Efficient Classifier Training to Minimize False Merges in Electron Microscopy Segmentation

Toufiq Parag
HHMI Janelia Research Campus
Ashburn, VA
paragt@janelia.hhmi.org

Dan C. Ciresan, Alessandro Giusti
IDSIA, USI-SUPSI
Lugano, Switzerland
{dan, alessandrog}@idsia.ch

Abstract

The prospect of neural reconstruction from Electron Microscopy (EM) images has been elucidated by the automatic segmentation algorithms. Although segmentation algorithms eliminate the necessity of tracing the neurons by hand, significant manual effort is still essential for correcting the mistakes they make. A considerable amount of human labor is also required for annotating groundtruth volumes for training the classifiers of a segmentation framework. It is critically important to diminish the dependence on human interaction in the overall reconstruction system. This study proposes a novel classifier training algorithm for EM segmentation aimed to reduce the amount of manual effort demanded by the groundtruth annotation and error refinement tasks. Instead of using an exhaustive pixel level groundtruth, an active learning algorithm is proposed for sparse labeling of pixel and boundaries of superpixels. Because over-segmentation errors are in general more tolerable and easier to correct than the under-segmentation errors, our algorithm is designed to prioritize minimization of false-merges over false-split mistakes. Our experiments on both 2D and 3D data suggest that the proposed method yields segmentation outputs that are more amenable to neural reconstruction than those of existing methods.

1. Introduction

One important task for neural reconstruction from Electron Microscopy (EM) is to extract the anatomical structure of a neuron by accurately assigning regions of EM images to corresponding cells. Due to the size and number of EM images typically required for a useful dense reconstruction, it is impractical to manually perform such task. Recent studies on neural reconstructions or connectomics [30][12] apply automated segmentation algorithms for determining cell morphology. The result of such an automated segmentation algorithm is not free of errors, which is why a reconstruction approach must either manually correct the mistakes made by these algorithms [30], or conform them to a skeleton representation generated earlier by hand [12].

In addition, there have been many notable works addressing one or multiple processes constituting an overall segmentation algorithm. Existing algorithms such as[14][7][19] for pixel classification; [22][21] for effective generation of over-segmentation; [28][16][1] for isotropic 3D supervoxel clustering; [31][9] for co-segmentation for anisotropic data report impressive performances on different kinds of EM datasets. Many of these novel approaches are motivated by the methods in natural image segmentation and evaluate output accuracy using error measures popular in computer vision literature, e.g., Rand Error (RE) of [16], Variance of Information (VI) of [1][28].

Ideally, an automated segmentation should attain 100% accuracy – its output should be free of both types of segmentation errors, namely false merge (under-segmentation) and false split (over-segmentation). However, it is not realistic to expect (near) 100% accuracy in practice; given the performances of the existing state of the art algorithms, one can generally assume that their outputs need to be corrected afterwards. Then, from a connectomics point of view, a segmentation algorithm should be designed to minimize manual labor (or algorithmic complexity) required for correcting its output[15].

To the best of our knowledge, there has not yet been a study analyzing the effect of segmentation errors on the effort necessary to correct them. Although error quantities, such as Rand Error (RE) [16], provide a coarse assessment of the mistakes an algorithm makes, they are unable to conclusively forecast the amount of work required for refinement. As an example, inaccurately combining two regions of sizes A and B would incur the same RE value as incorrectly splitting one region of size A+B into two parts. However, rectifying these two mistakes demands significantly different amount of work [5]. The high RE of a false split of two large bodies, e.g., $A = B = 10000$, on a 512×512 image disproportionately penalizes the effort to correct such

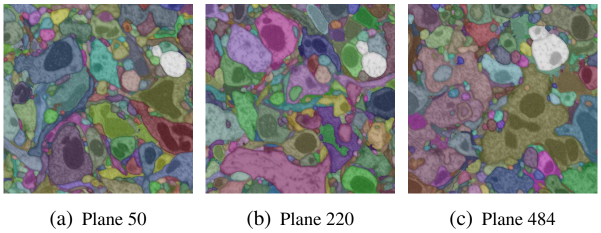


Figure 1. 3D Segmentation output on 3 planes from a volume of 500 images. Each individual neuron has been colorized with a different color. The two adjacent regions, colored in white, in the top-right, are in fact parts of two different neurons which have been falsely merged. Manually correcting this under-segmentation error is much more labor-intensive than correcting a false split.

error.

From a reconstruction perspective, an over-segmented result is preferred over an under-segmented one because a fragmented set of regions can be refined by automated methods such as agglomeration [25][27][28] or co-segmentation [9], but an under-segmented region can only be fixed by a human expert. Even for a human expert, identifying and correcting false merges is more difficult than correcting false split [5]. This difficulty is more pronounced in 3D volume segmentation than it is in 2D segmentation. Consider separating the two regions falsely connected through 450 planes (from 50 to 500) of a 520^3 volume by a segmentation method as displayed in Figure 1. The authors of [25][27][28] were aware of this issue and reported the two types of error rates separately for performance assessment. The study of [14] attempts to reduce false merges by identifying the locations vital for preserving topology given exhaustive groundtruth of the data.

Another desirable property of the EM segmentation algorithms is to be able to train the necessary components efficiently without compromising accuracy. An efficient training is perhaps essential for large scale reconstruction where one may anticipate learning the predictors multiple times for different neuropils. A quick segmentation result may also assist the neurobiologist to decide the optimal sample preparation that would maximize segmentation accuracy. But, training existing segmentation algorithms [14][7][19] remains a significant bottleneck in connectomics [11] due to the time and effort necessary for generating the groundtruth and time complexity of training the classifier (e.g., deep neural networks).

