
# Automatic 3D Neuron Tracing from Optical Microscopy Images

by

Zhangcan Ding

*National Centre for Computer Animation*

Faculty of Media & Communication

Bournemouth University

A thesis submitted in partial fulfilment of the requirements of Bournemouth University for the degree of *Doctor of Philosophy*

*June. 2021*

# Copyright Statement

# Acknowledgements

First of all, I would like to thank my supervisors, Dr.Hongchuan Yu, Professor Jianjun Zhang and Prof Hanchuan Peng for their help and support during my PhD study. Particularly, I would like to give very special thanks to my supervisor Dr.Hongchuan Yu for giving me the chance to start this journey.

I would like to thank the senior researchers in SEU: Zhi Zhou, Yimin Wang, Liya Ding, Peng Xie, Zongcai Ruan. Thank you for your patience and guidance.

I also would like to thank Dr.Yinyu Nie, Dr.Jinglu Zhang, Dr.Yunfei Fu, Yao Lyu who has been working closely with me for this dissertation. Thank you for your insightful advice and kind help.

I woud like to thank my friends: Ruibin Wang, Yu Zhou, Tongxin Chen, Nan Xiang, Mengqing Huang, Tingting Li, Xilin Dai, Wei Xiong. Thank you for bringing on joy and laughter.

I would like to thank my colleagues and friends in SEU: Qiang Ouyang, Shengdian Jiang, Sujun Zhao, Siqi Cheng, Peng Wang, Xuan Zhao, and Linus Manuben-Gil. You made my time in SEU a lifelong memory.

I would like to thank our postgraduate research administrators: Sonia Ashby and Sunny Choi. Thank you for always being patient, helpful and considerate.

I would like to thank my parents. Thank you for always standing behind me, trusting me and encouraging me.

At last I would like to thank my love of my life, Binglei Wu. I love you with all my heart.

# Abstract

Neuron tracing is the process of reconstructing three-dimensional morphology of neurons from microscopy images. It is essential for delivering more comprehensive understanding of the relationship between neuronal structure and function, which is the fundamental to know how the brain works. However, currently neuron tracing remains a challenging task, due to the natural complexity of neuronal structure, inadequate available data and computational limitation. In recent years, many automatic neuron tracing methods have been developed in the research field, with limited success on specific issues. The lack of a robust neuron tracing method with more general applicability greatly restrains systematic characterisation and analysis on neuronal morphology.

To address aforementioned challenges, we first establish a pipeline to generate more standard data, in which we specifically propose a novel approach for automatic refinement on semi-manual reconstruction. Following the pipeline, we manage to generate more than 1000 full morphology data. Second, based on the generated standard reconstruction, we conduct a systematic and quantitative analysis to identify the most critical obstacles in neuron tracing. Third, we propose a novel neuron tracing method by embedding occupancy learning with curve skeleton extraction, which tackles the major issues of weak and punctuated signal, as concluded from the previous analysis. We curated a large dataset to train and test the model. The experimental results show it exceeds other counterpart approaches in most terms of evaluation metrics. At last, we propose a novel learning model for automatic

neuron tracing, which learns to directly extracts the skeleton from a raw image. It addresses the main issue of close but irrelevant signal, as concluded previously. We train and bench test it on the curated dataset, as well as a public dataset. Experiments show it achieves state-of-the-art performances in all cases.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Motivation

More than 100 years ago, Ramón y Cajal first observed the polarised shape of a neuron, and demonstrated the dendrites and axon as input and output sides of signal processing respectively (y Cajal 1911). Fig 1.1 illustrates a typical structure of a single neuron. The discovery was based on neuron tracing i.e. reconstruction: to follow all the dendritic branches and long axon projection till every end of axon terminals to formulate a single tree structure that exactly represents the morphology of a neuron (Schwartz 1993). Through the 3D neuronal morphology, Cajal predicted the neuronal connections and elucidate the circuit logic of several brain components e.g. hippocampus (Ramón y Cajal 1928). Ever since then, 3D morphology of a neuron has been recognized as a defining feature of neuronal types and its functional role in the neural circuit (Peng et al. 2015). Characterising the morphology of neurons is so essential as it delivers more comprehensive understanding of the neuronal diversity across different nervous system, and contributes to the major goal of modern neuroscience, which is to clarify how brain functions (Donohue and Ascoli 2011).

Despite the early success, the knowledge of the diversity of neuronal morphology is still limited. In modern digital era, to systematically characterise the morphology of neurons, a typical workflow consists of five major steps: labelling, imaging, tracing, registration and classification (Peng et al. 2020a).

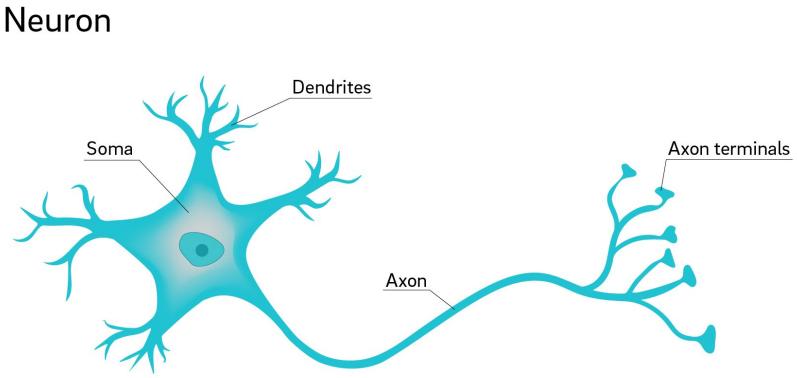


Figure 1.1: A simple illustration of a neuronal structure (Puppo et al. 2018).

Fig 1.2 illustrates the pipeline of the process. Neurons must first be labelled with a dye or transgenic tracer to reveal the structure. Next, optical microscopy methods are utilised to capture the image. Then, a computational tracing tool extracts the geometry of the neuron from the image data. The traced or reconstructed results are then registered to a common template for final classification and analysis. In recent years, the upstream process of neuron tracing i.e. labelling and imaging has made substantial advancement: novel transgenic or viral Cre delivery techniques e.g. TIGRE2.0 (Daigle et al. 2018, Veldman et al. 2020) have enabled efficient, sparse yet consistent labelling of a wide range of neuronal types across multiple brain area; high-resolution imaging and high throughput imaging techniques e.g.fluorescence micro-optical sectioning tomography (fMOST) generate massive amount of whole- brain fluorescent datasets at micron level (Gong et al. 2016, Economo et al. 2016). However, the neuron tracing or reconstruction remains a key bottleneck in the whole pipeline, significantly limiting downstream analysis of neuronal morphology. Currently, neuron tracing mainly relies on human labour for annotation, which is tedious, time-consuming and prone to error (Acciai et al. 2016). It is therefore highly desirable for the development of automatic neuron tracing methods to standardise and accelerate the pro-

cess.

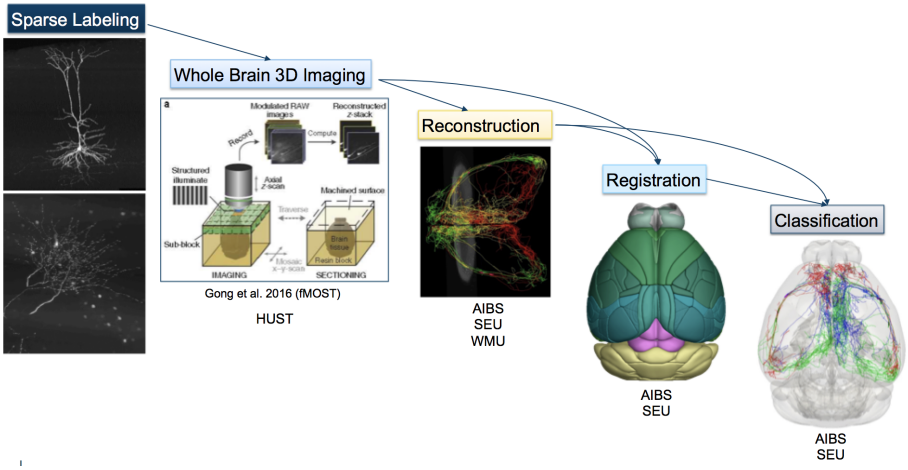


Figure 1.2: Pipeline of characterising neuronal morphology.

In recent years,to motivate the advancement of robust neuron tracing methods, DIADEM (Digital Reconstruction of Axonal and Dendritic Morphology challenge) (Ascoli 2008) and BigNeuron project (Peng et al. 2015) has been launched to benchmark several automated algorithms. A few efficient automatic tracing methods are developed such as APP2 (Xiao and Peng 2013), Neutube (Feng et al. 2015) and SmartTracing (Chen et al. 2015). In respect of simple and relatively small images, those algorithms are able to generate meaningful reconstruction. Fig 1.3 shows an accurate reconstruction from an optical image. Nevertheless, their performance on relatively complicated and large images are unconvincing (Peng et al. 2017).

In the last few years, deep learning algorithms have achieved a great success in various computer vision tasks and gains its popularity in neuron tracing. Li (Li et al. 2017) combined a 3D convolutional neural network (CNN) for image segmentation with several tracing methods and improved the performance. Zhao (Zhao et al. 2019 2020) proposed a framework to train a 3D CNN from pseudo labels produced by any traditional tracing methods, and mutually complement them to progressively improve reconstruction. Whereas, deep learning methods involved in neuron tracing are still under-explored because of the limited amount of public data.

Particularly, each automatic tracing method is designed for some specific issues in neuronal image e.g. weak signal, punctuated branches etc, and yields in more general circumstance. The occurrence of those issues and the cause of the failures is unknown for further analysis and improvement.

To resolve aforementioned challenges and limitations, our motivation is to propose novel neuron tracing methods addressing the most critical issues in the research field whilst maintains applicability in more general cases, towards a better understanding of neuronal morphology. To achieve this goal, it requires more data with accurate reconstruction for training and validation purposes, and a systematic failure analysis on advanced tracing methods to reveal our target.

Figure 1.3: A real neuron in optical image and its reconstruction (2D projection of a 3D image and reconstructoin) (Liu 2011).

## 1.2 Main Challenges

The difficulty of neuron tracing is ascribed to:

- The natural complexity of neuronal structure: neuron structure varies from functional role and areas of the brain in the nervous system. Neurons in Thalamic areas tend to have highly interwoven dendrites, causing severe merging-branches problems during reconstruction, as shown in Fig 1.4. Whereas neurons projected to cerebral cortex have large amount of axon termini, where signals are weak to detect and reconstruct.

- The image quality varies under different imaging condition e.g. sample preparation, imaging system, cell types etc. As a result, a same axon could be continuous in one image but discrete in another, sometimes even appearing to be punctuated and dislocated segments due to dissection and slicing of brain specimen (Peng et al. 2017), as shown in Fig 1.5. It hereby makes a complete tracing hard to achieve.

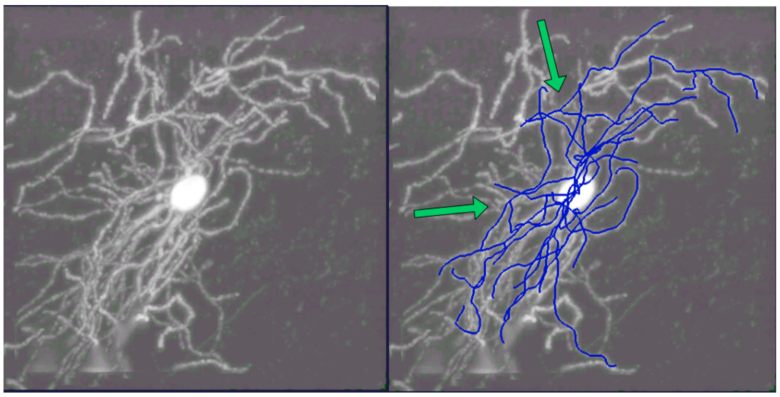

Figure 1.4: An example of the dendritic area of a neuron in optical image and a false tracing. The tracing is incomplete with merging branch where the green arrows point at. Note that this is a 2D projection of a 3D image and reconstruction.

- Specifically to large whole-brain image data, thousands of component blocks are imaged first then stitched together to constitute a 3D image stack (Zhou et al. 2018a). Time to image each component varies and thus the background is uneven, making an inconsistent background in the whole image stack. Currently no tracing method can adaptively adjust the contrast of different region and extract the morphology correctly.

- Lack of failure analysis on existing neuron tracing methods. Tens of neuron tracing methods are developed to tackle different issues occurred in neuronal image. But the frequency of occurrence of those issues is not clear, and the main cause of failure remains unknown.

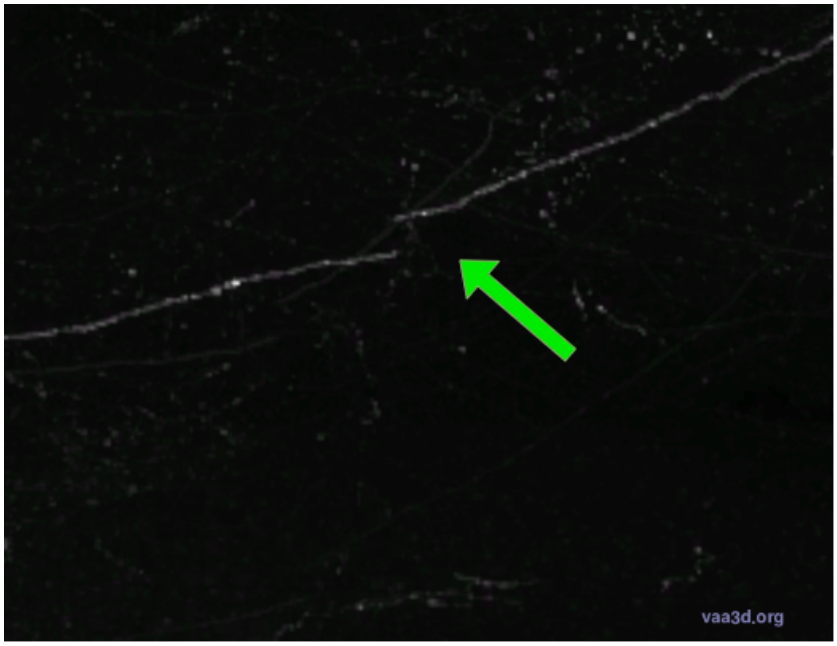


Figure 1.5: A punctuated and dislocated axon long projection area.

- Computational limitation: the raw image for reconstruction is 3D volumetric data, and the memory footprint of voxel representations grows cubically with resolution, limiting any implementation to a small size. On the contrary, an image for a single neuron can easily reach $1000 \times 1000 \times 1000$ voxels.

- Public data imperfection: Although DIADEM (Ascoli 2008) and Bigneuron (Peng et al. 2015) provides neuronal images and their corresponding manual reconstruction for algorithms development, the quantity of the public samples is relatively limited for learning purposes. DIADEM and Bigneuron provides 370 and 166 neurons repectively. This limits further development of learning-based algorithms.

## 1.3 Aims and Objectives

This thesis aims to develop efficient automatic 3D tracing algorithm to accurately reconstruct the three-dimensional morphology of a single neuron.

Based on our aim, we identify our research questions and objectives as follows:

- **What is the main obstacle in automatic neuron tracing?** Since the launch of DIADEM challenge, tens of automatic neuron tracing methods have been proposed, achieving varying degrees of success on different issues. But we still lack a knowledge about the main obstacles in neuron tracing. It is fundamental to identify the most critical issue to tackle to accomplish our aim.

- **How to generate more standard data for algorithm development?** Learning-based methods largely rely on massive amount of annotated data. While in the research field, the amount of available standard data is limited. The generation of more standard reconstructions, with accurate label is a key procedure and challenging task to accomplish in our research .

- **How to propose an automatic tracing tool which tackles the main obstacle?** After we identify the main obstacle in neuron tracing and generate sufficient amount of data, it is challenging to propose a novel approach targeting at the obstacle while maintaining applicability in general cases recurrently.

Motivated by aforementioned challenges, this thesis aims to achieve three major objectives:

- Our first objective is to generate more standard reconstructions for algorithm development. In particular, we focus on designing a pipeline to systemically produce accurate reconstruction. This objective is achieved in Chapter 3.

- The second objective is to systemically and quantitatively analyse failed cases of existing auto tracing methods, to know the main obstacle in the research field. We focus on establishing a workflow for automatic false samples detection and collection, followed by manual inspection and classification. This objective is explored in Chapter 3.

- Finally, we aim to develop efficient automatic 3D neuron tracing algorithm. We target at the main issues identified in Chapter 3 and propose our methods accordingly. We test them on different datasets to test their robustness. This objective is addressed in Chapter 4 and Chapter 5.

## 1.4 Contributions

The main contributions of this thesis is to propose novel automatic 3D neuron tracing methods from optical microscopy images. Our main concern is to reveal the main cause of failed tracing and develop algorithm accordingly to improve. The detailed contributions in each chapter are summarised as follows:

- In Chapter 3 we propose a pipeline for complete morphology reconstruction of single neurons. The pipeline consists of four stages as initial reconstruction, semi-manual reconstruction, cross-check and automatic refinement. Specifically, we propose a novel approach for the automatic refinement procedure, by a 2-time optimal path finding and deformation. Following the pipeline, we manage to generate more than 1,000 standard reconstructions, which significantly eases the limitation by lack of available data in the research field. With the data acquired, we conduct a systematic and quantitative failure analysis regarding existing automatic tracing approaches. more than 10,000 samples of false tracing are collected, manually inspected and classified. The analysis indicates that the main obstacles in neuron tracing are the close but irrelevant signals, as well as weak and punctuated relevant signals. To our best knowledge, the quantitative failure analysis is the first attempt in the research field.

- In Chapter 4 we propose a fully-automatic neuron tracing method targeting the major obstacles found in Chapter 3. We jointly embed occupancy learning with skeleton extraction to reconstruct neurons from 3D

optical microscopy images. It is the first attempt to address the sparseness in neuron reconstruction via predicting the occupancy of balanced input points sampled in 3D space. By leveraging occupancy learning with an equal point sampling strategy to control the data distribution in the input, it tackles the root causes of the imbalance problem, and eliminates the sparse redundancy. We trained and test the model on our dataset curated in Chapter 3. The experimental results show it outperforms other existing tracing methods.

- In Chapter 5 we propose another novel automatic 3D neuron tracing approach, to further tackle the main obstacle revealed in Chapter 3. We propose an deep learning model for neuron tracing, by learning skeleton extraction directly from a raw 3D volumetric image. To our knowledge, it is the first attempt in the research field. We design a set of deforming line primitives and a regularised loss function to preserve topology, overcoming the critical issue of false tracing caused by close but irrelevant signal. We evaluate our method with different dataset, and the experimental results show that it outperforms existing tracers in a variety of image scales and neuron types.

## 1.5 List of Publications

- Peng, Hanchuan, Peng Xie, ..., Zhangcan Ding, ..., Yun Wang, Hongkui Zeng. "Morphological diversity of single neurons in molecularly defined cell types." Nature 598.7879 (2021): 174-181.
- Zhang, Zhouzhou, Xiao yao, Zhangcan Ding, Tianyi Huang, Yan Huo, Runan Ji, Hanchuan Peng, Zengcai V. Guo, "Multi-scale light-sheet fluorescence microscopy for fast whole brain imaging." Frontiers in neuroanatomy (2021): 63.
- Zhangcan Ding, Hongchuan Yu, Yinyue Nie, Jinlu Zhang, Jianjun Zhang, Liya Ding, "3D Neuron Tracing via Occupancy Learning and Curve Skeleton Extraction". International Conference on Medical Image Computing and Computer Assisted Intervention 2021, submitted.

## 1.6 Outline of Thesis

This thesis is organised as follows:

- Chapter 2 reviews existing neuron tracing methods and deep learning applications in the research field.
- Chapter 3 includes the method for standard reconstruction data generation and a quantitative failure analysis.
- Chapter 4 proposes an automatic 3D neuron tracing method via occupancy learning and curve skeleton extraction.
- Chapter 5 proposes an further improved automatic tracing method via learning topology-preserving skeleton extraction.

# Chapter 2

# Literature Review

In this chapter we discuss previous works related to neuron tracing. First, we review the workflow of characterising neuronal morphology and demonstrate how neuron tracing influences as a core stage of it. Then we introduce several existing neuron tracing methods, with their applicability and performance. Finally, we review corresponding deep learning based applications in neuron tracing.

## 2.1 Neuronal Morphology Characterisation

In late 19th century, Santiago Ramón y Cajal foundationally revealed that the nervous system is made up of discrete individual cells (y Cajal 1911). These cells i.e. neurons are characterised with distinct input and output sides, the dendrites and axons respectively. By drawing the structures of thousands of different neurons, Cajal demonstrated the extraordinary diversity of neuronal morphology across brain regions and species. According to the neuronal morphology, he categorised a few of neuron types and successfully predicted their connections and elucidated their functions in some specific brain regions e.g. hippocampus, retina (Ramón y Cajal 1928). These outstanding discoveries were awarded with Nobel Prize in 1906, and ever since then, neuronal morphology has been recognised as a key feature to determine its neuronal type, connectivity, synaptic integration, and most importantly, its functional role in the neural circuits (Wang et al. 2019b).

Despite the preliminary success, now more than 100 years have passed but we still lack an accepted catalogue of neuron types (DeFelipe et al. 2013). The progress is greatly limited by the insufficient exploration of neuronal morphology at the level of single neuron. In modern digital era, to systematically characterise the morphology of neurons, a typical workflow consists of five steps: labelling, imaging, tracing, registration and classification. Fig 1.2 illustrates the pipeline of the process (Peng et al. 2020a).

Neurons must first be fluorescent labelled with a dye or transgenic tracer to reveal structure. Next, optical microscopy methods are utilised to capture the image. Then, a computational tracing tool extracts the morphology of the neuron from the image. The reconstructed results is then registered to a common template for final classification and analysis, to know its connectivity and function. In recent years, the upstream process of neuron tracing i.e. labelling and imaging has made substantial advancement: novel transgenic or viral Cre delivery techniques e.g. TIGRE2.0 (Aransay et al. 2015, Han et al. 2018, Lin et al. 2018, Veldman et al. 2020) have enabled efficient, sparse yet consistent labelling of a wide range of neuron types across multiple brain area; high-resolution and high throughput imaging techniques, such as fluorescence micro-optical sectioning tomography (fMOST) (Gong et al. 2016) and MouseLight (Economo et al. 2016), generate terabytes of high-resolution whole-brain fluorescent datasets at micron level. Massive amount of data intuitively helps the analysis of neuronal morphology. However, the neuron tracing or reconstruction remains a key bottleneck in the whole pipeline, significantly limiting downstream analysis and comparison of neuronal morphology. Currently, neuron tracing mainly relies on human labour for annotation, which is tedious, time-consuming and prone to error (Acciai et al. 2016). Aimed at stimulating advancement of automated morphology reconstruction, In 2009 the DIADEM Challenge (Gillette et al. 2011), short for Digital reconstruction of Axonal and DEndritic Morphology, was launched to identify an robust morphology reconstruction tool that could improve the efficiency for manual editing by 20 times. Considering the large amount of time involved in manual tracing as it might require days and weeks to complete reconstruction of a single neuron, the factor looks

conservative. However, none of the finalists achieved this landmark. The outcome indicates the situation of automating 3D neuron tracing as an open problem. It is therefore highly desirable for the development of automatic neuron tracing methods to standardise and accelerate the process.

## 2.2 Neuron tracing methods

Available neuron tracing methods vary in their degree of automation (Meijering 2010, Donohue and Ascoli 2011, Svoboda 2011) . Regarding to the amount of human intervention required, four main classes can be categorised (Myatt et al. 2012):

- Manual tracing
- Semi-Manual tracing
- Semi-Automatic tracing
- Fully-Automatic tracing

### 2.2.1 Manual Tracing Methods

The first ever neuron reconstruction is done by Cajal manually, with pencil and ink depictions. Armed with a rudimentary microscope, he drew and presented the neuronal structure by free hand. An example of his drawings is shown in Fig 2.1, and it can be seen that the neuronal morphology is already close to the real structure. To further improve the accuracy, in later era manual tracing is implemented with camera lucida: anatomists employ prisms to visually overlay the microscope image onto a piece of paper, and delineates it by hand. This technique was primarily proposed for 2D tracings, but can be used for 3D reconstruction in a more complicated manner: the microscope images are traced and serially aligned and projected onto a film, and the results are filmed for visualisation (Senft 2011). The processing is highly time-consuming and prone to error.

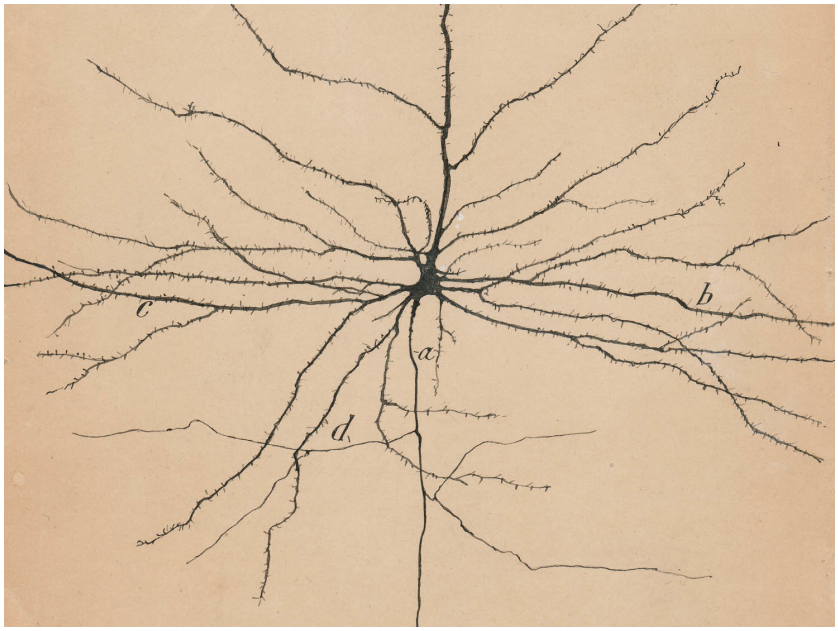


Figure 2.1: A neuron structure manually drawn by Santiago Ramón y Cajal (Swanson et al. 2017).

