A Graph-theoretical Approach for Tracing Filamentary Structures in Neuronal and Retinal Images

Jaydeep De, Li Cheng, Xiaowei Zhang
Bioinformatics Institute, A*STAR, Singapore
jaydeepd,chengli,zhangxw@bii.a-star.edu.sg
Feng Lin
Nanyang Technological University, Singapore
asflin@ntu.edu.sg
Huiqi Li
Beijing Institute of Technology, China
huiqili@bit.edu.cn
Kok Haur Ong, Weimiao Yu
Institute of Molecular and Cell Biology, A*STAR, Singapore
khong,wmyu@imcb.a-star.edu.sg
Yuanhong Yu, Sohail Ahmed
Institute of Medical Biology, A*STAR, Singapore
mike.yu,sohail.ahmed@imb.a-star.edu.sg

***Abstract*— The aim of this study is about tracing filamentary structures in both neuronal and retinal images. It is often crucial to identify single neurons in neuronal networks, or separate vessel tree structures in retinal blood vessel networks, in applications such as drug screening for neurological disorders or computer-aided diagnosis of diabetic retinopathy. Both tasks are challenging as the same bottleneck issue of filament *crossover*s is commonly encountered, which essentially hinders the ability of existing systems to conduct large-scale drug screening or practical clinical usage. To address the filament crossovers' problem, a two-step graph-theoretical approach is proposed in this paper. The first step focuses on segmenting filamentary pixels out of the background. This produces a filament segmentation map used as input for the second step, where they are further separated into disjointed filaments. Key to our approach is the idea that the problem can be reformulated as label propagation over *directed graph*s, such that the graph is to be partitioned into disjoint sub-graphs, or equivalently, each of the neurons (vessel trees) is separated from the rest of the neuronal (vessel) network. This enables us to make the interesting connection between the tracing problem and the digraph matrix-forest theorem in algebraic graph theory for the first time. Empirical experiments on neuronal and retinal image datasets demonstrate the superior performance of our approach over existing methods.**

***Index Terms*— Directed graph theory, tracing filamentary structures, neuronal tracing, retinal blood vessel tracing**

I. Introduction

Diseases of the central nervous system (CNS) such as Parkinson's and Alzheimer's are caused by loss of neurons and their connections. To identify drugs to treat CNS diseases, it is important to conduct high throughput screens (HTS) of microscopic neuronal images, which naturally requests for automated neuronal tracers. However, most existing systems are semi-automatic, where human guidance is frequently required during the tracing process. The widely used ImageJ plugins such as NeuriteTracer [1], Simple Neurite Tracer [2], as well as the Vaa3D neuron module [3] all fall under this category. Meanwhile, there are a few state-of-the-art systems that are able to trace neuronal structures automatically, including both academic efforts such as NeuroCyto [4] and the commercial product — the neurite tracing module of Metamorph NX [5]. Nevertheless, they often fail to reliably trace neurons in the presence of neurite *crossover*s, i.e. overlaps or touching neurites.

A very similar situation has also been observed in tracing retinal blood vessel trees from fundus images. There, topological and geometrical properties of retinal vessel trees provide valuable clinical information in diagnosing diseases such as proliferative diabetic retinopathy, glaucoma, and hypertensive retinopathy. Take as an example the disease of proliferative diabetic retinopathy, a leading cause of blindness in the working-age population of most developed countries. It is the result of progressive damage to the network of tiny blood vessels that supply blood to the retina, and is specifically characterized by the formation of newly formed vessels in the retina [6]. This thus depends on the description of blood vessel tree structure in clinical diagnosis, and as a result, calls for proper tracing of the vessel trees from fundus images. Similar to what has happened to neuronal tracing, existing methods often fail to trace properly with the occurrence of *crossover* [7] at the junctions, as it is challenging to predict

whether the filaments contacting a junction belong to the same tree or different trees, and for the latter case, to which tree each filament belongs.

This work aims at automated tracing of filamentary structures in neuronal as well as fundus images. In particular, we focus on addressing the bottleneck crossover issue. One important observation is that local and global contextual information is crucial to resolve the crossover issue. For example, at a junction, it is very helpful to go beyond the current filament and examine the angular, morphological, and textural properties of all filaments of the junction. This information is unfortunately largely ignored by current tracers. Here we consider a two-step tracing approach that takes into account both local and global contextual information of the neuronal and the vessel network: The first step takes the raw image as input and produces a pixel-based segmentation map. After skeleton extraction, a novel *graph representation* is formed in the second step, where each filament in the skeleton map becomes a node, and a contact between two adjacent filaments could be translated to directed edges of these two nodes. Furthermore, the root nodes are naturally identified as either the DAPI tagged soma (aka cell body) in neuronal images, or the filaments touching the optic disk area in fundus images, and are further labeled with their unique IDs. The number of subgraphs to-be-found in the filamentary network thus equals the number of root nodes. This naturally gives rise to a directed graph (or interchangeably, *digraph*) representation. The tracing problem is now formulated as label propagation [8] on digraphs: The goal becomes that of propagating the subgraph labels from known root nodes to the rest of the digraph, such that the digraph is partitioned into disjoint sub-graphs, which in turn delivers tracing result of the filamentary network. This allows us to consider and make novel usage of the established matrix-forest theorem [9] studied in algebraic digraph theory [10].

The main contributions of this paper are as follows. First, our approach, and in particular the second step, offers a principled way of addressing the tracing with crossover problem. By connecting to the well-established algebraic digraph theory [9], as well as the transductive inference in machine learning [11], local and global contextual information can be considered explicitly. We expect the graph representation, and the algebraic graph theory connection can open doors to more insightful understanding of the tracing problem at hand. Complexity analysis and generalization error analysis also provide useful characterization of the proposed approach. Second, our segmentation step is carried out with graph Laplacian based regularization, which facilitates the overall graph-theoretical framework considered in our tracing system. Third, we provide an in-house neuronal microscopic image dataset dedicated to the task of neuronal tracing. The dataset together with the gold-standard manual annotations have been made publicly available [1]. We wish it can provide an option for researchers in the field to compare performance on the same ground.

[1]Downloadable at http://web.bii.a-star.edu.sg/~chengli/tracing/.

Before proceeding into the detailed account of our approach, it is useful to briefly review the related research efforts.

II. Related Work

In what follows we will provide a concise report of the related efforts in neuronal and retinal tracing. As our approach relies on label propagation on (di-)graphs, it is also of interest to discuss the related main efforts along this direction in section II-B.

A. Neuronal and Retinal Tracing

The problems of neuronal and retinal blood vessel tracing are usually studied separately. So we will begin our discussions on retinal tracing, which is followed by neuronal tracing.

Existing efforts in retinal vessel analysis can be roughly categorized into two groups, segmentation-based and tracking-based. The segmentation-based methods often use pixel classification [12]–[15] to produce a binary segmentation, where a pixel is classified into either filament or non-filament. For instance, the work of [14] utilizes a neural network learning scheme for pixel classification by combining gray-scale and moment invariants-based features, meanwhile [15] is an unsuerpervised method based on wavelet thresholding. The tracking-based methods [7], [16]–[21], on the other hand, usually start with a seed and track the intended filaments based on local intensity or texture information. The method of [20], for example, is based on ridge analysis and affinity measure based method to form spanning trees from obtained ridge points. Inspired biologically by the cortical orientation columns in primary visual cortex, [21] advocates a special Euclidean group SE(2) based retinal vessel tracking system. For a more comprehensive and dedicated survey of retinal vessel analysis, interested readers might refer to the work of Fraz et al. [22].

There have also been similar developments in tracing neurons especially from 2D neuronal images, such as [1]–[5], [23], [24]. Well-known ImageJ plugins include the NeuronJ [23] and the Simple Neurite Tracer [2], which are primarily manual tracing tools, hence are rather limited when working with neuronal images containing complex structures. The NeuriteTracer [1] is also an ImageJ plugin, which is instead an automated neuronal segmentation method but unfortunately still fails to separate each neuron from the rest when they are overlapped. Therefore it is primarily for population average information instead of achieving cell level resolution. NeuroCyto [4] and Metamorph NX [5], nonetheless, are able to automatically quantify cell level information. In NeuroCyto [4] this is achieved based on topological image analysis and filamentary foreground segmentation. Meanwhile, as a commercial product, the inner working of Metamorph NX remains unknown. Besides, a probabilistic approach is adopted in [24], which connects the set of sampled points with an optimal tree which best preserves the filamentary structure. For a comprehensive survey on the topic of neuronal tracing, we suggest the paper of Meijering [25] as well as [26]. It is worth noting that in recent years, a number of dedicated methods have been devoted to 3D neuronal tracing, including [3],

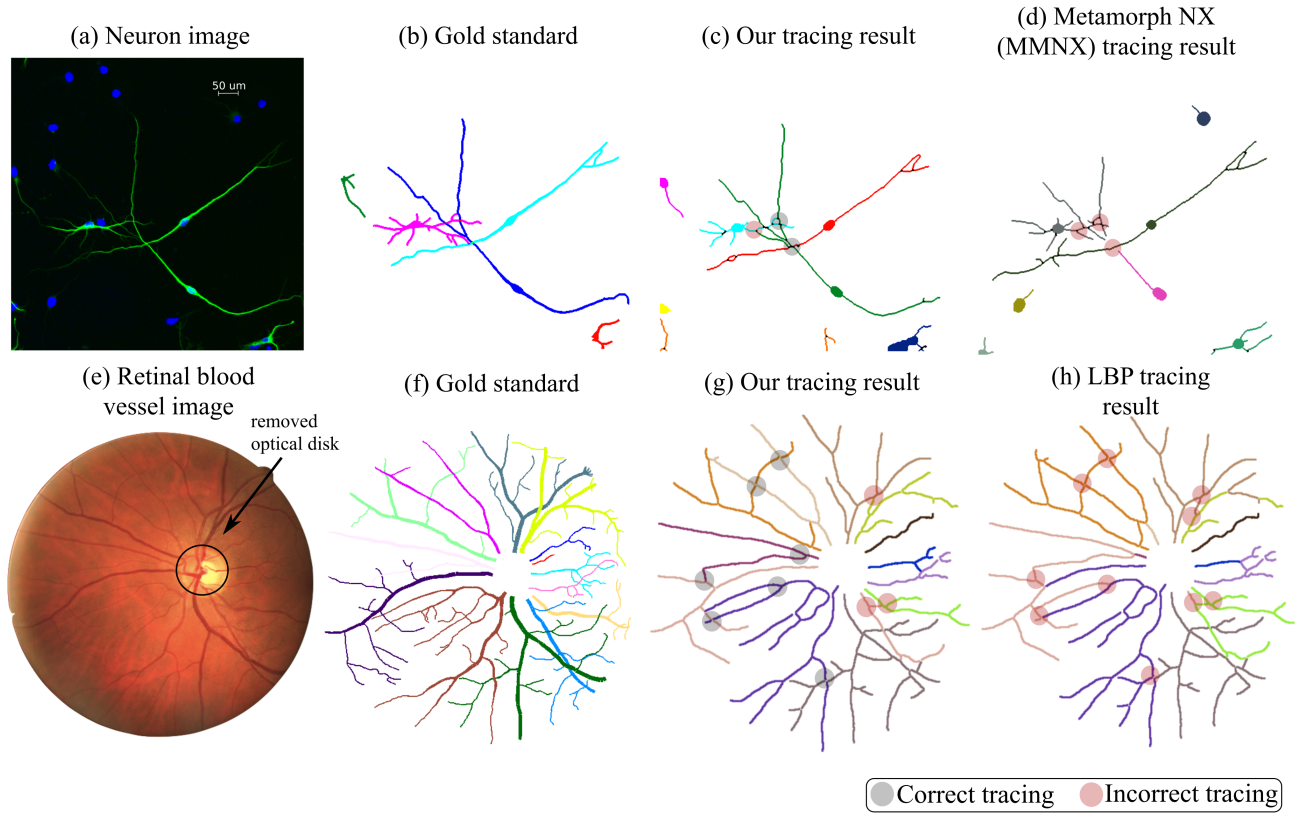

Fig. 1: Exemplar images for neuronal and retinal vessel tracing, as well as comparisons of our system vs. the state-of-the-arts. Selected correctly and incorrectly traced filaments are highlighted at junctions with grey and pink colored disks, respectively. Here gold standards are the manual annotations (i.e. reference results) that ideally should be equivalent to the ground-truths which are unfortunately impossible to obtain in practice. Best viewing in color.

