

Automatically Improving 3D Neuron Segmentations for Expansion Microscopy Connectomics

by Albert Gerovitch

Abstract

Understanding the geometry of neurons and their connections is key to comprehending brain function. This is the goal of a new optical approach to brain mapping using expansion microscopy (ExM), developed in the Boyden Lab at MIT to replace the traditional approach of electron microscopy. A challenge here is to perform image segmentation to delineate the boundaries of individual neurons. Currently, however, there is no method implemented for assessing a segmentation algorithm's accuracy in ExM. The aim of this project is to create automated assessment of neuronal segmentation algorithms, enabling their iterative improvement. By automating the process, I aim to devise powerful segmentation algorithms that reveal the "connectome" of a neural circuit. I created software, called SEV-3D, which uses the pixel error and warping error metrics to assess 3D segmentations of single neurons. To allow better assessment beyond a simple numerical score, I visualized the results as a multilayered image. My program runs in a closed loop with a segmentation algorithm, modifying its parameters until the algorithm yields an optimal segmentation. I am further developing my application to enable evaluation of multi-cell segmentations. In the future, I aim to further implement the principles of machine learning to automatically improve the algorithms, yielding even better accuracy.

Introduction

Studying the mechanisms of the brain is the central goal of neuroscience. Currently, however, neuroscientists lack the crucial ability to visualize brain cells, or neurons, in great detail. Recently, a new approach called Expansion Microscopy (ExM) [1] was introduced, allowing us to obtain images of the brain down to the spatial resolution of ~60 nanometers using conventional optical microscopes. The ability to use optical microscopes represents a key improvement over previous approaches based on electron microscopy, because they can capture multiple fluorescent colors simultaneously, and also can resolve features in three dimensions without the need to slice the brain tissue into nanoscale sections.

It is critical to understand how neurons are connected, as this would help explain how they interact, and would ultimately uncover how the brain functions. This requires not only data collection to obtain raw images of the neurons, but also a computational approach to extract individual cell shapes and connections from these images. My goal is to develop an optimal approach to extracting neuron shapes from ExM optical images. This paper describes the first step in this process: focusing on implementing methods to automatically assess and improve the quality of such image segmentations.

Researchers have developed several preliminary segmentation algorithms to isolate neurons in expansion microscopy images [2]. Currently, however, there is no method implemented for assessing segmentation algorithms' ability to capture entire neurons accurately, without omitting important features or adding extraneous areas. Even small errors could cause large-scale distortions. For example, a few errors in the shape of a long neural wire could misrepresent many downstream connections.

The strategy is to develop an assessment method with a closed loop between the human user and the segmentation algorithm. Such a method would evaluate the algorithm's segmentation of the raw data, modify algorithm parameters to improve the segmentation, and yield an optimal segmentation for the user. The remainder of the segmentation could be completed by a human, as the task would have become much

easier. With additional programming techniques, the algorithm would automatically self-improve, automating the entire process (as seen in **Figure 1**).

I have created a Java program, called “Segmentation Evaluation and Visualization in 3D” (SEV-3D), which takes the first steps in this process. It uses several standard image comparison techniques to score segmentation algorithms and yield an optimal segmentation based on a given set of parameters. I use two metrics for evaluation: pixel error and warping error. My software takes ground truth and a proposed segmentation as input, runs the selected metric to compare the two images, and outputs the numerical score, as well as a multilayered image to visualize potential errors in the 3D segmentation. The program runs in a closed loop with a segmentation algorithm, modifying its parameters until the algorithm yields an optimal segmentation. The goal of this program is to find the combination of parameters that yields the best neuron segmentation. Though SEV-3D is able to process any image, so far I have focused on single-cell data. In the future, I plan to expand my study to multi-cell data and to implement machine learning techniques to automatically improve the segmentation algorithm.

1 Methods

1.1 Expansion Microscopy

Traditional light microscopy allows resolution only up to 300nm. Expansion microscopy is a new method of looking at microscopic structures, including neurons, developed in the Boyden lab [1]. It works by physically expanding tissue, which allows for large 3-dimensional images in color and at high resolutions (see **Figure 2**). 5x expansion microscopy can see up to $300/5 = 60\text{nm}$, which is 5 times better than the resolution achieved by conventional light microscopy (see **Figure 3**).

