

Automated Segmentation of 3D Punctate Neural Expansion Microscopy Data

Zachary Steinberg

MIT PRIMES, September 2017

Abstract

*Well, IExM's like no other.
But existing techniques, we've discovered
will not work here quite right
since it's not black-and-white.
So we present one that takes color.*

The comprehensive study of multiple-neuron circuits, known as connectomics, has historically been hampered by the time-consuming process of obtaining data with perfect morphological reconstructions of neurons. Existing attempts to automate the reconstruction of synaptic connections have used electron microscope data to some success, but were limited due to the black-and-white nature of such data and the computational requirements of supervised learning. Now that multicolor data is available at 20nm resolution via Expansion Microscopy (ExM), creating an automated, reliable algorithm requiring minimal training that can process the future petabytes of neural tissue data in a reasonable amount of time is an open problem. Here, we outline an automated approach to segment neurons in a 20x expanded hippocampus slice expressing Brainbow fluorescent proteins. We first use a neural network as a mask to filter data, oversegment in color space to create supervoxels, and finally merge those supervoxels together to reconstruct the 3D volume for an individual neuron. The results demonstrate this approach shows promise to harness ExM data for 3D neural imaging. Our approach offers several insights that can guide future work.

Introduction

The human brain features billions of neurons. Each one of them is connected to several others by filament-like structures called axons, which contain synapses on their ends that send and receive electrical signals through the brain. This simple mechanism, multiplied by the 1000 connections between each of the 100 billion neurons in human brain, controls all of the complex behavior of the brain.

It's reasonable to assume that if one had direct knowledge of which neurons each neuron in the brain connected to, one would be able to model the brain's functions and study them further. The field of *connectomics* is a subfield of neurobiology that aims to learn more about the brain by studying multiple-neuron circuits. Currently, the field is in its infancy as existing methods of obtaining brain circuits have not developed far enough.

In order to obtain images and discern the locations of neurons, microscope imaging is required. Generally, in order to “image” a volume, a microscope takes multiple 2D cross-sections, which are then assembled into a 3D image. Conventional optical microscopes must deal with the diffraction limit of visible light, and thus cannot discern structures smaller than their ~300nm resolution¹. Imaging with conventional optical microscopes is therefore problematic when trying to image *dendritic spines* (one of many locations where neurons can connect to one another via synapses) as they are often are fractions of micrometers thick. One technique commonly used^{5,9} for neural imaging is electron microscopy, or “EM”, imaging (Figure 1a). EM microscopes first stain neurons with heavy metals such as osmium^{9,11} or lead¹¹, then bounce electrons off those metals in order to build up an image of where cell membranes lie.

However, this method comes with several disadvantages, chief among them its slow speed. EM imaging is very slow because “averaging tens or hundreds of thousands of single-particle images... is necessary to achieve high resolution”¹⁰. Kasthuri et al (2015)⁹ used state-of-the-art techniques to image at $9 \mu\text{m}^2/\text{s}$, a rate that would take 30 hours to image a cubic millimeter. Such an approach may work for small brain sections, but the time it would take to image an entire brain with EM would be exorbitant. Additionally, EM imaging is an all-or-nothing approach: either all cell membranes are stained, or no cell membranes are. As a result, EM results in very dense images (Figure 1a), with many neurons shown inside a small region. Finally, because EM images register either the presence or absence of reflected electrons, EM images by their very nature are black-and-white.

One promising alternative is Expansion Microscopy (ExM)³. With this method, a tissue sample to be imaged is embedded in a hydrogel, and then physically expanded in space by adding water. This method is well suited for neurons because the expansion makes dendritic spines big enough to be imaged with conventional light-based microscopes, which are much faster than electron microscopes. Thanks to recent developments¹, ExM can be iterated to obtain up to 20x expansion. As a result, ExM images are able to show a greater amount of magnified detail than EM with a higher throughput.

Additionally, because light-based microscopes image in color, it’s possible to image multiple color channels at once, which can be used to provide even more information about neurons than is possible through electron microscopes. Informally, this allows multiple data sources to be overlaid on top of one another in the same way that a computer displays color by combining a red image, a green image, and a blue image, each distinct from one another.

The possibility of multi-channel color data naturally raises the question of what additional information can be obtained using multiple channels. One common technique is to stain a neuron with multiple antibodies and then attach colored fluorescent proteins to the antibodies so they can each be tracked separately. Most approaches to multi-channel segmentation can be grouped into one of two

categories: Either they stain different biologically relevant parts of the cell, or they combine multiple colors for the same feature. Some common stains for this first approach include cell membrane stains and cytosolic stains, as well as DAPI stains which can detect the presence of DNA. Membrane stains stain the outer cell membrane, appearing as an outline of the neuron in the eventual image, while cytosolic stains stain the *cytosolic fluid* in the interior of a neuron (without staining any of the organelles, which appear as “holes” in the resulting image).

One example of the second category of multi-color stains, those that apply multiple colors for the same feature, is a technique called Brainbow². By using viral recombination to infect each neuron with a distinct combination of colors, it effectively colors each neuron a different color, allowing one to differentiate neurons simply by color (Figure 1b). Brainbow also allows for the possibility of *sparseness*: as opposed to *dense* images, which show a majority (or perhaps all) of the neurons in a given region, a sparse image only features a few of the neurons in an area, “hiding” others from view. Brainbow is able to generate sparse images by adjusting both the concentration of virus injected and the distance between the injection site and the imaged tissue, leaving many neurons unfluorescent.

