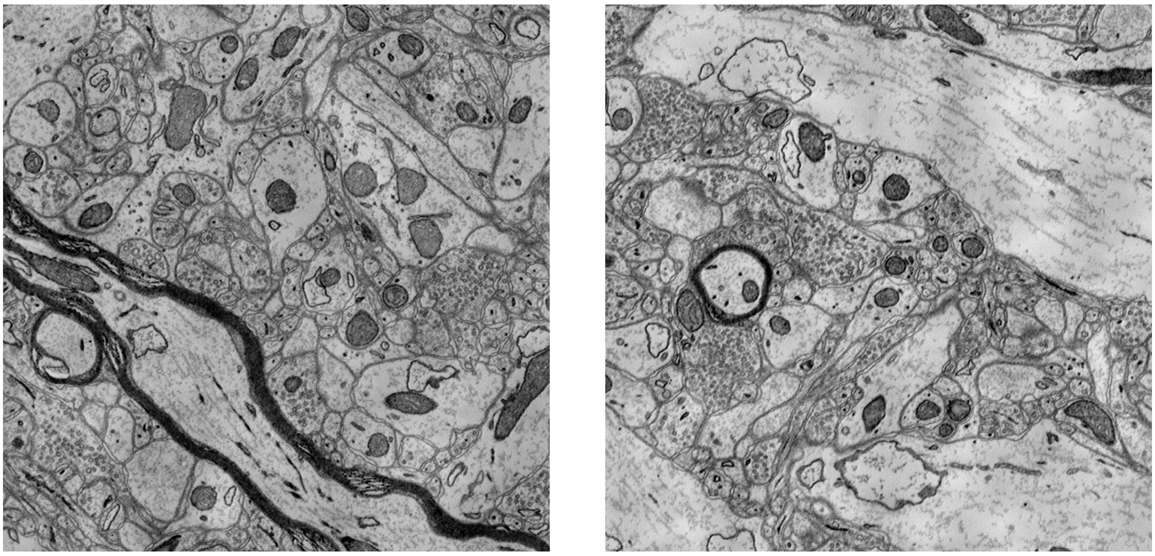
3-D Reconstruction of the Connectome Using Dense Correspondence

Vanessa Tan (Computer Science '14)

CS 91r Report, Fall 2013

1 Introduction

With recent advances in electron microscopy (EM), new technologies have enabled the production of high-resolution images of thinly sectioned brain tissue. Manually reconstructing a map of neural circuitry (*connectome*) from these images, however, has become a major bottleneck, given the magnitude of new datasets. The development of reliable, automatic algorithms for segmenting and reconstructing neurons in 2D scans would have enormous implications for our understanding of learning and memory. Thus, the overall objective of my project is to examine the automated 3D reconstruction problem from the perspective of feature correspondence.



(a)

(b)

Figure 1: Two examples of EM image slices.

1.1 Datasets

The data utilized consists of EM image stacks of thinly sliced mouse cortex (Figure 1). Due to the anisotropic data of such datasets – with significantly higher resolution in the x and y dimensions than in the z dimension – the 3D segmentation process is split into two tasks: first segmenting the individual 2D frames, and then linking the corresponding 2D objects to reconstruct neurons. A high-level overview of the workflow is shown in Figure 2.

