

VORONOI-BASED SKELETONIZATION ALGORITHM FOR SEGMENTING THE NETWORK OF BIOLOGICAL NEURONS

Kotsur D. V. – M. Sc., Post-graduate student, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Ukraine.

Tereshchenko V. M. – Dr. Sc., Professor, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, Ukraine.

ABSTRACT

Context. The problem of automated processing and analysis of microscopy image data is of high relevance due to its extreme impact on the research and recent developments in the field of biology and medicine. Efficient image processing algorithms facilitate the development of new medical diagnostic tools and therapy processes. They help us to broaden our knowledge of underlying mechanisms and processes inside living organisms. The primal focus of this paper is the processing of the microscopy images of the biological neural network. This aims to facilitate further studies of biological neural network that would lead to the development of better methods for diagnosis, prevention and cure of the related deceases.

Objective. The goal of the work is to development of an efficient image processing algorithm for segmenting the network of biological neurons based on the fluorescent microscopy image data.

Method. The introduced algorithm for segmenting the network of biological neurons comprises several steps. Firstly, we apply image processing routines, which aim to enhance the quality of the image data and extract the contours of the biological neural network. Then we construct the skeleton of the network applying the Voronoi diagram for line segments extracted from the object's contours. We employ Voronoi skeleton to identify the cellular somas and differentiate them from axons and dendrites.

Results. The developed Voronoi-based algorithm allows us to segment individual neurons, localize their somas, axons and dendrites and extract graph representation of the neural network. The underlying Voronoi diagram data structure allows us to compute such graph efficiently in $O(N \log N)$ operations (where N is a number of contour points). The proposed segmentation method was implemented in the C++/Python programming language and evaluated on the fluorescent images from CellImageLibrary (CIL).

Conclusions. The proposed segmentation method aims to facilitate studies of biological neural networks. It computes segmentation of the network of biological neurons in $O(N \log N)$ operations using the Voronoi diagram data structure. This data structure, in turn, gives us an attributed graph representation of the segmented network. Therefore, classical graph processing algorithms can be applied to analyze the neural and compute such network's characteristics as the number of connections between individual neurons, the shortest signal transduction path between two neurons, etc.

KEYWORDS: microscopy, image processing, Voronoi diagram, Voronoi polygon, neurons, segmentation, graph.

ABBREVIATIONS

VD is a Voronoi diagram;
VP is a Voronoi polygon;
BFS is a breadth-first search algorithm;
DFS is a depth-first search algorithm;
DCEL is a doubly-connected edge list data structure;
CIL is the Cell Image Library resource;

NOMENCLATURE

q is a point in R^2 ;
 R^2 is a two-dimensional space;
 $\|\cdot\|$ is Euclidean norm in R^2 ;
 $C(t)$ is a parametric curve in R^2 ;
 $C_i(t)$ denotes i^{th} parametric curve;
 M is a number of curves in a set;
 $I_{1,M}$ is an index set from one to M ;
 S is a set of M parametric curves in R^2 ;
 $VR(C_i)$ is a VR associated with a curve C_i ;
 $VP(C_i)$ is a VP associated with a curve C_i ;
 $VD(S)$ is a VD of a set of parametric curves S ;
 $\text{deg}(v)$ is a degree of a graph vertex v ;

m_i is a number of points sampled on $C_i(t)$;
 $l(t)$ is an arc-length of a curve $C(t)$;
 $\kappa(t)$ is a curvature of a curve $C(t)$;
 P_{C_i} is a set of discrete points on sampled a curve C_i ;
 γ is a "significance" coefficient in a parametrization;
 T_m is a threshold for bifurcation points merging.

INTRODUCTION

The nervous system comprises a large number of neurons forming a branched connected network, which is responsible for signal transduction and cognitive abilities of the living organisms [1]. The comprehensive understanding of the structures, functions, processes and mechanisms related to the biological neural networks requires analysis tools, which would enable a fast and precise processing of large quantities of available measurements. This, in turn, would facilitate the development of effective diagnosis, treatment and prevention methods against the deceases related to the neural system.

In comparison to other measurement techniques, fluorescent microscopy imaging allows us to capture the structural information of the neural network and visually investigate such characteristics of the network as topology, the density of the neurons, formation of synapses,

etc. However, a manual extensive analysis of the large neural network based on the obtained microscopy data could be extremely time-consuming. Therefore, one requires fast automated image processing tools, which would extract the topology and structural properties of the neural network from the fluorescent image data. The most suitable data structure for representing the topology and the structure of the network is an attributed graph. Such representation allows us to use numerous graph mining algorithms for the neural network analysis (e.g., Dijkstra algorithm for finding the shortest path or Ford-Fulkerson [2] algorithm for investigating the number of direct connections between two neurons).

Therefore, suitable image processing algorithm should segment fast and precisely the network of biological neurons and extract the graph-based representation of the network capturing its structural properties.

The Voronoi skeleton data structure meets these requirements, since it's based on the Voronoi diagram for line segments [3], it allows us to extract a graph representation of the network in terms of DCEL and it can be efficiently computed in $O(N \log N)$ operations, where N is a number of input generators. Moreover, Voronoi skeleton allows us to obtain a thin-line representation and store such structural information as local radii of the neuron bodies and radii of the individual protrusions (axons and dendrites). This could be employed to classify cellular bodies from their protrusions and to compute various network properties (such as the width of the axons, etc.).

The object of study is the segmentation of the network of biological neurons based on the fluorescent microscopy image data.

The subject of study is efficient Voronoi-based methods for segmenting the network of biological neural network based on the fluorescent microscopy images, which allow extracting the topology of the network suitable for further analysis using graph theory.

The known methods are based on morphological thinning [4–7] or tracing techniques [8–12]. The first class of methods is known to give errors on the step of vectorization and graph constructions. The latter class of methods requires a lot of computational resources.

The purpose of the work is to establish an efficient Voronoi-based algorithm for segmenting the network of biological neurons based on the fluorescent microscopy image data.

1 PROBLEM STATEMENT

Given a fluorescent microscopy images depicting the biological neural network, segment the individual somas, axons and dendrites, construct the graph representing the connectivity of the network, which can be used to apply graph processing algorithms and establishing neural network's structural properties.

The segmentation problem reduces to the following tasks: (1) compute the binary image of neural network; (2) construct thin skeleton based on the binary image,

extract the topology of biological neural network; (3) identifying cellular bodies and classify the protrusions.

In this paper we focus on Voronoi-based skeletonization algorithm as it allows us to obtain directly the graph representation of the neural network and avoid the step of vectorization, which is necessary for tracing- and thinning-based algorithms.

2 REVIEW OF THE LITERATURE

Typically, the workflow for transforming fluorescent image into a binary mask consists of the following steps [13]: image denoising, image enhancement and binarization. Then the skeleton of the binary mask is constructed (either by using contours of the mask or the region of the mask itself).

Skeletonization methods. There are three main classes of the skeletonization methods. The first class consists of the algorithms, which use morphological thinning technique [4–7]. This technique iteratively constructs the skeleton in the same resolution as original binary mask. Therefore, vectorization of the skeleton and extraction of the graph is required. This might induce additional computational errors due to irregularities in the skeleton.

The algorithms of the second class [8–12] employ pre-computed distance (potential) map and trace the skeleton along the maximal ridges of the map. Some versions of such method still require vectorization.

The third class of the algorithms employs the approximate Voronoi diagram and requires extracting the contours of the binary object [14–16]. Then the skeleton of the planar object is obtained as an intersection of approximate Voronoi diagram with the object's domain. However, at the discretization step such approach should consider the spatial complexity of the object [17].

In this paper we focus on the latter class of the Voronoi-based algorithms, since they don't require the vectorization step and give us directly a graph representation of the object's skeleton.

Currently there are many efficient algorithms for computing Voronoi diagrams for simple objects (e.g., points, line segments). For example, Fortune's algorithm [18] and "divide and conquer" algorithm [19] allow us to construct a Voronoi diagram for the set of N points/line segments in $O(N \log N)$. However, usually in practical problems objects have more sophisticated curved shape. It in turn leads to investigation of the Voronoi diagram for complex-shaped objects.