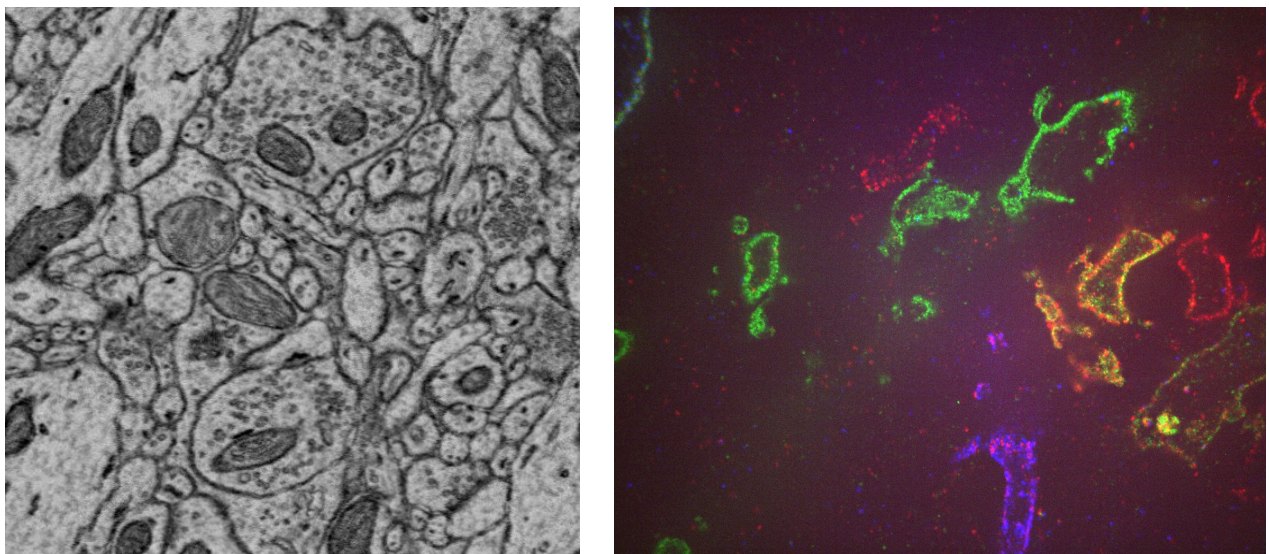


Figure 1: a) A 2D slice of EM data⁸. This image is dense: every single neuron in its volume is visible. b) A 2D slice of 20x expanded Brainbow data¹, equivalent to 8.25nm x 8.25nm of pre-expansion neural tissue. The red, green, and blue channels of the image represent

three different fluorescent proteins, appearing in different combinations in different neurons due to viral recombination. Notice how the cell membranes are punctate and spotty (exemplified by the center-top red neuron).

Segmenting 20x expanded ExM data poses many challenges. Though algorithms exist for segmenting black-and-white EM data^{5,9}, they don't take advantage of the new color information available through ExM. Additionally, the sheer size of the image – the full microscope data shown in Figure 1 takes up almost a gigabyte – means that any algorithm, which must iterate over all pixels, will take a long time to run on such big images. (Extrapolating, we remark that a cubic millimeter of 20x ExM data would take up 53.7 zettabytes.) Additionally, the light that is captured in an ExM image must come from somewhere, generally a fluorescent protein. At low levels of expansion, the points of light blend together and produce a smooth, easy-to-trace line, but at 20x expansion, the finite amount of fluorescent molecules begin to separate and neuron membranes no longer appear solid. These *punctate* membranes appear to be made of patchy blobs of color interspersed with background, which poses a problem because computer vision techniques have a hard time tracing broken lines. As an aside, we note that one could avoid this problem by downsampling until the membrane appears unbroken, but doing so would be equivalent to merely expanding to a lesser factor than 20x, and such a method would not be taking advantage of the additional level of detail that 20x expansion provides.

Segmentation, informally, is the process of starting from a microscope image, which consists of individual pixel values, and then figuring out which pixels belong to which neurons. Currently, one common technique is to have a human painstakingly trace a microscope image, slice by 2D slice, until the volume is complete. This method results in very accurate segmentation, to the point where often human-annotated data is referred to as “*ground truth*”. However, humans have a very slow throughput, requiring droves of laborers (sometimes known as “grad students”) and hours of work to annotate relatively small datasets. One recent paper⁵ from the Seung lab reported a state-of-the-art segmentation of 440 μm^3 took months of man-hours: “producing the gold standard segmentation required a total

~900 tracer hours, while producing the bootstrapped ground truth required ~670 tracer hours". At this rate, the time required to human-annotate all 1200 cubic centimeters of human brain tissue⁶ would be almost 20 times the age of the universe. In order to obtain the large volumes necessary to make connectomics feasible, any process would need to be automated.