A highly curated exhaustive groundtruth, such as those offered by the segmentation challenges (e.g., ISBI 2012 2D, SNEMI 2013 3D), demands extensive effort. Provided necessary resources, it is possible to generate a reasonable groundtruth by iteratively refining segmentation on a small volume with an interactive labeling tool such as ilastik [29]. This label set is expected to contain a small degree of tolerable noise but is efficient to generate. Some recent algorithms [1][25][27][28] have utilized interactively generated groundtruth to train the necessary tools for segmentation. However, these algorithms inherently rely on highly expert annotators or neurobiologists in order to produce a useful annotation efficiently (by finding out the minimal area to label for the prediction-correction scheme). Automated algorithms are expected to diminish such dependency on human expertise. As an alternative to exhaustive labeling, Jones et.al. [18] presented a method for sparsely labeling the membrane locations based on appearance similarity to user annotated examples. A completely semisupervised approach like [18] will be sensitive to the penalty parameter and has a risk of introducing noises that are too difficult for a classifier to tolerate.

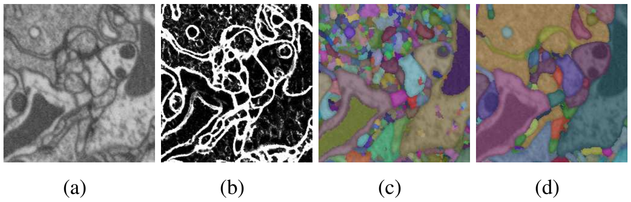


Figure 2. Workflow of a standard EM segmentation framework: (a) input → (b) pixelwise classification (white: membrane, black: non-membrane) → (c) over-segmentation → (d) final segmentation.

We adopt a standard EM segmentation approach [16][1][25][28], as illustrated in Figure 2, where the confidence values of a pixelwise classifier [1] are utilized to generate an initial over-segmentation of the dataset. The over-segmentated image or volume is then refined by aggregating superpixels with the help of a superpixel boundary classifier. In this paper, we propose an algorithm for training pixel and superpixel boundary classifiers. The classifiers are trained to attain two desirable properties of an EM segmentation method:

1. Maximize efficiency: the proposed algorithm employs active learning for classification. Instead of requiring an exhaustive pixel-level groundtruth, *our algorithm automatically determines a small fraction of samples that are critical for training* the pixel and superpixel boundary classifiers ($< 1\%$ for pixel and $< 20\%$ for superpixel boundary). These examples are identified using the disagreement between two predictors: a) a classifier being updated iteratively, and b) a semisupervised label propagation algorithm [4] predicting labels based on feature similarity. Unlike [18], all our training examples are labeled by an annotator.

2. Minimize false-merge: without exhaustive groundtruth, it is not possible to locate the topologically critical pixels using the method of [14]. We hypothesize that emphasizing on the detection of membrane pixels over other types would reduce the amount of false merges. Accordingly, our training protocol is designed to be biased towards more accurate learning of membrane class than the remaining categories.

We empirically demonstrate the advantages of the proposed method over the state of the art techniques for neural reconstruction from both 2D and 3D EM data. The over-

[1] We adopt multi-class pixel classification, as is explained later.

all active learning algorithm is defined in Section 2. Sections 2.2 and 2.3 explain how our active training approach is adapted for pixel and superpixel boundary classification. The following section (Section 3) discusses the experimental setup and reports the results. Finally, Section 4 concludes with a discussion summarizing our findings.

2. Proposed Active Labeling Framework

The segmentation scheme we adopt consists of pixel classification followed by a superpixel clustering by means of a superpixel boundary classifier. We propose an active strategy to train both the pixel and superpixel boundary classifiers. The goal of an active learning method is to identify a few examples – crucial for training a classifier – from a pool of unlabeled samples. The proposed active classification scheme identifies the challenging examples from the dataset and requests their labels from user. Given the labels for the query examples, the algorithm reconfigures its predictors and identifies a new set of queries in a repetitive fashion.

With the aim of locating these challenging examples, we estimate the class label of any unlabeled point by two predictors having substantially different views of the dataset. One predictor is a classifier (Random Forest (RF) [3] in our experiments) trained from an initially available subset of datapoints $X_l \subset X = \{x_1, \dots, x_n\}$ and their labels Y_l. The other predictor is a novel variant of semisupervised label propagation algorithm [33][4], that assumes a cluster formation of similar datapoints in feature space. While the classifier assesses the class of an unlabeled example by a discriminative set of rules learned so far, the label propagation technique extrapolates a prediction based on feature similarity among the datapoints.

A training sample is considered to be challenging if the class suggested by feature similarity is different from that calculated by the discriminative rules and vice versa[2]. The disagreement among these two types of estimates is quantified by a ranking formula. The first few examples in descending order of disagreement measure are presented to the user as queries. The set X_l is augmented by this new annotated queries and the whole process is repeated until a predefined stopping criterion is satisfied.

In Section 2.1, we propose the semisupervised label propagation method for a multiclass setting to facilitate the multiclass approaches of [27][25]. The strategies for query generation and initialization are different for pixel and superpixel boundary classification and are explained in Sections 2.2 and 2.3 respectively.

