A Context-aware Delayed Agglomeration Framework for EM Segmentation

Toufiq Parag[1], Anirban Chakraborty[2⋆], and Stephen Plaza[1]

[1] Janelia Farm Research Campus-HHMI, Ashburn, VA, USA.
[2] University of California, Riverside, CA, USA.
paragt@janelia.hhmi.org

Abstract. This study addresses the segmentation problem of Electron Microscopy (EM) images by an extended hierarchical agglomerative merging. For the overall segmentation methodology, we propose a context-aware algorithm that clusters over-segmented regions of sub-classes (representing different biological entities) in different stages. A delayed scheme for agglomerative clustering is proposed where the fusion of the newly formed bodies is postponed in order to allow them to grow larger and produce more accurate region boundary predictions. We show significant improvements in segmentation accuracies attained by the proposed approaches over existing standard methods on both 2D and 3D datasets.

1 Introduction

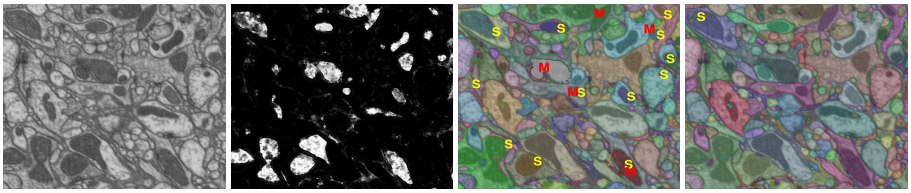
Connectomics is an emerging field in neuroscience where the goal is to discern neural connectivity in an organism. Recent advances of Electron Microscopy (EM) techniques have enabled us to image the neuronal bodies and their components in unprecedented level of details. The sizes of such datasets suggest that (semi-) automated region labeling or segmentation is the most viable strategy to conduct biological analysis on them. The outputs of such automated algorithms require manual correction afterwards [1].

Image segmentation for natural scenes have a long history in computer vision literature and there have been many successful methods that generate impressive segmentation results [2][3][4]. In recent years, there have been many fruitful attempts to identify meaningful regions in neuronal images collected by variants of EM technology as well [5][6][7][8] [9][10]. Most of these studies initially apply a pixel (or voxel)-wise classifier to generate the boundary confidence at any location and produce an initial (over)segmentation through methods such as Watershed [11]. Different approaches resort to different methods to refine the initial region labeling in order to generate the final segmentation. For anisotropic datasets, where depth resolution (z-dimension) is coarser than planar resolution (x, y dimensions), this step will be incorporated or followed by a registration process.

Several pixel-wise classifiers recommend increasingly complex pixel-wise detector models to determine whether each pixel[3] belongs to a cell membrane or not (binary classification) without any distinction among the intra-class sub-structures [12][13]. In

⋆ Work done during an internship at Janelia Farm Research.

[3] Unless it is obvious from the context, we denote locations on both 2D or 3D EM data as pixels in this manuscript.



(a) grayscale image (b) mitochondria probability (c) output of [14] (d) output proposed method

Fig. 1: Improved segmentation accuracy by context-aware approach on FIBSEM data.

contrast, some recent studies [6][14] apply relatively simpler classifiers (in terms of model size, learning time and convenience) to classify each pixel of the EM data into multiple classes to represent different sub-structures, e.g., cell boundary, cytoplasm, mitochondria etc. The results of [6][14] reveal that using prior domain knowledge to divide a problem into multiple components can achieve high segmentation quality with simpler classifier models requiring less computation.

However, we believe the studies of [6][14] do not extract the full benefit of multi-class predictions on EM data. Regardless of the quality of the multiclass pixel classifier, the algorithms in [14][6] do not distinguish between regions of one sub-structure (e.g., cytoplasm) to those of another (e.g., mitchondria) in the superpixel level. That is, the classification is divided into multiple sub-classes in pixel-level, but the subsequent assignment or agglomeration step does not utilize this additional information to achieve the final segmentation. This often leads to sub-optimal performances by these methods. For example, Figure 1(a) shows an EM image (plane in a 3D volume) where the mitochondria sub-class probabilities are depicted in Figure 1(b). By not using this sub-class explicitly in the agglomeration step, the final output of [14] failed to merge many regions into the correct cell (marked by 'S') and connected them to wrong cells (marked by 'M') as shown in Figure 1(c).

