_i k_j}{2m}, \tag{3}$$

where k_i is the sum of weights of edges linked to vertex i, and $m \in E$ is the sum of all edge weights. Notice, for unweighted graphs m is the number of edges in E (edge cardinality), and k_i is the number of edges linked to i (vertex degree).

According to the modularity criterion, a good division of a graph into clusters is one where the number of internal edges is higher than the number of external ones (edges connecting clusters). In that case, the value of Q is greater than zero, which indicates deviation from randomness. In practice, a value of $Q \geq 0.3$ is a good indicator of a strong clustering structure [17].

C. Multi-level Optimization of Modularity

Since the maximization of modularity is an NP-Hard problem [18], several algorithms have been proposed to approximate the optimal solution [7]. A fast and popular method is the fast greedy (FG) algorithm [17], which can achieve high values of Q. Nevertheless, the method proposed by Blondel et al. (2008), called multi-level optimization of modularity (MOM) [18], yields higher values of Q than the FG algorithm. The, MOM is also a greedy approach and consists of two steps:

1) *Modularity Optimization:* Every vertex $i \in V$ is labeled as a cluster of one element only. For each neighbor j of i, the gain of modularity is evaluated (Equation 4) by assigning the cluster of j to i. The vertex i is placed in the cluster for which the gain of modularity is maximal, but only when the gain is positive. This process is repeated for all vertices until no more improvement occurs.
2) *Cluster Aggregation:* Construct a new weighted graph, where vertices are the clusters resulting from the first step. The edge weights are given by the sum of external edges (edges connecting vertices that belong to different clusters), and internal edges lead to a self-loop (an edge that connects a vertex to itself), with the weight given by the sum of its respective internal weights.

The gain in modularity is computed by moving vertex i into a neighboring cluster C, defined as:

$$\begin{aligned} \Delta Q = & \left[\frac{\sum_{in} + k_{i,in}}{2m} - \left(\frac{\sum_{tot} + k_i}{2m} \right)^2 \right] \\ & - \left[\frac{\sum_{in}}{2m} - \left(\frac{\sum_{tot}}{2m} \right)^2 - \left(\frac{k_i}{2m} \right)^2 \right], \end{aligned} \tag{4}$$

where $\sum_{in}$ is the sum of internal edges of cluster C, $\sum_{tot}$ is the sum of weights of edges incident to vertices in C, k_i is the degree of vertex i, $k_{i,in}$ is the sum of weights of edges from i to vertices in C, and m is the sum of the weights of all edges.

D. Seed-based Multi-level Optimization of Modularity

The original MOM algorithm automatically determines both the number of clusters and their sizes. This can be a useful feature in applications in which prior information is not available. Nevertheless, in our application, it is important to include information about the object of interest (cellular layer) in order to limit the number of clusters up to two and guide the clustering process using manually labeled vertices as "seeds".

Let F be the set of foreground and B the set of background vertices with labels x_F and x_B, respectively. As in the original MOM algorithm, the graph is initially divided into as many clusters as there are vertices. During *modularity optimization*, vertex labels are stored in a *membership* vector $\mathbf{x}$ of size n that defines the current cluster of each vertex $i \in V$. Originally, $\mathbf{x}$ is defined by

$$x_i = i. \tag{5}$$

Labels change at each modularity optimization step.

To include the seed vertices in the process, we define a seed vector $\mathbf{b}$ of size n:

$$b_i = \begin{cases} 1, & \text{if } i \in x_F, \\ 2, & \text{if } i \in x_B \ , \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and we re-define the initial configuration of the membership vector $\mathbf{x}$ as

$$x_i = \begin{cases} b_i, & \text{if } b_i > 0, \\ i+2, & \text{otherwise} \end{cases} \tag{7}$$

From the re-definition of the membership vector, vertices seeded as either foreground or background remain unchanged throughout the entire process. Our goal is to divide the graph into two sets of vertices. However, the original MOM algorithm automatically defines the number of clusters. We force the algorithm to aggregate vertices until only two clusters remain, even when the gain in modularity is not positive. Once the process has terminated, the membership vector contains only two labels (1 and 2).

E. Morphological Sharpening

Super-pixels normally converge with high precision to boundaries of distinct objects in an image, e.g., bones, organ or metal implants. However, cellular boundaries are highly irregular, and super-pixels may not fit their boundaries precisely in some regions, as illustrated in Figure 3.a. To improve boundary handling, we apply a morphological sharpening operator in a post-processing step, as described by Schavemaker et al. (2000) [19]. This operator is defined by the following transformation:

$$\varepsilon[\Omega](\mathbf{x}, \rho) =$$

$$\begin{cases} \mathbf{F}^{\oplus}(\mathbf{x}, \rho), & \mathbf{F}^{\oplus}(\mathbf{x}, \rho) - \mathbf{F}(\mathbf{x}, 0) < \mathbf{F}(\mathbf{x}, 0) - \mathbf{F}^{\ominus}(\mathbf{x}, \rho), \\ \mathbf{F}^{\ominus}(\mathbf{x}, \rho), & \mathbf{F}^{\oplus}(\mathbf{x}, \rho) - \mathbf{F}(\mathbf{x}, 0) > \mathbf{F}(\mathbf{x}, 0) - \mathbf{F}^{\ominus}(\mathbf{x}, \rho), \\ \mathbf{F}(\mathbf{x}, 0), & \text{otherwise} \end{cases} \tag{8}$$

where Ω is the input image and $\mathbf{x}$ is the pixel position, ρ is a scale parameter of the morphological scale space, and $\mathbf{F}^{\oplus}(\mathbf{x}, \rho)$ and $\mathbf{F}^{\ominus}(\mathbf{x}, \rho)$ are operators for gray-scale dilation and erosion operations defined by:

