

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

**Skeleton Prediction of 3D Neuronal
Structures**

Alok Verma

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Biomedical Computing

**Skeleton Prediction of 3D Neuronal
Structures**

**Skelettprognose von 3D Neuronalen
Strukturen**

Author: Alok Verma
Supervisor: Prof. Dr. Laura Leal-Taixé
Advisor: Prof. Dr. Hanspeter Pfister and Dr. Donglai Wei
Submission Date: Submission date

I confirm that this master's thesis in biomedical computing is my own work and I have documented all sources and material used.

Munich, Submission date

Alok Verma

Acknowledgments

Abstract

To understand the internal workings of brain neuroscientists have been trying to reconstruct its ‘neural circuitry’, also termed as a ‘connectome’ which illustrates how neurons are connected with each other.

A connectome can be constructed by segmenting neurons in a 3D electron microscopy(EM) image and then by finding their inter-connections. Past decade has seen increasing use of Deep Learning based methods to improve segmentation of such EM images and it has been quite indispensable in improving the state of the art in EM segmentation.

But the neuron tracing problem is a hard problem to solve in itself. This can be attributed to its large volume size, multiple image artifacts, numerous closely intertwined segments of vivid shapes etc. [show images illustrating problems](#). Hence, the results of most algorithms are either not satisfactory or they are too slow and entail complicated postprocessing steps, limiting their scope to only inside a computer science lab, and far away from being used as a general tool directly by neuroscientists for new images.

This work tries to develop a new idea of building connectomes, bypassing the segmentation step. It builds upon from existing methods and approaches in EM segmentation but also leverages the tools and tricks of Machine Learning methods from natural image domain.

The first part discusses the relevant works from the Connectomics area, their advantages and pitfalls. Apart from that, it also delves into skeletonization methods used for natural images, which inspires the proposed method.

Contents

Acknowledgments	iii
Abstract	iv
1 Introduction	1
1.1 Connectomics	1
1.2 Connectomics Pipeline	1
1.3 Quirks of Connectomics Data	2
1.4 Segmentation Methods of EM	4
1.4.1 Boundary based Methods	4
1.4.2 Discriminative Learning?	5
1.4.3 Error detection and Error Correction	5
1.4.4 Flood Filling Network	5
1.5 Skeletons in Connectomics	6
1.5.1 State-of-the-art Skeletonization methods in Natural images . . .	6
2 Method	7
2.1 Encoding and Decoding Skeletons	7
List of Figures	9
List of Tables	10
Bibliography	11

1 Introduction

1.1 Connectomics

To understand the internal workings of brain neuroscientists have been trying to reconstruct its 'neural circuitry' also termed as a connectome, which illustrates how neurons are connected with each other. The first ever brain to be studied at this intricate detail was that of C. Elegans in the 1980's [whiteCElegans] with serial section electron microscopy(EM) images and the neurons were traced manually.

Since then, EM methods have improved many folds, generating petabytes of data, even for a small Fruit Fly braincite [Drosophila paper](#). So, the process of segmenting and tracing neurons have been delegated to Machine Learning(ML) algorithms, with humans pitching in only for error correction or initial training data generation.

Neuron tracing problem is a hard nut to crack. This can be attributed to its large volume size, multiple image artifacts, numerous closely intertwined segments of vivid shapes etc. [show images illustrating problems](#). Hence, the results of most algorithms are either not satisfactory or they are too slow and entail complicated postprocessing steps, limiting their scope to only inside a computer science lab far away from being used as a tool directly by neuroscientists.

This work tries to build upon from existing method and approaches which researchers have been working on from deca

1.2 Connectomics Pipeline

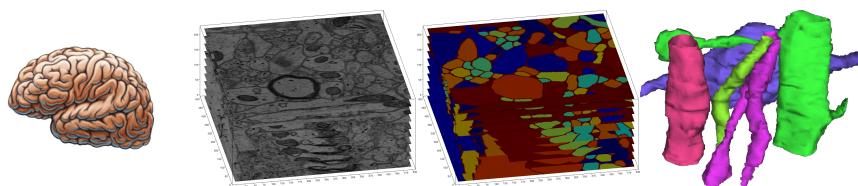


Figure 1.1: Connectomics Pipeline.