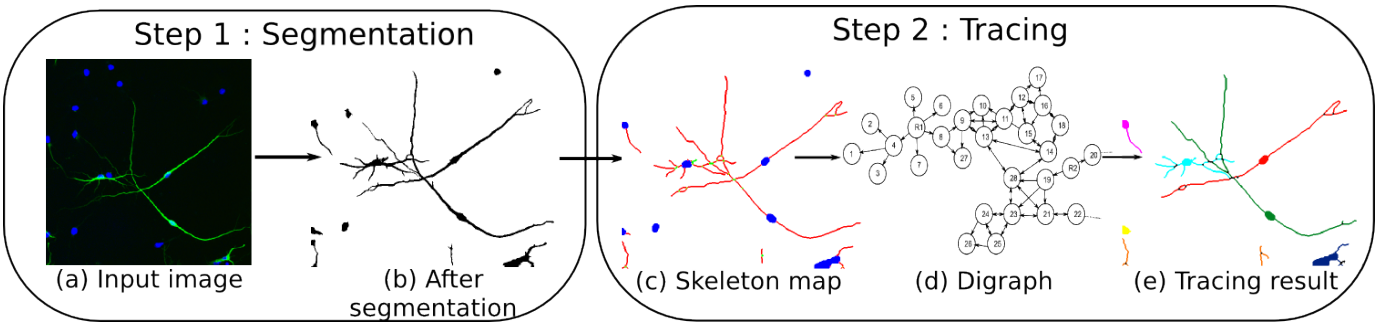

Fig. 2: Overview of our two-step tracing method.

[26]–[29], to name a few. The DIADEM challenge [30] has also been organized with dedicated datasets that focus on 3D neuronal tracing. However, since we focus on 2D images, they are beyond the scope of this paper.

There have also been efforts in considering both tracing problems under more general viewpoints. For example, a well-known local Hessian based method is developed by Frangi et al. [31] that is applicable to both situations. An interactive method is also presented in [32] using minimal math and anisotropic enhancement. Furthermore, [33] shows instead a discriminative method to learn convolutional features using gradient boosting regression technology. A review of existing developments are attempted in [34] from the general perspective of tracing vessels.

The crossover issue has been widely observed, and are discussed by existing research efforts including [20], [21], [24], [29]. Closely related work includes [35] where graphical models are considered rather than the digraph-theoretical approach proposed here. An early version of this work also appears in [36], which differs from this work in a number of aspects including the segmentation step, digraph construction protocol, generalization error analysis, as well as empirical evaluations. Moreover, both [35] and [36] only focus on the problem of retinal vessel tracing, instead of the broader view as we have considered in this paper.

B. Label Propagation on Graphs

Very few label propagation methods are specifically dedicated to digraphs, including notably [37], [38]. The most related work is that of [37], which generalizes their earlier framework of regularized risk minimization on undirected graphs [39] to digraphs by discriminatively normalizing in-links and out-links, as well as adopting the symmetrized graph Laplacian of [40]. One major difficulty in learning with digraphs lies in the asymmetric nature of the weight matrix introduced by these directed edges or links. This is often regarded as cumbersome when aligning with the key concepts developed for undirected graphs that are symmetric by nature, such as graph Laplacians. It thus leads to the widely adopted symmetrization trick in e.g. the construction of graph Laplacians [40], or co-link similarity matrices [41]. Unfortunately important information conveyed by edge directions are lost. In contrast, our approach directly works with asymmetric matrices, which is the key in preserving edge directions. In addition, the construction in both methods [37], [38] rely on *strongly connected* digraphs, that is, there always exists a directed path from any node to any other node of the graph. Since in practice the digraphs are usually not necessarily strongly connected, a teleporting random walk trick (e.g [42]) is adopted by inserting bi-directional edges between *all* node pairs with a *tiny* weight. The resulting method thus works with non-sparse matrices, instead our algorithm works with a much broader set of digraphs called *weakly connected* digraphs, i.e. for an arbitrary partitioning of the digraph node set into two subsets, there is at least an edge connecting them regardless of the edge direction. This enable our algorithm to well-preserve the sparse graph structures and edge directional information of the input, which we believe is an important advantage over existing label propagation methods on graphs (being either digraphs or undirected graphs). This hypothesis has indeed been empirically validated throughout extensive experiments where our label propagation method consistently outperforms existing label propagation methods when operating on the same ground.

III. Our Approach

The main problem involves tracing and separating filamentary structures into disjoint set of filaments, each starting from a labeled filament and forming on its own an individual object (neuron or retinal vessel tree). As illustrated in Fig. 1 as well as mentioned previously, the primary difficulty is to resolve the challenging crossover issue. Fig. 2 describes the pipeline of our tracing system consisting of *two* main steps: the *segmentation* step for segmentation and skeleton extraction, as well as the *tracing* step where the digraph is constructed from the previous step and the tracing task is cast as digraph-based label propagation using Matrix-forest theorem. In what follows we will elaborate on these two steps.

A. Segmentation Step: Penalized Likelihood Estimate with Graph Laplacian

Denote $\{1, 2, \dots K_s\}$ a set of class labels, and $\mathbf{x} = \{x_1, x_2, \dots x_{n_s}\}$ a set of n_s data points. Further, consider a set of K_s class-priors $\{p(k)\}_{k=1}^{K_s}$, and a set of class-conditional density functions $\{p(x_i|k)\}_{k=1}^{K_s}$. It is standard to formulate the Gaussian mixture model (GMM) as the following form of likelihood functions (e.g. [13])

$$p(k|x_i) = \frac{p(x_i|k)\, p(k)}{\sum_{k=1}^{K_s} p(x_i|k)\, p(k)}.$$

An interesting observation is that the above equation can be re-cast as an equivalent matrix representation

$$T = M^{-1}Q,$$

where $T = [t_{ik}]$ and $Q = [q_{ik}]$ are $n_s \times K_s$ matrices, with $t_{ik} = p(k|x_i)$ and $q_{ik} = p(x_i|k)\, p(k)$, and $M = \mathrm{diag}\left(\sum_k p(x_i|k)\, p(k)\right)$ a $n_s \times n_s$ diagonal matrix.

Consider now a graph representation of the set of n_s data points $\mathbf{x}$, with n_s vertices and a $n_s \times n_s$ symmetric weight matrix $W_s = [w_{s,ij}]$ to characterize the set of edges. Let $D_s = \mathrm{diag}(\sum_j w_{s,ij})$ be the diagonal degree matrix. The graph Laplacian $L_s = D_s - W_s$ acts as a smoothing operator on the function space on this graph. This inspires us to propose a graph Laplacian regularized GMM by solving the following augmented objective function:

$$T^* = \arg\min_{\hat{T}} \left(\|\hat{T} - M^{-1}Q\|_M^2 + \gamma\, \mathrm{tr}\left(\hat{T}^T L_s \hat{T}\right)\right),$$

where $\gamma > 0$ is a tuning parameter, $\mathrm{tr}(\cdot)$ is the matrix trace operator, and $\|\hat{T} - M^{-1}Q\|_M^2 := \mathrm{tr}((\hat{T} - M^{-1}Q)^T M(\hat{T} - M^{-1}Q))$.

Since the above objective function is a quadratic function of $\hat{T}$, there exists a unique matrix-valued solution, $T^* = [t_{ik}^*]$, which can be obtained by solving the system of linear equations

$$(M + \gamma L_s)T^* = Q,$$

where $(M + \gamma L_s)$ is a positive definite matrix. It is interesting to note that the solution to this penalized form remains a valid probability function, which can be verified straightforwardly by observing

$$\begin{aligned} T^*\mathbf{e}_{K_s} &= (I + \gamma M^{-1}L_s)^{-1} M^{-1} Q \mathbf{e}_{K_s} \\ &= (I + \gamma M^{-1}L_s)^{-1}\mathbf{e}_{n_s} = \mathbf{e}_{n_s}, \end{aligned}$$

where $\mathbf{e}_{K_s}$ and $\mathbf{e}_{n_s}$ are column vectors of K_s and n_s elements with all elements being 1, respectively, and the last equality holds since $(I + \gamma M^{-1}L_s)\mathbf{e}_{n_s} = \mathbf{e}_{n_s}$.

Finally, the class labels are obtained by row-wisely identifying the element with the maximum probability value,

$$y_i = \arg\max_k \; t_{ik}^*.$$

To illustrate the effect of the proposed penalized segmenter, Fig. 3 presents an exemplar neuronal image: Comparing to the existing GMM likelihood based segmenter as shown in the top row, it is clear that our segmenter delivers a much smoother and less noisy segmentation result, as displayed in the bottom row. Note similar results are also obtained for retinal images. Implementation detail and quantitative evaluation of the segmentation step are put off to the later section on empirical evaluations.

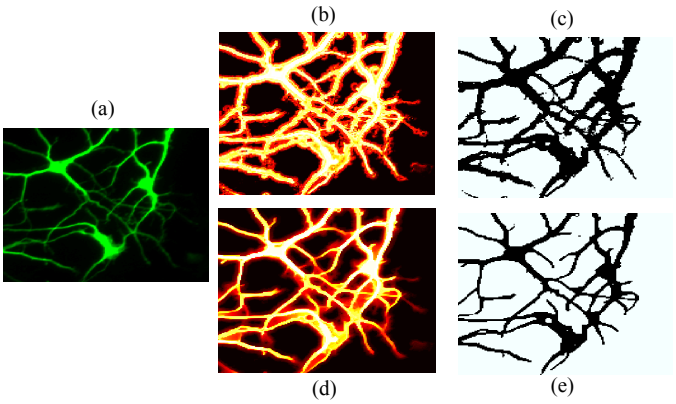

Fig. 3: Comparing segmentation results of GMM likelihood estimate vs. our penalized likelihood estimate. Given a cropped input neuronal image in (a), there are noticeable over-smoothing effects in the heatmaps of filament probability images of the GMM likelihood estimate (b) when comparing to those of the proposed penalized likelihood estimate (d). The segmentation results of the GMM likelihood estimate (c) thus contain excessive false positives when comparing to the proposed penalized likelihood estimate (e).

B. Tracing Step: Label Propagation on Digraphs by the Matrix-forest Theorem

Our tracing step starts by converting the segmentation result to its digraph representation. The tracing problem can therefore be formulated as label propagation on digraphs, which is subsequently addressed by our matrix-forest theorem algorithm.

1) Preparation: Segmentation → Digraph Representation: The segmentation result is converted to its digraph representation by two intermediate stages: First, the segmented image undergoes a thinning stage to a skeleton map; Second, a digraph is constructed using the skeleton map. Details are as follows:

Segmentation result → Skeleton map: Applying the segmentation step on an input image provides us the filamentary foreground segmentation. This is then followed by medial axis transform and postprocessing to produce a skeleton map of 1-pixel thickness. The obtained skeleton pixels can be divided into the following categories:

- *Root pixels* – For neuronal tracing, the root pixels are the DAPI-tagged nuclei pixels; For retinal vessel tracing, the root pixels are the pixels touching the boundary of the optical disk.
- *Body pixels* – Pixels with two neighbours.
- *Junction pixels* – Pixels with three or more neighbours.
- *Terminal pixels* – Pixels with only one neighbour and are not the root pixels.

Furthermore, a *filament* in our context refers to a set of connected body/root/terminal pixels, and a *junction* is the set of connected junction pixels, as illustrated in Fig. 4 in red and green, respectively. In this paper, a filament is also interchangeably referred to as a filamentary segment or simply a segment.

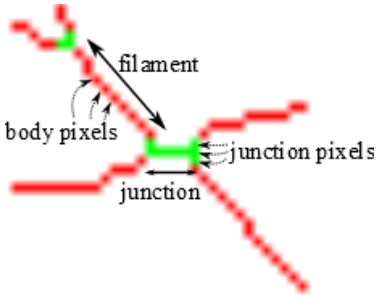

Fig. 4: Example of filament (in red) and junction (in green).

