

Learning to Boost Filamentary Structure Segmentation

Lin Gu

Bioinformatics Institute, A*STAR, Singapore

gulin@bii.a-star.edu.sg

Li Cheng, *Senior Member*

Bioinformatics Institute, A*STAR, Singapore

School of Computing, National University of Singapore, Singapore

chengli@bii.a-star.edu.sg

Abstract—The challenging problem of filamentary structure segmentation has a broad range of applications in biological and medical fields. A critical yet challenging issue remains on how to detect and restore the small filamentary fragments from backgrounds: The small fragments are of diverse shapes and appearances, meanwhile the backgrounds could be cluttered and ambiguous. Focusing on this issue, this paper proposes an iterative two-step learning-based approach to boost the performance based on a based segmenter arbitrarily chosen from a number of existing segmenters: We start with an initial partial segmentation where the filamentary structure obtained are of high confidence based on this existing segmenter. We also define a scanning horizon as epsilon balls centred around the partial segmentation result. Step one of our approach centers on a data-driven latent classification tree model to detect the filamentary fragments. This model is learned via a training process, where a large number of distinct local figure/background separation scenarios are established and geometrically organized into a tree structure. Step two spatially restores the isolated fragments back to the current partial segmentation, which is accomplished by means of completion fields and matting. Both steps are then alternated with the growth of partial segmentation result, until the input image space is entirely explored. Our approach is rather generic and can be easily augmented to a wide range of existing supervised/unsupervised segmenters to produce an improved result. This has been empirically verified on three specific filamentary structure segmentation tasks: retinal blood vessel segmentation as well as 2D and 3D neuronal segmentations, where noticeable improvement has been shown over the original state-of-the-arts.

I. INTRODUCTION

This work aims to address the problem of image-based filamentary structure segmentation in 2D or 3D. In particular, we focus on the challenging issue of preserving weak foreground signals, i.e., small and thin filaments from ambiguous backgrounds. This problem is fundamental in a rather broad range of applications such as neuronal tracing from microscopic images [1], retinal blood vessel tracing in retinal scans [2], as well as reconstruction of human vasculature such as 2D digital subtraction angiography and 3D magnetic resonance angiography [3]. Difficulties of this problem lie in the high variability of filament shape, texture and thickness, which is further complicated by the often noisy and cluttered background that at times could even confuse a trained eye [4].

Existing methods can be roughly categorized into three types: Hessian-based, model-based and learning-based. Hessian-based models make use of the second order derivatives either to guide the development of snake [5], to detect filament edges [6], or to combine with the eigenvalues [3] for segmenting filamentary structures. They however often lack the flexibility to tackle irregular filamentary structures. Model-based methods instead emphasize on fitting filaments with known geometric shapes. Zhao *et al.* [7] regard individual fibres of neurons as connected tubular shapes which are assembled to form neuronal tree structures. One widely used unsupervised method, optimally oriented flux (OOF) [8], is based on the assumption of circular filament cross-sections, which is further extended in Turetken *et al.* [9] to segment filamentary structures through a set of regularly spaced anchor points. Learning-based methods [10], [11], [12], [13], on the other hand, advocate the automation of the feature learning process. For example, The method of [13] employs a gradient boosting framework to optimize filters and often produces the state-of-the-art performance. Existing filamentary structure segmentation methods usually work very well when the filamentary structured foregrounds are of high contrast or with clear boundary from the backgrounds. Their performance nevertheless deteriorates dramatically when dealing with small & thin filaments. This is often further complicated with cluttered and ambiguous backgrounds, which are not uncommon in real-world images.

In addition to the basic segmentation result, many of the aforementioned methods also produce a pixelwise *confidence map*, despite the heterogeneous nature of these methods. A confidence map is a spatial mapping with each image pixel assigned a non-negative score, which is larger if this pixel more likely belongs to the filamentary structure foreground, or lower if the other way around. This concept has in fact been adopted by existing methods under different names, such as vesselness [3] and turbularity score [9], [12]. It has been observed that when placing a sufficiently high threshold on the confidence map, many such methods are able to output solely true positive foregrounds which produces a partial segmentation that usually contains the main trunk (i.e. the long & thick filaments). Similarly, when placing a sufficiently low threshold, many of these methods are able to produce

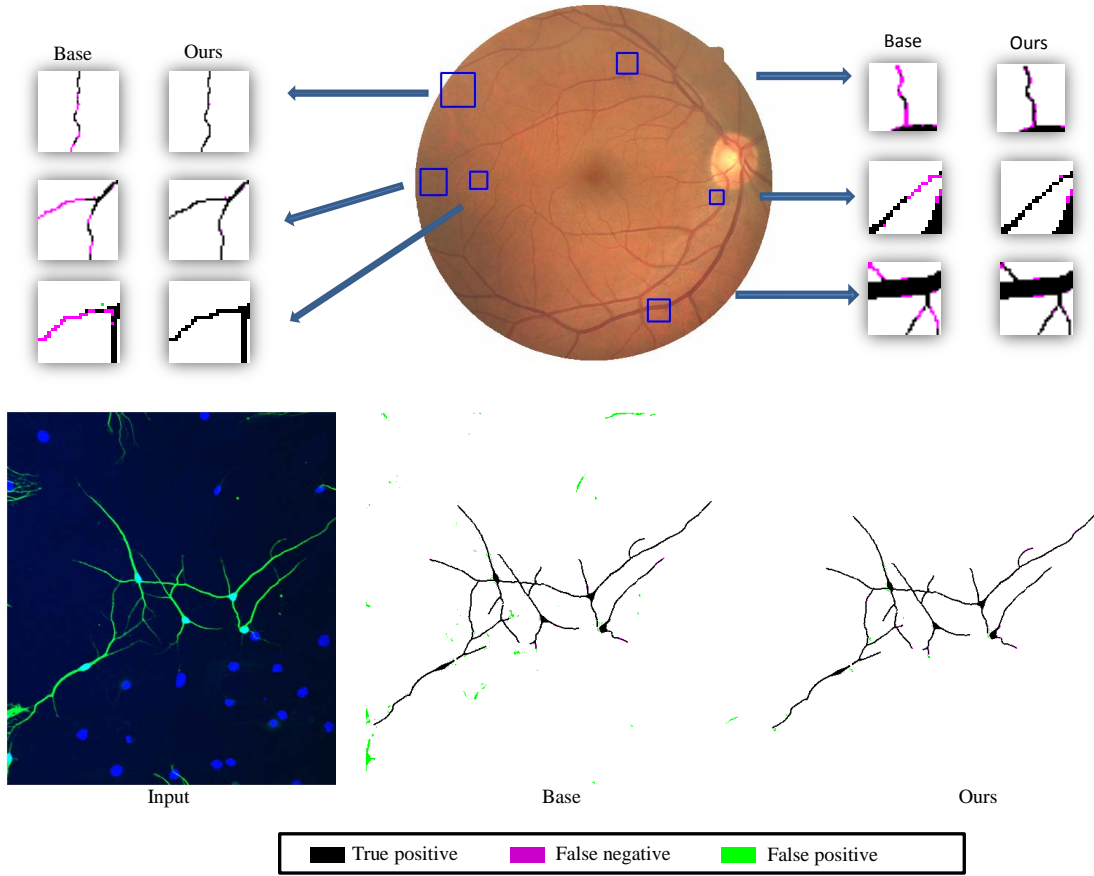


Fig. 1: Exemplar results of our approach on two applications: (first row) 2D retinal and (second row) 2D neuronal images. For each application, the results of the base segmenter [13] and results of our boosted are displayed. Note the blue channel in the neuronal image indicate the DAPI-tagged nuclei which can be ignored in our context. In each of the results, Black pixels refer to true positive, magenta pixels are for false negative (missing), while lime pixels are for false positive (false alarm).

