January 01, 2009

Automated detection and time lapse analysis of dendritic spines in laser scanning microscopy images

Jie Cheng
Northeastern University

Automated detection and time lapse analysis of dendritic spines in laser scanning microscopy images

A Thesis Presented

by

Jie Cheng

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

in Electrical Engineering

in the field of

Communication and digital signal processing

Northeastern University

Boston, Massachusetts

April 2009

Abstract

The branches extending from the cell body of neurons, the dendrites, receive more than 90% of the synaptic contacts made into that neuron. In many neurons of the mammalian brain, excitatory synapses involve specialized structures called dendritic spines that protrude from the dendrites and contain the molecules and organelles involved in the postsynaptic processing of the synaptic information. Neuron morphology, as captured in part by the structure of these spines, is illustrative of neuronal function and can be instrumental in better understanding the dysfunction seen in neurodegenerative conditions such as Alzheimer's and Parkinson's disease. Hence researchers have shown great interest in quantitatively studying dendritic spine morphology and density both statically and as a function of time. Such studies are typically carried out through the analysis of data collected from a range of microscopy modalities including confocal laser scanning microscopy (CLSM) and two-photon laser scanning microscopy (2PLSM).

Due to the size and complexity of these data sets, manually analyzing the morphological changes of dendritic spines is very time consuming. In the thesis, we describe robust, automated approaches for dendritic spine detection and measurement that are especially suitable to the batch processing of large neuronal image data sets. Our work is roughly divided into three related components. First, we focus on an image processing pipeline we have developed for the neuroinformatics processing system released from our lab called Neuron Image Quantitator (NeuronIQ), an integrated

system for automatic dendrite spine detection, quantification, and analysis. Second, to further improve detection results and solve a collection of related "hard problems" (such as disconnected spine segmentation) faced by existing automatic or semi-automatic methods, a post-processing segmentation algorithm based on a Maximum a Posteriori - orientated Markov random field (MAP-OMRF) is discussed in detail. Finally, we will present an efficient particle filter-based algorithm that is capable of tracking morphological changes in the spines over time. Possible future topics will be discussed at the end of the thesis.

Acknowledgements

First of all, I am grateful to my wife and my parents for always being there when I needed them, when I was going through some hard time during my Ph.D. study. I can hardly come to this point without their full supports.

My time in Northeastern University has been an unforgettable experience in my life. I thank my academic advisor Prof. Eric Miller, for his guidance along the way of my academic pursuit here, for his generous help in obtaining my scholarship, for his great efforts in the whole process of writing this thesis. I have learned a lot from his serious scholarly research attitude. Also many thanks to other thesis committee members Prof. Dana Brooks and Prof. Jennifer Dy. I give special thanks to Prof. Stephen Wong and Prof. Xiaobo Zhou, who have been constantly helping and supporting me for the past three years when I did my research in Professor Wong's lab. This thesis is completed with the support of NIH R01 LM009161 and funds from Harvard Center for Neurodegeneration and Repair (now Harvard Neurodiscovery Center).

I also would like to thank our collaborators: Bernardo Sabatini, Veronica Alvarez and Michelle Witt in Harvard Medical School. They provided me with great help in acquiring the data and validating the results and also offered me many valuable suggestions. Last but not least, I thank my former and current lab mates in the bioimaging analysis group of CBI: Xiaoyin Xu, Yong Zhang, Hai Fu and Zheng Yin. It has been a pleasure to work with them. Many ideas in this thesis actually came up after the active discussions with them.

Contents

Chapter 1 Introduction

1.1 Biological problems

Neurons are typically composed of a soma, a dendritic tree and an axon [1]. The soma is the central part of the neuron, which contains the nucleus of the cell. The dendrites are cellular extensions of a neuron with many branches, where more than 90% of input to the neuron occurs. Axons are the primary transmission lines of the neurons. Typical axons are usually about 1μm across, but may be up to several feet in length. Electrical stimulation is transmitted through an axon of upstream neurons onto dendrites via synapses (Figure 1.1).

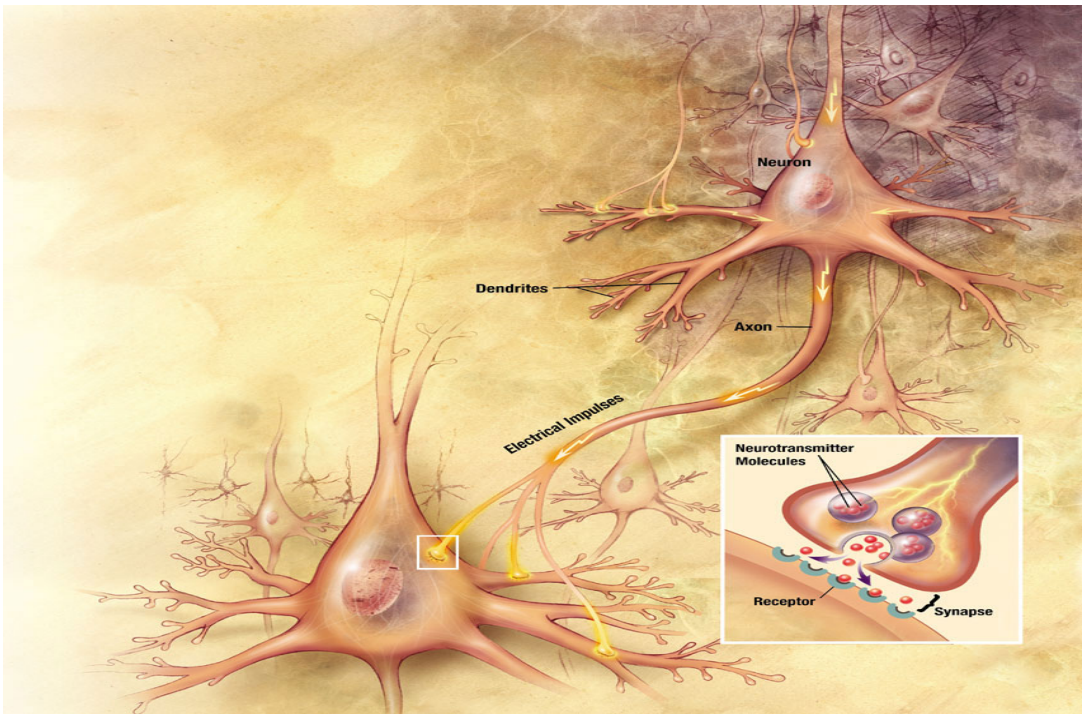


Figure 1.1: A typical neuron cell and how signals propagate down an axon to the cell body and dendrites of the next cell.[1] Soma is the large blob in the center of neuron which connects all dendrites and axons.