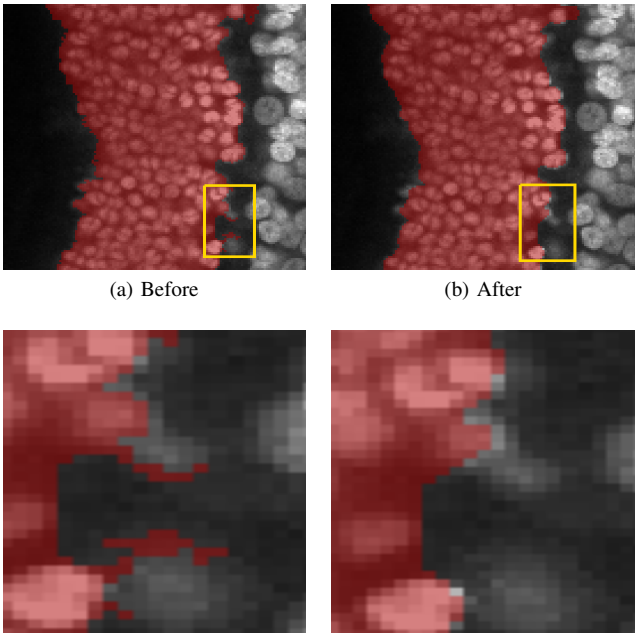
$$\mathbf{F}^{\oplus}(\mathbf{x}, \rho) = (f \oplus g^{\rho})(\mathbf{x}), \tag{9}$$

$$\mathbf{F}^{\ominus}(\mathbf{x}, \rho) = (f \ominus g^{\rho})(\mathbf{x}), \tag{10}$$

performed with a parabolic structuring function given as

$$g^{\rho}(\mathbf{x}) = -\frac{1}{2\rho}\mathbf{x}^{\mathbf{T}}\mathbf{x}. \tag{11}$$

The dilation and erosion operators replace the gray-scale value of a pixel by the maximum or minimum of the gray-scale values in its neighborhood. The function $\mathbf{F}(\mathbf{x}, 0)$ is equal to the input image Ω. Figure 3 shows an example of a result obtained with the sharpening operator.



(a) Before (b) After

(c) Before (d) After

Fig. 3. Sharpening operator. (a) Before application of operator and (b) after application of operator. A close-up view of the erosion result is shown in (c) and (d), yellow frames in (a) and (b) respectively.

III. Method

The diagram of Figure 4 illustrates the method proposed in this paper.

1) *Super-pixel generation:* we apply the super-pixels algorithm, described in Section II-A, to reduce the quantity of redundant information.
2) *Graph building:* a weighted graph, in which vertices are super-pixels instead of single pixels, is created. More details about our graph creation strategy are given in Section III-A.
3) *Seeding process:* a user selects the region of interest by placing seeds on it using brush strokes as showed in Figure 4.4.
4) *Graph clustering:* the segmentation of the regions of interest is performed by our seed-based algorithm introduced in this paper. It divides the graph into two clusters of vertices (foreground and background in the image). Seeds are used to guide the vertex agglomeration process so that vertices belonging to the cluster associated with foreground seeds correspond to the region of interest.
5) *Morphological sharpening:* inaccuracies in the super-pixel fitting process are corrected with the morphological sharpening operator described in Section II-E.
6) *User interaction:* steps 3 to 5 can be performed iteratively to improve segmentation.

A. Graph Building Strategy

Let $G = (V, E, W_E)$ define a weighted graph in which V is the set of vertices defining super-pixels Ω_I, E is the set of edges defining connected super-pixels, and $w_{IJ} \in W_E$ is the weight assigned to an edge (I, J). A vertex J is connected to the vertex I if J is inside a squared region of edge length r