The construction of the brain connectomes starts by slicing the brain tissue as thin as possible and imaging the 2D slices using an electron microscope, typically at resolution

of 3 – 10 nanometers. After imaging, the 2D images are stacked, aligned and processed to remove noise and other artifacts.cite alignment papers, ask Adi. Finally, this 3D image is segmented, often by doing multiple, laborious iterations of automatic segmentation and manual error correction to obtain a connectivity graph of the neuronal processes. Figure 1.1 shows a simplified succinct connectomics pipeline, starting from the brain tissue, alignment and segmentation. After segmentation multiple other analysis can be performed, which would entail more processing.

1.3 Quirks of Connectomics Data

A EM image stack can contain numerous of segments which are arbitrarily and closely intertwined with each other. The membranes demarcating different cells can be thin and tough to distinguish even by trained humans. Figure 1.2 shows a EM dataset cite SNEMI illustrating thin boundaries, closely packed nature of those neuronal segments and intertwined structure of the segments.

Apart from previously mentioned properties, EM datasets especially from serial section EM usually have some more quirks like:

- The resolution of the 3D image may not be isotropic. For serial section EM, the in-plane resolution along X and Y axis is usually 2 to 5 times higher than the Z resolution which is governed by the slice thickness
- 2D image slices can have large artifacts due to physical folds or knife marks while imaging.
- There can be sharp discontinuities across slices due to misalignment, missing slices etc.
- The boundaries of segments can break at some points or get blurred and hard to distinguish.
- The size in pixels of one volume can be in the order of $10K * 10K * 5K$ voxels and a single segment can encompass from one diagonal end to the other.

Thus, EM datasets are much more complex to segment compared to other natural non medical imaging datasets, and thus specialized algorithms need to be developed to tackle them.

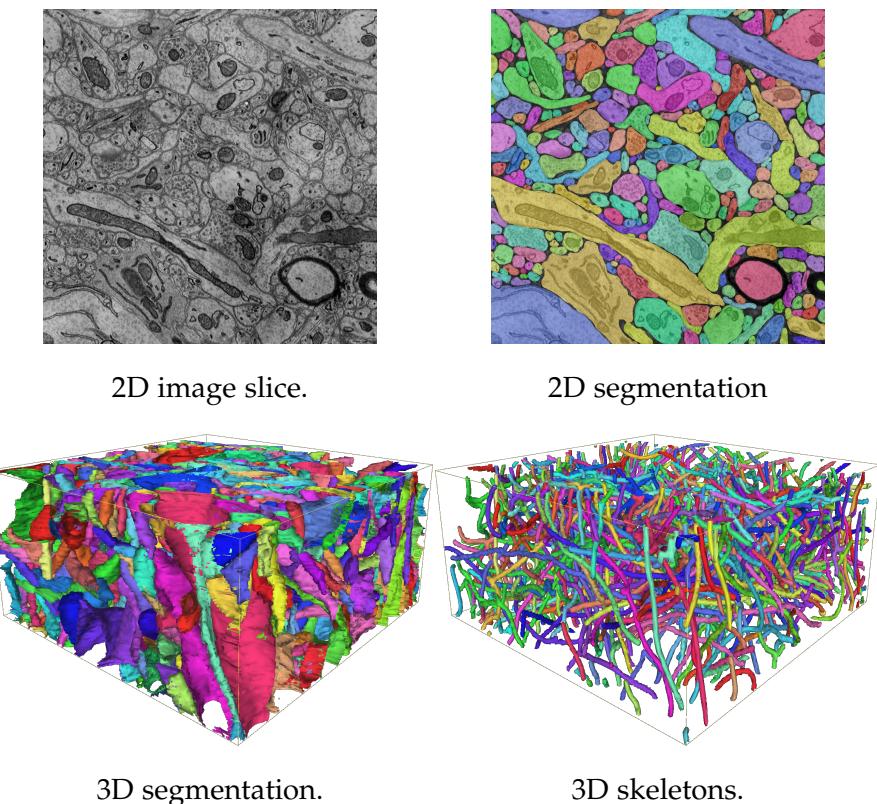


Figure 1.2: The SNEMI Dataset. It was created as part of a Connectomics challenge to advance segmentation methods [Cite SNEMI](#). The dimensions are relatively small - $100 * 1024 * 1024$ voxels with resolution of $30 * 6 * 6$ nm.