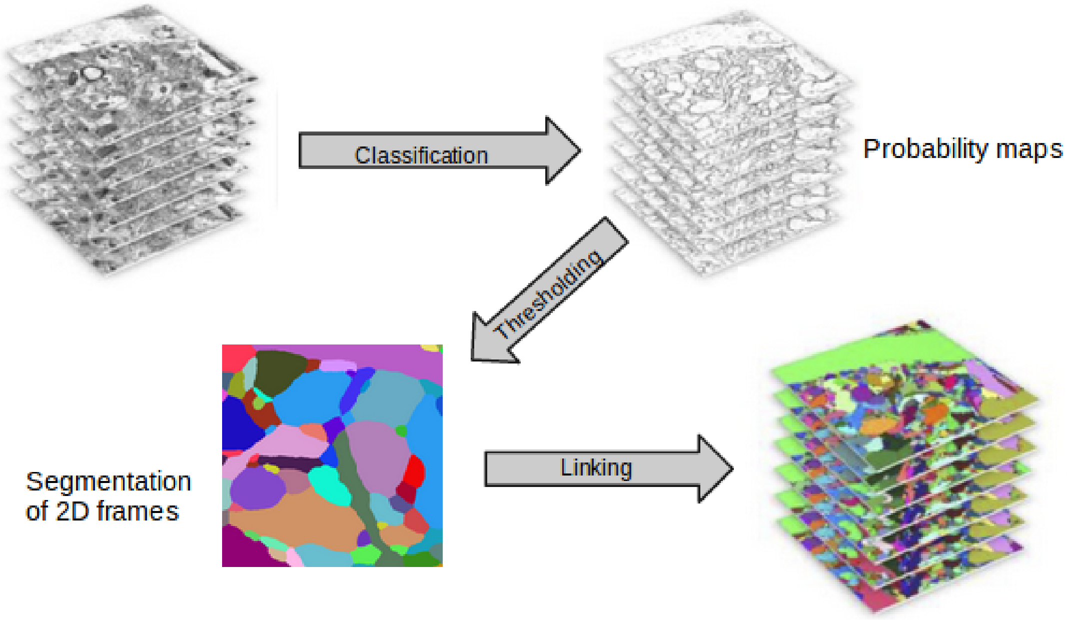


Figure 2: High-level algorithm for 3-D segmentation. Image source: [2].

The first of two datasets, referred to as here as ISBI, was publicly released by researchers at MIT in conjunction with the IEEE International Symposium on Biomedical Imaging for the “ISBI Challenge: 3D segmentation of neurites in EM images” [1]. Of the 200 1024x1024 pixel mouse cortex images included, a training set was included for 100, consisting of 2D and 3D ground-truth segmentations labeled by human experts. Sample images, along with their corresponding ground truth delineations, are shown in Figures 3 and 4.

Note that the images from the ISBI dataset lack large amounts of extracellular space separating each cell. Moving forward, however, techniques have been able to preserve the natural spacing between cells, resulting in more visually distinct objects to the human eye. Referred to