[1] The original image is listed at http://en.wikipedia.org/wiki/Neuron

The dendritic spine is a small (sub-micrometer) membranous extrusion of the dendrites that contains the molecules and organelles involved in the postsynaptic processing of the synaptic information. Typically dendritic spines have a bulbous head which is connected to the parent dendrite through a thin spine neck. Thus, physically the spines are connected to the dendrite although some spines seem 'detached' from the dendrite because of the small size of a spine and the low resolution in the z-direction in the 3D neuron image stacks, which can be observed in Figure (1.2). Dendritic spines possess a variety of shapes and can be categorized as different types, such as mushroom spines, thin spines and stubby spines, etc. However, the classification of spines is not strict. Electron microscopy studies have shown that there is a continuum of shapes between these categories. Time-lapse studies in the brains of living animals have shown that spines come and go, with the larger mushroom spines being the most stable over time.

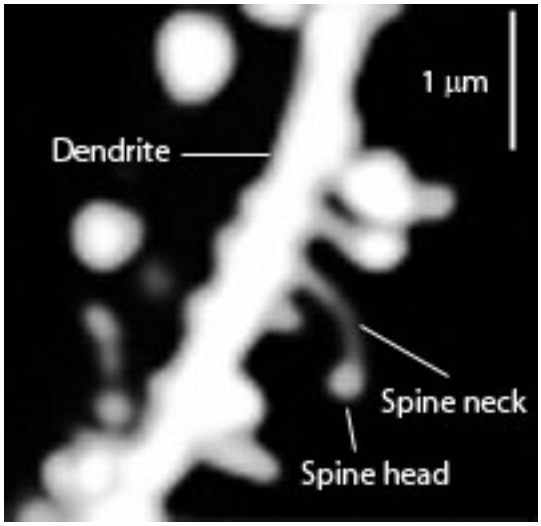


Fig. 1.2: Dendritic spines in two photon laser scanning microscopy[2]

[2] The original image is listed at http://en.wikipedia.org/wiki/Dendritic_spine

There is some evidence that differently shaped spines reflect different developmental stages of a synapse. It has been shown that the volume of spines can change depending on the types of stimuli that are presented to a synapse [11]. The remarkable capacity of dendritic spines to change shape rapidly, viz. the spine plasticity, is implicated in motivation, learning, and memory [12]-[22]. In particular, long-term memory is believed in part related to the growth of new dendritic spines or the enlargement of pre-existing spines, which reinforces a particular neural pathway. By strengthening the connection between two neurons, the pre-synaptic cell can more efficiently activate the post-synaptic cell.

The abnormalities in dendritic spine morphologies are believed to be associated with a variety of brain disorders. In particularly, neuron morphology is illustrative of neuronal function and can be instructive in the dysfunction seen in neurodegenerative conditions such as Alzheimer's disease and Parkinson's disease [23]-[33]. Cognitive disorders such as autism, mental retardation and fragile X Syndrome [34]-[39], may also be resultant from abnormalities in dendritic spines, such as the number of spines and their maturity status. For example, immature spines have impaired synaptic signaling compared with matured spines. Fragile X Syndrome is characterized by an overabundance of immature spines in cortical dendrites.

Although biologists have shown great interest in quantitatively studying dendritic spine morphology, the analysis of neuronal microscope images has remained largely

manual. Even with the computer assistance, such analyses are still extremely time-consuming, and subject to investigator bias, i.e., results cannot be easily confirmed by other investigators. Thus, automation is in great need for quantitatively analyzing dendritic spines. Automation is especially important for the analysis of the innumerable time-lapse images acquired to study the morphological changes of individual spines and the dynamics that underlie spine plasticity. The need becomes more and more urgent after the scope and practicality of time-lapse microcopies have been enormously extended with the development of high-sensitivity charge-coupled device (CCD) video cameras and automated image capture and analysis facilities.

1.2 Background

1.2.1 Neuron imaging techniques

Dendritic spines are very small membranous protrusion from a neuron's dendrite with spine head volumes ranging 0.01 μm^3 to 0.8 μm^3. Observing these tiny structures in vitro[3] or in vivo[4] was made possible only after modern fluorescence microscopes, such as confocal laser scanning microscopy (CLSM) and two-photon laser scanning microscopy (2PLSM) were introduced. These commonly used neuron imaging techniques will be briefly discussed in this section. We start by introducing the fluorescence technique which is closely related to these modern microscopes.

[3] A technique of performing given procedures in a controlled environment outside of a living organism
[4] A technique of performing given procedures in or on the living tissue of a whole living organism, which is opposed to doing experiments on a partial or dead one or in a controlled environment

1.2.1.1 Fluorescence microscope

Fluorescence is a luminescence that is mostly found as an optical phenomenon in cold bodies. Fluorescence occurs when a molecule relaxes to its ground state after the molecular absorption of a photon, which triggers the emission of another photon with a longer wavelength [2]. Usually the absorbed photon is in the ultraviolet range, and the emitted light is in the visible range. Fluorescence microscope [3] is a light microscope used to study properties of organic or inorganic substances using the phenomena of fluorescence.

Fluorescence microscopy of tissues, cells or subcellular structures is accomplished by labeling an antibody with a fluorophore (a fluorescent chemical group) through a simple chemical reaction [46]. The antibody then tags the molecules by finding its target antigen within the sample. The specimen is illuminated with light of a specific wavelength which is absorbed by the fluorophores, causing them to emit longer wavelengths of light with different colors. The illumination light is separated from the much weaker emitted fluorescence through the use of an emission filter.