The question "What differentiates a good segmentation from a bad segmentation?" naturally arises from such a definition. We remark that humans are very good at tracing things by eye, and therefore overlaying a segmentation transparently over the source data will reveal at a glance how accurate a segmentation is. While others have produced quantitative metrics for segmentation quality such as Rand Error and Warping Error, we concern ourselves with just two qualitative metrics that are easy to evaluate at a glance: First, a segmentation should not erroneously claim two neurons belong to the same neuron (the "Bijection Criterion"), and second, a segmentation should cleanly separate a membrane from the background (the "Separation Criterion").

Our goal, therefore, is to make an automated, high-throughput process to segment neurons.

We present a process that takes 20x ExM Brainbow data and segments it in an automated, start-to-finish process. Through trial and error and multiple iterations over the course of a year, we went through a number of techniques in order to better automate the process. Our approach used 3 steps: First we used a neural net to throw out likely non-membrane pixels, then we oversegmented the resulting membrane candidates with k-means, and finally we grouped together clusters with similar colors to produce a final segmentation.

Methods

Overview

First, we give an overview of our algorithm, and then give examples of several iterations we made on this algorithm in order to showcase some of the problems we faced and how we solved them.

Our algorithm consists of 3 main steps:

- Use a neural net as a mask to throw out likely non-membrane candidates
- Oversegment with k-means to cluster pixels with both spatial and color information
- Merge together clusters that represent the same neuron

Before we discuss the specifics of our algorithm, we note that the sheer amount of data poses problems for the runtime of segmentation algorithms. Because an algorithm to segment 20x ExM data must iterate over every pixel of gigabyte-sized images, its runtime will be very sensitive to changes in the input values. The algorithm we use for clustering, k-means, has a runtime of “ $O(nkt)$, where n , k , t equal the number of iterations, clusters, and data points”⁴. Any pre-processing that can reduce the amount of input data points by half will reduce t and therefore the overall runtime by half as well. Therefore, it is advisable to do any possible pre-processing that reduces the amount of pixels that need to be considered. In such a pre-processing pipeline, avoiding false negatives – throwing away a pixel that does belong to a neuron – is more important than avoiding false positives – incorrectly identifying a pixel as relevant, which can be eliminated in the actual processing step.

We chose to pre-process our data by using a neural net to filter out likely non-membrane candidates. Since neural nets output a continuous value that varies between two extremes, we trained a neural net to differentiate between “membrane” and “not membrane”, and then we threw away all pixels under a certain threshold to keep the pixels classified as “membrane”. We run 50x50 pixel patches through the neural net to output a mask that ranges from 0-1 where 0 is “not membrane” and 1 is “membrane”, and then throw away any pixels that are below 0.5. The number 0.5 is a parameter and was chosen arbitrarily here, although we note because the sigmoid activation function of our neural net “pushes” to the extremes, there is not a large difference between 0.5 and, say, 0.4.

Next, we used k-means to segment the steps. K-means takes an arbitrary amount of vectors and clusters them together into k clusters by iteratively moving the means to minimize the sum of the

euclidean distances between the means and the vectors. Informally, k-means starts with some randomly-placed centerpoints, and then moves them until they are the closest to as many points as possible. In order to take advantage of both the color and spatial coordinates, we eventually settled on concatenating the 3 spatial coordinates (XYZ) of each pixel to the 3 color coordinates (represented as RGB) of each pixel, forming 6-dimensional vectors. K-means uses *euclidean distance*, a generalization of the familiar pythagorean theorem to an arbitrary number of dimensions, as the metric to minimize. As an aside, we note that normalizing data prior to using k-means is crucial, a lesson painfully learned during the course of our work.

In our approach, we used k-means to create an oversegmentation. Setting the parameter k too low will cause unwanted background pixels to be overzealously added to the groups due to their spatial proximity, so in order to obtain a cleaner split along the line between membrane and non-membrane, we sought to increase k. This *oversegmentation* partitions one neuron into multiple clusters.

Finally, we used connected components to merge together multiple clusters into one neuron. First we used connected components in 3D to isolate the individual “islands” that represent neurons; two clusters separated by thrown-away background should obviously not be merged together. Then we compared the euclidean distance of each cluster’s mean color, and merged clusters together if the euclidean distance between the two is less than a parameter, set at 0.2. This ensured two touching neurons of different colors are not mistakenly merged.

Iterations

Our first major problem arose at the k-means stage. After running k-means, the clusters’ colors appeared to have no grounding in the actual colors of the neurons they represented (Figure 2). Additionally, the borders between clusters appeared to take the form of straight lines – unusual for

biological data. To analyze further, we performed a max-intensity projection on the 3D output volume, and obtained an image with the telltale look of a voronoi pattern. (Figure 3)

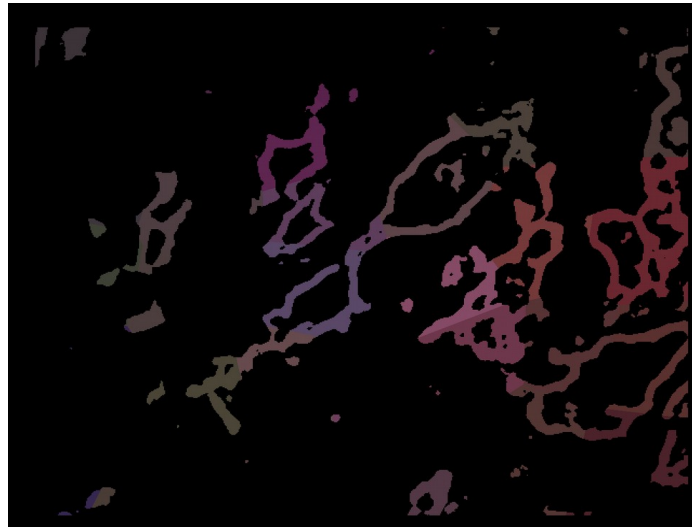


Figure 2: The result of running k-means without normalization. Segmentation output did not take color into account.