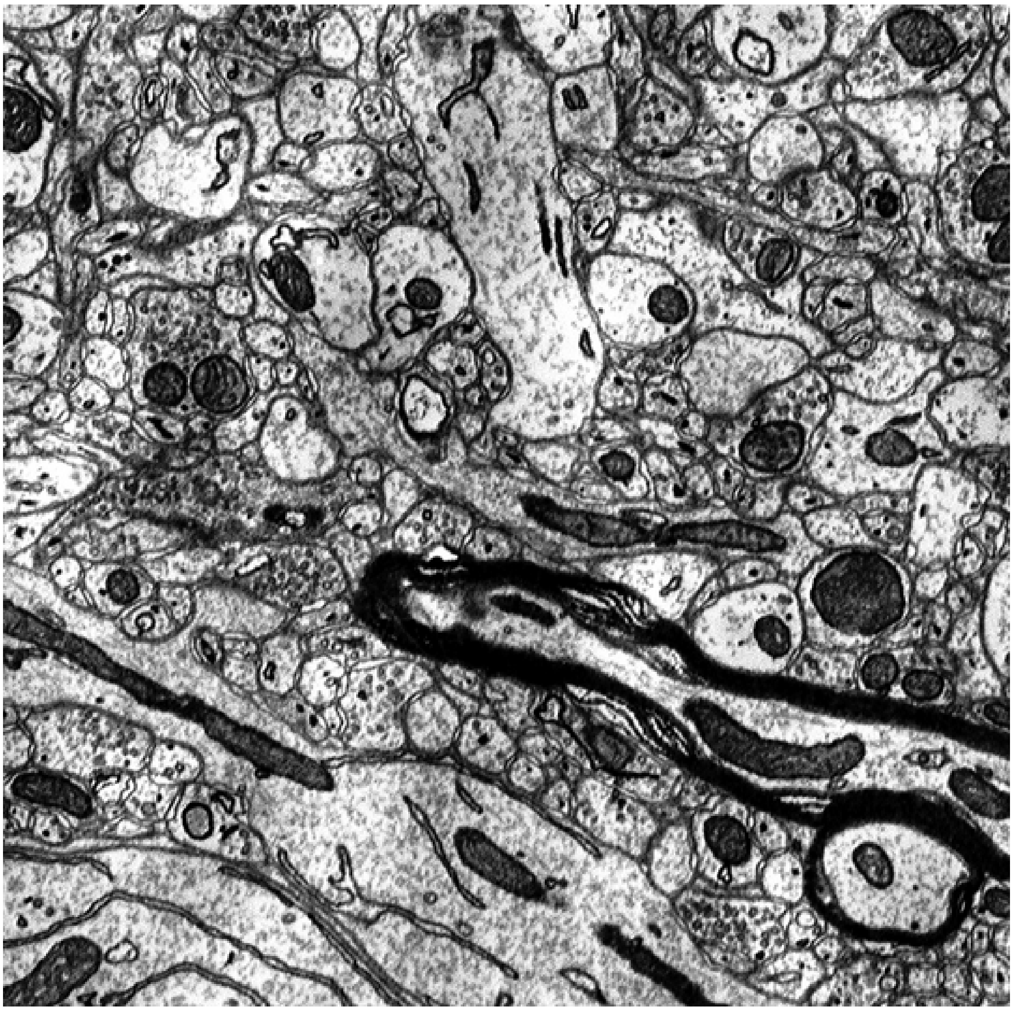
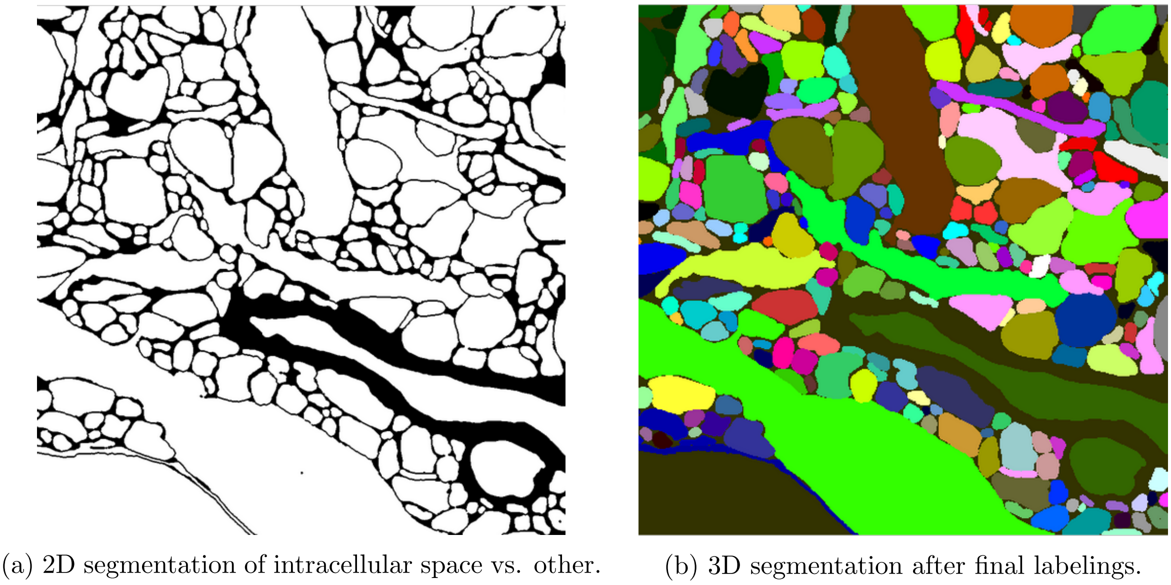


