

REVIEW

Automated Neuron Tracing Methods: An Updated Account

Ludovica Acciai¹ · Paolo Soda¹ · Giulio Iannello¹ 

Published online: 22 July 2016
© Springer Science+Business Media New York 2016

Abstract The reconstruction of neuron morphology allows to investigate how the brain works, which is one of the foremost challenges in neuroscience. This process aims at extracting the neuronal structures from microscopic imaging data. The great advances in microscopic technologies have made a huge amount of data available at the micro-, or even lower, resolution where manual inspection is time consuming, prone to error and utterly impractical. This has motivated the development of methods to automatically trace the neuronal structures, a task also known as neuron tracing. This paper surveys the latest neuron tracing methods available in the scientific literature as well as a selection of significant older papers to better place these proposals into context. They are categorized into *global processing*, *local processing* and *meta-algorithm* approaches. Furthermore, we point out the algorithmic components used to design each method and we report information on the datasets and the performance metrics used.

Keywords Neuron tracing · Neuron morphology · Digital reconstruction · Bioimage informatics · Neuroscience · BigNeuron

Introduction

The recent advances in microscopic imaging systems have made it possible to collect images at the macro-, the meso- and the microscale (Oh et al. 2014; Silvestri et al. 2012), changing how biologists visualize and study cellular structures. Among the numerous applications, recent efforts have been directed towards one of the foremost challenges in neuroscience, which is understanding how the brain works. In this respect, since manual reconstruction is time consuming and prone to error, the early attempts to obtain digital reconstruction of neuronal network can be traced back to 1965 (Glaser and Van der Loos 1965). The currently available high-resolution microscopic techniques allow to image a portion of the 86 billion neurons in the brain (Herculano-Houzel 2009), producing terabyte-sized volumes of images (Lichtman and Denk 2011). Meanwhile, the relevant improvements in computer science and computer vision in both computational power and algorithm sophistication observed in the last decades have increased the possibility to effectively exploit these data.

Reconstructing the neural morphology asks for the development of neuron tracing methods, i.e. the methods that automatically trace neurites. Despite the efforts reported in the scientific literature, this topic is still one of the main challenges in computational neuroscience (Donohue and Ascoli 2011; Meijering 2010). This is also testified by the DIADEM challenge (Ascoli 2008), short for Digital reconstructions of Axonal and DEndritic Morphology, that was launched in 2009 and concluded one year later. Its intent was to identify the most critical practical obstacles that needed to be overcome to advance this research area towards full automation. However, none of the finalist algorithms reached the originally projected goal of a 20-fold speed-up in the reconstruction process compared to manual

✉ Giulio Iannello
g.iannello@unicampus.it

Ludovica Acciai
l.accai@unicampus.it

Paolo Soda
p.soda@unicampus.it

¹ Department of Engineering, University Campus Bio-Medico of Rome, Roma, Italy

reconstruction (Liu 2011; Peng et al. 2015). Nevertheless, this challenge succeeded in stimulating a burst of progress as proven by the wheat of new methods that have been presented in recent years and by the launch of the BigNeuron project in March 2015, which aims in its first phase at gathering a worldwide community to define and advance the state-of-the-art of single neuron reconstruction (Peng et al. 2015).

Four papers gave a picture of the advances in this field until 2013 (Donohue and Ascoli 2011; Halavi et al. 2012; Meijering 2010; Parekh and Ascoli 2013). Meijering surveyed contributions published before 2010, discussing image pre-processing, soma segmentation methods, neuron tracing methods, quantitative measures of neuronal morphology, related software tools, and morphology databases (Meijering 2010). One year later, Donohue and Ascoli (2011) summarized the issues involved in automated reconstruction and shortly overviewed the available techniques. In 2012 Halavi et al. (2012) reviewed the general trends on specific animal species, brain regions and neuron types, whereas in 2013 Parekh and Ascoli summarized tools and resources for digital reconstruction (Parekh and Ascoli 2013). It is worth noting that none of these manuscripts entirely focused on neuron tracing methods and, therefore, a detailed survey is not available. On these grounds we decided to review the literature on neuron tracing.

Study Selection

Our literature research was performed from March 2015 to the end of October 2015 in the medical database Medline (1948–2015) and in the wide-ranging scientific database Web of Science (1970–2015) via PubMed and Google Scholar online services. Search terms were based on a combination of the following keywords: neuron, neurites, axons, dendrites, tracing, tracking, segmentation, DIADEM challenge and BigNeuron. After title, abstract and keyword screening, this research returned the four reviews (Donohue and Ascoli 2011; Halavi et al. 2012; Parekh and Ascoli 2013; Meijering 2010) and another twenty-six papers, whose full-text content was deeply analyzed. Ten are new papers published after the aforementioned reviews; another three were not found in previous surveys even if they appeared before them and they have been included in this work. Thirteen are discussed here and they are also cited in one existing review at least, we included them as they are the most widely cited in the literature and in the new published papers, and also because previous surveys only quickly overviewed them.

Taxonomy

The taxonomy we present here divides the methods into *global processing*, *local processing*, and *meta-algorithm*

approaches. Global approaches process the whole image, whereas the local ones explore an image only around relevant structures. Meta-algorithm approaches do not rely on a particular tracing algorithm, but rather enhance in some respect any existing method to manage large-scale images. They propose a strategy to apply any tracing algorithm designed either for reducing the computational workload or for dealing with image variability.

We also characterize methods pointing out the algorithmic components used to design each algorithm, i.e. the *functional elements* composing the tracing pipeline. In this respect, we identify six relevant algorithmic components: skeletonization, image transforms, seed generation, graph algorithms, deformable curves and supervised learning. While skeletonization, seed generation, graph algorithms and deformable curves are self-explanatory terms, the expressions “image transforms” and “supervised learning” here have the following meanings. The former indicates that a tracing method also leverages on a representation of the grey-level image in a new space, e.g. gradient vector field (Yuan et al. 2009). The latter means that a tracing method at a certain stage analyzes training data and produces an inferred function, which can be used for taking decisions on unseen traces. For instance, supervised classifiers have been used to estimate the quality of trace segments (Türetken et al. 2013) or to merge branches (Gala et al. 2014).

We also notice that a method can apply both a *pre-processing step* and a *post-processing step*. The first aims at removing noise, discarding undesirable features and enhancing the structures of interest. The second wishes to refine the neuronal tree and/or to discard false positive traces. It is straightforward that such a characterization is not applicable to meta-algorithm approaches, as they consist in a computational strategy to apply any tracing algorithm.

Table 1 groups the surveyed papers into the three main categories mentioned at the beginning of this section. It also reports the algorithmic components they exploited, and it points out whether a method employs a pre- and/or post-processing step. Furthermore, when multiple papers of the same group describe the same method, or even its evolution, only the most recent contribution is reported in Table 1 for the sake of presentation. Nevertheless, the older ones are cited when describing the method.

The rest of the manuscript is organized as follows: the next section gives a brief overview on the datasets used in the literature. Section “[Validation Metrics](#)” provides details on the metrics proposed for performance evaluation. In Section “[State of the Art](#)” we present a comprehensive description of the most relevant methods in the field, whereas the “[Discussion](#)” reports considerations about the state-of-the-art in neuron tracing in the light of the surveyed literature. Finally, Section “[Ongoing and Future Directions](#)” concludes the manuscript providing ongoing and future directions in the field.