This paper introduces a context-aware agglomeration methodology to utilize the prior knowledge of sub-classes within the dataset to improve the segmentation quality. Different sub-structures are clustered in different phases in the proposed segmentation algorithm. We adopt an Agglomerative or Hierarchical clustering framework [7][8] due to its advantages such as low space, time complexity and flexibility to tune for over/under segmentation. We develop a two-pass agglomeration policy where the (estimated) non-mitochondria regions are combined together in the first phase and then the remaining mitochondria bodies are absorbed into corresponding cell cytoplasm. Our proposed context aware approach significantly reduces the false split and merge errors (example shown in Figure 1(d)) provided fairly accurate sub-structure detection. In addition, this strategy substantially reduces the training data requirement and predictor model complexity which in turn provides almost 30-fold increase in learning speed. The findings of this study further inspired us to design an interactive training algorithm [15] for region boundary predictor with only few samples ($< 20\%$) and has a potential to eliminate the necessity for a completely labeled groundtruth data for EM segmentation.

We also propose a modified version of the hierarchical clustering algorithm to minimize under-segmentation errors since these errors are conventionally costlier to correct

than the over-segmentation errors during the subsequent manual correction stage. In order to minimize the number of false merges, we 'delay' the merge decisions over a certain set of boundaries to be resolved at a later time. This adaptation of agglomerative clustering is motivated by the observation that region boundary predictors render a more accurate decision for newly merged bodies at later stages of agglomeration than those in the earlier [7]. The proposed modification reduces the number of false merges without increasing over-segmentation compared to the traditional agglomerative scheme of [8]. In addition, we also empirically show that our agglomeration approach performs better than a Global scheme [10] on our datasets and attempt to analyze why it does so.

2 Delayed Agglomerative Clustering

Let us suppose the initial over-segmentation process generated N superpixels $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ on an EM dataset with M neurites (neuronal regions) where $N \gg M$. Let $L(S)$ be the neurite region that S actually belongs to. Our goal is to iteratively merge these N superpixels such that each $S_i, i = 1, 2, \ldots, N$ is merged into its corresponding $L(S_i)$.

Algorithm 1: Delayed Agglomerative Segmentation

Input: $S_1, S_2, \ldots, S_N$ and confidence function h.
Output: $R_1, R_2, \ldots, R_{N'}$

forall the i, edge e **do** $R_i = S_i$, Flag(e) = WHITE **repeat**
  $W = \{e \in E \mid \text{Flag}(e) = \text{WHITE}\}$;
  if W empty **then**
    forall the e : Flag(e) = GRAY **do**
      if $h(e)$ within range **then** Flag(e) = WHITE
  $e^* \equiv \{R_i, R_j\} = min_{e \in W}\ h(e)$;
  Merge R_j to R_i and update W ;
  forall the $R_b \in Nbr(R_j)$ **do**
    if $h(\{R_i, R_b\}) > h(\{R_j, R_b\})$ **then**
      Flag$(\{R_i, R_b\})$ = WHITE;
    else
      Flag$(\{R_i, R_b\})$ = GRAY;
until *StoppingCriterion*;

We denote a boundary between two oversegmented regions by a pair of regions $e \triangleq \{S_i, S_j\}$ and the set of all such boundaries by E. In a graph representation, each of the regions S_i is considered to be a node and the boundary or face between two regions is regarded as an edge – a notation we will be using throughout the paper. Also, let the boundary label map $B : \mathcal{S} \times \mathcal{S} \rightarrow \{0, 1\}$ assign a 1 to a boundary that actually separates one neurite region from another and a 0 to the boundary incorrectly generated due to over-segmentation. In agglomerative clustering methods, a real-valued superpixel boundary confidence function $h : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}$ approximates $B(e)$. Starting with the e with lowest $h(e)$, these algorithms sequentially combines its two neighboring regions and immediately updates the boundaries and corresponding $h(e)$ for the new

body. The studies of [9][8][14] have convinced the EM segmentation community to apply a region boundary classifier as the confidence function $h(e)$.

Our adaption of segmentation also starts with the boundary with lowest $h(e)$ (Line 6 in Algorithm 1)) and absorbs region R_j into R_i. However, unlike conventional hierarchical clustering, we do not consider all the new boundaries $\{R_i, R_b\}$ between the recently merged R_i and its updated neighbors R_b. Instead we maintain a set of edges W and insert the new edge $\{R_i, R_b\}$ only if its confidence increases from that of $\{R_j, R_b\}$ after R_j is absorbed into R_i (Line 10 in Algorithm 1)). The faces, for which $h(\{R_i, R_b\})$ decreases from previous value, are kept aside until there are no members left in W and the modified confidence on $\{R_i, R_b\}$ is within the operating range(Line 12 in Algorithm 1)).