One approach for constructing the Voronoi diagram for curvilinear polygons was introduced by Ramamurthy and Farouki in their papers [20, 21]. The authors proposed an incremental algorithm for constructing the exact Voronoi diagram and medial axis for planar domains with curved boundaries. Their method uses an approximate algorithm for computing curve-curve bisectors [22] and also Newton-Raphson iterative scheme for computing exact bifurcation points. It was shown that the complexity of such algorithm is $O(N^2)$, where N is a number of sites.

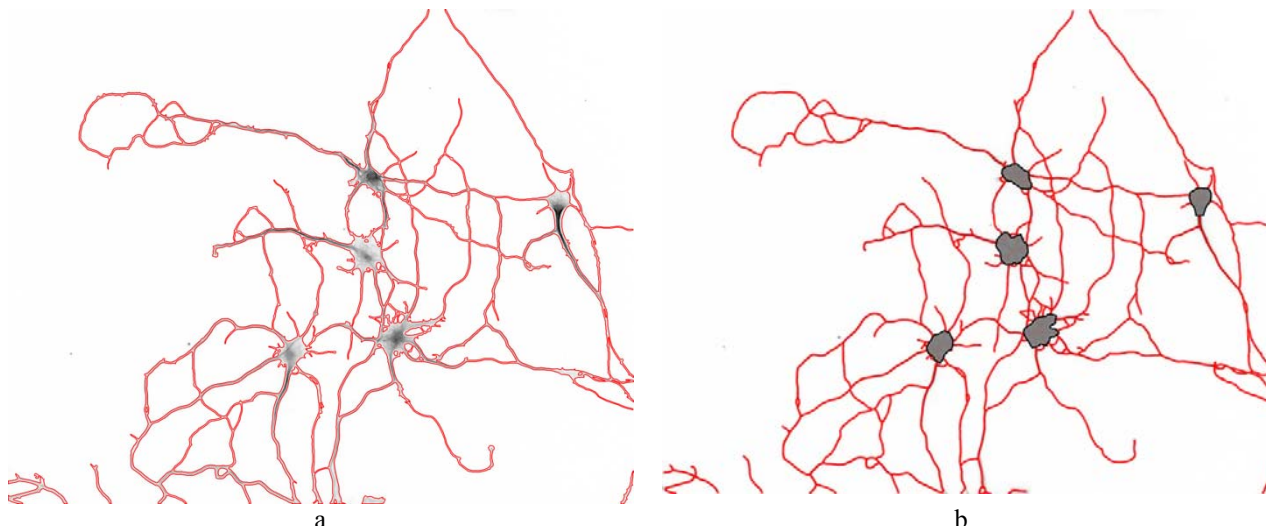


Figure 1 – The main steps of our segmentation algorithm: a – image processing and contours extraction; b – Voronoi-based skeleton construction and cell bodies segmentation (dark gray regions).

In the paper [23] authors proposed a randomized incremental method for constructing the Voronoi diagram for curved objects, which has complexity $O(N \log N)$, where N is the number of curves. However, authors assume that some operations (e.g., finding intersection point of two curves) can be performed in $O(1)$, although in practice these operations are time-consuming and sometimes require a sophisticated iterative implementation.

The difficulties in computing exact Voronoi diagrams led to study of approximate Voronoi diagrams. One class of approximate algorithms aims to discretize a spatial domain and build Voronoi diagram based on a finite pixel grid [24], [14] (Algorithm 4.9.2) or as in case of [25] on an exponential grid. Another class of approximate algorithms simplifies the object's representation (for example, curves can be approximated by a set of points or line segments). Then the Voronoi diagram for simple objects (points / line segments) is constructed and processed such that an approximate Voronoi diagram for curves is obtained [14] (Algorithm 4.9.1), (see also [26]).

3 MATERIALS AND METHODS

In this section we provide a brief description of our segmentation algorithm. Its main steps are the following:

1) the fluorescence image is processed as described in the section “Image processing routines”: image is enhanced and binarized, the contours of somas, axons and dendrites are extracted and interpolated using cubic splines (see Fig. 1 a);

2) voronoi-based skeleton for the network represented by a set of parametric curves (contours) S is constructed (see section “Skeleton constructing algorithm – Description”);

3) using skeleton graph from Step 2 we classify points corresponding to the somas/cell bodies (see Fig. 1 b), axons and dendrites (see section “Segmentation of biological neurons”); localize the regions corresponding to somas.

Fluorescent microscopy images can be exposed to the noise and/or photo bleaching, which makes it difficult to use these images in further computations (e.g., for contour tracing). In order to suppress the artifacts obtained during the image acquisition, we perform additional image enhancement steps.

The comprehensive workflow of image processing routines is described below:

1) [Optional] Original image (see Fig. 2 a) is denoised using TV-regularization [27];

2) the brightness of dim axons and dendrites in the image from Step 1 is enhanced using Frangi's filter [28] (as shown in Fig. 2 b);

3) the images from Steps 1 and 2 are binarized using Otsu's global thresholding [13]. Then the resulting binary masks are combined using morphological union operation (see Fig. 2 c);

4) based on the binary mask from Step 3, we trace the contours of the object's boundary. The contours are oriented such that interior part of the objects is always to the left;

5) finally, we interpolate each contour using parametric periodic cubic spline (see Fig. 2 d). Such parametric representation is used later for Voronoi-based skeleton construction;

Skeletonization algorithm. Let's define distance between a parametric curve $C(t)$, $t \in [0,1]$ and a point $q \in R^2$ as follows:

$$d(q, C) = \min_{t \in [0,1]} \|q - C(t)\|. \quad (1)$$

The value $t^* \in [0,1]$, which minimizes the Equation (1) is called footpoint of point q on the curve $C(t)$. Note that a curve can have more than one footpoint [20].

Let $S = \{C_i(t), t \in [0,1] | C_i(0) = C_i(1), i \in I_{1,M}\}$ be a set of continuous parametric closed curves in R^2 .

