# Learning to Boost Filamentary Structure Segmentation

Lin Gu,   Li Cheng

Bioinformatics Institute, A*STAR, Singapore

gulin,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 base segmenter arbitrarily chosen from a number of existing segmenters: We start with an initial partial segmentation where the filamentary structure obtained is 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 specific filamentary structure segmentation tasks: retinal blood vessel segmentation as well as neuronal segmentations, where noticeable improvement has been shown over the original state-of-the-arts.*

## 1. Introduction

This work aims to address the problem of image-based filamentary structure segmentation. In particular, we focus on the challenging issue of preserving weak foreground signals, i.e., small and thin filaments from ambiguous back-grounds. This problem is fundamental in a rather broad range of applications such as neuronal tracing from microscopic images [17], retinal blood vessel tracing in retinal scans [9], as well as reconstruction of human vasculature such as 2D digital subtraction angiography and 3D magnetic resonance angiography [10]. 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 [1].

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 [27], to detect filament edges [2], or to combine with the eigenvalues [10] 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.* [29] 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) [13], is based on the assumption of circular filament cross-sections, which is further extended in Turetken *et al.* [26] to segment filamentary structures through a set of regularly spaced anchor points. Learning-based methods [22, 21, 25, 6], on the other hand, advocate the automation of the feature learning process. For example, the method of [6] 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 and 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 meth-
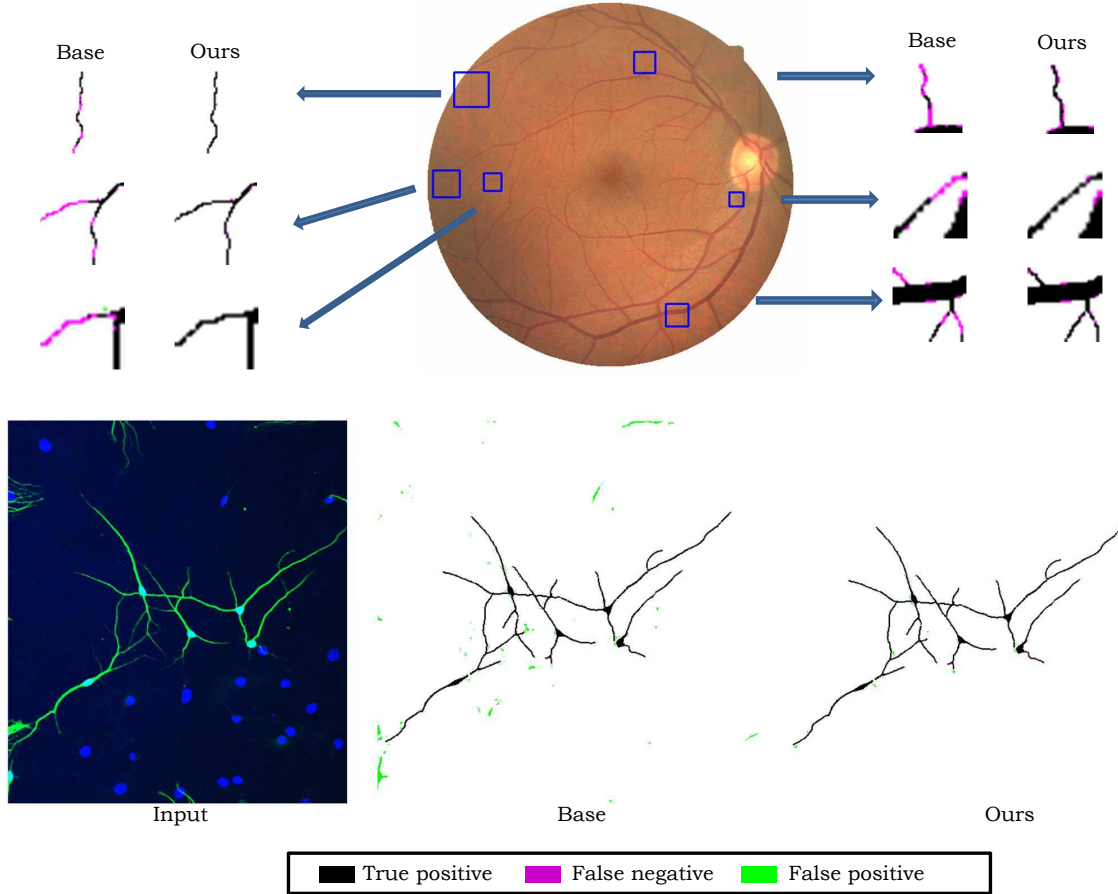
Figure 1: Exemplar results of our approach on two applications: (first row) retinal and (second row) neuronal images. For each application, the results of the base segmenter [6] and results of our boosted are displayed. Note the blue channel in the neuronal image indicate the DAPI-tagged nuclei which can be ignored in our context. In each of the results, Black pixels refer to true positive, magenta pixels are for false negative (miss), while lime pixels are for false positive (false alarm).
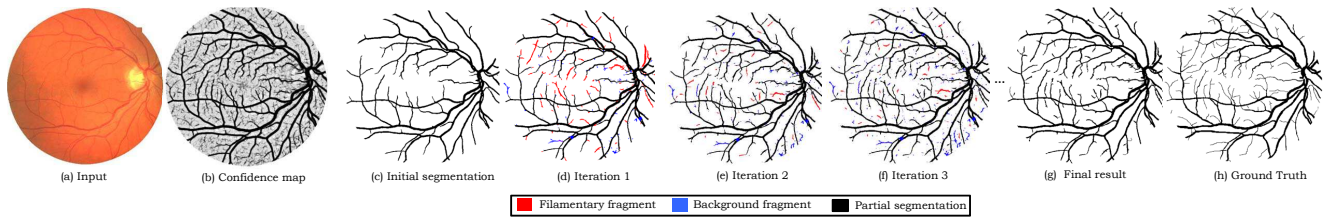