Table 1 List of the reviewed papers divided by category

	Paper	Skeletonization	Image transforms	Seeds generation	Graph algorithms	Deformable curves	Supervised learning	Pre-processing	Post-processing
Global processing	Yuan et al. (2009)	✓	✓		✓			✓	✓
	Chothani et al. (2011)	✓				✓	✓		
	Wang et al. (2011)		✓	✓		✓		✓	
	Lee et al. (2012)	✓			✓				✓
	Myatt et al. (2012)			✓	✓				
	Türetken et al. (2013)		✓	✓		✓		✓	
	Xiao and Peng (2013)		✓		✓				✓
	Yang et al. (2013)	✓						✓	✓
	Basu et al. (2014)				✓	✓			
	Dietenbeck et al. (2014)			✓		✓	✓	✓	✓
	Gala et al. (2014)			✓	✓		✓		
	Sui et al. (2014)			✓		✓			
Local processing	Al-Kofahi et al. (2002)			✓					
	Srinivasan et al. (2007)			✓					
	Zhang et al. (2007)			✓					
	Bas and Erdogmus (2011)			✓	✓				
	Zhao et al. (2011)			✓	✓				
	Choromanska et al. (2012)			✓					
Meta-algorithms	Zhou et al. (2015b)								
	Zhou et al. (2015a)				Not applicable				
	Chen et al. (2015)								

Datasets

The importance of reconstructing the brain neuronal network has stimulated the collection of datasets dedicated to neuron tracing. It is possible to distinguish between primary and secondary data (Meijering 2010): the former usually consists of the raw images, whereas the latter includes any information derived from the primary data. Their public availability has two main benefits. On the one hand, it saves time and resources, that is, there is no need to prepare and to image new samples or pay for them, so researchers can focus on their particular algorithms and implementations. On the other hand, and this is even more important, the use of the same datasets facilitates the comparison of different approaches and gives insight into the abilities of the different methods.

While readers interested in the secondary data can find a detailed description of publicly available resources in Parekh and Ascoli (2013), researchers dealing with neuron tracing would be interested in finding primary open-data. In this respect, the DIADEM challenge (Ascoli 2008) was the

first relevant image repository publicly available up to the end of the first decade of the new millennium, allowing to test and to benchmark the tracing methods. We also observe that in the last few years the use of other private or public datasets has increased.

The methods we surveyed here were tested using not only the five datasets of the DIADEM challenge, but also another two public image datasets and further thirteen private repositories. Although a detailed review of these resources is out of the scope of this manuscript, Table 2 reports their details available within the scientific community in terms of types of microscopy, species, brain regions, type of neurons, voxel size, size and number of stacks. Furthermore, the last column of the table specifies the papers using a dataset in a row.¹

¹ The dataset used in Zhou et al. (2015b) is not reported in Table 2 since no information is available.

Table 2 Datasets used by the reviewed papers

Acronym	Availability	Microscopy	Species	Region	Type of neuron	Voxel size	# stacks	Average stack dimension	Used by
DIADeM ^a									
CA3	Public	Brightfield	Sprague-Dawley rat	Hippocampus	Interneuron	$0.217 \times 0.217 \times 0.33 \mu\text{m}^3$	9	$5092 \times 2820 \times 107$	(Choromanska et al. 2012; Zhao et al. 2011)
CCF	Public	Brightfield	Rat	Cerebellar cortex	-	$1 \times 1 \times 8.8 \mu\text{m}^3$	3	$6073 \times 3480 \times 29$	(Choromanska et al. 2012; Wang et al. 2011; Zhao et al. 2011)
L6	Public	Two-photon	Mouse	Visual cortex	Layer 5 neurons	$0.294 \times 0.294 \times 1 \mu\text{m}^3$	16	$512 \times 512 \times 58$	(Choromanska et al. 2012; Chothani et al. 2011; Gala et al. 2014; Zhao et al. 2011)
NMF	Public	Confocal	Mouse	Interscutularis muscle	Axonal trees	$0.2 \times 0.2 \times 0.2 \mu\text{m}^3$	152 + 156	$1024 \times 1024 \times 110$ and $512 \times 512 \times 90$	(Chothani et al. 2011; Dietenbeck et al. 2014; Wang et al. 2011)
OP	Public	Confocal	Drosophila	-	Olfactory axonal projection	$0.33 \times 0.33 \times 1 \mu\text{m}^3$	9	$512 \times 512 \times 78$	(Basu et al. 2014; Choromanska et al. 2012; Chothani et al. 2011; Gala et al. 2014; Lee et al. 2012; Sui et al. 2014; Wang et al. 2011; Xiao and Peng. 2013; Yang et al. 2013; Zhao et al. 2011)
M1	Private	Two-photon	Rat	Hippocampus	-	$0.125 \mu\text{m}^2$ pixels with z-step of $0.5 \mu\text{m}$	-	-	(Yuan et al. 2009)
M2	Private	Confocal	Rat	Somatosensory cortex	Layer 5 pyramidal neuron	$0.09 \times 0.09 \times 1 \mu\text{m}^3$	-	-	(Yuan et al. 2009)
M3	Private	Brightfield	Sprague-Dawley rat	Hippocampal CA1 area	Pyramidal neuron	-	5	$2862 \times 1649 \times 86$	(Myatt et al. 2012)
M4	Private	Brightfield	Pig	Piriform cortex	-	-	1	$3840 \times 3072 \times 99$	(Myatt et al. 2012)
M5	Private	Brightfield	Rat	-	-	-	6	-	(Türeken et al. 2013)
M6	Private	-	Mouse	Visual cortex	-	-	4	-	(Türeken et al. 2013)
M7	Private	Confocal	Wistar rat	-	-	-	20	$512 \times 480 \times 301$	(Al-Kofahi et al. 2002)
M8	Private	Confocal	Mouse	-	-	$0.1 \times 0.1 \times 0.2 \mu\text{m}^3$	3	-	(Srinivasan et al. 2007)

Table 2 (continued)

Acronym	Availability	Microscopy	Species	Region	Type of neuron	Voxel size	# stacks	Average stack dimension	Used by
M9	Private	Multi-channel fluorescence	-	Hippocampus	-	-	-	-	(Zhang et al. 2007)
M10	Private	Confocal	Mouse	-	Neuromuscular projection	-	1	$342 \times 342 \times 62$	(Bas and Erdognus 2011)
Non-mammal NM1	Public ^b	confocal	Drosophila	-	-	$0.32 \times 0.32 \times 1 \mu\text{m}^3$	120	-	(Chen et al. 2015; Lee et al. 2012)
NM2	Private	confocal	L. luctuosa	-	-	-	-	-	(Xiao and Peng, 2013; Yang et al. 2013; Zhou et al. 2015a)
NM3	Public ^c	-	Drosophila	-	-	-	-	-	(Xiao and Peng 2013)
NM4	Private	confocal	Drosophila	-	-	$0.17 \times 0.17 \times 1.0 \mu\text{m}^3$	-	-	(Yang et al. 2013)

^ahttp://diademchallenge.org/data_sets.html^b<http://flycircuit.tw/>^c<http://fweb.janelia.org/cgi-bin/flew.cgi>

Validation Metrics

The chief method for evaluating a tracing procedure is to compare the results with a gold standard (GS) reconstruction. However, the way to compare the reconstructed neuron tree and the GS is not unique, and different metrics can be found in the literature. The following paragraphs describe five different metrics, which were proposed and/or used in the reviewed papers. Table 3 summarizes the metrics used by each manuscript to measure the performance.

