Active Learning and Proofreading for Delineation of Curvilinear Structures

Agata Mosinska*, Jakub Tarnawski**, and Pascal Fua

École Polytechnique Fédérale de Lausanne

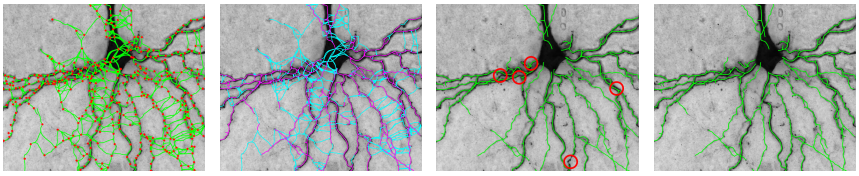
Abstract. Many state-of-the-art delineation methods rely on supervised machine learning algorithms. As a result, they require manually annotated training data, which is tedious to obtain. Furthermore, even minor classification errors may significantly affect the topology of the final result. In this paper we propose a generic approach to addressing both of these problems by taking into account the influence of a potential misclassification on the resulting delineation. In an Active Learning context, we identify parts of linear structures that should be annotated first in order to train a classifier effectively. In a proofreading context, we similarly find regions of the resulting reconstruction that should be verified in priority to obtain a nearly-perfect result. In both cases, by focusing the attention of the human expert on potential classification mistakes which are the most critical parts of the delineation, we reduce the amount of required supervision. We demonstrate the effectiveness of our approach on microscopy images depicting blood vessels and neurons.

Keywords: Active Learning, Proofreading, Delineation, Light Microscopy, Mixed Integer Programming

1 Introduction

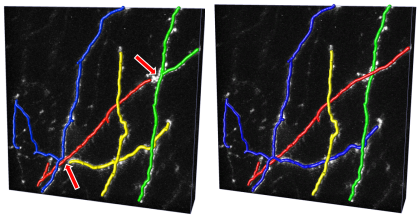
Complex and extensive curvilinear structures include blood vessels, pulmonary bronchi, nerve fibers and neuronal networks among others. Many state-of-the-art approaches to automatically delineating them rely on supervised Machine Learning techniques. For training purposes, they require *annotated* ground-truth data in large quantities to cover a wide range of potential variations due to imaging artifacts and changes in acquisition protocols. For optimal performance, these variations must be featured in the training data, as they can produce drastic changes in appearance. Furthermore, no matter how well-trained the algorithms are, they will continue to make mistakes, which must be caught by the user and corrected. This is known as *proofreading* – a slow, tedious and expensive process when large amounts of image data or 3D image stacks are involved, to the point that it is considered as a major bottleneck for applications such as neuron reconstruction [11].

* Supported by the Swiss National Science Foundation.
** Supported by ERC Starting Grant 335288-OptApprox.



(a) (b) (c) (d)

Fig. 1: Delineation workflow. (a) Input image with overcomplete graph overlaid. (b) The high-probability edges are shown in purple and the others in cyan. (c) Automated delineation, with connectivity errors highlighted by red circles. (d) Final result after proofreading. All figures are best viewed in color.



(a) (b)

Fig. 2: Misclassifying even a few edges may severely impact the final topology. (a) The two edges indicated by the red arrows are falsely labeled as negatives. As a result, two pairs of unrelated branches (green and yellow) are merged. (b) The true connectivity is recovered after correcting the two edges.

In other words, human intervention is required both to create training data before running the delineation algorithm and to correct its output thereafter. Current approaches to making this less tedious focus on providing better visualization and editing tools [4,11]. While undoubtedly useful, this is not enough. We therefore propose an Active Learning (AL) [13] approach to direct the annotator's attention to the most critical samples. It takes into account the expected change in reconstruction that can result from labeling specific paths. It can be used both for fast annotation purposes and, later, to detect potential mistakes in machine-generated delineations.

More specifically, consider an algorithm such as those of [12,17,16,9,10], whose workflow is depicted by Fig. 1. It first builds a graph whose nodes are points likely to lie on the linear structures and whose edges represent paths connecting them. Then it assigns a weight to each edge based on the output of a discriminative classifier. Since the result is critically dependent on the weights, it is important that the classifier is trained well. Finally, the reconstruction algorithm finds a subgraph that maximizes an objective (cost) function dependent on the edge weights, subject to certain constraints. However, even very small mistakes can result in very different delineations, as shown in Fig. 2.