Effectively, the proposed strategy 'delays' the merging of new edges $\{R_i, R_b\}$ resulting from a merge: either due to increase of $h(\{R_i, R_b\})$ or deliberately if $h((\{R_i, R_b\}))$ decreases. In effect, this design postpones the merge decisions on the newly formed bodies for a later time to avoid making wrong decisions on smaller superpixels which could multiply with each iteration. As the average size of the superpixels grows larger, they are expected to generate more discriminative features for the face classifier to separate.

2.1 Complexity

Asymptotically, the running time of the delayed algorithm remains the same as the traditional agglomerative clustering in the worst case. In a priority queue implementation, instead of adding the adjacent boundaries to the queue, the delayed algorithm stores them in a separate list. Later, building a queue from this list would require $O(n_1)$ time where the length n_1 of new list must be smaller than that of the previous one (which contains all edges): $n > n_1$.

However, in practice, we observe a reduction in the running time of delayed agglomeration. Notice that, a subset of adjacent boundaries are not pushed back or updated into the queue (Line 13 of Algorithm 1). We may as well apply a simple trick to avoid updates at each merge altogether: instead of increasing key of the edges with increased h value (Line 11 of Algorithm 1), we can check and increase the key only when it becomes a candidate for merge (Line 7 of Algorithm 1) or in Line 6 when it being considered to be inserted into W. Thus, we can reduce the computation by $O(dnlogn)$ where d is the degree of S_2 and n is the queue size.

3 Context-aware Segmentation

There are at least three reasons for separating the agglomeration of cytoplasm and mitochondria regions: (1) The mitochondira-cytoplasm borders indeed have strong feature similarity with cell membranes although we need to dissolve the former and retain the latter. Figure 2 shows the histogram of confidence levels of a trained cell boundary predictor for actual cell boundaries and mitochondria-cytoplasm borders. The overlap in the range $0.1 \sim 0.5$ suggests that absorbing mitochodria into correct cells would produce a large number of false merges of among neural regions.

(2) The distribution of same features computed on cytoplasm and mitochondria will be substantially different from each other. Combining these two types of feature value distribution may lead to failure of the boundary detector to identify false boundaries between cytoplasm superpixels such as the one on the lower left of Figure 1(c). (3) There exists a high probability of incorrectly connecting two neurons when the mitochondria are closely located to the cell membrane or other mitochondria regions from neighboring cells, often blurring the boundary. Figure 1(c) shows two such locations marked as M at the center-left and top positions. Despite this, past studies do not distinguish between the two sub-classes of superpixels for segmentation purposes. The work of [10] ignores mitochondria regions altogether from the segmentation and evaluation process. Our examination of both isotropic and ansitropic data suggests cell structures cannot be meaningfully identified without these regions and it is non-trivial to combine motochhondria detection with a segmentation that ignores it in order to produce the final segmentation.

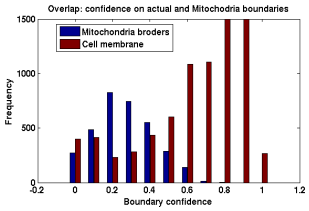


Fig. 2: Predicted boundary confidences on true cyotplasm-mitochondria borders and correct cell boundaries. The plot is clipped at $y = 1500$ for better visualization.

In our algorithm, we separate the set $\mathcal{S}_m$ of potential mitochondria superpixels from the set $\mathcal{S}_c$ of potential cyptoplasm superpixels assuming the existence of an effective mitochondria detector (e.g., [16]). The edges among regions in $\mathcal{S}_c$, are agglomerated first using a trained superpixel face classifier. We train a Random Forest (RF) [17] classifier to act as the boundary confidence function for clustering the set $\mathcal{S}_c$ of cytoplasm superpixels. During h_c training, mitochondria-cytoplasm borders are treated the same way as cell membrane.

