

# Neural Circuit Reconstruction from Electron Microscopy Images

C. Jones, T. Liu, M. Seyedhosseini, T. Tasdizen

## Connectomics:

- Motivation
  - Brain circuit models
  - Understanding wiring defects
- Electron microscopy
  - High resolution
  - High throughput techniques for large volumes (~20 Tb)
- Dense reconstruction
  - Segment each neuron in the volume
  - Find synapses between the segmented neurons
  - Manual segmentation is slow

## Multi-scale contextual model:

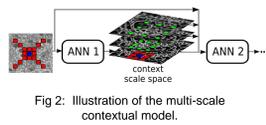


Fig 2: Illustration of the multi-scale contextual model.

## Multi-class multi-scale contextual model:

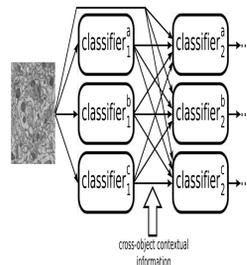


Fig 4: Illustration of the multi-class contextual model.

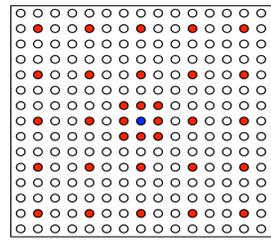


Fig 3: Sparse sampling vs multi-scale sampling.

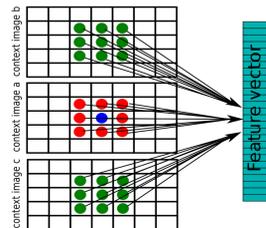


Fig 5: The multi-class feature pooling scheme.

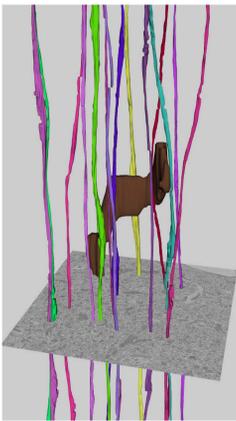


Fig 1: 3D neuron reconstruction.

## Partial Differential Equation (PDE) Processing

- The goal with this step is to close gaps that were left in membrane labels after learning.
- Closing gaps in the membrane improves the quality and usability of the resulting label when doing connectomics
- Some induced oversegmentation can be removed by using the watershed merge tree and boundary classifier

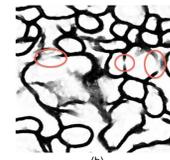
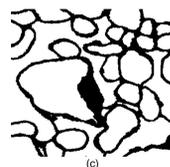
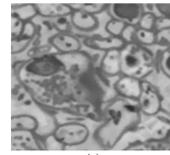


Fig. 8: Example of (a) original EM image, (b) ground truth labels, and (c) initial learning stage output with membrane gaps circled in red

## PDE Update Equation

$$\partial f = -\eta \lambda_1 + \chi \lambda_2 + \alpha \left[ \nabla f \cdot \frac{\nabla f}{|\nabla f|} + \beta \nabla f \cdot \nabla G \right]$$

1. Growth terms.  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of the derivative matrix.  $\lambda_1$  darkens the membrane and  $\lambda_2$  causes growth at terminal points such as gaps in the membrane
2. Area term. This term results in a minimization in the area of each cell enforcing smooth boundaries.
3. Boundary term.  $G$  is created from the gradient of the original image resulting in this term enforcing the membrane edges following edges in the original image

## Representation of the Result

- This method is sensitive to the number of iterations
- We replace each membrane pixel with the number of iterations required for that pixel to cross an intensity threshold divided by the total number of iterations
- The final result can either be thresholded by a smart user to provide the best segmentation or further improved upon by performing region merging classification

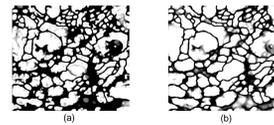


Fig. 9: Example of (a) result before replacement, and (b) result after replacing with the number of iterations

## Membrane Gap Closing Example

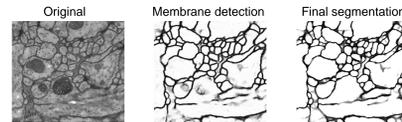


Fig. 10: PDE Results

## Watershed Merge Tree and Boundary Classifier

- Watershed transform generates initial over-segmentations and region merging hierarchy.
- Watershed merge tree: representation of region merging order.
- Boundary classifier:
  - Predict about each merge/split.
  - Random forest classifier with 141 features (geometry/intensity/texture/merge saliency).