### 2.2.2 Semi-Manual Tracing Methods

This kind of method involves with computer software to reconstruct the digital neuron structure by hand, typically beginning at the soma and trace the whole structure following the signal in the microscope image. A typical illustration is shown in Fig 2.2. Specifically, this is the mainstream for reconstructing the "gold-standard" neuronal morphology in current research field (Peng et al. 2011b, Mosinska et al. 2017, Haehn et al. 2018).

Neuron_Morpho (Brown et al. 2005) is a sufficient tool to trace neuron and largely contributes to the widely used NeuronMorpho database (Ascoli et al. 2007). The idea is similar to primary manual tracing. The acquired stacked image tiles for the target neuron is loaded and merged on computer, and then projected on the X-Y planar as a map for visualisation on monitor. The user has to scroll through each cross-sectional Z slice to locate and profile the object of interest. Along with a mouse drawing, the tool records the line and measures the neurite diameter along the line and save the information. Apparently, if the image is relatively large and the neuron is complicated,

the operation would be cumbersome and low-throughput. Furthermore, this essentially 2D process intuitively biases the understanding of intrinsic 3D properties of neuronal morphology.

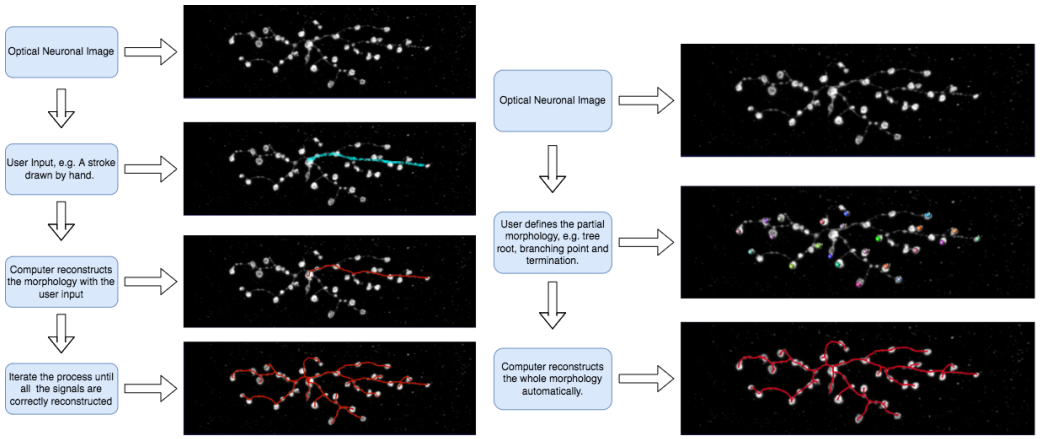


Figure 2.2: An comparison of semi-manual tracing (left) and semi-automatic tracing (right).

To overcome the obstacle, Virtual Finger (VF) (Peng et al. 2014b) in vaa3D (Peng et al. 2010b, 2014a) was developed to efficiently explore the complicated 3D image content. VF aims to map the identified 2D user input via computer screen back to the 3D volumetric space of the image. It is implemented through a 3D point-pinpointing algorithm (PPA) and curve-drawing algorithm (CDA). As illustrated in Fig 2.3, after the user inputs a mouse stroke on 2D screen, the method uses the trajectory of cursor locations sampled from the stroke, and generates a shooting ray orthogonal to the screen. The PPA locates the curve knot along each shooting ray by maximal intensity. Then a 3D curve is generated in a piece-wise manner, connecting each consecutive knot pair by finds the shortest geodesic path between them via Dijkstra algorithm (DIJKSTRA 1959). The geodesic distance between two immediately adjacent voxel $v$ and $u$ is defined as:

$$e(v,u) = d(v,u) \cdot \left( \frac{g_{\mathrm{I}}(v) + g_{\mathrm{I}}(u)}{2} \right) \tag{2.1}$$

where $d(v, u)$ denotes the Euclidean distance between $v$ and $u$ and $g_I(.)$ is defined as:

$$g_{\mathrm{I}}(x) = \exp\left(\lambda_{\mathrm{I}}\left(1 - \frac{I(x) - I_{\min}}{I_{\max} - I_{\min}}\right)^2\right) \tag{2.2}$$

where $\lambda_{\mathrm{I}}$ is a weighting factor, $I(x)$ denotes the intensity,$I_{min}$ and $I_{max}$ is the minimal and maximal intensity in the image.

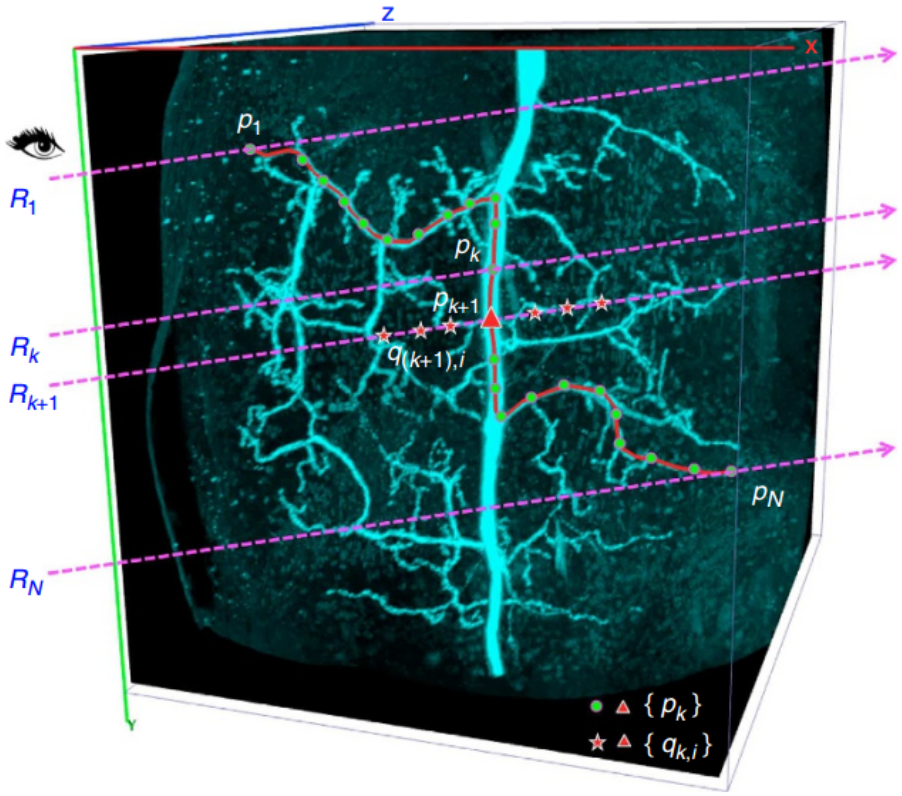


Figure 2.3: illustration of curve drawing through Virtual Finger using one mouse stroke painted in the 2D projection of a 3D neuronal image. $R_1$ - $R_N$ represents the shooting rays orthogonal to the computer screen and parallel to each other. $p_1$ - $p_n$ represents the estimated 3D location of each curve knot,corresponding to each shooting ray respectively. $q_(k, i)$ are the one-voxel evenly spaced 3D locations along the kth rays (Peng et al. 2014b).

Therefore VF can generate 3D curves points and regions-of-interest with a single mouse operation and greatly increases the efficiency of reconstruction.

Along with the Virtual Finger, a multi-resolution/multi-scale octree representation of an image stack called TeraFly (Bria et al. 2016) is proposed to explore and reconstruct massive volumetric image data smoothly. For a given large image data, TeraFly iteratively downsamples it by 2 times on each X,Y and Z dimensions to a relatively small volume (typically, 512*512*256 voxels), the final and intermediate smaller volumes are all saved to files and indexed using a tree data structure, in which each tree node stands for an image block. As illustrated in Fig 2.4, a whole mouse brain image data (about 2 terabyte) is converted in such pyramid structure for visualisation and reconstruction from coarse (low-resolution) to fine (high-resolution) level. And with the help of VF, the user can navigate of any region of interest and reconstruct neuronal morphology. This combination of vaa3d-TeraFly technique enables researchers to visualise and reconstruct neuronal morphology accurately and efficiently, for any given size. This method is used in Chapter 3 for our data generation.

Although Virtue Finger and TeraFly has achieved great accuracy and efficiency in practical use, the 2D working environment in the context of traditional desktop software still constrains the profiling and understanding of neurons in 3D image. If a neuron is highly complex at some point or the signal is relatively weak, resulting in an ambiguous structure, it will even pile up the difficulty to observe and manipulate higher dimensional data via a lower dimensional interface (McDonald et al. 2020). In such case, virtue reality or immersive environment has been introduced to overcome the limitations by 2D desktop interaction and visualisation modalities. TeraVR (Wang et al. 2019a) is proposed in line with such demand, as it provides the user a combined stereroviews to visualise and interact with the data in 3D directly. The overall scheme of TeraVR is shown in Fig 2.5. Integrated with the vaa3d-TeraFly framework, TeraVR enables observation and reconstruction on ultra large data. Particularly, in areas with complicated branching patterns, weak signals, overlapping neurites, or multiple densely packed neurons, which are common issues in 2D interface, TeraVR outperforms traditional semi-manual tracing methods and enables accurate and fast tracing. Also, it provides collaboration mode for multiple annotators to instantly share and proofread the

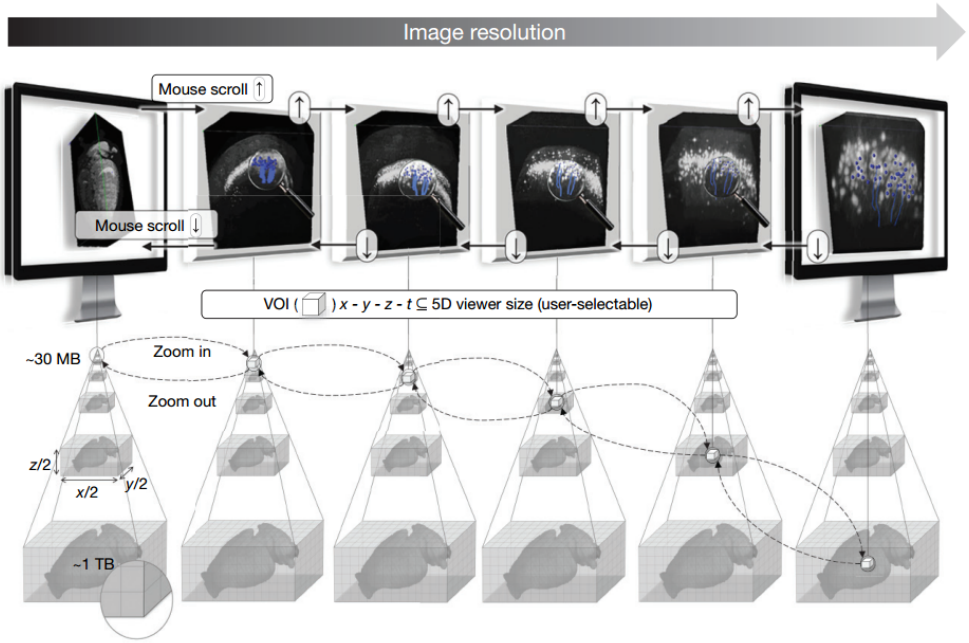


Figure 2.4: The progressive 3D image visualisation and exploration by TeraFly. With zoom-in and zoom-out, the multi-resolution image tiles are demonstrated in higher/lower resolution (Bria et al. 2016).

reconstruction. It largely extends the applicability of vaa3d-TeraFly technique. This technique is also employed in Chapter 3 to reconstruct our data, specifically in relatively complex areas.

### 2.2.3 Semi-Automatic Neuron Tracing Methods

Semi-automatic methods requires the user to define some partial morphology, such as identifying the tree root, branching point and termination, and the neurite paths are traced by the computational tool. A typical illustration is shown in Fig 2.2.Note that compared with semi-manual neuron tracing, this type of methods changes the way of user input, but it also relies on computational tool to conduct the tracing part.

In 2004, Meijering et al. (Meijering et al. 2004) presented a semi-automatic tool to trace individual neurites in 2D microscopy image. Based on the LiveWire algorithm (Barrett and Mortensen 1997, Falcão et al. 1998), the

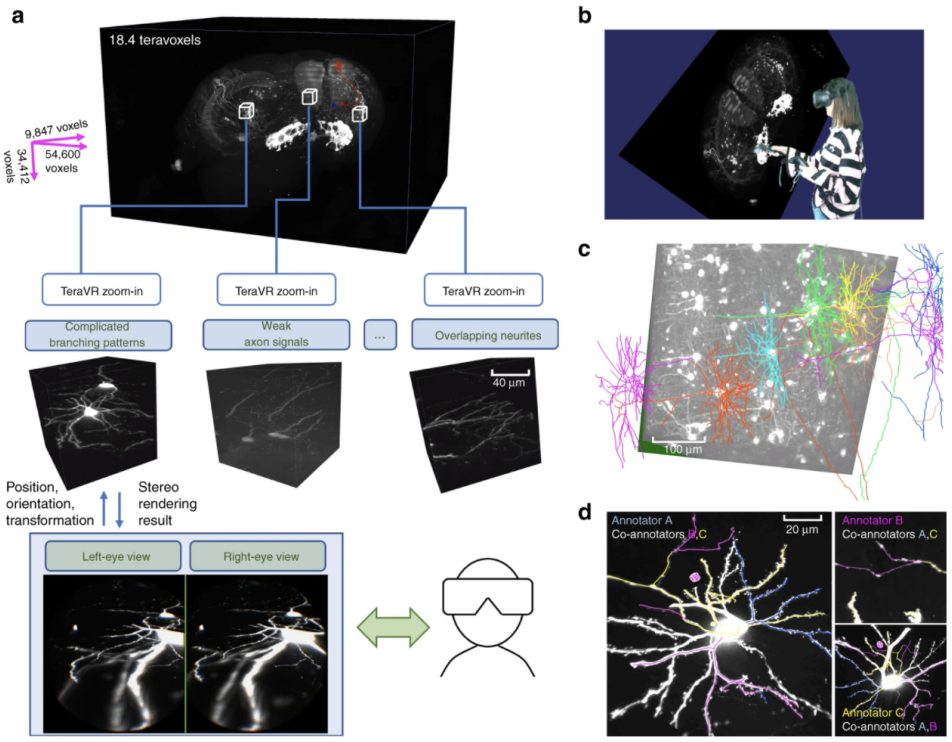


Figure 2.5: The overall scheme of TeraVR (Wang et al. 2019a).

tracing procedure consists of two main phase: First, the tool preprocesses the input image automatically by an steerable Gaussian filters (Freeman and Adelson 1991), giving a likelihood of each individual pixel belonging to a neurite; then the user gives a start and end point, and the tool will find the optimal path between them, linking consecutive ridge pixels at the centre of the signal. Compared with manual tracing, this method improves reproducibility and reduces human labour intensiveness. However,the 2D working environment brings ambiguities regarding the branching or crossing of neurites, and obstacles the understanding of 3D structure of real neuron.

To extend Meijering's method to a 3D image stack, Neuromantic (Myatt et al. 2012) was proposed in 2012. Sharing a similar idea as previous method, Neuromantic requires user input to define the starting and ending point. Then the tool automatically identifies pixels within the image stack that are likely to belong the the neuronal structure, and furthermore finds the

most likely flow direction for each pixel by analysing the eigenvectors of the estimated Hessian matrix. By transforming the given image to a graph structure, with each node corresponding to a pixel, the tool employs the Dijkstra's algorithm (DIJKSTRA 1959) to find the lowest cost route between the user-defined start and end point.

Overall, semi-automatic neuron tracing methods reduces the human labour to a certain degree, but compared with the advanced semi-manual methods, the accuracy is surpassed as a result of the automatic tracing step without, particularly in the noisy and fuzzy image area. So it is not widely used as an alternative for neuron tracing in real world.

### 2.2.4 Fully-Automatic Neuron Tracing Methods

Fully-automatic neuron tracing methods extracts the neuronal morphology with minimal user input. The ever increasing amount and volume of available microscope image data calls for automation for neuron tracing. Thanks to DIADEM challenge (Gillette et al. 2011) and BigNeuron project (Peng et al. 2015), a few interesting automatic algorithms have been proposed. Currently, these methods can be divided into three categories: global-processing, local-processing, and meta-methods (Acciai et al. 2016).

- **Global processing methods** extract 3D morphology of a single neuron from its global signal distribution in the optical image. A straightforward implementation is to use a segmentation algorithm to obtain a binary foreground/background image, and apply a 3D binary skeletonisation algorithm to extract the neurite structure (Dima et al. 2002, Weaver et al. 2004, Evers et al. 2005, Narro et al. 2007). However, a sufficient binarisation can be difficult to achieve. To enhance the line structure when doing skeletonisation, Hessian-based vesselness methods are employed (Abdul-Karim et al. 2005, Yuan et al. 2009, Vasilkoski and Stepanyants 2009), but it requires prior information and is computationally expensive when the image is large. Alternatively, it can be converted to a shortest path problem by transforming the image into

an graph, with designed node and edge representation, and uses Dijkstra's algorithm (Dijkstra 1959) to extract the skeleton (Zhang et al. 2007a, Peng et al. 2011a). However, the performance of these methods largely depends on the geodesic distance definition and seed (soma and termini) placement. Moreover, the computational storage to hold the graph will be intensive if the image is relatively large.

- **Local processing methods** start from a given location (seed) and explores an image only around relevant structure. An early representative is proposed by AL-Kofahi et al. (Al-Kofahi et al. 2002) that uses template fitting to determine the tracing direction . The template consists of four parallel edge detectors, and each of them used a greedy search to find a part of the branch boundary. This method can only trace one branch each time and must be cooperated with a branch point detection method to complete the tracing. Zhao et al. (Zhao et al. 2011)proposed a similar approach but defined a 3D cylindrical template, assuming that a neurite segment can be modelled a series of elliptical cylinders. For each branch to be traced, the algorithm needs at least one seed, which are chosen as local maxima among the points with a function of intensity and local geometry features above a threshold. The tracing procedure starts from fitting the cylinder template on defined point $c_i$, and then define the next point $c_{i+1}$ shift from $c_i$ one step along the axial direction, and iterates these steps until the fitting score is lower than a threshold. When all the seeds are traced or covered, the tracing ends. Feng et al. (Feng et al. 2015) further improved the algorithm by replacing the cylindrical mode with a tree model and proposed Neutube. However, it is still prone to error in areas where the signals are weakly continuous or multiple branches are densely packed.

- **Meta-methods** enhance any existing tracing to extend its applicability. Chen et al. (Chen et al. 2015) proposed SmartTracing as a self-learning framework with any existing neuron tracing method to enhance their performance. It first produces an initial neuron reconstruction with the user-provided method, and accordingly, it estimates

a confidence score regarding the likelihood of reconstructed unit to identify some reliable portion of reconstruction. Then it uses the reliable parts as training exemplars in a machine learning framework to extract wavelet features and predict the probability map of the whole image for the final tracing. Another direction of enhancement is to apply any existing method to trace in large image volumetric data. Whether the algorithm are global-process or local-process, they are mostly designed for relatively small volumetric data set. The DIADEM competition (Gillette et al. 2011) aims to do the reconstruction on optical images up to several gigabyte. However, with the significant advancement in microscopic technique, now neuroscientists can collect data on whole mouse brain which is terabyte-sized. Under this circumstance, most solely-used neuron tracing methods will fail. Zhou et al. (Peng et al. 2017) proposed a framework called Ultratracer that could aggregate a variety of different previous tracing methods and enable it to reconstruct very large neuron morphology. The core idea of Ultratracer is to sequentially trace small blocks of a 3D image. As illustrated in Fig 2.6, given a large-scale 3D image and the soma location, it first extracts a small cuboid block centred in the soma and do the reconstruction using any previous tracing algorithms. Then, it finds the terminal points on the six cuboid faces, and processes the next blocks centred in those points. It iterates until no terminal points are detected and assemble the final reconstruction. Thus, it could extend any previous neuron tracing algorithms to handle massive amount of data. In Chapter 3 we adopted this method to generate auto reconstructions from full mouse image data. In Chapter 5, we integrate our method into Ultratracer for tracing whole mouse-brain image data.

## 2.3 Deep Learning Applications in Neuron Tracing

In recent years, the rapid advancement in convolutional neural network (CNN) has enabled deep learning methods to outperform traditional methods in

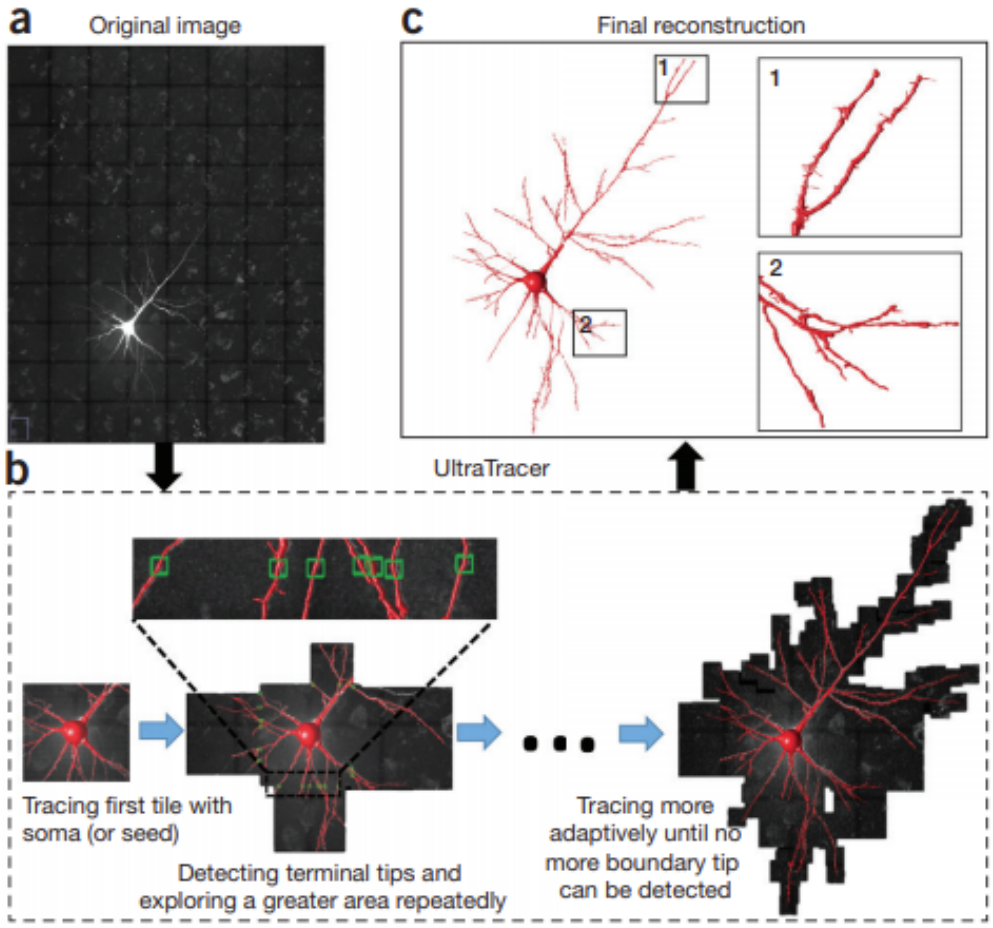


Figure 2.6: pipeline of ultratracer tracing a large 3D volumetric image. (Peng et al. 2017)

many pattern recognition and computer vision applications (Ji et al. 2012, Krizhevsky et al. 2012, Zeiler and Fergus 2014, Zeng et al. 2015). Taking advantages of its ability of capturing highly non-linear relationship between inputs and outputs (LeCun et al. 2015), deep learning based techniques can be introduced into neuron tracing procedures to enhance the performance, either as a pre-processing step for neuronal image segmentation (Li et al. 2017, Li and Shen 2020, Yang et al. 2021), or post-processing step for segments connection and refinement (Zhou et al. 2018a). Here we review some

related deep learning applications in neuron tracing.

### 2.3.1 Image Segmentation

Li et al. (Li et al. 2017) proposed a fully convolutional network (FCN) with 3D convolutions for segmenting the neuronal morphology from optical microscopy images. The network consists of several fully convolution and deconvolution layers to ensure the feasibility of end-to-end training. After inference, it generates a probability map $P$ with the same-size as the input image $I$. Instead of directly tracing on the probability map, the author combines the original image and the probability map to form an adjusted image $F(x)$ by:

$$F(x) = \alpha\tilde{I}(x) + (1-\alpha)\tilde{P}(x) \tag{2.3}$$

where $\alpha$ is a weight to adjust the contribution of image intensity and probability map. This combined representation sufficiently suppresses the noise signal while keeping the global structure and is used in our method in Chapter 4 for testing purposes. The author applied several existing tracing methods onto the enhanced images and find most of their performances have been improved compared with tracing on original images.

Nevertheless, to ensure a meaningful result, FCN requires large amount of training samples, considering the numerous parameters in the model. In the case of neuron tracing, it becomes a bottleneck due to the limit amount of available annotated data. After the success of U-net (Ronneberger et al. 2015) in 2015, a few of U-net variants have been applied to neuron tracing for image segmentation, for better efficiency of utilising the limited annotated samples. Li et al. (Li and Shen 2020) proposed a 3D U-net-based network called 3D U-net plus focusing on tracing neuron in tangled neuronal image. As illustrated in Fig 2.7, as a variant of U-net, the network consists of encoder-decoder architecture and feature fusion. The encoder extracts hierarchy features and the decoder gradually recovers the probability map for the image. A skip pathway is added between the symmetric module to concatenate the features extracted from an encoder to that from the corresponding decoder. Additionally, it replaces the original convolution and

pooling layers with atrous convolution (Holschneider et al. 1990, Chen et al. 2017) and Atrous Spatial Pyramid Pooling (ASPP) (He et al. 2015) respectively to capture more global and contextual information. Similarly, Yang et al. (Yang et al. 2021) proposed a 3D nested U-net (Unet++) (Zhou et al. 2018b) based approach, re-designing the skip pathway with a series of nested and dense modules to reduce the semantic gap between the features maps of the corresponding encoder-decoder. It claims to improve the accuracy and length of neuron reconstruction as a preprocessing procedure for tracing.