1.2.1.2 Confocal laser scanning microscope

In a conventional wide-field microscope, thick specimens will produce an image that represents the sum of sharp image details from the in-focus region combined with blurred images from the neighboring out of focus regions. Although this effect does not significantly deteriorate images at low magnification (10x and below), it will obviously

degrade the images acquired with high magnification objectives. Specimens having a thickness greater than three to five microns will produce images in which out-of-focus fluorescence tends to obscure details in the actual image plane. Contrast of the in-focus image will be reduced because of the effect of a blurred background.

To address this issue, the easiest solution is to modify the specimen by slicing it into very thin sections, but it will not work for living cells or tissue sections in culture. Another solution is to modify the microscopy techniques. Confocal laser scanning microscopy (CLSM) is such a technique which helps to obtain high-resolution in-focus optical images of thick specimens through a process known as optical sectioning.

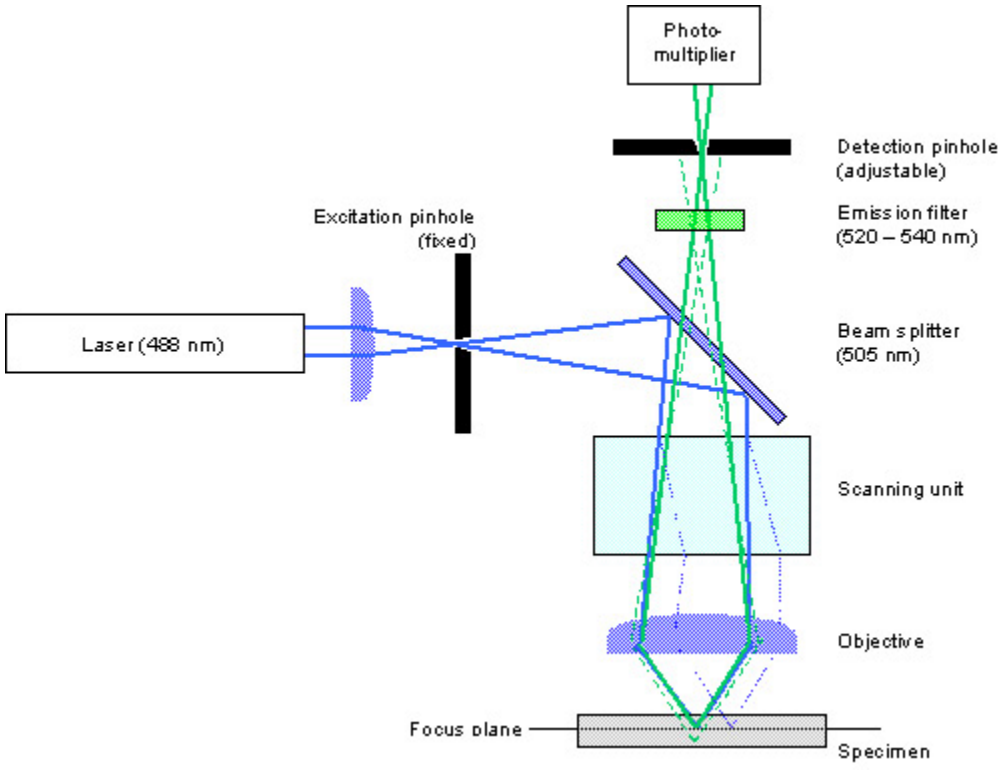


Fig. 1.3: Light path and image formation in a CLSM[5]

[5] The original image is listed at http://www.staff.kvl.dk/~als/confocal.htm

Several techniques are implemented to make CLSM able to offer observation of thin optical sections in thick specimens [4]. The light source is a laser that produces high-intensity, coherent light of a defined wavelength. Between laser and beam splitter, where the light is mirrored into the objective, a pinhole produces a sufficiently thin laser beam, which is later focused by objective projects into a focal volume within a fluorescent specimen. Since the emission pinhole is in a confocal position to the excitation pinhole, only the excited fluorescence from that point can be detected by the photomultiplier. Moreover, a special aperture in front of the photomultiplier is added to exclude any remaining out-of-focus fluorescence. With all the above mentioned techniques, out-of-focus fluorescence becomes a minor problem for CLSM imaging.

1.2.1.3 Two-photon laser scan microscopy (2PLSM).

In 1990, Denk et al introduced a two-photon excitation method into laser scanning fluorescence microscopy that allows imaging living tissue up to a depth of one millimeter [5]. The concept of two-photon excitation is based on the idea that two photons of low energy can excite a fluorophore in a quantum event, resulting in the emission of a fluorescence photon with a higher energy than either of the two excitatory photons [6][7]. Since the fluorescence intensity varies quadratically with the excitation intensity, the fluorescence intensity falls off sharply with distance from the focal point, resulting in a high degree of rejection of out-of-focus objects.

The probability of the near-simultaneous absorption of two photons is extremely low [7]. Therefore a high flux of excitation photons, which are usually provided by a

femtosecond laser, is typically required. The fluorescence from the sample is collected by a high-sensitivity detector, such as a photomultiplier tube. The observed light intensity becomes one pixel in the eventual image. The whole image is acquired by scanning the focal point throughout a desired region of the sample.

A significant advantage of multiphoton microscopy is the reduction of photobleaching of the sample [5]-[6]. Photobleaching is the photochemical destruction of a fluorophore which is a problem for fluoscence microscopes, since fluorescent molecules will eventually be destroyed by the light exposure necessary to stimulate them into fluorescing. The problem is more serious in time-lapse microscopy because of the long time exposure. Photobleaching of the entire specimen is actually the limiting factor in a CLSM: although the detected fluorescence in a confocal microscope originates at the focal point, the entire volume in the cone of illumination is illuminated by the excitation laser beam. However, it is possible for a multiphoton scanning microscope system to limit the excitation to the focal volume [5]-[6][6], which will restrict the photobleaching to the vicinity of the focal point. Thus 2PLSM is especially suitable for time-lapse imaging.