Figure 3: A max-intensity projection of the same dataset as Figure 2 reveals a voronoi diagram, showing a segmentation that does not reflect the varying color values of each pixel.

A voronoi pattern is formed by clustering each point according to their closest mean in space, so the fact that one appeared was a clue that the color coordinates of each point were not factoring into the

distance metric used by k-means. Upon further investigation, we discovered that in our 6-dimensional vectors (created by concatenating 3 spatial coordinates, the location of the pixel in the image, and 3 color coordinates, one per channel, in the order Z-Y-X-R-G-B), the color coordinates were limited to a range of 0 to 1 while the spatial coordinates had a range of 0 to ~1000. As a result, the color terms simply weren't big enough to significantly contribute to the euclidean distance metric k-means used.

To fix this problem, in our next iteration we standardized each channel⁷ before feeding the data into k-means. For each channel, we saved the mean and standard deviation, then replaced each observation with its z-score: $x_{\text{new}} = (x - \text{mean}) / \text{std}$. This gives every channel an equal standard deviation of 1 and ensures each channel will affect the result equally. After computing k-means, we unstandardized each channel using the saved values. This iteration successfully segmented individual neurons apart (Figure 4). However, the clusters erroneously classified the dark background pixels that were accepted by the mask as part of the neuron – note the upper red neuron, for example, which consists of one cluster even though the edges of that cluster should be separated into another background cluster). This undesirably violates the Separation Criterion (as defined in the Results section).



Figure 4: Iteration 2. Different clusters are represented by the RGB component of the centroid. Black represents a pixel thrown out by the neural net mask. The rightmost green neuron consists of several slightly different shades of green, indicating the different clusters. Background pixels that have not been thrown out by the mask are erroneously classified as part of the neuron, undesirably violating the Separation Criterion.

In order to allow our algorithm to more easily differentiate background from membrane, we reasoned additional pre-processing could make a measure of background-likeness to show up as a single channel in the Euclidean distance metric, instead of a parameter spread across multiple channels. Since Brainbow expresses the presence of a neuron via the presence of one or more channels, one good candidate for such a channel would be a “brightness”: the sum of all channels, which would be close to 0 if a pixel was part of a background but not otherwise. Since the Brainbow data we obtained, used only 3 channels, we settled on the Hue-Saturation-Value (HSV) color space because the 3rd coordinate, “value”, represents the overall “brightness” of a pixel, and therefore could plausibly differentiate between dark background pixels and lighter membrane pixels.

In 3 color dimensions, transforming a pixel image to a coordinate system with a lightness component (such as HSV) is equivalent to taking the RGB cube and rotating it so the black corner is down and the white corner faces up. In more than 3 color dimensions, an analogous transformation would be any conformal transformation that maps the point with color coordinates [1,1,1,1...] to [1,0,0,0...] and the point with color coordinates [0,0,0,0...] to [0,0,0,0...]. The first component then becomes an indicator of the “lightness” of the point. This criterion does not strictly determine a single transformation with this property in $n > 3$ dimensions, but they are all merely hyper-rotations and stretchings of the same transformation around the lightness axis.

However, transforming to HSV did not do much to help k-means differentiate the background from the membrane. We reasoned that perhaps some coordinates were unneeded and were introducing unwanted variance into the euclidean distance, and therefore throwing away some extra, unneeded components might help. To do this, we set one coordinate of the HSV coordinates of each pixel to 1.0 prior to segmentation.

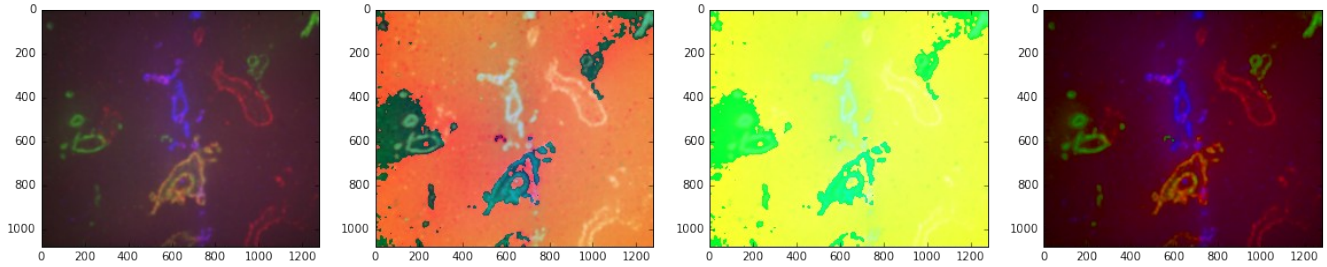


Figure 5: An illustration of Iteration 3 and throwing away coordinates. a) A 2D slice of the raw data. b) The slice converted to HSV, where red=H, green=S, and blue=V. Note the green patches – these arise from the discontinuity in hue addressed in Iteration 4. c) Saturation=1 for all points. d) Image 5c converted back to RGB. To the eye, the neurons appear to pop out from the background more than 5a, but this did not substantially affect the resulting segmentation.