Figure 2: An illustration of the pipeline of our approach on a retinal image. From top-left to bottom-right: (a) Input image; (b) The confidence map from a base segmenter. Here pixels with lower intensity values correspond to higher confidence scores. (c) Initial partial segmentation obtained by applying a sufficiently high threshold in the confidence map. (d-f) Intermediate results of iterations 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.

ods. 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 [10] and tubulary score [26, 25]. 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 and thick filaments). Similarly, when placing a sufficiently low threshold, many of these methods are able to produce 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. As shown in Figs. 1, empirically our approach is demonstrated to boost the performance of a wide spectrum of exemplar existing methods such as [2, 13, 6], which contains supervised and unsupervised methods, as well as Hessian-based, model-based, and learning-based methods.

## 2. Our Approach

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.2, 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.2(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.2(c)) via a latent classification tree model. The latent classification tree (LCT) 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.
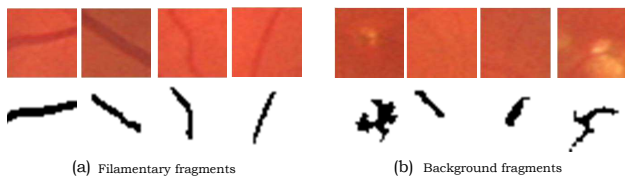
Figure 3: (a) and (b) present exemplar filamentary fragments vs. background fragments encountered during step one, respectively where the first row shows the input image patch and the second row shows the fragments.

### 2.1. Preprocessing

For an input image, its confidence map is obtained by applying a base segmenter. By placing a threshold $\tau_h$, we obtain an initial partial segmentation that comprises a set of pixels with confidence values exceeding $\tau_h$. Note $\tau_h$ is sufficiently high to ensure that only filamentary foreground pixels are selected. In a similar manner, we can also define another threshold $\tau_l$ to be sufficiently low for pure backgrounds. At this moment, a partial foreground segmentation is obtained, while the rest pixels remain undecided. We also define a scanning horizon as an epsilon ball $B$ around the current partial segmentation with a searching radius $\epsilon$.