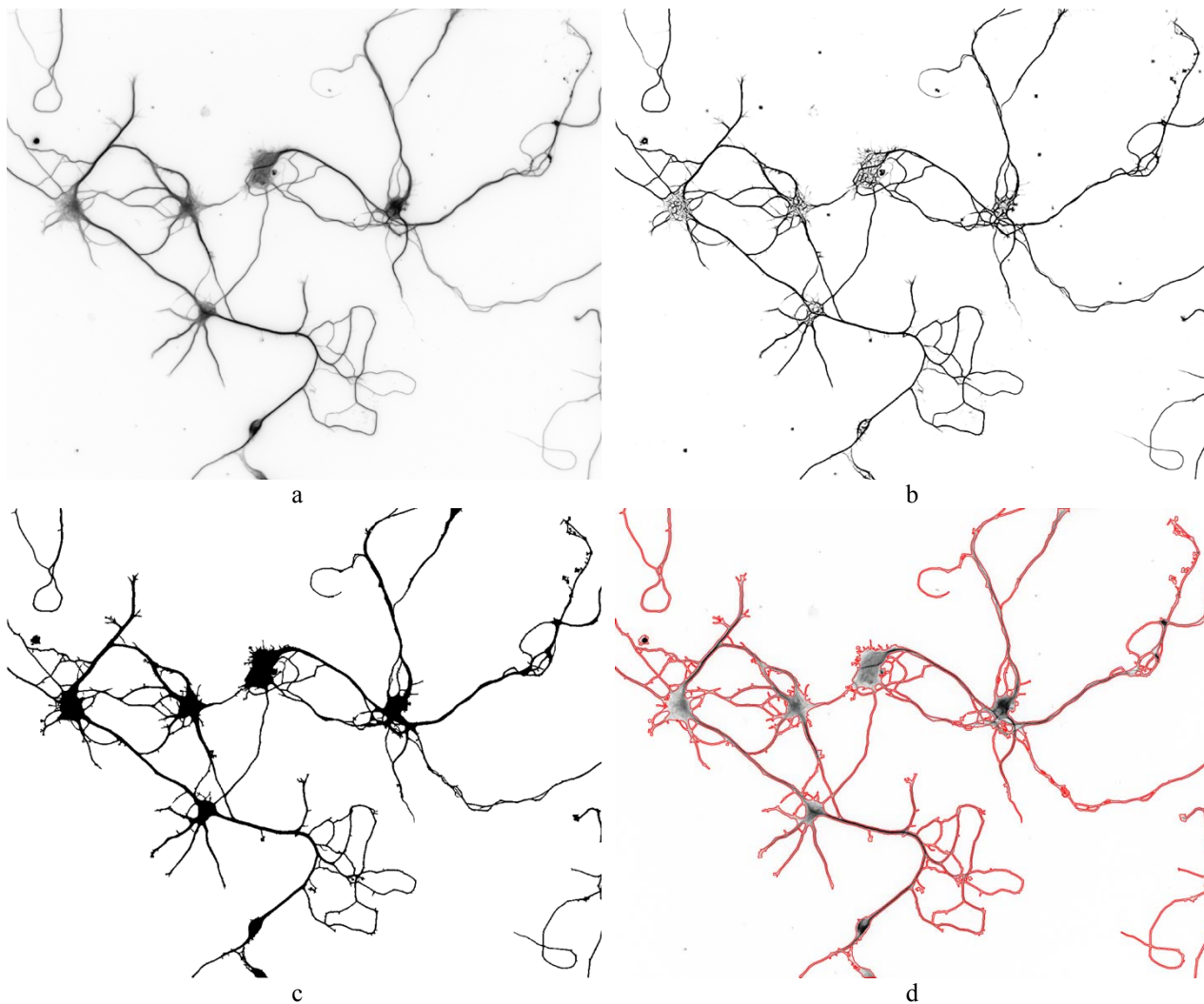


Figure 2 – The original image (a) is processed using Frangi's ridge enhancement filter (b). The result of this procedure is used to produce the binary mask (c). Then we use obtained mask to trace the contours of biological neural network (d).

Definition 1. The Voronoi region $VR(C_i)$ of a curve C_i is defined as follows:

$$VR(C_i) := \{q \in R^2 \mid d(q, C_i) \leq d(q, C_j), \forall j \in I_{1,M} \setminus \{i\}\}. \quad (2)$$

Definition 2. The Voronoi polygon $VP(C_i)$ associated with a curve C_i is defined as a boundary of $VR(C_i)$. According to [20] it is customary to call $VP(C_i)$ a “polygon”, although it has curved edges.

Definition 3. The Voronoi diagram $VD(S)$ of a set of parametric curves S is defined as follows:

$$VD(S) = \bigcup_{i \in I_{1,M}} VP(C_i). \quad (3)$$

Definition 4. The Bisector of two curves $C_i(s)$ and $C_j(t)$, where $s, t \in [0,1]$, is defined as a set of points that are equidistant from these two curves:

$$\{q \in R^2 \mid d(q, C_i(s)) = d(q, C_j(t)), i, j \in [0,1]\}. \quad (4)$$

Definition 5. The Self-bisector of a curve $C(t)$, where $t \in [0,1]$, is defined as a set of points, which have at least two footpoints on $C(t)$:

$$\{q \in R^2 \mid d(q, C(t)) = d(q, C(u_1)) = d(q, C(u_2))\}, \quad (5)$$

where $u_1 \neq u_2$ are two distinct footpoints on $C(t)$.

Thus, Voronoi polygon $VP(C_i)$ of curve C_i consists of a subset of (possibly trimmed) bisectors between and all other curves from S . It also contains self-bisectors of curves $C_i(t), i = 1, \dots, M$. According to [23] self-bisectors are of great importance, since they capture the connectivity of the object's domain.

Definition 6. The junction point connecting three or more bisectors (including self-bisectors) of the Voronoi diagram is called Voronoi vertex.

In this section we present the main steps of the Voronoi-based skeletonizing algorithm. Steps (1–3) of the algorithm are based on the Algorithm 4.9.1 from [14].

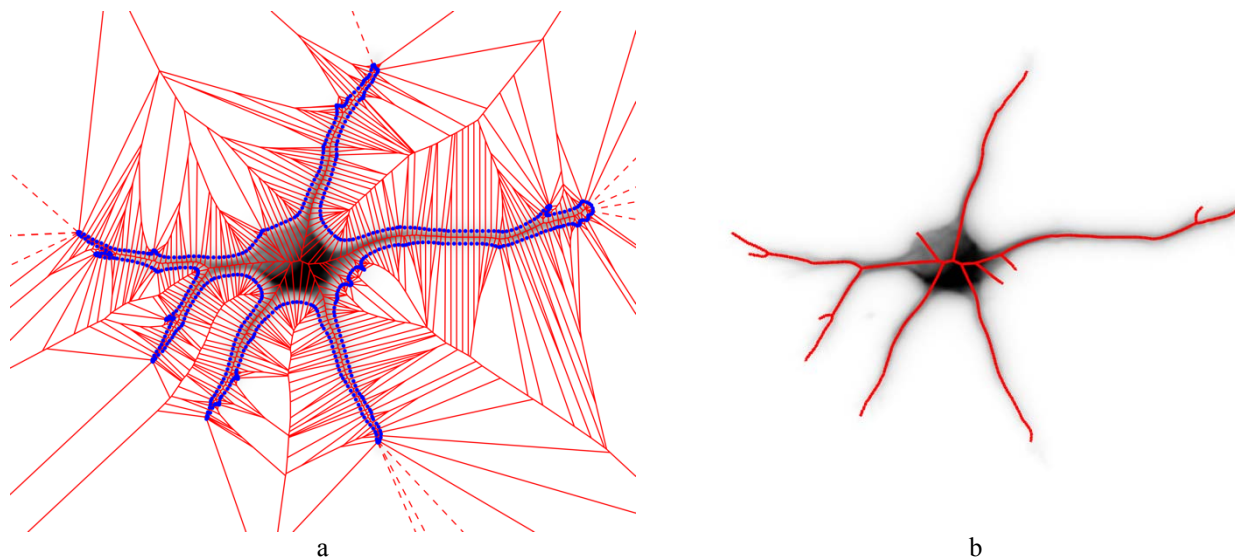


Figure 3 – The Voronoi diagram for sampled points (a) and corresponding Voronoi-based skeleton (b).

In comparison to Algorithm 4.9.1 from [14], we have modified Step (3) in order to preserve self-bisectors and handle possible topological errors. The main steps of skeleton construction algorithm are the following:

1. Discretize curve $C_i(t), i = 1, \dots, M, t \in [0, 1]$ and represent it as a set of points:

$$P_{C_i} = \{C_i(t_1), C_i(t_2), \dots, C_i(t_{m_i})\}, \quad (6)$$

where $0 = t_1 < t_2 < \dots < t_{m_i} = 1$ (Fig. 3a, b).

2. Construct Voronoi diagram for points $VP(P)$, where $P = P_{C_1} \cup P_{C_2} \cup \dots \cup P_{C_M}$, using Fortune's algorithm [18]. Example is shown in Fig. 3 a.

3. Process Voronoi diagram $VD(P)$ as follows:

3.1. Apply depth-first search algorithm to classify all vertices of Voronoi diagram $VD(P)$ as "inner" or "outer" w.r.t. the object interior. Firstly, we initialize DFS algorithm [2] from "outer" vertex (outside the convex hull of P , which can be obtained from $VD(P)$). If DFS is crossing the border of object change the label of the vertex to the opposite ("inner" to "outer" and vice versa), otherwise propagate current label. The cross-check can be performed fast if we store in a hash-table the pairs of Voronoi vertices and corresponding to them consecutive VPs.

3.2. Remove edges of $VD(P)$ for every pair of successive sampled points on a curve $C_i(t_j), C_i(t_{j+1})$, where $i \in I_{1,M}, j = 1, \dots, (m_i - 1)$. Remove all Voronoi vertices and respective edges classified as "outer". The resulting graph is denoted as $VD'(P)$;

3.3 Determine bifurcation points of the Voronoi-based skeleton $VD(S)$ using $VD'(P)$: traverse all vertices of VD using breadth-first search (BFS), mark v as a bifurcation point if $\deg(v) > 2$;

3.4. If a pair of neighboring bifurcation points is connected by polyline edge shorter than a give threshold value $T_m > 0$ then the corresponding polyline edge is removed and the pair is represented by one vertex (see Figure 3, B);

4. [Optional] Approximate edges of the processed Voronoi graph by Bezier splines using algorithm [29].

Remark 1. Step 3.2 can be performed faster if we organize data into two hash-tables. The first hash-table maps indices of sampled points to the compound structure comprising: index of a respective curve, indices of the next and previous sampled points of the corresponding curve (if applicable). The second hash-table maps indices of a curve to indices of sampled points of these curve (one point per curve is sufficient).