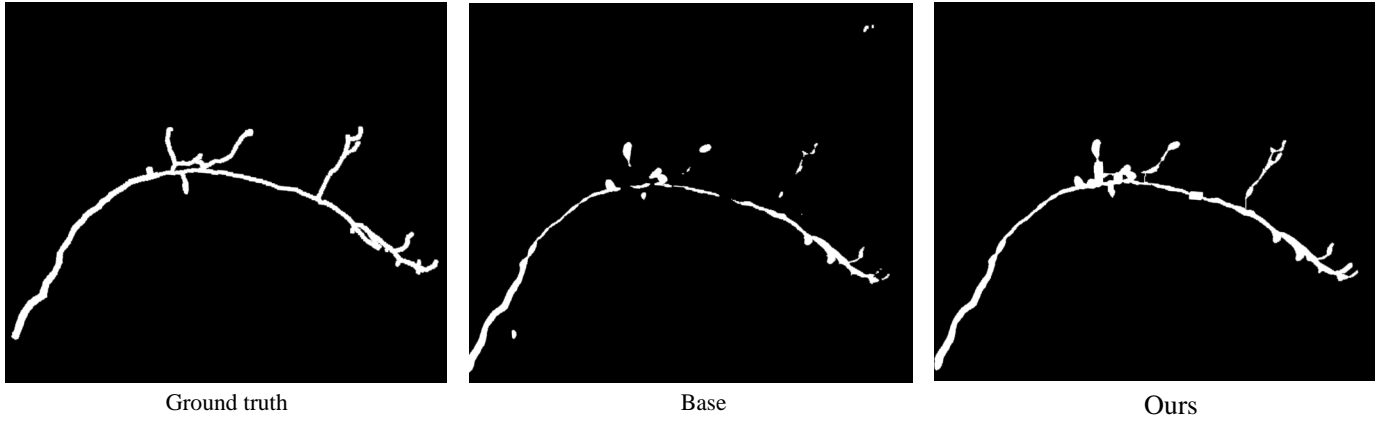


Fig. 2: An example of our approach on 3D neuronal images. The result of a base segmenter [8], and our boosted result are shown.

only true negative backgrounds. The observations inspires us to propose a data-driven learning approach aiming at boosting the performance of these existing segmentation methods. Our approach specifically focuses on detecting and restoring small foreground filaments from ambiguous backgrounds, a bottleneck issue of many state-of-the-art methods. To achieve this, an iterative two-step pipeline is proposed. More specifically, it contains a latent classification tree model that allows a data-driven geometrical organization of filamentary patterns. This

facilitates the decomposition of the challenging detection task into a large number of local figure/background separation sub-problems, organized in a tree structure, where the geometric similarities between the sub-problems are preserved as the topological distances between tree nodes. Step one focuses on detecting filamentary fragments under a predefined scanning horizon, i.e., epsilon balls centred around current partial segmentation result. Step two spatially restores these newly detected fragments by connecting them back to the current

partial segmentation. Then the two steps are alternated until the image space is entirely explored. Empirically our approach is demonstrated to boost the performance of a wide spectrum of exemplar existing methods such as [6], [8], [10], [13], which contains supervised and unsupervised methods, as well as Hessian-based, model-based, and learning-based methods.

to do: related work on 3D neurite segmentation and DIADEM challenge

II. OUR APPROACH

Existing filamentary structure segmentation methods usually work well when the filamentary structured foregrounds are of high contrast or with clear boundary from the backgrounds. Their performance nevertheless decreases dramatically when dealing with small & thin filaments as well as ambiguous backgrounds. Nevertheless, when placing a sufficiently high (or low) threshold on the confidence map, many such methods are able to produce solely true positive foregrounds (or true negative backgrounds). In addition, when placing a global threshold on the remaining pixels, one would always end up with detecting fragment candidates that are a mixture of filamentary fragments and background fragments.

Motivated by these observations, based on one of the existing filamentary structure segmentation methods that meet the above quantifications (which we will refer to as a base segmenter), our approach can be regarded as a value-added process to improve its segmentation performance. As illustrated in Fig.3, provided with an input image and the confidence map output by the base segmenter, the pipeline of our approach consists of the following steps:

Preprocessing: Obtain the partial segmentation (black-colored main trunk in Fig.3(c)) by placing a sufficiently high threshold on the confidence map. Define a scanning horizon as epsilon balls centered around current partial segmentation in the image space.

Step One: In the remaining low confidence regions, detect the filamentary fragments (red components in Fig.3(c)) via a latent classification tree model. The latent classification tree model is learned based on a large number of distinct local figure/background separation scenarios, which are geometrically organized into a tree structure. The detected filamentary fragments are on the other hand usually isolated from the main trunk due to missing edges.

Step Two: Grow current filamentary structure by restoring the detected filamentary fragments, i.e., connecting them back to the main trunk. This is achieved by making novel usage of the matting technique guided with the completion fields of these filamentary fragments.

Progress Check: Update the scanning horizon. Go back to step one if the image space has not been entirely explored, otherwise terminate.

A. Preprocessing

For an input image, its confidence map is obtained by applying a base segmenter. by placing a threshold τ_h we obtain a partial segmentation that comprises a set of pixels

with confidence values exceeding τ_h . Note τ_h is sufficiently high to ensure that only filamentary foreground pixels are selected. In a similar manner, we can also define another threshold τ_l to be sufficiently low for pure backgrounds. At this moment, a partial foreground segmentation is obtained, while the rest pixels remain undecided. Our task then becomes that of (step one) detecting filamentary fragments from backgrounds and (step two) connecting them back to current foreground segmentation. For this purpose we define a scanning horizon as an epsilon ball B around the seed with a searching radius ϵ .

B. Step One: Detect Filamentary Fragments via latent classification tree (LCT) Model

As displayed in Fig.3(c), the long & thick thick filaments (i.e. main trunk) are usually successfully detected as the initial partial segmentation, while the bottleneck issue becomes that of identifying and reclaiming the small and thin fragments. Given a base segmenter, starting from an initial partial segmentation, we focus on the weak confidence pixels within horizon B . The aim of step one is to detect filamentary fragments from backgrounds in those pixels with confidence values in the range of (τ_l, τ_h) . This is a rather challenging problem, mainly due to significant variability of local geometrical shapes and textures upon which we rely to identify and reclaim filament foregrounds from the ambiguous backgrounds. Simply placing a global threshold in the range of (τ_l, τ_h) will inevitably results in the detection of fragment candidates which are a mixture of filamentary fragments (coloured red) and ambiguous background fragments (coloured blue) as illustrated in Fig.3(c). To address this problem, we instead propose a data-driven strategy to learn distinct local shape and texture scenarios of filamentary fragments, with the underlying assumption that for any test instance, similar scenarios would usually exist in training data. This naturally corresponds to a large number of distinct local figure/background separation scenarios, which are established and geometrically organized into a tree structure, a latent tree model. This divide-and-conquer strategy becomes beneficial in several aspects: It helps to reduce the complexity of the original problem to something that can be managed case by case; It also facilitates the introduction of specific completion fields that each dedicates to a group of filament fragments having similar shapes and textures.