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

As displayed in Fig.2(c), the long and 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, the aim of step one is to detect filamentary fragments from backgrounds in those pixels with weak confidence values in the range of $(\tau_l, \tau_h)$ within horizon $B$. 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 this range will inevitably result in the detection of fragment candidates which are a mixture of filamentary fragments (coloured red) and ambiguous background fragments (coloured blue) as illustrated in e.g. Fig.2(d-f). 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 possessing similar orientations, shapes and textures.

#### 2.2.1 Identify Distinct Local Filament Shapes and Textures by Filament Tokens

Following the idea of sketch tokens [14], we will identify distinct local filamentary fragments that are referred to as *filament tokens*. It works by grouping similar filamentary fragments in terms of their shapes, textures, and orientations
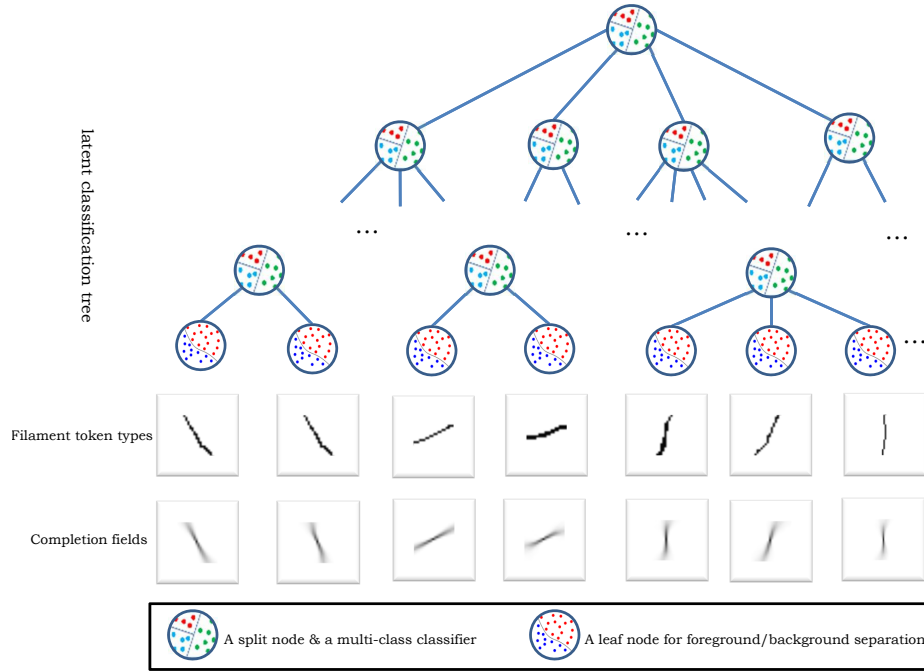
**Figure 4:** An illustration of the latent classification tree (LCT) model in step one: A latent split node which comprises a subset of similar tokens and is split into multi-branches according to a multi-class classifier; A leaf node corresponds to a local foreground/background separation problem, 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 forgeround/background separation problem.

in image patches 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 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 images are fixed to $31 \times 31$. $T$ is set to 180.

### 2.2.2 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 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 topol-

ogy structure within the token types. This inspires us to consider the latent tree models [8, 19], which in our context is a tree-structured probabilistic organization of these *local* figure/background separation scenarios as being the leaf nodes of Fig. 4, 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 [8]: 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 aforementioned HOG features describing the two token types.

Fig.4 illustrates an exemplar CLNJ tree, where each split 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 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.
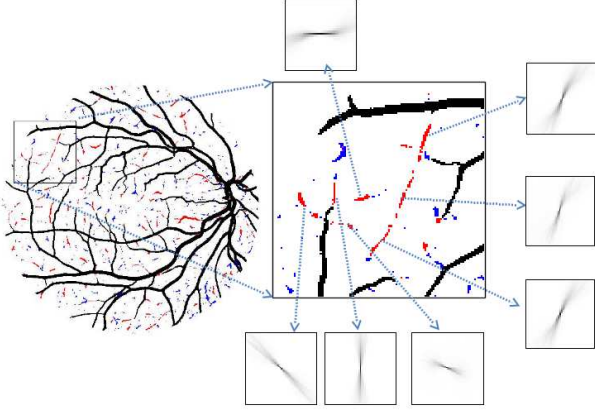


Figure 5: An example of completion fields of detected filamentary fragments.