Distance Metric

The most widely used way to compare the reconstructed and the GS traces is based on the measure of a spatial error. This spatial error is commonly defined as the deviation of the reconstructed centerline to the GS centerline. Different Authors have used the following approaches to compute the distance metric (DM):

- DM1** The nearest point in the GS for each point in the reconstructed trace are located; then the error can be computed as the Euclidean distance in the xy plane and along the z axis (Myatt et al. 2012).
- DM2** Given the GS trace and the reconstructed trace, two different volumes are obtained using the following procedure. Each volume contains a neuronal tree obtained interpolating the nodes of a trace and then blurring this skeleton. Then two different similarity factors are gauged to estimate how well the GS matches the reconstruction and how well the reconstruction matches the GS, respectively (Choromanska et al. 2012).
- DM3** A bidirectional nearest neighbor search is performed in Gala et al. (2014), Lee et al. (2012), Wang et al. (2011), Xiao and Peng (2013). To this aim, in Wang et al. (2011), Xiao and Peng (2013) the two traces (reconstructed and GS) are resampled so that the distance between adjacent points is 1 voxel. Then, the metric computes the average Euclidean distance of all the nodes in the reconstructed traces to their nearest point in the GS, and vice versa. Thus, the spatial distance between the two traces is the average of these two measures.
- DM4** The error is computed as the ratio between the area bounded by the reconstructed trace and the GS trace to the length of the reconstructed trace (Zhang et al. 2007).

Branch Detection

In many cases some of the shortest terminal branches are missed in the reconstructed traces. To estimate this

Table 3 Validation metrics proposed and/or used in the reviewed papers

	Paper	Metrics				
		Distance	Branch detection	Length	Diadem	NetMets
Global processing	Yuan et al. (2009)					
	Chothani et al. (2011)					★
	Wang et al. (2011)	★		★	★	
	Lee et al. (2012)	★				
	Myatt et al. (2012)	★				
	Türetken et al. (2013)				★	★
	Xiao and Peng (2013)	★				
	Yang et al. (2013)		★			
	Basu et al. (2014)				★	
	Dietenbeck et al. (2014)			★		
	Gala et al. (2014)	★				
	Sui et al. (2014)					
Local processing	Al-Kofahi et al. (2002)					
	Srinivasan et al. (2007)	★				
	Zhang et al. (2007)	★		★		
	Bas and Erdogmus (2011)					
	Zhao et al. (2011)				★	
	Choromanska et al. (2012)	★	★	★		
Meta-algorithms	Zhou et al. (2015b)	★				
	Zhou et al. (2015a)	★				
	Chen et al. (2015)					

error it is useful to compare the number of detected end points (**BD1**) (Choromanska et al. 2012) or the number of detected branches (**BD2**) (Yang et al. 2013) in the reconstructed traces and GS. Furthermore, Choromanska et al. (2012) measured the precision, the recall and the accuracy of branching detection, counting as false positives the incorrectly detected branching regions, as true positives the correctly detected branching regions, and as false negatives the missing branches (**BD3**).

Length Metric

This metric estimates the difference in length between the GS and the reconstructed traces. It can be measured as one minus the ratio between the length of the reconstructed trace and the length of the GS trace (**LM1**) (Zhang et al. 2007). Wang et al. (2011) proposed another length metric (**LM2**), which was also adopted by Dietenbeck et al. (2014). Naming as “correct reconstructed trace” the portion of the reconstructed trace that overlaps the GS trace, **LM2** computes the precision as the ratio between the correct reconstructed trace length and the overall reconstructed trace length. Furthermore, the recall is defined as the ratio between the length of the correct reconstructed trace and the length of the GS trace.

DIADEM Metric

The DIADEM metric was released as an open source code to assess the performance of DIADEM challenge attendees’ algorithms (Gillette et al. 2011). This metric compares two neuron reconstructions on the basis of topology matches. For each node in the GS the corresponding node in the automatic reconstruction is searched, looking only in a restricted region centered on the GS point. Furthermore, the metric weighs each comparison result taking into account the degree of each node, i.e. the number of branches to which the node leads: as a consequence the mismatch of a node leading to more than one branch is worst than the mismatch of a node in a terminal branch. The metric scores if the reconstructed trace can capture the actual neuron topology, looking at whether the GS and reconstructed traces match in a neighborhood of each node.

NetMets Metric

This metric computes two separate scores named as geometry and connectivity measures, which are available online² (Mayerich et al. 2012).

²<http://www.davidmayerich.net/resources/software.shtml>.

Given two traces T_1 and T_2 , the geometric measure estimates the ratio between the length of the fibers in T_1 that do not exist in T_2 and the length of T_2 . Considering as T_1 the reconstructed trace and as T_2 the GS, this ratio estimates the geometry false positive rate. The geometry false negative rate is estimated reversing the roles of the reconstructed and GS traces.

The connectivity measure is given by the false positive and false negative rates using the conventional definition derived from a 2×2 confusion matrix. To this aim, a graph is built from each trace using the terminal points and branching points as vertices. The false negatives are the number of edges in the GS that are not in the reconstructed trace, the false positives are the number of edges in the reconstructed trace that do not exist in the GS, whereas the true positives are the number of edges that are in both traces.

State of the Art

In this section we review the methods selected as described in the study selection. Within each of the three categories, the contributions are presented in a chronological order, reporting also the datasets and the performance metrics used, if available.

Global Processing Approaches

Yuan et al. (2009) presented a method for dendrites and dendritic spines tracing based on 3D skeletonization. The images are first pre-processed using a commercial deconvolution package (AutodeblurTM) and subsequently using an anisotropic diffusion algorithm to smooth the images preserving the useful edges. To extract the skeleton from the gray-scale pre-processed image the method detects critical points, namely saddles or attracting points, of the gradient vector field. The skeleton is built starting from a saddle point and moving along its positive eigenvectors with one voxel step until an attractive point is encountered. This process is repeated until all the saddle points have been used as starting seed points. Neurite traces are obtained in three steps. Firstly, all the skeleton points are considered as vertices; secondly, the intensity weighted minimum spanning tree (IW-MST) is computed and, thirdly, IW-MST is refined using the minimum descriptor principle, since it might contain some errors. The method was tested on the M1 and M2 mammals datasets.

Chothani et al. (2011) combined an active contour approach with a learning-based branch merging procedure. First, to enhance the linear structures, the image is filtered using the LoG filter. Then, the voxel-coding algorithm (Zhou and Toga 1999) is applied to detect the centerline. However, due to noise and imperfect labeling, the obtained

trace may contain erroneous branches and small loops. A more accurate reconstruction is obtained sequentially applying two runs of the active contour method. The first run is initialized with the centerline detected by the voxel-coding algorithm. As in Vasilkoski and Stepanyants (2009), the first fitness function contains two terms, one related to the intensity along the trace, the other to the elasticity. The second run applies a fitness function designed to improve the placement of branch- and end-points. Finally, a branch merging procedure is performed. To this aim, the end-points of the branches are identified and grouped into spatial clusters. Within each cluster the optimal merging scenario, e.g. the lowest cost scenario, is determined. This procedure is based on a human inspired cost function, whose parameters are computed by the perceptron learning algorithm (Engel and Van den Broeck 2001) during user-assisted branch merging procedure. The Authors tested this method on OP, L6 and NMF datasets from the DIADEM challenge, and the performance was evaluated with the DIADEM metric (Gillette et al. 2011). It is worth noting that this work is a more sophisticated version of the method presented in Vasilkoski and Stepanyants (2009) and it was one of the DIADEM challenge finalist algorithms. Indeed, it contemplates two active contour steps rather than one, and it uses a cost function to determine the best branching topology in the detected tree.