In the second step, the mitochondria-cytoplasm edges are merged utilizing a different confidence function h_m. In order to absorb mitochondria into corresponding cells, we apply the delayed-agglomeration algorithm with a small alteration. The *set of candidate edges W only contains the edges between mitochondria and cytoplasm*, i.e., $W = \{\{S_c, S_m\} \mid \text{type}(S_c) = \text{Cyto}, \text{type}(S_m) = \text{Mito}, \text{Flag}(\{S_c, S_m\}) = \text{WHITE}\}$ (i.e., mitochondria-mitochondria edges are not considered). Biologically, each mitochondrion should reside within a cell body. Therefore, boundary confidence for mitochondria merging should reflect how much a mitchondria is contained within a cytoplasm. For any edge $\{S_m, S_c\}$ with a mitochondria superpixel S_m and a cytoplasm superpixel S_c, the confidence is defined as $h_m(\{S_m, S_c\}) = 1 - \frac{\text{length}(\{S_m, S_c\})}{\sum_i \text{length}(\{S_m, S_i\})}$.

4 Experiments and Results

4.1 Volume Over-segmentation and training:

We learn a classifier to assign each individual pixel into multiple categories such as cell boundary, cytoplasm, mitochondria and mitochondria boundary etc. using the interactive tool Ilastik [18]. In effect, this pixelwise detector is a Random Forest (RF) classifier [17] trained on a few sparse samples from the dataset. The locations with lowest pixelwise cell boundary prediction are utilized as markers for the Watershed algorithm [11] to produce the over-segmentation of the volume.

The set $\mathcal{S}_m$ of probable mitochondria superpixels is populated with all regions having mean mitochondria probability (estimated by pixelwise classifier) above a certain threshold. The rest of the superpixels constitute the set $\mathcal{S}_c$ of possible cytoplasm regions. The training set for superpixel boundary classifier h_c consists of all boundaries among members of $\mathcal{S}_c$. Similar to [9][14], each cytoplasm superpixel edge is represented by the statistical properties of the multiclass probabilities estimated by Ilastik. The statistical properties include mean, standard deviation, 4 quartiles of the predictions generated for the data locations on the boundary, two regions as well as the differences of these region statistics. All of these features can be updated in constant time after a merge – a property which improves the efficiency of the segmentation algorithm substantially.

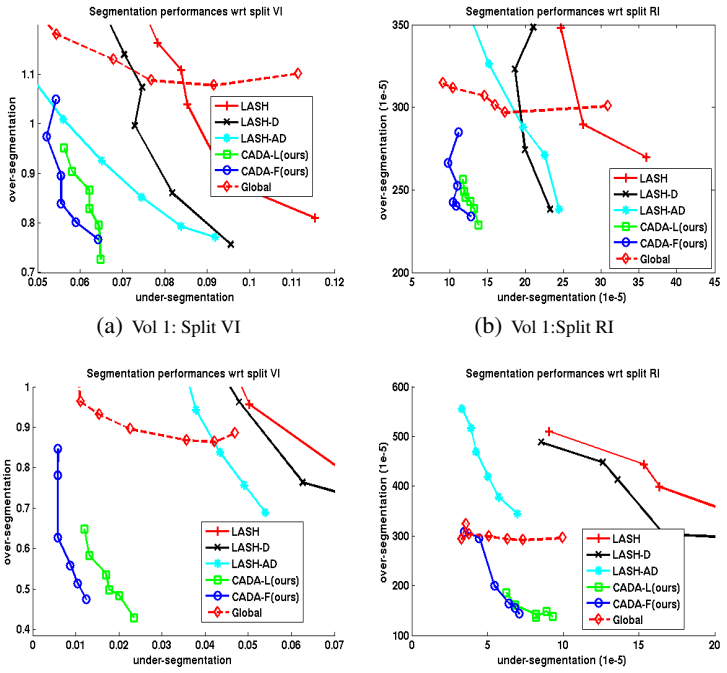
4.2 Segmentation Performances-FIBSEM data:

The first set of experiments was conducted on isotropic datasets from fruit fly visual system imaged at 10 nm resolution using FIBSEM technology. This data is segmented as a volume (i.e., 3D segmentation) and both the voxelwise multi-class predictor and the supervoxel boundary classifier are learned on one 250^3 volume and applied on two 520^3 test volumes.