Confidence map          Completion fields

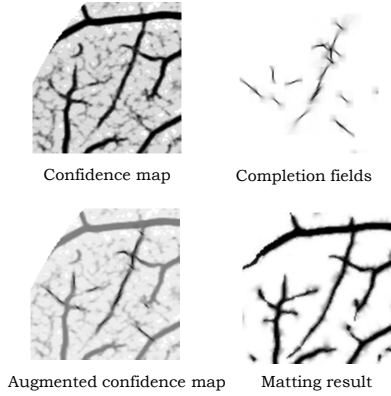Augmented confidence map          Matting result

Figure 6: An example of restoring the missing connections between the detected filamentary fragments and main trunk using the matting technique.

### 2.2.3 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 these classifiers, namely the filamentary-structure features and the low-level features, as below.

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.2(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 of the 2D area 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 inter-connecter. 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 the objective function of $\sum_{\mathbf{x}' \in \mathcal{S}_{s,b}} - \log(c(\mathbf{x}'))$. 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 images, the Kernel Boost Features [6] are utilized as follows. The Kernel Boost method is firstly trained on the set of filamentary structured 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. Similar to the filament token parameters discussed previously, the patch size for 2D images are fixed to $31 \times 31$.

### 2.3. 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.

### 2.3.1 Augmented Confidence Map by the Completion Fields

The concept of completion fields is inspired by the work of [28] 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 tex-

tures, 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 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. 5. 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. 6. 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 $\bar{c}(\mathbf{x})$.

### 2.3.2 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 $\tau_l$). 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 $\alpha$ values of the unknown pixels as a composition of foreground and background layers, i.e. $I(\mathbf{x}) = \alpha(\mathbf{x})F(\mathbf{x}) + (1 - \alpha(\mathbf{x}))B(\mathbf{x})$, where $\mathbf{x}$ indexes a particular 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 $\delta$ to categorize into foreground and background pixels as those with $\alpha$ 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 [7], where the $\alpha$ values are computed as $\alpha = (L_c + \lambda M)^{-1}(\lambda \mathbf{v})$. Let $n$ denote the number of pixels in an 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 $j$, $D = [d_{ij}]$ is the diagonal degree matrix of $K$, $d_{ii} = \sum_j k_{ij}$. $\lambda > 0$ is a scalar constant. 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.
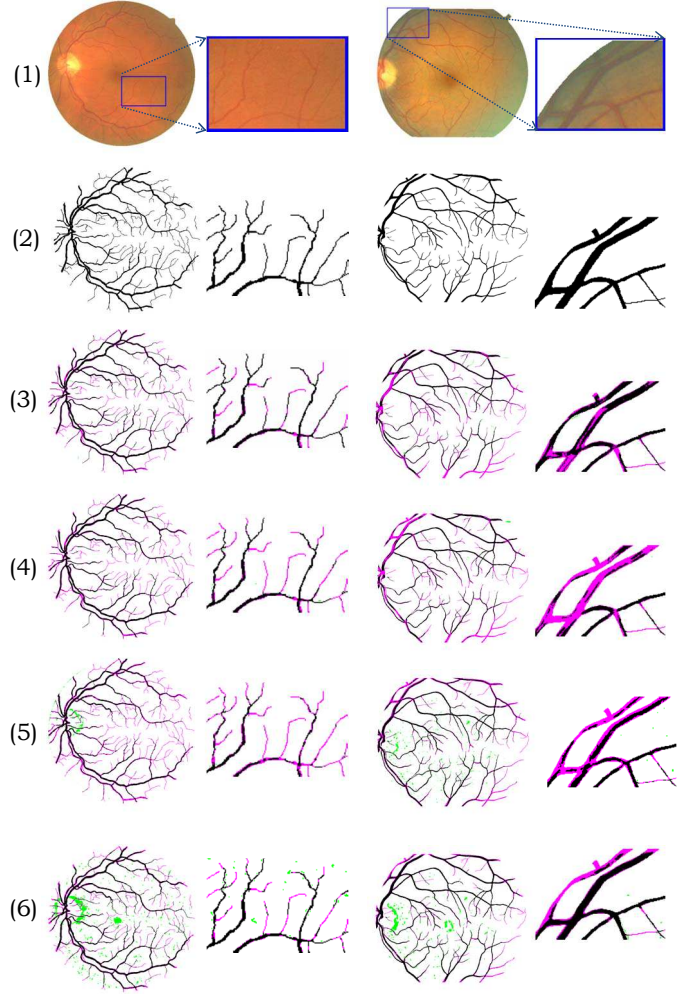