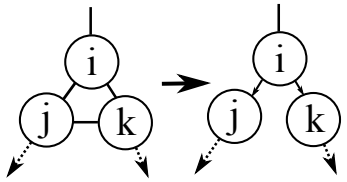

Fig. 5: Synthetic example of rule 3.

Constructing the digraph: Now we start to construct the digraph $G = (V, E, W)$, with V, E, W being the node set, edge set, and weight matrix, respectively. Each node in the graph corresponds to a *filament*; There is an edge between two nodes if these two filaments are directly connected in the skeleton map via a *junction*; A root node is a filament containing root pixels, while a terminal node is a filament containing terminal pixels. This gives rise to essentially an undirect graph $G_U = (V_U, E_U)$, from which our digraph representation G can be obtained by the following intuitive rules:

1. A root node possesses only outgoing edges, while a terminal node contains only incoming edges.
2. If a k-clique ($k \geq 3$) of G_U has more than $k-2$ terminal nodes, then the edge among these terminal nodes will be removed. For example, if a 4-clique has three terminal nodes, then the edge between those three nodes will be removed.
3. In a 3-clique of nodes i, j and k, if the nodes j and k has outgoing connections to rest of the graph other than i, then (i, j) and (i, k) edge will be $i \to j$ and $i \to k$ respectively. The edge (j, k) will be removed.
4. All remaining undirected edges are converted to a pair of directed edges with opposite directions.

Fig. 6 illustrates an example of executing this digraph construction stage. After filamentary foregrounds (b) are obtained from input image (a), its skeleton map is obtained (c), and for a zoomed-in crop of the filamentary foregrounds, its undirected graph G_U and finally the digraph representation G is constructed, where the root nodes are denoted as R1 and R2, respectively. The clique of nodes $\{1, 2, 3, 4\}$ in G_U of (e) is converted to its corresponding directed subgraph in (f) marked within a dotted boundary, obtained by applying rule 2. A synthetic example of rule 3 is shown in Fig. 5. Note the introduction of the intermediate undirected graph G_U here is to make the process intuitive and easy to understand. In practice, the digraph representation G as in Fig. 6(f) can be directly obtained using the skeleton map as in Fig. 6(d).

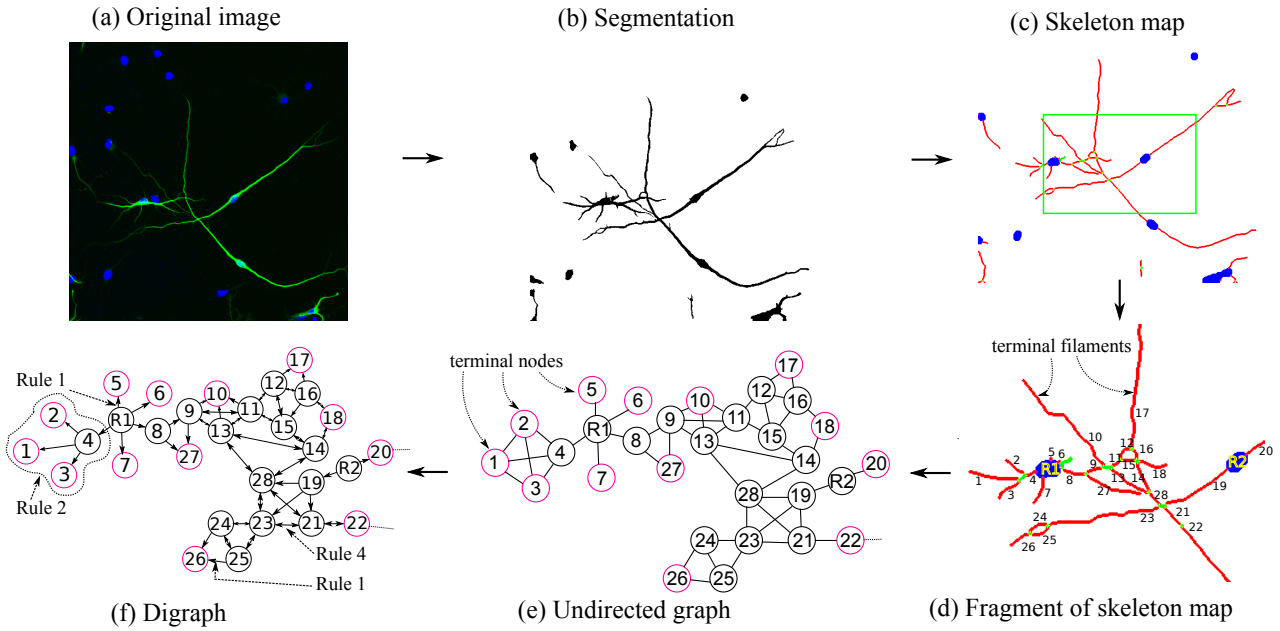

Fig. 6: Running example of digraph preparation process in the tracing step. (a) Original image. (b) Segmentation result. (c) Skeleton map, with (d) being a fragment inside the green box. (e) Undirected graph G_U. (f) Digraph G.

The weight matrix $W = [w_{ij}]$ is computed as follows: w_{ij} is set to 0 if the i-th and j-th filaments involved are not contacting (i.e. immediately connected via a junction). Due to the edge sparsity nature of our digraph representation, majority of the matrix elements are therefore 0. A weight matrix element may attain a positive values only if the two filaments are contacting, which can be obtained based on the angle θ_{ij} between these two filaments as

$$w_{ij} = \exp\Big(-f\big(\theta_{ij}\big)\Big).$$

Note θ_{ij} is in the range of [0, π], and the induced energy function takes the form

$$f(\theta_{ij}) = \begin{cases} -\frac{\sin(\theta_{ij})}{k} & \text{if } \theta_{ij} \in [0, \theta_c) \\ -\frac{\sin(\theta_c)}{k} & \text{if } \theta_{ij} \in [\theta_c, \arccos(\frac{-\sin\theta_c}{k^2})) \\ k\,\cos(\theta_{ij}) & \text{if } \theta_{ij} \in [\arccos(\frac{-\sin\theta_c}{k^2}), \pi]. \end{cases}$$

It is worth noting that each element in the weight matrix is designed to capture the following intuition: For two consecutive filaments i and j belonging to the same vessel tree or neuron, they should not bend too much. In other words, the angle between these two filaments, θ_{ij}, tends to be close to π. In our context it amounts to endow its corresponding w_{ij} a higher weight value, and the other way around if there is an abrupt bend (i.e. a tiny weight value if θ_{ij} is an acute angle). The meaning of the two tuning constants, θ_c and k, are illustrated in Fig. 7 for both the energy function f and the weight w_{ij}: θ_c marks the turning point in the function f, while k brings in the weight difference between highest and the lowest values. Throughout empirical experiments, θ_c and k are fixed to $\pi/7$ and 5, respectively.

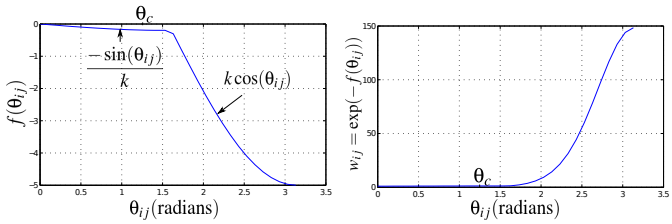

Fig. 7: Plot for the energy function $f(\theta_{ij})$ and the corresponding weight w_{ij}, both as functions of θ_{ij}, the angle between the i-th and j-th filaments.

2) Label Propagation by the Matrix-forest Theorem on digraphs: Given this digraph G, denote the out-degree of each node v_i as $d_i^+ = \sum_{j=1}^n w_{ij}$. Further, define the out-degree matrix as $D = \mathrm{diag}(d_1^+, \cdots, d_n^+)$, or in matrix forms, $D = \mathrm{diag}(W\mathbf{e}_n)$. Denote the digraph Laplacian $L = D - W$. A row-stochastic transition probability matrix $P = [p_{ij}]$ can be constructed as $p_{ij} = w_{ij}/d_i^+$, or equivalently as $P = D^{-1}W$. Note that undirected graphs can be regarded as special digraphs characterized algebraically by their symmetric weight matrix W, i.e. the symmetric pair w_{ij} & w_{ji} correspond to bi-directional edges with equal weights. We focus here on a transductive inference scenario where labels from the set of few labeled nodes V_l are to be propagated to the rest unlabeled nodes V_u, with $V = V_l \cup V_u$. These labels are of multiclass nature, where each class corresponds to an individual neuron or vessel tree. To simplify the notations we assume V_l contains the first l nodes, $V_l = \{v_1, \ldots, v_l\}$. To accommodate label information, define a label matrix $Y = [y_{ik}]$ of size $n \times K$ (assuming there are K class labels available), with each entry y_{ik} containing 1 if node i belongs to V_l and is labeled with class k, and 0 otherwise. Also define the length n ground-truth (i.e. gold-standard) label vector $\mathbf{y}$ that contains two disjoint parts $\mathbf{y}_l$ and $\mathbf{y}_u$: $\mathbf{y}_l$ is the input label vector of length l over

Algorithm 1 Label Propagation by Matrix-Forest Theorem of Digraphs (MFTD)

Input: A digraph $G = (V, E, W)$, label information Y, $\mathbf{y}_l$, and $\alpha \in (0, 1)$.
Output: $\mathbf{y}_u^*$
1. Compute the out-degree matrix D.
2. Compute the affinity matrix by $A = SY$ and (1) (or (2)).
3. Predict $\mathbf{y}_u^*$: Compute the i-th entry by (4), for any unlabeled node $v_i \in V_u$.

the set of labeled nodes, with each entry y_i for the input class assignment of node $v_i \in V_l$; $\mathbf{y}_u$ is the hold-out ground-truth label for the unlabeled nodes, i.e. a vector of length $n - l$. Similarly, define the initial label vector $\hat{\mathbf{y}}$ containing also two parts, $\hat{\mathbf{y}}_l := \mathbf{y}_l$ and $\hat{\mathbf{y}}_u = \mathbf{0}$, where $\mathbf{0}$ is an all-zero vector of length $n - l$. Define the prediction vector $\mathbf{y}^*$ with also two parts $\mathbf{y}_l^* := \mathbf{y}_l$, as well as $\mathbf{y}_u^*$ of length $n - l$, containing the prediction results, where each y_i^* denotes the predicted class assignment for a node $v_i \in V_u$.

The proposed label propagation algorithm (shown in Algorithm 1 and referred to as MFTD) is derived based on matrix-forest theorem [9] of algebraic digraph theory [10] as follows. The forest matrix is defined as

$$S_1 = (I + \alpha L)^{-1}, \tag{1}$$

a normalized forest matrix where each (i, j)-th entry denotes the number of spanning trees rooted at node i that also include the j-th node, as in Theorem 4 of [9]. In a way it can be viewed as a generalization of the celebrated matrix-tree theorem (e.g. [43]) from undirected graphs to digraphs. Let w_{max} denote the entry in W containing the strongest signal, i.e. $w_{\max} = \max_{i,j} |w_{ij}|$. By varying the preprocessing schemes of normalizing W, we have two variants: $\mathrm{MFTD_a}$ starts with a preprocessing effort to normalize W, $W \leftarrow \frac{W}{W_{\max}}$; $\mathrm{MFTD_b}$ considers a different normalization of W as $W \leftarrow D^{-1}W$ instead.