We have compared the following algorithms in this study: 1) LASH: Standard agglomeration with an RF supervoxel classifier learned based on the iterative procedure of [8]. 2) LASH-D: LASH classifier with delayed agglomeration (proposed extension). 3)LASH-AD [14]: an agglomerative method with repetitive learning phases like LASH, except it accumulates the training sets of multiple phases. 4) CADA-F: Proposed two stage delayed agglomeration with standard RF learned using training set accumulation similar to LASH-AD. 5) CADA-L: Proposed delayed agglomeration with a depth-limited RF (depth =20) learned *without* training set accumulation. 6) Global: the globally optimal closed-surface segmentation proposed in [10]. For [10], the boundary confidences were generated by the CADA-L predictor.

Split versions of variance of information (VI) [19] and Rand Index (RI) [20], as described in [14], were selected to evaluate segmentation errors. We plot the contribution of under-segmentation and over-segmentation to VI (i.e., $H(\text{result} \mid \text{groundtruth})$ and $H(\text{groundtruth} \mid \text{result})$) on x and y-axis respectively in Figures 3(a) and 3(c) for test Volumes 1 and 2. Motivated by the calculation of RI, we plot the ratio ($\times 10^{-5}$) of pairs of voxels falsely merged and split by different algorithms in similar plots in Figures 3(b) and Figures 3(d). In these figures, an ideal algorithm should achieve a zero value for both over and under-segmentation. For all algorithms except the Global method, each point in a plot refers to the boundary confidence threshold $\delta_c \in [0.1, 0.2]$ which was used as stopping criterion for cytoplasm merging. For [10], we instead changed the value of the bias parameter in weight calculation within the range $[0.2, 0.9]$. During segmentation, the variants of proposed algorithm finished agglomeration (for a specific δ_c) within $3 \sim 6$ minutes on an Intel Xeon 3.16 GHz CPU with 64 GB memory.

As the plots show, both the delayed agglomeration and two-phase segmentation process attained significant improvement over past methods; compare the performance of LASH (red +) with LASH-D (black x) and that of LASH-AD (cyan *) with CADA variants. Compared to the rest of the techniques, the two variants of proposed methods, namely CADA-L and CADA-F, appear to achieve the most favorable segmentations



(a) Vol 1: Split VI (b) Vol 1:Split RI

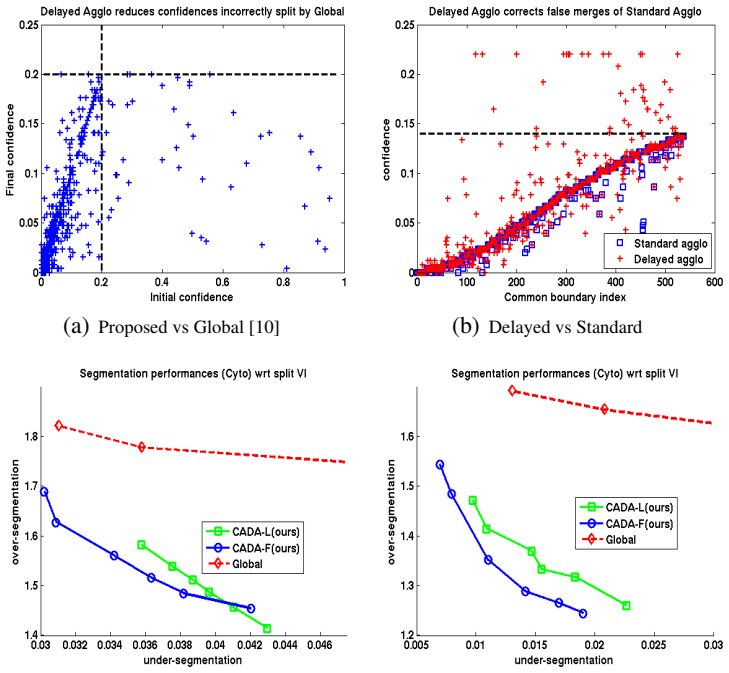
(c) Vol 2: Split VI (d) Vol 2:Split RI

Fig. 3: Segmentation error in terms of split-VI (left column) and split-RI (right column)on two FIBSEM volumes. Top: Test volume 1 and bottom: Test volume 2.

by reducing the over-segmentation steeply without increasing the false merge numbers much. We have observed similar performances over multiple runs of each segmentation algorithm. Qualitative results in Figure 1 offers an intuitive explanation for the better performances of CADA variants over the rest of the methods that are context invariant.