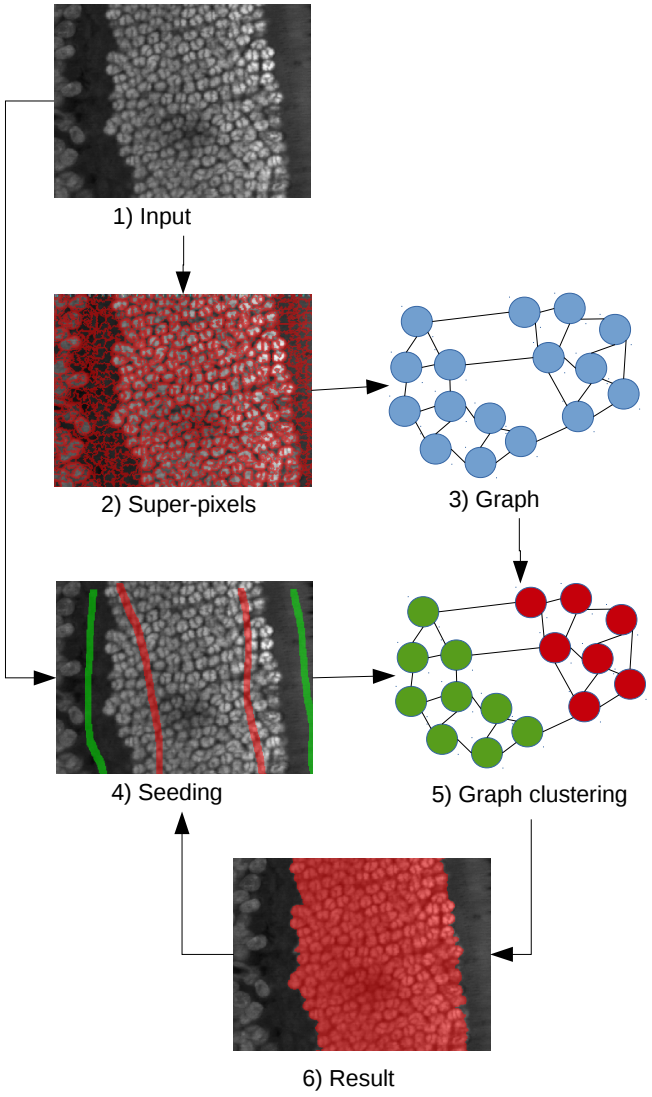


Fig. 4. An illustrative diagram of the proposed retinal layer segmentation method: first, similar and adjacent pixels are grouped into super-pixels; then a graph, whose vertices are super-pixels (steps 2 and 3), is created. The graph is then divided into two clusters using user-placed seeds to guide the clustering process (steps 4 and 5). These clusters give the corresponding segmentation of the input image. The morphological operator is finally applied (step 6). Users may iteratively improve the segmentation by placing and/or removing seeds.

(in pixels) and whose center is aligned with the Ω_I's spacial center. As shown in Figure 5, the set of super-pixels within this squared region is defined as the neighborhood V_I.

Edge weights are computed using an adapted version of the Laplace distribution function defined as:

$$w_{IJ} = \frac{1}{2b_I} \exp\left(-\frac{p(\| \Omega_I - \Omega_J \|^2 + \| \mathbf{x}_I - \mathbf{x}_J \|^2)}{b_I}\right), \tag{12}$$

where Ω_I is the mean color (intensity, for gray-scale images) of the I^{th} super-pixel, $\mathbf{x}_I$ is the spatial center of the super-pixel I, p is a parameter that penalizes small differences (intensity and position) among super-pixels. Parameter b_I can be estimated by the following maximum likelihood:

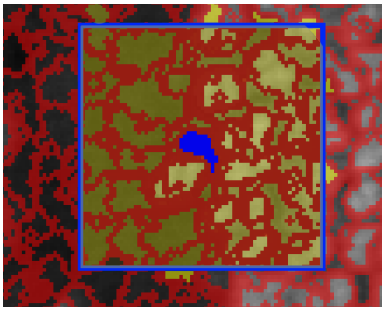


Fig. 5. Illustration of neighboring super-pixels. An edge between vertices I (super-pixel blue) and J (yellow) is created if their respective spacial centers are within a squared region of side length r.

$$b_I = \frac{1}{n} \sum_{J \in V_I} \| \Omega_J - \Omega_I \|^2, \tag{13}$$

where V_I is the super-pixel neighborhood of I and n is the size of V_I. Recall that the modularity measure (Equation 2) evaluate the quality of a certain division of a graph by looking at the edge-weights. Ideally, "different", but connected vertices, should have as low as possible edge-weights so that modularity value is not significantly incremented. On the other hand, similar vertices should have higher edge-weights in order to increase modularity. With the Laplace-based function, this behavior is attained as the function decays exponentially to zero when a variable is slightly far off the *Location parameter* [20] (super-pixel I in our application).

IV. Results and Validation

Our experiments were performed with two retinal image datasets with their respective best-expected segmentation, the so-called *ground truth* (GT), manually created by experts [12]. The datasets are collections of confocal microscopy images of an animal retina, taken at cellular resolution. For simplicity, we refer to the datasets as A and B. Dataset A is composed of 50 gray-level images, with their respective GTs, of size 512×512 and 768×512, with 0.3528 micrometer x 0.3528 micrometer pixel size. Dataset B is composed of 92 RGB color images of size 990×660 and 1226×817, with 0.3572 micrometer x 0.3704 micrometer pixel size. The datasets have differently sized images, and there are no duplicates.

To quantitatively evaluate the accuracy of our method we compare our segmentations with their respective GT data. We measure the similarity between them using two metrics commonly used in medical image segmentation, the Jaccard index and the Dice coefficient, defined as follows:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{14}$$

$$D(A,B) = \frac{2|A \cap B|}{|A| + |B|}, \tag{15}$$

where A and B are images, and $|A|$ and $|B|$ are their respective numbers of pixels. The Jaccard index and Dice coefficient are in the interval $[0,1]$. If A and B are empty, then $\{J|D\} = 1$.

In addition, we use the Hausdorff distance [21] to compute the difference before and after applying the morphological sharpening operator. The Hausdorff distance is defined as

$$H(A,B) = \max(h(A,B), h(B,A)), \tag{16}$$

where A and B are images and h is defined as

$$h(A,B) = \max_{a \in A} \min_{b \in B} \parallel a - b \parallel,$$

with a and b being pixels in A and B, respectively [22].

We have performed our experiments on a Linux workstation (Intel Core i7-2600 CPU 3.40GHz x 4 with 16GB memory). The C++ source code included the National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) [22] and the C++ template library of the linear algebra package Eigen [23]. We have used a painting tool, available in ITK-snap, to place the seeds.

A. Parameter optimization

Establishing parameter values is a time-consuming task. Our method uses several parameters during the segmentation process. Most of them were studied, and optimal range values for them are known. We use the parameter values for super-pixels and the sharpening operator as discussed in [2], [6], [24]. Table I shows the parameters employed.

TABLE I
PARAMETER VALUES FOR SUPER-PIXELS (SP) AND MORPHOLOGICAL SHARPENING (MS).