1.4 Segmentation Methods of EM

1.4.1 Boundary based Methods

To segment multiple 3D segments as shown in Figure 1.2 one of the usual methods is to first perform semantic segmentation into two classes - foreground, which consists of all the internal pixels of segments, and background, which consists of the extracellular space. Given such a perfect semantic mask, it is trivial to obtain the instance segmentation of each cell using simple connected component analysis. One can then match segments from different slices using intersection of union scores. [cite boundary matching paper ask donglai how boundary mask is segmented?](#)

Another commonly used approach for EM 3D segmentation is to represent boundaries as affinity graphs, first introduced by [Tur+10]. In its simplest form affinity graphs are similar to boundary masks - if a voxel and its immediate next neighbor lie in same segment then a high value, usually 1, is stored at that voxel. In this scenario the affinity graph would be of 3 channels, one each for the immediate next neighbour along X, Y and Z. Affinity graphs can be created for distant neighbors also, leading to more channels, and encoding long distance connectivity, useful for thin elongated segments, used by [Lee+17]. After obtaining such soft partition graph, the final segmentation can be done using standard graph partitioning methods or even connected components analysis, as shown in [Tur+10] or 3D affinity based watershed transform, used by [Lee+17] and [ZS15].

A common underlying idea in both methods is that they break down the segmentation problem into two steps. First is a local prediction step, where boundary is predicted based on local image textures and edges. Second is somewhat more global, which looks at connectivity of voxels and performs hierarchical clustering of oversegmented clusters based on shape and connectivity. The dismantling of the segmentation problem helps to tackle multi instance segmentation problem and makes it amenable for Deep Learning (DL) based methods.

A major issue with these pipeline processes is that a minute leak in the boundary segmentation can lead to false merge. To somewhat avoid such false merges, the hyper-parameters of watershed clustering are tuned so as to keep the results optimally over-segmented. This is done keeping in mind that its easier to correct false splits for humans rather than fixing false merges. Nevertheless, manually agglomerating over-segmentation is still laborious and not extendable to larger volumes. Second issue, encountered in DL based boundary prediction methods with a $L1$ or $L2$ loss is that it cannot force the network to learn the shape of segments, a cue very frequently used by human annotators to distinguish ambiguous boundaries. This is due to the fact that we are only forcing the network to learn local boundary predictions, which a

model can learn to predict based on textures only which is not enough. Apart from this, models are trained to reduce the mean loss over the entire dataset, this means it may learn to generate boundaries for most parts of the segment correctly, but may miss some boundaries. Even those small errors can snowball into large false merges in post-processing steps as explained earlier. [Show images for segmentation error cases?](#). Recently MALIS loss was proposed to avoid this. In practice its a weighted loss which penalizes errors causing topological errors heavily compared to others.

1.4.2 Discriminative Learning?

1.4.3 Error detection and Error Correction

It is unrealistic to obtain perfect boundary predictions directly from a single Deep Net. For perfect results an enormous corpus of training data with a verastile loss function is needed which can force the network to learn to look for all cues a human annotators employs like looking for texture, boundary, continuity across slices, and shape etc, which is unrealistic. So, recent works have explored the idea of error detection and correction of pre-segmented volumes [ZTS17], [Mat+19]. [ZTS17] trains a model to first identify split and merge errors and then trains another model for creating a correct segmentation at those locations. While, the method in [Mat+19] only solves false splits by computing skeletons and searching for matching skeletons near its ends. It computes a weighted graph with nodes comprising of segments and the edges weighted by the probability of segments being combined together. The merges are finally computed by partitioning the graph.

1.4.4 Flood Filling Network

All the previous discussed methods[cite segmentation papers](#) were tuned and tested on relatively small scale data, but advancements in EM helped create larger and larger datasets reaching hundreds of Gigabytes, and the previous state-of-the-art couldn't catch up to produce acceptable results. But a recent method called Flood Filling Network (FFN) [Jan+18] from Google was able to produce acceptable results for the first time on volumes of approx size $10K * 10K * 6K$ voxels. Their method segments one object at a time, growing its mask sequentially, until it traces the entire object. Initially a small seed is marked inside a segment, and a mask is predicted using a Deep Net for that single segment in a small field of view around that seed, further seed points are found inside the predicted mask by jumping a fixed step size in all possible directions. But similar to previous approaches neither can this method predict a mask completely accurately in one-shot. So, multiple iterations are run at different scales and seeding order to create a over-segmentation and then an extensive agglomeration step

is performed using the same model. It is interesting to note that while this method is the state-of-the-art in connectomics, but apparently no other lab than Google has been able to use it for any large dataset. One reason being, with just 1000 P100 GPUs it needs 16 hours to run inference on the large volume[Jan+17]. So, currently its hard to put forth FFN as a general tool for Neuroscientists and a urgent need remains to create a segmentation tool which is feasible and easy to use.