1.2 Metrics

The goal of metrics of segmentation performance is to compare segmentations from a computer algorithm with a human ground truth. This ground truth is usually generated by a human by segmenting raw data by hand. Alternatively, a data simulator could be used to generate simulated raw data and to produce its ground truth.

There are four major error types that a metric should consider: additions, deletions, splits (incorrect boundaries), and mergers (incorrect gaps). An ideal metric should tolerate minor differences, such as additions and deletions, but strongly penalize topological mistakes, such as splits and mergers. For example, a pixel addition to an object could still preserve its general shape, while if the change creates a new object or merges two objects, this could be a critical error. In the application, I focus on two evaluation metrics: pixel error and warping error.

1.2.1 Pixel error

Pixel error is generally considered to be the easiest method of comparing two images. It simply counts the fraction of pixels where the two images differ. However, it equally penalizes minor errors and topological differences.

1.2.2 Warping error

Warping error tolerates minor disagreements and strongly penalizes topological errors (see **Figure 4**). Instead of noticing only pixel differences, it focuses on entire objects in an image and evaluates the topological disagreement between them (see **Figure 5**) [3].

The algorithm for warping a binary image L^* onto a binary image T , based on mask image M , was adapted from [4]:

$warp(L^* \in B, T \in A, M \in B)$

$L := L^*$

do

```

S := simple(L) ∩ M
i := argmaxj ∈ S |ti - li|, randomly breaking ties
if |ti - li| > 0.5
    li := 1 - li
else
    return L
end

```

1.3 Raw Data

Data from a microscope without any edits is called raw data. In this project, obtaining raw data was a critical step, as it was the basis for both the ground truth and segmentations. The ground truth is created from the raw data by a human or simulation. A segmentation algorithm is then run on the raw data to propose a segmentation.

1.3.1 Single-cell Image

In this project, the focus was primarily on single-cell data. Raw data covered just one neuron, making only two possibilities for every pixel - 1 (neuron) or 0 (background). The segmentation algorithm can make just two types of mistakes: labeling a neuron as background (a false negative) or labeling background as a neuron (a false positive).

1.3.2 Multi-cell Image

The long-term aim of this project is to expand assessment capabilities to multi-cell data as well. With multi-cell data, there is more than one neuron in the raw data, making more possibilities for the labeling of every pixel. The segmentation algorithm, in addition to incorrectly distinguishing background from neuron, can also make the mistake of counting a part of one neuron as part of another.

1.3.3 Simulated Data

This project uses simulated microscopy data provided by the Boyden Lab. This data is generated by an automatic computer algorithm, without actual images from a microscope. It provides simulated raw data and its ground truth segmentation. An annotated electron microscopy volume was used to produce a simulation, using a specific mathematical model to simulate the effects of the microscope and the sample expansion on the image. Since the electron microscopy image was previously annotated by humans, there is a ground truth already available for the simulated sample. Segmentation algorithms run on the simulated data.

1.4 Segmentation Algorithms

Segmentation algorithms were provided by the Grossman Center for the Statistics of Mind at Columbia University. They interpret a raw image of a neuron, or several neurons, and return a proposed segmentation.

In this project, the SEV-3D program compares the proposed segmentation to a ground truth using the error metrics and produces a numerical score and visualization of error. Then, in a closed-loop system, the SEV-3D assessment module modifies parameters until the algorithm yields the best segmentation. In a long-term goal of this project, a machine learning module would automatically provide feedback, allowing the algorithm to learn and improve. In addition, this feedback would allow further self-improvement of the simulator to produce more realistic images.

1.5 Java Programming Language

The Java programming language was used for this project. Java's high-level capabilities for image processing were critical to the function of the application. Standard Java libraries were used to enable the manipulation of multi-layer images. The ImageJ API (source: <http://www.java2s.com/Code/Jar/i/Downloadij135jar.htm>) was used to open images and convert them to multi-dimensional arrays that could then be read and modified by the Java code.

1.6 TIFF Images

All data for this project was in 3-D, so multilayered images were required. The TIFF format allows images with many slices, which was a very effective method of storing and exporting data. However, TIFF images come in various compressions and types, so SEV-3D initially standardizes the raw data so it is compatible with the segmentation algorithm.