Wang et al. (2011) proposed a 3D neuron tracing algorithm based on open-curve active contour, also named as open-curve snake model. The use of the snake model for neuron tracing was previously proposed by Schmitt et al. (2004) and it was also used in Chothani et al. (2011), Vasilkoski and Stepanyants (2009). After preprocessing, the gradient vector flow is computed to be used as deforming force in the open curve snake. Subsequently, the Frangi's vesselness filter (Frangi et al. 1998) is applied and the resulting image is binarized using the graph-cut segmentation algorithm (Boykov et al. 2001). Next, the seed set is defined applying a user selected method (i.e. maximum vesselness points) to the binarized image. The seeds are then sorted by a priority criterium and the open snake tracing model is initialized at the first seed point. At each iteration the snake grows while the snake energy function is minimized. The procedure stops either when the maximum number of iterations is reached, or when the snake hits the image boundaries, or if the snake length remains unchanged for a predefined number of iterations. Next, the snake is validated, and only if it satisfies a set of rules it is not rejected. If the snake is not rejected, the seed list is updated and any seed closer to the new trace than a user-defined threshold is deleted. The procedure starts again from the next seed and this process continues until no seeds are left. The method detects a branch point as the collision point of two snakes. The algorithm was tested on a synthetic dataset and on OP,

CCF and NMF datasets from the DIADEM challenge. Different methods were used to evaluate the performance: the Authors used the DM3 metric on a synthetic dataset, the DIADEM metric (Gillette et al. 2011) on CCF dataset and the LM2 metric on NMF dataset. It is worth noting that this is one of the DIADEM challenge finalist algorithms.

Lee et al. (2012) designed a method based on the shortest path algorithm to reconstruct the neuronal structure. This method is an updated version of the algorithm proposed in Lee et al. (2008). To gather candidate neuron voxels the skeleton is computed slice-by-slice thinning a binary image obtained by 3D thresholding. Examining the neighborhood of each skeleton voxel, the set of candidate end-points and the set of candidate branch-points are built. Furthermore, among the candidate neuron voxels the soma location is identified. Then, given the soma location as source point, the shortest path to all the end points is obtained with the Dijkstra's algorithm (Dijkstra 1959), where edge weights are computed as a function of the Euclidean distance between vertices and of the closeness to a branch-point, i.e. edges close to a branch-point have a larger weight allowing to keep the appropriate branch locations. Finally, a post-processing step aims at removing the end-points that are too close to the longest path among all the shortest paths computed, which could be false candidate end points. However, this post-processing could remove fine structures, making this method inadequate for a study where the fine details are important. In comparison with Lee et al. (2008), we notice that the Authors presented a more elaborate thresholding approach and a different strategy to compute edge weights. This method was originally designed for confocal image stacks, and it was tested on NM1 dataset and validated on OP dataset from the DIADEM challenge using the DM3 metric.

In Myatt et al. (2012) a semi-automatic approach needing to manually set both the start and the end points of the neurite, is presented. The first step identifies a set of seed points, using steerable gaussian filters to estimate their neuriteness. Next, the growth direction of the neurite is estimated at each candidate seed location using the eigenvectors of the Hessian matrix. Finally, the algorithm estimates the least cost path between manually provided start and end points. To this aim, the Dijkstra's algorithm (Dijkstra 1959) is run on the set of seed points using a combination of the neuriteness and the estimated growth direction as weights. The Authors tested this method on two different image stacks: M3 and M4 datasets. The performance was evaluated using the distance metric DM1. It is worth observing that the Authors released an implementation of this method within the Neuromantic software (www.reading.ac.uk/neuromantic).

Türetken et al. (2013) proposed an automated approach to delineate complex and potentially loopy networks. This

work improved both the algorithm originally proposed by the same Authors in the DIADEM challenge (Türetken et al. 2011) and its second version available in Türetken et al. (2012). The algorithm starts computing a tubularity measure for every voxel, which estimates the voxel likelihood of being on the centerline of a curvilinear structure. Then, considering regularly spaced high tubularity voxels as vertices, a graph is built using a shortest path approach with geodesic distances as weights. The obtained graph is an over-complete representation of the network. The final trace is obtained solving a mixed integer problem to compute the maximum-likelihood sub-graph of this graph. They considered two log likelihood term. The first accounts for the quality of geodesic paths, and to this aim probabilistic weights are assigned to pairs of consecutive edges using a path classification approach (Türetken et al. 2012). The second one is a prior term that penalizes unwarranted bifurcations or terminations. It is worth noting that in the DIADEM version of the method, which was one of the finalist algorithms, the final trace is detected using the k-MST of the graph (Türetken et al. 2011). The Authors deemed that the k-MST does not guarantee the global optimum of the solution and they also proved that the classification approach is more robust to image noise than spanning tree methods, since the latter rely on averaging tubularity scores. Furthermore, the main difference with their previous work (Türetken et al. 2012) is that they do not constrain the reconstruction to be a tree, but allow it to contain cycles. The comparison with this earlier version shows that it is more effective to relax the tree constraint. The method was tested on M5 and M6 mammal datasets, on a retinal ganglion dataset, and on images of loopy roads.³ The DIADEM metric was gauged on M5 and M6 datasets, whereas the NetMets metric was evaluated only on M5 dataset.

Xiao and Peng (2013) proposed an automatic algorithm for neuron tracing based on hierarchical pruning of a gray-scale weighted image distance-tree. This is a new version of the all-path-pruning algorithm (Peng et al. 2011) that aims at generating a more accurate reconstruction within a shorter amount of time. This method is also known as “all-path-pruning 2.0”. Thresholding is the first step providing background and foreground. Then a gray-weighted distance transform is computed for all the foreground pixels, so their intensity is replaced by the sum of pixels intensity along the shortest path from the pixel under consideration to the background. An initial neurite reconstruction is given by the shortest path from a single point to the other foreground pixels, obtained using the Dijkstra's algorithm (Dijkstra 1959). Since the result is an over-reconstruction of the neuron, the

³We do not report further details on these two datasets since they are not neuronal images datasets.

initial tree is dismantled to the level of individual neuron segments, each one connecting two branching points, sorted in hierarchical order: the first and most important segment is the longest path from the source node to the furthest leaf node. This segment is removed from the initial tree and the longest path search is repeated until no points are left in the initial tree. Then, a pruning procedure is performed to discard less important segments that overlap with more important segments. This method was tested on DIADEM OP dataset and on NM2 and NM3 datasets. The performance of this algorithm was evaluated using the distance metric DM3.