However, this did not have much impact on the resulting segmentation, as measured by the Bijection and Separation Criteria defined in the Results section. We tested dropping both hue and saturation, but neither of these seemed to stop the segmentation from dragging in unwanted background pixels.

At this stage, we also noticed that HSV’s discontinuity in hue was adversely affecting the segmentation. Remember that euclidean distance measures the absolute value of the numerical difference between coordinates. In HSV coordinates, because hue is effectively an angle along the color wheel, a hue of 100% represents the same color as a hue of 0%. However, naïvely using the euclidean distance without correcting for HSV’s modular arithmetic means these two colors would be viewed as 100 units apart instead of the correct 0 units apart. This is an undesirable result.

Here we introduce the notion of “color wheel space”. Intuitively, one can imagine this as the XY-coordinates obtained by locating a color on a color wheel. Explicitly, it takes this form:

$$[\text{value} * \cos(\text{hue} * 2\pi), \text{value} * \sin(\text{hue} * 2\pi), \text{saturation}]$$

With this transformation, the discontinuity in hue is eliminated, and the saturation component can still be removed if desired by disregarding the third component.

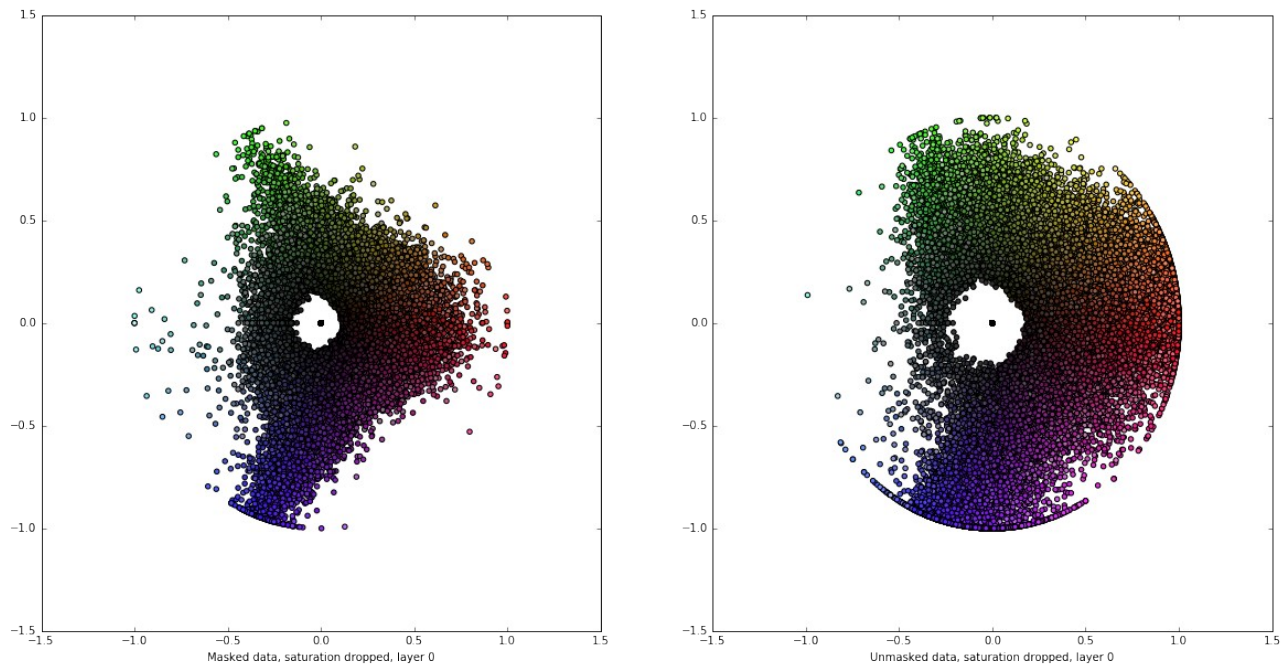


Figure 6: Color wheel space plots of every pixel in one 2D slice, with saturation dropped. a) Color space coordinates for every pixel that is considered background by the mask, 50% threshold. b) Plot of all pixels not considered background by the mask, 50% threshold.

Finally, because the oversegmentation step has spread an individual neuron across multiple clusters, we must merge those individual clusters into clusters that each represent one neuron. Because k-means outputs a mean alongside its classifications, we can extract the average spatial position and average color of all pixels in a cluster, and use this information to determine which clusters belong to the same neuron. Intuitively, neurons that are not touching one another should never be part of the same neuron, so we run connected components on the mask generated in step 1 of our procedure to calculate an adjacency matrix for each cluster: 1 if two clusters are part of the same “island”, and 0 if they are not.