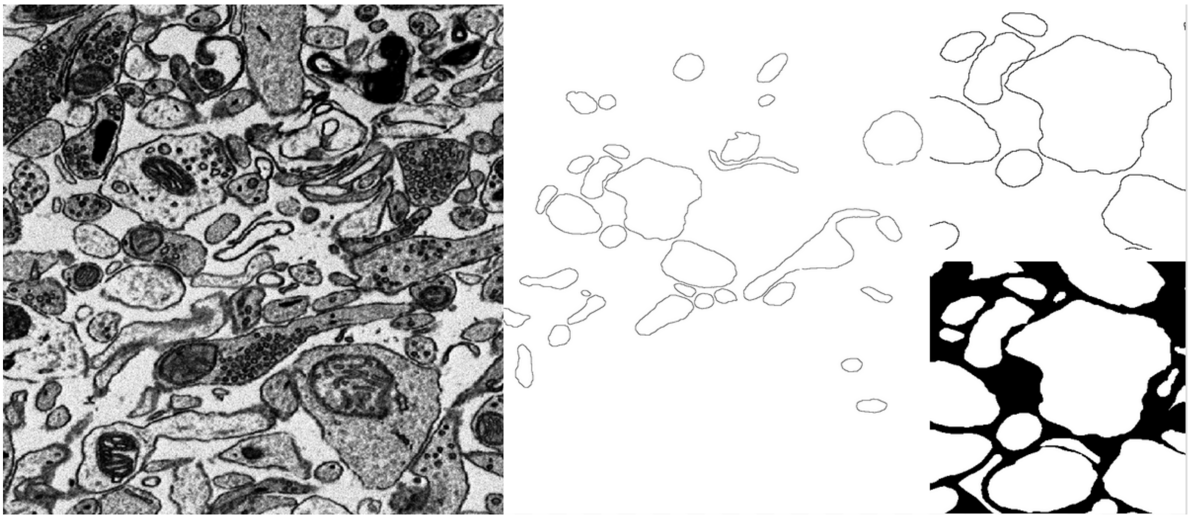
Figure 3: Sample EM image frame from the ISBI 2013 dataset.



(a) 2D segmentation of intracellular space vs. other.

(b) 3D segmentation after final labelings.

Figure 4: Ground truth labelings for the ISBI dataset.



(a) Sample ECS image frame.

(b) Partial membrane labeling with 350x350px insets.

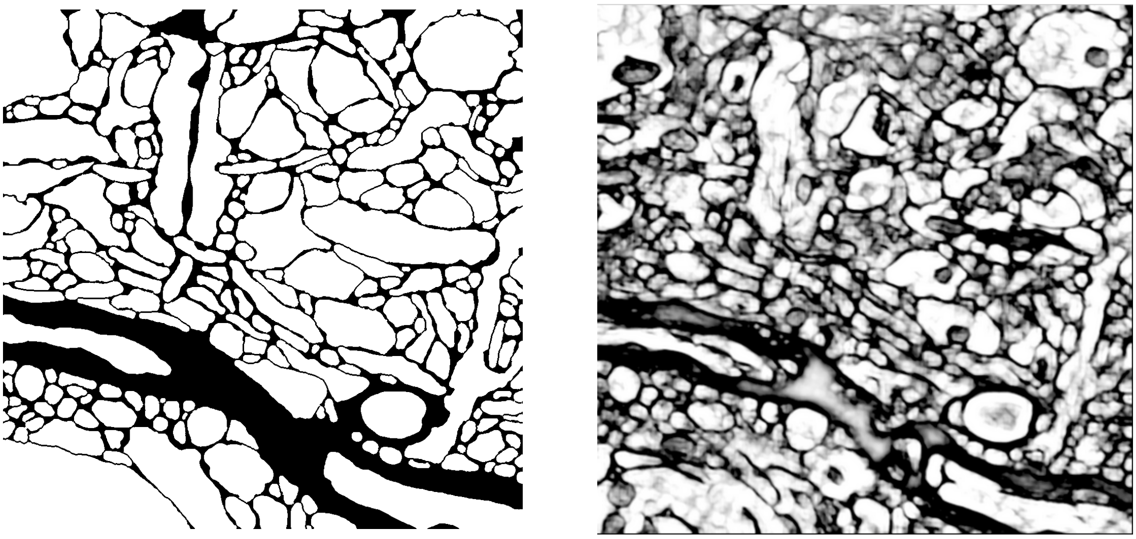
Figure 5: Sample ECS image frame and corresponding ground truth labeling.

here as *ECS*, a small dataset of 30 sparsely labeled images has been made available, including a combination of both expert and non-expert annotation. Due to the minimal amount of ground truth, only a 350x350px portion of the stack was used. An example image with annotations are shown in Figure 5. In addition, more data of this variety is being made available by the Lichtman lab, moving away from images as dense as in the ISBI dataset.