It can be shown that under normalization scheme of $W \leftarrow D^{-1}W$, the forest matrix becomes $S_1 = (1 - \tau)(I - \tau P)^{-1}$, with $\tau = \frac{\alpha}{1+\alpha}$. Further, let $\tilde{L} := \lim_{\alpha\to\infty}(I + \alpha L)^{-1}$, which is a matrix of normalized rooted spanning forests. Both S_1 and $\tilde{L}$ has a number of interesting properties [44]: Each entry of both matrices is non-negative, and both matrices are row-stochastic; $\tilde{L}$ resides in the null space of digraph Laplacian L, as $L\tilde{L} = \tilde{L}L = 0$; $\mathrm{rank}(L) = n - \mathrm{rank}(\tilde{L})$; $L + \beta\tilde{L}$ is non-singular for any $\beta > 0$, and is the "complementary perturbation of L" [45]. Indeed, this brings forward the second forest matrix,

$$S_2 = (L + \beta\tilde{L})^{-1}, \tag{2}$$

which is also termed the matrix of dense forest in [9]. When the same preprocessing schemes of normalizing W are applied to the forest matrix S_2, two additional variants are similarly obtained, and are denoted as $\mathrm{MFTD_c}$ & $\mathrm{MFTD_d}$, respectively.

One can interpret the (i, j)-th entry s_{ij} of the forest matrix S (being either $S := S_1$ or $S := S_2$) as quantifying the accessibility of a particle from a node v_i to visit node v_j along the digraph structure. This provides a notion of affinity from state i to j. The intuition is, if a state j is close to the initial state i in terms of digraph structure, it will be visited by the particle more often than if it is far away from i, i.e., we visit our close relatives more often than our distant ones. Now define the affinity matrix as

$$A = SY, \tag{3}$$

a matrix of size $n \times K$, with each entry a_{ik} being associated with an affinity score of state i belonging to class k. To infer $\mathbf{y}_u^*$ of the unlabeled states V_u, our algorithm predicts each entry's class assignment by identifying a label with the largest affinity score,

$$\mathrm{y}_i^* = \arg\max_k a_{ik}, \qquad \forall v_i \in V_u. \tag{4}$$

Now we have all the ingredients ready to present our digraph-theoretical label propagation algorithm, MFTD, as described in Algorithm 1. With a slight abuse of notation, MFTD is also used to refer to our entire tracing system.

Computational complexity As the forest matrix S (being either $S := S_1$ or $S := S_2$) is invertible, denote its inverse as $E := S^{-1}$. Usually it is computationally more efficient to evaluate E than S, since $E = I + \alpha L$ or $E = L + \beta\tilde{L}$. Let us investigate the complexity of Algorithm 1, which is clearly dominated by the cost of computing the affinity matrix A in (3). In fact, it can be accomplished by instead solving the linear system of

$$EA = Y,$$

thus eliminating the need to explicitly compute any matrix inverse. For a general dense matrix E, the computational time is $O(n^3 + n^2K)$. This is about the same complexity of [37], one of our main competing methods. Fortunately, E (as well as S) is usually a sparse matrix in our context, which can be exploited to reduce the computational time. There are many efficient solvers for large sparse linear systems, including both direct [46], [47] and iterative methods [48], [49]. In our implementation, we adopt the direct solver UMFPACK [46] which exists as a built-in routine (for LU, backslash, and forward slash functions) in MATLAB. The specific complexity depends on the size (n), the number of non-zero entries and the sparsity pattern of E, which remains a challenging task to provide an explicit complexity measure dedicated to our context.

PAC-Bayesian label propagation bound A PAC-Bayesian error bound is provided here for our MFTD algorithm (both (1) and (2)). This bound is an adaptation of the PAC-Bayesian bound for general transductive learning developed in [50], which improves the PAC-Bayesian bound of [51].

In the transductive setting, a learning method is given the full sample $V = V_l \cup V_u$ and labels of the labelled data (i.e., y_i for $v_i \in V_l$), and learns a classifier $h : V \to \mathcal{Y} := \{1, 2, \cdots, K\}$ that correctly classifies the unlabelled data in V_u. The hypothesis space of classifiers determined by the learning method is denoted as $\mathcal{H}$. For any classifier $h \in \mathcal{H}$, we define the *test error* as $\mathcal{L}_{l,n}(h) := \frac{1}{n-l}\sum_{i=l+1}^{n} \ell(h(v_i), \mathrm{y}_i)$

with respect to the $0/1$ loss function ℓ satisfying $\ell(h(v_i), \mathrm{y}_i) = 1$ if $h(v_i) \neq \mathrm{y}_i$ and $\ell(h(v_i), \mathrm{y}_i) = 0$ otherwise. Similarly, we define the *empirical error* of h as $\hat{\mathcal{L}}_{l,n}(h) := \frac{1}{l}\sum_{i=1}^{l} \ell(h_i, \mathrm{y}_i)$, the *full sample error* of h as $\mathcal{L}_n(h) := \frac{1}{n}\sum_{i=1}^{n} \ell(h_i, \mathrm{y}_i)$. Therefore, we have

$$\mathcal{L}_n(h) = \frac{1}{n}\left(l\hat{\mathcal{L}}_{l,n}(h) + (n-l)\mathcal{L}_{l,n}(h)\right). \tag{5}$$

In the error bound presented in the next theorem, we show that the test error can be upper bounded by the empirical error plus some complexity term.

Theorem 1: For any deterministic classifier h determined by our Algorithm 1 and any $\delta \in (0,1)$, with probability at lest $1-\delta$ over random draws of V_l from V, the following bound holds

$$\mathcal{L}_{l,n}(h) \leq \hat{\mathcal{L}}_{l,n}(h) + \sqrt{\frac{1}{2(1-\frac{l}{n})}\left(\ln\frac{n}{l} + \frac{1}{l}\ln\frac{C(l,n)}{\delta} + \ln(Ke)\right)}, \tag{6}$$

where $C(l,n) = (\sqrt{2}\ln l + 8)\sqrt{l(1-\frac{l}{n})}$.

Proof: Our proof is based on the PAC-Bayesian bound for *Gibbs classifier* $h_{\mathbf{q}}$ presented in Corollary 7 of [50]. For a given distribution $\mathbf{q}$ on $\mathcal{H}$ and a sample v, $h_{\mathbf{q}}$ randomly draws a classifier h according distribution $\mathbf{q}$ and classifies v as $h(v)$. The test error of $h_{\mathbf{q}}$ is defined as $\mathcal{L}_{l,n}(h_{\mathbf{q}}) := \frac{1}{n-l}\sum_{i=l+1}^{n}\mathbb{E}_{h\sim\mathbf{q}}\ell(h(v_i), \mathrm{y}_i)$. The empirical error $\hat{\mathcal{L}}_{l,n}(h_{\mathbf{q}})$ and full sample error $\mathcal{L}_n(h_{\mathbf{q}})$ are similarly defined. An adaptation of Corollary 7 of [50] leads to the following result: For any *prior distribution* $\mathbf{p}$ on $\mathcal{H}$ and any $\delta \in (0,1)$, with probability at lest $1-\delta$ over random draws of V_l from V, we have

$$\mathcal{L}_n(h_{\mathbf{q}}) \leq \hat{\mathcal{L}}_{l,n}(h_{\mathbf{q}}) + \sqrt{\frac{1-\frac{l}{n}}{2l}\left(KL(\mathbf{q}||\mathbf{p}) + \ln\frac{C(l,n)}{\delta}\right)}, \tag{7}$$

where $KL(\mathbf{q}||\mathbf{p})$ is the Kullback-Leibler divergence between two distributions and $C(l,n)$ is defined as in (6).

For any deterministic classifier $h \in \mathcal{H}$, we can derive an error bound for h by choosing the posterior distribution $\mathbf{q}$ as the one which assigns probability 1 to h. In this way, all errors of $h_{\mathbf{q}}$ reduce to the corresponding errors of h and the KL-divergence term $KL(\mathbf{q}||\mathbf{p})$ reduces to $\ln(1/\mathbf{p}(h))$, which implies that

$$\mathcal{L}_n(h) \leq \hat{\mathcal{L}}_{l,n}(h) + \sqrt{\frac{1-\frac{l}{n}}{2l}\left(\ln\frac{1}{\mathbf{p}(h)} + \ln\frac{C(l,n)}{\delta}\right)}. \tag{8}$$

Now, we propose a technique for selecting a prior distribution $\mathbf{p}$ over $\mathcal{H}$ based on the full sample V. It is noted that Algorithm 1 is deterministic and generates only one classifier h once V_l and the corresponding labels are given. Since there are $\binom{n}{l}$ ways to draw V_l from V and there are K possible labels for each sample in V_l, it implies that $\mathcal{H}$ has $\binom{n}{l}K^l$ hypotheses. That is, $|\mathcal{H}| = \binom{n}{l}K^l$. We then define the prior distribution $\mathbf{p}$ to be an uniform distribution over $\mathcal{H}$, or equivalently, $\mathbf{p}(h) = \frac{1}{\binom{n}{l}K^l}$ for any $h \in \mathcal{H}$. Substituting $\mathbf{p}(h)$ into (8) and using inequality $\binom{n}{l} \leq (en/l)^l$ leads to

$$\mathcal{L}_n(h) \leq \hat{\mathcal{L}}_{l,n}(h) + \sqrt{\frac{1-\frac{l}{n}}{2}\left(\ln\frac{n}{l} + \frac{1}{l}\ln\frac{C(l,n)}{\delta} + \ln(Ke)\right)}. \tag{9}$$

In addition, we have for any constant c that $\mathcal{L}_n(h) \leq \hat{\mathcal{L}}_{l,n}(h) + c$ is equivalent to $\mathcal{L}_{l,n}(h) \leq \hat{\mathcal{L}}_{l,n}(h) + \frac{1}{1-\frac{l}{n}}c$, which together with inequality (9) leads to the desired result. ■

Notice that when $\beta_l \leq l/n \leq \beta_u$ for any constant β_l, β_u satisfying $0 < \beta_l \leq \beta_u < 1$ (e.g., $l/n = \beta_0$ the number of labelled sample is a constant proportion of the number of full sample), the complexity term (i.e., the second term on the right-hand side of (6)) converges to $\sqrt{\frac{\ln(\frac{Ke}{\beta_l})}{2(1-\beta_u)}}$ as $n \to \infty$, which means that in such a case the test error $\mathcal{L}_{l,n}(h)$ will not excess the empirical error $\hat{\mathcal{L}}_{l,n}(h)$ by a constant value.

IV. Empirical Experiments

In this section, two types of datasets are used for empirical evaluations, which includes in particular our in-house dataset of neuronal images. We further carry out a series of experiments to evaluate and analyze the performance of the competing methods.

A. The Datasets

The following two sets of datasets are employed throughout our experiments on tracing with neuronal and retinal images.

Our In-house Dataset of Microscopic Neuronal Images To facilitate the analysis of neuronal tracing systems, we make available our annotated neuronal dataset[2], wishing this can provide an option for researchers in the field to compare performance on the same ground. While our dataset focuses on mouse embryonic neural stem cells, we would like to point out that we intentionally retain a variety of cellular morphologies, such that the images in our dataset could be sufficiently representative, and the tracing performance is expected to be similar when carried out on other neuronal datasets. Our data acquisition follows the following protocol: First the cells are grown in-vitro into neurospheres. Single cells are then collected from these neurospheres and seeded onto PLL/laminin (with $10\mu g/ml$ concentration, from Invitrogen Inc. [52]) coated 13 mm coverslips at 1×10^4 cells/coverslip, and cultured in differentiation medium for 7 days. Then the cells are stained with anti-βIII tubulin (with 1:200 concentration, from Sigma-Aldrich Inc. [53]) with secondary antibodies Alexa Fluor (AF)-488 goat anti-mouse, while the cell nuclei are counter-stained with DAPI. Furthermore, two microscopic imaging systems are used: The first is an Olympus Fluoview FV1000 laser scanning confocal microscope with photomultiplier tube detector (PMT), while the second is an Olympus lx-83 live-cell inverted with Coolsnap HQ2 camera. Both are with a 12-bit A/D converter, and with a magnification

[2]Our annotated neuronal dataset as well as performance evaluation code are publicly available at http://web.bii.a-star.edu.sg/~chengli/tracing/.

of 20x. Note to evaluate the robustness of the tracers, two image resolutions are used in NeuB2: For a subset of 30 image, the resolution is set to 0.333 micrometer per pixel, while for the rest 68 images, it is set to 0.994 micrometer per pixel. In practice, two batches of neurons are cultured, with *batch 1* of 112 images acquired with the first microscope, and *batch 2* of 98 images obtained using the second microscope. We subsequently refer to these two batches as NeuB1 and NeuB2, respectively. It is worth noting that images of the second batch are more challenging, with more noticeable background noises and more blurred foregrounds. It nevertheless contributes to our dataset with added variability, which we feel could be very helpful to examine the reliability of the tracing systems. These images are further annotated manually with an annotation tool, Neuromantic [54], to produce the final manual annotations as gold standard. For both batches, one third of the images are reserved for training, while the annotations of the rest two third images are retained for performance evaluation purpose. This amounts to 675 and 543 neurons for NeuB1 and NeuB2, respectively. In addition, as displayed in the top-left panel of Fig. 1, the numerous blue-colored DAPI stained cell bodies outnumber the neurons presented in the image. This is due to the existence of non-neuronal supporting cells (i.e. glial cells) such as astrocyte and oligodendrocyte cells, whose cell bodies are also DAPI stained but containing no neurites (which are green-colored). To address this issue, a simple yet effective image processing strategy is adopted in our tracing system to obtain the neuronal cell bodies: The potential cell bodies from DAPI channel (i.e. blue channel) are obtained via fixed thresholding, while the counterparts in the tubulin stained green channel are retrieved by performing morphological operations with structured element on the segmented foregrounds. The final neuronal cell bodies are then detected by taking logical and operation on two separate resultant images, and by morphological postprocessing operation for possible hole-fillings.