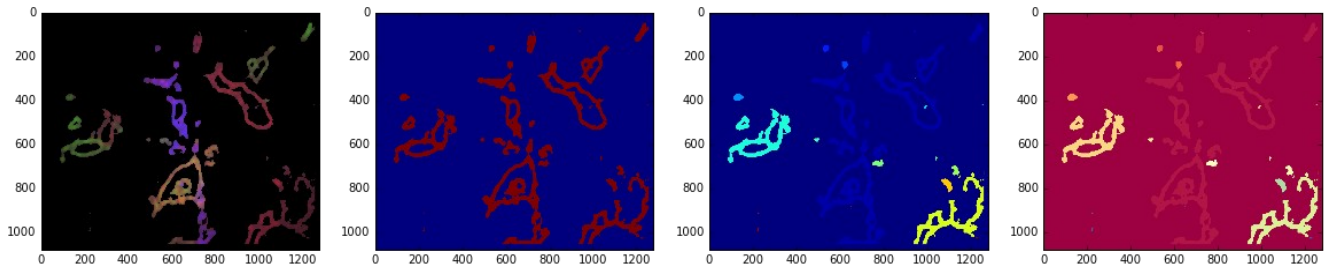


Figure 7: An illustration of connected components. a) The masked data. b) The mask, in black-and-white. c) A 2D slice of the results of using connected components, where two pixels are colored the same color if it is possible to reach one from the other without going through the masked-out region. The majority of the components have the same color even though many of them are separated in this 2D slice because they are connected in 3D. Random colors have been given to each cluster. d) The same plot with a different color scheme.

Then we used this adjacency matrix to merge together clusters based on their average colors, using the adjacency matrix to ensure no two clusters that are in separate “islands” are ever merged. Our first implementation computed the euclidean distance, in RGB, between the colors of each centroid, and repeatedly merged clusters if they were less than a parameter of 0.2.

Results

Before proceeding, we articulate two qualitative criteria by which to judge segmentations. As we are presenting a new pipeline operating on new data, no baseline results exist against which a quantitative metric can be compared to. At this stage, the best method by which to measure results is to manually review segmentations using domain knowledge. First, a segmentation should not erroneously claim two neurons belong to the same cluster (the “Bijection Criterion”), and second, a segmentation should cleanly separate a membrane from the background (the “Separation Criterion”).

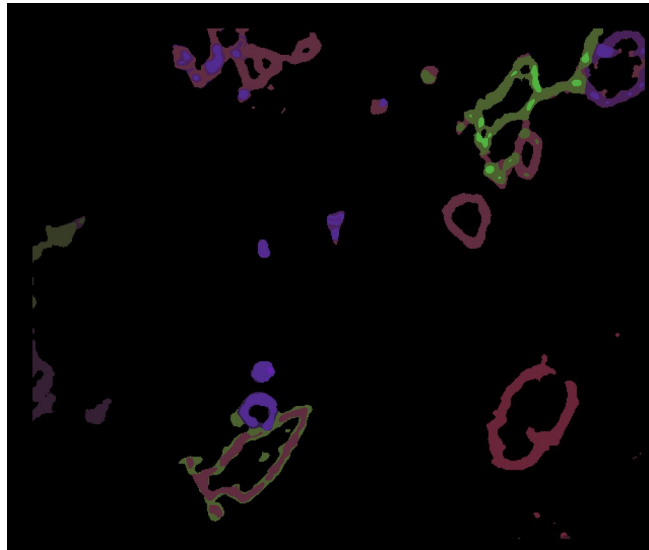


Figure 8a: Layer 22 of our algorithm's output. The top-right and bottom-left of this image feature multiple neurons of different colors in close proximity, correctly separated as per the Bijection Criterion. Most of these clusters do not separate the background into a separate cluster, violating the Separation Criterion. Of those that do, the top right cluster only features disparate blobs instead of the clean split that the human eye registers as correct. Only the bottom red cluster separates the background on either side of its red membrane into another cluster, and even so, the red membrane is not the solid line a human would expect it to be.

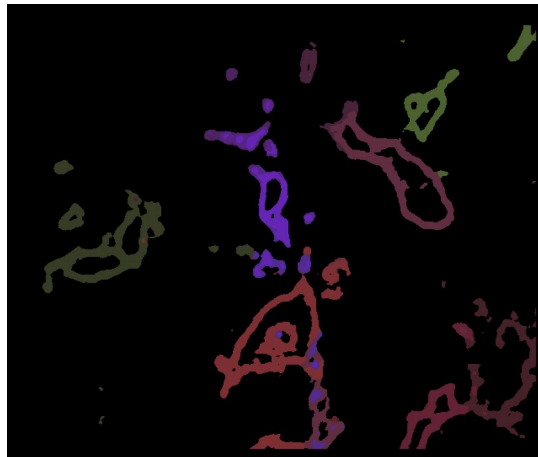


Figure 8b: A 2D slice of our algorithm's output. The leftmost green cluster is in fact two differently-colored neurons merged together, violating the Bijection Criterion. If a violation of the bijection criterion occurs, the segmentation becomes inaccurate, introducing errors into the eventual connectome.