Remark 2. In the Step (3.2) we obtain Voronoi diagram $VD'(P)$. It can possibly contain vertices of degree 1, which are isolated vertices (e.g., centers of circles); vertices of degree greater than 2 (bifurcation points) and also vertices of degree 2. The latter constitute the approximate polyline edge of the VD for curves. At Step 4 we approximate these polyline edges by Bezier splines.

Discretizing parametric curves. At the Step (1) we sample points on curve $C_i, i \in I_{1,M}$ as stated by Equation (6). Note that, a sampling method has to consider the shape of the curve, otherwise topological inconsistencies may occur [17]. Therefore, uniform parametrization $|t_{j+1} - t_j| = \text{const}$, where $j = 1, \dots, (m_i - 1)$ as well as an arc-length parametrization (cf., Fig. 4 a) of a curve $\|C_i(t_{j+1}) - C_i(t_j)\| = \text{const}$, $j = 1, \dots, (m_i - 1)$ won't give satisfactory results [17].

In order to obtain a point set P_{C_i} representing the curve's shape, we used the method based on mixture of curvature and arc-length normalized parametrizations (cf., Figure 4, B) as described in [30] (Equation (8) in [24]).

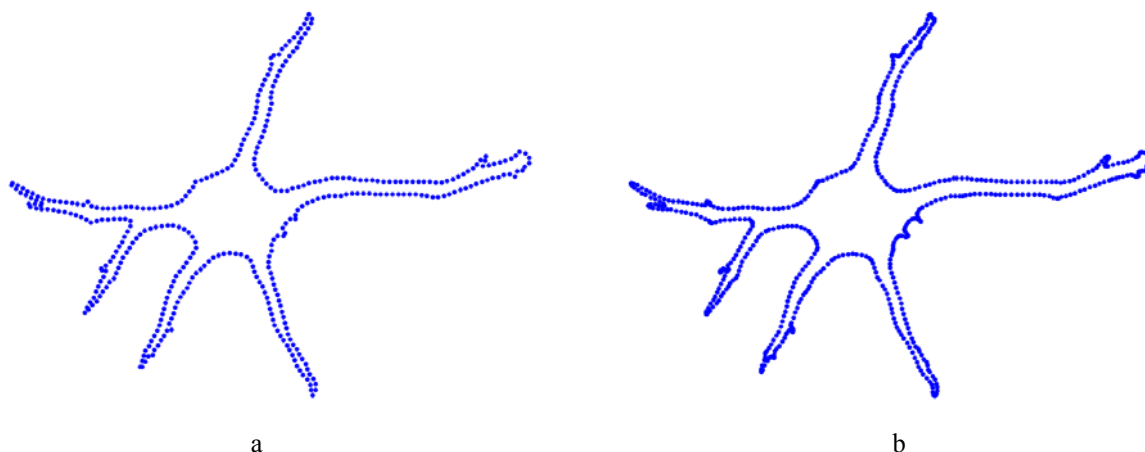


Figure 4 – Application of different parameterization methods for point sampling: a – is an arc-length parameterization (constant distance between two consecutive points); b – shows a mixed parameterization, which is a combination of curvature- and arc-length parameterizations. (The difference between (a) and (b) is well visible around the tips of dendrites)

In comparison to [30], where authors just averaged curvature and arc-length parameterizations, we use the following equation:

$$p(t) = l(t) + \gamma \kappa(t) \frac{l(t)}{\kappa(t)}, \quad (7)$$

where $\gamma > 0$ controls the “significance” of the curvature term (similar to Equation (7) in [24]). In case of $\gamma = 0$ we obtain just an arc-length parametrization.

Thus, if we parametrize a curve in such way, it will behave as an arc-length parametrized curve in “flat” regions with a small curvature ($\kappa(t) \approx 0$). However, in regions with a large curvature ($\kappa(t) \gg 0$), the density of the sampled points will increase (controlled by γ).

Let m_i be a number of points, which we have discretized on a curve $C_i(t)$, $N = m_1 + m_2 + \dots + m_M$ is a total number of points. In the following lemma(s) we analyze the complexity of each step of our algorithm.

Lemma 1. Let $C_i(t)$ be a curve represented by Bezier spline, then Step 1 of our algorithm can be computed in $O(N \log N)$ in the worst-case.

Proof. The worst-case scenario appears when we construct bisector between a point and a curve represented by the rest $N-1$ points. Assuming that the number of control points of $C_i(t)$ is proportional to N , one point of $C_i(t)$ can be computed in $O(\log N)$. Then for N points can be computed in $O(N \log N)$.

Lemma 2. Voronoi diagram for N points at the Step (2) of our algorithm can be computed in $O(N \log N)$.

Proof. At the Step 2 we just apply well-known sweep-line algorithm for Voronoi diagram construction, which is described in detail in [18]. Its complexity in case of N input points is $O(N \log N)$.

Lemma 3. Processing of the Voronoi diagram for points at the Step 3 can be performed in $O(N)$.

Proof. Let’s analyze the operations of Step (3) in detail:

Step 3.1: The complexity of BFS is $O(N)$. At every step of BFS we check if we cross the border of the object, which is $O(1)$ due to the fact that all pairs of border vertices are stored in a hash-table. Thus, total complexity of Step 3.1 is $O(N)$.

Step 3.2: We can look through all pairs of consecutive sampled points for every curve in $O(N)$ operations using the hash-tables as described above. One edge can be removed in $O(1)$ operations just by reassigning pointers in doubly-connected edge list (DCEL). Thus, total complexity of Step 3.1 is $O(N)$.

Step 3.3: The complexity of BFS for a graph with N vertices and E edges is $O(N+E)$. In case of Voronoi diagram E proportional to N . Thus, complexity of Step 3.2 is $O(N)$.

Step 3.4: The length of Voronoi edges between two bifurcation points can be already computed in Step (3.1). Checking the condition that the length of an edge is less than T_m for all edges can be done in $O(N)$. Two Voronoi vertices can be merged into one in $O(1)$ by reassigning pointers of DCEL. Thus, the total complexity of Step 3.3 is $O(N)$.

Since the complexities of Steps 3.1, 3.2, 3.3, 3.4 are equal to $O(N)$, the total complexity of Step 3 is $O(N)$.

Theorem 1. The complexity the proposed algorithm for constructing an approximate Voronoi diagram for a set of M curves and N sampled points is $O(N \log N)$.

Proof. Follows directly from Lemmas 1–3.

Remark. In the current analysis we didn’t take into account the optional Step (4), where we approximate the edges of the resulting skeleton by splines. The fitting of Bezier spline at Step (4) is performed using the recursive “divide and conquer” algorithm, which is described in [29]. In the worst-case scenario this procedure works in $O(N^2)$. Therefore, the average time complexity is $O(N \log N)$.