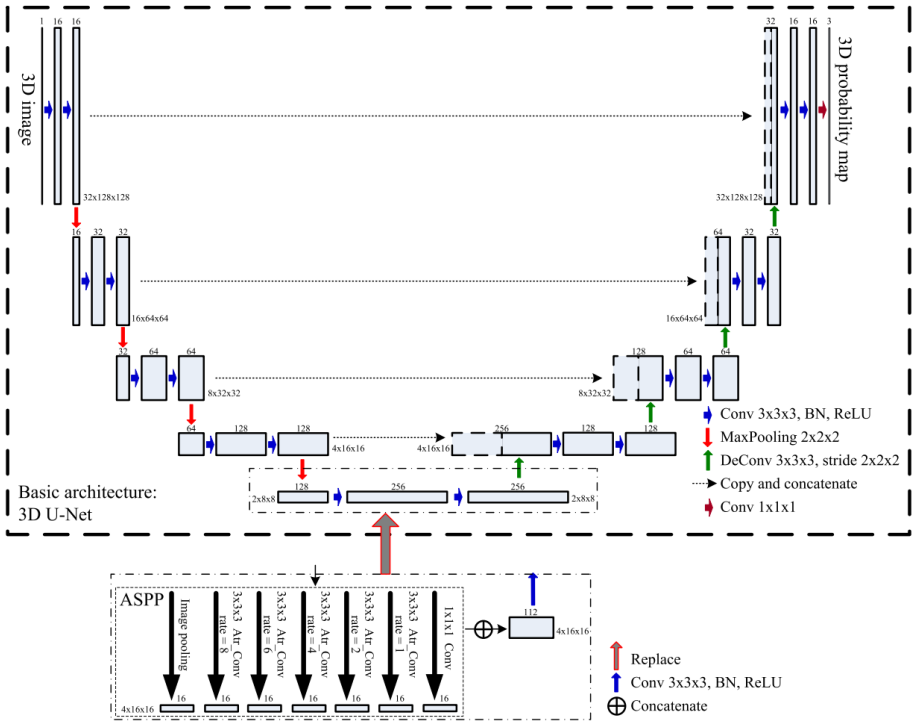


Figure 2.7: structure of a U-Net based deep neural network. Each box represents a multi-channel feature map. Top of each box: channel number. Lower left and lower right of the box: the z-y-x-size (depth-height-width) (Li and Shen 2020).

Despite of the higher efficiency of data utilisation, the limit amount of annotated data still significantly constrains further improvement on neuronal image segmentation. Under this circumstance, a few of weakly supervised

learning approaches have been developed. Zhao et al. (Zhao et al. 2019 2020) proposed a progressive learning framework for neuronal reconstruction. As illustrated in Fig 2.8, the network mainly consists of a segmentation module, an image enhancement module and a tracing module. The segmentation module takes an input image through 3D deeply supervised network (DSN) (Dou et al. 2017) to output a probability map. Then the image enhancement module combines it with the original image as Equation 2.3. Next, The tracing module applies a traditional tracing method onto the enhanced image and produces pseudo labels for each voxel, which is then fed back to train the segmentation network. With more iterations of the network training, the network is cable to capture more distinctive and long-distance trace progressively. Huang et al. (Huang et al. 2020) proposed a similar framework, taking advantages of both Residual CNN and traditional tracing. Additionally, it adds a refinement step after tracing for more accurate pseudo-label. However, these methods largely rely on the performance of the selected tracing methods and quality of original image, and tend to fail if the initial trace is over-reconstructed.

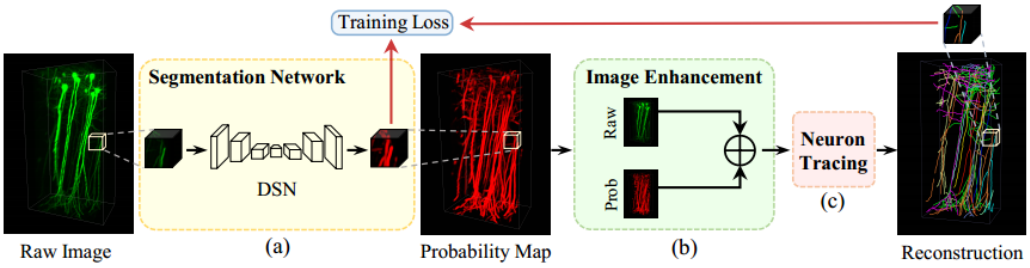


Figure 2.8: structure of a deep learning network mutual with a traditional tracing method (Li and Shen 2020).

### 2.3.2 Neuron Tracing Refinement

Zhou et al. (Zhou et al. 2018a) proposed DeepNeuron, a deep learning based toolbox for post processing after neuron tracing. For under-reconstruction, where there are false negative gaps between the neuronal segments, it is able

to find the continuity of neurites and connect the broken segments through a trained deep learning based Minimal Spanning Tree (DMST). It employs a revised Siamese network (Bromley et al. 1993, Chopra et al. 2005), in which two subjects are fed into the network to find the similarity or the relationship between them. After the inference, a dissimilarity score is given and segments pairs with low error will be connected through the Minimal Spanning Tree algorithm (Pettie 2008). For over-reconstruction, where false positive trace occurs, it is able to prune the false reconstructed components. By transforming the pruning task into a classification problem, it trains a CNN network to classify each tree node as false positive or true positive and prune the false ones. However, due to the limitation of GPU memory, DeepNeuron uses 2D CNN models. The input image is first projected on 2D and after inference mapped back to 3D. It apparently loses some degree of the rich information in 3D, and the projection and mapping procedure can bring distortion to the neuronal morphology.

## 2.4 Neuron Tracing Project and Dataset

In this section, we will introduce some projects in neuron tracing and their dataset which is widely used in the research field, as well as in this dissertation. Furthermore, we will discuss how they motivate our works in Chapter 3.

### 2.4.1 DIADEM

In 2009, the DIADEM Challenge (Gillette et al. 2011), short for Digital reconstruction of Axonal and DEndritic Morphology, was launched to stimulate robust tracing with minimum human involvement. It provides a dataset for the participants to develop their algorithms and bench test the performance. The dataset consists of 6 sets of data collected from different animal species (e.g. rat, drosophila, mouse, cat), brain regions (e.g. cerebellar,hippocampal,neocortical), structures (e.g.axons,projection fibre) and imaging methods (e.g. transmitted light brightfield, confocal microscopy). Each neuron has its image stack along with a "gold standard" reconstruction. An example is shown in Fig 2.9. The quality of the "gold standard"

reconstruction is controlled by a workflow collaborated by the data providers and the peer reviewing community. A same neuron is manually or semi-manually traced by two independent people to gauge the subjectivity. The agreed neuronal reconstruction is then undergone thorough checks by peer review, to eliminate operator mistakes such as missing branches, incorrect termination or connectivity. Each proposed edit was systemically verified with the original annotators to confirm if it purposed to be traced that way or it is indeed an error. By minimising the influence of human mistakes, the quality of the dataset is ensured. The design of cross-check is also exploited in our work in Chapter 3. However, the dataset is set for development of conventional automatic neuron tracing algorithms and only contains 370 relatively small neurons (most of them size up to 512x512x512). Therefore, it is not ideal for development of deep-learning based neuron tracing methods.

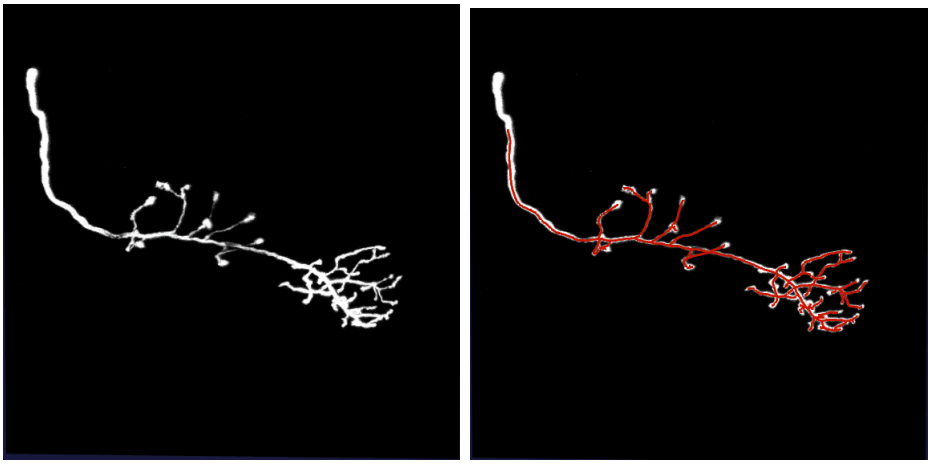

Figure 2.9: A drosophila olfactory axonal projection image stack (left), with a corresponding "gold standard" reconstruction (right) (Brown et al. 2011).

### 2.4.2 BigNeuron

Although the DIADEM challenge stimulates progress of automatic neuron tracing development, the original goal of the contest is not reached, which is to speed up the tracing process by 20 times compared with manual tracing. One practical limitation of DIADEM was that the reconstruction methods

were proposed in different programming languages, ran on different software platforms and followed different protocols for data I/O. In such way, a direct comparison of different methods remained as a main problem, and further blocked aggregation and consolidation of the cumulative progress on automated neuron tracing in a practically useful application for neuroscience. One solution for this is to propose a common, versatile software to aggregate different tracing methods, data formats and data I/O protocols. This will allow bench test of any algorithm on large-scale neuron datasets for effective validation.

Therefore, the BigNeuron project (Peng et al. 2015) was launched in 2015 to achieve this target. It aims to gather a worldwide community to define and advance the state-of-the-art of neuron tracing, by bench-testing any automatic neuron tracing methods on any dataset following a standard data protocols and evaluation metric. The first phrase of the BigNeuron project is to release a relatively small "gold standard" dataset to boost the algorithms development. The dataset is manually annotated and cross-checked, following similar workflow as DIADEM. After a few algorithms were developed and verified their usability, they will be applied on any unreconstructed raw images to output a consensus reconstructions. Then, based on the consensus reconstruction, manual correction and reconstruction is conducted to output a standard reconstruction, with minimal human labour involvement thanks to the initial auto-reconstruction. This new robust reconstruction will then extent the dataset as well as further development. An illustration of the process is shown in Fig 2.10.

However, the BigNeuron still remains as an unsolved challenge. Now the dataset consists of 166 neuron reconstructions along with their corresponding image stacks, across different species and brain regions. The limited amount of available data still restrains further development of automatic neuron tracing, particularly deep-learning based approaches.

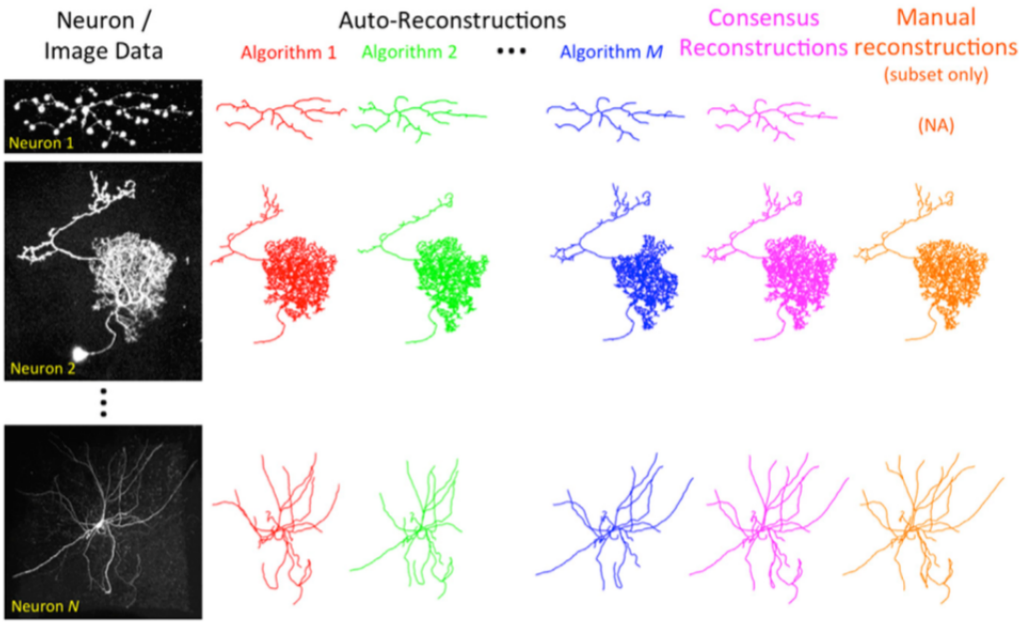


Figure 2.10: The BigNeuron gathers different automatic reconstruction methods and output a high-quality initial reconstruction. With minimal human labour for manual reconstruction, it will output a robust reconstruction and contribute to a common large dataset (Peng et al. 2015).

### 2.4.3 BICCN Brain Image Library (BIL)

In 2017, Brain Initiative Cell Census Network (BICCN) was launched to build a comprehensive understanding of cell types in mouse, monkey and human brain (Ecker et al. 2017). It integrates centres and laboratories around the world to generate large-scale datasets, methods, tools for the goal. All of the results are shared with the community. Among the datasets, we choose the whole mouse brain images for reconstruction in Chapter 3. A typical whole mouse brain image is shown in Fig 2.11. Such an image occupies around 2 Terabyte, and approximately 2000 neurons in it are sparsely labelled and imaged by the fluorescence micro-optical sectioning tomography (fMOST) system (Gong et al. 2016)with physical resolution as $0.35\mu$m $\times$ $0.35\mu$m $\times$ $1\mu$m. There are over 250 mouse brain images available in Brain Image Library (BIL) (Benninger et al. 2020), which is a public repository

under BICCN. Such amount of data is sufficient for our deep-leaning based algorithm development, but it requires precise reconstructions which is done in Chapter 3.

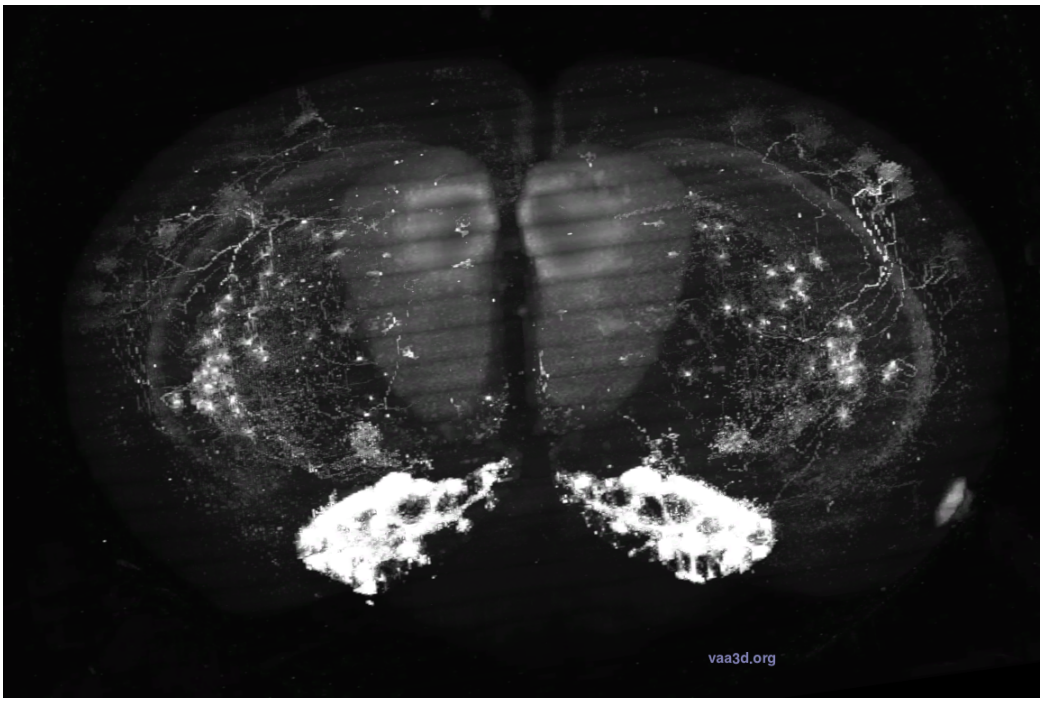


Figure 2.11: A typical whole mouse brain image data by BICCN BIL (Benninger et al. 2020).

### 2.4.4 NeuroMorpho

NeuroMorpho (Ascoli et al. 2007) is an open repository for single neuron reconstructions. It was launched in 2006 to provide free access to a variety of digital tracings of neuronal morphology online. The database is mainly contributed by data described in peer-reviewed publications. Today it has became the largest public database with over 80,000 neurons from diverse animal species and brain regions (Akram et al. 2018). Our reconstructions in Chapter 3 also contribute to this database.

Although the dataset is sufficiently large, the data are all reconstructions without corresponding image stacks, which means it cannot be directly used

for development of tracing algorithm. However, analysis on existing reconstructions in NeuroMorpho is still meaningful for our project. For example, a quantitative morphological measurements among a specific neuron type will give a common idea of how it should be structured. This will greatly help the quality check of any newly reconstructed neurons in the same type. Also we conclude some common features of neuronal reconstruction, which inspires the design of loss function in Chapter 5.

## 2.5 Data Format and Evaluation Metrics

To describe the neuronal morphology, few data formats have been proposed such as Neurolucida DAT/NRX/ASC/XML (MBF Bioscience, Williston, VT USA), GENESIS XML (Taylor et al. 1986). To fix any irregularities and ensure a base level of homogeneity and compatibility with different platform, NeuroMorpho database (Ascoli et al. 2007) and BigNeuron (Peng et al. 2015) select SWC format (Cannon et al. 1998) as the standard description of neuronal reconstruction, for visualisation and testing purposes. In this thesis, we also choose SWC format for any neuron tracing output. A typical SWC file and its visualisation are shown in Fig 2.12. A SWC file describes a simple directed tree model, and stores a points matrix with seven columns follows. Each row represents a node traced along the tree. Each column has following meaning:

1. ID: node identifier. A positive integer.
2. TypeID: type identifier. A positive interger denotes the type of the node, categorised into root, soma, axon, dendrite, apical dendrite and other custom types.
3. x: X-position in micrometers.
4. y: Y-position in micrometers.
5. z: Z-position in micrometers.
6. r: Radius in micrometers.

7. ParentID: Parent sample identifier. This defines the how nodes are connected to each other. Multiple nodes can refer to a same ParentID, which denotes the branch point. No loop is allowed.

```
#name
#comment
##n,type,x,y,z,radius,parent
1 2 0.72501 391.08 25 1 -1
2 2 4.2166 383.5 27 1 1
3 2 8.8883 376.49 28 1 2
4 2 14.136 370.06 30 1 3
5 2 19.991 361.9 32 1 4
6 2 22.903 356.08 33 1 5
7 2 25.815 350.22 35 1 6
8 2 28.151 343.79 37 1 7
9 2 31.063 338.54 38 1 8
10 2 34.582 331.53 40 1 9
11 2 36.342 326.86 41 1 10
12 2 37.494 323.37 42 1 11
13 2 38.678 319.28 43 1 12
14 2 40.983 313.46 45 1 13
15 2 42.743 310.54 46 1 14
16 2 45.078 304.11 48 1 15
17 2 46.838 298.86 48 1 16
18 2 49.75 294.77 49 1 17
19 2 52.086 290.1 50 1 18
20 2 54.422 284.27 52 1 19
21 2 56.181 279.6 53 1 20
22 2 59.093 273.78 54 1 21
23 2 63.189 267.35 55 1 22
24 2 68.437 263.83 56 1 23
25 2 73.108 257.43 57 1 24
26 2 80.692 250.99 59 1 25
27 2 85.94 246.32 60 1 26
28 2 98.196 238.16 61 1 27
29 2 105.78 232.91 62 1 28
30 2 116.31 227.64 63 1 29
31 2 129.14 218.32 64 1 30
32 2 138.48 213.65 65 1 31
33 2 155.41 209.56 66 1 32
34 2 165.33 207.22 66 1 33
35 2 172.91 203.13 67 1 34
36 2 179.34 201.4 67 1 35
37 2 176.4 192.63 68 0 36
38 2 175.82 186.81 69 0 37
39 2 178.74 182.13 70 1 38
40 2 181.07 180.37 72 1 39
41 2 186.35 179.19 75 1 40
42 2 188.08 178.61 77 1 41
```

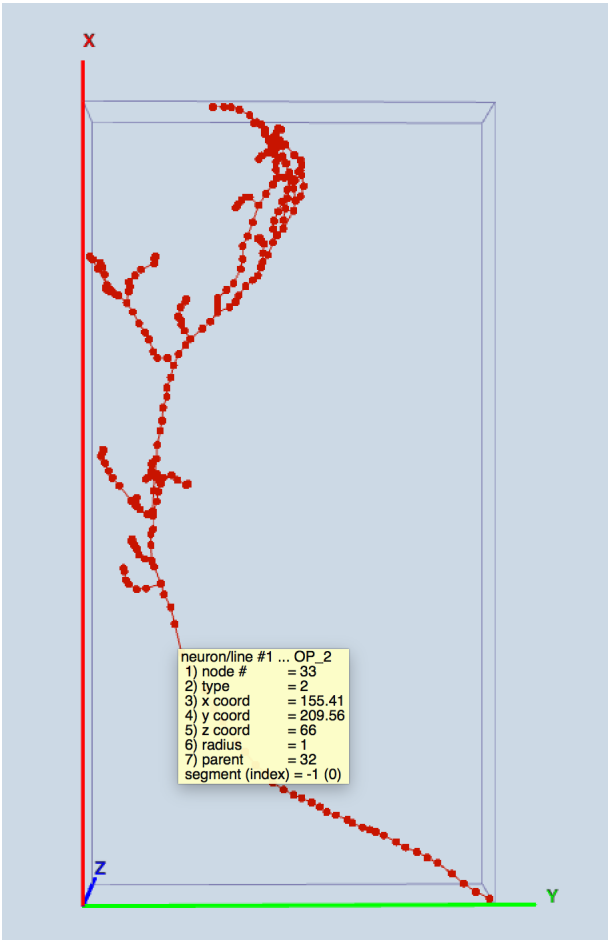


Figure 2.12: A swc file (partially displayed) and its visualisation. Note that each node is sized up for better view. Nodes with parental relationships are connected with a line.

To evaluate any existing or novel tracing approach, a common method is to compare the output swc file with the gold-standard (GS) reconstruction. There are several metrics to be used as following:

1. **Distance Metric**: the most common way to compare a reconstruction with the GS is based on the spatial error, which is defined as the de-

viation of the reconstructed centreline to the GS centreline. Different literature papers have adopted following ways to compute the distance metric:

(a) For each node in the reconstructed trace, it finds its nearest point in the GS. Then the spatial error is calculated as the average Euclidean distance between these nearest node pairs in the (Myatt et al. 2012).

(b) For the reconstructed and the GS trace, both of them are interpolated between the nodes to obtain a skeleton. Then two similarity factors are calculated to estimate how well the trace matches the GS and the GS matches the trace, respectively (Choromanska et al. 2012).

(c) The reconstructed and the GS traces are resampled so the distance between adjacent node is 1 voxel. Then two distances are calculated by averaging the Euclidean distance of all nearest nodes pairs from trace to the GS and vice versa. The final spatial error is the average of these two distances (Wang et al. 2011, Lee et al. 2012, Wang et al. 2011, Gala et al. 2014, Xiao and Peng 2013). It is also the metric we choose to evaluate our tracing algorithm proposed in Chapter 4 and Chapter 5.

2. **Length Metric**: this metric compares the difference of length between the reconstructed and the GS trace. A natural definition of spatial error is one minus the ratio between the length of reconstructed trace and the GS trace (Zhang et al. 2007b). Another approach is to focus on the correct reconstructed trace portion where the the two traces overlap, and calculate the ratio between the correct length and the overall GS trace length (Wang et al. 2011, Dietenbeck et al. 2014).

3. **DIADEM Metric**: the DIADEM metric compares two neuron reconstructions based on topology matches. For each node in a reconstruction, when locating nearest node in the other trace, only a restricted

area will be searched and a high penalty will be added if missing. Moreover, the metric takes into account the importance of each node and weights the spatial error accordingly: a mismatch of branch node will lead higher spatial error than a mismatch of a termination (Gillette et al. 2011).

# Chapter 3

# Data Generation and Analysis

## 3.1 Introduction

Neuron tracing is a key procedure in the workflow of neuronal morphology characterisation. In recent years, the substantial advancement in upstream procedure i.e. fluorescent labelling and microscopic imaging have enabled generation of massive amount of high-resolution and large-scale neuronal images (Acciai et al. 2016). Under this circumstance, manual tracing becomes time-consuming, error-prone and impractical. This has motivated the development of automatic neuron tracing methods. In 2009, the DIADEM Challenge (Gillette et al. 2011), short for Digital reconstruction of Axonal and DEndritic Morphology, was launched to call for novel approaches towards fully automatic tracing. The original goal of the challenge is to speed up the tracing process by 20 times compared with manual tracing without heavy loss of accuracy. Unfortunately, none of the finalist algorithms achieved the target. The fail of these traditional algorithms can be ascribed to the complicated nature of neuronal structure, large diversity of neuron types, varying image condition, inevitable image noise, and other causes of ambiguity in the image data (Radojevic and Meijering 2019).

Over the last few years, the rapid advances of deep learning based techniques have made a huge leap in identifying, classification and quantify patterns in optical microscopy image (Shen et al. 2017, Haan et al. 2019, Rivenson and Ozcan 2018). Rather than designing the features of the objects

based on domain-specific knowledge in traditional methods, deep learning exploit hierarchical features representation directly from the data (LeCun et al. 2015). The characteristic of this enables deep learning to achieve state-of-the-art performance in a variety of applications Collobert and Weston (2008), Sutskever et al. (2011), Hinton et al. (2012), Szegedy et al. (2013), Taigman et al. (2014), Zhang et al. (2015). A few of deep learning methods have been applied into neuron tracing, either as pre-processing for image segmentation Li et al. (2017), Li and Shen (2020) or a post-processing for tracing refinement (Zhou et al. 2018a), achieving enhanced performances. However, an effective deep learning requires large amount of training datasets, with quality annotation. In the case of neuron tracing, it remains as a main issue due to the limit amount of available annotated data. This greatly obstacles a further development and improvement of automatic neuron tracing algorithms.