2 Results and Discussion

2.1 Application and Results of Metrics

The created application, “Segmentation Evaluation and Visualization in 3D” (SEV-3D), runs the pixel error and warping error metrics on a segmentation and compares the result with a ground truth. SEV-3D takes a proposed segmentation TIFF image file, a ground truth TIFF image file, a selected metric, a toggle for generating a visualization, and a range of input parameters for the segmentation algorithm (as seen in **Figure 6**). Once the program finds the parameters that yield an optimal segmentation, the results for the best segmentation are returned to the user in two formats: a numerical score and a visualization of the error.

Figure 7 is an example of raw data, **Figure 8** is an example of ground truth, and **Figure 9** is a proposed optimal segmentation. These images have multiple layers, and the scores are computed for all layers, but only one layer is shown here for comparison.

For a video demonstration of the operation of SEV-3D, see <https://youtu.be/Uu1tSK36AOk>. The video shows how the program is run, manipulating two parameters (*sigma* and *spatialDistanceUpperBound*) with two proposed values for each. Then, the video displays the assessment results after the four runs. After that, the program outputs the optimal segmentation with its score (9.700E-5) and visualizes it, going through multiple layers of the image. The visualization shows no areas with error. False negatives would have been marked blue (“Only Truth”); false positives would

have been marked green (“Only Segmentation”). The white color indicates the area where the segmentation and ground truth coincide.

2.1.1 Scores

If the user does not choose to export a visualization of the error, the program will simply output the score of the selected error metric, pixel error or warping error, on its selected optimal segmentation. A lower error score indicates a better segmentation, while a higher error score shows that there were many errors.

2.1.2 Visualization

A numerical score is not the best form of feedback for an algorithm developer. For the human, it is essential to see which parts of the image the algorithm is segmenting incorrectly.

The application has an option to export a TIFF image file with a visualization of the error. The pixel error between the two images can be seen on **Figure 10**. The warping error is presented on **Figure 11**.

While pixel error considers all pixels equally, the warping error algorithm recognizes if there is a gap in the membrane of the neuron, and does not fill the inside.