2.1. Proposed Multiclass Label Propagation

Let us suppose, we have n datapoints x_i that we wish to classify into one of the k classes. Let $\mathbf{f}_i$ denote the indicator variable for datapoint x_i: $f_i^c = 1$ if x_i is classified to class c and rest of its values are 0. We wish to assign 'similar' datapoints into the same class, i.e., the pairs of samples x_i and x_j with large feature similarity quantified by w_{ij} should belong to the same class. We propose to attain this by minimizing the following cost.

$$J(\mathbf{f}) = \sum_{i\sim j} w_{ij} \left[\frac{\mathbf{f}_i}{\sqrt{d_i}} - \frac{\mathbf{f}_j}{\sqrt{d_j}}\right]^T \left[\frac{\mathbf{f}_i}{\sqrt{d_i}} - \frac{\mathbf{f}_j}{\sqrt{d_j}}\right] \quad (1)$$

$$= 2\sum_i \mathbf{f}_i^T \mathbf{f}_i - 2\sum_{i\sim j} \frac{w_{ij}}{\sqrt{d_i}\sqrt{d_j}} \mathbf{f}_i^T \mathbf{f}_j. \quad (2)$$

In this cost function, we normalize the weight w_{ij} by the corresponding degree $d_i = \sum_j w_{ij}$ to balance the effects of disparity in class sample size. The cost is summed over all neighboring $i \sim j$ that possess a feature similarity above a certain predefined value. Using a matrix notation for the indicator variables, $F = [\mathbf{f}_1^T, \dots \mathbf{f}_n^T]^T$, we can write this cost function as

$$J(F) = 2\,\mathrm{Tr}\{FF^T(I - D^{-0.5}WD^{-0.5})\}, \quad (3)$$

where I and D are the identity and diagonal degree matrices respectively. By relaxing the values of F to be nonnegative real-valued numbers $f_i^c \geq 0$ and differentiating wrt F, one can compute the system of linear equations needed to be solved for determining F. Of course, the minimization is constrained by label consistency among the values of $\mathbf{f}_i$, i.e., $F\mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is a vector of all 1's.

$$\frac{\partial J}{\partial F} = 0 \implies \left(I - D^{-0.5}WD^{-0.5}\right)F = 0 \quad (4)$$

An efficient solver for Equation 4 is essential to build an interactive interface of our method. By avoiding the factorization of matrices with thousands of variables, iterative techniques can produce a solution significantly faster than the closed form methods with the same level of accuracy (up to a certain error tolerance). A stationary iterative formulation of this equation would repeatedly update the solution using the following formula [20].

$$F_{next} = D^{-0.5}WD^{-0.5}F \quad (5)$$

This iteration will converge if: 1) the absolute value of the eigenvalues of $D^{-0.5}WD^{-0.5}$ is bounded by 1, and 2) $I - D^{-0.5}WD^{-0.5}$ is non-singular [20]. Since there is no bipartite connected component in the graph corresponding to W, the first condition is satisfied [6]. We add a small perturbation to the quantity $D^{-0.5}WD^{-0.5}$ to attain non-singularity. One must also satisfy the label consistency constraint $F\mathbf{1} = \mathbf{1}$ to reach a meaningful solution.

In our active learning setting, the algorithm is given the labels for m out of n examples (where $m << n$) at the beginning of the process. We set the known labels in F and iterate Equation 5 followed by a projection onto $F\mathbf{1} = \mathbf{1}$

[2] For the interested readers, we illustrate the intuition behind our query generation approach on a synthetic dataset in[26].

Table 1. Multiclass Label Propagation algorithm

```
Algorithm: Multiclass Label Propagation
repeat
  1. Set the known labels in $F$.
  2. Update solution by Equation 5.
  3. Project onto $F\mathbf{1} = \mathbf{1}$
until convergence
```

until convergence for computing the unknown label confidences. The algorithm is outlined in Table 1 and has similarity to a past approach for efficient label propagation on large dataset [32].

After a query set is annotated by user, the linear equations in 4 need to be solved again. Instead of starting the solver algorithm (Table 1) from scratch, we begin with the most recently converged F as the initial solution. Such a warm start brought about a significant speed-up without altering the output in our experiments.

2.2. Active Learning for Pixel Classification

In pixel classification, each datapoint x_i of the above formulation corresponds to a pixel. We will denote a pixel by a different literal u_i to distinguish it from it from superpixel boundary defined later. In our design, each pixel is classified into one of the four classes: membrane, cytoplasm, mitochondria, mitochondria border [27].

Initial Subset Selection : Equal size subsets of samples, one for each class, are selected from the dataset to constitute the initial dataset X_l for label propagation. In the interactive setting, the user will be required to select the initial X_l using a GUI.

In an attempt to maximize the detection of membrane pixels, the initial training set for the RF classifier is constructed from a subset of X_l that contains different number of examples for different classes. In the following text, we describe how the pairwise similarity values in W are utilized to determine the sample proportion for different categories.

Introducing indicator vectors $\boldsymbol{\alpha}^m$ and $\boldsymbol{\alpha}^o$ for membrane class m and other classes o respectively, one can determine the sample proportion by solving an optimization problem. The value of $\alpha_i^m = 1$ if the i-th membrane example in X_l is selected and $\alpha_i^m = 0$ otherwise. The following formulation will select of largest subset of initial samples that will prevent misclassification of any member of class m in a nearest neighbor classifier setting.

$$\max_{\boldsymbol{\alpha^o},\boldsymbol{\alpha^m}} \sum_i \alpha_i^o + \alpha_i^m$$
$$\text{s.t.} \sum_{l:y_l=m} \alpha_l^m w_{ij} \alpha_j^m \geq \sum_{i:y_i=o} \alpha_i^o w_{ij} \alpha_j^m, \quad \forall_j \; y_j = m$$
$$\alpha_i^m, \alpha_i^o \in \{0,1\} \qquad (6)$$

Here, $y_i \in \{m, o\}$ indicates whether u_i belongs to membrane or other categories. In practice, we compute a suboptimal solution to this problem for efficiency. In our solution, $\alpha_i^m = 1$ for all i. We then greedily select examples for each class o to increase Cut(m, o) as long as Vol$(o) \leq$ Vol(m); we refer the reader to [24] for the definitions of these terms and to comprehend the motivation behind our heuristics.