1) *Identify Distinct Local Filament Shapes and Textures by Filament Tokens:* Following the idea of sketch tokens [14], we would identify distinct local filamentary fragments that are referred to as *filament tokens*. It works by by grouping similar filamentary fragments in terms of their shapes, textures, and orientations in image patches (i.e. 2D image patches or 3D image cubes) centered on the centerline of filaments obtained from the set of labeled training images. A wide spectrum of features could be applicable in our context, while the histogram of gradient (HOG) feature is employed as being relatively insensitive to slight spatial offsets. Note the HOG feature is directly operated on the ground-truth segmentation to extract local shapes of various filamentary fragments. This is followed by k-means clustering to partition these filamentary

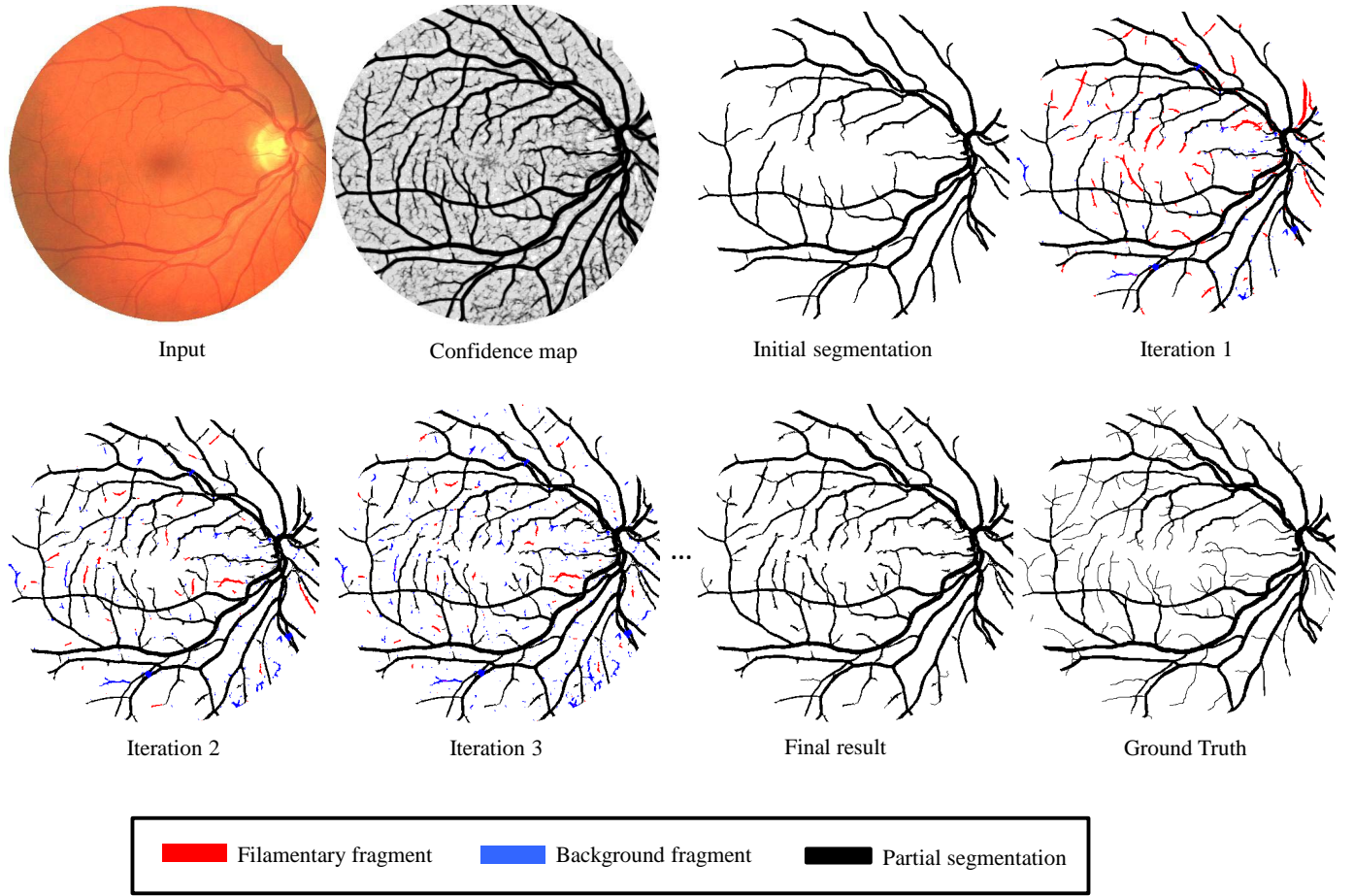


Fig. 3: An illustration of the pipeline of our approach on a 2D retinal image. From top-left to bottom-right in scan-line order: (a) Input image; (b) The confidence map from a base segmenter. Here pixels with lower intensity values correspond to higher confidence scores. (c) Initial partial segmentation obtained by applying a sufficiently high threshold in the confidence map. (d-f) Intermediate results of iteration 1-3, where the result of step 1 is shown for each iteration. (g) The final reconstructed filamentary structure, in comparison with (h) the ground truth.

fragment examples into T token types based on their HOG features. Each token type corresponds to a specific group of filamentary fragments with similar geometry pattern. In this paper, the patch size for 2D images are fixed to 31×31 and for 3D images is fixed to $31 \times 31 \times 15$. T is set to 180.

2) *Detect Filamentary Fragments by LCT Model*: When the filament tokens are prepared, the remaining problem of step one becomes that of solving a large number of token type dependant figure/background separation scenarios. A straightforward approach would be that of employing a multi-class classifier such as support vector machine or random forest with respect to token types. This approach nevertheless treats each token type as an independent one, which ends up tackling an unnecessarily hard classification problem with large number of token types. In fact, these token types inherently possess hierarchical structures that characterize their non-uniform pairwise distances. Thus it would be advantageous to exploit the topology structure within the token types. This inspire us to consider the latent tree models [15], [16], which in our context is a tree-structured probabilistic organization of these *local* figure/background separation scenarios, where the foregrounds of each scenario come from one unique token type, and the

backgrounds are formed by nearby background fragments. The tree structure is obtained by the Chow-Liu neighbor joining (CLNJ) method [15]: A Chow-Liu tree is constructed over all the token types to provide guidance on the groups of token types that tend to be topologically close to each other. Provided with the Chow-Liu tree and started from each of the individual token types as a leaf node, a neighbor joining strategy is thus recursively employed to build sub-trees in a bottom-up manner by connecting a group of several closest neighbors at a time, until an entire tree is formed in the end. In this paper, the distance between a pair of token types is determined by the Euclidean distance of the afore-mentioned HOG features describing the two token types.

Fig.5 illustrates an exemplar CLNJ tree, where each split node (including the root node) possesses an internal multi-class classifier to assign a fragment candidate to the corresponding sub-tree it belongs to. At test run, a fragment candidate starts at the root node of the LTM, and descends to one of the sub-trees following decision made by invoking the current split node's classifier. This process is repeated until the candidate reaches a leaf node. Each leaf node also maintains a binary classifier with the purpose of retrieving true filament