3 Illustrations

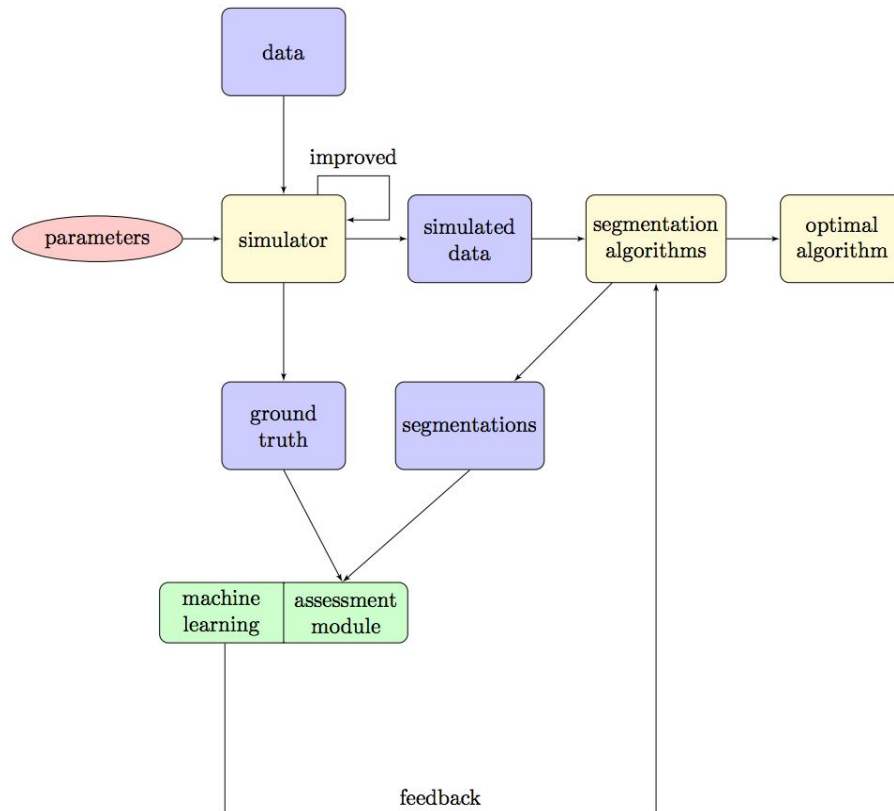


Figure 1. Diagram of Segmentation and Feedback Process. This diagram depicts the flowchart of this project. Yellow boxes represent existing programs, blue boxes represent image data, and the green box is the SEV-3D software I have created. Data and parameters are passed into the simulator, which returns ground truth and simulated data. The simulated data is passed to segmentation algorithms, which create segmentations. These segmentations and the ground truth are passed to the SEV-3D assessment module that communicates with my machine learning software, which automatically improves the segmentations algorithms. This finally yields an optimal algorithm.

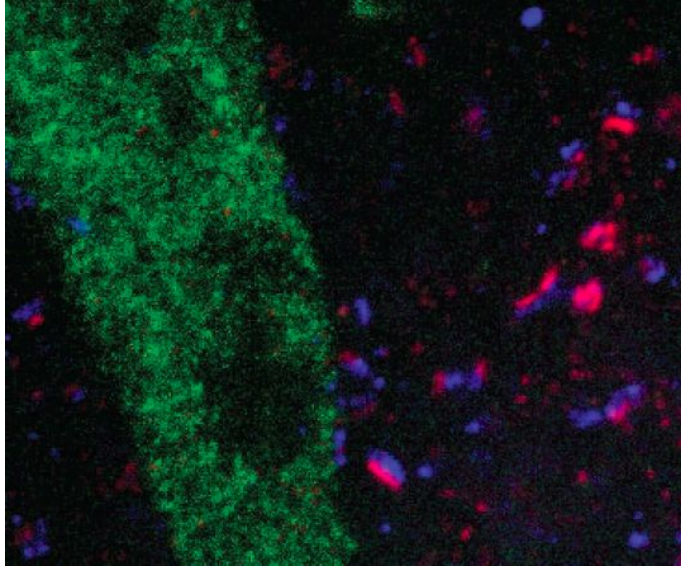


Figure 2. Close-up Expansion Microscopy Image. This is an example of a high-resolution color image obtained from zooming in on expansion microscopy data. Similar neurons were analyzed in this project (image courtesy of the Boyden lab [1]).

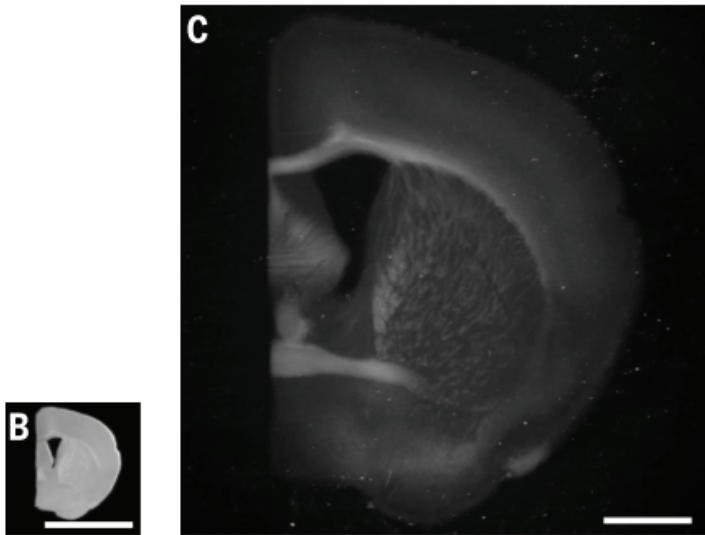
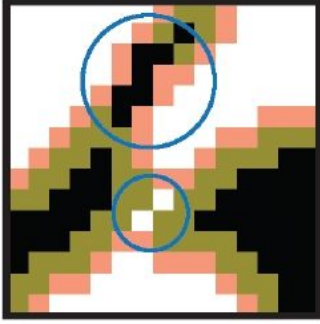


Figure 3. Light Microscopy vs. Expansion Microscopy. The B image is data from light microscopy, while the C image is from expansion microscopy. The C image is 5 times larger than the B image, as the white stripe is the same length in both, and it is much more detailed (image courtesy of the Boyden lab [1]).