Our main insight is that the decision about which edges to annotate or proofread should be based on their influence on the cost of the network. Earlier methods either ignore the network topology altogether [5] or only take it into consideration locally [8], whereas we consider it globally. Our contribution is therefore

a cost- and topology-based criterion for detecting attention-worthy edges. We demonstrate that this can be used for both AL and proofreading, allowing us to drastically reduce the required amount of human intervention when used in conjunction with the algorithm of [16]. To make it practical for interactive applications, we also reformulate the latter to speed it up considerably – it runs nearly in real-time and it can handle much larger graphs than [16].

2 Attention Mechanism

2.1 Graph-Based Delineation

Delineation algorithms usually start by computing a tubularity measure [7,15,14], which quantifies the likelihood that a tubular structure is present at a given image location. Next, they extract either high-tubularity superpixels likely to be tubular structure fragments [12,17] or longer paths connecting points likely to be on the centerline of such structures [6,3,9,16]. Each superpixel or path is treated as an edge e_i of an over-complete spatial graph $\mathcal{G}$ (see Fig. 1(a)) and is characterized by an image-based feature vector $\mathbf{x}_i$. Let $\mathcal{E}$ be the set of all such edges, which is expected to be a superset of the set $\mathcal{R}$ of edges defining the true curvilinear structure, as shown in Fig. 1(d). If the events of each edge e_i being present in the reconstruction are assumed to be independent (conditional on the image evidence $\mathbf{x}_i$), then the most likely subset $\mathcal{R}^*$ is the one minimizing

$$c(\mathcal{R}) = \sum_{e_i \in \mathcal{R}} w_i \quad \text{with} \quad w_i = -\log \frac{p(y_i = 1|\mathbf{x}_i)}{p(y_i = 0|\mathbf{x}_i)} \,, \tag{1}$$

where $w_i \in \mathcal{R}$ is the *weight* assigned to edge e_i and y_i is a binary class label denoting whether e_i belongs to the final reconstruction or not. This optimization is subject to certain geometric constraints; for example, a state-of-the-art method presented in [16] solves a more complex Mixed Integer Program (**MIP**), which uses linear constraints to force the reconstruction to form a connected network (or a tree). We were able to reformulate the original optimization scheme and obtain major speedups which make it practical even when delineations must be recomputed often. **MIP** yields better results than using a more basic method Minimum Spanning Tree with Pruning [6], while also being able to handle non-tree networks. Let us remark that finding the minimizing $\mathcal{R}$ is trivial to parallelize.

The probabilities appearing in Eq. 1 can be estimated in many ways. A simple and effective one is to train a discriminative classifier for this purpose [3,17,16]. However, the performance critically depends on how well-trained the classifier is. A few misclassified edges can produce drastic topology changes, affecting the entire reconstruction, as shown in Fig. 2. In this paper we address both issues with a single generic criterion.

2.2 Error Detection

The key to both fast proofreading and efficient AL is to quickly find potential mistakes, especially those that are critical for the topology. In this work, we take *critical mistakes* to mean erroneous edge weights w_i that result in major changes to the cost $c(\mathcal{R}^*, \mathbf{W})$ of the reconstruction. In other words, if changing a specific weight can significantly influence the delineation, we must ensure that the weight is correct. We therefore measure this influence, alter the edge weights accordingly, and recompute the delineation.

Delineation-Change Metric We denote by $\mathcal{R}^*$ the edge subset minimizing the objective (cost) function $c(\mathcal{R}, \mathbf{W}) = \sum_{e_i \in \mathcal{R}} w_i$ given a particular assignment $\mathbf{W}$ of weights to edges in $\mathcal{G}$. Changing the weight w_i of an edge e_i to w_i' will lead to a new graph with an optimal edge subset $\mathcal{R}_i'$. We can thus define a delineation-change metric, which evaluates the cost of changing the weight of an edge $e_i \in \mathcal{E}$:

$$\Delta c_i = c(\mathcal{R}^*, \mathbf{W}) - c(\mathcal{R}_i', \mathbf{W}') \; . \tag{2}$$

If $\Delta c_i > 0$, the cost has decreased; we can conjecture that the overall reconstruction benefits from this weight change and therefore the weight value may be worth investigating by the annotator as a potential mistake. The converse is true if $\Delta c_i < 0$. In other words, this very simple metric gives us a way to gauge the influence of an edge weight on the overall reconstruction.