Query Generation : Let the vector $\mathbf{p}_i$ denote the prediction confidences generated by the classifier for an unlabeled pixel u_i, where p_i^c corresponds to the confidence towards class c. If one wishes to compute the over-segmentation from the classifier probability for membrane class p_i^m, it is favorable to have $p_i^m > p_i^o$, $o \neq m$ for all membrane pixels. For a pixel u_i from the other classes, the deviance $p_i^o - p_i^m$ should be maximized instead. We define a margin vector $\bar{\mathbf{p}}_i$ wrt class m consisting of these quantities defined as follows.

$$a = \arg\max_c p_i^c$$
$$\bar{p}_i^c = \begin{cases} 0, & c \neq a \\ p_i^m, & c = a = m \\ p_i^a - p_i^m, & c = a \neq m \end{cases} \qquad (7)$$

Let $\bar{\mathbf{g}}_i$ be the margin wrt class m computed for u_i in a similar fashion from the real-valued outputs of multiclass label propagation algorithm. The disagreement $\delta(u_i)$ between these two estimates is computed by the dot product of their differences.

$$\delta_{\text{pixel}}(u_i) = (\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_i)^T (\bar{\mathbf{g}}_i - \bar{\mathbf{p}}_i). \qquad (8)$$

The margins $\bar{\mathbf{g}}_i$ and $\bar{\mathbf{p}}_i$ are modeled to capture the overlap in confidence the two predictors have between membrane and other classes. The disagreement $\delta_{\text{pixel}}(u_i)$ between these two margins will increase when the confidence distributions deviate from one another. A few unlabeled samples with largest disagreement value $\delta_{\text{pixel}}(u_i)$ will be selected as the next set of queries to be presented to the user. After the termination of the training process, the real-valued confidences of the classifier (RF in our case) are used for the subsequent tasks.

2.3. Active learning for Superpixel Boundary Classification

The output confidence of pixel classifier (RF in our cases) is utilized to generate an over-segmentation of the image or volume (see Figure 2). In order to aggregate the fragments into actual cell regions, each boundary between two superpixels of this over-segmentation needs to be classified as true or false boundary. We employ a superpixel boundary classifier (RF) that is also trained using the active learning method. For this training, each datapoint v_i corresponds to a superpixel boundary.

Initial Subset Selection : In order to reduce redundancy, the initial labeled set X_l was populated by the centers of the output of a clustering algorithm such as k-means.

Query Generation : Given the real valued confidences q_i from the current classifier and the estimates h_i of the label propagation method, we use the following formula to compute disagreement between them.

$$\delta_{\text{sp}}(v_i) = (q_i - h_i)^2. \tag{9}$$

Note that, since there are only two classes, values of both q_i and h_i are scalar for superpixel border classification. A few samples with largest $\delta_{\text{sp}}(v_i)$ are selected as the next query set to be annotated. After the training terminates, the real valued predictions from RF are used for superpixel clustering.

3. Experiments and Results

The proposed algorithm has been tested for both 3D volume and 2D image segmentation problems. In the following, we will describe the experimental setup, i.e., computation of the intermediate quantities, feature representation etc. for pixel and superpixel boundary classification. The Sections 3.2 and 3.3 report the results on 3D and 2D data respectively.

3.1. Experimental Setup

Pixel classification : As noted earlier, each pixel was classified into four classes: membrane, cytoplasm, mitochondria, and mitochondria border. A pixel is represented by features similar to those utilized in ilastik [29], e.g., gaussian smoothing, gradient magnitude, laplacian of gaussian, hessian of gaussian and its eigenvalues, structure tensor and its eigenvalues etc. computed at different scales. The similarity values for a pair of examples $\{u_i, u_j\}$ were generated by gaussian distance between their feature representations: $w_{ij} = \exp\{-\frac{1}{2}(\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)^T \Sigma^{-1} (\boldsymbol{\phi}_i - \boldsymbol{\phi}_j)\}$ where $\boldsymbol{\phi}_i$ are the feature values of u_i and Σ is the covariance matrix among all feature vectors.

Superpixel boundary classification : Given the pixel detection result, we utilize the predicted confidence values of the membrane class for generating an over-segmentation by the watershed algorithm [2]. In order to generate the watershed, we used all the pixels (or clusters or pixels larger than size 3) with RF confidence for membrane class $p_i^m < 0.01$. For superpixel clustering, we follow a context-aware agglomeration approach of [27] that was designed to prevent under-segmentation by delaying some merge decisions during agglomeration. This agglomeration scheme first clusters the cytoplasm superpixels together using a superpixel boundary predictor and then absorbs the mitochondria bodies into the agglomerated cytoplasm regions based on their degree of inclusion. A superpixel boundary predictor for this setup considers the cell boundary as well as the border between mitochondria and cytoplasm as true boundaries and only the borders between over-segmented cytoplasm superpixels as false boundaries.

Each boundary is represented by the statistical properties of the multiclass probabilities estimated by the pixel detector. The statistical properties include mean, standard deviation, 4 quartiles of the predictions generated for the data locations on the boundary, two regions it separates as well as the differences of these region statistics [27]. All of these features can be updated in constant time after a merge – a property which improves the efficiency of the segmentation algorithm substantially. The affinity values between two suprepixel boundaries were computed by the same formula used for pixel classification.