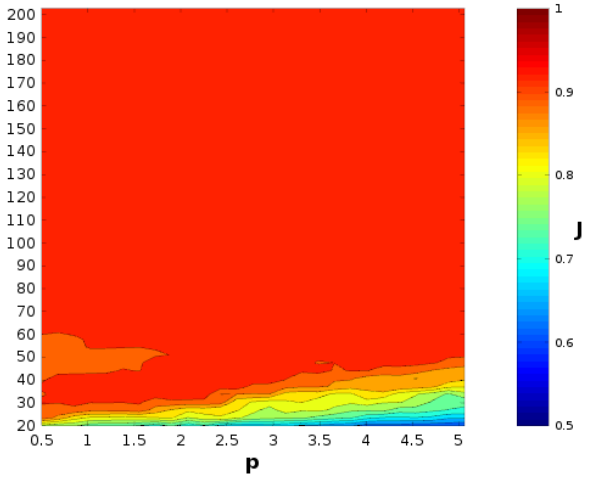
Method	Parameter	Value
SP	p_1	1
SP	p_2	0.5
SP	l	10
SP	it	5
MS	ρ	1
MS	it	1

To construct a graph from super-pixels involves the usage of two parameters: the squared neighborhood side length r and the penalty p, see Section III-A. To evaluate which parameter setup yields the best segmentation quality, an empirically defined range of values is considered:

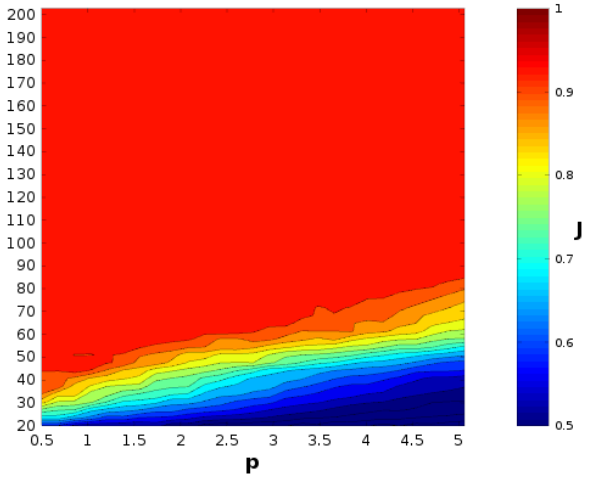
$$r = [20 - 200], \text{ step size } 1, \text{ range size } 181$$
$$p = [0.5 - 5.0], \text{ step size } 0.1, \text{ range size } 46.$$

We have evaluated datasets A and B for an extensive set of possible combinations of parameter values for r and p. The number of segmented images for datasets A and B are $249,780$ and $457,930$, respectively. For each dataset we randomly selected 60% for training. We used the Jaccard index to compute accuracy of results. Figure 6 shows a heat-map with the average accuracy result for the training sets.

In general, most parameter combinations resulted in high accuracy values ($J > 0.9$), implying that our approach is quite insensitive to changes in parameter values. Best segmentation accuracy ($J \simeq 1.0$) can also be obtained with a wide range of parameter values. For all other experiments we have chosen the parameter values listed in Table II:



(a) Dataset A



(b) Dataset B

Fig. 6. Heat-map for results obtained by using different values of parameters r and p for the training sets A and B. Variable r is the neighborhood's side length, p is the Laplace penalty, and J is the resulting average Jaccard index value. The majority of the segmentation results have Jaccard index values above 9.0, which indicates high accuracy. Jaccard index values closer to 1.0 (dark red regions) can be produced for a wide range of combinations of r and p values. In this example, Jaccard index values (lowest ones shown in dark blue) are not below 0.5.

TABLE II
PARAMETER SETUP FOR THE GRAPH BUILDING PROCESS

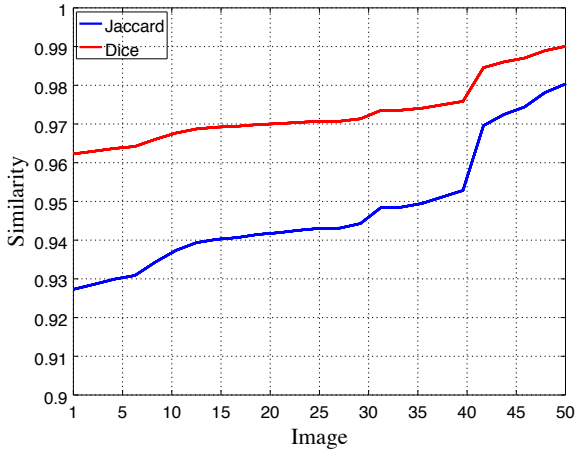
Parameter	Value
r	60
p	2

In addition, we have evaluated the entire datasets A and B using the parameter setup defined by Tables I and II. Figure 7 shows the similarity curves for the Jaccard index

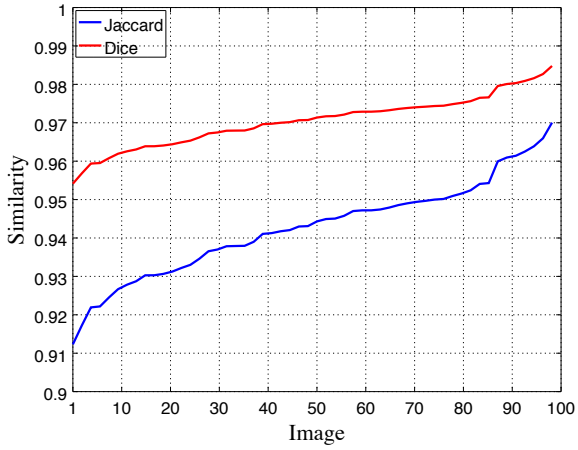
and Dice coefficient. In all cases, the segmentation accuracy is greater than 0.9. Table III provides the corresponding mean and standard deviation values.

TABLE III
MEAN μ AND STANDARD DEVIATION σ USING THE PARAMETER VALUE SETUP DEFINED IN TABLES I AND II.