2 2D Segmentation

Segmenting a 2D frame into objects generally involves using a classifier to distinguish cell membranes. For my experiments, I trained an existing SVM using “Coxlab features” to classify each pixel. Because the SVM uses a binary classification – here, “membrane” vs. “non-membrane” – the output is a probability map where a high probability for a pixel indicates high likelihood of being intracellular space. During training, all extracellular space (which was minimal in the ISBI dataset) was conflated with “membrane” for simplicity.

A sample probability map output for an image in the ISBI dataset is shown in Figure 6. A respective example is shown for an image from the ECS dataset in Figure 7. Note that because the amount of extracellular space is nontrivial in the ECS dataset, conflating membrane and extracellular space is no longer satisfactory, as intra- and extra-cellular space are not easily



(a) Ground truth membrane labeling.

(b) Probability map output.

Figure 6: Comparison of ground truth and predicted membrane pixels for an image from the ISBI dataset.

distinguished. In the future, the classifier will need to handle three labels in order to achieve good results.

For continued experiments with the ISBI dataset, both *watershed transforms* and *dual-thresholding* methods were used to generate segmentations. Overall, watersheds performed relatively well, but no single threshold produced a satisfactory single segmentation. Using *multiple* possible 2D segmentations in a probabilistic pipeline, as promoted by [2] [3], and [4], is a potential future direction.

3 3D Linking

3.1 Feature tracking

Initially, my goal was to develop *feature tracking* algorithms for 3D linking. Feature or object tracking is frequently employed in video analysis, and consists of selecting and following the trajectory of an object from frame to frame [7]. The image stack of neuronal pathways can be interpreted as a tracking problem with “motion” of cells in a third spatial dimension instead

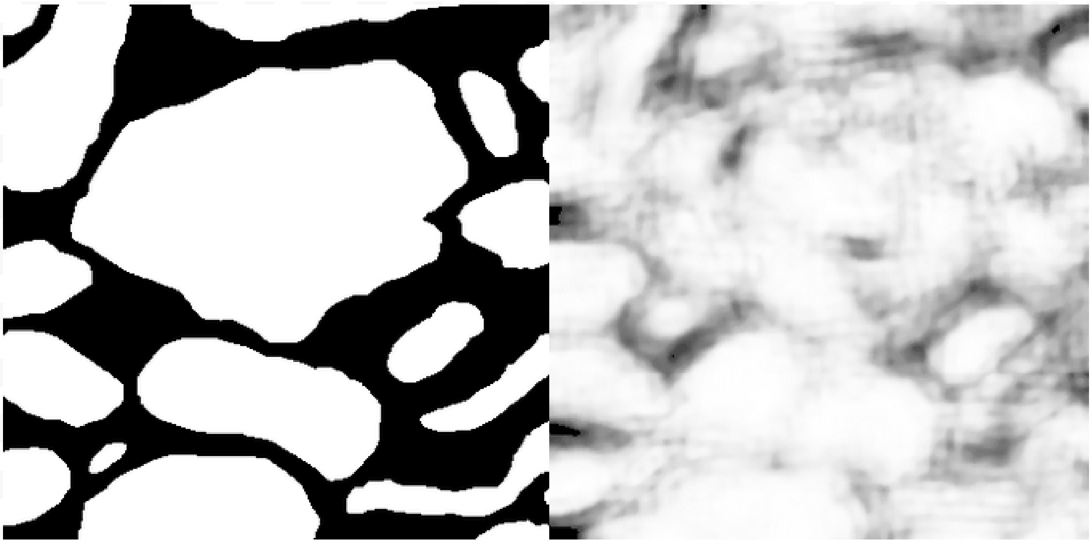
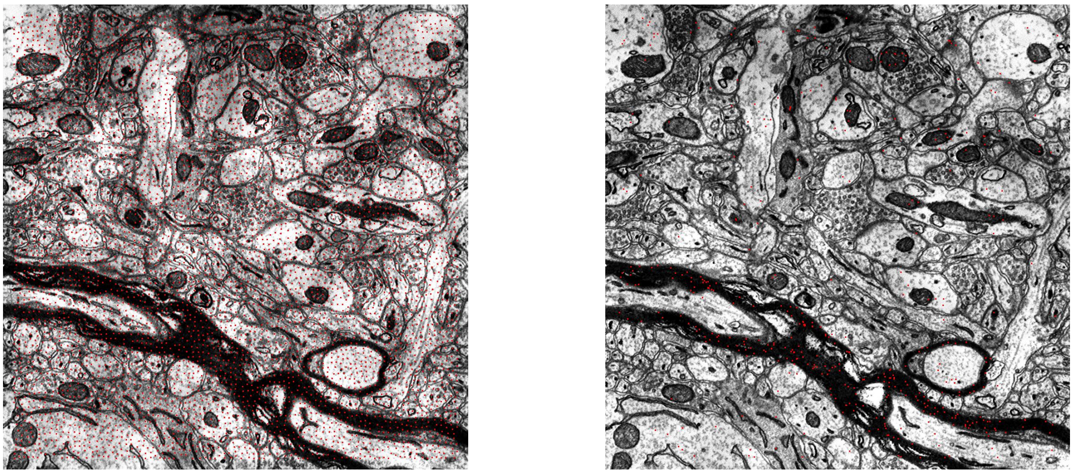