[6] This is because those multiphoton systems can use excitation laser with lower energy and longer wavelengths, which scatter less as they pass through tissue. In addition, since the light does not need to travel through the scanning mirrors or confocal aperture before reaching the detector , the detector can be located as close to the objective as possible for maximum efficiency.

1.2.2 Image acquisition

All the images processed in this thesis are obtained by our collaborator: Bernardo Sabatini's lab[7]. Brain slices from the hippocampus were prepared from rat pups (P7) and cultured as described in Alvarez et al [8]. Slices were transfected with a green fluorescent protein (GFP) expressing vector and pyramidal neurons which were identified based on their characteristic morphology at 7-20 days post-transfection (DPT). Neuronal morphology was studied using 2-photon laser scanning microscopy (2PLSM) and a custom-built microscope [9] with a water immersion objective (Olympus LUMPlanFI/RI 60x, NA=0.9). The excitation wavelength was 910 nm provided by a Verdi 10-V.-Mirra laser (Coherent). Measurements performed on 100 nm diameter yellow-green fluorescent microspheres (FluorSpheres, Molecular Probes) indicated that the point-spread function of the microscope placed a lower limit on measurable widths of 550 nm.

Images (512 x 512 pixels) of apical and basal dendrites[8] of hippocampal pyramidal neurons were acquired at zoom 3 and 5 (image field, 64x64 μm and 42x42 μm, respectively). The 3D image stacks were 16-bit grey-scale with 1 μm optical section spacing. The analyzed dataset included a variety of genotypes to ascertain how well our algorithm detected spines with a wide distribution of morphologies. The manual analysis of spine density, length, and width were measured using custom software [10] by observers who were blind to the genotype. Spine lengths were measured from the

[7] http://www.hms.harvard.edu/dms/neuroscience/fac/sabatini.html
[8] An apical dendrite and a basal dendrite are dendrites emerge from the apex or base of a pyramidal cell respectively.

junction with the dendritic shaft to the tip. To determine head width and primary dendrite thickness, the fluorescence was measured in a line across each structure and the width of the distribution where fluorescent intensity fell to 30% of maximum was calculated.

The photomultiplier (PMT) plays an important role during image acquisition with a CLSM. The intensity of the images is controlled by increasing or decreasing the voltage of the photomultiplier. It is also possible to change the offset of the PMT, which can soften or harden the contrast in an image. By adjusting the voltage and offset of a PMT, we can produce an image with a black background and sufficiently outstanding fluorescence. Sometimes if the fluorescence is weak, the photomultiplier has to work at its limits. This will make the background spotted with white pixels caused by spontaneous electron delivered by thermal processes in the PMT. Improvement of the image quality in this case is by “averaging” several images. According to our experiment, median filter has a very good performance to remove this kind of noise.

1.3 Objectives

Here we are interested in designing a pipeline for automatic neuron image processing which will release the biologists from heavy manual burden for quantitative analysis of dendritic spines. The approaches should be automatic and able to deal with problems in various fields of image processing, such as denoising, segmentation, detection, and tracking (registration). They are supposed to be based on sound mathematical models which are able to describe the essential aspects of neuron images

with reasonable simplification. Most of the proposed algorithms are built based on the Bayesian models with combining various prior knowledge and information of observed data, from which optimal solutions are found. Last but not least, the algorithms should be able to deal with application problems in the real world and be validated by comparing with the manual results of different images acquired under various conditions.

Recently, some automatic dendritic spines analysis algorithms have been proposed [40]-[44]. However, problems still exist and improvements are needed (please refer to Section 3.1 for the detailed discussion of those algorithms). Compared with the existing algorithms, we expect the solutions proposed in this thesis to:

1. reduce the missing spines and the false positives;
2. segment the spines with much higher accuracy, viz. the shapes of the segmented spines are more similar to their actual shapes;
3. measure the spines more consistently, viz. less human interventions which might induce the bias;
4. be robust to different imaging conditions and resolutions;
5. be able to process the time-lapse images, viz. track the morphological changes of dendritic spines in time sequence.

1.4 Contribution

The algorithms proposed in this thesis can be roughly divided into three parts. In Chapter 3, we focus on the algorithms used in a neuroinformatics system called neuron

image quantitator (NeuronIQ)[9], an integrated data processing pipeline for automatic dendrite spine detection and quantitative analysis of spine morphometry [45]. The main purpose of NeuronIQ is to provide biologists an automated tool which is well suitable for batch processing a large dataset of image with little human interference. In Section 3.2, we introduce an automatic adaptive segmentation algorithm. In Section 3.3, the backbone extraction algorithm is presented. With the information obtained above, the detached and attached spine components are detected separately by using different approaches. The detached spine components are found by local signal to noise ratio (SNR) analysis, and the attached spine components are detected using geometric properties of dendrites. The spine detection algorithms are discussed in Section 3.4. Certain post-processing algorithms, such as combination of broken spine components and measurement of spine density, are described in Section 3.5. In Section 3.6 the results show that NeuronIQ has obviously improved compared with other existing methods.

In order to further improve the detection results and solve some common problems (e.g. distorted segmentation and obviously degraded detection performance for images with lower resolution) faced by existing automatic or semi-automatic methods, a post-processing algorithm based on a maximum a posteriori - orientated Markov random field (MAP - OMRF) framework is proposed in Chapter 4, which improves the detection results for images acquired at various resolutions. In Section 4.2, the statistical model of OMRF will be discussed in detail. It is shown that the segmentation and detection

[9] NeuronIQ is a tool developed by the bioimaging analysis group of CBI in Radiology Department at Methodist Hopital. The latest version of NeruonIQ can be downloaded at: http://www.methodisthealth.com/tmhs/basic-right.do?channelId=-90590&contentId=191151&contentType=GENERIC_CONTENT_TYPE

results can be improved by combining the orientation information of both dendrite and spines, as well as the local intensity distribution information. To further increase the speed of processing and make the detection more robust, a region of interest (ROI) estimation algorithm based on iterative spine background correction is also presented. Obtaining the optimal solution will be discussed in Section 4.3. A knowledge-guided iterative conditional mode (KICM) algorithm is described, which is time efficient and can obtain better detection results for microscope neuron images compared with normal ICM algorithm.