Figure 7: Performance on retinal datasets of DRIVE and STARE datasets. The first two columns correspond to DRIVE and the last two correspond to STARE. For each dataset, first column shows the full image while the second column presents a detailed zoom-in view of a cropped region. From top-down, rows are in the following order: (1) Input image. (2) Ground-truth annotation. (3) our method based on [6]. (4) Kernel Boost [6]. (5) OOF [13]. (6) IUWT [2]. True positive is denoted in black, false positive in cyan, and false negative in green.

Table 1: Performance on the two retinal testbed 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 [6] | OOF [13] | IUWT [2] |
|---|---|---|---|
| Base | 78.77 | 72.86 | 72.20 |
| Ours-nCnM | 79.61 | 73.01 | 73.99 |
| Ours-CnM | 79.96 | 73.10 | 74.17 |
| Ours | 80.92 | 73.25 | 74.28 |
| % gain | 2.15 | 0.39 | 2.08 |

Table 2: Accuracy comparison of methods on retinal testbeds.

|  | DRIVE | | | STARE | | |
|---|---|---|---|---|---|---|
|  | Prec | Recall | Acc | Prec | Recall | Acc |
| Ours based on [6] | 79.57 | **79.47** | **97.32** | 83.59 | 73.72 | **97.72** |
| Kernel Boost [6] | 80.31 | 77.58 | 97.19 | **84.54** | 72.29 | 97.64 |
| N4-fields [11] | **81.57** | 78.39 | 96.59 | - | - | - |
| OOF [13] | 73.12 | 65.43 | 95.84 | 73.03 | 68.09 | 94.68 |
| Soares [22] | 69.43 | 74.25 | 94.42 | 75.68 | 74.40 | 94.80 |
| Bankhead [2] | 70.27 | 71.77 | 93.71 | 67.46 | **78.97** | 94.59 |
| Ricci [20] | - | - | 95.63 | - | - | 95.84 |
| Marin [15] | - | 70.67 | 94.52 | - | 69.44 | 95.26 |
| Mendonca [18] | - | 73.44 | 94.52 | - | 69.96 | 94.40 |
| Martinez-Perez [16] | - | 72.46 | 93.44 | - | 75.06 | 94.10 |

# 3. 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 datasets as follows.

## 3.1. Datasets

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

**Two Retinal Vessel Datasets: DRIVE and STARE** For retinal blood vessel tracing, the publicly available testbeds of DRIVE [23] and STARE [12] 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 Neuronal Microscopic Images** To facilitate the analysis of 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. It contains 112 images of mouse embryonic neural stem cells with annotated ground-truths, where one third of the images are reserved for training. Meanwhile, the rest two third images containing in total 675 neurons are retained for test evaluation purpose.