Figure 7: Comparison of ground truth and predicted membrane pixels for an image from the ECS dataset.

of motion over time. Because certain previous methods (e.g. [2]) chose to ignore intracellular texture-based information, including the presence of organelles and vesicles, this feature tracking approach aimed to take advantage of such information.

I began by adapting the *KLT feature tracking* algorithm [8], which relies on spatial intensity information to detect interest points. Additionally, I experimented with a variation of this tracking framework using SIFT feature vectors for point detection [9]. For both methods, I selected features to track for each pair of frames, computing how frequently the correspondences correctly linked 2D components belonging to the same 3D object (Figure 8). While both methods performed comparably well linking most objects, they performed poorly with respect to smaller objects, achieving no greater than 65% accuracy on the bottom 25% of objects ranked by size. This posed a significant challenge, since many elongated, thin structures (known as "dendritic spine necks") appear in the data.



(a) Interest points identified by KLT tracker.

(b) Tracked points from subset of frame (a).

Figure 8: KLT-tracked correspondences in frame (b) for a subset of points from frame (a).

3.2 Optical Flow and SIFT Flow

Initial experiments with KLT revealed that tracking of small objects was maximized by increasing the density of tracked keypoints, without greatly sacrificing overall accuracy. As a result, I turned to approaches for *dense correspondence*. *Optical flow* algorithms, rather than selecting a few keypoints to track, compute the pixelwise motion of the entire image. First, we find the flow field that minimizes distance in terms of intensity between each pair of frames. This results in a field of displacement vectors used to create correspondences between pixels in pairs of frames.

$$\begin{aligned} E(\mathbf{w}) = & \sum_{\mathbf{p}} \min\Big(\big\|s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}(\mathbf{p}))\big\|_1, t\Big) + \\ & \sum_{\mathbf{p}} \eta\Big(|u(\mathbf{p})| + |v(\mathbf{p})|\Big) + \\ & \sum_{(\mathbf{p},\mathbf{q})\in\varepsilon} \min\Big(\alpha|u(\mathbf{p}) - u(\mathbf{q})|, d\Big) + \min\Big(\alpha|v(\mathbf{p}) - v(\mathbf{q})|, d\Big) \end{aligned}$$