In Chapter 5 we will mainly focus on discussing the time lapse image processing. A revised iterative closest point (ICP) algorithm with feature selection for global registration is presented first in Section 5.2. Compared with the normal ICP algorithm, the registration of dendrites can be obviously improved. Then, the particle filter based tracking algorithm is described in detail, which can accurately track and measure the dendritic spines simultaneously.

In Chapter 6 we will first give a brief summary about the relationship of the proposed algorithms. Then some unsolved problems are discussed in detail, which are possible future topics based on the work presented in Chapter 3-5.

References

[1] L.R. Squire, Fundamental Neuroscience, Third Edition, published by Elsevier (2008)

[2] B. Valeur, Molecular Fluorescence: Principles and Applications, published by Wiley-VCH (2001)

[3] J.R. Lakowicz, Principles of Fluorescence Spectroscopy, Third Edition, Plenum Press, New York (2006)

[4] J.B. Pawley, Handbook of Biological Confocal Microscopy, Third Edition, Berlin: Springer (2006)

[5] W. Denk, J.H. Strickler, W.W. Webb, Two-photon laser scanning fluorescence microscopy, Science, 248(4951): 73-76 (1990)

[6] J. D. Bhawalkar, A. Shih, S. J. Pan, W. S. Liou, J. Swiatkiewicz, B. A. Reinhardt et al, Two-photon laser scanning fluorescence microscopy – from a fluorophore and specimen perspective, Bioimaging, Special issue: Nonlinear optical microscopy, 4(3): 168-178 (2001)

[7] Göppert-Mayer M, Über Elementarakte mit zwei Quantensprüngen, Ann Phys. 9: 273–95 (1931)

[8] A.V. Veronica, A.R. Dennis, B.L. Sabatini, Retraction of synapses and dendritic spines induced by off-target effects of RNA Interference, J. of Neurosci., 26(30): 7820-25 (2006)

[9] A.G. Carter, B.L. Sabatini, State dependent calcium signaling in dendritic spines of striatal medium spiny neurons, Neuron 44(3): 483-93 (2004)

[10] S.F. Tavazoie, V.A. Alvarez, D.A. Ridenour, D.J. Kwiatkowski and B.L. Sabatini, Regulation of neuronal morphology and function by the tumor suppressors Tsc1 and Tsc2. Nature Neurosci., 8: 1727-34 (2005)

[11] E.A. Nimchinsky, B.L. Sabatini, K. Svoboda, Structure and function of dendritic spines, Annu. Rev. Physiol. 64:313–53 (2002)

[12] R. Yuste, T. Bonhoeffer, Morphological changes in dendritic spines associated with long-term synaptic plasticity, Annu. Rev. Neurosci., 24:1071–89 (2001)

[13] P. Andersen, A. Soleng A, Long-term potentiation and spatial training are both associated with the generation of new excitatory synapses, Brain Res. Rev. 26:353–59 (1998)

[14] V.Y. Bolshakov, S.A. Siegelbaum SA, Regulation of hippocampal transmitter release during development and Long-Term Potentiation, Science 269:1730–34 (1995)

[15] C. Boyer, T. Schikorski, C.F. Stevens, Comparison of hippocampal dendritic spines in culture and in brain. J. Neurosci. 18:5294–300 (1998)

[16] F. Engert, T. Bonhoeffer, Dendritic spine changes associated with hippocampal longterm synaptic plasticity, Nature 399:66–70 (1999)

[17] M. Fischer, S. Kaech, D. Knutti, A. Matus, Rapid actin-based plasticity in dendritic spine, Neuron 20:847–54 (1998)

[18] B. Lendvai, E. Stern, B. Chen, K. Svoboda, Experience-dependent plasticity of dendritic spines in the developing rat barrel cortex in vivo. Nature 404:876–81 (2000)

[19] M.B. Moser, M. Trommald, P. Andersen, An increase in dendritic spine density on hippocampal CA1 pyramidal cells following spatial learning in adult rats suggests the formation of new synapses, Proc. Natl. Acad. Sci. 91:12673–75 (1994)

[20] R. Yuste, A. Majewska, K. Holthoff, From form to function: calcium compartmentalization in dendritic spines, Nat. Neurosci. 3:653–59 (2000)

[21] M. Masanori, Factors critical for the plasticity of dendritic spines and memory storage, Neurosci. Res. 57(1): 1-9

[22] K.J. Lee, J.G. Jung, T. Arii, K. Imoto, I.J. Rhyu, Morphological changes in dendritic spines of Purkinje cells associated with motor learning, Neurobiol Learn Mem., 88(4):445-50 (2007)

[23] B.T. Human, Molecular and anatomical studies in Alzheimer's disease, Neurologia. 16(3): 100-04 (2001)

[24] A.V. Veronica, A.R. Dennis, B.L. Sabatini, Retraction of synapses and dendritic spines induced by off-target effects of RNA Interference, J. of Neurosci., 26(30): 7820-25 (2006)

[25] S. Knafo et al, Widespread changes in dendritic spines in a model of Alzheimer's disease, Cerebral Cortex, 19(3):586-592 (2009)

[26] P. Mehraein, M. Yamada, T. Dziduszko, Quantitative study on dendrites and dendritic spines in Alzheimer's disease and senile dementia, Adv Neurol. 12:453-8 (1975)

[27] M. Knobloch, I.M. Mansuy, Dendritic spine loss and synaptic alterations in Alzheimer's disease, Molecular Neurobiology, 37(1):73-82 (2008)

[29] I. Wasowicz et al, Quantitative age-related changes in apical dendrites and dendritic spines of CA1 pyramidal neurons among senescence accelerated mice (SAMP1TA/Ngs), Mechan. of Ageing and Develop., 90(1): 63-73 (1996)

[30] D.L. Moolman, O.V. Vitolo, J.P. Vonsattel, M.L. Shelanski, Dendrite and dendritic spine alterations in Alzheimer models. J Neurocytol. 33:377–387 (2004)