- in simple points
- out simple points

Figure 4. Warping Error. Warping error penalizes topological differences, while tolerating variations in simple points. If pink pixels, or “in” simple points, are added to the image, this would not create or delete objects from the segmentation, so warping error would not penalize a segmentation algorithm if these pixels are accidentally added. Similarly, if green pixels, or “out” simple points, are removed, this would not create or delete objects, so warping error would not penalize a segmentation algorithm for deleting these pixels. However, if one of the black pixels is removed or added onto the white, a new hole or object would be created, and warping error would penalize such a mistake very strongly (image adapted from [5]).

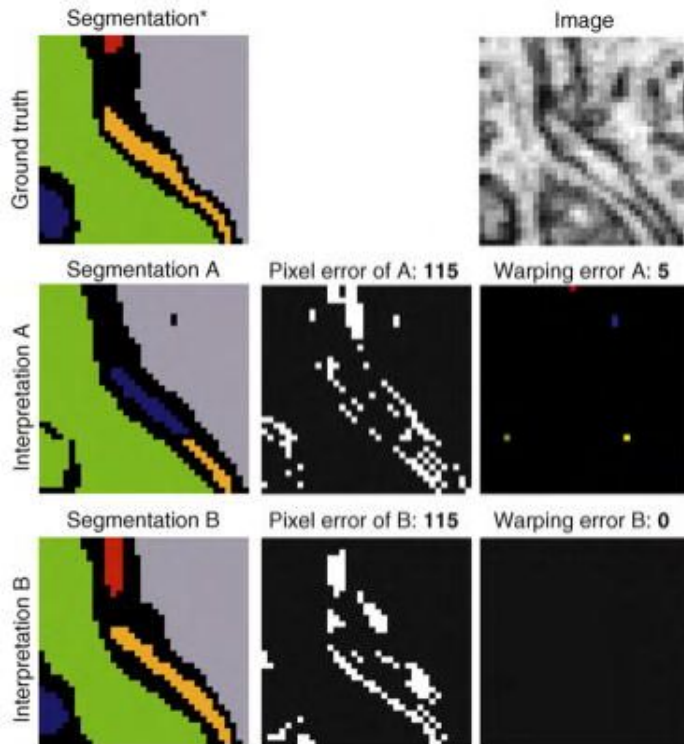


Figure 5. Comparing Segmentation Metrics. The ground truth is used to evaluate the accuracy of two segmentation algorithms (A and B) in segmenting the black and white raw image in the top right corner. In the evaluation of the segmentations created by algorithms A and B by using pixel error, they score equally. Segmentation A, however, visibly differs from the ground truth, while Segmentation B is relatively similar. Warping error, unlike pixel error, penalizes only topological errors (red = deletion, blue = addition, green = merger, yellow = split) (image adapted from [4]).

```
String groundTruthPath = "/Users/Albert/Desktop/simulation2_gt.tif";
String rawDataPath = "Users/Albert/Desktop/simulation2.tif";
String error = "Warping Error";
boolean visualize = true;
String visualizationOutput = "/Users/Albert/Desktop/visualization.tif";
Integer[] sigma = {500, 1500};
Integer[] spatialDistanceUpperBound = {25, 30};
```

Figure 6. Application Input. As input, the user defines the ground truth and segmentation file-paths, chooses an error metric, decides whether to visualize and where to output the image, and then defines the test values for two parameters (*sigma* and *spatialDistanceUpperBound* in this case).