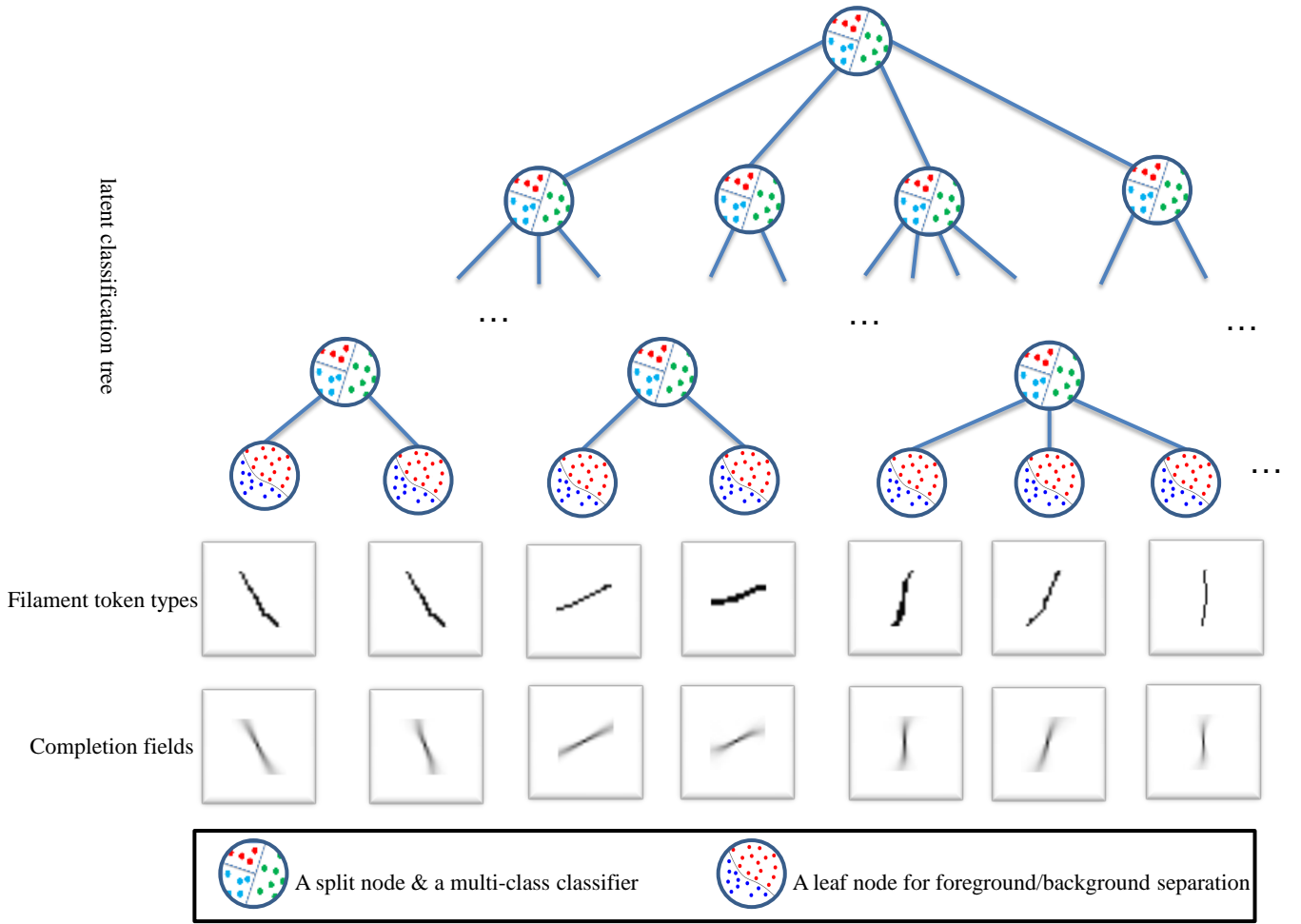


Fig. 5: An illustration of the latent classification tree (LCT) model in step one: A latent split node which comprises a subset of similar tokens and is split into multi-branches according to a multi-class classifier; A leaf node corresponds to a local foreground/background separation problem, with the set of foreground filaments precisely corresponding to a unique token type, as displayed in the second row. For each token type, its completion field is presented in the third row. In the legend: a split node contains a multi-class classifier, while a leaf node has its local foreground/background separation problem.

fragments from backgrounds under its local context. if it passes this verification process, the token type of the leaf node will be assigned to the fragment candidate. Note that these classifiers could employ an arbitrary multi-class classification method. After empirical evaluation we settle down with the random forest, which gives the best performance in our context.

In what follows, we would specify the set of features engaged by these classifiers used in our LCT model.

3) *Features for the LCT Model:* Once the tree structure of the LCT model is settled, we are left to decide the classifiers utilized in our LCT model. Two types of features are considered by the classifiers utilized in our LCT model, namely the filamentary-structure features and the low-level features.

Denote s a fragment candidate, for any pixel in the fragment $\mathbf{x} \in s$, let $c(\mathbf{x})$ be its confidence score. As displayed in Fig.3(c), shapes of the filamentary fragments often preserve the filamentary structures as being thin and elongated, while the background fragments could be of arbitrary shape. This motivates us to consider the following features that capture such shape discriminations: Given a fragment candidate s , we first approximate a typical filament fragment by an ellipse,

and denote as two features the major and minor axes of the surrogate ellipse, $f_1(s)$ and $f_2(s)$. Second, the size (2D area or 3D volume, depending on the context) as well as the accumulative confidence score of the fragment are calculated, as $f_3(s) = \sum_{\mathbf{x} \in s} 1$, $f_4(s) = \sum_{\mathbf{x} \in s} c(\mathbf{x})$. Third, ideally a true filament fragment would connect naturally with the main branch (i.e. partial segmentation) by a smooth curvilinear interconnector. This inspires us to consider using the Dijkstra's algorithm to find a path $\mathcal{S}_{s,b}$, a sequence of connected pixels along the possible path, with each denoted as \mathbf{x}' , that attaches the fragment candidate s back to current main branch b by minimizing following objective function

$$\sum_{\mathbf{x}' \in \mathcal{S}_{s,b}} -\log(c(\mathbf{x}')). \quad (1)$$

Intuitively this amounts to encouraging a short path along high confidence pixels. For the obtained optimal path, its objective value becomes $f_5(s)$ and its average curvature is $f_6(s)$.

In addition to above filamentary-structure features, we also consider the low-level raw features. For 2D images, the Kernel Boost Features [13] are utilized as follows. The Kernel Boost method is firstly trained on the set of filamentary structured

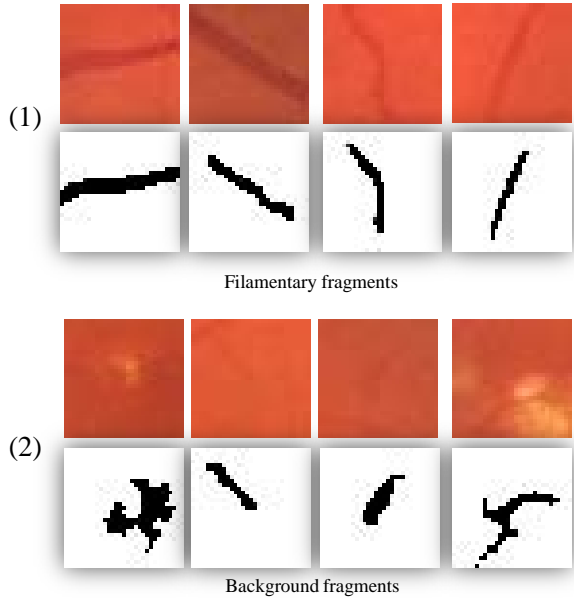


Fig. 4: (1) and (2) present exemplar filamentary fragments vs. background fragments encountered during step one, respectively where the first row shows the input image patch and the second row shows the fragments.

training images to yield discriminative convolutional filters. The obtained filters are then applied on a 2D image patch centered on the specific fragment candidate s , which gives the kernel boost features as the filtered output. The low-level feature of 3D images are simply the intensity value of image cube of a fixed size. Similar to the filament token parameters discussed previously, the patch size for 2D images are fixed to 31×31 and cube size is fixed to $31 \times 31 \times 15$ for 3D images.

C. Step Two: Restoring the Filamentary Fragments

Given the filamentary fragments detected from step one, in step two, we elaborate on connecting these isolated fragments back to the main trunk. The leading difficulty here lies in the fact that in the confidence map, the detected filament fragments and current partial segmentation are spatially separated by an uninformative zone of weak confidence values. We thus consider the incorporation of *completion fields* in the augmented confidence map that provides prior information on how to join the fragments back to the main trunk. This is followed by a novel utilization of the matting technique to connect fragments back to the main trunk.

1) Augmented Confidence Map by the Completion Fields:

The concept of completion fields is inspired by the work of [17] and others on contour completion, and is used in our context to provide prior information to guide the joining of the detected fragments back to the main trunk. The filament tokens not only group similar filamentary fragments into distinct token types according to their shapes and textures, they also provide a good opportunity to establish the completion fields as follows. Each token type corresponds to a group of similar filamentary fragments from the training dataset, while each fragment is associated with an image patch (i.e. a 2D image patch or a 3D image cube) and its corresponding ground-truth label patch, where background and foreground pixels

are labeled as 0 and 1, respectively. Now, for each token type, its completion field is obtained by taking the average of these label patches. As a result, each detected filamentary fragment from test run has its completion field as illustrated in Fig. 6. Moreover, a global completion field is obtained by summing up all the local completion fields from individual filamentary fragments. In other words, if a pixel is contained in the completion fields of multiple fragments, its score in the global completion field is obtained by simply adding up all the individual scores. In this manner, when there are a sequence of nearby but broken fragments presented along a missing filament, the influence of their completion fields can be considered altogether, as illustrated in Fig. xxx. The augmented confidence map $\bar{c}(\cdot)$ is therefore realized by adding up the confidence map and the global completion field scores for each pixel, where both the confidence map and the global completion field are normalized separately to within the range of $[0, 1]$. For any pixel \mathbf{x} , its augmented confidence score is $\bar{c}(\mathbf{x})$.