It is worth mentioning that, in a two stage segmentation scheme, the performance of a depth limited RF (i.e., CADA-L, green square), learned without accumulating training set over multiple passes, is very similar to that of the standard RF (CADA-F, blue circle) trained over cumulative learning passes. Training full-depth RF (CADA-F) needs more than 3 hours whereas training a depth limited single iteration (CADA-L) requires ≤ 5 minutes.

Global vs Proposed: The split-VI plot in Figures 4(c) and (d) show that both variants of the proposed CADA algorithm generates significantly low under and over-segmentation errors than those of Global [10] method in clustering *cytoplasm regions only (mitchondria not merged)*. In order to analyze why this happens, we plot the initial confidences of $h_c(e)$ on all e incorrectly split (over-segmented) by the Global method but correctly merged by proposed algorithm on x-axis of Figure 4(a). The y-axis of Figure 4(a) corresponds to $h_c(e)$ at the time e was correctly merged by the proposed method. The threshold on boundary confidences to stop agglomeration was $\delta_c = 0.2$. Notice that, the agglomerative process correctly reduced the confidences of many false boundaries that received a high score by the predictor at the beginning (high x value but low y value). This refinement is possible through the evolution of the superpixels in the agglomera-



(a) Proposed vs Global [10] (b) Delayed vs Standard

(c) Vol 1: Split VI cytoplasm (d) Vol 2: Split VI cytoplasm

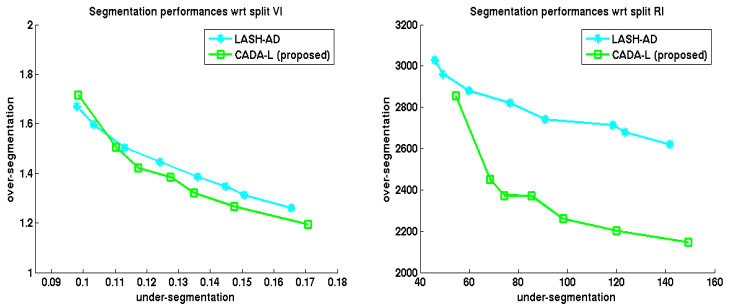
Fig. 4: (a) False splits of Global method corrected by proposed CADA-L, (b) False merges of Standard agglomeration corrected by delayed method, (c) - (d) split-VI of cytoplasm segmentation of two FIBSEM volumes.

tive process – an advantage the Global method of [10] cannot benefit from. Although Global method could identify many false boundaries in the volume, the enclosed region of Figure 4(a) suggests it also generated many false positives for membranes with low confidence score $h_c(e)$.

Delayed vs standard agglomeration: In order to illustrate the improved accuracy attained by the delayed agglomeration over the standard one, we collected all *faces that were incorrectly dissolved by standard agglomeration algorithm* (LASH) and examine their confidences under a delayed scheme (LASH-D) operating at $\delta_c = 0.14$ The confidences (clipped to 0.25) of these 534 edges generated by standard and delayed agglomeration are plotted in Figure 4(b) in blue square and red $+$ respectively. The proposed delayed agglomeration accurately increased the confidences h_c of many of these faces, among which, 41 exceeded the threshold of 0.14 (green line) and avoided a false merge. In addition to these common supervoxel edges, the standard and delayed algorithms independently generated 163 and 4 more incorrect merges respectively.

4.3 Segmentation Performances-ssTEM data:

We also report some preliminary results on a different data modality, namely ssTEM 2D images. The segmentation is performed on each image (without connecting them across planes) and Figures 5(a) and (b) plots the average of split-VI and split-RI errors over 15 images of size 1000×1000 of the proposed CADA-L and LASH-AD [14] methods.



(a) TEM images: split-VI (b) TEM images: Split RI

Fig. 5: Segmentation errors in terms of split-VI (a) and spli-RI (b) on TEM data.

Although the performance of out mrthod CADA-L is marginally better than LASH-AD in terms of split-VI, the former seems to perform better than LASH-AD in terms of split-RI (Figure 5(b)) suggesting that it is segmenting smaller regions better. The result of the Global method [10] were too poor to show on this plot. While a more accurate mitochondria predictor could potentially reduce the segmentation errors of the proposed method (as FIBSEM results suggest), context-invariant algorithms such as LASH-AD would be less effective around mitochondria regions. It is worth mentioning here that, compared to LASH-AD, CADA-L used less than half of the training examples (42.46%) collected without training iterations (i.e., significantly more efficient in training).