Dataset	Jaccard		Dice	
	μ	σ	μ	σ
A	0.948	0.016	0.973	0.008
B	0.942	0.011	0.970	0.006



(a) Dataset A



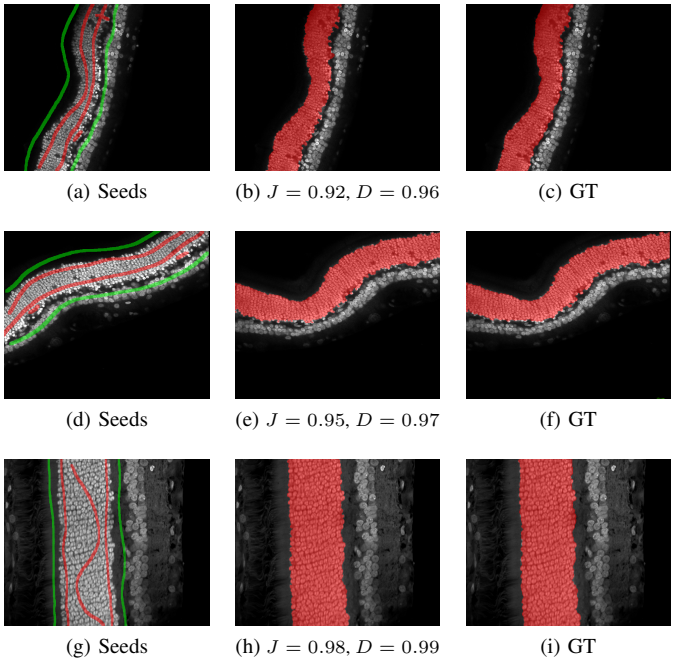
(b) Dataset B

Fig. 7. Similarity curves, Jaccard index (J) and Dice coefficient (D), obtained using the parameter values provided in Tables I and II. For best plotting, values are sorted by increasing order of similarity range, from left to right. All results are greater than 0.9. Axis x is the image label (an image's index).

Figures 8 and 9 show three representative segmentation examples considering the similarity curves shown in Figure 7. For both datasets we provide the results for the lowest, middle and highest similarity values.

B. Morphological Sharpening

We have evaluated the influence of the morphological sharpening (MS) operator on segmentation quality as well.

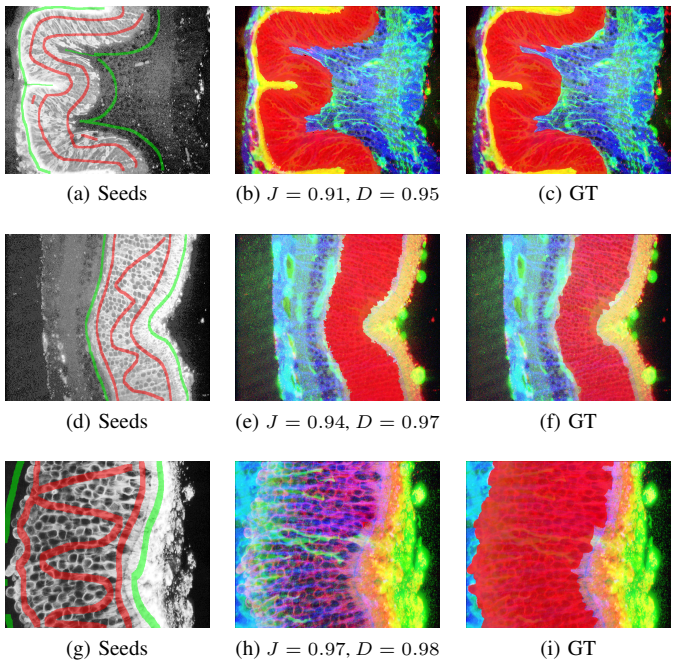


(a) Seeds (b) $J = 0.92$, $D = 0.96$ (c) GT

(d) Seeds (e) $J = 0.95$, $D = 0.97$ (f) GT

(g) Seeds (h) $J = 0.98$, $D = 0.99$ (i) GT

Fig. 8. Dataset A: Lowest, middle and highest similarity values (from top to bottom), obtained with the parameter values defined in Tables I and II.



(a) Seeds (b) $J = 0.91$, $D = 0.95$ (c) GT

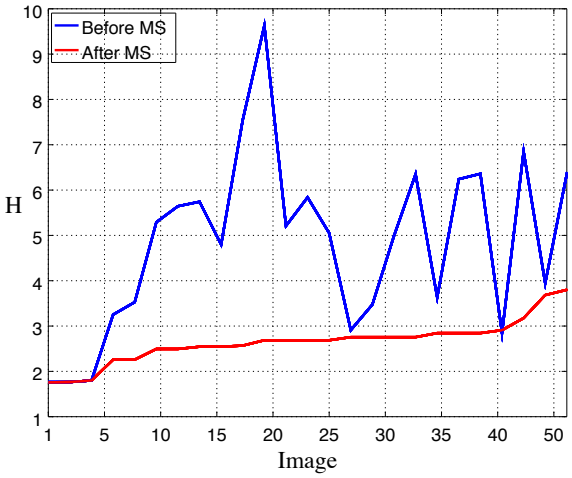
(d) Seeds (e) $J = 0.94$, $D = 0.97$ (f) GT