2) *Restoring the Filamentary Fragments by Matting*: Denote the known background and foreground pixels as 0 and 1, respectively, and the rest as unknown. At this moment, many of the foreground pixels are known, including the detected filamentary fragments and the main trunk as discussed previously. A number of background pixels can also be obtained by those with sufficiently low confidence scores (below τ_l). It remains to decide on the rest unknown pixels. Interestingly our current situation resembles that of the seemingly unrelated matting problem, which aims at finding the α values of the unknown pixels as a composition of foreground and background layers, i.e.

$$I(\mathbf{x}) = \alpha(\mathbf{x})F(\mathbf{x}) + (1 - \alpha(\mathbf{x}))B(\mathbf{x}), \quad (2)$$

where \mathbf{x} indexes a particular (2D or 3D) pixel location, $\alpha(\mathbf{x}) \in [0, 1]$ is its alpha matte value, $F(\cdot)$ and $B(\cdot)$ denote the foreground and background layers, respectively. The filamentary fragments can then be connected back to the main trunk by imposing a threshold δ to categorize into foreground and background pixels as those with α values $\geq \delta$ and $< \delta$, respectively.

While a wide range of image matting techniques can be applied, in this paper we adopt the KNN matting of [18], where the α values are computed as

$$\alpha = (L_c + \lambda M)^{-1}(\lambda \mathbf{v}). \quad (3)$$

Let n denote the number of pixels in an (2D or 3D) image. Here M is a $n \times n$ diagonal matrix, with $M(i, i) = 1$ if pixel i is a foreground or a background pixel, and $M(i, i) = 0$ otherwise. \mathbf{v} is a binary vector of pixels corresponding to the known foregrounds and backgrounds. The clustering Laplacian $L_c = (D - K)^T(D - K)$, where $K = [k_{ij}]$ a $n \times n$ affinity matrix with each entry k_{ij} being the affinity value between pixels i and j , $D = [d_{ii}]$ is the diagonal degree matrix of K , $d_{ii} = \sum_j k_{ij}$. $\lambda > 0$ is a scalar constant. More details of the KNN matting technique [18] are relegated to Appendix . With a slight abuse of notation, the affinity matrix in our context is defined as $k_{ij} = \bar{c}(i) + \bar{c}(j)$, where i and j index two pixel locations that are within the completion fields and with their

augmented confidence scores above $\tau_m \in (\tau_l, \tau_h)$, a scalar threshold.

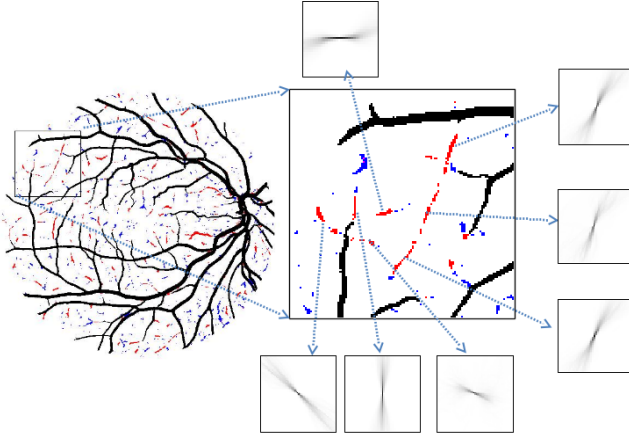
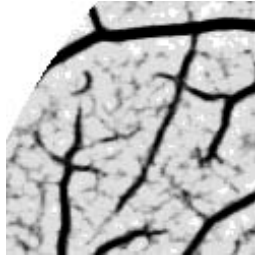


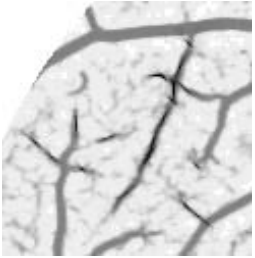
Fig. 6: An example of completion fields of detected filamentary fragments.



Confidence map



Completion fields



Augmented confidence map



Matting result

Fig. 7: Show a running example of matting and how the missing connections are restored.

III. EMPIRICAL EXPERIMENTS

The performance of our approach is evaluated on two types of filamentary structured segmentation applications, namely retinal blood vessel segmentation and neuronal segmentation, which contain both 2D and 3D datasets as follows.

A. Datasets

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

Two 2D Retinal Vessel Datasets: DRIVE and STARE For retinal blood vessel tracing, the publicly available testbeds

of DRIVE [19] and STARE [20] 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.

Our In-house Datasets of 2D Neuronal Microscopic Images To facilitate the analysis of 2D neuronal segmentation systems, we make available our annotated neuronal dataset, wishing this can provide an option for researchers in the field to compare performance on the same ground. 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 $10\mu\text{g}/\text{ml}$ concentration, from Invitrogen Inc) 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 secondary antibodies Alexa Fluor (AF)-488 goat anti-mouse, while the cell nuclei are counterstained with DAPI. In the end, 112 images are acquired and manually annotated, where one third of the images are reserved for training. Meanwhile, the ground-truths of the rest two third images are retained for performance evaluation purpose, which amount to 675 neurons.

3D Neuronal Dataset from the DIADEM Challenge Besides 2D images, our approach is also evaluated on the 3D neuronal images (also referred to as image stacks), namely the OPF dataset from the DIADEM challenge datasets [21]. It consists of 9 drosophila olfactory axonal projection image. Following the DIADEM Grand Challenge Completion, we took the first three images for training and reserved the rest of 6 for test evaluation.

B. Evaluation Details and Analysis

Base Segmenters The following six segmentation methods are considered as the base segmenters in this paper:

- (1) Kernel Boost [13] which learn discriminative convolutional filters based on the Gradient Boosting framework.
- (2) Optimally Oriented Flux (OOF) [8], an unsupervised method with manually designed filter, widely used for delineating tubular structures.
- (3) IUWT [6], an unsupervised method based on isotropic undecimated wavelet transform.
- (4) Eigen [3], a multiscale Hessian-based method.
- (5) T2T [22], a supervised method based on pixel classification, medial sub-tree generation, and global tree linking components.
- (6) LEFD [23] which is an unsupervised model-based method. Due to the capacity of individual methods, most existing methods work only for one or two out of the three application scenarios, with the notably exception of OOF [8]. On the other hand, unfortunately OOF is usually not the best performer.

Parameters & Performance Evaluation Criteria The internal parameters in our approach are set to the following values in this paper. thresholds $\tau_h = 0.9$, $\tau_m = 0.7$, $\tau_l = 0.3$, $\epsilon = 5$, $\delta = 0.9$, trade-off value $\lambda = 1000$, and the token

types $T = 180$. Typically, our method ends within 5 iterations. For fair comparison, the internal parameters of the comparison methods are fixed to values that have been tuned to give the best overall results.