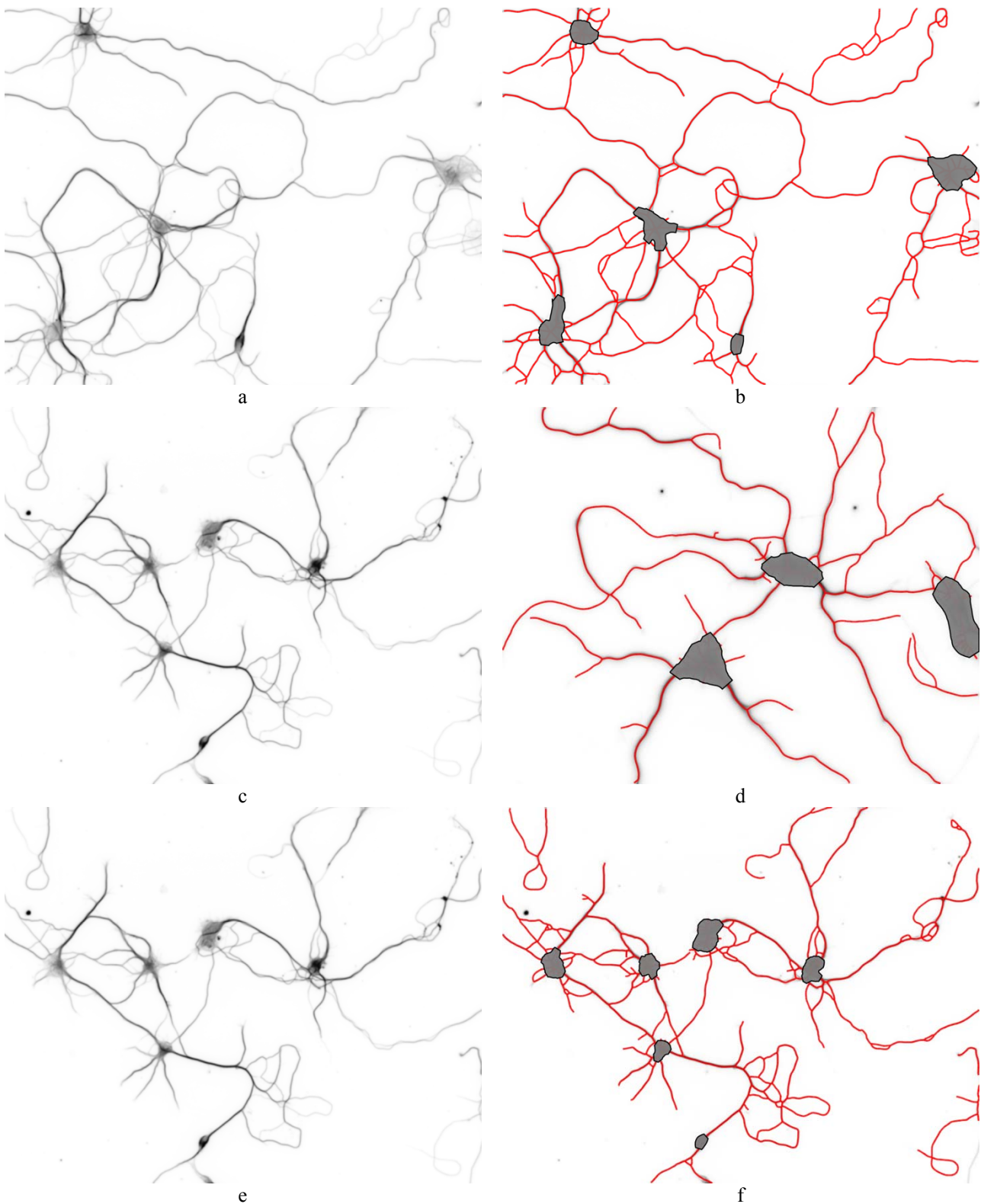


Figure 6 – Examples of input fluorescent microscopic images (left column – a, c, e); Segmented axons, dendrites (red lines, right column – b, d, e) and cell bodies (gray regions, right column – b, d, e)

Segmentation of biological neurons. From the Step 2 of the algorithm described above we employ skeleton to identify cellular bodies (somas). Since every Voronoi vertex is associated with corresponding circle, which

encloses (at least) three generator points of Voronoi diagram, we can differentiate the cellular body vertex from axon/dendrite vertex by comparing the radius of the circle with some threshold.

If we consider all radii of all Voronoi vertices of the skeleton, we can apply (similar as for image binarization) Otsu's thresholding [13]. This would allow us to find an optimal value of the radius of the circle corresponding to the cellular body. Then using such procedure, we can classify all Voronoi vertices as "body" vertex or "non-body" vertex. Using BFS algorithm all connected "body" vertices (with no gaps between them) can be grouped into the connected components (clusters). At the last step we build a star-shape polygon [31] for each component. This polygon would represent the body of the neuron.

4 EXPERIMENTS

The proposed segmentation algorithm was implemented in C++ and Python programming languages. The image processing workflow as described in the corresponding section was implemented using Python programming language. We used "skimage" library to investigate and set up the pipeline of image filters from denoising to contours extraction. We have implemented the Voronoi skeletonization algorithm (including the Voronoi diagram for line segments and Voronoi graph post-processing routines) in C++ programming language and used STL library for containers. Finally, C++ code was integrated with Python code using intermediate Cython programming language. C++ code was compiled as a separate library with Python interface. In order to visualize the results of the segmentation algorithm, we used Python's "matplotlib" graphics library.

The performance of the implemented method in terms of execution time (s) was tested on Linux platform, which has Intel Core i7 2.2GHz and 8Gb RAM (for details, please, see Figure 5).

In order to test and visually evaluate the performance of the segmentation algorithm we processed 20 images depicting the network of biological neurons as well as individual neurons. Typical resolution of the image is 1300x1300 pixels. The examples of input fluorescent microscopic images are available in free access at the web resource of the CellImageLibrary [32]. We used images of Dieter Brandner and Ginger Withers: *Rattus*, multipolar neuron (2010). The CIL IDs are the following: 8785, 10109, 10112, 10113, 10118, 10203, 10218, 10219, 10223, 10224, 10225, 10226, 10227, 10228, 10229, 10234, 36182, 36183, 36184, 36185.

For the contour discretization we used the "significance" parameter $\gamma = 0.5$. The threshold for merging the neighboring bifurcation points was set to $T_m = 1.0$.

5 RESULTS

We have tested the execution time of our segmentation algorithm and averaged the results for all CIL images mentioned above. The average time required to process one single image using the filter pipeline as described in the section "Image processing routines" is ~ 1.32 s. In order to evaluate the execution time of the Voronoi skeletonization algorithm we have discretized the contours by varying the number of sampled points from 700 to 5500. For each fixed number of sampled points (input sites) we

measured the execution time and averaged over all images. The results of the evaluation are shown in Fig. 5. Typically, the average object contains from 800 to 1500 points after sampling using the "mixed" parametrization. Therefore, the average execution of the whole segmentation routine is approximately ~ 2.7 s.

Due to the absence of the ground truth labels for segmented neurons we performed a manual visual investigation and evaluation of the segmentation results. In particular we investigated the error related to the neuron bodies' identification. Firstly, we have manually counted the number of false positives (FP) – the number of protrusive regions (axons or dendrites) of wrongly detected as cell bodies, false negative (FN) – the number of cell bodies, which were not identified. After computing the precision-recall metrics we obtained the following results: Precision (P) = 95.33% and Recall (R) = 93.32%.

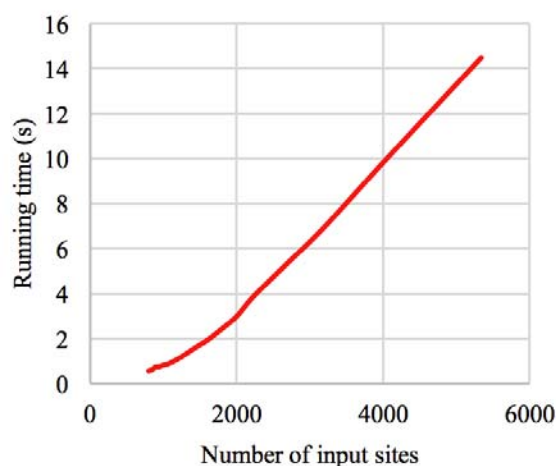


Figure 5 – Average running time of the method depending on the number of contours (20 images).

The examples of input fluorescent microscopic images (left column) and segmentation results (right column) obtained with our method are shown in Figure 6. In order to illustrate the results in this paper we used a subset of the originally collected CIL dataset. The subset consists of the CIL images with the following IDs: CIL: 8785, 10219, 10113, 10109.

6 DISCUSSION

The developed segmentation algorithm allows us to obtain directly the graph representation of the neural network. In contrast, thinning-based algorithms [4–7] perform the construction of the binary skeleton of the shape. In this case the resulting skeleton has the same resolution as an input image. Therefore, one needs to apply vectorization techniques and extract the graph representation from the binary skeleton. This in turn may induce additional errors due to vectorization. Other class of algorithms – tracing methods [8–12] – performs the computation of the skeleton iteratively by tracing each individual branch of the skeleton. Such methods may recover the topology of the underlying graph. However, this requires

additional checking procedures (e.g., collision tests), which might be extremely time-consuming in case of large image data.

We have estimated the average execution time of thinning methods [4–7] with vectorization routine and tracing methods [8–12]. First class of methods performs in average ~ 2.5 s. The average execution time of the tracing-based methods [8–12] varies significantly from 5s to 60s per image. Therefore, the proposed Voronoi skeleton-based segmentation algorithm allows us to segment both – precisely and fast.