(g) Seeds (h) $J = 0.97$, $D = 0.98$ (i) GT

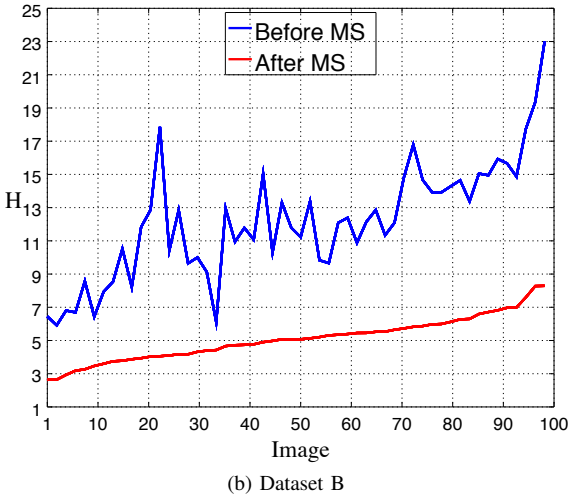
Fig. 9. Dataset B: Lowest, middle and highest similarity values (from top to bottom), obtained with the parameter values defined in Tables I and II. Segmented layer in semi-transparent red. This dataset consists of RGB color images.

Since the MS operator erodes only boundaries in a resulting segmentation, we have used the Hausdorff distance (H) to measure the difference between "before operator application" and "after operator application." Figure 10 shows the resulting Hausdorff curves for datasets A and B.

After applying the MS operator, imperfections in boundary regions are corrected, and segmentation accuracy is significantly improved, which is reflected in the Hausdorff curves.



(a) Dataset A



(b) Dataset B

Fig. 10. Morphological sharpening via erosion: Before-operator-application and after-operator-application comparison using Hausdorff distance.

C. Comparison with Related Methods

We have compared our method with two related methods also based on user-placed seeds and graph clustering. The first one, called *One-cut*, was proposed by Tang et al. (2013) [25] and extends the *min-cut* method introduced by Boykov et al. (2001) [5]. The second one, called *Laplacian coordinates*, was described by Casaca et al. (2014) [26] and uses an energy functional based on the Laplace matrix and a minimization approach using Cholesky decomposition.

We have selected these methods as they use a strategy very similar to ours. The literature essentially covers three basic methods to segment the retina layer, see Section I, and these methods were evaluated with at least one dataset that we also used to evaluate our method. We have not compared our method to these methods as our approach is based on a user-guided strategy. We believe that it would not be a fair or meaningful to compare our approach with entirely automatic methods and/or methods based on deep learning.

To perform a fair comparison, we optimized the parameter values required by both related methods instead of directly using the parameter values as suggested by the authors. We performed optimization with the training set and seeds of dataset A as described in Section IV-A. The One-cut method uses two parameters: *number of bins* and *color separation*; the Laplacian Coordinates method uses only the parameter *beta*. We evaluated the parameter value choices for the following intervals: One-cut parameter values were chosen from the ranges $bins = \{16-255\}$ and $color = \{0.1-1.0\}$; Laplacian coordinates parameter values were chosen from the range $beta = \{300-600\}$. These ranges are based on the original parameter values suggested by the authors. Best results were obtained for these parameter values: One-cut: $bins = 50$, $color = 0.5$; Laplacian Coordinates: $beta = 600$.

To compare our method with the other methods work, we segmented all images of dataset A with all methods, using the Jaccard index as similarity (quality) metric. For the entire dataset A we used the same seed set-up – as much as possible – and the parameter values resulting from the optimization. Figure 11 shows the resulting Jaccard index curves.

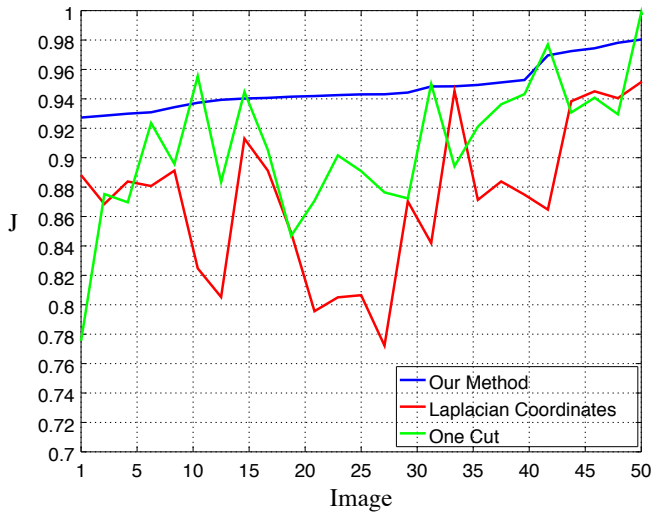


Fig. 11. Results for Laplacian coordinates, One-cut and our segmentation method. Variable J (y-axis) is the Jaccard index, and the numbers on the x-axis are the image IDs.

We could not perform a similar experiment with dataset B. This dataset consists of larger images ($\geqslant 990, \geqslant 660$) that cannot be handled by Laplacian coordinates or one-cut segmentation methods. In contrast, our method can segment very large images.

Our method outperforms the considered related methods not only because we use a morphological sharpening operator, but also because it is less dependent on the seeds. In general, we do not need to place as many seeds as other methods require, and our method does not require a user to place many seeds trying to comprise the entire object of interest, as other methods commonly do. This fact explains the relatively low segmentation accuracy of the two considered related methods. If one wanted the two considered related methods to achieve performance comparable to ours, one would have to use more seeds and a higher-precision placement of the seeds.

D. Scalability Analysis

The super-pixels algorithm (SUTP) has linear complexity $O(n \times it)$, where n is the number of pixels in the input image

and it is the number of iterations. The community detection algorithm (MOM) has, on average, near-linear complexity $\approx O(n)$ for sparse graphs, where n is the number of vertices [18]. In our graph-building strategy vertices are connected only in a local neighborhood, which invariably results in a sparse graph. Moreover, our method uses super-pixels instead of individual pixels, thereby drastically reducing graph cardinality. The morphological sharpening operator also has linear complexity $O(n)$, where n is number of input image pixels.