Particularly, all the existing tracing methods, whether involving deep learning or not, are proposed for some specific issues e.g. weak signal (Peng et al. 2011a, Li et al. 2019), noisy background (Peng et al. 2011a, Huang et al. 2020). However, there lacks a quantitative analysis on how these issues effect the neuron tracing and the frequency of occurrences. Therefore, it still remains as an open problem to identify the most critical obstacle towards a better tracing performance, which limits any further algorithm development in a more targeted way.

In this chapter, we propose a pipeline for data generation of complete neuronal morphology, including procedures as automatic tracing, manual refinement, and a novel automatic reconstruction refinement, as illustrated in Fig 3.1. Following the pipeline, we generate over 1000 standard single neuronal morphology data from whole mouse brain image, for analysis and model training. Based on the image obtained, we applied several neuron tracing method and compare the results with the newly generated dataset to detect the consensus false structure and generate samples of false tracing. We systemically inspect the false samples and categorise the failure types, for later algorithm development and testing purposes.

In this regard, we claim our contributions as following:

1. We propose a pipeline for reconstruction of complete neuronal morphology. Following the pipeline, we generate more than 1000 robust reconstructions for analysis and algorithm development.

2. We propose a novel approach for data refinement to produce more accurate and smooth reconstruction based on a 2-time optimal path finding algorithm and deforming step.

3. To our best knowledge, it is the first attempt to systemically classify false types and attributions in neuron tracing.

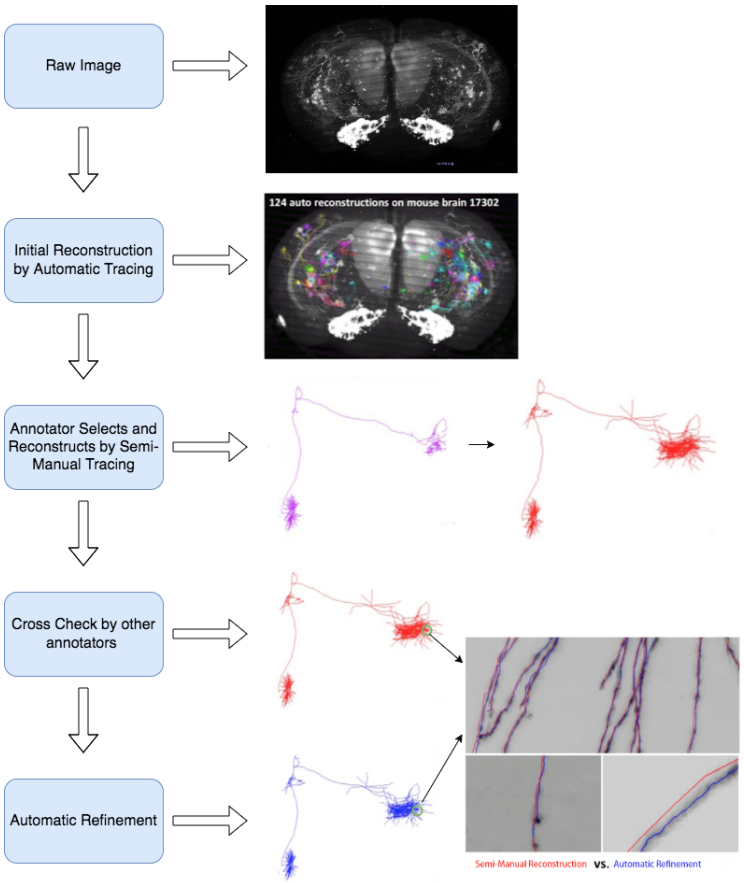


Figure 3.1: Our proposed pipeline for data generation.

## 3.2 Data Generation

### 3.2.1 Full Neuronal Morphology Data Generation Pipeline

As illustrated in Fig 3.1, we establish a pipeline for complete neuronal morphology reconstruction with high quality, which consists of four stages:

1. **Initial neuron reconstruction**: in this stage, we run an existing auto tracer on the raw whole mouse brain image to generate initial reconstructions. Specifically in our case, the raw images data are obtained from BICCN database (web portal:https://biccn.org/). Neurons in a whole mouse brain are sparsely labelled and imaged by the fluorescence micro-optical sectioning tomography (fMOST) system (Gong et al. 2016) with physical resolution at $0.35\mu$m $\times$ $0.35\mu$m $\times$ $1\mu$m. In each sample brain image, there are approximately 2000 neurons to be selected for tracing. The image is converted to TeraFly format (Bria et al. 2016) in a hierarchical structure for multi-resolution visualisation and manipulation in vaa3d (Peng et al. 2010b). With the help of the vaa3d-TeraFly (Peng et al. 2014c) tool mentioned in Section 2.2.2, hundreds of soma locations are pin-pointed as the roots to start automatic tracing. The auto tracing is conducted by Ultratracer+APP2 (Xiao and Peng 2013, Peng et al. 2017).

2. **Semi-manual reconstruction**: in this stage, annotators refine the initial reconstructions by semi-manual tracing. Annotators first observe and select a few of meaningful initial reconstructions, which means they only require some small amount of human labour to refine. Then annotators conduct the semi-manual tracing with vaa3d-Terafly tool (Peng et al. 2014c) to obtain a complete neuronal reconstruction.

3. **Cross check**: In this stage, following the spirit of DIADEM (Gillette et al. 2011)mentioned in Section 2.4.1, to eliminate the mistakes by human operations, we set up a cross check process to control the quality of the reconstruction. After semi-manual reconstruction, the result is sent to two independent annotators to check and correct. Any edit will

be rolled back to the original annotators for confirmation. Only when all of the annotators agree with the reconstruction, it will be sent to next stage.

4. **Automatic Refinement**: in this stage, the reconstruction is processed by an automatic method to produce a final result where the trace lies on the centre of the neuronal signal. The details of the method are discussed in Section 3.2.2.

Following the workflow of the proposed pipeline, we reconstruct 1700 complete single neuronal morphology data for further analysis and algorithm development. Fig 3.2 presents a few of the reconstructions along in the mouse brain data. The dataset are contributed to the SEU-ALLEN Joint Center database (https://braintell.org/projects/fullmorpho/) and NeuroMorpho (https://neuromorpho.org/).

### 3.2.2 Automatic Data refinement

According to the piratical inspection, a drawback of the vaa3d-Terafly (Peng et al. 2014c) is that at some areas, there is a offset of the trace to the signal centre, as seen in Fig 3.1. Although it does not change the path of the neurite or the whole structure, it does effect any downstream usage such as estimating node radius along the neurite or serving as training samples for learning models. To address the issue, we propose a automatic refinement method based on a 2-time optimal path finding and deforming algorithm.

A simple illustration of our method is shown in Fig 3.3. For a given neuronal trace after semi-manual reconstruction,

1. we fist split it into branch segments according to the tree hierarchy, maintaining its topology.

2. For each segment, we evenly distribute a series of anchor points along the reconstruction with equal space between.

3. Then we implement an optimal path finding algorithm between two consecutive anchor points and deform the curve for optimisation. The

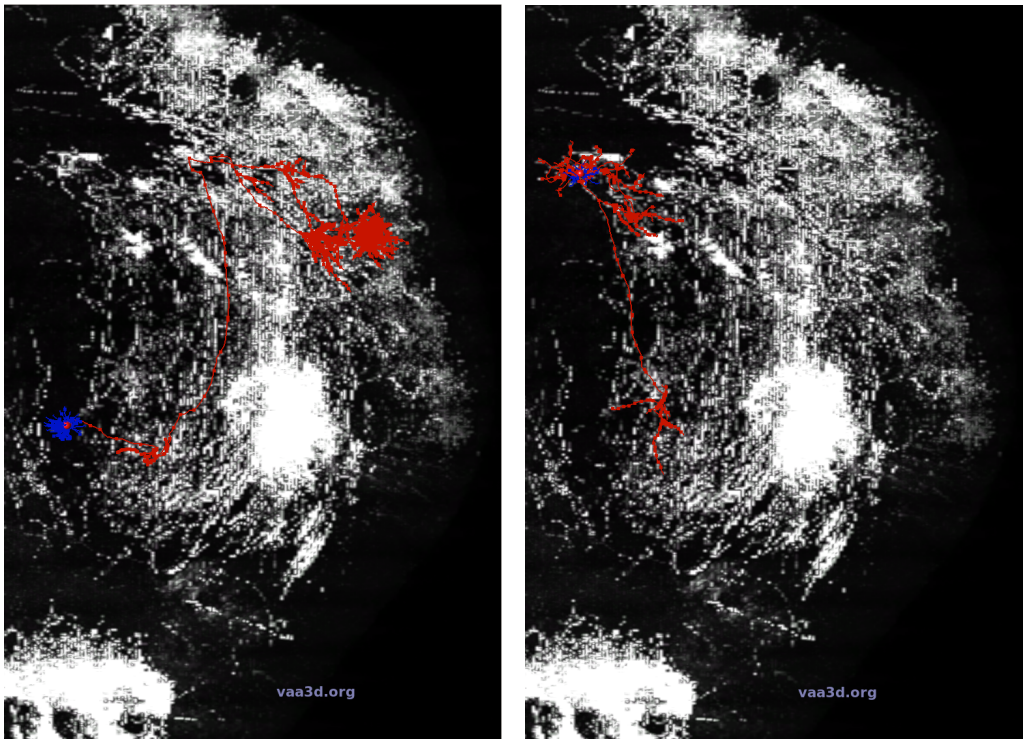


Figure 3.2: two complete neuronal morphology generated by the whole pipeline of the proposed pipeline, projected on 2D. The bright field: the image signals. Blue curves: dendritic branches. Red curves :axon long projection and termini. Red sphere: soma location.

red dots and line illustrate the anchor points and the segment after 1st refinement. The refinement method will be introduced later in details.

4. Next, we select a new set of anchor points at the centre location between the previous set of anchor points (yellow dots) to conduct the second run of the refinement, with same method to produce a final result (yellow line).

5. After the 2-time refinement, the offset line is readdressed to the centre of the image signal.

The refinement method consists of two steps: path finding and deforming. Path finding step is based on graph construction and shortest path finding algorithm. Similar to the approaches in semi-automatic neuron tracing (Peng

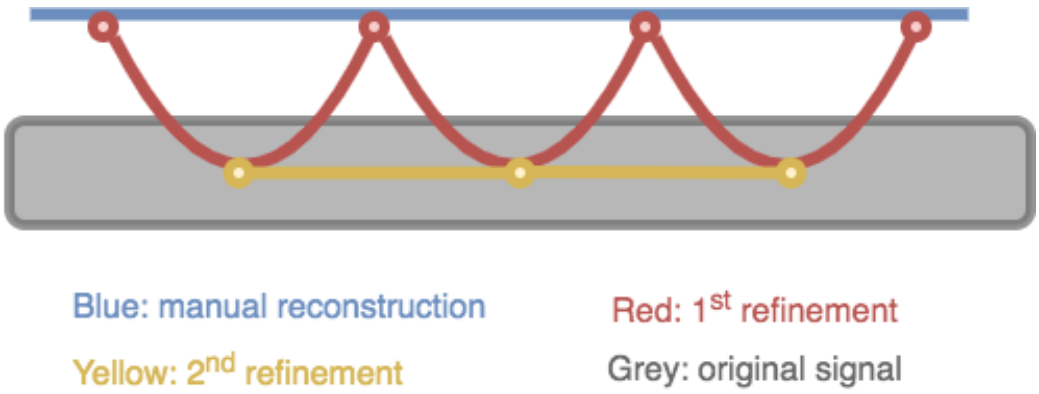


Figure 3.3: A simple illustration of the 2-time refinement approach. The blue line is the manual traced segment which is offset to the grey signal, after the refinement step, it lies in the centre of the signal as the yellow line represents.

et al. 2010a, Myatt et al. 2012), we construct a undirected graph $G = (V, E)$ from the image voxels, where $V$ is the set of vertexes standing for an image voxel and $E$ is the edge between two vertexes only if they are immediate spatial neighbours in image voxel grid. The edge weight is designed as a geodesic function, combining the euclidean distance with image voxel intensity as:

$$e\left(v_0, v_1\right) = \left\|v_0 - v_1\right\| \cdot \left(\frac{g_I\left(v_0\right) + g_I\left(v_1\right)}{2}\right) \tag{3.1}$$

where $v_0, v_1$ denote two adjacent voxels at location $v_0 = (x_0, y_0, z_0)$ and $v_1 = (x_1, y_1, z_1)$ which have an edge between them. $g_i(.)$ is defined as:

$$g_I = \exp\left(\lambda_I \left(1 - I(p)/I_{\max}\right)^2\right) \tag{3.2}$$

in which $I(p)$ denotes the intensity at location $p$ and $I_{\max}$ denotes the maximum intensity among the whole image.

After the graph construction, we implement the Dijkstra algorithm (DIJKSTRA 1959) to find the shortest path.

Then the deforming step is conducted for a more accurate and smooth result. We deform the curve with energy minimisation, to reach a local optimisation for better smoothness. Using the same energy function design

as (Peng et al. 2010a), we define the the overall energy function for the curve as:

$$E_{GD} = \sum_{k=1}^{K-1} \exp\left(E_I(k) + E_C(k)\right) \|C_k - C_{k+1}\| \tag{3.3}$$

where $\{C_k, k = 1, \ldots, K\}$ are a series of control points evenly spaced to represent the continuous curve $C$. And $E(.)$ each represents a energy term as following:

$$E_I(k) = \lambda_I \left(1 - \frac{I\left(C_k\right)}{\max I\left[\Theta\left(C_k, r\right)\right]}\right)^2 \tag{3.4}$$

where $\lambda_I$ is a weight factor set to 10. $I(C_k)$ is the intensity at the control point $C_k$ and $\max I\left[\Theta\left(C_k, r\right)\right]$ is the local maximum intensity around $C_k$. The other energy term represents the metric of closeness to local centres as:

$$E_C(k) = \frac{\lambda_C \sum_{q \in \Theta(C_k, r)} \|C_k - q\|^2 I(q)}{\sum_{q \in \Theta(C_k, r)} I(q)} \tag{3.5}$$

where $\lambda_C$ is a weight factor set to 1. $q$ is the neighbouring voxels within the radius $r$ around control points $C_k$.

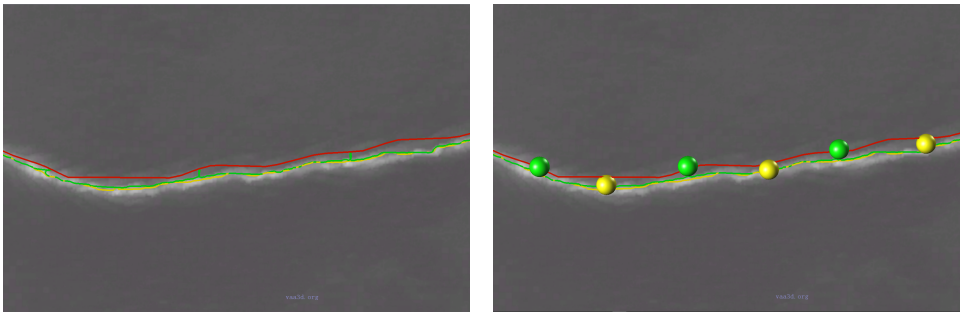

Figure 3.4: A refined neurite in a local area. Red line: Semi-manual reconstruction. green sphere :anchor points selected in the 1st run of the refinement, the point is sized up for best view. Green line : trace after 1st run of refinement. Yellow sphere : anchor points selected in the 2nd run of the refinement, the point is sized up for best view. Yellow line : trace after 2nd run of refinement.

The energy function is minimised via gradient descent in each iteration to update the control points. Thus we obtain a final curve which is closer

to the signal centre and more smooth. A qualitative comparison is shown in Fig 3.4 in practical application.

To evaluate the refinement, we compare the refined reconstructions with their corresponding semi-manual reconstructions. We calculate the average image intensity change of the voxels at the nodes' positions before and after refinement. We also detect nodes with position changed over 2 voxels, since distance less than 2 voxels is visually indistinguishable. These nodes are defined as "different structure" and their portion and intensity change are also calculated. We evaluate 200 reconstructions and the result is given in Table 3.1. According to the result, reconstruction after refinement has been shifted closer to the signal, as the distinct increase of intensity indicates.

| Entire Intensity Change Average | Different Structure Portion | Different Stucture Intensity Change Average |
|---|---|---|
| +4.5472 | 0.1197 | +19.4925 |

Table 3.1: The evaluation of the refinement.

## 3.3 Data Analysis

### 3.3.1 Automatic False Samples Detection and Collection

Since we have generated sufficient amount of data for our algorithm development, there is another problem to further clarify, which is what is our main goal to achieve, or what is the most severe obstacle to overcome towards a novel tracing method. As mentioned before, several tracing methods are proposed for addressing specific issues but the frequency of the issue occurrence and the extent of improvement after solving the problem is still unknown. In this case, we apply some widely-adopted neuron tracing methods i.e. Neutube (Feng et al. 2015), SmartTracing (Chen et al. 2015), APP2 (Xiao and Peng 2013), on the image data obtained in Section 3.2.1, and compare the tracing results with the standard neuronal morphology data, here "standard"

means the quality of the data is ensured by the whole workflow involving manual inspection and cross check. To detect the false structure and collection false samples for further inspection and analysis, we design a method in following procedures as illustrated in Fig 3.5:

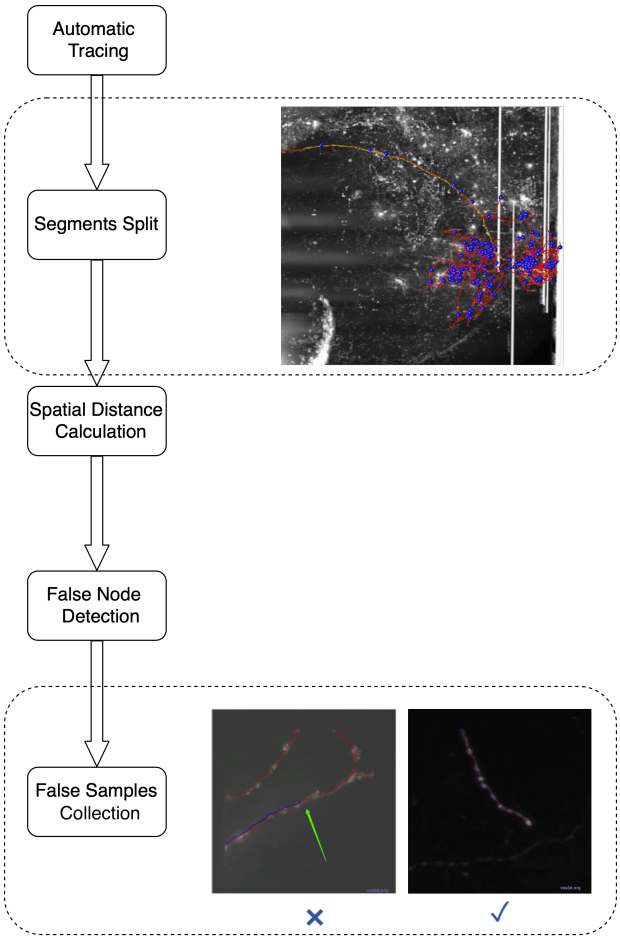


Figure 3.5: The workflow of proposed false sample detection and collection. Top image: reconstructions split into segment with the breaking points; red and yellow lines are standard reconstruction and auto reconstruction respectively; blue spheres represent the breaking points of the branch segments. Bottom box: segments compared with standard reconstruction to detect the false nodes; red lines are standard reconstruction and blue lines are auto reconstruction; a false case and a correct case are shown from left to right.

1. **Automatic tracing**: we run Neutube (Feng et al. 2015), SmartTrac-

ing (Chen et al. 2015), APP2 (Xiao and Peng 2013) respectively on the input whole mouse brain image, in the framework of Ultratracer (Peng et al. 2017). Since the user-input parameters of these methods influence the tracing performance, we set all parameters as default in vaa3d platform (Peng et al. 2010b) to obtain a general result.

2. **Segments split**: After tracing, a neuronal tree is obtained. We split the tree into branch segments following its hierarchy, without disturbing the topology. If a segment is too long, it will be split into shorter segments less than a pre-set length. Same method is applied to the standard reconstruction.

3. **Spatial distance calculation**: using the metric mentioned in section 2.5, we calculate the spatial distance between two corresponding segments from auto tracer and that from standard reconstruction. Following a top-down manner, the comparison starts from segments with highest level in the parent and child hierarchy, to the termini branch.

4. **False node detection**: segment with spatial distance to the standard segment over 5 voxels is determined as false. If a segment is false, the child segments of it will no longer be compared and the false segment is recorded. Next we calculate the spatial distance of each node on the false segment to the nearest node on the standard segment, the first node with spatial distance over 5 voxels to the standard segment is recorded.

5. **false samples collection**: After detecting the false node, we crop a cubic image block with size 128× 128 ×128, centring the node. The corresponding reconstruction in the image block is also cropped. The image/reconstruction dataset is output for inspection and analysis in the next stage.

Following the workflow of the false sample detection and collection, we obtain over 10,000 false samples for analysis.

### 3.3.2 False Types Classification

Based on the the collected false samples, we manually inspect and classify the false types. Since no systematic analysis has been done before in the research field, we classify the false samples due to their appearance and causes empirically. The classifications and their details are described as following:

1. **False Positive type**: false positive types are over-reconstructed neuronal traces with redundant branches compared with the real morphology. As shown in Fig 3.6, we analyse and conclude the possible causes of this kind of fail as:

   (a) Close but irrelevant signal: during tracing procedure, the tracer falsely identify a close signal from other neurons as relevant and construct a bifurcation and redundant branch, which is a recurrent problem in neuron tracing.

   (b) Noise: the tracer falsely takes the noise in the background as neuron signal and therefore forms a redundant structure. It occurs frequently in areas with high noise/signal ratio.

   (c) Boundary effect: an artificial branch is generated due to the Ultratracer (Peng et al. 2017) framework. The Ultratracer framework processes the tracing method in a series of 3D image tiles, of which the trace sometimes show an offset on the cuboid faces. When starting the tracing procedure in the next consecutive tile, the offset node acting as starting point will cause a redundant structure.

2. **False Negative type**: false negative types are under-reconstructed neuronal traces with missing branch. As shown in Fig 3.7, we analyse and conclude the causes as:

   (a) Weak signal: if the intensity along a neurite signal is too low, the tracer is prone to regard it as background noise and not following the path, resulting in a missing branch. It is a recurrent problem in neuron tracing.

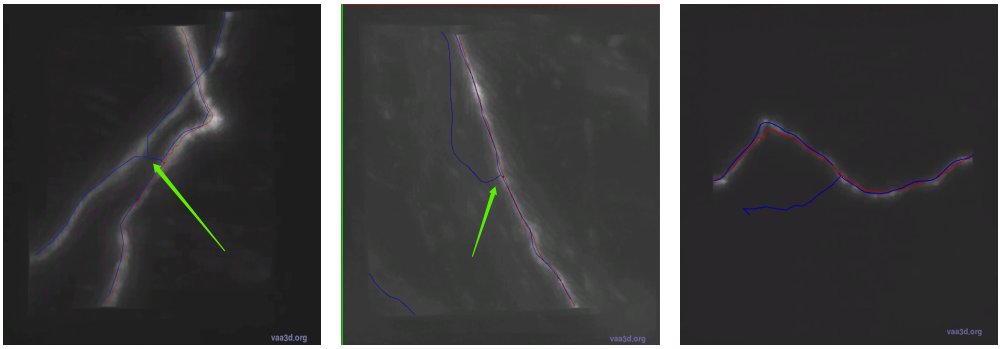

Figure 3.6: False positive (FP) type with different causes. From left to right: FP caused by close signal; by background noise; by boundary effect. The red marker is the false node detected. Red lines are standard reconstruction. Blue lines are automatic reconstruction.

(b) Punctuated signal: the tracing stops due to a gap between the signal. It can be seen in areas where the neurites are punctuated due to imperfect imaging.

(c) Boundary effect: as mentioned in the previous part, a node is offset due to the boundary effect by Ultratracer (Peng et al. 2017). If the node, acting as starting point during the tracing procedure in the next image tile, is largely offset to signal, it is not able to start the tracing, resulting in a break of branch and an offset termini.

(d) Other reason: this kind of false negative samples have consistent and bright signal, without showing any offset node, but stop midway. The reason remains unknown and needs further analysis.

3. **False Positive & Negative** : false positive and negative types are traces combine over-reconstruction and under-reconstruction simultaneously. This is mainly caused by close signal, where the tracer accidentally goes along signal of another relevant neuron and misses branches of its own, as illustrated in Fig 3.8.