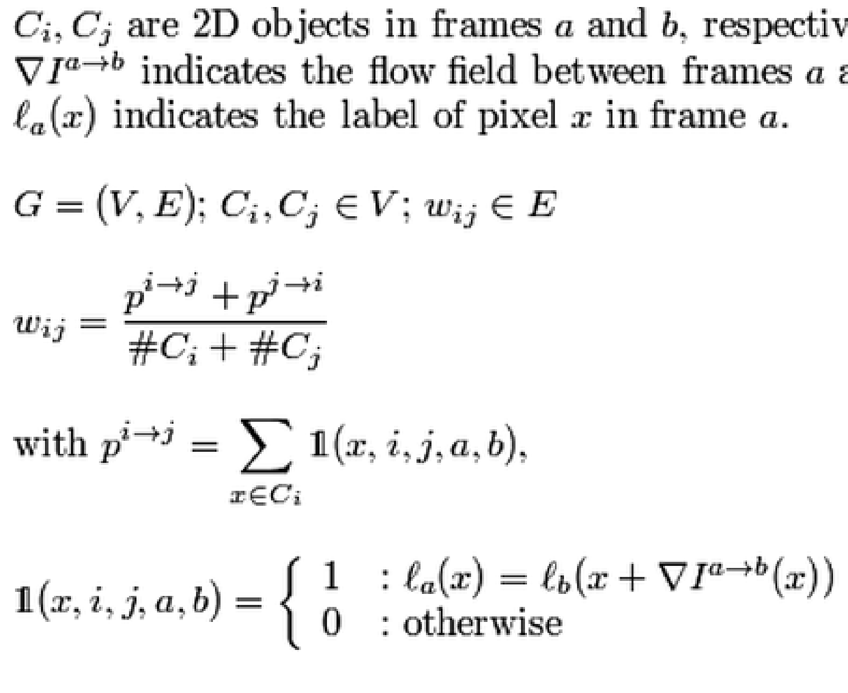
Figure 9: Energy function minimized by SIFT flow, where s_1 and s_2 give the SIFT vectors of a given pixel p; η is the small displacement constant; α is a smoothness function; and w is the flow field.

Utilizing a variation of this method, *SIFT flow* [6], achieved greater correspondence accuracy than tracking methods and did not miss small structures. Like optical flow, SIFT flow optimizes for a flow field that minimizes distance between two images, but uses SIFT features rather than

RGB or intensity values (Figure 9).

3.3 Global Graphical Linking Approach

Using SIFT flow-based correspondences as indicators for links between frames, I implemented and performed initial testing on a graphical model that determines 3D objects from 2D components. At a high level, the graph includes an edge between every 2D object in one frame and every 2D object in the adjacent frame; the edge weights between two objects are determined by the proportion of pixelwise correspondences connecting those two objects. Edge weights exceeding a threshold are deemed connections, and the graph is appropriately colored so that objects connected by an edge have the same color. For a more detailed description of the edge weight computations, see Figure 10.



C_i, C_j are 2D objects in frames a and b, respectively;
$\nabla I^{a \to b}$ indicates the flow field between frames a and b;
$\ell_a(x)$ indicates the label of pixel x in frame a.

$$G = (V, E);\ C_i, C_j \in V;\ w_{ij} \in E$$

$$w_{ij} = \frac{p^{i \to j} + p^{j \to i}}{\#C_i + \#C_j}$$

$$\text{with } p^{i \to j} = \sum_{x \in C_i} \mathbb{1}(x, i, j, a, b),$$

$$\mathbb{1}(x, i, j, a, b) = \begin{cases} 1 & : \ell_a(x) = \ell_b(x + \nabla I^{a \to b}(x)) \\ 0 & : \text{otherwise} \end{cases}$$

Figure 10: Equations for computing edge weights (w_{ij}).

Experiments. Using ground-truth 2D segmentations for 100 ISBI frames, I tested this method using a held-back portion for the validation set to tune the thresholds. Figure 11 displays the results of these experiments through a stack of 50 frames and a stack of 75 frames, using four metrics: variation of information, split error, merge error, and “rand error” as defined by the ISBI challenge [1].

# frames	Rand Err	VI	Split Err	Merge Err
50	0.2577	3.634	6.574	0.258
75	0.458	1.862	4.556	0.542

Figure 11: Results of algorithm for 25 and 75 frames.

In addition, Figure 12 displays visualized results from these experiments each showing a subset of the reconstructed neurons.