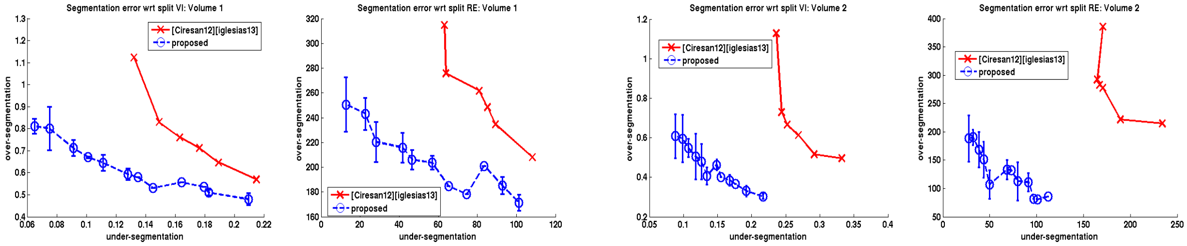
3.2. Result on 3D segmentation

We have tested our algorithm for 3D volume segmentation on Focused Ion Beam Serial Electron Microscopy (FIBSEM) isotropic images collected from fruit fly retina with a resolution of $10 \times 10 \times 10$nm. One 250^3 volume and two 520^3 volumes were used as training and test datasets respectively. The proposed algorithm does not need an exhaustive pixel-level groundtruth. However, for this particular experiment, instead of presenting queries to an annotator, we read off their labels from a noisy pixel level groundtruth generated earlier for another study [27]. Each of the segmentation tasks, namely pixel classification, over-segmentation and subsequent context-aware agglomeration were performed in 3D.

The performance of our algorithm was compared against a combination of [7] and [25] that has been one of the top scorer of the SNEMI 3D segmentation challenge 2013 (http://brainiac2.mit.edu/SNEMI3D). The neural net for pixel prediction was trained with the same techniques described in [7][10]. In order to further improve the quality of the probability maps, the outputs on rotated images were averaged together [8]. The watersheds were generated in the same manner as those of the proposed method and then the agglomeration technique of [25] was applied for superpixel clustering. Our effort to test the capability of [13], which is an extended version of [14], has not yet yielded results comparable to [7][25]; the supplementary material [26] discusses some reasons behind such performance

We report the under- and over-segmentation errors separately because under-segmentation is costlier than the other in terms of manual correction. Given a groundtruth, GT, and a segmentation, SG, split versions of variance of information (VI) [23] and Rand Error (RE) [16] were selected for performance evaluation. For split-VI, the over and under-segmentation are quantified by the conditional entropy $H(\text{GT} \mid \text{SG})$ and $H(\text{SG} \mid \text{GT})$ respectively. The over-segmentation and under-segmentation quantities in Rand Error are the ratios of pixel pairs within same cluster in GT but different cluster in SG and vice versa.

The proposed algorithm has been trained and applied 6 times to assess its consistency. In each training pass, we randomly subsampled a set of pixels from the whole training



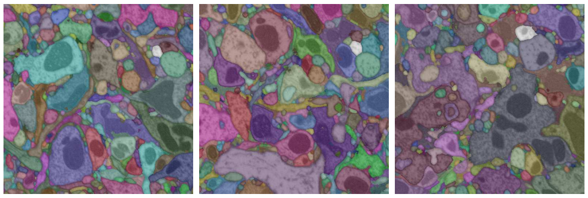
(a) Split-VI test vol1 (b) split-RE test vol1 (c) Split-VI test vol2 (d) Split-RE test vol2

Figure 3. Quantitative evaluation of competing methods on two FIBSEM test volumes. Left and right pairs of plots show the split-VI and split-RE errors of two methods on volume 1 and 2 respectively.

set so that the weight matrix W used in label propagation contains $\sim 0.5\%$ nonzero values and still fit in the available memory. The remaining parameters of the proposed active learning scheme are fixed to initial set size $= 4000$ (1000 each class), query set size $= 10$, number of queries $= 800$ for all the experiments reported in this paper. For the superpixel boundary learning, the parameters are set for all experiments to initial set size $= 3.5\%$ of total number of boundaries, query set size $= 10$, number of total boundaries labeled $= 15\%$ of all examples ($10000 \sim 14000$ in total). With our current implementation, the computation of pixel and superpixel training scheme needed around 24 hours on a 32 and 16 core cluster node respectively.

In Figure 3, we plot the split versions of error measures: x and y axes correspond to under- and over-segmentation errors respectively. Ideally, a segmentation algorithm should attain an error rate of 0, and therefore be plotted at the origin of the graph. For both the proposed and that of [7][25], the points on the plot were calculated by varying the stopping point of the agglomeration algorithm. The curve corresponding to the proposed method is an average of performances on 6 trials. On the two FIBSEM test volumes, the proposed algorithm (blue -o-) consistently produced lower false merge errors than that (red -x-) of [7][25] at the same over-segmentation error level.