All algorithms used in our method are of linear complexity. Further, a significant resolution reduction is achieved via the use of super-pixels. As a consequence, our method can segment large images (1226×817) in a few seconds (two seconds, on average, using the hardware configuration described in the beginning of this section). Images of low resolution are segmented in less than a second.

Table IV provides the number of pixels, the number of super-pixels, and the average processing time for the two datasets A and B.

TABLE IV
AVERAGE PROCESSING TIME (IN SECONDS) FOR DATASETS A. NOTICE THE NUMBER OF SUPER-PIXELS CORRESPOND TO THE NUMBER OF VERTICES IN THE GRAPH.

Dataset	Input Pixels	Super-pixels	Time
A	262,144	2,621	1
B	1,001,642	10,016	2

The use of super-pixels and the linear complexity of our community detection algorithm allow our method to segment very large images efficiently, outperforming or overcoming limitations of the considered related methods. The first stage of our method (super-pixels generation) constitutes about 70% of total computing time, and it is performed only once. The second stage (graph clustering) accounts for the remaining processing time and is performed iteratively to fine-tune segmentation.

E. Usability test

Twenty different volunteers with no specific knowledge of the application domain were selected to perform usability tests. They were given a set of 15 random images and were told to draw strokes on the background and foreground. We did not provide any instructions regarding the number of strokes to be drawn. The mean time for all users was 6.025 seconds. Figure 12 shows the results of the mean time in seconds per user. The accuracy (Jaccard index) is shown in Figure 13. Considering that our volunteers had no knowledge about the application domain, we observed a slight reduction in accuracy, but quality was still preserved.

We have studied the relationship between user time spent and resulting Jaccard values, performing usability tests with 20 volunteers. The two users who spent about 13 seconds, obtained Jaccard indices greater than 0.9. The other 18 users spent between 2 to 9 seconds; 9 of these users obtained Jaccard indices greater than 0.9, and the others obtained

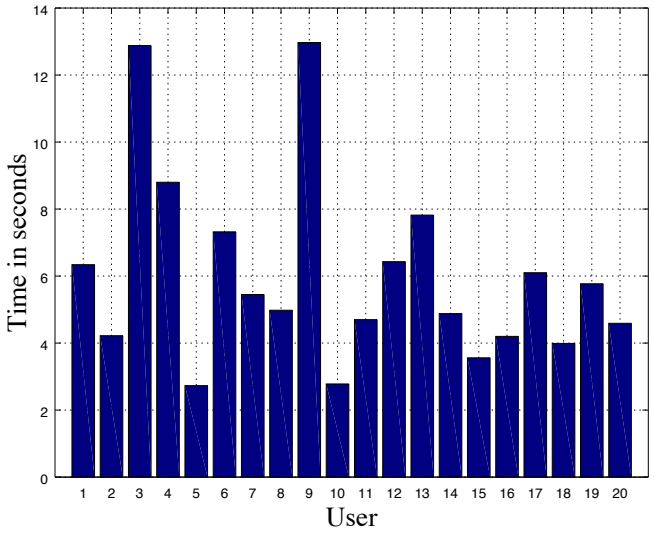


Fig. 12. Time in seconds for usability tests with 20 volunteers having no knowledge about the application domain. Volunteers were given 15 random images and were told to draw an arbitrary number of strokes on the background and foreground. Time refers to the mean time per user.

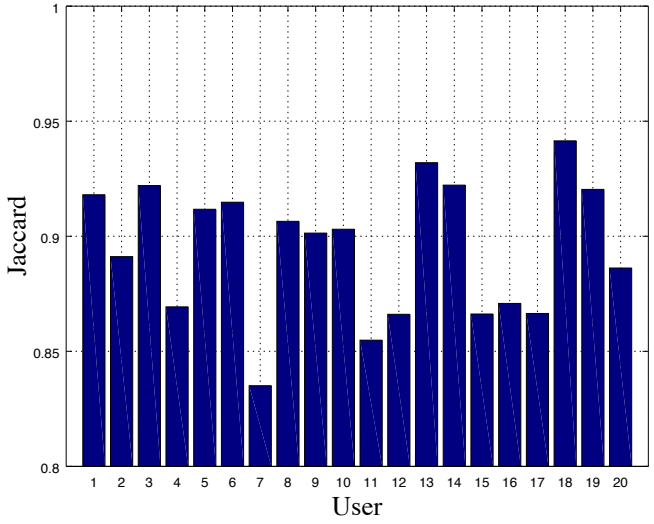


Fig. 13. Jaccard values for usability tests with 20 volunteers having no knowledge about the application domain. Volunteers were given 15 random images and were told to draw an arbitrary number of strokes on the background and foreground. Values refer to the mean value per user.

indices between 0.83 and 0.89. These results are good, despite the fact that these users worked with the tool for the first time.

V. CONCLUSION

We have introduced a new interactive and highly efficient method for retinal layer segmentation for color and gray-scale images. Our approach consists of three main steps:

1) Pre-segmentation performed with super-pixels.
2) Graph modeling and segmentation into two clusters using seeds.
3) Morphological erosion to correct possible imperfections in cell boundary regions.

Users can iteratively place and/or remove seeds in order to improve a segmentation.