By the Bijection Criterion, our algorithm works well in most cases. Many neurons have been correctly identified. Figure 9 shows the darker green cluster visible in the top right of Figure 8a in 3D. We reiterate that this green neuron has been extracted from a 20x ExM dataset using a start-to-finish automated pipeline. Note that this neuron fails the Separation Criterion – as visible in figure 8a, the

cluster has expanded to fill all of the non-black pixels, those which have marked by the mask as non-membrane candidates – but this is of a lesser concern because it does not greatly affect the extracted morphology.

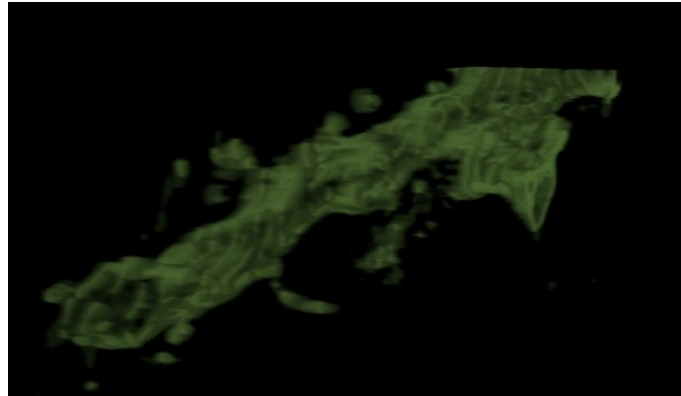


Figure 9: One post-merged cluster, correctly comprised of one neuron. The horizontal line at the top of the neuron is the top of the original dataset. Image is a still of a 3D turnaround created by extracting one cluster and rendering it in 3D using FIJI. Note the dendritic spines, normally too small to see with light microscopes, are visible thanks to 20x ExM.

However, many clusters fail the bijection criterion. Figure 10 presents an example of improper merging: several red-colored neurons have been merged together despite clearly being 4 different neurons to the naked eye. The middle-bottom of the image features portions of a blue neuron that has been improperly marked as part of the same red neuron. This has arisen despite the presence of an upper bound on the distance in space centroids must be from one another in order to merge. Since our merge step is an iterative process that repeats until merging is completed, we hypothesize that small islands in between the neurons may have been improperly merged, forming a daisy-chain that could have “bridged the gap”.

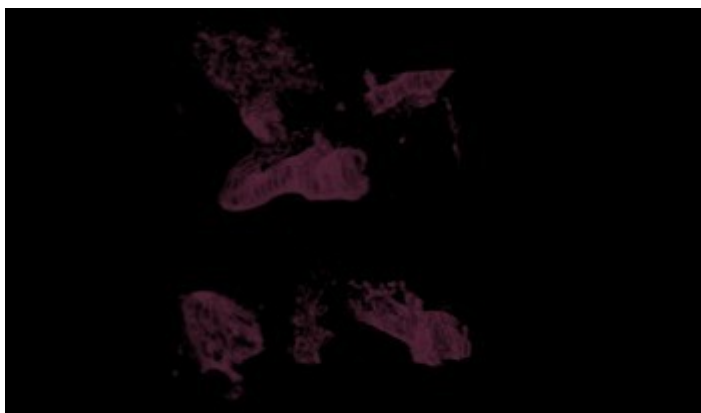


Figure 10: One cluster that our algorithm created by mistakenly merging several separate neurons that were similarly-colored.

Future Work

There are a number of ways to potentially improve both future segmentation algorithms and the data being segmented, ranging from adding additional stains to leveraging existing biological knowledge.

The input data can be improved to reduce some of the difficulties we have faced in this paper. One major challenge for segmentation of 20x ExM images is the punctate nature of the data. If membranes were unbroken, segmentation would be far easier and algorithms like the watershed algorithm could be used. This could be arranged by amplifying the existing fluorescent molecules after a round of expansion, or by adding more Brainbow virus every iteration of expansion. The key challenge is to maintain the same combination of colors across multiple injections of Brainbow virus, however, which may prove difficult. Perhaps a single injection of Brainbow virus post-expansion could result in a more uniform cell membrane.

We note that iExM is a destructive form of imaging: it cannot be done in situ, so one cannot obtain a connectome *and* measure electrical potentials, which may prove problematic because neural circuits may vary from organism to organism. A technique to obtain Brainbow ExM images without

killing the brain would allow electrical monitoring of neuron spikes in addition to connectome analysis. Among other things, this would allow experimental validation of a connectome and allow researchers to more easily discern the functions of neurons. While far-fetched, such a technique would prove useful for later analysis once a connectome has been developed.

The strategies we have used so far have pulled from general computer vision staples. However, we are not merely segmenting general structures – we are specifically segmenting neurons. Because the field of neuroscience already contains a wealth of information about the structure of neurons, future algorithms should be able to take advantage of this domain-specific knowledge. We will denote this domain-specific knowledge by the term “biological prior”^{*}.

One way to take advantage of biological priors is to consider the topology of neurons. For example, since neurons do not form synapses with themselves, if a proposed connectome involves a “loop” of connections, a future algorithm could detect that as evidence of a merge error. The known thicknesses of an axon or dendritic spine are other examples of useful topological information not currently considered by algorithms. Persistent homology is another interesting avenue that uses morphological operations to discern the structure of an object, where it may be possible to incorporate the biological prior that neurons are unbroken manifolds.