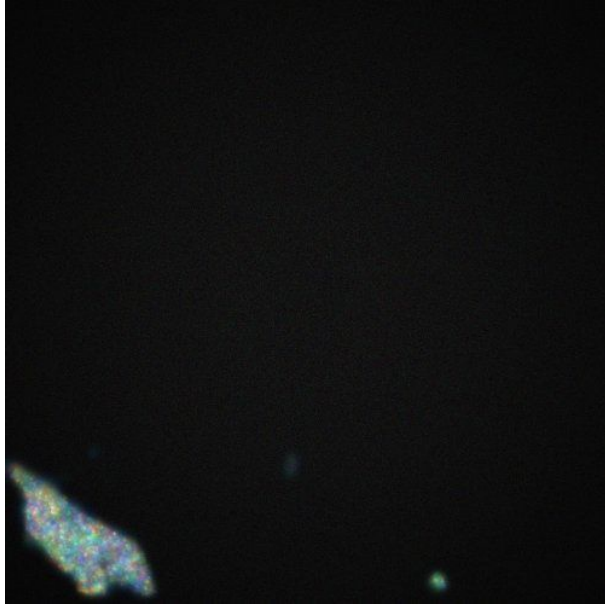


Figure 7. Raw Data. This image is an example of raw data produced by a simulator. See animation of the sequence of slice images going down a full 3-D stack at <https://www.dropbox.com/s/z0stindu0mhx114>. (Image courtesy of the Boyden lab.)



Figure 8. Ground Truth. This image is an example of ground truth produced by a simulator. See animation of the sequence of slice images going down a full 3-D stack at <https://www.dropbox.com/s/nbwsfcwctc1iyx59>. (Image courtesy of the Boyden lab.)



Figure 9. Proposed Segmentation. This image is an example of the optimal segmentation produced by SEV-3D. See animation of the sequence of slice images going down a full 3-D stack at: <https://www.dropbox.com/s/8wgovn0owe6a8ro>. (Image courtesy of the Boyden lab.)

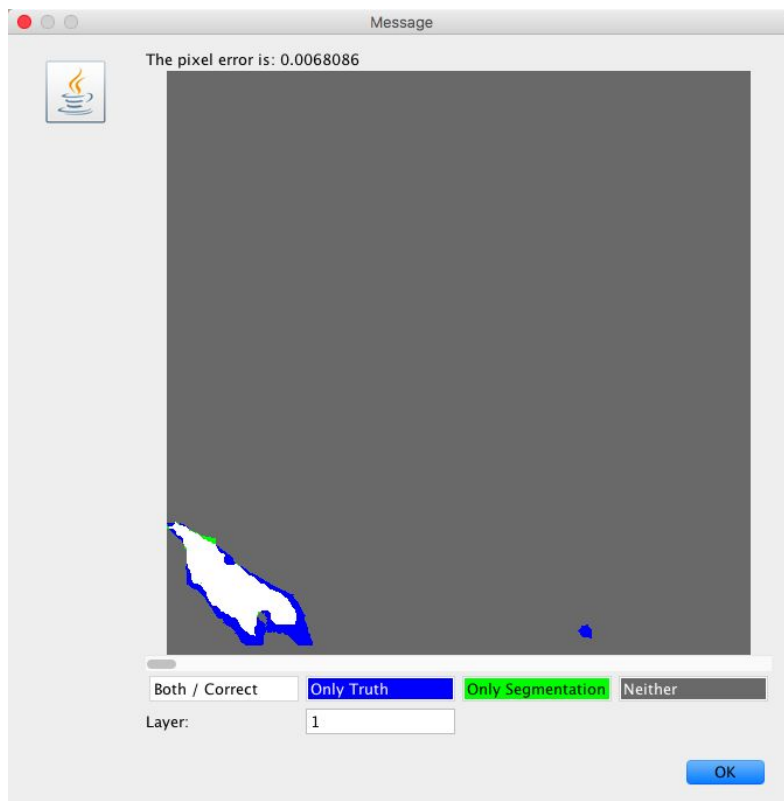


Figure 10. Pixel Error Visualization when comparing **Figure 8**, as the ground truth, to **Figure 9**, as the proposed segmentation. (Animation going down full 3-D stack:

<https://www.dropbox.com/s/atr9l2ys9kxogyt>)