Changing the Weights For our cost change criterion to have practical value, we must alter weights in such a way that Δc_i is largest for edges which require the opinion of an annotator. To achieve this, we must not only flip the sign of the weight (which would imitate the classifier assigning it to the opposite class), but also increase the absolute value of likely mistakes. Without this, many of the mistakes with $|w_i| \approx 0$ could be omitted due to smaller values of Δc_i compared to edges with larger $|w_i|$, which are much less likely to be mistakes.

The above requirements can be satisfied with the following transformation:

$$w_i' = \begin{cases} A + w_i & \text{if } w_i > 0, \\ B + w_i & \text{if } w_i < 0. \end{cases} \tag{3}$$

Intuitively, the proposed transformation corresponds to swapping the probability distributions from which the weights of positive and negative edges come. It is equivalent to ranking every edge according to how much the current weight must be increased (decreased) to remove (add) it from the solution.

We take A and B to be the 10% and 90% quantiles of the weight distribution (for robustness to outliers). These are near-extreme values of the weights for the positive and negative classes respectively, which we use as attractors for w_i': for small positive w_i we want w_i' to be close to A, and for negative ones to B instead. The weight change is therefore likely to yield a significant Δc_i for probable mistakes.

Finally, for edges whose weight is negative but which nevertheless do not belong to the graph, we take Δc_i to be w'_i to ensure that it is positive and that more uncertain edges are assigned higher Δc_i.

3 Active Learning and Proofreading

AL aims to train a model with minimal user input by selecting small subsets of examples that are the most informative. Formally, our algorithm starts with a small set of labeled edges S_0. We then repeat the following steps: At iteration t, we use the annotated set of edges S_{t-1} to train classifier C_{t-1} and select one or more edges to be labeled by the user and added to S_{t-1} to form S_t. The edge(s) we select are those that maximize the criterion Δc of Eq. 2.

By contrast, proofreading occurs *after* the classifier has been trained and a complete delineation has been produced. At this point, the main concern is not to further improve the classifier, but simply to correct potential mistakes. Therefore, the most crucial edges are those that are misclassified and whose presence or absence affects the topology of the delineation the most. To find them, we again compute the Δc value for each edge. However, some edges could have a high Δc because they are misclassified, even though they do not influence the topology of the final delineation.

To focus on potential mistakes that do affect the topology strongly, we rely on the DIADEM score [1], which captures the topological differences between trees, such as connectivity changes and missing or spurious branches. It ranges from 0 to 1; the larger the score, the more similar the two trees are. More specifically, let $\mathcal{R}^*$ be the optimal tree given the edge weights, and let $\mathcal{R}'_i$ be the tree we obtain when changing the weight of edge e_i from w_i to w'_i, as described in Section 2.2. To measure the importance of each edge, we compute the score

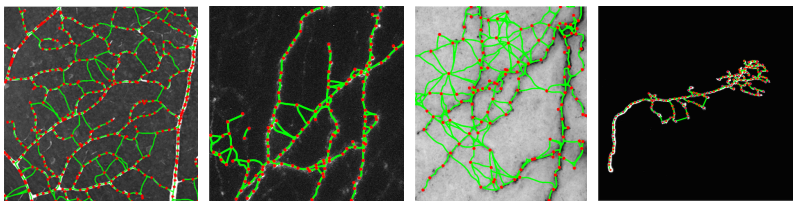
$$s_i = \frac{\Delta c_i}{\text{DIADEM}(\mathcal{R}^*, \mathcal{R}'_i)} \tag{4}$$

and ask the user to check the highest-scoring one. The edge is assigned a weight equal to A or B from Section 2.2 according to the user's response. We then recompute $\mathcal{R}^*$ and repeat the process. Traditional proofreading approaches require the user to visually inspect the whole image. By contrast, our user only has to give an opinion about one edge at a time, which is automatically selected and presented to them.

4 Results

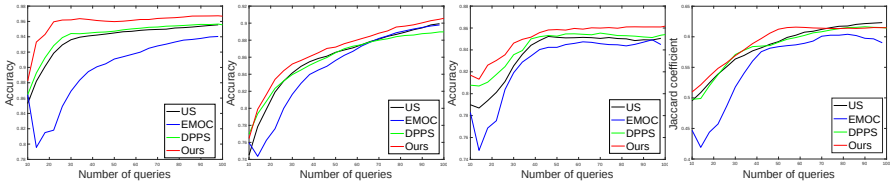
We tested our approach on 3-D image stacks depicting retinal blood vessels, rat brain axons and dendrites, and drosophila olfactory projection fibers [1] obtained using either 2-photon or brighfield microscopes, shown in Fig. 3.