It is worth noting that existing biological priors may be disproven as imaging techniques advance. Any attempt to use biological priors in imaging techniques may end up marking real biological structures that do not match the priors as errors, perpetuating the flawed priors into the observed data. Future algorithms should allow their priors to be wrong by using techniques such as gaussian probability distributions instead of hard-and-fast rules.

^{*} *The term “biological prior” has two advantages: firstly, it references the analogous bayesian prior, and secondly, when described to a lay audience, it can be shortened to the snappy “bior”.*

Additional channels in the data would also aid immensely in segmentation. A cytozolic stain in addition to Brainbow for example, would highlight the inside of every neuron, providing even more data that an algorithm can take advantage of. The biological prior that cytozolic fluid lies within the membrane means such a stain could be used to inform an algorithm where neurons lie (either as a metric to judge segmentations post-merge or a replacement for the neural net mask after a bit of dilation). The absence of cytozolic stain might also be able to detect membrane merge errors: if two centroids were mistakenly connected that belonged to two different neurons, cytozolic fluid would not connect the two insides of the cells. Similarly, any stain that stained the fluids outside of a neuron would help to eliminate background pixels more easily, and synapse stains could provide targets for dendritic spines. Additional channels and varying types of stains would be useful to algorithm-designers both for error-correction and inside the algorithms themselves.

Finally, to store and keep track of multiple channels obtained through multiple techniques, additional metadata tags may prove useful to keep track of the origin of various channels. If a file consists of 3 Brainbow membrane stains and one cytozolic stain, an algorithm should be able to look at the data file's metadata tags to see a table of what each channel represents, and perhaps even what molecule was binding. Such metadata may prove useful for large-scale automated segmentation tools.

As ExM grows in popularity, we anticipate further developments in imaging techniques with excitement.

Conclusions

Our goal was to take advantage of the color information available through ExM and develop a completely automated method to segment 20x ExM data, which has not been previously accomplished. We developed an algorithm that took an input image and segmented it from start-to-finish, which

segmented many neurons correctly. However, it was not able to easily separate membrane from background, and occasionally merged neurons improperly.

Based on our experiences, we also present two useful criteria for expert manual review of segmentations: First, a segmentation should not erroneously claim two neurons belong to the same cluster (the “Bijection Criterion”), and second, a segmentation should cleanly separate a membrane from the background (the “Separation Criterion”). We also present the idea of a “color wheel space”, a variant of HSV more suited for algorithms.

We are not sure if this mask-oversegment-connect pipeline has promise. Nevertheless, we believe that the problems we have encountered in our quest will remain problems for other researchers in the field, and the ideas we have proposed to surmount them have value. Already, we have demonstrated the ability to segment neurons within a human lifetime without a human, and our process can be easily scaled up. With further decades of iteration towards minimally supervised learning algorithms, we look forward to a time when we can easily map every neuron present in the entire brain.

Acknowledgements

We would like to thank Daniel Goodwin for mentorship throughout this project, Ed Boyden for helpful feedback and guidance, Grace Huynh for first suggesting the use of biological priors, and the MIT PRIMES program for this research opportunity.

References

1. Chang, Chen et al. Iterative Expansion Microscopy. *Nature Methods* 14, 593-599 (2017).
2. Livet et al. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature* 450, 56-62 (2007)
3. Chen, F., Tillberg, P.W., E.S. Expansion microscopy. *Science* 347, 543–548 (2015).
4. Davenport, Kevin. The Cost Function of k-means. <http://kldavenport.com/the-cost-function-of-k-means/> (2014). Accessed 9 September 2017.
5. Zung, Tartavull, Seung. An Error Detection and Correction Framework for Connectomics. (2017) [arXiv:1708.02599](https://arxiv.org/abs/1708.02599) [cs.CV]
6. Marblestone, Zamft, et al. Physical Principles for Scalable Neural Recording (2013). [arXiv:1306.5709](https://arxiv.org/abs/1306.5709) [q-bio.NC]
7. Fowler, Cohen, Jarvis. *Practical Statistics for Field Biology*, 2nd ed (1998). Wiley. ISBN 978-0-471-98296-8.
8. Lee et al. ‘*stack1-image.tif*’ from “NIPS 2015 Training Data”. <http://seunglab.org/data/>
9. K. J. Hayworth, J. L. Morgan, R. Schalek, D. R. Berger, D. G. C. Hildebrand, and J. W. Lichtman. Imaging ATUM ultrathin section libraries with WaferMapper: a multi-scale approach to EM reconstruction of neural circuits. *Frontiers in Neural Circuits*, (2014).
10. W. Kuhlbrandt. The Resolution Revolution. *Science* 343, 1443 (2014).
11. Tapia, J.C., Kasthuri, N., Hayworth, K.J., Schalek, R., Lichtman, J.W., Smith, S.J., and Buchanan, J. (2012). High-contrast en bloc staining of neuronal tissue for field emission scanning electron microscopy. *Nature Protocols* 7, 193–206.