[31] T.L. Spires-Jones, M. Meyer-Luehmann, J.D. Osetek, P.B. Jones, E.A. Stern, B.J. Bacskai, B.T. Hyman, Impaired spine stability underlies plaque-related spine loss in an Alzheimer's disease mouse model. Am J Pathol. 171:1304–1311 (2007)

[32] C. R. Gerfen, Indirect-pathway neurons lose their spines in Parkinson disease, Nature Neurosci. 9: 157-158 (2006)

[33] P. Anglade, A. Mouatt-Prigent, Y. Agid, E. C. Hirsch, Synaptic plasticity in the caudate nucleus of patients with Parkinson's disease, Neurodegeneration, 5(2): 121-128 (1996)

[34] A. Y. Hung et al, Smaller Dendritic Spines, Weaker Synaptic Transmission, but Enhanced Spatial Learning in Mice Lacking Shank1, J. of Neurosci., 28(7):1697-1708 (2008)

[35] C.M. Durand, Mutations in the gene encoding the synaptic scaffolding protein SHANK3 are associated with autism spectrum disorders. Nat Genet 39:25–27 (2007)

[36] T.V. Bilousova, L. Dansie, M. Ngo, J. Aye, J.R. Charles, D.W. Ethell, I.M. Ethell, Minocycline promotes dendritic spine maturation and improves behavioural performance in the fragile X mouse model, J. of Med. Genetics 46:94-102 (2009)

[37] S.A. Irwin, R. Galvez, W.T. Greenough, Dendritic spine structural anomalies in fragile-X mental retardation syndrome, Cereb Cortex. 10(10):1038-44 (2000)

[38] R. Galvez, W.T. Greenough, Sequence of abnormal dendritic spine development in primary somatosensory cortex of a mouse model of the fragile X mental retardation syndrome, Amer. j. of medical genetics, 135(2): 155-160 (2005)

[39] T.A. Comery, Abnormal dendritic spines in fragile X knockout mice: Maturation and pruning deficits, Proc. Natl. Acad. Sci. 94: 5401–5404 (1997)

[40] I. Y. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, and K. Svoboda, An image analysis algorithm for dendritic spines, Neur. Comput. 14: 1283–310 (2002)

[41] X. Xu, J. Cheng, R. M. Witt, B. L. Sabatini, S. T.C. Wong, A shape analysis method to detect dendritic spine in 3D optical microscopy image, Biomedal Imaging: Macro to Nano,. 3rd IEEE International Symposium, Apr. 6, pp554 – 557 (2006)

[42] C. M. Weaver, P. R. Hof, S. L. Wearne, W. Lindquist, Automated algorithms for multiscale morphometry of neuronal dendrites, Neur. Comput. 16: 1353-83 (2004)

[43] W. Bai, X. Zhou, L. Ji, J. Cheng, and S. T. Wong, Automatic dendritic spine analysis in twophoton laser scanning microscopy images, Cytometry A 71: 818-826 (2007)

[44] Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjeroh, and S. T. Wong, Dendritic spine detection using curvilinear structure detector and LDA classifier. Neuroimag. 36: 346–360 (2007)

[45] J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, and S. T. Wong, A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *J. of Neurosci. Methods* 165: 122-134 (2007)

[46] K.R. Spring, M.W. Davidson, Introduction to fluorescence microscopy, http://www.microscopyu.com/articles/fluorescence/fluorescenceintro.html

Chapter 2: Technical background

Segmentation plays a very important role in the automatic spine detection. Good segmentation robust to different imaging conditions can greatly reduce the difficulty of the post-processing (combining broken pieces, registration, measurement, etc.) and improve the efficiency of the whole algorithm. In the first part of this chapter, we will briefly survey the segmentation methods being used in medical image processing.

One effective approach to automatically detect and measure the morphological changes of the spines is to track the spines in time-lapse imagery. In the second part of this chapter, we will discuss the tracking algorithms with a focus on the state space based Kalman filter and particle filter.

2.1 Segmentation

Medical image segmentation is a process to label each voxel in a medical image based on its tissue type or anatomical structure [70]. The structures of interest include the organs, bones, vessels, or abnormalities such as tumors and cysts [2]-[6]. The basic purpose of medical image segmentation is to assist doctors in evaluating the medical imagery. In this section we will first review some popular segmentation methods proposed in the literature, and then we will focus our discussion on the model based image segmentation methods.

2.1.1 Thresholding

Thresholding is one of the simplest methods of image segmentation. The segmentation is achieved by grouping pixels into different classes based on the preset intensity (or other quantifiable features) values - thresholds. Thresholding is an effective segmentation method if different structures in the image have obvious different intensity distributions. There can be one threshold (binary partitioning), or several thresholds (multiple thresholding). The threshold can be global (global thresholding), or decided locally (adaptive thresholding). A survey on thresholding techniques can be found in [1].

Although they have been successfully applied in medical image processing [2]-[6], there is a major drawback of threshold-based methods: they are sensitive to noise and intensity inhomogeneities because such methods are unable to take into account the spatial information of an image. In other words, only based on the intensity, they often lack the sensitivity (the rate of correct detection) and specificity (the complement of wrong detection rate) required for accurate segmentation.

2.1.2 Clustering

Clustering algorithms are basically unsupervised classification methods, in which only unlabeled examples are given. There are three commonly used clustering algorithms: k-means algorithm, fuzzy c-means algorithm, and the expectation-maximization (EM) algorithm. The k-means algorithm [7]-[8] assigns each pixel to the cluster (a collection

of data who have the similar features) whose centroid is the nearest. The centroids are recomputed after the assignment of all pixels. The two steps are repeated until some convergence criterion is met. Despite its simplicity, one major drawback of k-means algorithm is that its performance largely depends on the initial random assignments. Besides, the global minimum of variance cannot be assured although the intro-cluster variance is minimized.