Two Fundus Image Datasets: DRIVE and STARE For retinal blood vessel tracing, the publicly available testbeds of DRIVE [55] and STARE [56] datasets are used. The DRIVE and STARE datasets contain 40 and 20 color retinal fundus images, respectively. Both datasets have their own fixed partitions of train and testing subsets, each contains 20 and 10 images, respectively.

B. the Segmentation Step: Implementation Details and Evaluations

During segmentation step, the set of data points, $\mathbf{x} = \{x_1, x_2, \dots x_n\}$, contains features of all the pixels in training images. The number of class categories is naturally set to $K_s = 2$, as the focus here is to segment out the filamentary structures as foreground, with the rest pixels being background. More specifically, each data point $x_i \in \mathbb{R}^5$ containing five features: The first feature is the normalized image intensity, and the next four are normalized maximum Gabor filter responses among 18 directions in $[0, \pi]$ for 4 different scales. Our GMM likelihood model follows that of [13]. Throughout our experiments, a mixture of 15 Gaussian components is employed which is empirically to lead to good performance. The set of points (i.e. pixels) n_s is set to 10^6, obtained by randomly selecting pixels from training images. The Laplacian is obtained from an 8-neighbor grid graph following the image structure, where each image pixel corresponds to a node, and the weight matrix is constructed by $w_{s,ij} = w_{s,ji} = \exp(-\beta_s \|x_i - x_j\|^2)$ for each $(i,j)-$th element with $\beta = 5000$. Furthermore, the segmentation related parameter is set as $\gamma = 1$. Performance evaluation is reported using the average F-1 score over the test images. Table I provides an comparison of our segmentation results to the GMM-based method (referred to as Soares) [13] as well as FRVD [15], OOF [57], Frangi [31], four state-of-the-art methods for retinal blood vessel segmentation. The method of Soares [13] is based on Gabor wavelet and GMM models, and is the closest to our segmentation module. FRVD [15] utilizes a wavelet-based thresholding and edge refinement scheme, meanwhile, OOF or optimally oriented flux method is widely known for its ability to enhance and detect irregular filamentary structures and usually outperforms the traditional Hessian based methods. Frangi [31] is a local Hessian based method. The results of these comparison methods are reported by running their original source codes with parameters tuned for best performance. Empirically our segmentation module outperforms the comparison methods on both datasets of neurons and retinal vessels. This we believe could be due to the introduction of graph Laplacian based regularization which helps noticeably in providing smoother and cleaner segmentation results, as is visually illustrated in Fig. 3.

	Ours	**Soares [13]**	**FRVD [15]**	**OOF [57]**	**Frangi [31]**
NeuB1	0.76	0.74	0.69	0.51	0.59
NeuB2	0.83	0.79	0.66	0.61	0.76
DRIVE	0.78	0.77	0.71	0.73	0.56
STARE	0.77	0.76	0.64	0.69	0.57

TABLE I: Comparison of segmentation performance w.r.t. F1 score. See text for details.

C. the Overall Tracing System: Evaluation Details and Analysis

Parameters & Performance Evaluation Criteria The tracing-related internal parameters in our system are set to $\alpha = 10$ for MFTD$_a$ and MFTD$_b$, $\beta = 10^{-4}$ for MFTD$_c$ and $\beta = 1$ for MFTD$_d$. Performance evaluation is carried out using both the widely-used DIADEM score [30] and the relatively recent Netmets score [58]. DIADEM score compares the similarity between a ground truth filamentary object and the corresponding tracing result, and returns a score within $[0, 1]$, where 0 indicates a completely mismatch and 1 means a perfect match. Note the x-y threshold of the DIADEM score is fixed to 30 pixels throughout our experiments. The Netmets score, on the other hand, provides matching score based on geometric and connectivity related information, which is represented as both false positive and false negative rates. Unlike in DIADEM score, in Netmets lower value is better. In particular, in terms of connectivity false positive and false negative rates, Netmets score penalizes the extra or missing

junctions *twice* by considering extra or missing junctions and related filaments separately, which suggests the error rates would often be much higher than that of e.g. the DIADEM score where only *one* round of extra or missing junctions are considered. The DIADEM score has been the primary choice for performance evaluation of the tracing systems during empirical evaluations, while the Netmets score is also used in most of the occasions to provide a complementary source of information.

Four variants of the proposed system are considered in experiments, which are
(1) MFTD$_a$: $S := S_1$ from (1) with $W \leftarrow \frac{W}{W_{\max}}$.
(2) MFTD$_b$: $S := S_1$ from (1) with $W \leftarrow D^{-1}W$.
(3) MFTD$_c$: $S := S_2$ from (2) with $W \leftarrow \frac{W}{W_{\max}}$.
(4) MFTD$_d$: $S := S_2$ from (2) with $W \leftarrow D^{-1}W$.
As to be shown later, throughout empirical evaluations MFTD$_a$ is observed to consistently deliver the best results among the four, so by default we would also refer to our tracing system as MFTD$_a$, when the context is clear. The superior performance of MFTD$_a$ may contribute to its relative insensitive to local variations by dividing by the largest element of the row; In comparison, the denominator of each row in MFTD$_b$ is the row sum, which is more susceptible to local variations in each row as being e.g. either evenly distributed or few dominating signals mingled with many weak signals. In term of comparison methods, a suite of state-of-the-art commercial as well as related academic tracers are considered, including
(1) Label propagation method based on digraphs, including the Symmetrized Graph Laplacian (SGL) [37], and the Zero-mode Free Laplacian (ZFL) [38].
(2) Label propagation method based on undirected graphs: the Learning with local and global consistency method (LLGC) [39].
(3) Label propagation by inference on graphical Models [59] including Mean Field (MF), Loopy Belief Propagation (LBP) [35], and Tree Re-Weighted belief propagation (TRW) [59].
(4) State-of-the-art tracing systems including the commercially available neurite tracer module of Metamorph NX (MMNX) [5], as well as the academic tracing system NeuroCyto [4].
Note that SGL, ZFL, LLGC, LBP, MF, and TRW tracers are essentially label propagation methods already developed from machine learning community. They are implemented in a plug-in manner to replace the digraph algorithm in our tracing system, while the rest components of our system remain the same as those in our MFTD tracer. For methods operate only on undirected graphs (LLGC, LBP, MF, and TRW), the undirected graph G_u are obtained as discussed in Fig.6(e), and similarly, the rest of our system is still used as is. On the other hand, MMNX and NeuroCyto are self-contained neuronal tracing systems, so they are operated completely on their own. During the experiments, the proposed four variants and the comparing methods are all carried out in automated manner.

Effect of Varying α or β We first study the influence of the internal parameter (α or β) to our four MFTD tracing variants. As presented in Fig. 8, our tracing system MFTD$_a$ performs relatively robust when the internal parameter is varied within a rather broad range between 1 to 10^4. At the same time, the other MFTD tracing variants are relatively sensitive to the changes of internal parameters.

Effect of Training and Testing on Same or Different Batches From system robustness viewpoint, one might be curious about the impact of performing training and testing on slightly different sets of examples. Table II presents three such scenarios, namely NeuTr1Te1, NeuTr1Te2, NeuTr2Te2. Take NeuTr1Te2 for example, it refers to the scenario of performing training on the training neuronal images from batch 1 (i.e. NeuB1), while evaluation is carried out on the testing images from batch 2 (i.e. NeuB2). Not surprisingly, for all four variants of our tracing systems, the best results are always obtained from NeuTr1Te1, since images in NeuB1 are with less background noises and the foreground filamentary structures are less blurring. On the other hand, NeuTr2Te2 seems to be the most challenging scenario for all four competing methods. It is interesting to observe that the results on NeuTr2Te1 is quite good, almost on par with those on NeuTr1Te1. This we believe should be attributed to the relatively clean contents which make this batch much easier to be dealt with. For example, the Penalized likelihood model learned from the noisier NeuB2 images can still perform well on these less noisy NeuB1 images, although both batches are obtained from different microscopy systems. Unfortunately, it does not produce much meaningful results at all when training on NeuB1 and testing on NeuB2. This is not surprising, since it is widely known that a statistical model learned on a clean (thus easy) dataset usually performs poorly on unseen data that are with noticeably more noises (thus more difficult).

	MFTD$_a$	**MFTD$_b$**	**MFTD$_c$**	**MFTD$_d$**
NeuTr1Te1	0.52	0.51	0.51	0.41
NeuTr2Te1	0.51	0.49	0.50	0.38
NeuTr2Te2	0.26	0.24	0.25	0.19

TABLE II: DIADEM scores on the effect of training and testing on same or different batches.

Evaluations on Neuronal and Retinal Datasets
We further evaluate the performance of the proposed MFTD tracing variants as well as those competing systems on the four datasets mentioned above. As these datasets cover a range of filamentary structures with diverse morphologies, textures, and network complexities, collectively they form a reasonable testbed to evaluate performance of tracing systems on these representative scenarios.

Table III presents the DIADEM scores of the competing systems on both neuronal and retinal datasets. Overall graph-theoretical methods (including the three variants of our tracing systems except for MFTD$_d$, as well as SGL, ZFL, and LLGC) delivers the best results, followed by the graphical model methods (MF, LBP, and TRW). Note these methods all share the same segmentation step as well as the preparation process of converting the skeleton maps to undirected graphs or digraphs. On neuronal datasets, MMNX performs relatively on par with the graphical model methods, while NeuroCyto