The combined methods of [7][25] generally attained high quality segmentation in most areas of the test volumes. However, because they do not emphasize on the membrane class for training, their outputs were vulnerable to false merges near relatively weaker membranes. In Figure 1, we have displayed the false merge generated by [7][25] operating at agglomeration threshold 0.15 (highest point on the red curve of Figure 3(c)) on test volume 2. Segmentation produced by the proposed method did not reproduce this or any other false merges of similar size; the output of our method is shown in Figure 4(a)-(c) for the same three planes. The qualitative results from the proposed method was generated with an agglomeration threshold of 0.3 (halfway in the blue curves of Figure 3(c)). In both these images, the segmented regions are overlaid on the raw data with random color. Adjacent regions with same color may not always imply they are merged, please check [26] for the output and a visualization tool for a close inspection of results. In fact, both the test volumes were under-segmented in the watershed computed from [7]. The VI errors for under-segmentation for a watershed on [7] output were 0.132 and 0.236 respectively for two test volumes as opposed to 0.0188 and 0.0243 on average for those computed from our method. Such outcome may not be obvious from an examination of pixel probabilities computed by the proposed method and [7]; example predictions on Plane 484 are displayed in Figure 5(a)-(c). Indeed, the overall accuracy of our pixel detector is less than 90% on samples whose labels are unknown to the active algorithm. Although the deviation measure defined for active learning of pixel detection in Section 2.2 enables the identification of misclassified locations, the gain in classification accuracy is not the prominent factor contributing to the low under-segmentation error of our technique.

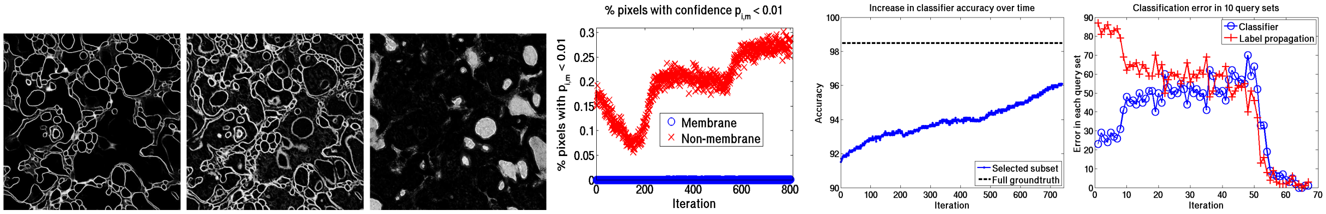


(a) Plane 50 (b) Plane 220 (c) Plane 484

Figure 4. (a)-(c):Result of the proposed method on the same three slices as displayed in Figure 1. The output contains no false merges of significant size.

The proposed pixel detection algorithm inherently minimizes the number of boundary pixels (and maximizes number of other types of pixels) receiving a confidence $p_i^m < 0.01$. Such an outcome is conducive to minimizing false merges in the consequent watershed method. In Figure 5(d) we plot the percentage of pixels of membrane (blue o) and other classes (red x) with $p_i^m < 0.01$ against the number of iterations. By construction, the algorithm starts with a very low, approximately 0.01%, of membrane pixels with $p_i^m < 0.01$. With the progression of the iterative updating of training examples, the proposed approach increases the percentage of other pixels with $p_i^m < 0.01$ while maintaining that for membrane pixels at the initial value.

In case of the superpixel boundary classifier, however, the training scheme effectively reduces the classification error in distinguishing false boundaries from the correct ones. In Figure 5(e), we plot the increase in accuracy of the classifier being actively trained (blue curve) and that of the one



(a) Membr.from [7] (b) Prop. membr. (c) Prop. mitochond. (d) % pixel $p_i^m < 0.01$ (e) SP accuracy (f) Error in SP query set

Figure 5. (a)-(c): Pixel predictor confidences on Plane 484. (a) by [7], (b) and (c) by proposed method for boundary and mitochondria class respectively. (d): percentage of pixels with $p_i^m < 0.01$ plotted against number of iterations. (e): increase in superpixel (SP) boundary classification accuracy with number of iterations. (f): prediction errors of the classifier and label propagation on every 10 query sets (100 samples) during superpixel boundary classification.

learned from all examples (black dashed line) on test samples. The plot shows a steady performance improvement with query iterations (x-axis). Interestingly enough, the error rates of both the predictors, namely the label propagation and the classifier, on query sets of images drops to zero after a certain number of iterations as shown in Figure 5(f). Such behavior has been observed in all the trials of superpixel boundary training and was utilized to determine a stopping criterion for training.

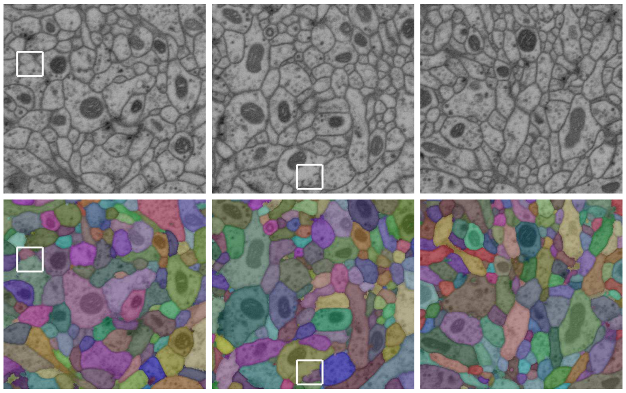
We have not reached to a point of zero error rates in query set for pixel classification. The error values of the proposed algorithm are found to be insensitive to the number of pixel queries in the range [700,1000] [26]. To further test the parameter sensitivity and robustness of our algorithm, we applied the proposed training with the exact same parameter on a 250^3 FIBSEM volume from a different region (mushroom body) of fly brain and produced almost perfect segmentation on a separate 512^3 mushroom body volume. Figure 6 shows outputs on some of the planes, note how the bias towards the membrane class of the proposed method resisted false merges on membrane gaps marked by white squares. Segmentation of all 512 images can be found in [26]. The VI error measures on a third 250^3 volume was $(0.085, 0.21)$.