An important aspect of the proposed skeletonization algorithm is the reparametrization of the periodic spline curves representing the contours of the neural network. In this paper we proposed a mixed parametrization, which is a combination of an arc-length parametrization and curvature-based parametrization. The first one (arc-length parametrization) allows us to sample points equidistantly along the spline curve. However, such approach tends to decrease the details of the shape (e.g., local bumps) and, thus, may lead to inconsistencies in the resulting skeleton. The second (curvature-based) parametrization approach tends to reduce the number of sampled points in the flat regions of the curve. However, it might also provide an excessive amount of point in the regions with high curvature, which may correspond to a random noise in the contour data. Thus, curvature-based method would make the skeletonization algorithm less robust and more sensitive to random noise (or errors of contour tracing algorithm). Therefore, we the proposed mixed parametrization is a compromise approach, which allows to control the influence of the random noise or errors based on the introduced curvature “significance” parameter $\gamma > 0$.

As it was also mentioned above, Voronoi diagram allows us to represent neural network as a graph with sub-pixel precision. This in turn allows us to apply such algorithms as Ford-Fulkerson [2] for finding the number of connections between two cells (see Fig. 7) or Dijkstra algorithm [2] to find the shortest signal transduction path.

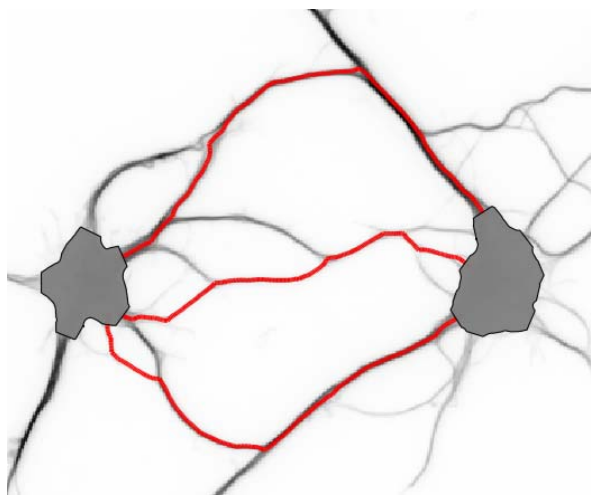


Figure 7 – Ford-Fulkerson algorithm was applied to find the connections between two neurons (gray regions)

CONCLUSIONS

The problem of automatic segmentation of the biological neural network based on the fluorescent microscopy images is solved.

The scientific novelty of obtained results is that we firstly propose the Voronoi-based method for segmenting the biological neural network using the fluorescent microscopy image data. The proposed method localizes the cellular bodies and distinguishes them from axons and dendrites.

The analysis of the image data is performed using the pipeline of the predefined image filters. In such way we obtain the contours of the neurons. Then the contours are interpolated by periodic cubic splines and the Voronoi skeleton of constructed. We also firstly introduced a mixed parametrization of contours for the problem of an approximate Voronoi diagram (skeleton) construction.

The proposed mixed parametrization allows us to sample contour points taking into account its local shape and controlling the influence of random noise and errors. The introduced $O(N \log N)$ Voronoi skeletonization algorithm allowed us to speed-up the computation of the neural network in comparison to the tracing methods. It also allows us to obtain directly the graph representation of the neural network with sub-pixel precision and apply numerous graph processing algorithm to analyze the properties of the network.

The practical significance of obtained results is that the software realizing the proposed segmentation algorithm is developed. The experiments showed that the algorithm allows us to segment the network of biological neurons fast and precisely (see section “Experiments”).

Thus, the algorithm can be applied to analyze the biological data and examine the properties of neural biological networks of living organisms. This in turn may facilitate the investigation in the field of microbiology and help to extend our knowledge and improve our understanding of the structures, functions, processes and mechanisms related to the biological neural networks.

Prospects for further research are to make further optimization of the proposed algorithm and apply the proposed algorithm to analyze the biological data using the graph processing algorithms for investigating the structural properties of the network of biological neurons.

ACKNOWLEDGEMENTS

This research was conducted as a part of the project “Development of a software-algorithmic environment of visualization and computer modeling for building the automated health recovery systems for military personnel” (beginning from 01.01.2019) at the department of Mathematical Informatics, Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv. The project was supported by Ukrainian Ministry of Education and Science.

REFERENCES

1. Mai J., Paxinos G. The Human nervous system, Third Edition. London, Academic Press, 2012, 1428 p.

2. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms, Third Edition. Cambridge, The MIT Press, 2009, 1312 p.
3. Berg M., Cheong O., Kreveld M., Overmars M. Computational Geometry: Algorithms and Applications, Third Edition. Berlin, Springer, 2008, 386 p. DOI: 10.1007/978-3-540-77974-2
4. Changxian S., Yulong M. Morphological thinning based on image's edges, *Communication Technology: International Conference, 22–24 October 1998 : proceedings*. Beijing, IEEE, 1998, Vol. 1, pp. 1–5. DOI: 10.1109/ICCT.1998.743232
5. Zhang T. Y., Suen C. Y. A fast parallel algorithm for thinning digital patterns, *Communications of the ACM*, 1984, Vol. 27, Issue 3, pp. 236–239. DOI: 10.1145/357994.358023
6. Liu L., Chambers E. W., Letscher D. et al. A simple and robust thinning algorithm on cell complexes, *Computer Graphics Forum*, 2010, Vol. 29, Issue 7, pp. 2253–2260. DOI: 10.1111/j.1467-8659.2010.01814.x
7. Saha P. K., Borgfors G., Baja G. S. A survey on skeletonization algorithms and their applications, *Pattern recognition letters*, 2016, Vol. 76, Issue C, pp. 3–12. DOI: 10.1016/j.patrec.2015.04.006
8. Stein A. M., Vader D. A., Jawerth L. M. et al. An algorithm for extracting the network geometry of three-dimensional collagen gels, *Journal of Microscopy*, 2008, Vol. 232, Issue 3, pp. 463–475. DOI: 10.1111/j.1365-2818.2008.02141.x
9. Zhou Z., Sorensen S. A., Peng H. Neuron crawler: An automatic tracing algorithm for very large neuron images, *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015, 1, pp. 870–874.
10. Xie J., Zhao T., Lee T. et al. Automatic neuron tracing in volumetric microscopy images with anisotropic path searching, *Medical Image Computing and Computer-Assisted Intervention: International Conference, 20–24 September 2010: proceedings*. Beijing, Springer, 2010, pp. 472–479. DOI: 10.1007/978-3-642-15745-5_58
11. Acciai L., Soda P., Iannello G. Automated Neuron Tracing Methods: An Updated Account, *Neuroinformatics*, 2016, Vol. 14, № 4, pp. 353–367. DOI: 10.1007/s12021-016-9310-0
12. Yang J., Gonzalez-Bellido P. T., Peng H. A distance-field based automatic neuron tracing method, *BMC Bioinformatics*, 2013, Vol. 14, No. 93. P. 93–104. DOI: 10.1186/1471-2105-14-93
13. Gonzales R. C., Woods R. E. Digital image processing, Third Edition. Missouri, Pearson Education, 2007, 976 p.
14. Okabe A., Boots BSugihara, K., Chiu S., Kendall D. G. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. London, J. Wiley & Sons, 2000, 696 p.
15. Ogniewicz R., Ilg M. Voronoi skeletons: theory and applications, *Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1992, 1, pp. 63–69.
16. Sharma O., Mioc D., Anton F. Voronoi diagram based automated skeleton extraction from colour scanned maps, *3rd International Symposium on Voronoi Diagrams in Science and Engineering : International Conference, 2–5 July 2006: proceedings*. Banff, IEEE, 2006, Vol. 1, pp. 186–195. DOI: 10.1109/ISVD.2006.39
17. Brandt J. W., Algazi V. R. Continuous skeleton computation by Voronoi diagram, *CVGIP: Image Understanding*, 1992, Vol. 3, Issue 55, pp. 329–338. DOI: 10.1016/1049-9660(92)90030-7
18. Fortune S. A sweepline algorithm for Voronoi diagrams, *Algorithmica*, 1987, Vol. 2, No. 1–4, pp. 153–174. DOI: 10.1007/BF01840357
19. Shamos M., Hoey D. Closest-point problems, *Proceedings of 16th Annual Symposium on Foundations of Computer Science*, 1975, 1, pp. 131–162. DOI: 10.1109/SFCS.1975.8
20. Ramamurthy R., Farouki R. T. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations, *Journal of Computational and Applied Mathematics*, 1999, Vol. 102, №1, pp. 119–141. DOI: 10.1016/S0377-0427(98)00211-8
21. Ramamurthy R., Farouki R. T. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries – II: Detailed algorithm description, *Journal of Computational and Applied Mathematics*, 1999, Vol. 102, No. 2, pp. 253–277. DOI: 10.1016/S0377-0427(98)00223-4
22. Farouki R. T., Ramamurthy R. Specified-Precision Computation of Curve/Curve Bisectors, *International Journal of Computational Geometry & Applications*, 1998, Vol. 8, No. 5–6, pp. 599–617. DOI: 10.1142/S0218195998000291
23. Alt H., Cheong O., Vigneron A. The Voronoi Diagram of Curved Objects, *Discrete & Computational Geometry*, 2005, Vol. 34, No. 3, pp. 439–453. DOI: 10.1007/s00454-005-1192-0
24. Hoff K., Culver T., Keyser J. et al. Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware, *Proceedings of 26th Annual Conference on Computer Graphics and Interactive Techniques*, 1999, 1, pp. 277–286. DOI: 10.1145/311535.311567
25. Har-Peled S. Geometric Approximation Algorithms, *Mathematical Surveys and Monographs*, 2011. 1, pp. 129–132.
26. Ho Y., Liu J. Collision-free curvature-bounded smooth path planning using composite Bezier curve based on Voronoi diagram, *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 2009, 1, pp. 463–468. DOI: 10.1109/CIRA.2009.5423161
27. [Kwak H. J., Lee S. J., Lee Y. et al. Quantitative study of total variation (TV) noise reduction algorithm with chest X-Ray imaging, *Journal of Instrumentation*, 2018, 1, pp. 1–14.
28. Frangi A. F., Niessen W. J., Vincken K. L. et al. Multiscale vessel enhancement filtering, *Proceeding of Medical Image Computing and Computer-Assisted Intervention*, 1998, 1, pp. 130–137.
29. Schneider P. An Algorithm for Automatically Fitting Digitized Curves, *Graphics Gems*. San Diego, Academic Press Professional, 1990, Section 11, pp. 612–626.
30. Pagani L., Scott P. J. Curvature based sampling of curves and surfaces, *Computer Aided Geometric Design*, 2018, Vol. 59, pp. 32–48.
31. Preparata F., Shamos M. Computational Geometry: An Introduction. Berlin, Springer, 1985, 390 p.
32. Cell Image Library [Electronic resource]. Access mode: <http://www.cellimagelibrary.org/home>