1.5 Skeletons in Connectomics

While EM segmentation is a major step in the connectomics challenge but it is not the final goal. Neuroscientists are interested to find out interconnections between neurons through synapses, and their geodesic distance to the soma. Such analysis naturally calls in for skeletonization of neurons, and a straight forward way is to skeletonize the predicted segmentations, for which many off the shelf algorithms already exist **Cite traditional skeletonization methods**. Recently, curated skeletonization methods for EM data have also been proposed. Apart from serving as a connectome analysis tool, skeletons are instrumental in improving the segmentations itself. Since, they capture a global topology of segments - branch points and end points - they can be a indicator of false split and false merges in segmentation **Show picture of FM and FS in skeletons?**. A recent method **cite brains skel paper** utilized skeleton end points to identify false splits and merged them using skeleton curvature information. Creating ground truth skeleton data is also easier than segmentations. Dedicated tools **cite KNOSSOS** exist for human annotators for neuron center line labelling. One paper [**Berning2015**] quotes manual skeletonization to be 25 to 100 times faster than segmentation labelling. It also proposes a semi-automatic segmentation method based on hand traced neurons. Also, the state-of-the-art FFN method, [Jan+17], [Jan+18] utilizes skeletons for validation and error metric (estimated run length) calculation. The metric they choose is tailored to report high error if the reconstructed segmentations are falsely merged or split, but it does not penalize small segmentation perturbations on the boundary, so the focus is more on improving the topology - or the centerline skeletons - rather than boundaries. In another kind of neural images obtained from Fluorescence confocal microscopy, Neuron tracing [**Kayasandik2018**] methods are being developed, which essentially is also skeletonization. So, skeletons are a useful representation for neurons and would be an important tool for connectomics in near future.

1.5.1 State-of-the-art Skeletonization methods in 2D Natural images

Since many natural image processing tools have been applied to EM images, it is pertinent to review existing 2D natural image to skeleton methods which can guide

development of 3D EM skeletonization method.

It would come as no surprise that all recent state of the art methods [**Shen2016**], [**Shen2017**], [**Ke2017**], [**Wan+19**], [**Xu2019**] are based on deep learning. Except [**Wan+19**], all cited methods formulate skeltonization as a pixelwise binary classification problem. [**Shen2016**], [**Shen2017**], [**Ke2017**] utilize multiple side outputs from different depths of the network to capture skeletons at all object scales. They also use cross entropy loss, which does not focus on any skeleton shape regularization, hence, leading to blurry and disconnected predictions. [**Wan+19**] on the other hand proposes to learn a vector field called flux, which points to the nearest skeleton pixel. A decoding step is required to extract skeletons from the field. [**Xu2019**] argues against such field prediction and proposes geometry aware loss function which penalizes blurry and disconnected outputs.

2 Method

This section proposes a novel method for instance skeletonization of neuronal structures in EM images. The method presented here is motivated by existing [Wan+19] learning based skeletonization method for natural images, but differs in two ways - First, it creates separate skeletons for each individual segment- we describe it as instance skeletonization. Second, its is applied on 3D EM images, which to the best of our knowledge has not been explored yet.

2.1 Encoding and Decoding Skeletons

Previous skeletonization methods [**Shen2016**], [**Shen2017**], [**Ke2017**], [**Wan+19**], [**Xu2019**] do not separate skeletons of multiple objects. Their prediction is a binary mask marking skeleton points of all salient objects as 'True'. But, in EM images, it is important to separate skeletons of one object instance from another. It is not straight forward to construct a Deep Learning model which can directly demarcate skeletons for every object instance. Hence, a system of encoding must be devised which allows to formulate a feasible loss metric for training and also to separate skeleton instances in a postprocessing step.