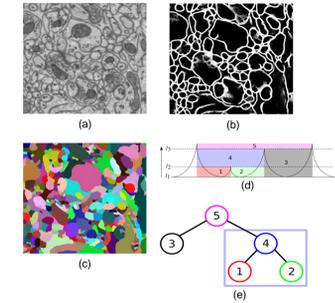


Fig. 15: Example of (a) original EM image, (b) membrane detection, (c) initial watershed over-segmentation, (d) region merging with water level rising and (e) watershed merge tree.

## Resolving Merge Tree

- Consistency constraint:
  - Any pixel should be labeled only once.
  - Once a node is selected, its ancestors and descendants must be removed.
- Node potential:
  - Probability that a node does not merge with its sibling and its children merge.
  - In Fig. 2 (b),  $P_{e,8} = (1 - P_{e,8}) P_{1,2}$ .
- Resolving merge tree via greedy optimization:
  - Pick the most potential node;
  - Remove its ancestors and descendants;
  - Repeat until no nodes are left.

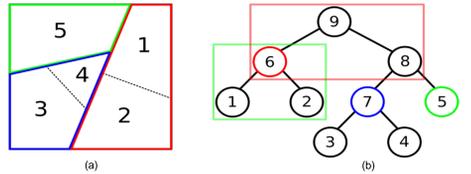


Fig. 16: Illustration of how (a) final segmentation is acquired by (b) resolving a merge tree.

## Region Merging Examples

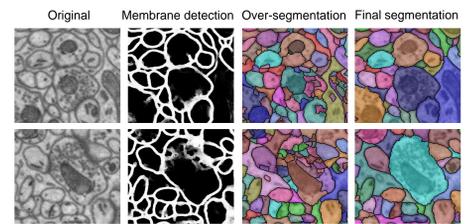


Fig. 17: Region merging results of two image regions (zoomed in).

## Results (mitochondria and synapse segmentation)

### Mouse neuropil (SBSFEM)

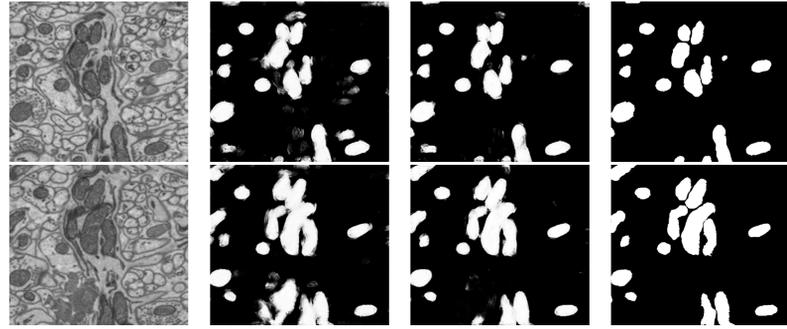


Fig 6: Results for mitochondria segmentation. (a) input image, (b) multi-scale model, (c) MCMS model, and (d) groundtruth image.

### Drosophila first instar larva ventral nerve cord (SSTEM)

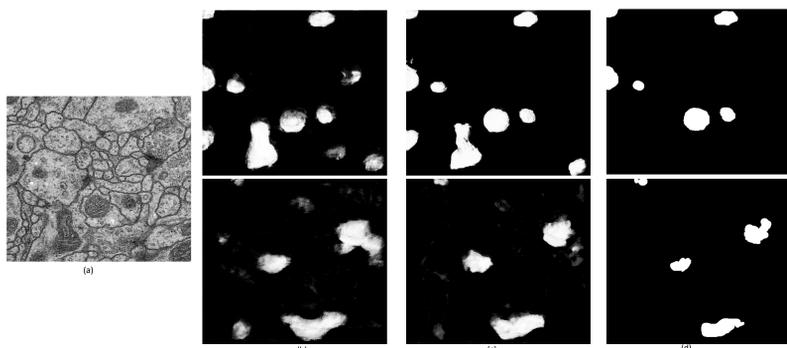


Fig 7: Results for mitochondria and synapse segmentation. (a) input image, (b) multi-scale model, (c) MCMS model, and (d) groundtruth image.