4. **Offset nodes**: offset nodes are where reconstructed nodes do not perfectly lie on the signal. This kind of false type does not cause severe

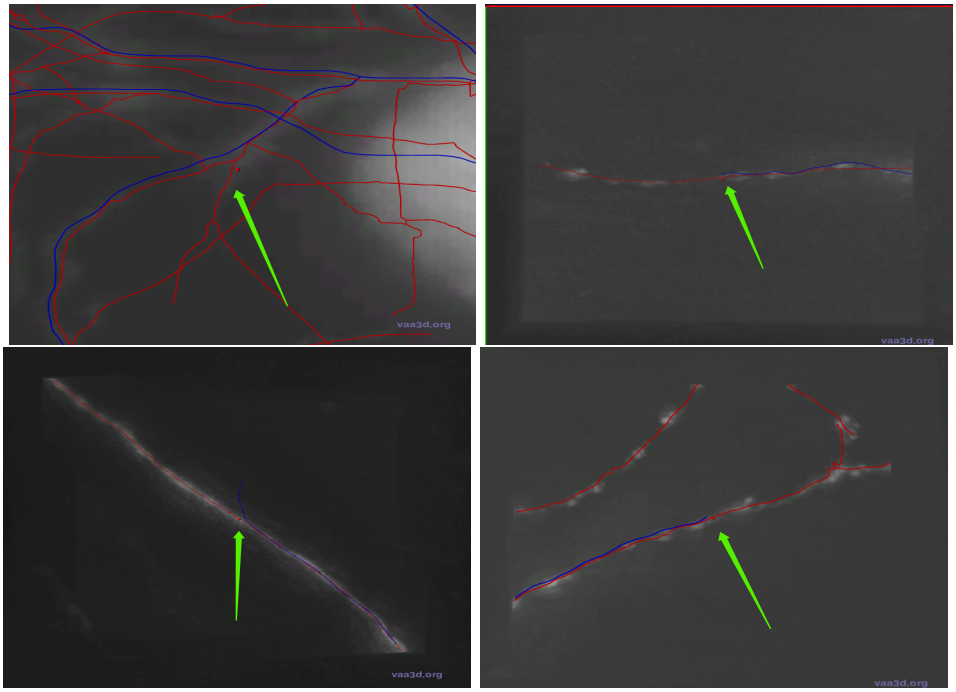

Figure 3.7: False negative (FN) type with different causes. From Top left to bottom right: FN caused by weak signal; by gap; by boundary effect; other FN types. The red marker is the false node detected. Red lines are standard reconstruction. Blue lines are automatic reconstruction.

errors nor changing the overall structure, and can be sufficiently optimised by our refinement approach. As shown in Fig 3.9, they can be concluded as:

(a) Boundary effect: if the aforementioned offset node generated by Ultratracer is not remote to the neuronal signal, the tracing in the next image tile can continue. if there is no overlap between the tracing in two consecutive tile, it will form a peak-shape.

(b) Off-centre nodes by tracer: this kind of fail is due to the tracing algorithm. The tracer fails to find the optimal local maximum position for the node.

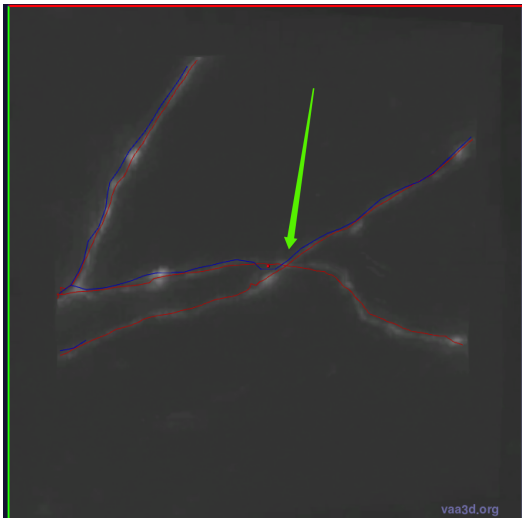


Figure 3.8: False positive & negative false type. The trace goes to another adjacent but irrelevant signal, missing its own branch. Red lines are standard reconstruction. Blue lines are automatic reconstruction.

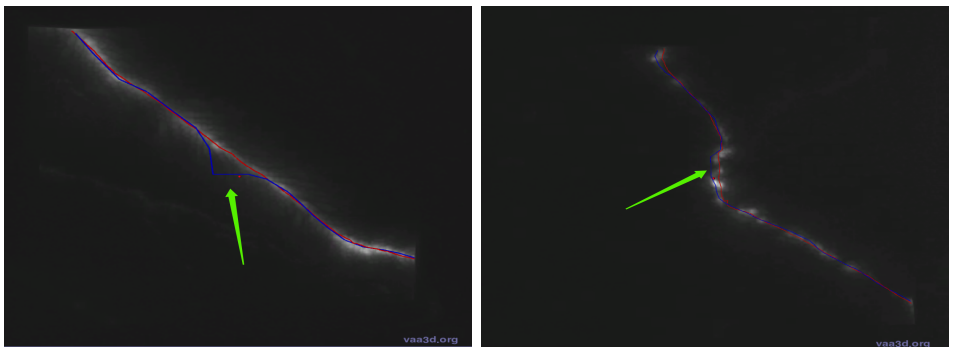


Figure 3.9: Offset nodes false types. From left to right: offset due to boundary effect by ultratracer; off-centre node reconstructed by tracing algorithm. Red lines are standard reconstruction. Blue lines are automatic reconstruction.

### 3.3.3 Quantitative Analysis

We count the frequency of occurrence of aforementioned false types, as the basis of our subsequent algorithm development and research. The result is shown in Table 3.2. Note that the "agreed" means cases where the three tracers fail at the same location.

According to the quantitative result, in terms of frequency of occurrences,

| Type | | Number | % | Neutube | SmartTracing | APP2 | Agreed | Agreed% |
|---|---|---|---|---|---|---|---|---|
| | False Positive | 4991 | 49.5% | 4635 | 4463 | 4487 | 3950 | 79.14% |
| 1.1 | Close irrelevant Signal | 3892 | 38.6% | 3654 | 3693 | 3728 | 3326 | 85.46% |
| 1.2 | Noise | 851 | 8.4% | 810 | 601 | 589 | 456 | 53.58% |
| 1.3 | Boundary effect | 173 | 1.7% | 171 | 169 | 170 | 168 | 97.11% |
| | False Negative | 1898 | 18.8% | 1456 | 1440 | 1619 | 1296 | 68.28% |
| 2.1 | Weak signal | 675 | 6.7% | 593 | 582 | 624 | 557 | 82.52% |
| 2.2 | Gap | 548 | 5.4% | 525 | 534 | 522 | 513 | 93.61% |
| 2.3 | Boundary Effect | 203 | 2.0% | 201 | 198 | 199 | 197 | 97.04% |
| 2.4 | Others | 472 | 4.7% | 137 | 126 | 274 | 29 | 6.14% |
| | False Positive & Negative | 967 | 9.6% | 926 | 937 | 941 | 912 | 94.31% |
| 3.1 | Close irrelevant signal | 967 | 9.6% | 926 | 937 | 941 | 912 | 94.31% |
| | Offset nodes | 2307 | 22.9% | 1473 | 1593 | 1575 | 1025 | 44.43% |
| 4.1 | Boudary effect | 688 | 6.8% | 681 | 679 | 678 | 674 | 97.97% |
| 4.2 | off centre by tracer | 1619 | 16.0% | 792 | 914 | 897 | 351 | 21.68% |
| | Sum | 10088 | | | | | 7183 | 71.20% |

Table 3.2: Quantity of false samples classification

the main obstacle toward a more accurate tracing method is the false positive trace caused by close but irrelevant signal, making up approximately 40% of the false cases, and three different tracers are all prone to error in that case. However, the potential harm of a false type should also be taken into account. A false positive reconstruction with redundant branches can be easily revised by a single delete operation by an annotator, as simple as a mouse click. However, for a false negative reconstruction, a missed branch at a high level of the neuronal tree is possible to cause a loss of all its child segments, resulting in a highly incomplete structure and massive amount of human labour to refine. A even worse situation is the false negative & positive false reconstruction, where the trace goes along another irrelevant signal at some point and continues to grow afterwards, which is very hard for an annotator to detect and takes huge amount of time to revise. And the high agreement percentage also indicates it is a general problem for these three auto tracers. Therefore, despite its relatively lower likelihood to occur, it should be regarded as one major obstacle to overcome in our tracing algorithm development. By contrast, offset nodes, although are more likely to appear as shown by the quantitative result, do not change the overall structure and

can be easily improved by our proposed refinement method mentioned in the previous section, so it will not be regarded as a main problem to address.

Overall, taking into account of the frequency of occurrences, the severity and the generality of the problem, we regard the main obstacles to tackle are the false positive and false positive & negative reconstruction caused by close irrelevant signal, and false negative tracing caused by weak signal and punctuated signal.

## 3.4 Summary

In this chapter, we propose a reconstruction pipeline to generate complete single neuronal morphology data, which includes initial reconstruction, semi-manual reconstruction and a novel automatic refinement approach. The novel automatic refinement approach is based on a 2-time optimal path finding and deforming algorithm. After the automatic refinement, we obtain more accurate reconstruction aligning the neuronal signal in the optical microscopy image. Following the pipeline, we generate over 2000 complete neuronal morphology data for analysis and algorithm development.

Based on the complete neuronal morphology data, we propose a workflow to detect and collect false samples generated by auto tracing. The workflow includes five stages as automatic tracing, segments split, spatial distance calculation, false node detection and false samples collection. We manually inspect 10,088 of these samples and classify them into false types with a quantitative analysis. To our best knowledge, It is the first attempt in the research field to do a quantitative failure analysis on automatic neuron tracing. Taking into account the frequency of occurrence and potential harm of the false cases, false positive & negative cases by close but irrelevant signal, and false negative caused by weak or punctuated signal are selected as our main issues to tackle in later algorithm development stage. A future work is to propose a novel model enabling to capture the weak or punctuated neurite signal whilst distinguish close but irrelevant signal.

# Chapter 4

# Automatic Neuron Tracing via Occupancy Network and Curve Skeleton Extraction

## 4.1 Introduction

The substantial advances in microscopic techniques have enabled the generation of massive amount of optical microscopic data, helping the analysis of neuronal structure and function (Peng et al. 2015). Among the analytic workflow, digitizing the 3D morphology of single neurons from optical images is essential, as it determines the functional role of a single neuron in a neural circuit (Peng et al. 2020a). However, the reconstruction of 3D morphology remains as a critical challenge, largely relying on human labour for annotation, which is time-consuming and prone to error. Therefore, it requires the development of automatic neuron tracing i.e. neuron reconstruction methods to standardize and accelerate the process (Acciai et al. 2016).

Numerous automatic methods have been developed to yield more efficient neuron tracing targeting a variety of issues, such as noisy background (Peng et al. 2011a), weak signal (Li et al. 2019). After a quantitative analysis in Chapter 3, we are acknowledged that the main issues in automatic neuron tracing are close but irrelevant branch signals and weak or punctuated neurite signals. It can be intuitively improved by a more accurate image segmentation. Most of automatic tracing methods involve a 3D fore-

ground/background segmentation during their tracing procedure, either using a threshold or a neural network to classify the foreground. But it can be challenging due to the severe data imbalance of the foreground/background voxels in 3D optical neuronal images. Typically, the positive signals only account for approximately 0.1% of the image. In such case, the weak neurite signal is prone to be classified as background, while if the threshold is set low to capture these weak signals, close but irrelevant branch signals are fused as foreground without discrimination. Also, the variety of imaging condition results in uneven background and punctuated segments, making it even more difficult to follow.

In this chapter, we propose a novel approach to trace neurons in 3D optical microscopic images, by occupancy learning (Mescheder et al. 2019) and curve skeleton extraction (Tagliasacchi et al. 2009). To train and test the occupancy network, we build a dataset with optical microscopic images with manually annotated foreground labels and reconstructions (SWC format) per image. A 3D implicit occupancy function is then learned to represent the continuous shape of a neuron, and to calculate the occupancy probability, which clarifies if a spatial voxel at any position is within 3D bounding volume of a neuron or not. With a strategy of equal input points sampling, it tackles the extreme foreground/background data imbalance of neuronal images, and efficiently captures weak signal. Then a curve skeleton extraction based on generalized rotational symmetry axis (ROSA) is applied for tracing. It exploits the neuron orientation and further refines on any incomplete area to produce robust tracing results. The experimental results show that, compared with existing tracing methods, our method shows superior performance in terms of IoU accuracy and neuron distance score.

In summary, our contributions are as follows:

- We jointly embed occupancy learning with skeleton extraction to reconstruct neurons from 3D optical microscopy images. To our best knowledge, it is the first attempt to address the sparseness in neuron reconstruction via predicting the occupancy probability.

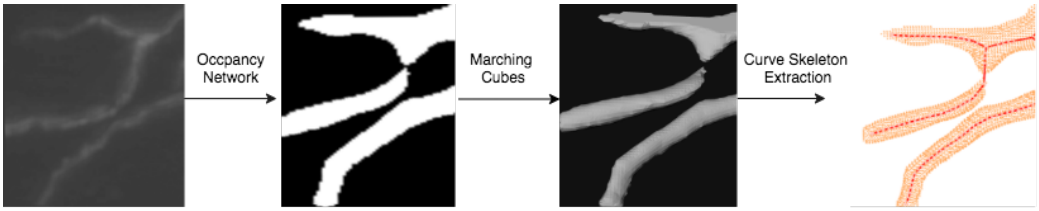


Figure 4.1: Workflow of our tracing method. From left to right, given a 3D raw image, an occupancy map is given by a learned occupancy network. Next, based on the occupancy map, an isosurface is reconstructed through Marching Cubes. Finally, neuron is traced by Curve Skeleton Extraction via ROSA.

- We address the data imbalance issue by leveraging occupancy learning with an equal point sampling strategy to control the data distribution in the input. It tackles the root causes of the imbalance problem, and captures weak signal efficiently.
- We experimentally evaluate the performance of our approach. It produces superior results compared to existing methods both qualitatively and quantitatively.

## 4.2 Methodology

Our approach mainly consists of three stages as Fig 4.1 illustrates.

1. The input image is voxelized and every point in voxel grids is evaluated by the occupancy network.
2. Every voxel is assigned with an occupancy. Marching Cubes algorithm is then employed to construct the isosurface mesh.
3. Curve skeleton extraction via ROSA algorithms is exploited for tracing.

### 4.2.1 Data Collection and Preparation

We build a dataset of raw 3D optical microscopic images with manually annotated foreground labels and reconstructions for training and testing. The raw images used are cut from high-resolution whole mouse brain optical images from sparsely labeled brains using fluorescence micro-optical sectioning tomography (fMOST)(Gong et al. 2016). The physical resolution of the image is $0.35\mu m \times 0.35\mu m \times 1\ \mu m$. which are open-source dataset available at BICCNs Brain Image Library (BIL) (available at https://www.brainimagelibrary.org). The neuron reconstruction in SWC format are contributed by SEU-ALLEN Joint Center, Institute for Brain and Intelligence (https://braintell.org/projects/fullmorpho), following our pipeline of reconstruction generation as in Chapter 3. The foreground label of images are derived from the reconstructions. The annotation/labelling, reconstruction and visualization are realized through Vaa3D (Peng et al. 2014c) platform along with the TeraFly (Bria et al. 2016) and TeraVR (Wang et al. 2019a) The software package is available through https://github.com/Vaa3D.

The raw image and its reconstruction are uniformly sampled with a size of $64^3$. We generate 50,000 samples. The train/test split is set to 42,000/8,000.

### 4.2.2 Occupancy Network

Existing neuron tracing methods segment foreground neurons either using a threshold or a classifier. In our study, the neuronal image has extreme class imbalance problems, which usually undermines the performance of existing methods. To this end, we employ Occupancy Network (OccNet) instead.

An occupancy network is described by the learnable mapping as:

$$f_\theta : R^3 \times \chi \rightarrow [0, 1] \tag{4.1}$$

The mapping is parametrised by $\theta$. It takes a 3D location $p \in R^3$ and an observation, $x \in \chi$ as the input. And it outputs the occupancy probability $o \in [0, 1]$ that describes whether the input location lies within or outside a continuous decision boundary. In our case, The observation $x$ is a 3D volume that is encoded by a 3D CNN to a global latent representation $x_l$. Then the occupancy determines if a location is within our neuron structure or not.

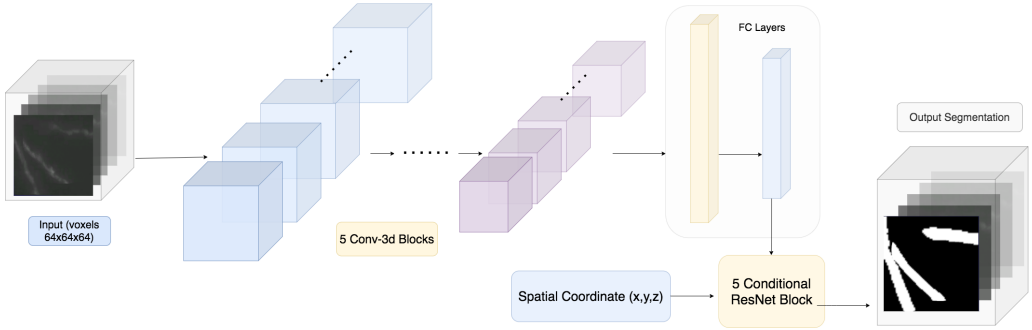


Figure 4.2: Network Structure of Occupancy Network.

**Network Structure.** As Fig 4.2 shows, the OccNet exploits 5× 3D convolutional blocks as encoder. ReLu is adopted as activation function following each block, which is connected with a multilayer perceptron (MLP) to produce a conditional feature. The conditional feature along with the spatial coordinates (x,y,z) are fed into 5× ResNet blocks with conditional batch normalization, to obtain the binary occupancy values (0 or 1).

**Sampling and Training.** To learn the parameters of the OccNet, we equally sample points inside and outside the neurite during training. This ensures an equal occurrence of classes and eliminates the class imbalance problem encountered in neuronal images. For the $i$-th sample in a training batch we sample $K$ points $p_{ij} \in R^3, j = 1, ..., K$ and evaluate the mini-batch loss at those sample points:

$$L_\beta(\theta) = \frac{1}{|\beta|} \sum_{i=1}^{|\beta|} \sum_{j=1}^{K} L(f_\theta(p_{ij}, x_i), o_{ij}), \tag{4.2}$$

where $x_i$ denotes the $i$'th observation of batch $\beta$, $o_{ij}$ is the ground-truth occupancy at point $p_{ij}$. $L(\cdot, \cdot)$ is a cross-entropy loss.

**Inference.** Given a raw image, we evaluate the OccNet for all voxels $\{p\}$. Then we exploit the Marching Cubes (Lorensen and Cline 1987) algorithm to extract an approximate isosurface, as illustrated in Fig 4.1:

$$\left\{p \in R^3 \mid f_\theta(p, x) = \tau\right\} \tag{4.3}$$

### 4.2.3 Curve Skeleton Extraction

A common strategy to model a neuron is to use a curved skeleton, or 'backbone' with the estimated width along the skeleton. To obtain such a skeleton, a typical routine is: 1) segment the image into foreground/background. 2) construct a undirected graph on the image foreground. Graph vertexes are image voxels, and edge weight are designed according to context geometric and intensity information. 3) Finding the shortest path from the root to the end location, and obtain the skeleton (Peng et al. 2010a). But the performance largely relies on the segmentation results and parameters chosen when defining the edge function during the graph construction. The tracing are prone to terminate if the neurite is imaged as punctuated and broken, resulting in incomplete reconstruction. To overcome these disadvantages, we exploit a curve skeleton extraction method via generalized *rotational symmetry axis* (ROSA), following our surface mesh reconstruction in the previous step.

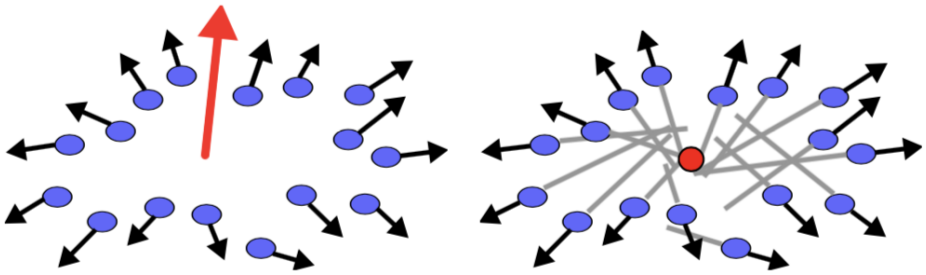

Figure 4.3: ROSA point definition in 3D: point that minimizes the sum of angular variance and sum of projected distance to the normal extensions.

The key insight of our method is that a curve skeleton is considered as a generalized rotational *symmetry axis* (ROSA) of a shape. Neurites are generally modelled as cylindrical compartments therefore complement the assumption (Peng et al. 2010a). For a local subset $S$ of orientated vertex in the input mesh, a *ROSA point* is defined as the most rotationally symmetric point about $S$. As shown in Fig 4.3, the ROSA point $p = (xp, vp)$, where

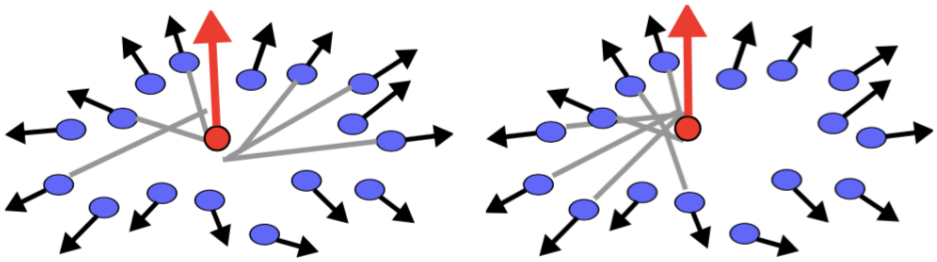

Figure 4.4: Robustness of ROSA points in incomplete structure (blue dots without grey arrows are missing points).

1. $xp$ denotes the position where sum of squared distances to the line extensions of the point normals in $S$ reaches minimum

2. $vp$ denotes the normal that minimizes the variance of the angles between $vp$ and the normals in S.

Fig 4.4 shows that ROSA point is stable with position and orientation when large amount of vertexes are missing.

Based on the ROSA formulation, skeleton extraction derives from finding ROSA points sets of the whole object. It is implemented with an iterative procedure via planar cuts. Each iteration for a given vertex, the algorithm finds the optimal cutting plane which appropriately describes its local rotational symmetry. The ROSA point is computed and all the vertexes on the optimal cutting plane will be anchored. The cuts end until all the points are anchored. For joint area where it lacks symmetry, a post-processing of thinning and re-centering are applied to obtain more meaningful skeletal points. An example in practice of how it further completes the incomplete area and traces the neuron is shown in Fig 4.1.

## 4.3 Evaluation

### 4.3.1 Experiment Settings

**Dataset Description:** As mentioned above, our dataset consists of raw images with size of $64^3$ , their corresponding foreground labels and reconstructions (SWC format). We use different combination of these data to train and test for different purposes:

- The raw images along with their foreground labels are for training and testing the OccNet and other CNN for comparison.
- The raw images along with the reconstruction (SWC format) are for testing the whole tracing approach, compared with other existing tracing methods.

**Implementation and Training Details:** The OccNet model is implemented via Pytorch and trained on a NVDIA GeForce RTX 2080 Ti graphic card. The dataset samples are split into 42000 for training and 8000 for testing. Duringf training, 1024/1024 positive/negative points are equally sampled. We employ Adam Optimizer with learning rate $lr = 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-8}$. The model reaches a stable score after approximately 50 epochs. And the best model with highest evaluation score was recorded for comparison experiments in the next stage.
**Evaluation Metrics:** To evaluate the performance of the whole workflow and OccNet, we adopt two different metrics:

1. The Intersect over Union (IoU) score is adopted to evaluate the outputs of segmentation methods, compared with the ground-truth.
2. The neuron distance metric is exploited when compared the final reconstruction (SWC format) with the correspondingly ground-truth reconstruction. Specifically, three distances scores defined in (Peng et al. 2010a) are calculated as:

(a) The entire structure average: it first calculates the minimal spatial distance between each pair of two nearest nodes from reconstruction and GT, and then averages all these reciprocal minimal spatial distances.

(b) different structure average : it only sums those reciprocal distances that are larger than 2 voxels, since the difference less than 2 voxels is hardly distinguished visually.

(c) percentage of different structure: it calculates the percentage of pairs of nodes whose reciprocal minimal spatial distances are more than 2 voxels.

### 4.3.2 Comparison Experiments

**OccNet Evaluation:** We compare the performance of OccNet with Unet (Ronneberger et al. 2015) and Attention U-net (Oktay et al. 2018) with different design of loss function e.g. Dice Loss (Sorensen 1948), Focal Loss (Lin et al. 2017) and clDice loss (Shit et al. 2021). We train different models on the 42000 split of training dataset and test it on the 8000 split of test dataset. Considering the sufficiently large dataset, the risk of overfitting is greatly reduced. Table 4.1 shows the score. A qualitative result is shown in Fig 4.5, the threshold used for making the binary map is 0.5 . Considering the extreme imbalance of the class in neuronal image (0.1%), the prediction of other network is prone to be false negative, while OccNet significantly improves the prediction accuracy and tackles the problem. The significant improvement compared with CNN-based U-Nets can be ascribed to the natural difference of network structure between occupancy network and CNN-based U-Nets. CNN-based U-Nets consist of an encoder-decoder scheme and the 3D shape is represented discretely (e.g. voxels in our case), while occupancy networks implicitly represent the 3D surface as the continuous decision boundary of a deep neural network classifier. This makes occupancy networks more capable of capturing weak or punctuated signals.

**OccNet+Tracing Evaluation:** Exploiting the same strategy as in (Li et al. 2017) and (Zhao et al. 2020), we processed segmentation through OccNet and

| Model | IoU( 100) |
|---|---|
| U-net | 6.12 |
| U-net with Dice Loss | 6.56 |
| Unet with Focal Loss | 6.04 |
| U-net with clDice Loss | 7.95 |
| Attention U-net | 9.44 |
| Attention U-net with Dice Loss | 10.01 |
| Attention Unet with Focal Loss | 9.46 |
| Attention U-net with clDice Loss | 12.62 |
| **OccNet** | **35.22** |

Table 4.1: OccNet compared with other segmentation Network

Figure 4.5: A qualitative comparison of segmentations. From left to right: raw image, ground-truth segmentation, segmentation via U-net, segmentation via Attention Unet and segmentation via OccNet.

fuse the generated mask with the raw image by equal weight as:

$$F(x) = \alpha\tilde{I}(x) + (1-\alpha)\tilde{P}(x) \tag{4.4}$$

Different tracing methods, APP2 (Xiao and Peng 2013), SmartTracing (Chen et al. 2015) and Neutube (Feng et al. 2015) are run on the fused image to see if the performance improves compared with that merely run on the raw image. To obtain a fair result, since these traditional tracing methods require careful parameter tuning, we use more than 5 set of parameters to find their optimal reconstruction result. Table 4.2. shows the neuron distance score compared with the ground-truth reconstruction. Fig 4.6 shows a qualitative comparison. According to the results, most of the tracing methods have a better performance combined with OccNet.