The fuzzy c-means algorithm is a generalized form of the k-means algorithm which allows the soft segmentations based on fuzzy set theory [9]-[10]: each pixel has a possibility of belonging to all clusters, rather than belonging exclusively to only one cluster. The probability of belonging is reciprocal to the pixel's distance to the cluster centroid. Like the k-means algorithm, fuzzy c-means algorithm also depends on the initial classification and the minimum is a local minimum.

Assuming that the data follow a Gaussian mixture model, the Expectation-Maximization (EM) algorithm [11]-[14] is a more statistically formalized method which applies the same clustering principles as the fuzzy c-means algorithm. Generally it has better convergence properties compared with fuzzy c-means algorithm. To better describe the EM algorithm, below we give a simple example of how to apply EM algorithm for brain MR image segmentation.

Suppose the intensity distributions of three types of brain tissues, i.e. white matter, gray mater and cerebrospinal fluid (CFS), are modeled as Gaussian distributions. Once

the parameters (means and variances) of all three tissue types are known, voxels can be segmented based on the intensity values. The EM algorithm can be used to find the maximum likelihood estimates (MLE) of those parameters. Let $\boldsymbol{\theta}$ be the parameter vector, $\mathbf{x}$ be the observed data and $\mathbf{z}$ be the unobserved data. In this case, $\mathbf{z}$ represent which of the three tissue types each voxel belong to.

The basic idea of EM algorithm is to alternate between estimating the expectation of the likelihood (E step) and computing the maximum likelihood estimates of the parameters (M step). The process is repeated until a preset criterion is met. The steps can be formulated as:

Expectation step: find the expectation of the log likelihood function given $\mathbf{x}$ under the current estimate of the parameters $\boldsymbol{\theta}^{(t)}$

$$Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) = E[\log L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) \mid \mathbf{x}, \boldsymbol{\theta}^{(t)}] \qquad (2.1)$$

Maximization step: find the parameters which maximize the Q function

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{(t)}) \qquad (2.2)$$

For brain MR image segmentation, the likelihood function is

$$L(\boldsymbol{\theta}; \mathbf{x}, \mathbf{z}) = \prod_{i=1}^{N} \sum_{j=1}^{3} \mathrm{I}(z_i = j) \tau_j f(x_i; \mu_j, \Sigma_j), \qquad (2.3)$$

and the parameter vector is

$$\boldsymbol{\theta} = (\tau_1, \tau_2, \tau_3, \mu_1, \mu_2, \mu_3, \Sigma_1, \Sigma_2, \Sigma_3), \qquad (2.4)$$

where N is the total number of voxels; I is an indicator function and f is the probability density function of a multivariate normal; τ_j is the prior probability of each type of tissue; μ_j and Σ_j are the mean and covariance matrix respectively.

In general, the performance of the clustering methods largely depends on the initial classification results. Furthermore, the lack of spatial modeling makes these methods sensitive to noise and intensity inhomogeneities.

2.1.3 Region growing

The simplest form of region growing method is the seeded region growing method. A set of seeds are firstly selected, then regions are iteratively grown by comparing their unallocated neighboring pixels [15]-[17]. The process continues until all pixels are allocated. To relax the requirement of the manually set seed points, split-and-merge method is proposed [18]. Seeds based region growing method is sensitive to noise which might cause the distortion of topology of the objects. To deal with this issue, a homotopic [19] and a fuzz theory based [20] region-growing algorithms are proposed.

2.1.4 Model based segmentation

Methods for medical image segmentation are often application-specific: prior knowledge for the particular objects of interest is applied. Assuming that objects of interest have a repetitive form of geometry, model based segmentation methods are

widely applied in medical image processing because of the ability of such methods to impose various constraints into the prior models. In this section, we will discuss two popular models widely applied in medical image segmentation: Markov random fields (MRF) and active contours.

2.1.4.1 Markov random fields

MRF models are widely used for image segmentation by encoding expected spatial correlations through contextual constraints of neighboring pixels. Usually, the correlation structure is such that pixels are expected to have the same or similar intensities, i.e. the image is piecewise constant or piecewise continuous.

Let $F = \{F_1,..., F_m\}$ be a family of random variables defined on the set S, a joint event is abbreviated as $F = f$ where $f = \{f_1,..., f_m\}$ is a configuration of F. F is said to be a MRF on S with respect to a neighborhood system N if and only if both the positivity

$$P(f) > 0, \forall f \in F, \tag{2.5}$$

and Markovianity

$$P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i}), \tag{2.6}$$

are satisfied. Here $S - \{i\}$ is the set difference, $f_{S-\{i\}}$ denotes the set of labels at the sites in $S - \{i\}$ and $f_{N_i} = \{f_{i'} \mid i' \in N_i\}$ stands for the set of labels at the sites neighboring i. The label at a site is dependent only on those at the neighboring sites.

The practical use of MRF models is largely ascribed to the equivalence between MRFs and Gibbs distributions [21]. A set of random variables F is said to be a *Gibbs random field* on set S with respect to the neighborhood N if and only if its configurations obey a Gibbs distribution. A Gibbs distribution takes the form

$$P(f) = \frac{1}{Z} e^{-\frac{1}{T}U(f)}, \qquad (2.7)$$

where $Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)}$ is a normalizing constant called the partition function, T is a constant called the temperature which is usually set as 1. $U(f)$ is the energy function. The energy $U(f) = \sum_{c \in C} V_c(f)$ is a sum of clique potentials $V_c(f)$ over all possible cliques C. The value of $V_c(f)$ depends on the local configuration on the clique c.

A clique for graph (V, E) is defined as a subset of sites in V, e.g., pair-site cliques are the subset of groups of two neighboring pixels. Here V contains the nodes, E determines the links between the notes according to the neighboring relationship. One commonly used MRF model is called auto-models, in which Gibbs energy consists of only single-site and pair-site clique potentials. The energy can be described as

$$U(f) = \sum_{\{i\} \in C_1} \alpha_i G_1(f_i) + \sum_{\{i,i'\} \in C_2} \beta_{i,i'} G_2(f_i, f_{i'}), \qquad (2.8)$$

where C_1 and C_2 are the collection of single-site and pair-site cliques respectively; α_i and $\beta_{i,i'}$ are the control parameters.