Yang et al. (2013) developed the DF-Tracing approach to tackle the neuron tracing challenge. It consists of three stages: (i) pre-processing to enhance line-like structures, (ii) skeletonization using two distance fields, (iii) assembling the spatially disconnected neuron segments obtained from the previous steps. In the first step the neuron signal is enhanced by a nonlinear anisotropic filter. Thresholding is then used to extract foreground objects, i.e. the neurites, as multiple disconnected segments. For each segment the set of boundary pixels B is defined as the pixels that have at least one neighbor in the background. The distance transform of an image region R with respect to another image region or point x is defined as a new image where the intensity of each voxel in R is replaced by the value of its shortest distance to x . Two distance transforms are then computed for each neuron segment: the distance transform of the neuron segment to the background, and the distance transform of the neuron segment to an arbitrary selected point in B . The skeleton of each neuron segment is obtained combining the results of these two distance fields. Finally the disconnected segments are merged at their convergence point, obtaining a tree-like structure. As the Authors reported, this method may fail on highly anisotropic 3D images. The method was tested on the non-mammals datasets NM2 and NM4 and on OP dataset from the DIADEM challenge. The performance of the algorithm was measured using the branch detection metric BD1.

Basu et al. (2014) presented an automated neuron tracing approach consisting of the following steps. First, the algorithm fits a set of spheres on the data, obtaining a set of objects that it considers as seeds. Furthermore, it simultaneously identifies the branch and terminal nodes. Next, an objective function consisting of two energy terms is computed on these objects. The first term is computed from the data and the second is a prior energy term that encourages survival of objects corresponding to critical points and discourages isolated objects. The configuration of the spherical objects that minimizes the objective function is the initial trace. Finally, the neuron tree is constructed as a minimum spanning tree. The method was tested on OP dataset from

the DIADEM challenge and the performance was evaluated with the DIADEM metric (Gillette et al. 2011).

Dietenbeck et al. (2014) described a method using both local and global features. It requires that the user identifies the number of fibers in the volume. First a pre-processing step is performed to enhance the fiber location via a random local probability filter (RLPF). Then, an SVM classifier is used to compute the posterior probability that a voxel x belongs to a fiber ($p_{svm}(x)$). To this aim, each voxel is represented in a feature space consisting of RLPF output combined with the output of steerable filters. Once $p_{svm}(x)$ is computed, x is considered a seed point if it is a local maximum and its $p_{svm}(x) > 0.5$. Next, particle filtering computes a connection map between nearby seeds. Finally, supervised seed clustering assigns each seed to its fiber. The method was tested on NMF dataset from the DIADEM challenge, and the performance was evaluated using the LM2 length metric. As the Authors reported, the method is not capable of fully automatically tracing the neuromuscular fibers since it needs multiple manual interactions per image volume.

Gala et al. (2014) proposed a methodology for neuron tracing based on two main steps: initial tracing and branch merging. In the former, they used the fast marching algorithm (Sethian 1999) initialized with multiple seed points to obtain an initial trace of the neurites. Even if this initial trace captures the structure of the neurites, it fails to detect the correct branching topology: for this reason the Authors dismantled the initial trace to the level of individual branches. In the latter step, the branches are first clustered using the relative distances of their terminal points. For each cluster nine features are computed; next, an SVM is applied on each cluster to determine how to connect the branches. An active learning approach (Settles 2012) is used; the Authors deemed that such an approach reduces the training time and the number of topological errors in automated traces with a few training examples. The method was tested on OP and L6 datasets from the DIADEM challenge, whereas its performance was evaluated using the DM3 distance metric. It is worth observing that the Authors released an implementation of this method in the NCTracer platform (www.neurogeometry.net).

Sui et al. (2014) proposed a neuron tracing method based on two main steps: seed points identification and neuron tracing. The first step is based on the spatial sliding volume filter (SVF), that is applied to detect the candidate seed points. Thus, the seed set is built sorting these seed candidates by their SVF response values. The second step solves the neuron tracing problem finding out the solution by minimizing an energy function of an open curve snake (Wang et al. 2011). The method was tested on OP dataset from the DIADEM challenge and on a synthetic dataset.

Local Processing Approaches

Al-Kofahi et al. (2002) proposed a neuron tracing method based on local processing. Assuming that the neurites cross section is elliptical and that they do not drastically change their growth direction, four templates are defined, namely left, right, top and bottom, that identify the corresponding boundary points as the points of their maximum template response. A set of seed points is needed to initialize the algorithm. To this aim, an over-complete seed set is given by the local maxima on a grid built on one 2D projection of the 3D image stack. The seeds are detected only in the xy plane; to determine the z coordinate of each point, the plane of the local maxima in the neighborhood of (x, y) is searched. Among all these candidates only the ones satisfying a set of conditions are considered as seed points, whereas the other are discarded. Once the set of seed points is defined, the algorithm iterates two main steps. Given a seed point \tilde{p}^i and the local growth direction \tilde{u}^i , the first step⁴ refines \tilde{p}^i and \tilde{u}^i computing the template responses, thus producing p^i and u^i . Next, the new point $\tilde{p}^{i+1} = p^i + \alpha^i u^i$ and direction $\tilde{u}^{i+1} = u^i$ are computed. These operations are iterated until one of the stopping criteria is reached (e.g. the traced structure's average intensity is one grey level lower than the local background intensity). The algorithm was tested on the mammals dataset M7.

Srinivasan et al. (2007) presented a hybrid method switching between 2D and 3D tracing. The 2D tracing method is an exploratory method based on the idea proposed by Al-Kofahi et al. (2002), but it is applied on 2D data (MIP). When a cross-over occurs, i.e. the current axon intersects a previously traced axon in the MIP image, the algorithm switches to 3D tracing. The 3D tracing method starts from a set of seed points determined by a modified mean-shift algorithm. Once the set of seeds has been defined, the actual 3D tracing is accomplished by segmenting the axons with the watershed algorithm guided by constraints. Two features are used to guide the growing: the orientation and the perimeter gauged from the ten previously segmented slices. Finally, since the result is a set of 2D and 3D traced segments, 2D segments are converted in 3D segments defining the third coordinate for each 2D point as the one corresponding to the maximum from which the 2D point was generated. The method was tested on the mammals dataset M8 and its performance was evaluated using the distance metric DM1.

Zhang et al. (2007) proposed a method working only on the MIP image of a 3D image stack. This proposal shares several aspects with the method proposed by Al-Kofahi

et al. (2002) and, consequently, also with the tracing method presented by Srinivasan et al. (2007). The method was tested on the mammals dataset M9, where the Authors considered only two randomly picked soma regions from each image. The performance was evaluated using the distance metric DM4 and the length metric LM1.

Bas and Erdogmus (2011) proposed a method relying on the notion of principal curves (Hastie and Stuetzle 1989). First, to reduce computational load, down-sampled data are projected to the closest principal curve using intensity values to estimate the sample probability distribution. Then, samples from the 1D principal set of the whole structure are selected as seed points. Starting from one seed at a time, branches are recursively traced by a Principal Curve Tracing (PCT) algorithm until a termination condition is met. After each branch termination, one more starting seed that has not been visited yet, and that is near already traced segments is chosen, and the PCT is run again. When a new starting point is chosen, the PCT is run in both directions, to find out if and where the segment under tracing connects with already traced structures. The tracing terminates when no more starting point candidates can be found. Since the recursive application of PCT may produce loops, especially because of the insufficient resolution in z direction, the method runs an MST algorithm on the traced segments, using voxel intensity values as weights. The topology of the reconstructed tree is not univocally determined; indeed it depends on the order with which seeds are considered for tracing. The method was tested on OP and CCF datasets from the DIADEM challenge and on the mammal dataset M10. It is worth noting that this is one of the DIADEM challenge finalist algorithms.