## 3.2. Evaluation Details and Analysis

**Base Segmenters** The following six segmentation methods are considered as the base segmenters in this paper: (1) Kernel Boost [6] which learn discriminative convolutional filters based on the Gradient Boosting framework. (2) Optimally Oriented Flux (OOF) [13], an unsupervised method with manually designed filter, widely used for delineating tubular structures. (3) IUWT [2], an unsupervised method based on isotropic undecimated wavelet transform. (4) Eigen [10], a multiscale Hessian-based method. (5) T2T [3], a supervised method based on pixel classification, medial sub-tree generation, and global tree linking components. (6) LEFD [4] which is an unsupervised model-based method. Due to the capacity of individual methods, most existing methods work only for one of the possible application scenarios, with the notably exception of OOF [13]. On the other hand, unfortunately OOF is usually not the best performer.

**Parameters and Performance Evaluation Criteria** The internal parameters in our approach are set to the following values in this paper. thresholds $\tau_h = 0.9$, $\tau_m = 0.7$, $\tau_l = 0.3$, $\epsilon = 5$, $\delta = 0.8$, 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 number of 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 [5, 24], 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 $r = 1.5$. Following [24], 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.

## 3.3. Retinal Blood Vessels Segmentation

Table 1 summarizes the quantitative evaluations of our approach when starting with various base segmenters on the

STARE retinal testbed, 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. This is in fact quite noticeable, given the huge amount of existing research efforts on these two datasets. As shown in Table 1, 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 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 2 presents a direct comparison with a number of additional state-of-the-art retinal segmentation methods. Here the boosted results of our approach based on [6] is regarded as our results, as it produces the best results among all base segmenters during Table 1. Note the performance evaluation criteria in Table 2 is switched to the standard accuracy metric, with the aim of best utilizing existing results as reported in literature. Here our results give the best accuracy over both datasets and all comparison methods. Our results also maintain a nice balance of precision and recall, while others are less well-balanced. For example, Bankhead [2] gives best recall on STARE with however a very poor precision. 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 whey 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. 7, overall our approach is able to recover more challenging filamentary fragments that are ignored by other methods including the base method of [6]. We also evaluate the robustness of our approach. As presented in Figure 8, our approach is quite insensitive w.r.t. varying the internal parameters K and T.
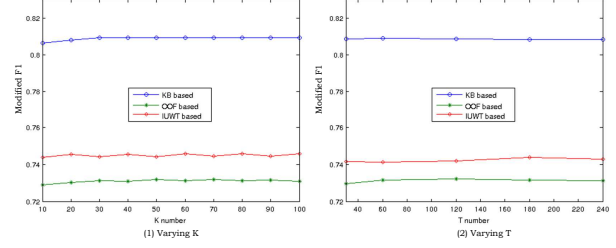


**Figure 8:** performance deviation with varying K and T values on the retinal testbed of STARE. The modified F1 measure (%) is used.

**Table 3:** Performance on 2D Neuronal dataset with different base methods using modified F1 measure.

|        | Kernel Boost [6] | OOF [13] | Eigen [10] | T2T [3] |
|--------|------------------|----------|------------|---------|
| Base   | 84.74            | 63.50    | 63.94      | 66.49   |
| Ours   | 86.80            | 66.03    | 66.85      | 66.91   |
| % gain | 2.06             | 2.53     | 2.91       | 0.42    |

### 3.4. Neuronal Segmentation

Table 3 shows that on average our approach produces around 2% performance gain in term of the modified F1 measure. Similar to the retinal datasets, our approach is able to boost the performance over different base methods ranging from unsupervised [13, 10] to supervised such as [6, 3]. Visually our result is often noticeably better than that of existing methods, as exemplified in Fig. 9.
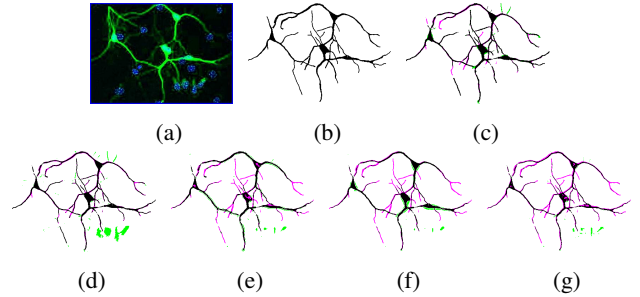


**Figure 9:** Performance of 2D neuronal dataset. (a) Input image where the contrast has been increased for visual presentation. (b) Ground-truth annotation. (c) Our method based on [6]. (d) Kernel Boost [6], (e) OOF [13], (f) Eigen [10]. (g) T2T [3]. True positive is denoted in black, false positive in cyan and false negative in green.