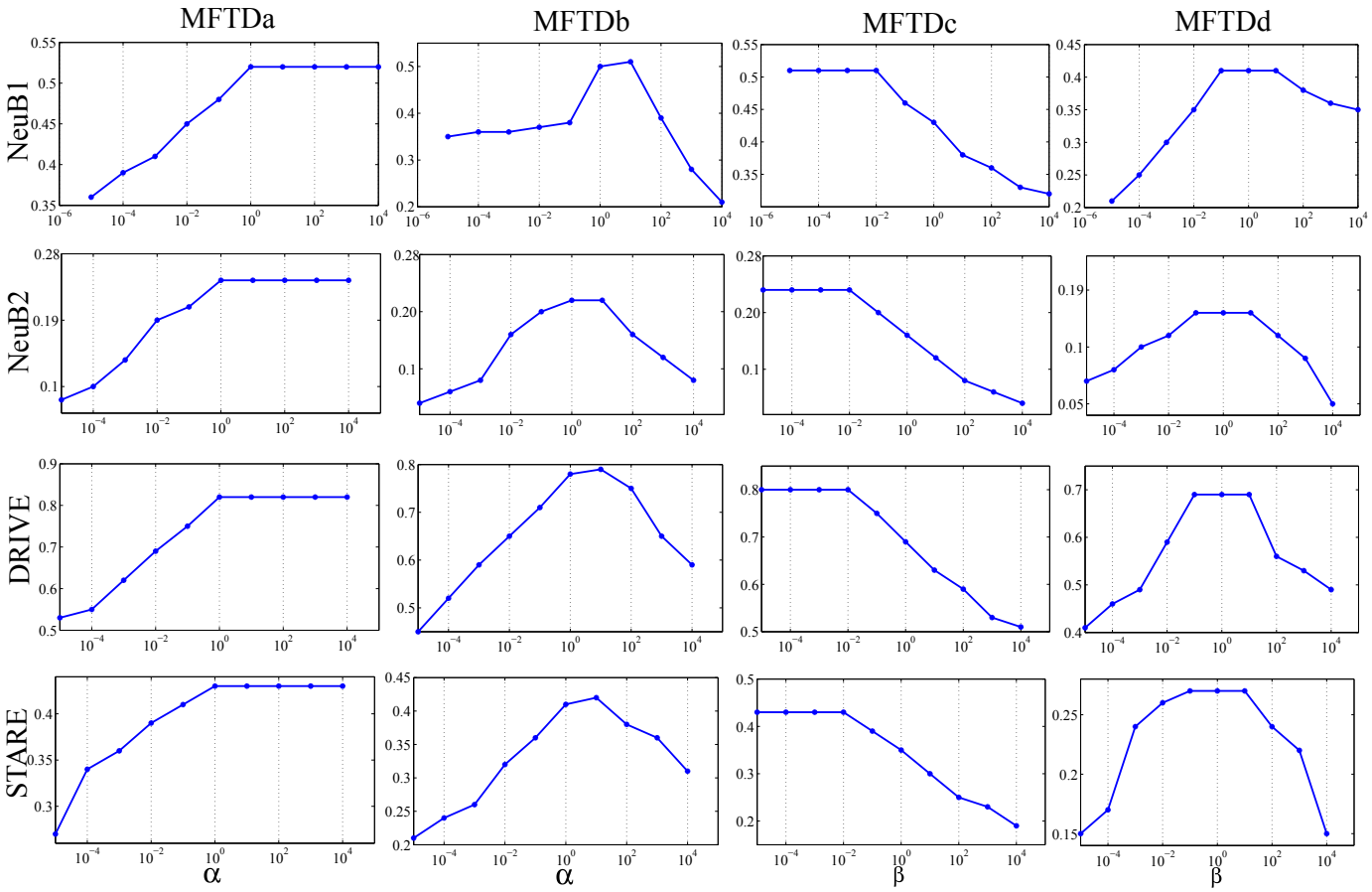

Fig. 8: Effect of varying the internal parameters α or β of our four MFTD tracing variants. From top to bottom, each row presents one of the four datasets, namely NeuB1, NeuB2, DRIVE, and STARE; From left to right, each column is for one of the four MFTD tracing variants. In each panel, α or β are presented in x-axis, while y-axis shows the DIADEM score performance.

	MFTDa	**MFTDb**	**MFTDc**	**MFTDd**	**SGL**	**ZFL**	**LLGC**	**MF**	**TRW**	**LBP**	**MMNX**	**NeuroCyto**
NeuB1	0.52	0.51	0.51	0.41	0.45	0.47	0.50	0.38	0.46	0.42	0.43	0.36
NeuB2	0.26	0.24	0.25	0.19	0.21	0.22	0.21	0.15	0.23	0.19	0.22	0.14
DRIVE	0.79	0.76	0.75	0.63	0.68	0.73	0.71	0.58	0.68	0.63	-	-
STARE	0.42	0.41	0.42	0.26	0.35	0.39	0.38	0.29	0.35	0.32	-	-

TABLE III: DIADEM scores of comparison methods on the neuronal and the retinal datasets. The first four columns are variants of our tracing system, while the next six columns are reported by replacing the tracing component of our system with different methods. The last two columns are results of two existing neuronal tracing systems.

	Ours	**Soares [13]**	**FRVD [15]**	**OOF [57]**	**Frangi [31]**
NeuB1	0.52	0.47	0.43	0.38	0.34
NeuB2	0.26	0.23	0.21	0.17	0.12
DRIVE	0.79	0.71	0.69	0.66	0.51
STARE	0.42	0.36	0.33	0.30	0.26

TABLE IV: Tracing performance of our system (MFTD$_a$) by replacing the segmentation component with different methods, evaluated in term of DIADEM scores. See text for details.

on the other hand delivers the worst DIADAM scores. Both methods however can not work with the retinal datsets. Overall our tracing system MFTD$_a$ consistently outperforms the other methods and especially existing tracing systems [4], [5], [35] by a rather large margin, which suggests the advantage of our digraph-theoretical approach for tracing filamentary structures. The statistical significance of the performance gain of MFTD$_a$ w.r.t. each of the comparison methods has been evaluated using the Wilcoxon signed rank test. It is found that at $p = 0.05$ significance level that in all cases, the performance gains are

statistically significant. Moreover, the better performance of MFTD$_a$ over MFTD$_b$ might due to that row-wise normalization tends to bias the solution toward local minimums. One main difference of our system MFTD$_a$ versus the graphical model methods (MF, TRW, and LBP) and LLGC are that the latter ones are all based on undirected graphs. Instead, the adoption of digraph representation in our system *retains* directional information related to the root and terminal nodes, which explicitly rules out those improper propagation paths in graphs and by doing so, helps to resolve confusions of contacting filaments alongside the junctions. Meanwhile, it is not surprising that existing digraph-based methods SGL and ZFL only delivers modest results, which we mainly attribute to the adoption of teleporting as well as symmetrization tricks that unfortunately damage useful information contained in the input raw signals. Table IV reports the final DIADEM score performance of our segmentation method vs. competing segmentation methods reported previously in the segmentation step. Here only the segmentation module of our system is replaced by the corresponding segmentation method, while the rest of the system remain unchanged. Empirically the final tracing result of the proposed segmentation algorithm clearly outperforms the others, which again suggests the applicability of our new segmentation module. Tables V, VI, VII and VIII also provide the results evaluated by Netmets, which provide more finely categorized evaluation details. Overall the performance of the comparison methods is similar w.r.t. DIADEM.

Figs. 9 and 10 present the visual tracing results of several images on neuronal datasets. It is not known how MMNX is operated internally for these tracing tasks as being a commercial software. Nonetheless, empirically we observe as in Fig. 9 that MMNX tends to separate the filamentary network by forming spatial clusters of subgraphs on clean and well-focused images, and as in Figs. 10 MMNX tends to misplace the neuronal labels of the filaments. These unfortunately will not necessarily render good results, as indicated by the amount of mistakes made in Figs. 9 and 10 marked out with the pink-colored disks. In contrast, the results of our tracing system MFTD$_a$ have consistently far less mistakes. As MMNX can not deal with retinal images, we instead compare with the LBP tracing system as shown in Fig.11, which can be regarded as an improved version of [35] since a number of components in our system (for example the segmentation step) are different from that of [35]. To facilitate better interpretation of the visual comparisons, neurons in Figs. 9, and 10 are tagged with their own DIADEM scores, with the corresponding IDs of these neurons marked in their original images displayed on the topmost row. Moreover, grey and pink colored disks are used to represent the correct and the wrong predictions on various crossover scenarios, respectively. For example, as shown in image (1) of Fig. 9 for NeuB1, there exists a crossover in the junction between neurons 2 and 3, which are successfully picked up by our system MFTD$_a$ and SGL, but is missed out by MMNX. Another example is in image (3) of Fig. 10 for NeuB2, where there exists a crossover in the junction between neurons 1 and 2. It is successfully picked up by MFTD$_a$, but is missed out by both SGL and MMNX. Many similar situations can be observed from the exemplar images in Figs. 9 and 10. It is also visually observed that over these neurons in both NeuB1 and NeuB2, consistently higher DIADEM scores are obtained by MFTD$_a$, which is followed by SGL, while MMNX often produces worst DIADEM scores. Again, this suggests that overall graph-theoretical approaches tends to deliver better results than MMNX, and our approach performs notably well when comparing to other methods. Visual inspections into retinal datasets are also presented in Fig. 11. Similar to the neuronal datasets, our system clearly produces better tracing results when comparing to the SGL and the LBP tracing systems. On the other hand, there are still mistakes made by our system: Among them, one major error source stems from the segmentation step where topological connections might sometimes be altered; Another error source is related to the construction of our current weight matrix where just angular property of neighbouring filaments in a junction is considered. These are the limitations of our current tracing system.

We also directly evaluate the crossovers where two or more filamentary objects (e.g. retinal blood vessels or neurons) touch or crossover each other. In Table IX, a crossover refers to a junction point, whereas a pair refers to a unique pair of filaments (as in Fig. 4) connected at a junction point that belongs to the same object (e.g. neuron). Note there will be only true positives and false negatives in this context. The accuracy in the table is thus calculated as: $\text{Accuracy} = \frac{\text{\#of true positives}}{\text{\#of (true positives + false negatives)}}$. Overall our approach (i.e. MFTDa) clearly outperforms MMNX in term of addressing the crossover issues.

V. Conclusion and Outlook

A graph-theoretical approach is proposed to trace filamentary structures of both neuronal and retinal blood vessel images. Key to our approach is the newly discovered connection between the tracing problem and the established algebraic digraph theory, which we hope may open doors for new insights into the tracing problem. The competitiveness of our approach is verified in various empirical experiments, with justification being provided. For future work, we would like to consider the inclusion of complementary information such as transition of thickness or image texture along a filament around a junction when constructing the digraph weight matrix. The performance of our tracing system could also be further improved by incorporating a more advanced segmentation module.

Acknowledgements

This research was supported by A*STAR JCO grant 1231BFG040. We would like to thank Dr. Lin Gu for helping with the 2D neuronal data processing.

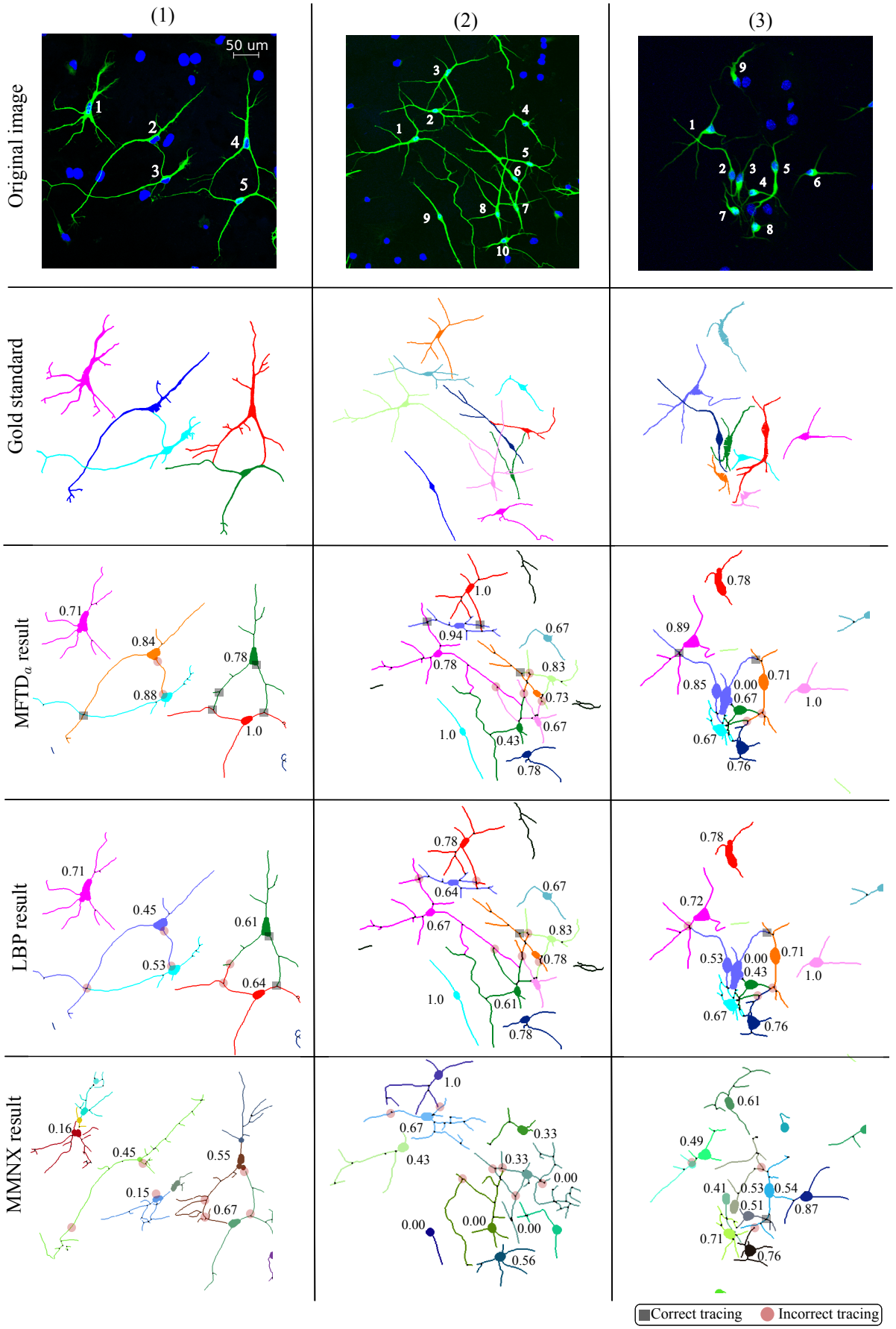

Fig. 9: Exemplar tracing results on the neuronal dataset NeuB1. Correct and incorrect tracing are highlighted by grey and pink colored disks, respectively. Numbers in the images denote DIADEM scores for each of the particular neurons. Best viewing in color.