Zhao et al. (2011) proposed a local tracing algorithm followed by the shortest path algorithm to optimize the trace (Zhao et al. 2011). Assuming that a neurite segment can be approximated by a series of elliptical cylinders, they defined a 3D cylindrical template. The algorithm requires at least one seed for each branch. Seed points are defined as local maxima among the points with both intensity and local geometry features above a threshold. The first seed s_1 is taken as starting point c^0 . The algorithm iterates the following steps: (i) fit the cylinder template on the current point c^i ; (ii) define the next point c^{i+1} moving from c^i one step along the axial direction. Note that this tracing procedure ends if the fitting score of c^i is lower than a threshold, or if previously traced regions are encountered. When the tracing started from s_1 ends, all the seeds covered by this trace are deleted from the seed set and the next trace begins from the new s_1 , until the seed set is empty. The result of this exploratory tracing procedure is a set of disconnected branch segments. An undirected graph is built considering each segment as a node, where a node is composed of three parts: two *ends* and a *body*. The edges between the nodes

⁴The interested readers may refer to equations 6 and 7 in (Al-Kofahi et al. 2002) for the formal presentation of this step.

are found solving a minimum weighted spanning tree problem, where the weights are defined as the distance between two points to be connected. The possible connections are only *end-to-end* and *end-to-body* and only one connection is possible between two different nodes. This method, that was one of the finalist algorithm of the DIADEM challenge, was tested on OP, L6, CA3 and CCF datasets from the DIADEM challenge, and the performance was evaluated with the DIADEM metric (Gillette et al. 2011). As the Authors reported, this method can fail when many branches are in a small region of the volume; and when the dendrite signal is not continuous due to sample preparation issues.

Choromanska et al. (2012) presented an algorithm that progressively extends the neuronal tree starting from a seed point and analyzing a set of local morphological properties. The main steps of the algorithm are: (i) the user manually sets the center and the radius R of a sphere; (ii) the set S of candidate seed points is given by all the points connected to the center of the sphere that are within a margin from the sphere surface; (iii) the center of the sphere is moved to the point in S that minimizes an objective function; (iv) the set S is updated; (v) if the spread γ of points in S is lower than a user defined threshold t , steps (iii) and (iv) are repeated, otherwise the algorithm runs steps (vi) and (vii); (vi) S is divided into n subsets of connected points and $\Gamma = \{\gamma_1, \dots, \gamma_n\}$ is computed; (vii) if $\gamma_i > t$ exists, the algorithm increases R by one and it executes from steps (ii), otherwise steps (iii)–(v) are executed for each subset. Note that at step (vii) the algorithm ends when R is equal to a maximum value defined by the user. The datasets used in the study are OP, L6, CA3 and CCF all from the DIADEM challenge. Tracing performance is evaluated using the distance metric DM1 and the branch detection metrics BD1 and BD3.

Meta-Algorithm Approaches

Zhou et al. (2015a) proposed a fully automatic tracing strategy to gain efficient computation for large-scale images. It first traces the projections on 2D planes, and then reconstructs the 3D neuronal tree using a reverse mapping technique, i.e. 3D Virtual Finger (Peng et al. 2014b). The 2D tracing can be performed with any neuron tracing method, for instance in Zhou et al. (2015a) the tracing framework was tested using all-path-pruning 2.0 (Xiao and Peng 2013). After reverse mapping, the MST method is used to connect all 3D curves to produce the final neuron reconstruction. However, 3D Virtual Finger has one major limitation, if two or more structures overlap in the 2D projection, only one would be traced in the 3D space. To address this issue, the Authors used multiple projections from different angles to make sure that all neuron structures can be detected. The method accuracy was computed on the NM2 dataset

using the DM3 metric considering as GS the output provided by all-path-pruning 2.0 with input terminal markers. The results show that such a framework has comparable tracing accuracy to other state-of-the-art methods, but lower computational cost in terms of memory usage and tracing speed.

Zhou et al. (2015b) presented an automatic tracing framework based on the core idea of sequentially analyzing small tiles. Given a large-scale image and a root point, e.g. the soma, it first extracts a small tile centered in the root and elaborates it using a 3D tracing algorithm. Then, it detects the terminal nodes in the current tile and processes only the tiles that are adjacent to these nodes. This framework is able only to trace the projections of a single neuron at a time, but the Authors claimed that it has a lower computational cost than the current state-of-the-art methods, and that it could be extended in principle to trace multiple neurons in parallel.

Chen et al. (2015) developed an automatic tracing framework aiming at overcoming the variability among methods given by the differences in image modality, image parameters or tissue processing protocol. First, a user-provided tracing method is applied to detect an initial reconstruction of the neuron. Then, the trace is decomposed in segments, and voxels belonging to segments labeled as *reliable* and *unreliable* according to a given criterion are used to build a training set. After training an SVM, the classifier distinguishes foreground voxels, i.e. neuron voxels, from background voxels. On the basis of the classifier output the original image is adjusted, i.e. all the intensity of all the background voxels is set to 0. Then, the selected tracing algorithm is applied on this adjusted image to obtain a better tracing result. The whole framework was tested on the NM1 dataset using as tracing algorithm all-path-pruning 2.0 (Xiao and Peng 2013). By visual inspection, the Authors noted that the framework is capable of correctly tracing neurons that present gaps, that are conversely traced as disjunct segments by all-path-pruning 2.0. Even if the idea of the framework is smart, and it would effectively allow an automatic and self-adapting tracing method, it has two major limitations. First, its effectiveness relies on the adopted tracing method, e.g. if such method is not capable to detect some regions of the neuronal tree, the classifier would probably not detect them. Second, it has non trivial computational costs, making unfeasible its application on large-scale images, unless a strategy is devised to limit its use to regions where it is really needed.

Discussion

Looking at the temporal evolution of the literature, up to 2014 the scientific community has observed a growing interest in global processing approaches compared to local

processing ones. We speculate that this trend was motivated by at least two reasons. First, global methods do not require a well defined starting point to trace tree-like structures. In this sense they look more general, since they seem better suited to trace also structures (e.g. muscular fibers, retinal ganglion cells) different from neurites outgoing from the soma of a neuron. Second, local methods may partially fail when structures are interrupted and/or when the signal is weak and it can be confused with the background. These situations occur more frequently in very large (even terabyte-sized) data that are being produced by current imaging technologies, which can contain multiple neurons and potentially long neurites. Indeed, such large images may present non-homogeneous regions and high variable signals due, for instance, to the clearing procedure adopted (Chung et al. 2013).

Global methods for their very nature are characterized by the presence of multiple functional elements since they tend to over-segment images and therefore need to filter or somehow adjust the first tracing. In this respect, it is worth noting that several of them require a post-processing step to further discard false positive segments. As to the functional elements used by global methods, no clear trend emerges. We only point out that the most recent proposals rely on more sophisticated mathematical tools, such as deformable curves and supervised learning. Conversely, local methods look more compact, without pre- and post-processing steps, managing signal variability and introducing a stopping criterion.

A functional element common to most global processing and local processing proposals is the generation of seeds that, however, plays different roles in the two approaches. In global processing methods, neurites are traced trying to optimally connect seeds, initially assumed as points belonging to relevant structures with high probability. These points are always automatically produced in a large number. In local processing approaches, seeds are the starting points of the tracing procedure and they are generated either manually or automatically. In the former case, this requires user intervention that can be unfeasible or time consuming with the large images nowadays acquired. In the latter case, automatic generation methods employed in local processing approaches tend to filter out a higher number of candidates, looking for seeds that localize somas, or at least that are likely to be the end points of thin structures.