The standard evaluation metric is *accuracy*, evaluated in term of the sum of all true positive and true negative counts divided by the total number of instances (i.e. pixels). This metric however will be severely biased when the filamentary foreground pixels are far less than the background pixels, which is often the case in our context. Besides, even manual “ground-truth” annotation might contain errors especially at foreground boundaries. These inspire us to adopt a *modified F1* measure to allow minor location offsets similar to [24], [25], as follows: When a segmented object is not exactly matched to the ground-truth annotation, it will give a miss as well as a false alarm. In light of this, we assign each miss pixel the distance of its nearest false alarm pixel. If there is a nearest false alarm within deviation range r , the miss pixel is still regarded as a true positive. The precision is thus computed as $\frac{tp}{t}$ where tp is the number of true positive pixels and the t is the total number of positive pixel. In our experiments, the deviation range is set to $r = 1.5$ for 2D applications and $r = 1.8$ for 3D applications. Following [25], the recall is computed based on the centreline of the true positive and ground-truth image. The modified F1 measure is then defined as the harmonic mean of the precision and recall. In all the experiments, the best such F1 score is reported. In addition, for 3D neuronal datasets we evaluate the metric of averaged absolute centerline deviation (AACD) as follows. Let \mathcal{P} and \mathcal{G} denote the sets of predicted and ground-truth foreground pixels, respectively. Denote a foreground pixel $x_p \in \mathcal{P}$ and similarly a background pixel $x_g \in \mathcal{G}$. Define $d(x_p, x_g)$ the 3D Euclidean distance between these two pixels. The AACD metric between the predicted and the ground-truth segmentation is defined by

$$\text{AACD}(\mathcal{P}, \mathcal{G}) = \text{avg} \left(\min_{x_p \in \mathcal{P}, x_g \in \mathcal{G}} (d(x_p, x_g)) \right), \quad (4)$$

where avg is an average operator.

C. 2D Retinal Blood Vessels Segmentation

Table I summarizes the quantitative evaluations of our approach with different base segmenters on two retinal testbeds: DRIVE and STARE, where the evaluation criterion is in term of the modified F1 score. Overall our approach is able to consistently improve the performance of various existing methods with up to 7% gain. This amount is in fact quite noticeable, given the huge amount of existing research efforts on these two datasets. As shown in Table I, the simplified variant of our approach, Ours-nCnM, skip the completion fields & matting in step two by simply connecting each of the fragments back to main trunk as a straight lines. The other variant, Ours-CnM, is obtained with completion fields but without matting, i.e. by utilizing the Dijkstra method instead of matting. Ours refers to the full version that comes with both completion fields and matting. It is observed that (1) Ours outperforms Ours-CnM, which outperforms Ours-nCnM, which in turn outperforms the base method. (2) Empirically

TABLE I: Performance on the two 2D retinal testbed DRIVE and STARE with different base methods. The modified F1 measure (%) is used. Ours-nCnM refers to the simplified variant of our approach without using completion fields and matting, while Ours-CnM instead refers to with completion field but without matting. Ours is the full version of our approach. The last row shows the performance increments of our full approach comparing with various base methods.

	Kernel Boost [13]	OOF [8]	IUWT [6]
Base	76.29 / 78.77	69.97 / 72.86	64.13 / 72.20
Ours-nCnM	79.12 / 79.63	71.22 / 73.28	71.79 / 74.15
Ours-MnM	79.44 / 81.14	71.93 / 75.91	72.25 / 74.29
Ours	80.11 / 81.93	72.12 / 76.84	72.65 / 74.58
gain	5.01 / 3.99	3.07 / 5.46	13.29 / 3.30

TABLE II: Accuracy comparison of methods on 2D retinal testbeds.

	DRIVE			STARE		
	Prec	Recall	Acc	Prec	Recall	Acc
Ours based on [13]	79.57	79.47	97.32	83.59	73.72	97.72
Kernel Boost [13]	80.31	77.58	97.19	84.54	72.29	97.64
N4-fields [26]	81.57	78.39	96.59	-	-	-
OOF [8]	73.12	65.43	95.84	73.03	68.09	94.68
Soares [10]	69.43	74.25	94.42	75.68	74.40	94.80
Bankhead [6]	70.27	71.77	93.71	67.46	0.7897	94.59
Ricci [27]	-	-	95.63	-	-	95.84
Marin [28]	-	70.67	94.52	-	69.44	95.26
Mendonca [29]	-	73.44	94.52	-	69.96	94.40
Martinez-Perez [30]	-	72.46	93.44	-	75.06	94.10

the mostly simplified variant accounts for 40% -80% of the performance gain, while the completion field and matting processes collectively leads to the rest gain. Moreover, Table II presents a direct comparison with a number of known and additional state-of-the-art retinal segmentation methods. Here the boosted results of our approach based on [13] is regarded as our results, as it produces the best results among all base segmenters during Table I. Note the performance evaluation criteria in Table II is switched to the standard accuracy metric, with the aim of best utilizing existing results as reported in literature. Here our approach gives the best results over both datasets and all comparison methods. A possible question one may raise is that the performance gain is much smaller here. This is in fact to be expected. As explained previously, one major drawback of this accuracy metric lies on its heavy bias toward the majority class when the class-dependent population becomes very unbalanced: As we have a lot more background pixels than filamentary foreground pixels, the number of false negative (missing) or even false positive (false alarm) instances actually do not matter much as long as we have sufficient true negative instances. This also explains why almost all methods can easily reach beyond 90% accuracy (as most background pixels can be classified relatively easily), while it is on the other hand so difficult to improve further when the accuracy goes over e.g. 97%. As can be seen in Fig. ??, overall our approach is able to recover more challenging filamentary fragments that are ignored by other methods including the base method of [13]. We also evaluate the robustness of our approach. As presented in Figure ??, our approach is quite insensitive w.r.t. varying the internal parameters K and T .

As can be seen in the representative samples, with the help of LTM, our method is able to recover more challenging filaments that are ignored by the base method [13]. Compared to two specially designed vessel segmentation methods [2], [10], our method is able to recover comparable amount of

TABLE III: Performance on 2D Neuronal dataset with different base methods. The modified F1 measure (%) is used. Ours-nCnM refers to the simplified variant of our approach without using completion fields and matting, while Ours-CnM instead refers to with completion field but without matting. Ours is the full version of our approach. The last row shows the performance increments of our full approach comparing with various base methods.

	Kernel Boost [13]	OOF [8]	Eigen [3]	T2T [22]
Base	84.74	63.50	63.94	66.49
Ours-nCnM	86.20	65.99	66.81	66.18
Ours-MnM	86.39	70.25	70.38	75.70
Ours	86.80	71.03	71.89	76.01
gain	2.43	11.86	12.43	14.32

filaments while avoiding bringing the ambiguous backgrounds.

D. 2D Neuronal Segmentation

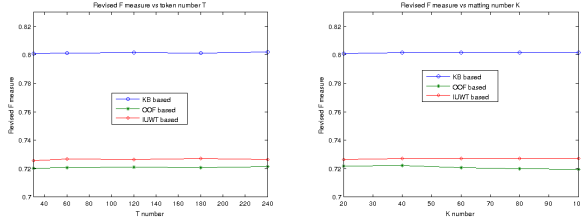


Fig. 9: Performance deviation with varying K and T values on 2D retinal testbed of DRIVE. The modified F1 measure (%) is used.

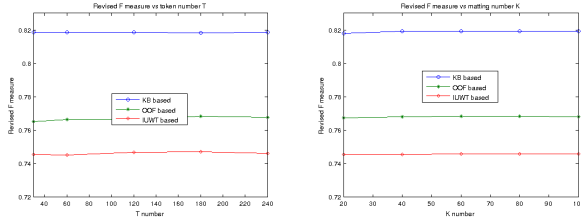


Fig. 10: Performance deviation with varying K and T values on 2D retinal testbed of STARE. The modified F1 measure (%) is used.

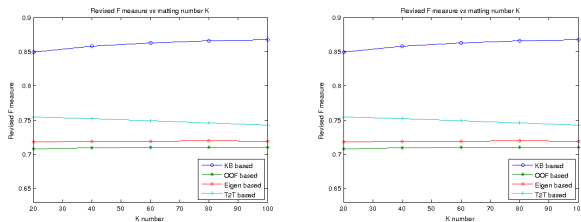


Fig. 11: Performance deviation with varying K and T values on 2D neuron testbed. The modified F1 measure (%) is used.