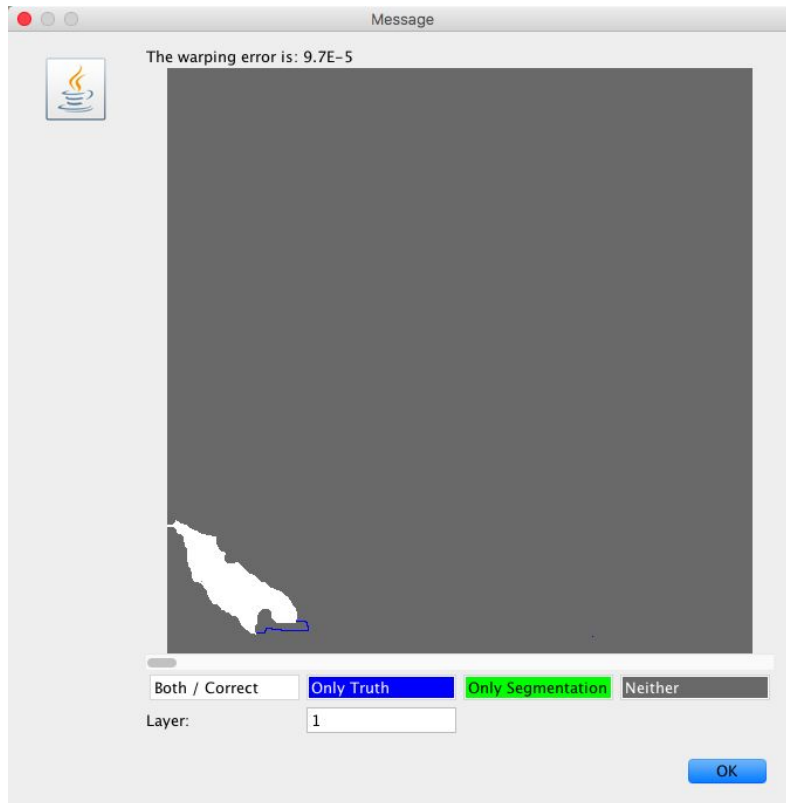


Figure 11. Warping Error Visualization when comparing **Figure 8**, as the ground truth, to **Figure 9**, as the proposed segmentation. (Animation going down full 3-D stack:

<https://www.dropbox.com/s/8e3ihlxq1jw4ucy>)

4 Conclusions and Future Work

4.1 Conclusions

Using my software, SEV-3D, I compared two error metrics, pixel error and warping error. I ran the evaluation software on a variety of simulated and actual data, and concluded that warping error gives a much more accurate representation of the algorithm's mistakes than pixel error. Warping error recognizes and strongly penalizes only topological differences, while pixel error counts all pixel differences equally. With

my software, I have created a tool for researchers to evaluate their segmentation algorithms. SEV-3D provides feedback in a visual form, allowing developers to see exactly which areas are missed by their algorithm. The software successfully runs a segmentation algorithm, evaluates single-cell segmentations, yields an optimal segmentation, and has some multi-cell capabilities that I plan to expand on later.

4.2 Future Work

In the future, I plan to further expand my closed-loop system to automatically improve the segmentation algorithm. The program would provide feedback to the developer on both multi-cell data as well as single-cell data, and would automatically determine which areas of the segmentation can be fixed by the computer. The rest of the segmentation, a relatively small part, could be completed by a human. With the use of other evolutionary programming techniques, this entire process could become fully automated. I plan to improve the operation of SEV-3D by using the principles of machine learning and letting the algorithm self-evaluate and then self-improve.

References

- [1] Chen, Fei, Paul W. Tillberg, and Edward S. Boyden. "Expansion microscopy." *Science* 347, no. 6221 (2015): 543-548.
- [2] Dr. Uygur Sümbül, Grossman Center for the Statistics of Mind, Columbia University, personal communication.
- [3] Jain, Viren, H. Sebastian Seung, and Srinivas C. Turaga. "Machines that learn to segment images: a crucial technology for connectomics." *Current opinion in neurobiology* 20, no. 5 (2010): 653-666.
- [4] Jain, Viren, Benjamin Bollmann, Mark Richardson, Daniel R. Berger, Moritz N. Helmstaedter, Kevin L. Briggman, Winfried Denk et al. "Boundary learning by optimization with topological constraints." In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2488-2495. IEEE, 2010.
- [5] "ImageJ." Topology Preserving Warping Error. Accessed September 14, 2016. http://imagej.net/Topology_preserving_warping_error.

Acknowledgements

This project was conducted at the MIT Synthetic Neurobiology Lab. Special thanks to Dr. Adam Marblestone for mentoring this project, Professor Ed Boyden for his support and guidance, and MIT PRIMES for providing this opportunity.