Focusing on 2015, we observe a raising interest in the meta-algorithms approaches. Two of them address issues related to computational complexity, trying to prevent expensive computations on the whole volumes or large parts of it. While the approach presented in Zhou et al. (2015a) might require too much 2D projections when multiple complex structures populate the 3D image, the idea proposed in Zhou et al. (2015b), which processes only relatively small

tiles where previously traced structures continue, seems more natural and adoptable by most approaches. Nevertheless, both global and local approaches should be redesigned when processing large images through tiling. Indeed, in this case the rationale behind the tiled approach is that somas of neurons of interest are first localized and then the tracing starts from them. This means that the existing methods may require some adaptations since the majority of them does not exploit the soma location.

Still in 2015, new learning-based approaches to segment thin structures have been proposed. Although they are not specifically focused on neuron tracing, their ability to provide a better segmentation of filamentary structures can be incorporated into any neuron tracing pipeline to boost its performance. In Gu and Cheng (2015) an iterative two-step learning-based approach processes the confidence map provided by an existing segmenter. In the first step, filamentary segments characterized by low confidence are detected using a structured learning approach. In the second step, detected fragments are connected to the already segmented structure using completion fields to augment the confidence map, followed by a new utilization of the matting technique. The two steps are then repeated until a termination condition is reached. In Sironi et al. (2015) a different approach is used to enhance the performance of a regression-based segmenter that provides the initial confidence map. Patches of the confidence map are then systematically replaced by their nearest neighbors in a set of ground truth training patches, which approximates the manifold of all admissible ground truth maps. Both these methods testify an emerging trend in segmentation algorithms, that is to model in a learning context the strong relationships existing among neighboring points. This may help to overcome some of the limitations exhibited by many neuron tracing methods.

Turning our attention to data employed to assess the method performance, we observe that for a long time the only publicly available datasets were released within the DIADEM challenge. Nevertheless, in the last few years the use of private or public datasets has increased, but in most cases they are only partially described (see Table 2). To make things worse, methods were often tested on datasets that differ in many respects: from almost isotropic images to highly anisotropic ones, images with different resolutions in xy , images of different sizes, etc..

Focusing now on performance evaluation, we notice that in many cases different metrics have been used. This makes it impossible to rank the methods, or even to compare the different approaches. Moreover, the used metrics suffer from the variability that can be expected in manually traced GSs: consequently, algorithm performance could be biased if a single GS was used in conjunction with distance-based metrics. An alternative could be to follow Gala et al. (2014), where multiple GSs are used, inter-GSs distance evaluated

and then used as a baseline to assess the effectiveness of a tracing method. We also observe that metrics based on distance are still prevalent, even if they may fail in evaluating the correct identification of neurites topology. Indeed, a correctly identified, but disconnected neuron trace, e.g. composed of multiple neuron segments, could lead to a very good score, even if neurite topology was completely missed. In this respect, we observed that only the NetMets metric takes into consideration both distance and topology, producing two separate scores (Mayerich et al. 2012). Furthermore, the reviewed papers do not include an analysis of the computational complexity. Only a few of them, and only recently, provide performance data in terms of measured execution times. This makes it currently impossible to compare the methods and to assess them in terms of scalability. For these reasons we deem that the definition of standard methods to effectively measure tracing performance and to enable comparisons is a topic deserving more attention.

Ongoing and Future Directions

Perspectives on neuron tracing have been proposed by the BigNeuron project, which was launched in March 2015 with the goal to “define and advance the state of the art of single-neuron reconstruction, develop a tool-kit of standardized reconstruction protocols, analyze neuron morphologies, and establish a data resource for neuroscience” (Peng et al. 2015). In the first, currently ongoing phase, the project addresses most of the issues raised in the introduction. Indeed, the project aims to define standardized protocols and evaluation methods of neuron tracing algorithms, to include them into a common software platform, to characterize their performance with respect to specific neuron image datasets, to release a rich collection of consensus reconstructions created from the output of individual algorithms. Moreover, a fairly rich dataset has been made available to the scientific community, simply requiring to join the project. The project also proposes to adopt the publicly available Vaa3D platform⁵ (Peng et al. 2014a) as a shared development environment to enable future benchmarking of neuron tracing methods.

To conclude this review, we would add a few considerations about the ultimate goals for which neuron tracing is getting so much interest. From a biological perspective, we can identify at least three kinds of problems that would benefit from digital neuron reconstruction: (i) characterizing cell types by morphology, including in wild-type and model animals, (ii) mapping projections between different brain regions, (iii) studying projections of single cells in

whole brain images. The state-of-the art of automatic neuron tracing is still in an early stage with respect to these goals. So far efforts have been mainly addressed to develop tools that can reconstruct the morphology of single neurons. However, on the one hand it is not clear what degree of precision is needed to provide a sufficient characterization of cell types, and, on the other hand, no algorithm is currently able to provide a reliable reconstruction without some degree of human intervention. As to another two issues, while computer assisted tracing of axon bundles is likely to be within the reach of current imaging technology and tissue clearing procedures, further technological advances are probably needed before single long-range projection can be automatically traced in whole brain acquisitions. Indeed, in this case even a single error in axon reconstruction would yield incorrect results. Such occurrences seem likely when taking into account current signal-to-noise ratio variability in large images as well as their voxel resolution, which is still hardly below one micron, especially in the *z* direction.

Information Sharing Statement

This work is entirely based on publicly available papers.

Acknowledgments The Authors are grateful to the anonymous referees for their valuable comments which greatly helped to improve the first version of this paper.

Compliance with Ethical Standards

Conflict of interests None declared. All the authors are supported by University Campus Bio-Medico of Rome.

References

- Al-Kofahi, K., Lasek, S., Szarowski, D.H., Pace, C.J., Nagy, G., Turner, J.N., Roysam, B., et al. (2002). Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Transactions on Information Technology in Biomedicine*, 6(2), 171–187.
- Ascoli, G.A. (2008). Neuroinformatics grand challenges. *Neuroinformatics*, 6(1), 1–3.
- Bas, E., & Erdogmus, D. (2011). Principal curves as skeletons of tubular objects. *Neuroinformatics*, 9(2-3), 181–191.
- Basu, S., Tsang, O.W., & Racocanu, D. (2014). Improved marked point process priors for single neurite tracing. In *2014 International Workshop on Pattern Recognition in Neuroimaging* (pp. 1–4): IEEE.
- Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 1222–1239.
- Chen, H., Xiao, H., Liu, T., & Peng, H. (2015). Smarttracing: self-learning-based neuron reconstruction. *Brain Informatics*, 2(3), 135–144.

⁵www.vaa3d.org.