We rely on the algorithm of [16] for the initial overcomplete graphs, the corresponding edge features and the final delineations. To classify edges as being likely to be part of an extended linear structure or not on the basis of local image evidence, we use Gradient Boosted Decision Trees [2].



(a) (b) (c) (d)

Fig. 3: Dataset images with the over-complete graphs overlaid. (a) *Blood Vessels.* (b) *Axons.* (c) *Brightfield Neurons.* (d) *Olfactory Projection Fibers.*



(a) (b) (c) (d)

Fig. 4: Active Learning. Accuracy as a function of the number of annotated samples. (a) *Blood vessels.* (b) *Axons.* (c) *Brightfield neurons.* (d) *Olfactory Projection Fibers.* The red curve denoting our approach is always above the others, except in the right-hand side of (d): because this is a comparatively easy case, the delineation stops changing after some time and error-based queries are no longer informative.

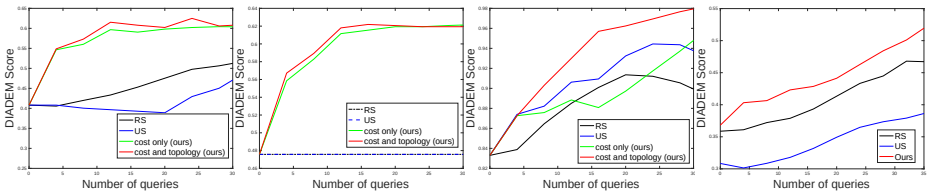
4.1 Active Learning

For each image, we start with an overcomplete graph. The initial classifier is trained using 10 randomly sampled examples. Then, we query four edges at a time, as discussed in Section 3, which allows us to update the classifier often enough while decreasing the computational cost. We report results averaged over 30 trials in Fig. 4. Our approach outperforms both naive methods such as Uncertainty Sampling (**US**) and more sophisticated recent ones such as **DPPS** [8] and **EMOC** [5]. **DPPS** is designed specifically for delineation and also relies on uncertainty sampling, but only takes local topology into account when evaluating this uncertainty. **EMOC** is a more generic method that aims at selecting samples that have the greatest potential to change the output.

4.2 Proofreading

For each test image, we compute an overcomplete graph and classify its edges using a classifier trained on 20000 samples. We then find four edges with the highest values of the score s_i of Eq. 4 and present them to the user for verification. Their feedback is then used to update the delineation.

The red curves of Fig. 5(a-c) depict the increase in DIADEM score. Rapid improvement can be seen after as few as 15 corrections. Fig. 6 shows how the



(a) (b) (c) (d)

Fig. 5: Focused proofreading. DIADEM score as a function of the number of paths examined by the annotator. (a) *Axons*. (b) *Brightfield Neuron*. (c) *Olfactory Projection Fibers*. (d) Combined AL and proofreading for *Axons*.

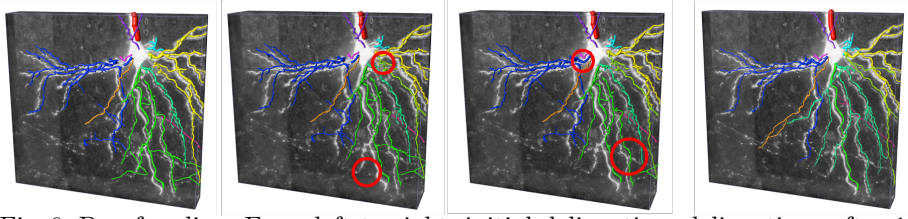


Fig. 6: Proofreading. From left to right: initial delineation, delineations after 10 and 20 corrections, and ground truth.

reconstruction evolves in a specific case. For analysis purposes, we also reran the experiment using the Δc criterion of Eq. 2 (cost-only) instead of the more sophisticated one of Eq. 4 (cost and topology) to choose the paths to be examined. This approach performs worse, particularly in the case of Fig. 5(c), because the highest-scoring mistakes are often the ones that tend to be in the **MIP** reconstruction both before and after correcting mistakes. It is therefore only by combining both cost and topology that we increase the chances that a potential correction of the selected edge will improve the reconstruction. By contrast, paths chosen by **RS** and **US** are not necessarily erroneous or in the neighborhood of the tree and investigating them often does not give any improvements.