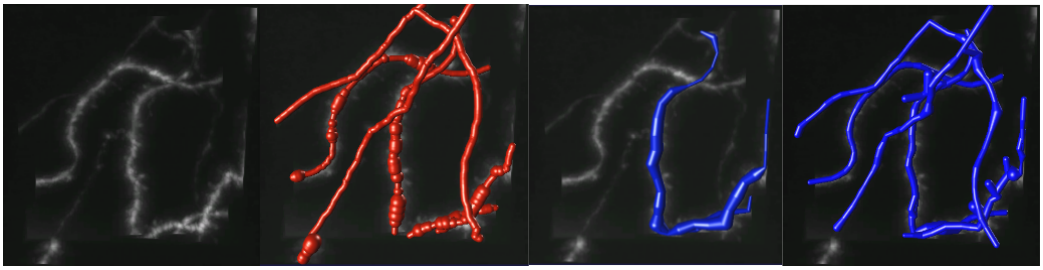

Figure 4.6: A qualitative comparison of OccNet+Tracing methods. From left to right: Raw image, Ground Truth Reconstruction, APP2, OccNet+App2. All the reconstructions are visualized via Vaa3d (Peng et al. 2010b).

| **Tracing methods** | **Entire structure average** | **Different structure average** | **% of different structure** |
|---|---|---|---|
| APP2 | 10.0591 | 18.5326 | 0.5349 |
| OccNet+APP2 | 4.1022 | 4.3812 | 0.2156 |
| SmartTracing | 21.5022 | 24.2611 | 0.6458 |
| OccNet+SmartTracing | 5.6247 | 5.9147 | 0.4586 |
| Neutube | 11.4993 | 12.4576 | 0.4851 |
| OccNet+Neutube | 3.7645 | 3.9515 | 0.1842 |
| **Our method** | **3.4471** | **3.6753** | **0.1883** |

Table 4.2: A comparison of our method with other tracing methods and OccNet+ Methods.

**OccNet+Curve Skeleton Extraction Evaluation:** we generate the skeleton with our whole workflow and compare it with OccNet plus any other tracing methods. The result is shown in Table 4.2 and Fig 4.7 gives a qualitative comparison. According to the previous experiment, OccNet can largely improve the performance of other existing tracers, and our proposed method which embeds OccNet with ROSA further outperforms these enhanced tracing algorithms. Compared with GT, our method is more likely to generate accurate reconstruction in terms of different structure and neuron distance.

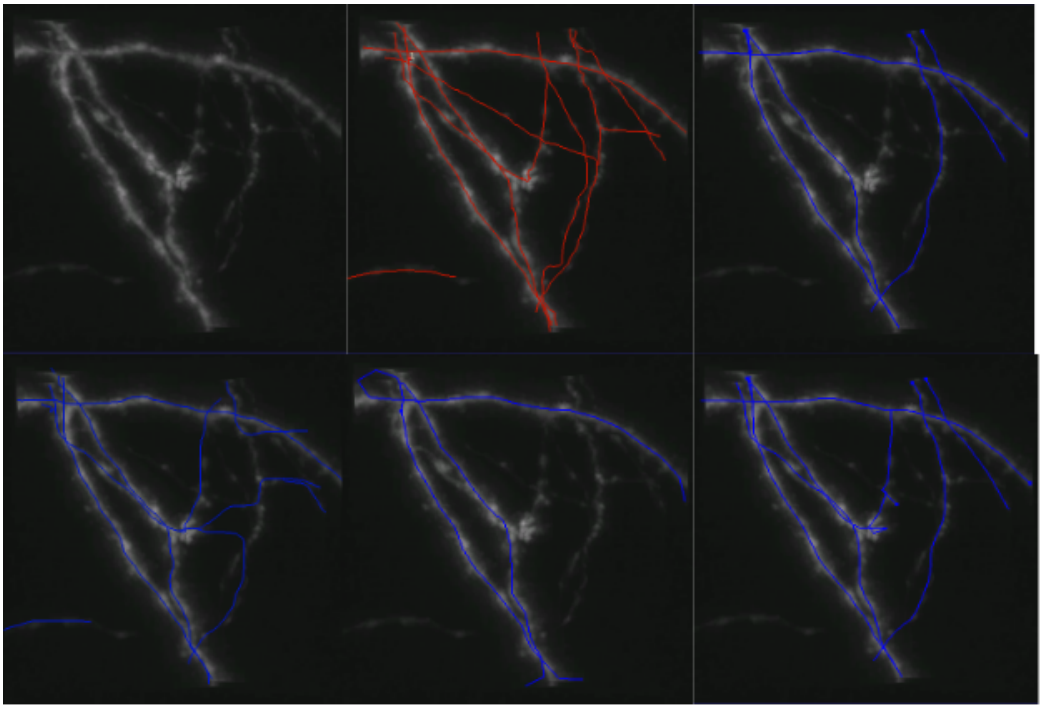

Figure 4.7: A qualitative comparison of our method with other tracing approaches. Note only skeletons are shown for better view. Top left to right: Raw image, Ground Truth, Occ+APP2, Bottom left to right: Occ+ Neutube, OccNet+SmartTracing, Our method.

## 4.4 Discussion

According to the experimental results, OccNet significantly increases the prediction accuracy of foreground in neuronal image, address the disadvantages of high-class imbalance. It can be ascribed to implicit representation of 3D shape by occupancy network. Along with OccNet, any other existing tracing methods achieve greater performance. Results also prove the effectiveness of our method, by surpassing most of the existing tracing methods in terms of neuron distance. However, according to our observation during the experiments, in some cases, some fine geometric details are missed, which requires further improvement.

## 4.5 Summary

In this chapter, we propose a novel approach for 3D neuron tracing. Our target is to tackle major obstacles in neuron tracing, which are weak or punctuated signals and close but irrelevant signals. Our solution is to enhance the image segmentation to capture relevant signals while discriminate irrelevant signals. However, the extreme imbalance of foreground/background classes in neuronal images has greatly limited the process, making any segmentation prone to false negative. Accordingly we propose a deep learning model based on occupancy learning to resolve the problem. Followed by a curve skeleton extraction, the method is able to trace complete neuron structure from highly sparse images. Experiments show that our method surpasses most of existing tracing methods in terms of accuracy and neuron distance score. A direction of future work is to further revise the occupancy network to enable its ability in fine geometry.

# Chapter 5

# Automatic Neuron Tracing via Learning Topology-Preserving Skeleton Extraction

## 5.1 Introduction

Neuron tracing or reconstruction, is a key step for characterising the 3D neuronal morphology, which is essential to understand how brain functions (Acciai et al. 2016). In decades, the tracing has been mainly processed by human labour. However, in recent years, the advancement of optical microscopy imaging technique have enable a burst generation of huge amount of high-resolution image data, with ultra-volume size. To analyse such amount of data at micron level becomes even more time-consuming, tedious and error-prone for manual processing. Therefore, it is highly desired for automation in the research field to accelerate and standardise the tracing process.

In last decade, several semi-automatic or fully-automatic tracing methods have been developed. These methods achieve varying degree of success tackling different specific obstacle, such as noisy background (Peng et al. 2011a), weak signal (Li et al. 2019). However, according to our quantitative analysis in Chapter 3, these methods are prone to fail in areas with irrelevant signal nearby, or where weak signal and punctuated signal occur. These issues limit further improvement on current tracing performance. One reasonable method to address is to enhance the image segmentation, as we proposed in

Chapter 4. Although with a more accurate performance, the method still traces neuron in a multi-stage manner, like most of other existing methods. In such multi-stage approach, the whole performance highly relies on the image segmentation result and the tracing procedure alone cannot be improved during training. Error in the segmentation will be accumulated in the later tracing process, while error in the tracing cannot be back-propagated to the previous stage for optimisation. Therefore, learning neuron tracing directly from raw image is a potential solution to further tackle the obstacles and improve the performance.

In recent years, skeleton extraction has achieved great breakthrough with the application of deep learning techniques. Several deep learning based methods are proposed for skeleton extraction from a variety of data input, such as RGB image (Shen et al. 2016, Ke et al. 2017, Liu et al. 2018, Wang et al. 2019c, Zhao et al. 2018, Xiao et al. 2019, Tang et al. 2021), binary image (Wang and Liu 2018, Nathan and Kansal 2019) , and point cloud (Qin et al. 2020, Chaudhury and Godin 2020, Nie et al. 2020). The success of these application has motivated us to extend its use to neuron tracing from 3D image voxels, to leverage the great modeling capacities of deep learning.

In this chapter, we propose an deep learning approach to trace neuron directly from an input 3D volumetric image data. We propose an encoder-decoder convolutional neural network (CNN), with deforming line primitives to form skeleton. With a designed regularised loss function, the topology of skeleton is well-preserved, overcoming the critical obstacle of false tracing to a nearby but relevant signal. We evaluate the algorithm using our generated dataset mentioned in Chapter 4 and a public dataset i.e. BigNeuron by comparing it with existing methods as well as our proposed method in Chapter 4. The experimental results show it outperforms other approaches and achieve state-of-the-art performance.

In summary, we claim our contribution as following:

- We propose a novel deep learning model for neuron tracing. To our best knowledge, it is the first attempt to learn skeleton extraction directly from a raw 3D volumetric image.

- We design a set of deforming line primitives and a regularised loss function to preserve topology, overcoming the critical issue of false tracing caused by close but irrelevant signal.
- We conduct thorough experiments to compare our approach with counterpart methods on different datasets. The experimental results manifest its robustness by achieving state-of-the-art performance.

## 5.2 Methodology

### 5.2.1 Data Collection and Preparation

We use the datasets generated and curated in Chapter 4.2.1 to train and validate our proposed learning model. Since the model does not involve image segmentation and output skeleton directly, the reconstructions are maintained in SWC format to leverage the natural presentation of skeletal nodes.

To further evaluate our method in a more general case, we use the public BigNeuron (Peng et al. 2020a) dataset for training and testing. The dataset has 166 gold standard reconstruction with corresponding optical microscopy image stacks, from a variety of neuron types, brain regions and species, with image size range from $256\times256\times64$ to $2000\times2000\times1000$. We uniformly sampled the image and reconstruction in the same way as in Chapter 4.2.1 to generate image/trace pair with same block size $64^3$.Since the images are highly sparse, only 3422 image blocks with reconstruction are obtained. After image augmentation, 17110 samples are generated.

### 5.2.2 Learning Topology-Preserving Skeleton Extraction

Given an input volumetric image $I$, our proposed network is trained to extract the skeletons directly. Since the skeletons in neuron tracing is fundamentally represented as undirected tree, learning such tree from image $I$ is computationally complex and costly. However, we are acknowledged that the skeleton of neurite satisfies geometric constraints to form a regular shape.

Therefore, we transform the problem into learning a set of pre-connected skeletal points from a series of input line primitives, thus making the skeleton extraction a standard point set regression problem.

**Overall Network Structure** The structure of our proposed deep neural network for neuron tracing is illustrated in Fig 5.1. An input volumetric image is fed into a 3D encoder to produce a latent vector. Then a set of points are sampled from line primitives, which along with the latent vector are fed into the multilayer perceptron (MLP) decoder to form the skeleton, where we design a regularised loss function to constrain the shape of the reconstruction. Details will be given in the later sections.

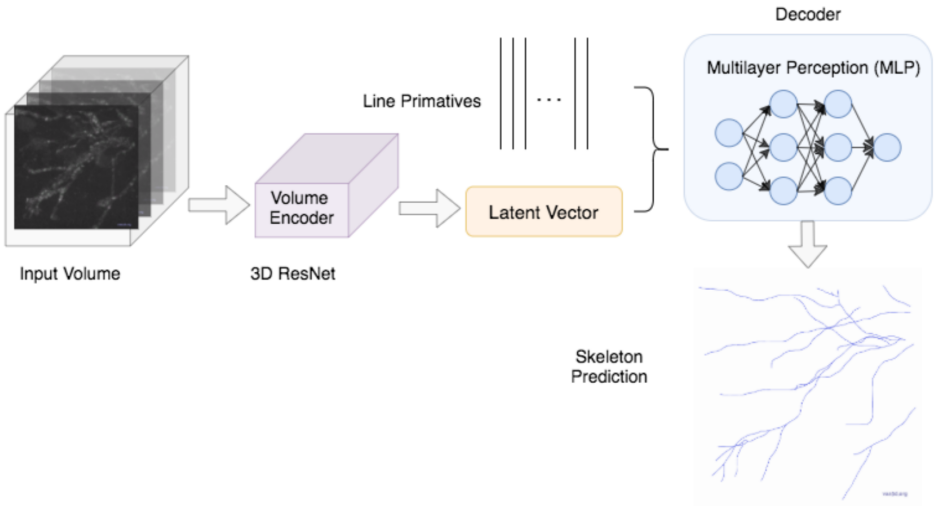


Figure 5.1: Network structure of our proposed deep neural network. Given an input volumetric image, the model is trained to regress a set of skeletons.

**Shape Encoding** The proposed network starts with a 3D convolutional voxel encoder to produce a m-dimension latent vector $f_{\boldsymbol{I}} \in \mathbb{R}^m$ representing surface shape of the objects in $I$, which is then used as input of the module for leaning skeleton, which is a point set $\mathcal{K} = \{\boldsymbol{p}_i \in \mathbb{R}^3\}_{i=1}^{n_\mathcal{K}}$, where $n_k$ denotes the number of points.

**decode a skeleton** Learning $\mathcal{K}$ from the latent vector $f_{\boldsymbol{I}}$ is a standard point set regression problem. Specifically, we define a set of 1D primitives of line

segment [0,1]. For each 1d primitive, we regularly sample points on it. Giving a sampled point $\boldsymbol{p}_i \in [0,1]$, we first map it to n-dimension space via a MLP $\varphi_\theta$ with parameter $\theta$ as :

$$\varphi_\theta\left(\boldsymbol{p}_i\right) = f_{p_i} \in \mathbb{R}^n \tag{5.1}$$

Then $f_{\boldsymbol{I}}$ and $f_{p_i}$ are concatenated and fed into another MLP $\varphi\prime_{\theta\prime}$ with parameter $\theta\prime$ to decode a point location in 3D space:

$$\varphi\prime_{\theta\prime}\left(f_{p_i} \oplus f_{\boldsymbol{I}}\right) = \boldsymbol{p}_i{}' \in \mathbb{R}^3 \tag{5.2}$$

A simple illustration of the process is shown as Fig 5.2.

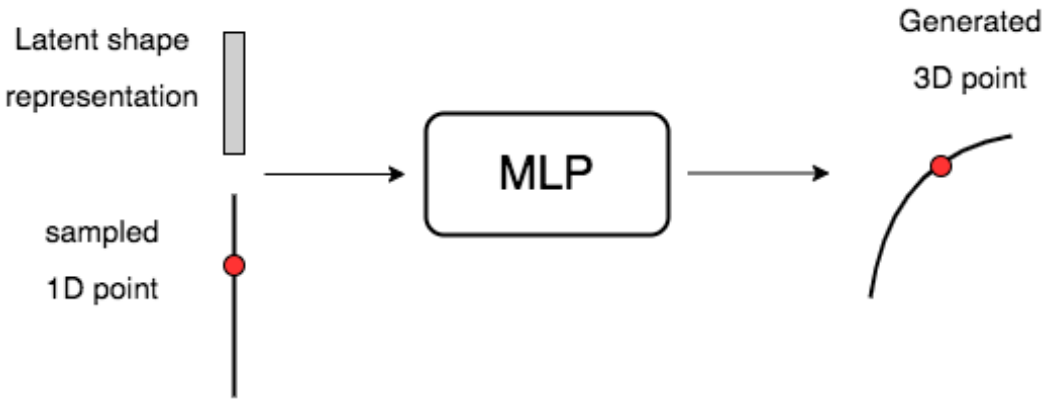


Figure 5.2: A simple illustration of decoding a skeleton. The input is a latent vector representing a shape and a sampled points on 1D line primitive.

Then we calculate the loss between a deformed sampled point and its corresponding skeleton points via a designed regularised loss function.

**Regularised Loss Function** Distribution of skeleton points satisfies geometric constraints that prevent them from free positioning in the 3D space. To further maintain the spatial relations between consecutive points and prevent irrational deformation to cause incorrect shape, we design a regularised loss function as:

$$L_\phi = L_{cd} + \lambda_1 L_{\text{smoothness}} + \lambda_2 L_{\text{edge}} \tag{5.3}$$

with

$$L_{cd} = \sum_{p \in \mathcal{P}} \min_{p^* \in \mathcal{P}^*} \|\boldsymbol{p} - \boldsymbol{p}^*\|_2^2 + \sum_{p^* \in \mathcal{P}^*} \min_{p \in \mathcal{P}} \|\boldsymbol{p} - \boldsymbol{p}^*\|_2^2 \tag{5.4}$$

$$L_{\text{smoothness}} = \sum_{\boldsymbol{p} \in \mathcal{P}} \left\| \boldsymbol{p} - \frac{1}{|\mathcal{N}(\boldsymbol{p})|} \sum_{\boldsymbol{p}' \in \mathcal{N}(\boldsymbol{p})} \boldsymbol{p}' \right\|_2^2 \tag{5.5}$$

$$L_{\text{edge}} = \sum_{p} \sum_{p' \in \mathcal{N}(\boldsymbol{p})} \|\boldsymbol{p} - \boldsymbol{p}'\|_2^2 \tag{5.6}$$

where Equation 5.4 is the loss term defined by chamfer distance (Watson and Gupta 1997), which is to regress the sample points close to its correction position on skeleton. $\mathcal{P}$ and $\mathcal{P}^*$ denote point sets on line primitives and ground-truth skeletal point sets respectively. Equation 5.5 represents the laplacian smoothness (Field 1988) for regularisation, where $\mathcal{N}(\boldsymbol{p})$ denotes neighbouring points of point $\boldsymbol{p} \in \mathcal{P}$ on the line primitive. Equation 5.6 s the loss term defined by edge length for regularisation, to penalise flying points causing long edge and preserve the shape. The overall loss function is weighted sum of all three losses as Equation 5.3, with hyperparameter $\lambda_1$ and $\lambda_2$. This loss design is similar to that of (Wang et al. 2018).

**Redundant Node Pruning** In inference, there is an issue that the number of points sampled from the line primitives are not perfectly matched to the number of nodes on a skeleton. Our strategy to address this issue is we train our model with a maximum sample points number, and prune the redundant node post inference. The pruning method is similar to that used in (Peng et al. 2011a).

First, the radius of every node on a skeleton is estimated by an range growth method. The method defines a sphere centred at a node and then grows gradually by a step of 1 voxel size, until 0.1% of the image voxels within the sphere are background voxels, which have lower intensity than the average intensity of the whole image. We go through every node on the generated skeleton, and remove those being heavily covered by other remaining nodes, as illustrated in Fig 5.3. After pruning, the neighbouring nodes of the removed node are connected to maintain the topology. Thus

we generate a simplified reconstruction with minimal nodes covering the full neuron region.

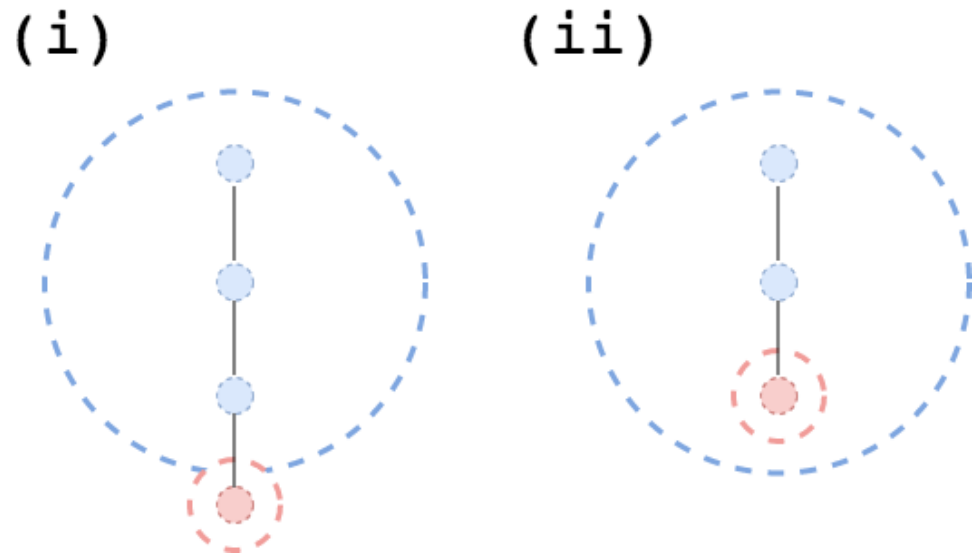


Figure 5.3: A simple illustration of redundant node pruning. Blue and red dots: nodes. Dotted circle: cover range of a node, according to its radius. (i) the red node will be kept (ii) the red node will be removed.

## 5.3 Evaluation

### 5.3.1 Experiment Settings

**Dataset Description** For evaluation using our curated dataset, we split 42,000 samples out of 50,000 for training and the remaining 8,000 for testing. For the BigNeuron dataset, we split 14374 samples for training and 2736 for testing. Note that our curated dataset originates from 1000 ground-truth reconstructions on whole mouse brain images. To further evaluate our method in ultra-volume image data on full morphology reconstruction, we test it on the same whole mouse brain images for another 200 neurons and compare the result with ground-truth reconstruction.

**Implementation and Training Details** For our proposed network, 3D ResNet18 (He et al. 2016, Hara et al. 2017) is utilised as the volumetric image encoder, which produces latent vector with dimension 512. The decoder to map the sample points to high dimension is MLP with 3 fully-connected (FC) layers, with respective 3, 64, and 128 neurons, ReLU activation is applied for the 3 layers. The decoder to decode the skeleton is MLP consisting of 4 FC layers, with respective 640, 256, 128 and 3 output neurons. ReLU activation is applied for the first 3 layers and tanh activation for the last layer. 30 line primitives, with 100 sampling points on each line are utilised as a side input to the decoder to form the skeleton. We set the weight in Equation 5.3, $\lambda_1$ and $\lambda_2$ as 0.3 and 0.1. We employ Adam Optimiser (Kingma and Ba 2014) with learning rate lr = 10-3 for 100 epochs. The model is implemented via Pytorch and trained on a NVDIA GeForce RTX 2080 Ti graphic card.

**Evaluation Metrics** To evaluate the performance of our neuron tracing methods, following metrics are employed in our comparison experiments:

1. The neuron distance metric is exploited when comparing the final reconstruction with the correspondingly ground truth (GT). Specifically, three distances scores defined in (Peng et al. 2010a) are calculated as:

   (a) The entire structure average: it first calculates the minimal spatial distance between each pair of two nearest nodes from reconstruction and GT, and then averages all these reciprocal minimal spatial distances;

   (b) different structure average : it only sums those reciprocal distances that are larger than 2 voxels, since the difference less than 2 voxels is hardly distinguished visually

   (c) percentage of different structure: it calculates the percentage of pairs of nodes whose reciprocal minimal spatial distances are more than 2 voxels.

2. The neuron length metric is also employed. Specifically, two length score are defined as:

(a) overall length score: it calculates the ratio of the length of reconstructed trace to that of GT (Zhang et al. 2007b).

(b) correct length score: it calculates the ratio of the correct portion of length reconstructed to that of GT (Wang et al. 2011, Dietenbeck et al. 2014).

### 5.3.2 Comparison Experiments

**Comparison on Curated Dataset** We compare our method with previously presented method in Chapter 4 and other deep learning methods in (Li et al. 2017, Zhao et al. 2019) using distance metrics. A quantitative result is shown in Table 5.1 and a qualitative comparison is shown in Fig 5.4. According to the results, our methods improve the tracing performance in all three domains and achieve state-of-the-art result. Note that in Fig 5.4, although close but irrelevant signal occurs, our method generates the correct trace without yielding an unreasonable shape like other tracers.

| **Tracing methods** | **Entire structure average** | **Different structure average** | **% of different structure** |
|---|---|---|---|
| Method in Chapter.4 | 3.4471 | 3.6753 | 0.1883 |
| Li et al. 2017 | 3.9975 | 4.1853 | 0.2658 |
| Zhao et al. 2019 | 3.6517 | 4.0125 | 0.1984 |
| **Our method** | **3.2143** | **3.4746** | **0.1751** |

Table 5.1: Comparison with other deep-learning based methods.