[Wan+19] proposes 'DeepFlux' which encodes skeletons of objects in 2D images using a \mathbb{R}^2 vector field. It defines a 2D context region around skeleton, and all \mathbb{R}^2 vectors in that context region point to the nearest skeleton pixel. This provides a easy non-mask based encoding to represent skeletons, and the objective of a Deep Net model would be to predict this field.

The proposed method is also inspired from this encoding, albeit for 3D skeletons and some other tweaks.

First, instead for creating a \mathbb{R}^2 field a \mathbb{R}^3 field for 3D skeletons is created. Ground Truth skeletons can either be manually generated by human annotators [cite KNOSSOS Tool](#), or by using segmentation to skeletonization algorithms. The context region for a skeleton is also defined in 3D using a sphere of Nnm radius, and finally for all points in the context region a \mathbb{R}^3 field is defined such that the vectors point to the nearest skeleton voxel. Such defined direction field is non-smooth if the skeletons are defined on a discrete grid. This is due to the fact that direction vectors will jump abruptly from one skeleton point to another. Learning a non-smooth field is not encouraged as

Deep Nets are usually fail to generate sharp fields. Hence, to create smoother field, first skeleton points are interpolated using splines and then a distance transform is calculated. Finally, the gradient of the distance transform defines a \mathbb{R}^3 encoding of the skeleton pixels. **create graphic of the flux and the context and directions**

To obtain the instance skeletons back from such an encoding, simple postprocessing steps can be used. First observation about the field is that, in a small neighborhood around the skeleton they point away from each other, while at other locations they are pointing almost in the same direction. This property can be used to identify skeleton voxels. The Divergence operator is apt for such field analysis. Divergence at diverging field location, which are skeleton points would be high, where as for all other location it would be low. Thresholding the divergence would allow to find the skeleton points. Finally for instance skeletonization, connected components analysis can be performed to separate skeleton instances.

The advantages for encoding and decoding the skeletons in such a manner are:

- Deep Net model has to learn to look for both global and local properties while predicting the field. This helps to avoid local false merges due to small membrane breaks, as seen in boundary based methods.
- Voxel wise loss function for the deep net can be easily constructed. The loss metric is agnostic of skeleton instances.
- Decoding methods - Divergence and Connected Components analysis - are intuitive and fast, unlike watershed for the case of boundary based methods.
- Predicted field can be useful to solve false merges and splits in later post processing steps.

2.2 Network Architecture

2.3 Training Objective

List of Figures

1.1	Connectomics Pipeline.	1
1.2	The SNEMI Dataset. It was created as part of a Connectomics challenge to advance segmentation methods Cite SNEMI . The dimensions are relatively small - $100 * 1024 * 1024$ voxels with resolution of $30 * 6 * 6$ nm.	3

List of Tables

Bibliography

- [Jan+17] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka, W. Denk, and V. Jain. “High-Precision Automated Reconstruction of Neurons with Flood-filling Networks.” In: (Oct. 2017). doi: 10.1101/200675.
- [Jan+18] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka, W. Denk, and V. Jain. “High-precision automated reconstruction of neurons with flood-filling networks.” In: *Nature Methods* (2018).
- [Lee+17] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung. *Superhuman Accuracy on the SNEMI3D Connectomics Challenge*. 2017. arXiv: 1706.00120 [cs.CV].
- [Mat+19] B. Matejek, D. Haehn, H. Zhu, D. Wei, T. Parag, and H. Pfister. “Biologically-Constrained Graphs for Global Connectomics Reconstruction.” In: *CVPR*. 2019.
- [Tur+10] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung. “Convolutional Networks Can Learn to Generate Affinity Graphs for Image Segmentation.” In: *Neural Computation* 22.2 (Feb. 2010), pp. 511–538. doi: 10.1162/neco.2009.10-08-881.
- [Wan+19] Y. Wang, Y. Xu, S. Tsogkas, X. Bai, S. Dickinson, and K. Siddiqi. “DeepFlux for Skeletons in the Wild.” In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [ZS15] A. Zlateski and H. S. Seung. *Image Segmentation by Size-Dependent Single Linkage Clustering of a Watershed Basin Graph*. 2015. arXiv: 1505.00249 [cs.CV].
- [ZTS17] J. Zung, I. Tartavull, and H. S. Seung. “An Error Detection and Correction Framework for Connectomics.” In: *CoRR* abs/1708.02599 (2017).