3.3. Result on 2D segmentation - ISBI 12

We have also tested the proposed method for 2D segmentation on datasets provided for ISBI 2012 segmentation challenge (http://brainiac2.mit.edu/isbi_challenge/home). The challenge website provides a training set of 30 annotated images, generated by serial section Transmission Electron Microscopy (ssTEM) from the ventral nerve cord (VNC) of the Drosophila larva. We remind the user that an exhaustive groundtruth is not required by the proposed strategy because it automatically identifies the pixels and superpixel boundaries that are needed to be labeled by an annotator. For convenience of experimentation, and to incorporate some mistakes a human annotator would make in the active learning setting, we generated a noisy groundtruth by performing a watershed with all cell interior pixels marked as seeds and read off labels from this groundtruth.

A similar set of 30 images, without the groundtruth, was also provided for test purposes. The proposed method was applied on this dataset with the same number of samples and iteration for pixel classification as mentioned in Section 3.2. The number of examples utilized for superpixel boundaries is also similar to those stated in Section 3.2. In Table 2, we show the quantitative measures of performances of our method, that of [7] and another baseline algorithm that uses all pixels for training the pixel detector (Random Forest) and the technique of [25] for superpixel boundary training. Since the groundtruth for the test dataset is not available, the split versions of VI and RE could not be computed. A qualitative inspection of the results [26] suggests that the difference in error values between our method and those of [7] was most probably caused by over-segmentation.

For complete neuron reconstruction, the 2D segmentation results on anisotropic images – such as those of ISBI 12 dataset – need to be connected across planes by a linkage algorithm. The linkage algorithms have been shown to refine some false split errors, but cannot recover from false merges [9][31]. It is therefore rational (and may even be necessary) to prevent under-segmentation at a cost of small over-segmentation rate. This strategy will be more effective on difficult areas of EM volume characterized by broken or

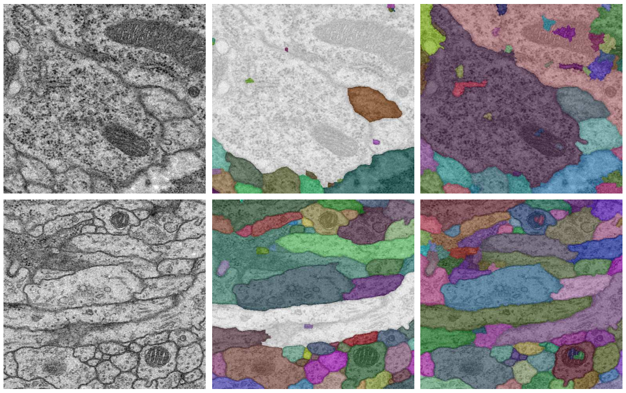


(a) Plane 22 (b) Plane 76 (c) Plane 300

Figure 6. Qualitative Result of the proposed method on mushroom body FIBSEM data with exact same parameter. White boxes mark gaps in the membrane where the proposed method successfully avoided false merge.

Table 2. Comparison of F-measure of Rand error provided by ISBI 2012 website.

	Proposed	[7]	All+ [25]
error	0.08	0.05	0.126



(a) Input (b) [7] (c) Proposed

Figure 7. Performance comparison with [7] on challenging regions of larva data. Regions highlighted in white (middle column) are falsely merged by watershed generated from [7].

hazy membranes or dark cell regions. We downloaded 20 images from two different regions of the whole larva dataset (http://fly.mpi-cbg.de/) and computed segmentation with predictors trained on the challenge data and the same set of parameters. For [7], the output was generated by applying watershed after thresholding the pixel prediction values at 0.3, same as that used to compute the winning entry of the ISBI 12 challenge.

As Figure 7 demonstrates, the proposed method prevents most of the false merges generated by [7] in these challenging areas and facilitates more accurate reconstruction through linkage algorithms like [9, 31]. An emphasis on learning the membrane class leads to a wall generally 'higher' than those from [7] around watershed basins. Results on all the 20 images can be found in [26].

Figure 8 shows images with some pixel locations (circle centers) selected as queries by our active pixel training method. Recall that the query set consists of the challenging examples – the locations where the estimation of the two techniques contradict each other. The regions covered by queries include patch between mitochondria and cell boundary, areas with darker shades. These regions often turn out to be misclassified (or receive low confidence) by a predictor trained in interactive setting of [29]. Example discrepancies between the confidences of label propagation and classifier used to determine these queries are plotted in [26].

4. Discussion

We have proposed a framework for training the necessary tools for an EM segmentation algorithm by acquiring some properties suitable for neural reconstruction. On one hand, the proposed method suggests a strategy to train without complete groundtruth by automatically selecting a small fraction of training examples. On the other hand, our algorithm is designed to minimize the false merge errors which are substantially more difficult to correct than the false split errors. The results demonstrate the merit of our method for neural reconstruction in comparison to the existing algorithms.

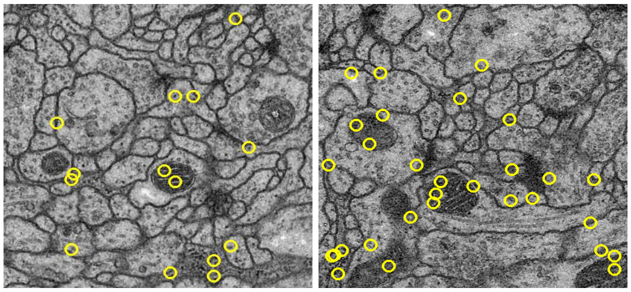


Figure 8. Sample queries determined automatically by the proposed method. Note how the queries were placed at challenging locations on images such as patch between mitochondria and cell boundary, areas with darker shades.