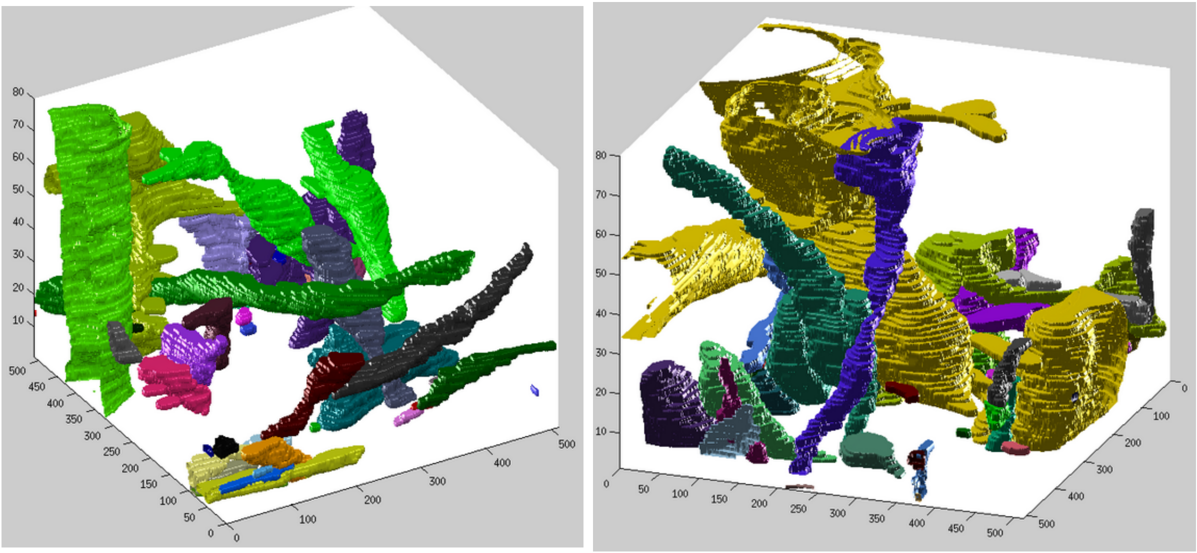


Figure 12: Two partial visualizations of reconstructed neurites.

3.4 Moving Forward: Feature Flow and Learned Graph Weights

Future work will focus on augmenting this algorithm in two ways:

1. *Better feature vectors.* The SIFT flow library is currently being modified to replace SIFT features with Cox lab features, which have been successful for various types of classification problems.

2. *Learned edge weights.* Currently, the edge weights are thresholded based on a validation set. However, there is much room to apply machine learning techniques, based on SIFT

flow data and additional features, to learn whether an edge should be part of the graph.

Finally, given the availability of more thoroughly annotated ECS datasets, further work on the 2D segmentation step will soon be possible.

4 Acknowledgements

I would like to thank Prof. David Cox for providing this research opportunity, and Walter Scheirer for his continuing mentorship throughout the semester!

References

[1] Shaar, Nadar. "SNEMI3D: 3D Segmentation of Neurites in EM Images." *IEEE International Symposium on Biomedical Imaging*, 2013. Web. 8 Nov. 2013.

[2] Kaynig, Verena, Amelio Vazquez-Reina, Seymour Knowles-Barley, Mike Roberts, Thouis R. Jones, Narayanan Kasthuri, Eric Miller, Jeff Lichtman, and Hanspeter Pfister. "Large-scale automatic reconstruction of neuronal processes from electron microscopy images." *arXiv preprint arXiv:1303.7186* (2013).

[3] Vazquez-Reina, Amelio, Michael Gelbart, Daniel Huang, Jeff Lichtman, Eric Miller, and Hanspeter Pfister. "Segmentation fusion for connectomics." In *2011 IEEE International Conference on Computer Vision (ICCV)*, pp. 177-184. IEEE, 2011.

[4] Funke, Jan, Bjoern Andres, Fred A. Hamprecht, Albert Cardona, and Matthew Cook. "Efficient automatic 3D-reconstruction of branching neurons from EM data." In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 1004-1011. IEEE, 2012.

[5] Kaynig, Verena, Thomas J. Fuchs, and Joachim M. Buhmann. "Geometrical consistent 3D tracing of neuronal processes in ssTEM data." In *Medical Image Computing and Computer-Assisted Intervention* 2010, pp. 209-216. Springer Berlin Heidelberg, 2010.

[6] Liu, Ce, Jenny Yuen, and Antonio Torralba. SIFT flow: Dense correspondence across scenes and its applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 33.5, pp. 978-994, 2011.

[7] Yilmaz, Alper, Omar Javed, and Mubarak Shah. Object tracking: A survey. In *ACM Computing Surveys (CSUR)* 38, no. 4.13, 2006.

[8] Shi, Jianbo, and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593-600, 1994.

[9] Lowe, David G. Distinctive image features from scale-invariant keypoints. In *International Journal of Computer Vision* 60, no. 2, pp. 91-110, 2004.