5 Conclusion

Our results indicate that a context-aware clustering of sub-classes such as cytoplasm and mitochondria can improve segmentation accuracy significantly given fairly accurate sub-structure detection. In addition to reducing the over- and under-segemntation errors, it demands fewer training examples (and no training iterations) which largely expedites the training procedure. Our analysis also illustrates how a delayed agglomerative procedure benefits from the intermediate boundary probabilities and corrects many false merge/split errors generated by existing algorithms. We strongly believe these findings would be very useful in designing more accurate segmentation techniques for EM volume segmentation.

We argue that, due to considerable ambiguity in appearances, it is only rational for an EM segmentation algorithm to be context-aware in each of its stages, i.e., in both pixel and superpixel levels (and in alignment for anisotropic data). Detecting the sub-classes and considering them separately as necessary is a highly effective strategy to supersede existing algorithms in accuracy and speed.

Acknowledgment

The authors would like to thank Pat Rivlin, Stuart Berg, Don Olbris, Bill Katz, Matt Saunders, Lei-Ann Chang of Janelia Farm Research Campus-HHMI for their help on proofreading, software development and useful discussions.

References

1. Takemura, S.Y., et al.: A visual motion detection circuit suggested by Drosophila connectomics. Nature **500**(7461) (2013) 175–181
2. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI, IEEE Transactions on **33**(5) (2011) 898–916
3. Ladicky, L., Russell, C., Kohli, P., Torr, P.H.S.: Associative hierarchical crfs for object class image segmentation. In: ICCV. (2009)
4. Krahenbuhl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: NIPS. (2011)
5. Funke, J., Andres, B., Hamprecht, F., Cardona, A., Cook, M.: Efficient automatic 3d-reconstruction of branching neurons from em data. In: CVPR. (2012)
6. Vazquez-Reina, A., Gelbart, M., Huang, D., Lichtman, J., Miller, E., Pfister, H.: Segmentation fusion for connectomics. In: ICCV. (2011)
7. Chklovskii, D.B., Vitaladevuni, S., Scheffer, L.K.: Semi-automated reconstruction of neural circuits using electron microscopy. Current Opinion in Neurobiology **20**(5) (2010) 667 – 675
8. Jain, V., Turaga, S.C., Briggman, K., Helmstaedter, M.N., Denk, W., Seung, H.S.: Learning to agglomerate superpixel hierarchies. In: NIPS 24. (2011) 648–656
9. Andres, B., Köthe, U., Helmstaedter, M., Denk, W., Hamprecht, F.: Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. Pattern Recognition **5096**(15) (2008) 142–152
10. Andres, B., Kroeger, T., Briggman, K., Denk, W., Korogod, N., Knott, G., Koethe, U., Hamprecht, F.: Globally optimal closed-surface segmentation for connectomics. In: ECCV. (2012)
11. Beucher, S., Meyer, F.: The Morphological Approach to Segmentation : The Watershed Transformation. Mathematical Morphology in Image Processing (1993) 433–481
12. Jain, V., Bollmann, B., Richardson, M., Berger, D., Helmstaedter, M., Briggman, K., Denk, W., Bowden, J., Mendenhall, J., Abraham, W., Harris, K., Kasthuri, N., Hayworth, K., Schalek, R., Tapia, J., Lichtman, J., Seung, H.: Boundary learning by optimization with topological constraints. In: CVPR. (2010)
13. Ciresan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS. (2012)
14. Nunez-Iglesias, J., Kennedy, R., Parag, T., Shi, J., Chklovskii, D.B.: Machine learning of hierarchical clustering to segment 2d and 3d images. PLoS ONE **8**(8) (08 2013)
15. Parag, T., Plaza, S., Scheffer, L.: Small smaple learning of superpixel classifiers for EM segmentation. In: MICCAI. (2014)
16. Lucchi, A., Smith, K., Achanta, R., Knott, G., Fua, P.: Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features. Medical Imaging, IEEE Transactions on **31**(2) (2012) 474–486
17. Breiman, L.: Random forests. Machine Learning **45**(1) (October 2001) 5–32
18. Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.A.: "ilastik: Interactive learning and segmentation toolkit". In: ISBI. (2011)
19. Meila, M.: Comparing clusterings by the variation of information. In: COLT'03. (2003) 173–187
20. Rand, W.M.: Objective Criteria for the Evaluation of Clustering Methods. Journal of the American Statistical Association **66**(336) (1971) 846–850