## 4. 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 achieve 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.

## Acknowledgements

## References

[1] M. Abramoff, M. Niemeijer, M. Viergever, and B. Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imag.*, 23(4):501–509, 2004.

[2] P. Bankhead, C. Scholfield, J. McGeown, and T. Curtis. Fast retinal vessel detection and measurement using wavelets and edge location refinement. *PLoS ONE*, 2012.

[3] S. Basu, A. Aksel, B. Condron, and S. Acton. Tree2Tree: Neuron segmentation for generation of neuronal morphology. In *ISBI*, 2010.

[4] S. Basu, C. Liu, and G. Rohde. Localizing and extracting filament distributions from microscopy images. *Journal of Microscopy*, 258(1):13–23, 2015.

[5] C. Becker, K. Ali, G. Knott, and P. Fua. Learning context cues for synapse segmentation. *IEEE Trans. Med. Imag.*, 32(10):1864–77, 2013.

[6] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua. Supervised feature learning for curvilinear structure segmentation. In *MICCAI*, 2013.

[7] Q. Chen, D. Li, and C. Tang. KNN matting. In *CVPR*, 2012.

[8] M. Choi, V. Tan, A. Anandkumar, and A. Willsky. Learning latent tree graphical models. *JMLR*, 12:1771–1812, 2011.

[9] J. De, H. Li, and L. Cheng. Tracing retinal vessel trees by transductive inference. *BMC Bioinformatics*, 15(20):1–20, 2014.

[10] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale vessel enhancement filtering. In *MICCAI*. 1998.

[11] Y. Ganin and V. Lempitsky. N4-fields: Neural network nearest neighbor fields for image transforms. In *ACCV*, pages 536–51. 2014.

[12] 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*, 19(3):203–10, 2000.

[13] M. Law and A. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *ECCV*. 2008.

[14] J. Lim, C. Zitnick, and P. Dollar. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013.

[15] 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.*, 30(1):146–58, 2011.

[16] 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.*, pages 47–61, 2007.

[17] E. Meijering. Neuron tracing in perspective. *Cytometry A.*, 77(7):693–704, 2010.

[18] A. Mendonca and A. Campilho. Segmentation of retinal blood vessels by combining the detection of centerlines and morphological reconstruction. *IEEE Trans. Med. Imag.*, 25(9):1200–13, 2006.

[19] R. Mourad, C. Sinoquet, N. Zhang, T. Liu, and P. Leray. A survey on latent tree models and applications. *JAIR*, 47:157–203, 2013.

[20] E. Ricci and R. Perfetti. Retinal blood vessel segmentation using line operators and support vector classification. *IEEE. Trans. Med. Imag.*, 26(10):1357–65, 2007.

[21] R. Rigamonti and V. Lepetit. Accurate and efficient linear structure segmentation by leveraging ad hoc features. In *MICCAI*, 2012.

[22] 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.*, 25(9):1214–22, 2006.

[23] 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.*, 23(4):501–09, 2004.

[24] 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.

[25] E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. Reconstructing loopy curvilinear structures using integer programming. In *CVPR*, 2014.

[26] E. Turetken, G. Gonzalez, C. Blum, and P. Fua. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2-3):279–302, 2011.

[27] Y. Wang, A. Narayanaswamy, C. Tsai, and B. Roysam. A broadly applicable 3-d neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3):193–217, 2011.

[28] L. Williams and D. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *neural computation*, 9(4):837–58, 2006.

[29] 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*, 9(2-3):247–261, 2011.