Table III shows that on average our approach produces around 2% performance gain in term of the modified F1 measure. Similar to 2D retinal datasets, our approach is able to boost the performance over different base methods ranging from unsupervised [8], [3] to supervised such as [13], [22]. This is also observed in the follow-up experiments to be described during 3D neuronal segmentation.

E. 3D Neuronal Segmentation

Experiments are conducted on the 3D Olfactory Projection Fibers (OPF) neuronal dataset from the DIADEM challenge, while the base methods are OOF [8], Eigen [3], and

TABLE IV: Performance on 3D Neuronal OPF dataset with different base methods using modified F1 measure.

	OOF [8]	Eigen [3]	LEFD [23]
Base method	49.17	52.27	54.50
Our method	53.26	57.27	58.56
gain	4.09	5.00	4.06

TABLE V: Performance on 3D neuronal OPF dataset with different base methods using AACD metric (in pixel unit).

	OOF [8]	Eigen [3]	LEFD [23]
Base method	3.4162	6.4849	2.5055
Our method	1.7860	1.7352	1.5915

LEFD [23]. As presented in Table IV when using the modified F1 measure, a performance boost of 4–5% is obtained from all three existing methods when our approach is additionally used. Moreover, a close examination of the amount of centerline deviation using the AACD metric as described in Table V reveals that our approach dramatically reduce the deviation by at least a half over the base methods.

IV. CONCLUSION AND OUTLOOK

To address the problem of image-based filamentary structure segmentation, we propose a value-added approach to improve over a broad set of existing segmenters, with an emphasis on addressing the challenging aspect of preserving small and thin filaments from ambiguous backgrounds. This is achieved by a learning based iterative pipeline that start from an initial partial segmentation, to detect filamentary fragments with a novel LCT model and to restore them back to the current partial segmentation and repeat until there is no change in the segmentation result. Our approach is empirically demonstrated to be capable of improving over a number of existing methods on very different applications. For future work, we plan to evaluate on new biomedical applications such as digital subtraction angiography and magnetic resonance angiography.

APPENDIX

Image matting regards an observed image I as a linear combination of foreground layer F and background layer B where the alpha matte $\alpha(x) \in [0, 1]$ determines the proportion the mixture for an unknown pixel indexed by x .

$$I(x) = \alpha(x)F(x) + (1 - \alpha(x))B(x) \quad (5)$$

TABLE VI: Performance on the 3D Big Neuron Challenge Dataset. The F1 measure (%) is used. The last row shows the performance increments of our full approach comparing with various base methods.

	Multi-Scale Enhancement	Fast Marching
Base	64.62	63.20
Ours	67.79	64.02
gain	4.91	1.3

TABLE VII: Performance on the 3D Big Neuron Challenge Dataset. The F1 measure (%) is used. The last row shows the performance increments of our full approach comparing with various base methods.

	Multi-Scale Enhancement	Fast Marching
Base	64.62	63.20
Ours	67.79	64.02
gain	4.91	1.3

TABLE VIII: Tracing Performance on the 3D Big Neuron Challenge Dataset. The F1 measure (%) is used. The last row shows the performance increments of our full approach comparing with various base methods.

	Multi-Scale Enhancement	Fast Marching
Base	43.47	44.69
Ours	49.59	50.59
gain	14.08	13.21

For example, known foregrounds are those of alpha matt $\alpha = 1$ while the known backgrounds are assigned 0. This leads us to use image matting techniques to explore the existence of filaments by their observed physical proportion rather than a binary segmentation.

Imaging matting refers to the problem of accurately estimating the composition of foregrounds and backgrounds. They are largely categorised into sampling based matting and propagation based matting. The sampling based matting collect the best representative samples to estimates the unknown pixel's alpha matt. On the other hand, the propagation based matting uses the affinity of pixels to propagate the alpha value from known region to unknown region. Setting the previously recovered fragment and bases as known foregrounds and the low score area ($p(x) < \tau_1$) as background, we seeks to estimate the alpha matt of challenging filaments in the rest of place. In this section, we capitalise the nonlocal principle which assumes the expected value of the alpha matt as

$$E[\alpha_i] \approx \sum_j \alpha_j K(i, j) \frac{1}{D_i} \quad (6)$$

$$D_i = \sum_j (j) K(i, j) \quad (7)$$

Where $K(i, j)$ is the affinity value between i and j . Representing all α values across the image as α , Equation 7 becomes

$$D_i \alpha_i \approx K(i, j)^T \alpha \quad (8)$$

In the form of matrix, Equation 78 are derived as

$$D\alpha \approx K\alpha \quad (9)$$

where $D = \text{diag}(D_i)$ is an $N \times N$ and K is an $N \times N$ affinity matrix, where N is the total number of pixels. The alpha matt α could thus be calculated by solving

$$\alpha^T L_c \alpha \approx 0 \quad (10)$$

where $L_c = (D - K)^T(D - K)$ is called the clustering Laplacian.

Ensuring the α is closed to the values of foregrounds and backgrounds in the known part, the final closed solution for the matting is

$$\alpha = (L_c + \lambda M)^{-1}(\lambda \mathbf{v}) \quad (11)$$

where M is a $N \times N$ diagonal matrix where $M(i, i) = 1$ if pixel i is in a known area and $M(i, i) = 0$ if not. The \mathbf{v} is a binary vector of pixels corresponding to the known foregrounds and backgrounds.

To recover the α from Equation 11, we define the affinity matrix based on the completion field of individual fragment s_k , where $K(i, j) = 0.5(E_k(i) + E_k(j) + c(i) + c(j))$ if both i, j is in its completion field and both of their confidence value are above threshold τ_m . Thus the completion field of fragment s_k could properly guide the alpha propagation. Restraining

the affinity relation inside the completion field as shown in Fig also enforces the sparsity which could greatly reduce the computation burden.

REFERENCES

- [1] E. Meijering, "Neuron tracing in perspective," *Cytometry A*, vol. 77, no. 7, pp. 693–704, 2010.
- [2] J. De, H. Li, and L. Cheng, "Tracing retinal vessel trees by transductive inference," *BMC Bioinformatics*, vol. 15, no. 20, pp. 1–20, 2014.
- [3] A. Frangi, W. Niessen, K. Vincken, and M. Viergever, "Multiscale vessel enhancement filtering," in *MICCAI*, 1998.
- [4] M. Abramoff, M. Niemeijer, M. Viergever, and B. Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 501–509, 2004.
- [5] 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.
- [6] P. Bankhead, C. Scholfield, J. McGeown, and T. Curtis, "Fast retinal vessel detection and measurement using wavelets and edge location refinement," *PLoS ONE*, 2012.
- [7] T. Zhao, J. Xie, F. Amat, N. Clack, P. Ahammad, H. Peng, F. Long, and E. Myers, "Automated reconstruction of neuronal morphology based on local geometrical and global structural models," *Neuroinformatics*, vol. 9, no. 2-3, pp. 247–261, 2011.
- [8] M. Law and A. Chung, "Three dimensional curvilinear structure detection using optimally oriented flux," in *ECCV*, 2008.
- [9] 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, no. 2-3, pp. 279–302, 2011.
- [10] J. Soares, J. Leandro, R. Cesar, H. Jelinek, and M. Cree, "Retinal vessel segmentation using the 2-d gabor wavelet and supervised classification," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1214–22, 2006.
- [11] R. Rigamonti and V. Lepetit, "accurate and efficient linear structure segmentation by leveraging ad hoc features," in *MICCAI*, 2012.
- [12] E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua, "Reconstructing loopy curvilinear structures using integer programming," in *CVPR*, 2014.
- [13] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua, "Supervised feature learning for curvilinear structure segmentation," in *MICCAI*, 2013.
- [14] J. Lim, C. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *CVPR*, 2013.
- [15] M. Choi, V. Tan, A. Anandkumar, and A. Willsky, "Learning latent tree graphical models," *JMLR*, vol. 12, pp. 1771–1812, 2011.
- [16] R. Mourad, C. Sinoquet, N. Zhang, T. Liu, and P. Leray, "A survey on latent tree models and applications," *JAIR*, vol. 47, pp. 157–203, 2013.
- [17] L. Williams and D. Jacobs, "Stochastic completion fields: A neural model of illusory contour shape and salience," *neural computation*, vol. 9, no. 4, pp. 837–58, 2006.
- [18] Q. Chen, D. Li, and C. Tang, "Knn matting," in *CVPR*, 2012.
- [19] 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. 501–09, 2004.
- [20] 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.
- [21] K. Brown, G. Barrionuevo, A. Canty, P. De, J. Hirsch, G. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. Ascoli, "The diadem data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions," *Neuroinformatics*, vol. 9, pp. 143–157, 2011.
- [22] S. Basu, A. Aksel, B. Condrón, and S. Acton, "Tree2tree: Neuron segmentation for generation of neuronal morphology," in *ISBI*, 2010.
- [23] S. Basu, C. Liu, and G. Rohde, "Localizing and extracting filament distributions from microscopy images," *Journal of Microscopy*, vol. 258, no. 1, pp. 13–23, 2015.
- [24] C. Becker, K. Ali, G. Knott, and P. Fua, "Learning context cues for synapse segmentation," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1864–77, 2013.
- [25] E. Turetken, C. Becker, P. Glowacki, F. Benmansour, and P. Fua, "Detecting irregular curvilinear structures in gray scale and color imagery using multi-directional oriented flux," in *ICCV*, 2013.
- [26] Y. Ganin and V. Lempitsky, "N4-fields: Neural network nearest neighbor fields for image transforms," in *ACCV*, 2014, pp. 536–51.