Received 18.01.2019.
Accepted 23.01.2019.

УДК 004.932.2:004.021

АЛГОРИТМ ПОБУДОВИ СКЕЛЕТОНУ ОБ'ЄКТА НА ОСНОВІ ДІАГРАМИ ВОРОНОГО ДЛЯ ЗАДАЧІ СЕГМЕНТАЦІЇ БІОЛОГІЧНОЇ НЕЙРОННОЇ МЕРЕЖІ

Коцур Д. В. – аспірант кафедри математичної інформатики факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, Київ, Україна.

Терещенко В. М. – д-р фіз.-мат. наук, професор, завідувач кафедри математичної інформатики, факультету комп'ютерних наук та кібернетики Київського національного університету імені Тараса Шевченка, Київ, Україна.

АНОТАЦІЯ

Актуальність. Проблема автоматизованої обробки та аналізу даних зображень з мікроскопу має велике значення з огляду на її значний вплив на дослідження та останні розробки в області біології та медицини. Ефективні алгоритми обробки зображень сприяють розробці нових медичних діагностичних засобів і терапевтичних методів, а також сприяють розширенню нашого знання про основні механізми і процеси в живих організмах. Первинним фокусом цієї роботи є обробка мікроскопічних зображень біологічної нейронної мережі. Робота має на меті полегшити подальші дослідження біологічної нейронної мережі, що призведе до розробки більш ефективних методів діагностики, профілактики та лікування пов'язаних із ними захворювань.

Метод. Запропоновано алгоритм сегментації мережі біологічних нейронів, що складається з декількох кроків. На першому кроці застосовуються процедури обробки зображень, які спрямовані на підвищення якості даних зображення та виділення контурів біологічної нейронної мережі. На другому кроці будується скелетон мережі, при цьому використовується діаграма Вороного для відрізків, що складають контур об'єкта. Скелетон на основі діаграми Вороного використовується на третьому кроці для ідентифікації клітинних тіл і виокремлення їх від аксонів і дендритів.

Результати. Розроблений алгоритм на основі діаграми Вороного дозволяє сегментувати окремі нейрони, локалізувати їх клітинні тіла, аксони, дендрити. Алгоритм також дозволяє представити нейронну мережу у вигляді графу. Структура даних діаграми Вороного дозволяє ефективно обчислити такий граф зі складністю $O(N \log N)$ операцій, де N кількість точок контуру. Запропонований метод сегментації був реалізований на мовах програмування C++/Python і протестований на флуоресцентних зображеннях отриманих з CellImageLibrary (CIL).

Висновки. Запропонований метод сегментації спрямований на полегшення вивчення біологічних нейронних мереж. Метод дозволяє швидко сегментувати біологічну нейронну мережу за $O(N \log N)$ операцій з допомогою діаграми Вороного для відрізків. Структура даних діаграми Вороного, у свою чергу, дозволяє отримати представлення відсегментованої нейронної мережі у вигляді графу з атрибутами. Таким чином, можуть бути застосовані класичні алгоритми обробки графів для аналізу нейронної мережі і обчислення її різноманітних характеристик (наприклад, знаходження числа зв'язків між окремими нейронами, найкоротшого шляху передачі сигналу між двома нейронами тощо).

КЛЮЧОВІ СЛОВА: мікроскопія, обробка зображень, діаграма Вороного, полігон Вороного, нейрони, сегментація, граф.

УДК 004.932.2:004.021

АЛГОРИТМ ПОСТРОЕНИЯ СКЕЛЕТОНУ ОБЪЕКТА НА ОСНОВЕ ДИАГРАММЫ ВОРОНОГО ДЛЯ ЗАДАЧИ СЕГМЕНТАЦИИ БИОЛОГИЧЕСКОЙ НЕЙРОННОЙ СЕТИ

Коцур Д. В. – аспирант кафедры математической информатики факультета компьютерных наук и кибернетики Киевского национального университета имени Тараса Шевченко, Киев, Украина.

Терещенко В. М. – д-р физ.-мат. наук, профессор, заведующий кафедрой математической информатики факультета компьютерных наук и кибернетики Киевского национального университета имени Тараса Шевченко, Киев, Украина.

АННОТАЦИЯ

Актуальность. Проблема автоматизированной обработки и анализа данных изображений с микроскопа имеет большое значение, учитывая ее значительное влияние на исследования и последние разработки в области биологии и медицины. Эффективные алгоритмы обработки изображений способствуют разработке новых медицинских диагностических средств и терапевтических методов, а также способствуют расширению нашего знания об основных механизмах и процессах, происходящих в живых организмах. Первичным фокусом этой работы является обработка микроскопических изображений биологической нейронной сети. Работа имеет целью облегчить дальнейшие исследования биологической нейронной сети, что приведет к разработке более эффективных методов диагностики, профилактики и лечения связанных с ними заболеваний.

Метод. Предложен алгоритм сегментации сети биологических нейронов, который состоит из нескольких шагов. На первом шаге применяются процедуры обработки изображений, направленные на повышение качества данных изображения и выделения контуров биологической нейронной сети. На втором шаге строится скелетон сети, при этом используется диаграмма Вороного для отрезков, составляющих контур объекта. Скелетон на основе диаграммы Вороного используется на третьем шаге для идентификации клеточных тел и выделения их от аксонов и дендритов.

Результаты. Разработанный алгоритм на основе диаграммы Вороного позволяет сегментировать отдельные нейроны, локализовать их клеточные тела, аксоны, дендриты. Алгоритм также позволяет представить нейронную сеть в виде графа. Структура данных диаграммы Вороного позволяет эффективно вычислить такой граф со сложностью $O(N \log N)$ операций, где N – количество точек контура. Предложенный метод сегментации был реализован на языках программирования C++/Python и протестирован на флуоресцентных изображениях полученных с CellImageLibrary (CIL).