**Comparison on full morphology reconstruction** to evaluate our method on full morphology reconstruction in whole mouse brain images, we integrate our method with Ultratracer to extend its applicability in such large-scale data. We test it on 200 neurons, and calculate the reconstructed length ratio as well as corrected length ratio, by comparing with GT. The quantitative result is shown in Table 5.2 and a qualitative comparison is shown in Fig 5.5. According to the result, compared with existing tracing methods,

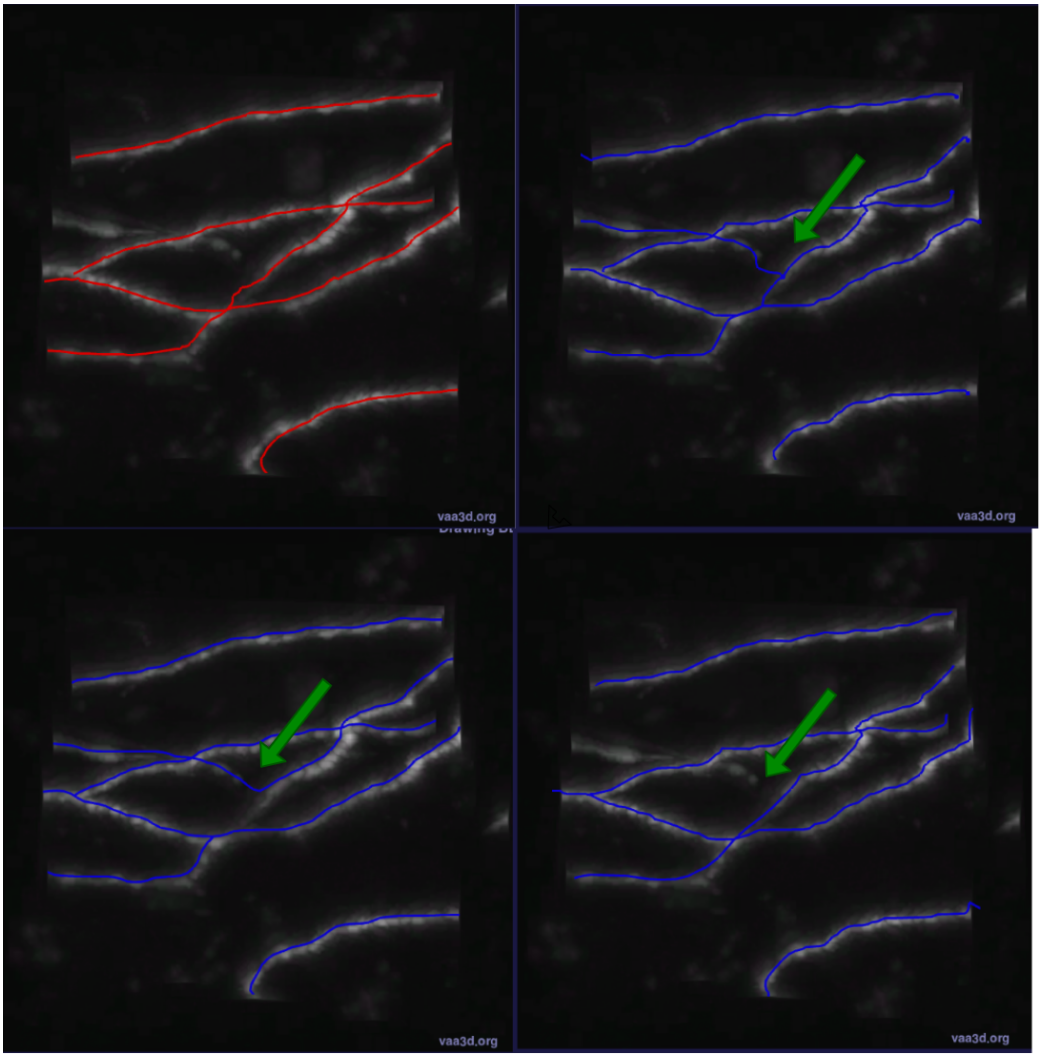


Figure 5.4: An example of comparison of tracing performance in area where close but irrelevant signal occurs. From top left to bottom right: GT, method in Chapter.4, Zhao et al.2019, our method. Note that in the arrow-pointing area, our method avoids a sharp turn or branch merging.

our proposed method is able to generate more meaningful reconstruction in extensive range.

**Comparison on BigNeuron Dataset** we further evaluate our method using the public dataset from BigNeuron. We compare the generated results with the other tracers in distance metrics of entire structure average, dif-

| | GT | Neutube | APP2 | SmarTracing | **Our Method** |
|---|---|---|---|---|---|
| Average Length | 175180 | 23185 | 18534 | 19647 | **29573** |
| Ratio of length to GT | 1 | 0.132 | 0.106 | 0.112 | **0.169** |
| Correct Length | 175180 | 17197 | 12423 | 16545 | **24396** |
| Ratio of correct Length to GT | 1 | 0.098 | 0.071 | 0.094 | **0.139** |

Table 5.2: A quantitative comparison result of our proposed method on full morphology reconstruction. All mentioned methods are applied within the Ultratracer framework.

| Method | Entire Structure Average | Different Structure Average | % of Different Structure |
|---|---|---|---|
| APP2 | 6.063 | 9.079 | 0.496 |
| Neutube | 12.741 | 21.379 | 0.434 |
| SmartTracing | 10.43 | 13.43 | 0.547 |
| Li2017 | 4.917 | 7.972 | 0.461 |
| Zhao2019 | 4.714 | 9.458 | 0.428 |
| **Ours** | **4.558** | **8.514** | **0.405** |

Table 5.3: A quantitative comparison with other tracers on BigNeuron Dataset

ferent structure average and percentage of different structure. The result is demonstrated in Table 5.3 and Fig 5.6 illustrates a qualitative comparison. The result shows that our method has achieved state-of-the-art performance, in a more general case across different neuron types and species.

### 5.3.3 Hyperparameter Search

The number of lines and number of points sampled per line are so important as they greatly affect the performance that they require a careful setting. Hence we conduct an experiment to quantify the effect of these parameters. We choose 4 different sets of number of lines and number of points per line and train each of them for 100 epochs on curated test dataset as in section 5.3.2. Time consumed of each epoch and neuron distance score compared with GT are recorded. The result is shown in Table 5.4. According to the result, small number of lines and sampled points result is trained faster but the accuracy is

| Number of Lines and Sample points | Entire Structure Average | Different Structure Average | % of Different Structure | Time per epoch (seconds) |
|---|---|---|---|---|
| 30, 50 | 3.5143 | 4.0265 | 0.212 | 68 |
| 30, 100 | 3.2143 | 3.4746 | 0.1751 | 163 |
| 30, 150 | 3.2141 | 3.4735 | 0.1743 | 279.5 |
| 40, 150 | 3.214 | 3.4735 | 0.1742 | 421 |

Table 5.4: A numerical comparison of how number of lines and sampled points affect the performance.

low. While large number of lines and points improve the accuracy, but after a certain point, the improvement becomes minor whilst the time consumed is significantly increased. This could be ascribed to the implementation of Chamfer loss in Equation 5.4, which has a compute cost that is quadratic in the number of input points. Considering the practical implementation, we choose 30 lines and 100 sample points per line as our default parameters through the whole comparison experiments.

## 5.4 Discussion

According to the experimental results, our proposed method has further improved the accuracy of automatic neuron tracing and achieves state-of-the-art performance on a variety of neuronal images with different. The effectiveness of our method is highly attributed to the learning model which directly extracts skeleton from volumetric image data, without error accumulating in different stages. Moreover, the design of line primitives as side input for decoding and the regularised loss function well preserve the topology of neuron tree. The design targets to the main obstacle in automatic neuron tracing, which is the close but irrelevant signals, as we discussed in Chapter 3. The improvement on tracing result proves the correctness of our analysis and discussion.

## 5.5 Summary

In this chapter, we propose a novel approach for 3D neuron tracing. Our target is to overcome the main obstacle in neuron tracing, which is the close but

irrelevant signals. Our model is designed to learn to directly extract skeleton from volume image, along with a design of line primitives and regularised loss function to preserve the topology. Experiments show that our methods achieve state-of-the-art performance. A direction of future work is to work on combine our work in Chapter 4 with Chapter 5, to enable learning from a enhanced image by occupancy network, which might further improve the accuracy.

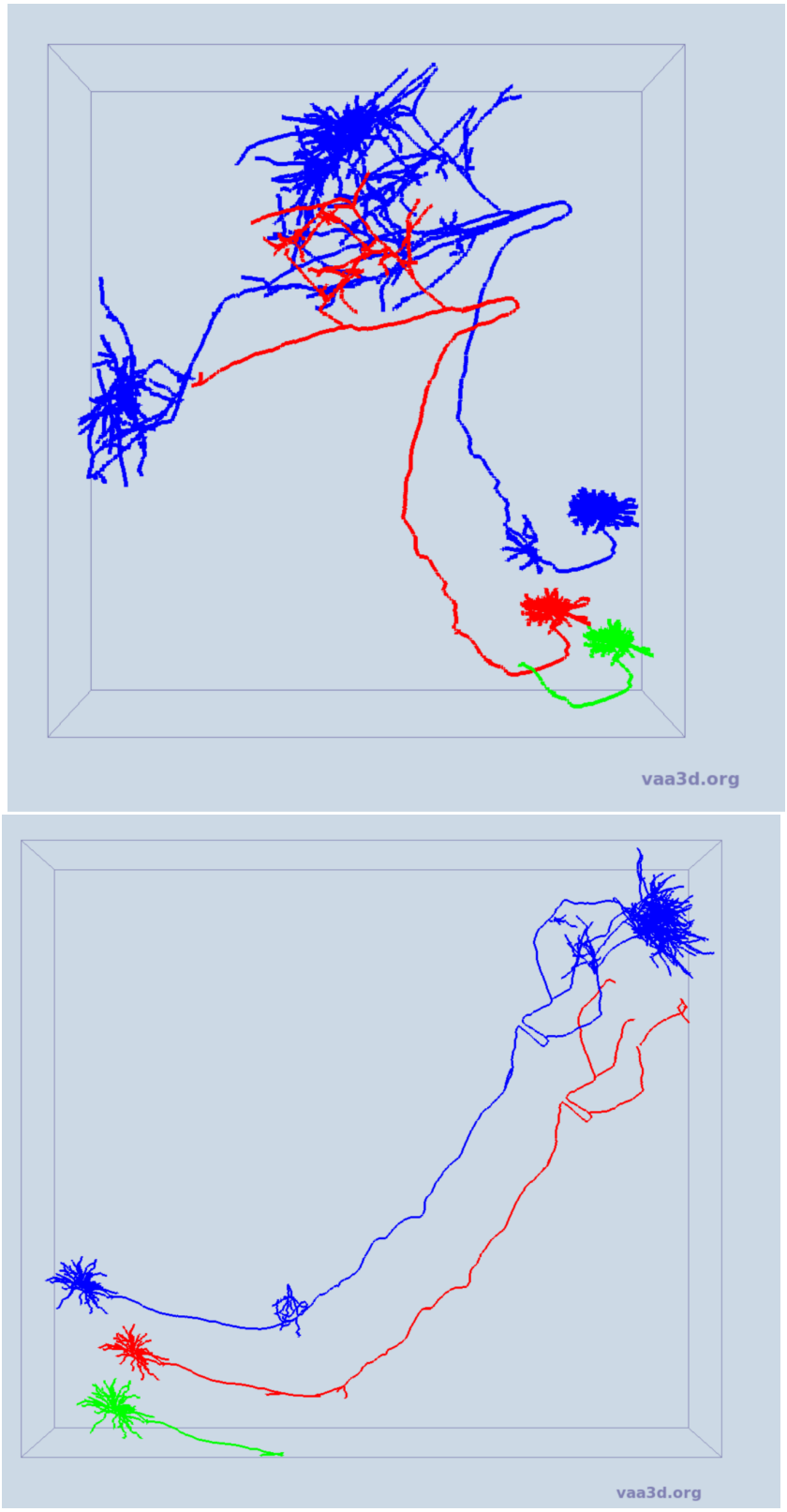




Figure 5.5: A qualitative comparison of our proposed method on full morphology reconstruction from whole mouse brain image. Only skeletons are shown for better visualisation. Red:our method. Blue: GT. Green: Ultra-tracer+APP2. The reconstructions are intentionally displaced for best view.

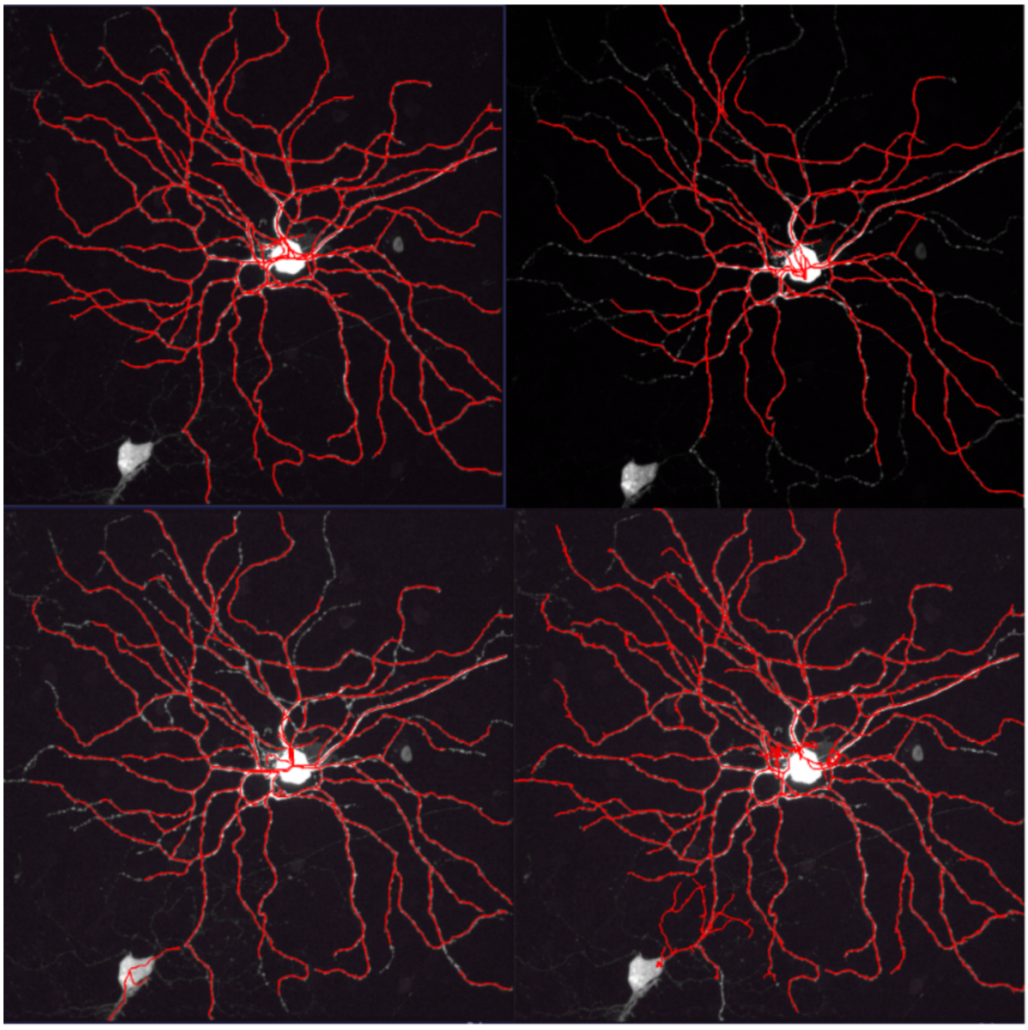

Figure 5.6: A qualitative comparison of our method with other tracers on a mouse retinal ganglion cell image from BigNeuron dataset. From top left to bottom right GT, Li 2017 et al, Zhao 2019 et al, our method.

# Chapter 6

# Conclusion and Future Work

In this chapter, we summarise our works and conclude the thesis. We also discuss the possible directions of future work to further improve on the research field.

## 6.1 Conclusion

This thesis targets at developing efficient and fully-automatic method for neuron tracing from optical microscopy image. In order to achieve this goal, we accomplish a few of works including data generation, failure analysis and algorithm development.

To prepare sufficient amount of data for neuron tracing algorithm development, we propose a pipeline for complete morphology reconstruction of single neurons. The pipeline consists of four stages as initial reconstruction, semi-manual reconstruction, cross-check and automatic refinement. Specifically, we propose a novel approach for the automatic refinement stage, by a 2-time optimal path finding and deformation, which leads to a final standard reconstruction with trace adhering to the centre of the neuronal signal. Following the pipeline, we manage to generate more than 2,000 standard reconstructions, which significantly eases the limitation of available data in the research field. With the data acquired, we conduct a systematic and quantitative failure analysis regarding existing automatic tracing approaches, to acknowledge the main obstacle toward a more efficient and accurate trac-

ing method. The analysis starts from a false sample collection, in which stage we establish a workflow and produce more than 10,000 samples for inspection. Then we manually check and classify the false samples into types according to the false appearance and the possible cause. The quantitative result indicates that the main obstacles in neuron tracing are the close but irrelevant signals, as well as weak and punctuated relevant signals. To our knowledge, the systematic and quantitative failure analysis is the first attempt in the research field and it determines the direction and target in our tracing method development in later stage. The details of data generation and failure analysis are presented in Chapter 3.

To overcome the aforementioned obstacles and achieve a superior tracing performance, we propose a fully-automatic neuron tracing method via occupancy learning and curve skeleton extraction. The idea is to enhance the image segmentation to capture weak and punctuated signals while discriminate irrelevant signals. Therefore we employ a deep neural network to learn an occupancy function to determine a continuous decision boundary which represents the surface of a neurite. Along with a balanced foreground/background input points sampling strategy, the occupancy network addresses the extreme class imbalance problem in neuronal image, and captures relevant signals efficiently. We also apply a curve skeleton extraction algorithm to trace the neurite. Leveraging the symmetric property from the cylindrical shape of the neurite, the approaches can refine any incomplete reconstruction where punctuated signal occurs. We train and test our method using a carefully curated dataset partitioned from reconstructions in Chapter. 3. The experimental results show that our proposed occupancy network largely outperforms existing deep learning models for image segmentation in terms of IoU score. And the whole tracing approach outperforms existing tracers in most terms of neuron distance metric. The details of the proposed method are discussed in Chapter 4.

Taking one step further from the previously proposed method in Chapter 4, we propose a fully-automatic neuron tracing method via learning a topology-preserving skeleton extraction. We aim to tackle the main obstacle i.e. irrelevant nearby signals in neuron tracing. Although the enhanced image

segmentation in previous method addresses this issue to some degree, it still remain a main challenge in automatic neuron tracing. Therefore, we propose an novel learning model to directly extract skeleton from the input optical microscopy image data, without any error accumulation occurring in other multi-stage methods. The network is designed in an encoder-decoder structure. A 3D convolutional based encoder is employed to learn latent shape encodings of the neuron, and with the latent feature vector, a multilayer perceptrons (MLPs) based decoder learns the deformation from a set of line primitives to form the skeleton while maintain the topology. A designed regularisation loss function further constrains the geometry of a neurite, which prevents the skeletal point from moving freely to form an unreasonable shape. In inference, we also add a node pruning procedure to remove any redundant skeletal points and obtain a simplified reconstruction with minimal nodes covering the full neuron region. We train our method using the dataset generated in Chapter 4 , as well as a public BigNeuron (Peng et al. 2015) dataset separately. The experimental results indicate that our method significantly outperforms existing methods, in cases of partial reconstruction on cropped 3D image block, full morphology reconstruction on whole mouse brain image and complete tracing on BigNeuron dataset, in different terms of neuron distance and length metrics. The details of the proposed method are presented in Chapter 5.

## 6.2 Limitations and Future Work

**Further improvement on fine geometry** As we presented and discussed in Chapter 4, we adopted an occupancy network to learn an continuous representations of 3D geometry for neuronal structure in volumetric image, and achieves superior accuracy in neuronal image segmentation. However, its performances in some fine and detailed area are unsatisfactory. The issue derives from a common limitation of this kind of occupancy network, which is the simple fully-connected structure. The fully connected network is incapable to integrate local information in the image voxels, or to incorporates inductive biases so yields in more general cases. This has degraded their

performance on structured reasoning (Peng et al. 2020b). In recent studies, some revised occupancy network (Jiang et al. 2020, Chibane et al. 2020) have been introduced to reconstruct more complex shape by preserving fine geometric details. The key idea of these methods is to use a convolutional volume decoder, exploiting the spatial translation equivariant property of convolutional neural network (CNN). A typical structure of a 3D convolutional volume decoder is illustrated in Fig 6.1. For a given 3D feature volume encoded by a 3D encoder, rather than directly decoding as in our method in Chapter 4, the volume decoder processes the feature volume using 3D U-nets to aggregate local and global information. For a query point $p \in R^3$, the point-wise feature vector $\psi(\mathbf{x}, \mathbf{p})$is obtained via trilinear interpolation. Then along with the feature vector, a query voxel at location p will be fed into a smaller fully-connected network to predict the occupancy by $f_\theta(\mathbf{p}, \psi(\mathbf{p}, \mathbf{x}))$. These models show a superior experimental results in terms of 3D reconstruction of indoor scenes and is promising to be extended to neuronal image segmentation as a future work.

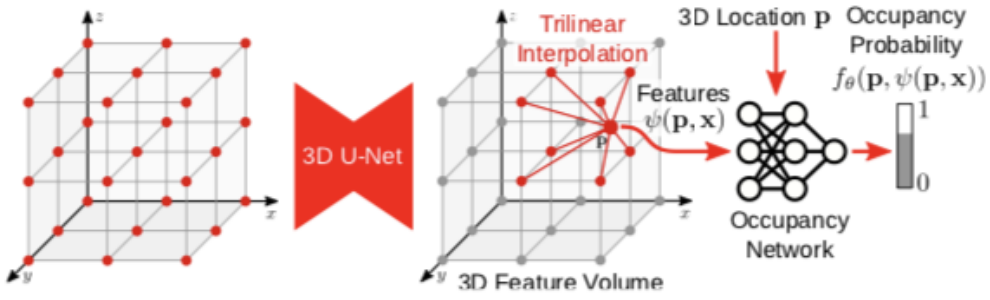


Figure 6.1: An illustration of a 3D convolutional volume decoder (Peng et al. 2020b).

**Multi-Neuron Distinction** In DIADEM and BigNeuron dataset, each image only contains one single neuron for tracing but it is not the case in whole mouse brain image. Although sparsely labelled, an average amount of 2000 neurons are imaged in a whole mouse brain, resulting in numerous areas with densely packed neurons. An example is shown in Fig 6.2. In such area, most

of automatic tracing methods will fail. An intuitive approach to distinguish multiple neurons is to conduct an instance segmentation. However, they do not have any semantic difference. We propose a method that could be implemented in the future to potentially resolve this issue. In a multi-neuron area, first we conduct a series of auto tracing, which starts from the soma location of each neuron respectively. Since the tracing results from each neuron are highly prone to overlap, we scan through each voxel location to records its coverage. Any voxel which is covered by more than two reconstructed will be fed into a classifier to determine its occupancy by a certain neuron. The classifier can be a learned model, or a simple decision tree using features such as a geodesic distance to each neuron, branch level on the reconstructed tree and so on. An efficient multi-neutron distinction will extend the applicability of any tracing method and help to generate more full morphology data of single neurons.

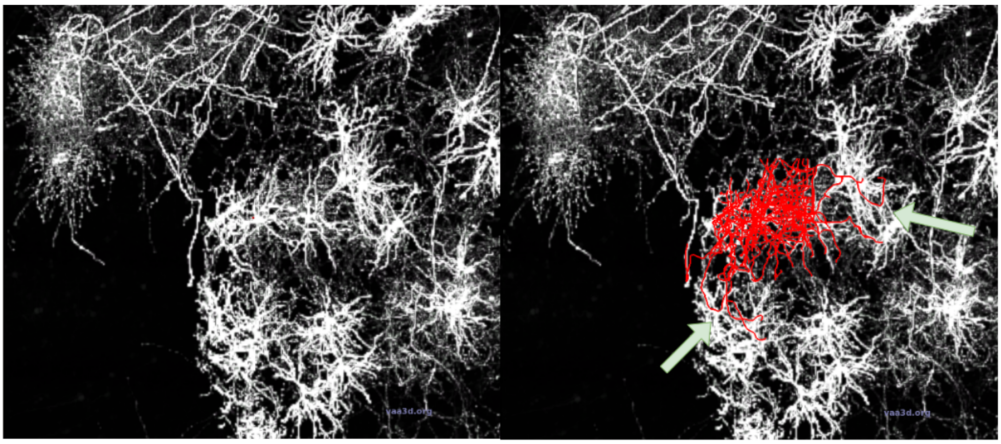

Figure 6.2: A failed example of neuron reconstruction in densely packed areas. Left image: the original image. Right image: a false reconstruction, with branching overlapping signals from adjacent neurons.

**Multi-scale learning** Due to the limitation of computation and memory, the image samples to train our proposed learning model are constrained to a relatively small size. Compared with the actual size of a neuron in

its full morphology, it definitely loses some degree of global information. Since the whole mouse brain image is converted to an octree structure with muliti-resolution in TeraFly (Bria et al. 2016) it motivates us to implement a multi-scale learning as a further improvement for neuron reconstruction. Fig 6.3 gives an overview of a multi-scale learning model (Chibane et al. 2020), given an input, it computes a 3D grid of multi-scale features, encoding global and local properties of the input shape. Then a set of deep features $\mathbf{F}_1(\mathbf{p}) \ldots \mathbf{F}_n(\mathbf{p})$is extracted from the grid at location $p$. A learned point occupancy decoder then takes these features to determine the occupancy of the point $p$. The original method takes a single voxel grid as input, encoded by a series of multi-dimension encoders. But it is possible for us to take a series of multi-resolution images as inputs, such as the multi-resolution images converted by TeraFly, and encode it using a series of fixed-dimension encoders to obtain the same deep features. Then we can better exploit the global shape information toward a better reconstruction.

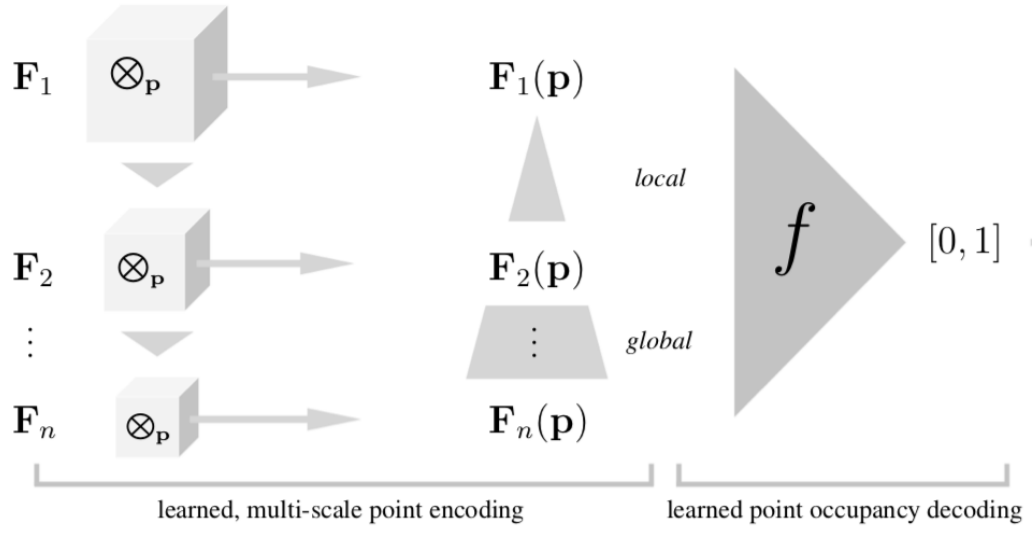


Figure 6.3: An overview of a multi-scale learning network (Chibane et al. 2020).