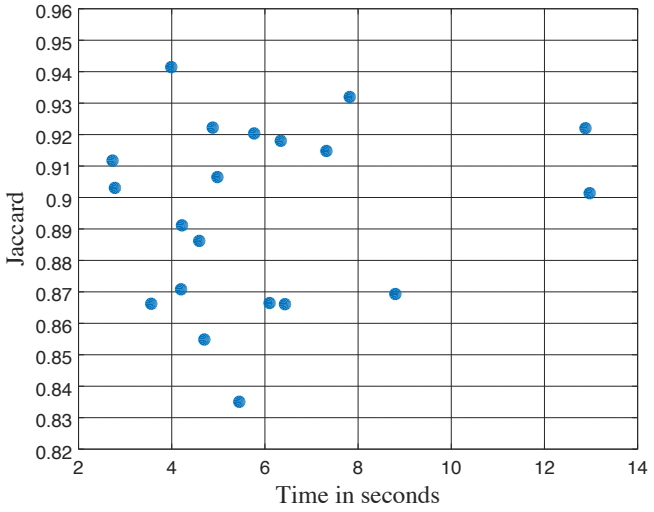


Fig. 14. Time[s] vs. Jaccard index values obtained by usability test with 20 volunteers having no knowledge about the application domain. Volunteers were given 15 random images and were asked to draw an arbitrary number of strokes on the background and foreground.

We have validate our approach with four experiments:

1) Evaluation of the parameter space.
2) Comparison with two related methods.
3) Evaluation of the influence of a morphological sharpening operator.
4) Usability test.

Our results show that our method produces a highly accurate segmentation of the retinal layer for color and gray-scale images. By using a morphological sharpening operator, complex circle-like contours (cell boundaries) are accurately segmented. A highly accurate segmentation can be produced with a wide range of parameter values, documenting the relative insensitivity of our method to parameter settings. In addition, our method requires about two seconds to process a large image, in addition to the seeding time of 6.025 seconds. This level of efficiency makes our method useful for retina segmentation tasks.

ACKNOWLEDGMENTS

This work was supported by the Brazilian research agency Sao Paulo Research Foundation, under FAPESP grant numbers 2012/24036-1 and 2018/06074-0.

REFERENCES

[1] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 315–337, 2000, pMID: 11701515.

[2] O. Cuadros Linares, G. M. Botelho, F. A. Rodrigues, and J. B. Neto, "Segmentation of large images based on super-pixels and community detection in graphs," *IET Image Processing*, pp. 60–80, 2017.

[3] M. Cabezas, A. Oliver, X. Lladó, J. Freixenet, and M. B. Cuadra, "A review of atlas-based segmentation for magnetic resonance brain images," *Computer Methods and Programs in Biomedicine*, vol. 104, no. 3, pp. e158–e177, 2011.

[4] N. R. Pal and S. K. Pal, "A review on image segmentation techniques," *Pattern Recognition*, vol. 26, no. 9, pp. 1277–1294, 1993.

[5] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 359–374, 2001.

[6] O. Cuadros Linares, G. Botelho, F. Rodrigues, and J. B. Neto, "Segmentation of large images with complex networks," *IEEE SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 24–31, 2012.

[7] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[8] N. Vu, P. Ghosh, and B. Manjunath, "Retina layer segmentation and spatial alignment of antibody expression levels," in *IEEE International Conference on Image Processing*, vol. 2. IEEE, 2007, pp. II–421.

[9] S. Lobregt and M. A. Viergever, "A discrete dynamic contour model," *IEEE Transactions on Medical Imaging*, vol. 14, no. 1, pp. 12–24, 1995.

[10] L. Bertelli, J. Byun, and B. Manjunath, "A variational approach to exploit prior information in object-background segregation: Application to retinal images," in *IEEE International Conference on Image Processing*, vol. 6. IEEE, 2007, pp. 61–64.

[11] L. Bertelli, P. Ghosh, B. S. Manjunath, and F. Gibou, "Reference-based probabilistic segmentation as non-rigid registration using thin plate splines," in *2008 15th IEEE International Conference on Image Processing*, Oct 2008, pp. 3052–3055.

[12] E. D. Gelasca, J. Byun, B. Obara, and B. Manjunath, "Evaluation and benchmark for biological image segmentation," in *IEEE International Conference on Image Processing*. IEEE, 2008, pp. 1816–1819.

[13] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels," cole polytechnique fdrale de Lausanne, Tech. Rep., 2010.

[14] Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2274–2282, 2012.

[15] C. Çığla and A. A. Alatan, "Efficient graph-based image segmentation via speeded-up turbo pixels," *IEEE International Conference on Image Processing*, pp. 3013–3016, 2010.

[16] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113, 2004.

[17] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, p. 066111, 2004.

[18] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, no. 10, p. P10008, 2008.

[19] J. G. Schavemaker, M. J. Reinders, J. J. Gerbrands, and E. Backer, "Image sharpening by morphological filtering," *Pattern Recognition*, vol. 33, no. 6, pp. 997–1012, 2000.

[20] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance*. Springer Science & Business Media, 2012.

[21] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.

[22] H. J. Johnson, M. McCormick, L. Ibáñez, and T. I. S. Consortium, *The ITK Software Guide*, 3rd ed., Kitware, Inc., 2013, http://www.itk.org/ItkSoftwareGuide.pdf, (accessed 08.16.17).

[23] G. Guennebaud, B. Jacob, P. Avery, A. Bachrach, and S. Barthelemy, "Eigen v3," http://eigen.tuxfamily.org, (accessed 08.08.17), 2010.

[24] R. Beare, "Morphology with parabolic structuring elements," *Insight Journal*, vol. 228, 2008.

[25] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, "Grabcut in one cut," in *International Conference on Computer Vision*, December 2013.

[26] W. Casaca, L. G. Nonato, and G. Taubin, "Laplacian coordinates for seeded image segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 384–391, 2014.