Выводы. Предложенный метод сегментации направлен на облегчение изучения биологических нейронных сетей. Метод позволяет быстро сегментировать биологическую нейронную сеть за $O(N \log N)$ операций с помощью диаграммы Вороного для отрезков. Структура данных диаграммы Вороного, в свою очередь, позволяет получить представление отсегментированной нейронной сети в виде графа с атрибутами. Таким образом, могут быть применены классические алгоритмы обработки графов для анализа нейронной сети и вычисления ее различных характеристик (например, нахождение числа связей между отдельными нейронами, кратчайшего пути передачи сигнала между двумя нейронами).

КЛЮЧЕВЫЕ СЛОВА: микроскопия, обработка изображений, диаграмма Вороного, многоугольник Вороного, нейроны, сегментация, граф.

ЛІТЕРАТУРА / LITERATURA

1. Mai J. The Human nervous system, Third Edition / J. Mai, G. Paxinos. – London: Academic Press, 2012. – 1428 p.
2. Introduction to Algorithms, Third Edition / [T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein]. – Cambridge, The MIT Press, 2009. – 1312 p.
3. Computational Geometry: Algorithms and Applications, Third Edition / [M. Berg, O. Cheong, M. Kreveld, M. Overmars]. – Berlin : Springer, 2008. – 386 p. DOI: 10.1007/978-3-540-77974-2
4. Changxian S. Morphological thinning based on image's edges / S. Changxian, M. Yulong // Communication Technology: International Conference, 22–24 October 1998 : proceedings. – Beijing : IEEE, 1998. – Vol. 1. – P. 1–5. DOI: 10.1109/ICCT.1998.743232
5. Zhang T. Y. A fast parallel algorithm for thinning digital patterns / T. Y. Zhang, C. Y. Suen // Communications of the ACM. – 1984. – Vol. 27, Issue 3. – P. 236–239. DOI: 10.1145/357994.358023
6. A simple and robust thinning algorithm on cell complexes / [L. Liu, E. W. Chambers, D. Letscher et al.] // Computer Graphics Forum. – 2010. – Vol. 29, Issue 7. – P. 2253–2260. DOI: 10.1111/j.1467-8659.2010.01814.x
7. Saha P. K. A survey on skeletonization algorithms and their applications / P. K. Saha, G. Borgefors, G. S. Baja // Pattern recognition letters. – 2016. – Vol. 76, Issue C. – P. 3–12. DOI: 10.1016/j.patrec.2015.04.006
8. An algorithm for extracting the network geometry of three-dimensional collagen gels / [A. M. Stein, D. A. Vader, L. M. Jawerth et al.] // Journal of Microscopy. – 2008. – Vol. 232, Issue 3. – P. 463–475. DOI: 10.1111/j.1365-2818.2008.02141.x
9. Zhou Z. Neuron crawler: An automatic tracing algorithm for very large neuron images / Z. Zhou, S. A. Sorensen, H. Peng // IEEE 12th International Symposium on Biomedical Imaging (ISBI). – 2015. – 1. – P. 870–874.
10. Automatic neuron tracing in volumetric microscopy images with anisotropic path searching / [J. Xie, T. Zhao, T. Lee et al.] // Medical Image Computing and Computer-Assisted Intervention: International Conference, 20–24 September 2010: proceedings. – Beijing : Springer, 2010. – P. 472–479. DOI: 10.1007/978-3-642-15745-5_58
11. Acciai L. Automated Neuron Tracing Methods: An Updated Account / L. Acciai, P. Soda, G. Iannello // Neuroinformatics. – 2016. – Vol. 14, № 4. – P. 353–367. DOI: 10.1007/s12021-016-9310-0
12. Yang J. A distance-field based automatic neuron tracing method / J. Yang, P. T. Gonzalez-Bellido, H. Peng // BMC Bioinformatics. – 2013. – Vol. 14, № 93. – P. 93–104. DOI: 10.1186/1471-2105-14-93
13. Gonzales R. C. Digital image processing, Third Edition / R. C. Gonzales, R. E. Woods. – Missouri : Pearson Education, 2007. – 976 p.
14. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams / [A. Okabe, B. Boots, K. Sugihara, S. Chiu, D. G. Kendall]. London : J. Wiley & Sons, 2000 – 696 p.
15. Ogniewicz R. Voronoi skeletons: theory and applications / R. Ogniewicz, M. Ilg // Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – 1992. – 1. – P. 63–69.
16. Sharma O. Voronoi diagram based automated skeleton extraction from colour scanned maps / O. Sharma, D. Mioc, F. Anton // 3rd International Symposium on Voronoi Diagrams in Science and Engineering : International Conference, 2–5 July 2006: proceedings. – Banff : IEEE, 2006. – Vol. 1. – P. 186–195. DOI: 10.1109/ISVD.2006.39
17. Brandt J. W. Continuous skeleton computation by Voronoi diagram / J. W. Brandt, V. R. Algazi // CVGIP: Image Understanding. – 1992. – Vol. 3, Issue 55. – P. 329–338. DOI: 10.1016/1049-9660(92)90030-7
18. Fortune S. A sweepline algorithm for Voronoi diagrams / S. Fortune // Algorithmica. – 1987. – Vol. 2, № 1–4. – P. 153–174. DOI: 10.1007/BF01840357
19. Shamos M. Closest-point problems / M. Shamos, D. Hoey // Proceedings of 16th Annual Symposium on Foundations of Computer Science. – 1975. – 1. – P. 131–162. DOI: 10.1109/SFCS.1975.8
20. Ramamurthy R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations / R. Ramamurthy, R. T. Farouki // Journal of Computational and Applied Mathematics. – 1999. – Vol. 102, № 1. – P. 119–141. DOI: 10.1016/S0377-0427(98)00211-8
21. Ramamurthy R. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries – II: Detailed algorithm description / R. Ramamurthy, R. T. Farouki // Journal of Computational and Applied Mathematics. – 1999. – Vol. 102, № 2. – P. 253–277. DOI: 10.1016/S0377-0427(98)00223-4
22. Farouki R. T. Specified-Precision Computation of Curve/Curve Bisectors / R. T. Farouki, R. Ramamurthy // International Journal of Computational Geometry & Applications. – 1998. – Vol. 8, № 5–6. – P. 599–617. DOI: 10.1142/S0218195998000291
23. Alt H. The Voronoi Diagram of Curved Objects / H. Alt, O. Cheong, A. Vigneron // Discrete & Computational Geometry. – 2005. – Vol 34, № 3. – P. 439–453. DOI: 10.1007/s00454-005-1192-0
24. Fast Computation of Generalized Voronoi Diagrams Using Graphics Hardware / [K. Hoff, T. Culver, J. Keyser et al.] // Proceedings of 26th Annual Conference on Computer Graphics and Interactive Techniques. – 1999. – 1. – P. 277–286. DOI: 10.1145/311535.311567
25. Har-Peled S. Geometric Approximation Algorithms / S. Har-Peled // Mathematical Surveys and Monographs. – 2011. – 1. – P. 129–132.
26. Ho Y. Collision-free curvature-bounded smooth path planning using composite Bezier curve based on Voronoi diagram / Y. Ho, J. Liu // Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation. – 2009. – 1. – P. 463–468. DOI: 10.1109/CIRA.2009.5423161
27. Quantitative study of total variation (TV) noise reduction algorithm with chest X-Ray imaging / [H. J. Kwak, S. J. Lee, Y. Lee et al.] // Journal of Instrumentation. – 2018. – 1. – P. 1–14.
28. Multiscale vessel enhancement filtering / [A. F. Frangi, W. J. Niessen, K. L. Vincken et al.] // Proceeding of Medical Image Computing and Computer-Assisted Intervention. – 1998. – 1. – P. 130–137.
29. Schneider P. An Algorithm for Automatically Fitting Digitized Curves / P. Schneider // Graphics Gems. – San Diego : Academic Press Professional, 1990. – Section 11. – P. 612–626.
30. Pagani L. Curvature based sampling of curves and surfaces / L. Pagani, P. J. Scott // Computer Aided Geometric Design. – 2018. – Vol. 59. – P. 32–48.
31. Preparata F. Computational Geometry: An Introduction / F. Preparata, M. Shamos. – Berlin : Springer, 1985. – 390 p.
32. Cell Image Library [Electronic resource]. – Access mode: <http://www.cellimagelibrary.org/home>