- [27] E. Ricci and R. Perfetti, "Retinal blood vessel segmentation using line operators and support vector classification," *IEEE Trans. Med. Imag.*, vol. 26, no. 10, pp. 1357–65, 2007.
- [28] D. Marin, A. Aquino, M. Gegundez-Arias, and J. Bravo, "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. 146–58, 2011.
- [29] 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.
- [30] M. Martinez-Perez, A. Hughes, S. Thom, A. Bharath, and K. Parker, "Segmentation of blood vessels from red-free and uorescein retinal images," *Med Image Anal.*, pp. 47–61, 2007.

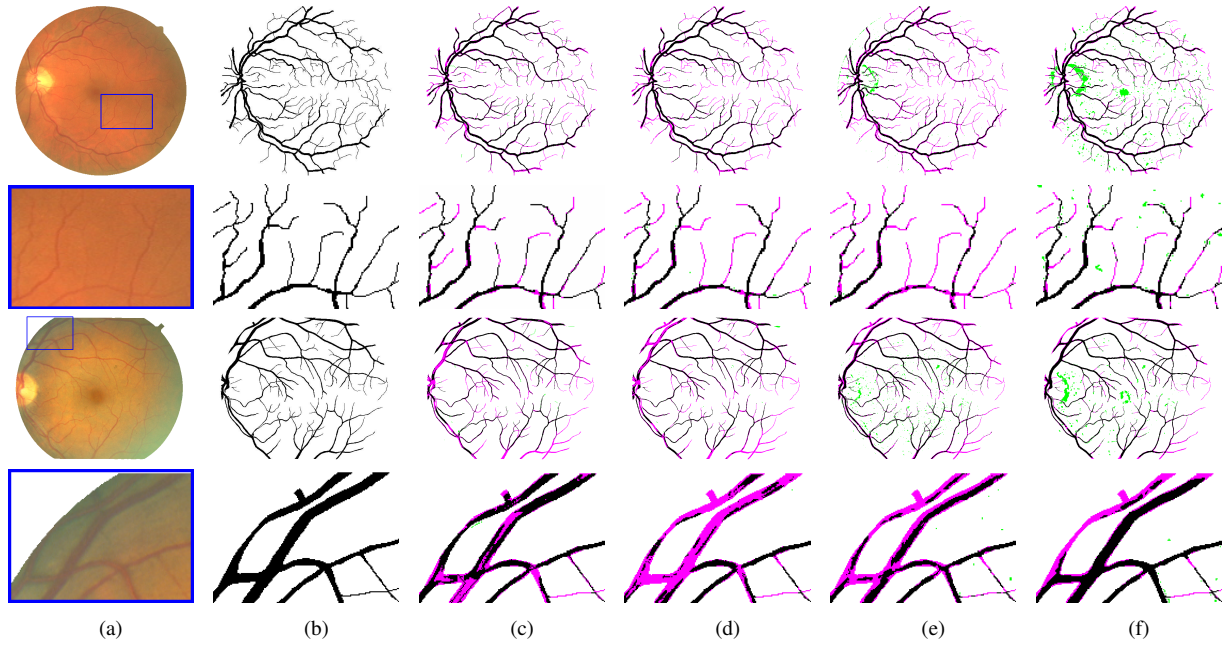


Fig. 8: Performance on 2D retinal datasets of DRIVE and STARE datasets. The top two rows correspond to DRIVE and the bottom two rows correspond to STARE. For each dataset, top row shows the full image while the bottom row presents a detailed zoom-in view of a cropped region. Columns are in the following order: (a) Input image. (b) Ground-truth annotation. (c) our method based on [13]. (d) Kernel Boost [13]. (e) OOF [8]. (f) IUWT [6]. True positive is denoted in black, false positive in cyan, and false negative in green.

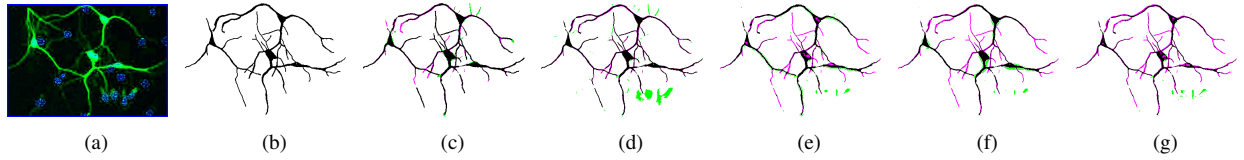


Fig. 12: Performance of 3D neuronal dataset. (a) Input image where the contrast has been increased for visual presentation. (b) Ground-truth annotation. (c) Our method based on [13]. (d) Kernel Boost [13]. (e) OOF [8]. (f) Eigen [3]. (g) T2T [22]. True positive is denoted in black, false positive in cyan and false negative in green.

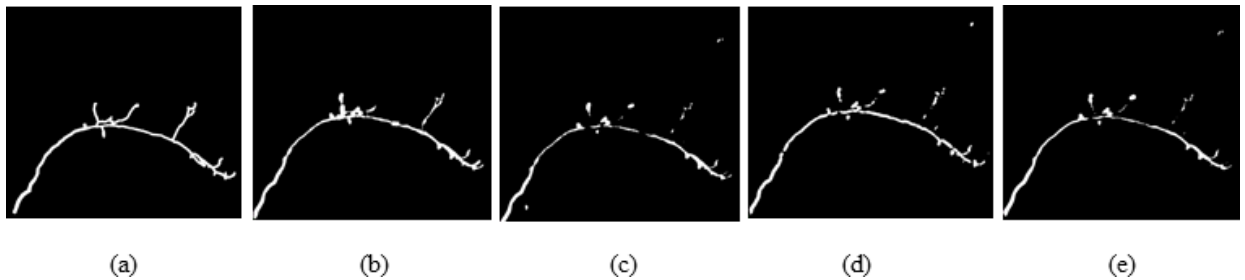


Fig. 13: Performance of 3D neuronal dataset. (a) Groundtruth annotation which has been expanded for visualisation, (b) Our method based on [8], (c) OOF [8], (d) Eigen [3]. (e) LEFD [23].