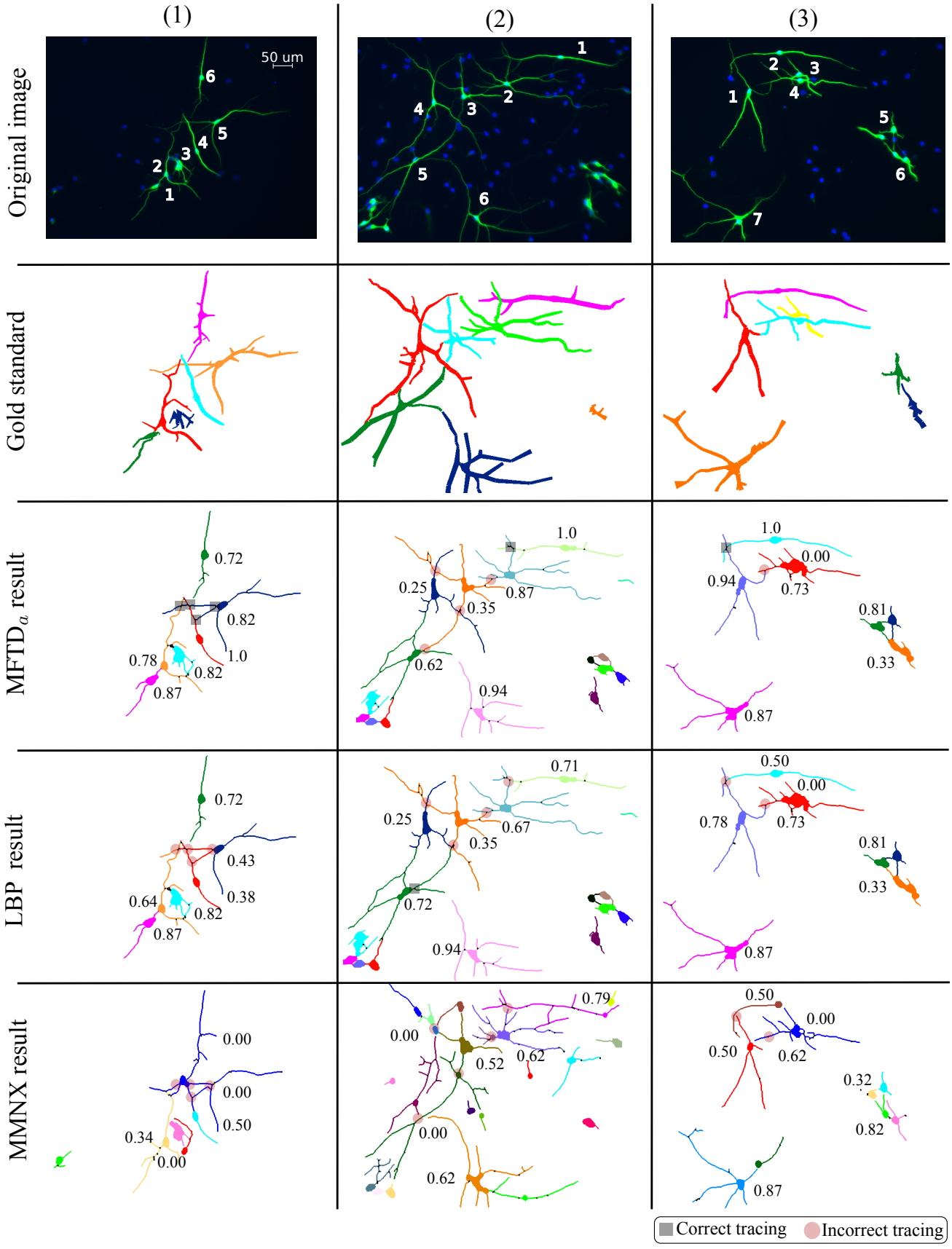

Fig. 10: Examples of tracing results on the neuronal dataset NeuB2. Correct and incorrect tracing are highlighted by grey and pink colored disks, respectively. Numbers in the images denote DIADEM scores for each of the particular neurons. Best viewing in color.

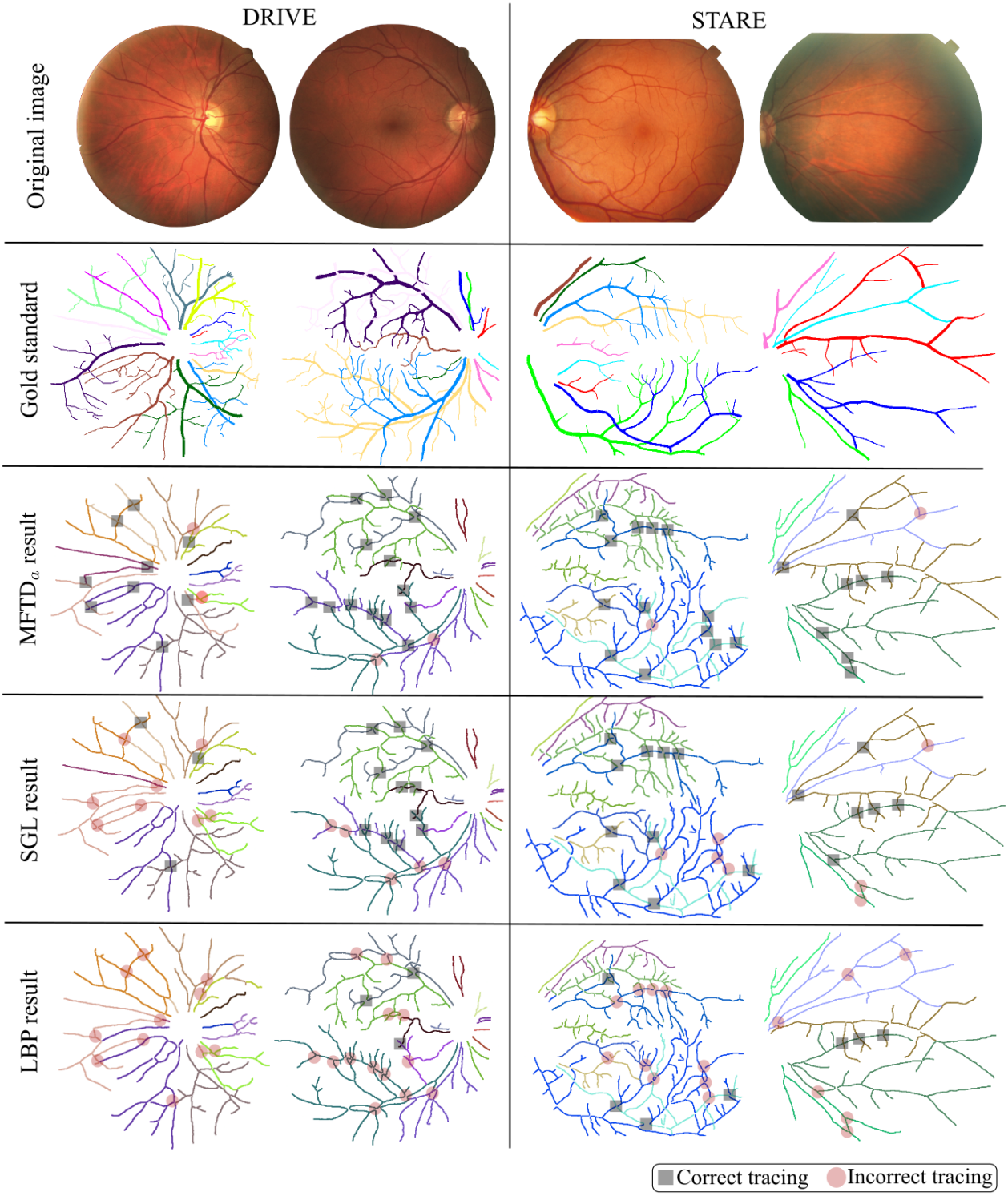

Fig. 11: Examples of tracing results on the retinal datasets DRIVE and STARE. Correct and incorrect tracing are highlighted by grey and pink colored disks, respectively. Best viewing in color.

	MFTDa	MFTDb	MFTDc	MFTDd	SGL	ZFL	LLGC	MF	TRW	LBP	MMNX	NeuroCyto
NeuB1	0.21	0.22	0.21	0.25	0.24	0.23	0.22	0.30	0.23	0.26	0.25	0.33
NeuB2	0.38	0.41	0.39	0.45	0.49	0.43	0.46	0.51	0.47	0.48	0.43	0.52
DRIVE	0.09	0.11	0.12	0.15	0.14	0.13	0.12	0.20	0.13	0.18	-	-
STARE	0.26	0.28	0.28	0.31	0.31	0.29	0.27	0.37	0.31	0.34	-	-

TABLE V: One of the four Netmets scores, geometric false positive rate, on the neuronal and the retinal datasets.

	MFTDa	**MFTDb**	**MFTDc**	**MFTDd**	**SGL**	**ZFL**	**LLGC**	**MF**	**TRW**	**LBP**	**MMNX**	**NeuroCyto**
NeuB1	0.20	0.22	0.21	0.26	0.25	0.23	0.23	0.29	0.25	0.27	0.26	0.32
NeuB2	0.36	0.38	0.37	0.43	0.42	0.40	0.43	0.47	0.43	0.45	0.42	0.51
DRIVE	0.08	0.10	0.10	0.13	0.13	0.12	0.12	0.18	0.15	0.16	-	-
STARE	0.25	0.27	0.26	0.30	0.29	0.28	0.35	0.31	0.33	0.31	-	-

TABLE VI: One of the four Netmets scores, geometric false negative rate, on the neuronal and retinal datasets.

	MFTDa	**MFTDb**	**MFTDc**	**MFTDd**	**SGL**	**ZFL**	**LLGC**	**MF**	**TRW**	**LBP**	**MMNX**	**NeuroCyto**
NeuB1	0.77	0.78	0.78	0.81	0.81	0.79	0.80	0.85	0.81	0.82	0.82	0.89
NeuB2	0.81	0.85	0.83	0.91	0.89	0.87	0.88	0.91	0.88	0.90	0.86	0.93
DRIVE	0.62	0.64	0.63	0.69	0.68	0.67	0.76	0.71	0.75	0.73	-	-
STARE	0.79	0.80	0.80	0.83	0.83	0.82	0.81	0.88	0.83	0.81	-	-

TABLE VII: One of the four Netmets scores, connectivity false positive rate, on the neuronal and the retinal datasets.