4.3 Complete Pipeline

In a working system, we would integrate AL and proofreading into a single pipeline. To gauge its potential efficiency, we selected 50 edges to train our classifier using the AL strategy of Section 3. We then computed a delineation in a test image and proofread it by selecting 35 edges. For comparison purposes, we used either our approach as described in Section 3, **RS**, or **US** to pick the edges for training and then for verification. In Fig. 5(d) we plot the performance as a function of the total number of edges the user needed to label manually.

5 Conclusions

We present an attention scheme that significantly reduces the annotation effort involved both in creating training data for supervised Machine Learning and in

proofreading results for delineation. It does so by detecting possibly misclassified samples and considering their influence on the topology of the reconstruction. Our method outperforms baselines on a variety of microscopy images and can be used in interactive applications thanks to its efficient formulation.

References

1. Ascoli, G., Svoboda, K., Liu, Y.: Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge (2010), `http://diademchallenge.org/`
2. Becker, C., Rigamonti, R., Lepetit, V., Fua, P.: Supervised Feature Learning for Curvilinear Structure Segmentation. In: MICCAI (September 2013)
3. Breitenreicher, D., Sofka, M., Britzen, S., Zhou, S.: Hierarchical Discriminative Framework for Detecting Tubular Structures in 3D Images. In: International Conference on Information Processing in Medical Imaging (2013)
4. Dercksen, V., Hege, H., Oberlaender, M.: The Filament Editor: An Interactive Software Environment for Visualization, Proof-Editing and Analysis of 3D Neuron Morphology. Neuroinformatics 12, 325–339 (2014)
5. Freytag, A., Rodner, E., Denzler, J.: Selecting Influential Examples: Active Learning with Expected Model Output Changes (2014)
6. Gonzalez, G., Fleuret, F., Fua, P.: Automated Delineation of Dendritic Networks in Noisy Image Stacks. In: ECCV. pp. 214–227 (October 2008)
7. Law, M., Chung, A.: Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In: ECCV (2008)
8. Mosinska, A., Sznitman, R., Glowacki, P., Fua, P.: Active Learning for Delineation of Curvilinear Structures. In: CVPR (2016)
9. Neher, P.F., Götz, M., Norajitra, T., Weber, C., Maier-Hein, K.H.: A Machine Learning Based Approach to Fiber Tractography Using Classifier Voting. In: Medical Image Computing and Computer-Assisted Intervention - MICCAI. pp. 45–52 (2015)
10. Peng, H., Long, F., Myers, G.: Automatic 3D Neuron Tracing Using All-Path Pruning 27(13), 239–247 (2011)
11. Peng, H., Long, F., Zhao, T., Myers, E.: Proof-Editing is the Bottleneck of 3D Neuron Reconstruction: the Problem and Solutions. Neur. Inf. 9(2), 103–105 (2011)
12. Santamaría-Pang, A., Hernandez-Herrera, P., Papadakis, M., Saggau, P., Kakadiaris, I.: Automatic Morphological Reconstruction of Neurons from Multiphoton and Confocal Microscopy Images Using 3D Tubular Models. Neur. Inf. pp. 1–24 (2015)
13. Settles, B.: Active Learning Literature Survey. Tech. rep., University of Wisconsin–Madison (2010)
14. Sironi, A., Turetken, E., Lepetit, V., Fua, P.: Multiscale Centerline Detection. PAMI (2016)
15. Turetken, E., Becker, C., Glowacki, P., Benmansour, F., Fua, P.: Detecting Irregular Curvilinear Structures in Gray Scale and Color Imagery Using Multi-Directional Oriented Flux. In: ICCV (December 2013)
16. Turetken, E., Benmansour, F., Andres, B., Glowacki, P., Pfister, H., Fua, P.: Reconstructing Curvilinear Networks Using Path Classifiers and Integer Programming. PAMI (2016)
17. Zegarra, J.A.M., Wegner, J.D., Ladicky, L., Schindler, K.: Mind the Gap: Modeling Local and Global Context in (Road) Networks. In: Pattern Recognition - 36th German Conference, GCPR 2014, Münster (2014)