The proposed approach is designed to expedite multiple components of the overall neural reconstruction effort that led to high impact research in neuroscience [30][12]. A faster training method will assist the imaging expert to determine the optimal sample quality based on actual segmentation results rather than the raw images. A comprehensive search for segmentation errors is not practical in connectomics [30]. The presence of no or minimal undersegmentation is a prerequisite for applying intelligent proofreading methods that avoids 100% screening of the volume [17]. Efficiency in creating a groundtruth and training will also be valuable for computing different predictors for different neuropils for large scale reconstruction. It will be interesting to see how other classification tools can be extended to achieve these properties as well.

Acknowledgment

Toufiq Parag gratefully acknowledges Pat Rivlin, Chris Ordish, Corey Fisher for assistance in data preparation; Stuart Berg, Steve Plaza for software support; Gary Huang for computing DAWMR output; Stephan Saalfeld for providing the access to the complete larva dataset.

References

[1] B. Andres, T. Kroeger, K. Briggman, W. Denk, N. Korogod, G. Knott, U. Koethe, and F. Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *ECCV*. 2012. 1, 2

[2] S. Beucher and F. Meyer. The Morphological Approach to Segmentation : The Watershed Transformation. *Mathematical Morphology in Image Processing*, pages 433–481, 1993. 5

[3] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001. 3

[4] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006. 2, 3

[5] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer. Semi-automated reconstruction of neural circuits using electron

microscopy. *Current Opinion in Neurobiology*, 20(5):667–675, Oct. 2010. 1, 2

[6] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, Dec. 1996. 3

[7] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *NIPS*, 2012. 1, 2, 5, 6, 7, 8

[8] D. C. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *MICCAI*, volume 2, pages 411–418, 2013. 5

[9] J. Funke, B. Andres, F. Hamprecht, A. Cardona, and M. Cook. Efficient automatic 3d-reconstruction of branching neurons from em data. In *CVPR*, 2012. 1, 2, 7, 8

[10] A. Giusti, D. C. Ciresan, J. Masci, L. M. Gambardella, and J. Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *ICIP*, 2013. 5

[11] M. Helmstaedter. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature Methods*, 10(6):501–7, 2013. 2

[12] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168–174, Aug. 2013. 1, 8

[13] G. B. Huang and V. Jain. Deep and wide multiscale recursive networks for robust image labeling. *CoRR*, abs/1310.0354, 2013. 5

[14] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, Briggman, et al. Boundary learning by optimization with topological constraints. In *CVPR*, 2010. 1, 2, 5

[15] V. Jain, S. Seung, and S. Turaga. Machine that learn to segment images: a crucial technology for connectomics. *Current opinion in Neurobiology*, 20:653–666, 2010. 1

[16] V. Jain, S. C. Turaga, K. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung. Learning to agglomerate superpixel hierarchies. In *NIPS 24*, pages 648–656. 2011. 1, 2, 5

[17] C. Jones, T. Liu, N. W. Cohan, M. Ellisman, and T. Tasdizen. Efficient semi-automatic 3d segmentation for neuron tracing in electron microscopy images. *Journal of Neuroscience Methods*, 246(0):13 – 21, 2015. 8

[18] C. Jones, T. Liu, M. Ellisman, and T. Tasdizen. Semi-automatic neuron segmentation in electron microscopy images via sparse labeling. In *ISBI*, pages 1304–1307, 2013. 2

[19] E. Jurrus, A. R. C. Paiva, S. Watanabe, J. R. Anderson, B. W. Jones, R. T. Whitaker, E. M. Jorgensen, R. Marc, and T. Tasdizen. Detection of neuron membranes in electron microscopy images using a serial neural network architecture. *Medical Image Analysis*, 14(6):770–783, 2010. 1, 2

[20] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Number 16 in Frontiers in Applied Mathematics. SIAM, 1995. 3

[21] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen. A modular hierarchical approach to 3d electron microscopy image segmentation. *Journal of Neuroscience Methods*, 226(0):88 – 102, 2014. 1

[22] T. Liu, E. Jurrus, M. Seyedhosseini, M. H. Ellisman, and T. Tasdizen. Watershed merge tree classification for electron microscopy image segmentation. In *ICPR*. IEEE, 2012. 1

[23] M. Meila. Comparing clusterings by the variation of information. In *COLT'03*, pages 173–187, 2003. 5

[24] M. Meila and J. Shi. A random walks view of spectral segmentation. In *AISTATS 2001*, 2001. 4

[25] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical clustering to segment 2d and 3d images. *PLoS ONE*, 8(8), 08 2013. 2, 3, 5, 6, 7

[26] T. Parag. What properties are desirable from an electron microscopy segmentation algorithm. *CoRR*, abs/1503.05430, 2015. 3, 5, 6, 7, 8

[27] T. Parag, A. Chakraborty, S. Plaza, and L. Scheffer. A context-aware delayed agglomeration framework for electron microscopy segmentation. *PLoS ONE*, 10(5), 2015. 2, 3, 4, 5

[28] T. Parag, S. Plaza, and L. Schefher. Small sample learning of superpixel classifier for em segmentation. In *MICCAI*, 2014. 1, 2

[29] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht. "ilastik: Interactive learning and segmentation toolkit". In *ISBI*, 2011. 2, 5, 8

[30] S.-Y. Takemura et al. A visual motion detection circuit suggested by Drosophila connectomics. *Nature*, 500(7461):175–181, 2013. 1, 8

[31] A. Vazquez-Reina, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister. Segmentation fusion for connectomics. In *ICCV*, 2011. 1, 7, 8

[32] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation, 2002. Tech Report CMU-CALD-01-107, School of Computer Science, Carnegie Mellon University. 4

[33] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining Active Learning and Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *ICML 2003 workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003. 3