	MFTDa	**MFTDb**	**MFTDc**	**MFTDd**	**SGL**	**ZFL**	**LLGC**	**MF**	**TRW**	**LBP**	**MMNX**	**NeuroCyto**
NeuB1	0.79	0.80	0.80	0.85	0.84	0.83	0.81	0.88	0.83	0.86	0.85	0.88
NeuB2	0.83	0.85	0.86	0.91	0.93	0.91	0.89	0.94	0.92	0.93	0.91	0.96
DRIVE	0.65	0.67	0.66	0.69	0.69	0.68	0.67	0.71	0.69	0.68	-	-
STARE	0.79	0.80	0.81	0.88	0.86	0.83	0.81	0.89	0.85	0.86	-	-

TABLE VIII: One of the four Netmets scores, connectivity false negativity rate, on the neuronal and the retinal datasets.

Datasets	#of crossovers	#of pairs	#of true positives		#of false negatives		Accuracy	
			Ours	MMNX	Ours	MMNX	Ours	MMNX
NeuB1	557	865	472	389	393	476	0.546	0.449
NeuB2	254	371	199	67	172	304	0.536	0.180
DRIVE	405	849	429	-	420	-	0.505	-
STARE	121	263	112	-	151	-	0.426	-

TABLE IX: A direct comparison of our approach (i.e. MFTDa) and MMNX on crossover accuracy. See text for details.

REFERENCES

[1] M. Pool, J. Thiemann, A. Bar-Or, and A. E. Fournier, "Neuritetracer: A novel ImageJ plugin for automated quantification of neurite outgrowth," *Journal of Neuroscience Methods*, vol. 168, no. 1, pp. 134–139, 2008.

[2] M. Longair, D. Baker, and J. Armstrong, "Simple neurite tracer: Open source software for reconstruction, visualization and analysis of neuronal processes," *Bioinformatics*, vol. 27, no. 17, pp. 2453–4, 2011.

[3] H. Peng, Z. Ruan, F. Long, J. Simpson, and E. Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnology*, vol. 28, no. 4, pp. 348–53, 2010.

[4] W. Yu, H. Lee, s. Hariharan, W. Bu, and S. Ahmed, "Quantitative neurite outgrowth measurement based on image segmentation with topological dependence," *Cytometry Part A*, vol. 75A, no. 4, pp. 289–297, 2009.

[5] "Metamorph NX," http://www.moleculardevices.com/systems/metamorph-research-imaging/metamorph-nx-microscopy-automation-image-analysis-software, 2015.

[6] K. Viswanath and D. McGavin, "Diabetic retinopathy: Clinical findings and management," *Community Eye Health*, vol. 16, no. 46, pp. 21–4, 2003.

[7] B. Al-Diri, A. Hunter, and D. Steel, "An active contour model for segmenting and measuring retinal vessels," *IEEE Trans. Med. Imag*, vol. 28, no. 9, pp. 1488–97, 2009.

[8] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*. MIT Press, 2006.

[9] R. Agaev and P. Chebotarev, "Spanning forests of a digraph and their applications," *Automation and Remote Control*, vol. 62, no. 3, pp. 443–66, 2001.

[10] F. Harary, R. Norman, and D. Cartwright, *Structural models : an introduction to the theory of directed graphs*. Wiley, 1965.

[11] V. Vapnik, *Statistical learning theory*. Wiley, 1998.

[12] A. Mendonca and A. Campilho, "Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1200–13, 2006.

[13] J. Soares, J. Leandro, R. Cesar, H. Jelinek, and M. Cree, "Retinal vessel segmentation using the 2D Gabor wavelet and supervised classification," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1214–22, 2007.

[14] D. Marin, A. Aquino, M. Gegundez-Arias, and J. Brav, "A new supervised method for blood vessel segmentation in retinal images by using gray-level and moment invariants-based features," *IEEE Trans. Med. Imag.*, vol. 30, no. 1, pp. 146–158, 2011.

[15] P. Bankhead, C. Scholfield, J. McGeown, and T. Curtis, "Fast retinal vessel detection and measurement using wavelets and edge location refinement," *PLoS ONE*, vol. 7, no. 3, p. e32435, 2012.

[16] A. Can, H. Shen, J. Turner, H. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Trans. Inf. Tech. in Biomed.*, vol. 3, no. 2, pp. 125–38, 1999.

[17] E. Grisan, A. Pesce, A. Giani, M. Foracchia, and A. Ruggeri, "A new tracking system for the robust extraction of retinal vessel structure," in *IEEE International Conf. of the Engineering in Medicine and Biology Society*, 2004, pp. 1620–3.

[18] E. Turetken, C. Blum, G. Gonzalez, and P. Fua, "Reconstructing geometrically consistent tree structures from noisy images," in *Medical Image Computing and Computer-Assisted Interventation (MICCAI)*, 2010, pp. 291–9.

[19] X. Xu, M. Niemeijer, Q. Song, M. Sonka, M. Garvin, J. Reinhardt, and M. Abramoff, "Vessel boundary delineation on fundus images using graph-based approach," *IEEE Trans. Med. Imag.*, vol. 30, no. 6, pp. 1184–91, 2011.

[20] E. Bas, E. Cansizoglu, D. Erdogmus, and J. Kalpathy-Cramer, "Retinal vasculature segmentation using principal spanning forests," in *IEEE International Symposium on Biomedical Imaging (ISBI)*, 2012, pp. 1792–5.

[21] E. Bekkers, R. Duits, T. Berendschot, and B. ter Haar Romeny, "A multi-orientation analysis approach to retinal vessel tracking," *Journal of Math. Imag. and Vision*, vol. 49, no. 3, pp. 583–610, 2014.
[22] M. Fraz, P. Remagnino, A. Hoppe, B. Uyyanonvara, A. Rudnicka, C. Owen, and S. Barman, "Blood vessel segmentation methodologies in retinal images - a survey," *Comput. Methods Prog. Biomed.*, vol. 108, no. 1, pp. 407–33, 2012.
[23] E. Meijering, M. Jacob, J. Sarria, P. Steiner, H. Hirling, and M. Unser, "Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images," *Cytometry Part A*, vol. 58, no. 2, pp. 167–76, 2004.
[24] E. Turetken, G. Gonzalez, C. Blum, and P. Fua, "Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors," *Neuroinformatics*, vol. 9, pp. 279–302, 2011.
[25] E. Meijering, "Neuron tracing in perspective," *Cytometry A*, vol. 77, no. 7, pp. 693–704, 2010.
[26] A. Choromanska, S. Chang, and R. Yuste, "Automatic reconstruction of neural morphologies with multi-scale graph-based tracking," *Frontiers in Neural Circuits*, vol. 6, no. 25, pp. 1–13, 2012.
[27] Y. Wang, A. Narayanaswamy, C. Tsai, and B. Roysam, "A broadly applicable 3-D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, no. 2-3, pp. 193–217, 2011.
[28] H. Xiao and H. Peng, "APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree," *Bioinformatics*, vol. 29, no. 11, pp. 1448–54, 2013.
[29] A. Santamaría-Pang, P. Hernandez-Herrera, M. Papadakis, P. Saggau, and I. Kakadiaris, "Automatic morphological reconstruction of neurons from multiphoton and confocal microscopy images using 3D tubular models," *Neuroinformatics*, vol. 13, no. 3, pp. 1–24, 2015.
[30] T. Gillette, K. Brown, and G. Ascoli, "The diadem metric: comparing multiple reconstructions of the same neuron," *Neuroinformatics*, vol. 9, no. 2-3, pp. 233–45, 2011.
[31] R. Frangi, W. Niessen, K. Vincken, and M. Viergever, "Multiscale vessel enhancement filtering," in *Medical Image Computing and Computer-Assisted Interventation (MICCAI)*, 1998, pp. 130–7.
[32] F. Benmansour and L. Cohen, "Tubular structure segmentation based on minimal path method and anisotropic enhancement," *Int. J. Comput. Vision*, vol. 92, no. 2, pp. 192–210, 2011.
[33] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua, "Supervised feature learning for curvilinear structure segmentation," in *Medical Image Computing and Computer-Assisted Interventation (MICCAI)*, 2013, pp. 526–33.
[34] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Computing Surveys*, vol. 36, pp. 81–121, 2000.
[35] J. De, T. Ma, H. Li, M. Dash, and L. Cheng, "Automated tracing of retinal blood vessels using graphical models," in *Scandinavian conf. on Image analysis*, 2013, pp. 277–89.
[36] L. Cheng, , J. De, X. Zhang, F. Lin, and H. Li, "Tracing retinal blood vessels by matrix-forest theorem of directed graphs," in *Medical Image Computing and Computer-Assisted Interventation (MICCAI)*, 2014, pp. 626–33.
[37] D. Zhou, J. Huang, and B. Schölkopf, "Learning from labeled and unlabeled data on a directed graph," in *International conference on Machine learning (ICML)*, 2005, pp. 1036–43.
[38] X. Cai, H. Wang, H. Huang, and C. Ding, "Simultaneous image classification and annotation via biased random walk on tri-relational graph," in *European Conference on Computer Vision (ECCV)*, 2012, pp. 823–36.
[39] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf, "Learning with local and global consistency," in *Neural Information Processing Systems (NIPS)*, 2004, pp. 321–8.
[40] F. Chung, "Laplacians and the Cheeger inequality for directed graphs," *Annals of Combinatorics*, vol. 9, no. 1, pp. 1–19, 2005.
[41] H. Wang, C. Ding, and H. Huang, "Directed graph learning via high-order co-linkage analysis," in *European Conference on Machine Learning (ECML)*, 2010, pp. 451–66.
[42] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford University, Technical Report, 1998.
[43] R. Brualdi and H. Ryser, *Combinatorial Matrix Theory*. Cambridge Uni. Press, 1991.
[44] P. Chebotarev and R. Agaev, "Forest matrices around the Laplacian matrix," *Linear Algebra and its Applications*, vol. 356, no. 1-3, pp. 253–74, 2002.
[45] C. Meyer and M. Stadelmaier, "Singular m-matrices and inverse positivity," *Linear Algebra and its Applications*, vol. 22, pp. 139–56, 1978.
[46] T. Davis, "Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.*, vol. 30, pp. 196–9, 2004.
[47] J. Demmel, S. Eisenstat, J. Gilbert, X. Li, and J. Liu, "A supernodal approach to sparse partial pivoting," *SIAM J. Matr. Anal. App.*, vol. 20, pp. 720–55, 1999.
[48] C. Paige and M. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, pp. 43–71, 1982.
[49] Y. Saad, *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.
[50] L. Bégin, P. Germain, F. Laviolette, and J.-F. Roy, "PAC-Bayesian theory for transductive learning," in *International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2014, pp. 105–13.
[51] P. Derbeko, R. El-Yaniv, and R. Meir, "Explicit learning curves for transduction and application to clustering and compression algorithms," *J. Artif. Intell. Res.*, vol. 22, pp. 117–142, 2004.
[52] "Invitrogen inc." http://www.lifetechnologies.com/, 2015.
[53] "Sigma-Aldrich inc." http://www.sigmaaldrich.com/, 2015.
[54] D. Myatt, T. Hadlington, G. Ascoli, and S. Nasuto, "Neuromantic - from semi-manual to semi automatic reconstruction of neuron morphology," *Frontiers in Neuroinformatics*, vol. 6, no. 4, pp. 1–14, 2012.
[55] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–9, 2004.
[56] A. Hoover, V. Kouznetsova, and M. Goldbaum, "Locating blood vessels in retinal images by piecewise threshold probing of a matched filter response," *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 203–10, 2000.
[57] M. Law and A. Chung, "Three dimensional curvilinear structure detection using optimally oriented flux," in *European Conference on Computer Vision (ECCV)*, 2008, pp. 368–82.
[58] D. Mayerich, C. Bjornsson, J. Taylor, and B. Roysam, "NetMets: software for quantifying and visualizing errors in biological network segmentation," *BMC Bioinformatics*, vol. 13, no. S8, pp. 1–19, 2012.
[59] M. Wainwright and M. Jordan, *Graphical models, exponential families, and variational inference*, ser. Foundations and trends in machine learning. Hanover: Now Publisher, 2008.