- Choromanska, A., Chang, S.F., & Yuste, R. (2012). Automatic reconstruction of neural morphologies with multi-scale tracking. *Frontiers in neural circuits* 6.
- Chothani, P., Mehta, V., & Stepanyants, A. (2011). Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9(2-3), 263–278.
- Chung, K., Wallace, J., Kim, S.Y., Kalyanasundaram, S., Andalman, A.S., Davidson, T.J., Mirzabekov, J.J., Zalocusky, K.A., Mattis, J., Denisin, A.K., et al. (2013). Structural and molecular interrogation of intact biological systems. *Nature*, 497(7449), 332–337.
- Dietenbeck, T., Varray, F., Kybic, J., Basset, O., & Cachard, C. (2014). Neuromuscular fiber segmentation through particle filtering and discrete optimization. In *SPIE Medical Imaging, International Society for Optics and Photonics* (pp. 90,340B–90,340B).
- Dijkstra, E.W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1), 269–271.
- Donohue, D.E., & Ascoli, G.A. (2011). Automated reconstruction of neuronal morphology: an overview. *Brain Research Reviews*, 67(1), 94–102.
- Engel, A., & Van den Broeck, C. (2001). *Statistical mechanics of learning*: Cambridge University Press.
- Frangi, A.F., Niessen, W.J., Vincken, K.L., & Viergever, M.A. (1998). Multiscale vessel enhancement filtering. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 1998* (pp. 130–137): Springer.
- Gala, R., Chapeton, J., Jitesh, J., Bhavsar, C., & Stepanyants, A. (2014). Active learning of neuron morphology for accurate automated tracing of neurites. *Frontiers in neuroanatomy* 8.
- Gillette, T.A., Brown, K.M., & Ascoli, G.A. (2011). The diadem metric: comparing multiple reconstructions of the same neuron. *Neuroinformatics*, 9(2-3), 233–245.
- Glaser, E., & Van der Loos, H. (1965). A semi-automatic computer-microscope for the analysis of neuronal morphology. *IEEE Transactions on Biomedical Engineering* (1), 22–31.
- Gu, L., & Cheng, L. (2015). Learning to boost filamentary structure segmentation. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 639–647).
- Halavi, M., Hamilton, K.A., Parekh, R., & Ascoli, G.A. (2012). Digital reconstructions of neuronal morphology: three decades of research trends. *Frontiers in neuroscience* 6.
- Hastie, T., & Stuetzle, W. (1989). Principal curves. *Journal of the American Statistical Association*, 84(406), 502–516.
- Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience* 3.
- Lee, P.C., Ching, Y.T., Chang, H., & Chiang, A.S. (2008). A semi-automatic method for neuron centerline extraction in confocal microscopic image stack. In *2008. ISBI 2008. 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (pp. 959–962): IEEE.
- Lee, P.C., Chuang, C.C., Chiang, A.S., & Ching, Y.T. (2012). High-throughput computer method for 3d neuronal structure reconstruction from the image stack of the drosophila brain and its applications. *PLoS Computer Biology*, 8(9), e1002658.
- Lichtman, J.W., & Denk, W. (2011). The big and the small: challenges of imaging the brain's circuits. *Science*, 334(6056), 618–623.
- Liu, Y. (2011). The diadem and beyond. *Neuroinformatics*, 9(2), 99–102.
- Mayerich, D., Björnsson, C., Taylor, J., & Roysam, B. (2012). Netmets: software for quantifying and visualizing errors in biological network segmentation. *BMC Bioinformatics*, 13(Suppl 8), S7.
- Meijering, E. (2010). Neuron tracing in perspective. *Cytometry Part A*, 77(7), 693–704.
- Myatt, D.R., Hadlington, T., Ascoli, G.A., & Nasuto, S.J. (2012). NeurOmatic—from semi-manual to semi-automatic reconstruction of neuron morphology. *Frontiers in neuroinformatics* 6.
- Oh, S.W., Harris, J.A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., Lau, C., Kuan, L., Henry, A.M., et al. (2014). A mesoscale connectome of the mouse brain. *Nature*, 508(7495), 207–214.
- Parekh, R., & Ascoli, G.A. (2013). Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron*, 77(6), 1017–1038.
- Peng, H., Long, F., & Myers, G. (2011). Automatic 3d neuron tracing using all-path pruning. *Bioinformatics*, 27(13), i239–i247.
- Peng, H., Bria, A., Zhou, Z., Iannello, G., & Long, F. (2014a). Extensible visualization and analysis for multidimensional images using vaa3d. *Nature Protocols*, 9(1), 193–208.
- Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., Zhou, Z., Gonzalez-Bellido, P.T., Oh, S.W., Chen, J., et al. (2014b). Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature communications* 5.
- Peng, H., Meijering, E., & Ascoli, G.A. (2015). From diadem to bigneuron. *Neuroinformatics*, 13(3), 259–260.
- Schmitt, S., Evers, J.F., Duch, C., Scholz, M., & Obermayer, K. (2004). New methods for the computer-assisted 3-d reconstruction of neurons from confocal image stacks. *NeuroImage*, 23(4), 1283–1298.
- Sethian, J.A. (1999). *Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science* Vol. 3: Cambridge University Press.
- Settles, B. (2012). Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1), 1–114.
- Silvestri, L., Bria, A., Sacconi, L., Iannello, G., & Pavone, F.S. (2012). Confocal light sheet microscopy: micron-scale neuroanatomy of the entire mouse brain. *Optics Express*, 20(18), 20,582–20,598.
- Sironi, A., Lepetit, V., & Fua, P. (2015). Projection onto the manifold of elongated structures for accurate extraction. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 316–324).
- Srinivasan, R., Zhou, X., Miller, E., Lu, J., Litchman, J., & Wong, S.T. (2007). Automated axon tracking of 3d confocal laser scanning microscopy images using guided probabilistic region merging. *Neuroinformatics*, 5(3), 189–203.
- Sui, D., Wang, K., Chae, J., Zhang, Y., & Zhang, H. (2014). A pipeline for neuron reconstruction based on spatial sliding volume filter seeding. *Computational and mathematical methods in medicine* 2014.
- Türetken, E., González, G., Blum, C., & Fua, P. (2011). Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2-3), 279–302.
- Türetken, E., Benmansour, F., & Fua, P. (2012). Automated reconstruction of tree structures using path classifiers and mixed integer programming. In *2012 IEEE Conference on Computer vision and pattern recognition (CVPR)* (pp. 566–573): IEEE.
- Türetken, E., Benmansour, F., Andres, B., Pfister, H., & Fua, P. (2013). Reconstructing loopy curvilinear structures using integer programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1822–1829).
- Vasilkoski, Z., & Stepanyants, A. (2009). Detection of the optimal neuron traces in confocal microscopy images. *Journal of Neuroscience Methods*, 178(1), 197–204.
- Wang, Y., Narayanaswamy, A., Tsai, C.L., & Roysam, B. (2011). A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3), 193–217.
- Xiao, H., & Peng, H. (2013). App2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11), 1448–1454.

- Yang, J., Gonzalez-Bellido, P.T., & Peng, H. (2013). A distance-field based automatic neuron tracing method. *BMC Bioinformatics*, 14(1), 93.
- Yuan, X., Trachtenberg, J.T., Potter, S.M., & Roysam, B. (2009). Mdl constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7(4), 213–232.
- Zhang, Y., Zhou, X., Degterev, A., Lipinski, M., Adjero, D., Yuan, J., & Wong, S.T. (2007). A novel tracing algorithm for high throughput imaging: Screening of neuron-based assays. *Journal of Neuroscience Methods*, 160(1), 149–162.
- Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., Long, F., & Myers, E. (2011). Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9(2-3), 247–261.
- Zhou, Y., & Toga, A.W. (1999). Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3), 196–209.
- Zhou, Z., Liu, X., Long, B., & Peng, H. (2015a). Tremap: Automatic 3d neuron reconstruction based on tracing, reverse mapping and assembling of 2d projections. *Neuroinformatics*, 1–10.
- Zhou, Z., Sorensen, S.A., & Peng, H. (2015b). Neuron Crawler: An automatic tracing algorithm for very large neuron images. In *ISBI*.