## Semi-Automatic Segmentation Method

We use a sparsely labeled image as a guide for using a min-cost path finding algorithm to create the membrane structure

### Guided Sparse Labeling

- Overlay gridlines on the original image
- User indicates where gridlines cross the cell membranes

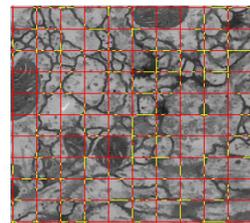


Fig 11: Example of grid labeling of cell membranes. Red shows unlabeled gridlines and yellow shows labeled grid lines.

### Min-cost Path Finding

- Use Dijkstra's algorithm for the path finding
- Computed the min-cost path between all pairs of membrane for a given grid square
- Cost function is such that pixels near in intensity to the labeled membrane will have a low cost and those further away will have a higher cost.

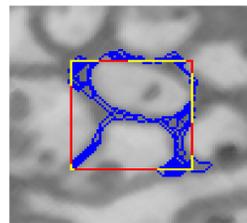


Fig 12: Example of min-cost path finding result for a single grid square

### Representation of the result

- Merge multiple paths along the same membrane using morphological processing
- Replace binary membrane labels with original intensity labels for removal of inaccurate paths through thresholding

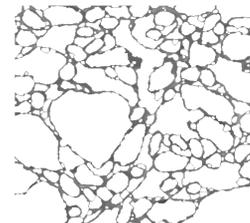


Fig 13: Example of grid labeling of cell membranes. Red shows unlabeled gridlines and yellow shows labeled grid lines.

## Semi-Automatic Segmentation Example

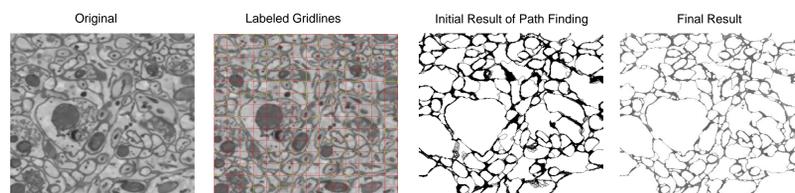


Fig. 14: Semi-automatic segmentation results for an example image.

## Results (membrane detection)

### Mouse neuropil (SBSFEM)

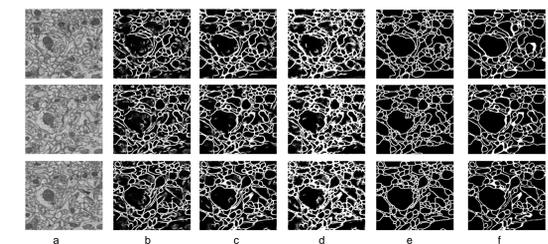


Fig 18: Results for membrane detection. (a) input image, (b) ANN series, (c) Multi-scale contextual model, (d) PDE post processing, (e) water shed merge tree, and (f) groundtruth image.

### Drosophila first instar larva ventral nerve cord (SSTEM)

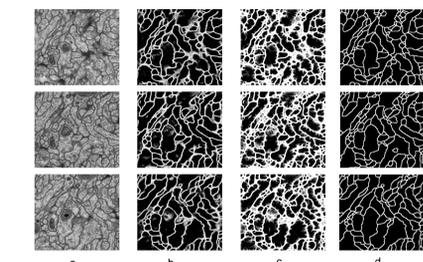


Fig 19: Results for membrane detection. (a) input image, (b) Multi-scale contextual model, (c) PDE post processing, and (d) water shed merge tree.

Table 1. Performance of the multi-scale contextual model and post-processing methods (pde + watershed merge tree) for the mouse neuropil SBSFEM dataset.

Method	Training		Testing	
	Rand Error	Pixel Error	Rand Error	Pixel Error
Multi-scale Contextual Model	0.2551	0.0512	0.2413	0.0510
Post-processing	0.1274	0.0716	0.1538	0.0745

Table 2. Testing performance of the multi-scale contextual model and post-processing methods (pde + watershed merge tree) for the Drosophila VNC sSTEM dataset.

Method	Training		Testing	
	Rand Error	Pixel Error	Rand Error	Pixel Error
Multi-scale Contextual Model	0.2084	0.0527	0.1312	0.0752
Post-processing	0.0378	0.0599	0.0770	0.1026

## Collaborators:

National Center for Microscopy and Imaging Research at UCSD



## Acknowledgment

This work was supported by NIH 1R01NS075314-01 (TT,MHE) and NSF IIS-1149299 (TT). We thank the Cardona lab at HHMI Janelia Farm for providing drosophila VNC dataset.