**Biological information exploit** Neurons of a certain type share some common biological characteristics, which can be further exploited as a topological

constraint for reconstruction. Some quantitative measurement, such as number of bifurcations, path length, ratio of the diameter between a father and its child branch can be either introduced to define a loss function term or a in-time evaluation metric during tracing.

# References

Abdul-Karim, M.-A., Roysam, B., Dowell-Mesfin, N. M., Jeromin, A., Yuksel, M. and Kalyanaraman, S., 2005. Automatic selection of parameters for vessel/neurite segmentation algorithms. *IEEE Transactions on Image Processing*, 14 (9), 1338–1350.

Acciai, L., Soda, P. and Iannello, G., 2016. Automated neuron tracing methods: an updated account. *Neuroinformatics*, 14 (4), 353–367.

Akram, M. A., Nanda, S., Maraver, P., Armañanzas, R. and Ascoli, G. A., 2018. An open repository for single-cell reconstructions of the brain forest. *Scientific data*, 5 (1), 1–12.

Al-Kofahi, K. A., Lasek, S., Szarowski, D. H., Pace, C. J., Nagy, G., Turner, J. N. and Roysam, B., 2002. Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Transactions on information technology in biomedicine*, 6 (2), 171–187.

Aransay, A., Rodríguez-López, C., García-Amado, M., Clascá, F. and Prensa, L., 2015. Long-range projection neurons of the mouse ventral tegmental area: a single-cell axon tracing analysis. *Frontiers in neuroanatomy*, 9, 59.

Ascoli, G. A., 2008. Neuroinformatics grand challenges.

Ascoli, G. A., Donohue, D. E. and Halavi, M., 2007. Neuromorpho. org: a central resource for neuronal morphologies. *Journal of Neuroscience*, 27 (35), 9247–9251.

Barrett, W. A. and Mortensen, E. N., 1997. Interactive live-wire boundary extraction. *Medical Image Analysis*, 1 (4), 331–341. URL https://www.sciencedirect.com/science/article/pii/S1361841597850050.

Benninger, K., Hood, G., Simmel, D., Tuite, L., Wetzel, A., Ropelewski, A., Watkins, S., Watson, A. and Bruchez, M., 2020. Cyberinfrastructure of a multi-petabyte microscopy resource for neuroscience research. *Practice and Experience in Advanced Research Computing*, 1–7.

Bria, A., Iannello, G., Onofri, L. and Peng, H., 2016. Terafly: real-time three-dimensional visualization and annotation of terabytes of multidimensional volumetric images. *Nature methods*, 13 (3), 192–194.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R., 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 737–744.

Brown, K. M., Barrionuevo, G., Canty, A. J., De Paola, V., Hirsch, J. A., Jefferis, G. S., Lu, J., Snippe, M., Sugihara, I. and Ascoli, G. A., 2011. The diadem data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9 (2), 143–157.

Brown, K. M., Donohue, D. E., D'Alessandro, G. and Ascoli, G. A., 2005. A cross-platform freeware tool for digital reconstruction of neuronal arborizations from image stacks. *Neuroinformatics*, 3 (4), 343–359.

Ramón y Cajal, S., 1928. Degeneration and regeneration of the nervous system.

y Cajal, S. R., 1911. *Histologie du système nerveux de l'homme & des vertébrés: Cervelet, cerveau moyen, rétine, couche optique, corps strié, écorce cérébrale générale & régionale, grand sympathique*, volume 2. A. Maloine.

Cannon, R., Turner, D., Pyapali, G. and Wheal, H., 1998. An on-line archive of reconstructed hippocampal neurons. *Journal of Neuroscience Methods*, 84 (1), 49–54. URL https://www.sciencedirect.com/science/article/pii/S0165027098000910.

Chaudhury, A. and Godin, C., 2020. Skeletonization of plant point cloud data in stochastic optimization framework. *bioRxiv*.

Chen, H., Xiao, H., Liu, T. and Peng, H., 2015. Smarttracing: self-learning-based neuron reconstruction. *Brain informatics*, 2 (3), 135–144.

Chen, L.-C., Papandreou, G., Schroff, F. and Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chibane, J., Alldieck, T. and Pons-Moll, G., 2020. Implicit functions in feature space for 3d shape reconstruction and completion. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6970–6981.

Chopra, S., Hadsell, R. and LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, IEEE, volume 1, 539–546.

Choromanska, A., Chang, S. and Yuste, R., 2012. Automatic reconstruction of neural morphologies with multi-scale tracking. *Frontiers in neural circuits*, 6, 25.

Collobert, R. and Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*, 160–167.

Daigle, T. L., Madisen, L., Hage, T. A., Valley, M. T., Knoblich, U., Larsen, R. S., Takeno, M. M., Huang, L., Gu, H., Larsen, R. et al., 2018. A suite of transgenic driver and reporter mouse lines with enhanced brain-cell-type targeting and functionality. *Cell*, 174 (2), 465–480.

DeFelipe, J., López-Cruz, P. L., Benavides-Piccione, R., Bielza, C., Larrañaga, P., Anderson, S., Burkhalter, A., Cauli, B., Fairén, A., Feldmeyer, D. et al., 2013. New insights into the classification and nomenclature of cortical gabaergic interneurons. *Nature Reviews Neuroscience*, 14 (3), 202–216.

Dietenbeck, T., Varray, F., Kybic, J., Basset, O. and Cachard, C., 2014. Neuromuscular fiber segmentation through particle filtering and discrete optimization. S. Ourselin and M. A. Styner, eds., *Medical Imaging 2014: Image Processing*, International Society for Optics and Photonics, SPIE, volume 9034, 71 – 80. URL https://doi.org/10.1117/12.2043257.

DIJKSTRA, E., 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271. URL http://eudml.org/doc/131436.

Dijkstra, E. W., 1959. A note on two problems in connexion with graphs. *Numerische mathematik*, 1 (1), 269–271.

Dima, A., Scholz, M. and Obermayer, K., 2002. Automatic segmentation and skeletonization of neurons from confocal microscopy images based on the 3-d wavelet transform. *IEEE Transactions on image processing*, 11 (7), 790–801.

Donohue, D. E. and Ascoli, G. A., 2011. Automated reconstruction of neuronal morphology: an overview. *Brain research reviews*, 67 (1-2), 94–102.

Dou, Q., Yu, L., Chen, H., Jin, Y., Yang, X., Qin, J. and Heng, P.-A., 2017. 3d deeply supervised network for automated segmentation of volumetric medical images. *Medical Image Analysis*, 41, 40–54. URL https://www.sciencedirect.com/science/article/pii/S1361841517300725, special Issue on the 2016 Conference on Medical Image Computing and Computer Assisted Intervention (Analog to MICCAI 2015).

Ecker, J. R., Geschwind, D. H., Kriegstein, A. R., Ngai, J., Osten, P., Polioudakis, D., Regev, A., Sestan, N., Wickersham, I. R. and Zeng, H.,

2017. The brain initiative cell census consortium: lessons learned toward generating a comprehensive brain cell atlas. *Neuron*, 96 (3), 542–557.

Economo, M. N., Clack, N. G., Lavis, L. D., Gerfen, C. R., Svoboda, K., Myers, E. W. and Chandrashekar, J., 2016. A platform for brain-wide imaging and reconstruction of individual neurons. *Elife*, 5, e10566.

Evers, J. F., Schmitt, S., Sibila, M. and Duch, C., 2005. Progress in functional neuroanatomy: precise automatic geometric reconstruction of neuronal morphology from confocal image stacks. *Journal of neurophysiology*, 93 (4), 2331–2342.

Falcão, A. X., Udupa, J. K., Samarasekera, S., Sharma, S., Hirsch, B. E. and de A. Lotufo, R., 1998. User-steered image segmentation paradigms: Live wire and live lane. *Graphical Models and Image Processing*, 60 (4), 233–260. URL https://www.sciencedirect.com/science/article/pii/S1077316998904750.

Feng, L., Zhao, T. and Kim, J., 2015. neutube 1.0: a new design for efficient neuron reconstruction software based on the swc format. *eneuro*, 2 (1).

Field, D. A., 1988. Laplacian smoothing and delaunay triangulations. *Communications in applied numerical methods*, 4 (6), 709–712.

Freeman, W. and Adelson, E., 1991. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13 (9), 891–906.

Gala, R., Chapeton, J., Jitesh, J., Bhavsar, C. and Stepanyants, A., 2014. Active learning of neuron morphology for accurate automated tracing of neurites. *Frontiers in neuroanatomy*, 8, 37.

Gillette, T. A., Brown, K. M., Svoboda, K., Liu, Y. and Ascoli, G. A., 2011. Diademchallenge. org: a compendium of resources fostering the continuous development of automated neuronal reconstruction.

Gong, H., Xu, D., Yuan, J., Li, X., Guo, C., Peng, J., Li, Y., Schwarz, L. A., Li, A., Hu, B. et al., 2016. High-throughput dual-colour precision imaging for brain-wide connectome with cytoarchitectonic landmarks at the cellular level. *Nature communications*, 7 (1), 1–12.

Haan, B., Rivenson, Y., Wu, Y. and Ozcan, A., 2019. Deep-learning-based image reconstruction and enhancement in optical microscopy. *Proceedings of the IEEE*, PP, 1–21.

Haehn, D., Kaynig, V., Tompkin, J., Lichtman, J. W. and Pfister, H., 2018. Guided proofreading of automatic segmentations for connectomics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 9319–9328. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00971.

Han, Y., Kebschull, J. M., Campbell, R. A., Cowan, D., Imhof, F., Zador, A. M. and Mrsic-Flogel, T. D., 2018. The logic of single-cell projections from visual cortex. *Nature*, 556 (7699), 51–56.

Hara, K., Kataoka, H. and Satoh, Y., 2017. Learning spatio-temporal features with 3d residual networks for action recognition. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 3154–3160.

He, K., Zhang, X., Ren, S. and Sun, J., 2015. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37 (9), 1904–1916.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29 (6), 82–97.

Holschneider, M., Kronland-Martinet, R., Morlet, J. and Tchamitchian, P., 1990. A real-time algorithm for signal analysis with the help of the wavelet transform. *Wavelets*, Springer, 286–297.

Huang, Q., Chen, Y., Liu, S., Xu, C., Cao, T., Xu, Y., Wang, X., Rao, G., Li, A., Zeng, S. and Quan, T., 2020. Weakly supervised learning of 3d deep network for neuron reconstruction. *Frontiers in Neuroanatomy*, 14, 38. URL https://www.frontiersin.org/article/10.3389/fnana.2020.00038.

Ji, S., Xu, W., Yang, M. and Yu, K., 2012. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35 (1), 221–231.

Jiang, C., Sud, A., Makadia, A., Huang, J., Nießner, M., Funkhouser, T. et al., 2020. Local implicit grid representations for 3d scenes. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6001–6010.

Ke, W., Chen, J., Jiao, J., Zhao, G. and Ye, Q., 2017. Srn: Side-output residual network for object symmetry detection in the wild. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1068–1076.

Kingma, D. P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097–1105.

LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *Nature*, 521, 436–44.

Lee, P.-C., Chuang, C.-C., Chiang, A.-S. and Ching, Y.-T., 2012. High-throughput computer method for 3d neuronal structure reconstruction

from the image stack of the drosophila brain and its applications. *PLoS Comput Biol*, 8 (9), e1002658.

Li, Q. and Shen, L., 2020. 3d neuron reconstruction in tangled neuronal image with deep networks. *IEEE Transactions on Medical Imaging*, 39 (2), 425–435.

Li, R., Zeng, T., Peng, H. and Ji, S., 2017. Deep learning segmentation of optical microscopy images improves 3-d neuron reconstruction. *IEEE transactions on medical imaging*, 36 (7), 1533–1541.

Li, S., Quan, T., Zhou, H., Yin, F., Li, A., Fu, L., Luo, Q., Gong, H. and Zeng, S., 2019. Identifying weak signals in inhomogeneous neuronal images for large-scale tracing of sparsely distributed neurites. *Neuroinformatics*, 17.

Lin, R., Wang, R., Yuan, J., Feng, Q., Zhou, Y., Zeng, S., Ren, M., Jiang, S., Ni, H., Zhou, C. et al., 2018. Cell-type-specific and projection-specific brain-wide reconstruction of single neurons. *Nature methods*, 15 (12), 1033–1036.

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, C., Ke, W., Qin, F. and Ye, Q., 2018. Linear span network for object skeleton detection. *Proceedings of the European Conference on Computer Vision (ECCV)*, 133–148.

Liu, Y., 2011. The diadem and beyond.

Lorensen, W. E. and Cline, H. E., 1987. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21 (4), 163–169.

McDonald, T., Usher, W., Morrical, N., Gyulassy, A., Petruzza, S., Federer, F., Angelucci, A. and Pascucci, V., 2020. Improving the usability of virtual reality neuron tracing with topological elements. *CoRR*, abs/2009.01891. URL https://arxiv.org/abs/2009.01891.

Meijering, E., 2010. Neuron tracing in perspective. *Cytometry Part A*, 77 (7), 693–704.

Meijering, E., Jacob, M., Sarria, J., Steiner, P., Hirling, H. and Unser, M., 2004. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*, 58A.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S. and Geiger, A., 2019. Occupancy networks: Learning 3d reconstruction in function space. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4460–4470.

Mosinska, A., Tarnawski, J. and Fua, P., 2017. Active learning and proofreading for delineation of curvilinear structures. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 165–173.

Myatt, D., Hadlington, T., Ascoli, G. and Nasuto, S., 2012. Neuromantic–from semi-manual to semi-automatic reconstruction of neuron morphology. *Frontiers in neuroinformatics*, 6, 4.

Narro, M. L., Yang, F., Kraft, R., Wenk, C., Efrat, A. and Restifo, L. L., 2007. Neuronmetrics: software for semi-automated processing of cultured neuron images. *Brain research*, 1138, 57–75.

Nathan, S. and Kansal, P., 2019. Skeletonnet: Shape pixel to skeleton pixel. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Nie, Y., Lin, Y., Han, X., Guo, S., Chang, J., Cui, S., Zhang, J. et al., 2020. Skeleton-bridged point completion: From global inference to local adjustment. *Advances in Neural Information Processing Systems*, 33, 16119–16130.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B. et al., 2018. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*.

Peng, H., Bria, A., Zhou, Z., Iannello, G. and Long, F., 2014a. Extensible visualization and analysis for multidimensional images using vaa3d. *Nature protocols*, 9 (1), 193–208.

Peng, H., Hawrylycz, M., Roskams, J., Hill, S., Spruston, N., Meijering, E. and Ascoli, G. A., 2015. Bigneuron: large-scale 3d neuron reconstruction from optical microscopy images. *Neuron*, 87 (2), 252–256.

Peng, H., Long, F. and Myers, G., 2011a. Automatic 3d neuron tracing using all-path pruning. *Bioinformatics*, 27 (13), i239–i247.

Peng, H., Long, F., Zhao, T. and Myers, E., 2011b. Proof-editing is the bottleneck of 3d neuron reconstruction: the problem and solutions. *Neuroinformatics*, 9 (2), 103–105.

Peng, H., Ruan, Z., Atasoy, D. and Sternson, S., 2010a. Automatic reconstruction of 3d neuron structures using a graph-augmented deformable model. *Bioinformatics*, 26 (12), i38–i46.

Peng, H., Ruan, Z., Long, F., Simpson, J. H. and Myers, E. W., 2010b. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28 (4), 348–353.

Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., Zhou, Z., Gonzalez-Bellido, P. T., Oh, S. W., Chen, J., Mitra, A., Tsien, R. W., Zeng, H., Ascoli, G. A., Iannello, G., Hawrylycz, M., Myers, E. and Long,

F., 2014b. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature Communications*, 5 (1).

Peng, H., Tang, J., Xiao, H., Bria, A., Zhou, J., Butler, V., Zhou, Z., Gonzalez-Bellido, P. T., Oh, S. W., Chen, J. et al., 2014c. Virtual finger boosts three-dimensional imaging and microsurgery as well as terabyte volume image visualization and analysis. *Nature communications*, 5 (1), 1–13.

Peng, H., Xie, P., Liu, L., Kuang, X., Wang, Y., Qu, L., Gong, H., Jiang, S., Li, A., Ruan, Z. et al., 2020a. Brain-wide single neuron reconstruction reveals morphological diversity in molecularly defined striatal, thalamic, cortical and claustral neuron types. *bioRxiv*, 675280.

Peng, H., Zhou, Z., Meijering, E., Zhao, T., Ascoli, G. A. and Hawrylycz, M., 2017. Automatic tracing of ultra-volumes of neuronal images. *Nature methods*, 14 (4), 332.

Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M. and Geiger, A., 2020b. Convolutional occupancy networks. *European Conference on Computer Vision*, Springer, 523–540.

Pettie, S., 2008. *Minimum Spanning Trees*, Boston, MA: Springer US. 541–544. URL https://doi.org/10.1007/978-0-387-30162-4_239.

Puppo, F., George, V. and Silva, G. A., 2018. An optimized structure-function design principle underlies efficient signaling dynamics in neurons. *Scientific reports*, 8 (1), 10460.

Qin, H., Zhang, S., Liu, Q., Chen, L. and Chen, B., 2020. Pointskelcnn: Deep learning-based 3d human skeleton extraction from point clouds. *Computer Graphics Forum*, Wiley Online Library, volume 39, 363–374.

Radojevic, M. and Meijering, E., 2019. Automated neuron reconstruction from 3d fluorescence microscopy images using sequential monte carlo estimation. *Neuroinformatics*, 17.

Rivenson, Y. and Ozcan, A., 2018. Toward a thinking microscope: Deep learning in optical microscopy and image reconstruction.

Ronneberger, O., Fischer, P. and Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention*, Springer, 234–241.

Schwartz, E. L., 1993. *Computational neuroscience*. Mit Press.

Senft, S. L., 2011. A brief history of neuronal reconstruction. *Neuroinformatics*, 9 (2-3), 119–128.

Shen, D., Wu, G. and Suk, H.-I., 2017. Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19 (1), 221–248. URL https://doi.org/10.1146/annurev-bioeng-071516-044442, pMID: 28301734.

Shen, W., Zhao, K., Jiang, Y., Wang, Y., Zhang, Z. and Bai, X., 2016. Object skeleton extraction in natural images by fusing scale-associated deep side outputs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 222–230.

Shit, S., Paetzold, J. C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Pluim, J. P., Bauer, U. and Menze, B. H., 2021. cldice-a novel topology-preserving loss function for tubular structure segmentation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16560–16569.

Sorensen, T. A., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5, 1–34.

Sutskever, I., Martens, J. and Hinton, G. E., 2011. Generating text with recurrent neural networks. *ICML*.

Svoboda, K., 2011. The past, present, and future of single neuron reconstruction. *Neuroinformatics*, 9 (2-3), 97.

Swanson, L. W., Newman, E., Araque, A. and Dubinsky, J. M., 2017. *The beautiful brain: the drawings of Santiago Ramón y Cajal*. Abrams.

Szegedy, C., Toshev, A. and Erhan, D., 2013. Deep neural networks for object detection.

Tagliasacchi, A., Zhang, H. and Cohen-Or, D., 2009. Curve skeleton extraction from incomplete point cloud. *ACM SIGGRAPH 2009 papers*, 1–9.

Taigman, Y., Yang, M., Ranzato, M. and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Tang, J., Han, X., Tan, M., Tong, X. and Jia, K., 2021. Skeletonnet: A topology-preserving solution for learning mesh reconstruction of object surfaces from rgb images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Taylor, L. M., Flanagan, D. P. and Mills-Curran, W., 1986. Genesis finite element mesh file format. Technical report, Sandia National Labs., Albuquerque, NM (USA).

Vasilkoski, Z. and Stepanyants, A., 2009. Detection of the optimal neuron traces in confocal microscopy images. *Journal of neuroscience methods*, 178 (1), 197–204.

Veldman, M. B., Park, C. S., Eyermann, C. M., Zhang, J. Y., Zuniga-Sanchez, E., Hirano, A. A., Daigle, T. L., Foster, N. N., Zhu, M., Langfelder, P. et al., 2020. Brainwide genetic sparse cell labeling to illuminate the morphology of neurons and glia with cre-dependent morf mice. *Neuron*, 108 (1), 111–127.

Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W. and Jiang, Y.-G., 2018. Pixel2mesh: Generating 3d mesh models from single rgb images. *Proceedings of the European conference on computer vision (ECCV)*, 52–67.

Wang, T.-Q. and Liu, C.-L., 2018. Fully convolutional network based skeletonization for handwritten chinese characters. *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wang, Y., Li, Q., Liu, L., Zhou, Z., Ruan, Z., Kong, L., Li, Y., Wang, Y., Zhong, N., Chai, R. et al., 2019a. Teravr empowers precise reconstruction of complete 3-d neuronal morphology in the whole brain. *Nature communications*, 10 (1), 1–9.

Wang, Y., Narayanaswamy, A., Tsai, C.-L. and Roysam, B., 2011. A broadly applicable 3-d neuron tracing method based on open-curve snake. *Neuroinformatics*, 9 (2), 193–217.

Wang, Y., Xie, P., Gong, H., Zhou, Z., Kuang, X., Wang, Y., Li, A.-a., Li, Y., Liu, L., Veldman, M. B., Daigle, T. L., Hirokawa, K. E., Qu, L., Lesnar, P., Jiang, S., Yu, Y., Wakeman, W., Zeng, S., Li, X., Yuan, J., Nguyen, T. N., Larsen, R., Kebede, S., Song, Y., Yin, L., Zhao, S., Feiner, A., Shen, E., Hill, C., Wang, Q., Mok, S., Sunkin, S. M., Josh Huang, Z., Esposito, L., Yao, Z., Hawrylycz, M. J., Tasic, B., Ng, L., Sorensen, S. A., William Yang, X., Harris, J. A., Koch, C., Luo, Q., Peng, H. and Zeng, H., 2019b. Complete single neuron reconstruction reveals morphological diversity in molecularly defined claustral and cortical neuron types. *bioRxiv*. URL `https://www.biorxiv.org/content/early/2019/06/20/675280`.

Wang, Y., Xu, Y., Tsogkas, S., Bai, X., Dickinson, S. and Siddiqi, K., 2019c. Deepflux for skeletons in the wild. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5287–5296.

Watson, P. and Gupta, K., 1997. Em-ann modelling and optimal chamfering of 90/spl deg/cpw bends with air-bridges. *1997 IEEE MTT-S International Microwave Symposium Digest*, IEEE, volume 3, 1603–1606.

Weaver, C. M., Hof, P. R., Wearne, S. L. and Lindquist, W. B., 2004. Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural computation*, 16 (7), 1353–1383.

Xiao, H. and Peng, H., 2013. App2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29 (11), 1448–1454.

Xiao, Y., Cai, Z. and Yuan, X., 2019. An improved skeleton extraction method via multi-task and variable coefficient loss function in natural images. *IEEE Access*, 7, 171272–171284.

Yang, J., Zhang, Y., and Yu and Zhong, N., 2021. Nested u-net architecture based image segmentation for 3d neuron reconstruction. *Journal of Medical Imaging and Health Informatics*, 11 (5), 1348–1356.

Yuan, X., Trachtenberg, J. T., Potter, S. M. and Roysam, B., 2009. Mdl constrained 3-d grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7 (4), 213.

Zeiler, M. D. and Fergus, R., 2014. Visualizing and understanding convolutional networks. *European conference on computer vision*, Springer, 818–833.

Zeng, T., Li, R., Mukkamala, R., Ye, J. and Ji, S., 2015. Deep convolutional neural networks for annotating gene expression patterns in the mouse brain. *BMC bioinformatics*, 16 (1), 1–10.

Zhang, J., Zong, C. et al., 2015. Deep neural networks in machine translation: An overview. *IEEE Intell. Syst.*, 30 (5), 16–25.

Zhang, Y., Zhou, X., Degterev, A., Lipinski, M., Adjeroh, D., Yuan, J. and Wong, S. T., 2007a. Automated neurite extraction using dynamic programming for high-throughput screening of neuron-based assays. *Neuroimage*, 35 (4), 1502–1515.

Zhang, Y., Zhou, X., Degterev, A., Lipinski, M., Adjeroh, D., Yuan, J. and Wong, S. T., 2007b. A novel tracing algorithm for high throughput imaging: Screening of neuron-based assays. *Journal of neuroscience methods*, 160 (1), 149–162.

Zhao, J., Chen, X., Xiong, Z., Liu, D., Zeng, J., Xie, C., Zhang, Y., Zha, Z.-J., Bi, G. and Wu, F., 2020. Neuronal population reconstruction from ultra-scale optical microscopy images via progressive learning. *IEEE Transactions on Medical Imaging*, 39 (12), 4034–4046.

Zhao, J., Chen, X., Xiong, Z., Liu, D., Zeng, J., Zhang, Y., Zha, Z.-J., Bi, G. and Wu, F., 2019. Progressive learning for neuronal population reconstruction from optical microscopy images. *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, Berlin, Heidelberg: Springer-Verlag, 750–759. URL https://doi.org/10.1007/978-3-030-32239-7_83.

Zhao, K., Shen, W., Gao, S., Li, D. and Cheng, M.-M., 2018. Hi-fi: Hierarchical feature integration for skeleton detection. *arXiv preprint arXiv:1801.01849*.

Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., Long, F. and Myers, E., 2011. Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9 (2-3), 247–261.

Zhou, Z., Kuo, H.-C., Peng, H. and Long, F., 2018a. Deepneuron: an open deep learning toolbox for neuron tracing. *Brain informatics*, 5 (2), 3.

Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N. and Liang, J., 2018b. Unet++: A nested u-net architecture for medical image segmentation. *Deep learning in medical image analysis and multimodal learning for clinical decision support*, Springer, 3–11.