The MRF based segmentation methods have been widely studied for various types of medical images obtained by a wide range of modalities. These includes, but are not limited to, cardiac imaging [22]-[23], brain imaging [24]-[30], digital mammography [31]-[32] acquired by magnetic resonance imaging (MRI), ultrasound, positron emission tomography (PET), and electron microscopy.

Within each of these fields, MRFs are used to address a wide range of image processing tasks. The main purpose for brain segmentation is to separate the brain into three kinds of tissues: gray matter (GM), white matter (WM) and cerebrospinal fluid (CSF). Although the finite mixture (FM) model[10] is the most commonly used model for statistical brain segmentation, it is sensitive to noise as a histogram-based model without spatial information. Various MRF algorithms are proposed to address the issue. Ruan et al [24] proposed a MRF based algorithm with multifractal analysis, which can effectively solve the problem of partial volume effects. Zhang et al [25] proposed a novel hidden Markov random field (HMRF) model to deal with the artifacts caused by biased field distortion. It has also been shown that mathematically the FM model is a degenerate version of the HMRF model. MRF approaches have also been extensively studied to segment brain voxels into active and inactive areas for the analysis of functional magnetic resonance imaging (fMRI) data [26]-[28]. By effectively fusing contextual dependencies within functional imaging data, the segmentation is much more robust to the noisy fMRI images. For cardiac image processing, Xiao et al [22] proposed a MRF-based method to address the problem of intensity inhomogeneities dominated

[10] The method to describe a probability density function as a combination of a set of probability density functions with different weights

by nonuniform beam attenuation within the body in the ultrasound B-mode images. Juslin et al [23] successfully applied the MRF models in an automatic method to extract the heart volume from the cardiac PET images. Detection of clusters of microcalcifications in digital mammograms is critical for the early diagnosis of breast cancer. However, due to the small size of microcalcifications and the poor quality of images, it is hard to find reliable methods for an automatic detection of such clusters. To address these problems, D'Elia et al [31] proposed a tree structured Markov random field algorithm which can effectively extract the elementary homogeneous regions of interest on the image. Li et al [32] proposed a multiresolution MRF model to extract the region of interest, which can increases the speed of processing and performance of detection.

The major difficulty for MRF models is the computational complexity. Usually, the local optimization algorithms such as iterative conditional modes (ICM) are applied to find the optimal solutions [24][25][28][32]. How to properly select the control parameters such as α and β in Eq. 2.4 is another common problem. A too high setting can result in an excessively smooth segmentation and cause the loss of important structural details.

To present how MRF is applied in medical image segmentation, here we give an example of MRF approach for brain MR image segmentation [71]. Suppose vector $\mathbf{y}$ contains all individual intensity y_i for each voxel i in the MR image. The task of segmentation is to determine the vector $\mathbf{x}$ so that for each pixel $x_i \in$ (WM, GM, CSF). To model intensity inhomogeneities in MRI image, which are caused by

inhomogeneities of the magnetic fields and the sensitivity profile of the receiving coil, a smooth inhomogeneity field $\mathbf{z}$ is proposed. For any observed MR intensities $\mathbf{y}$, the *a posteriori* probability of the segmentation $\mathbf{x}$ and the inhomogeneity $\mathbf{z}$ can be obtained by applying Bayes rule

$$p(\mathbf{x},\mathbf{z}\mid\mathbf{y}) \propto p(\mathbf{y}\mid\mathbf{x},\mathbf{z})\,p(\mathbf{x})\,p(\mathbf{z}), \tag{2.9}$$

where $p(\mathbf{y}\mid\mathbf{x},\mathbf{z})$, $p(\mathbf{x})$ and $p(\mathbf{z})$ are the priori probability of the segmented image, the inhomogeneity and the conditional probability of the observation. The optimal solution which maximizes the probability can be found by simulated annealing (SA) [27][30] or iterated conditional modes (ICM) algorithms [24][25][28][32].

The priori probability $p(\mathbf{x})$ can be modeled by MRF, namely neighborhood MRF. After defining a neighborhood system of six nearest neighbors ξ in 3-D (Figure 2.1), the probability can be expressed as

$$p(\mathbf{x}) \propto e^{-E(\mathbf{x})}, \tag{2.10}$$

with the energy

$$E(\mathbf{x}) = \sum_{i=1}^{M}\sum_{j\in\xi_i} e_{x_i x_j}, \tag{2.11}$$

where M is the total number of the voxels, ξ_i is the six-neighbor of voxel i, $e_{x_i x_j}$ is potential between two voxels, which is minimal for neighboring voxels of the same tissue ($x_i = x_j$) so that neighboring voxels are intent to be segmented as the same type.

The inhomogeneity probability $p(\mathbf{z})$ can also be modeled by MRF, namely inhomogeneity MRF. After the 6-neighbor is defined, we have

$$p(\mathbf{z}) \propto e^{-U(z)}, \qquad (2.12)$$

with the energy (in which small inhomogeneity corrections are more probable than large)

$$U(\mathbf{z}) = \alpha \sum_{i=1}^{M} \sum_{j=\xi_i} (z_i - z_j)^2 + \beta \sum_{i}^{M} z_i^2, \qquad (2.13)$$

where α and β are the weight factors.

The conditional probability $p(\mathbf{y} \mid \mathbf{x}, \mathbf{z})$ can be expressed by a Parzen-window distribution [72]

$$p(\mathbf{y} \mid \mathbf{x}, \mathbf{z}) = \frac{1}{n_{x_i}} \sum_{k=1}^{n_{x_i}} \frac{1}{(2\pi)^{d/2} \mid \Sigma \mid^{1/2}} e^{-\frac{1}{2}(y_i z_i - \bar{y}_{k,x_i})^t \Sigma^{-1} (y_i z_i - \bar{y}_{k,x_i})}, \quad (2.14)$$

where $\bar{y}_{k,x_i}$ is a set of n_{x_i} training points of tissue class x_i selected by a supervisor; the covariance matrix Σ is equal to σ^2 in the case of single-echo MR image ($d = 1$) and to times the 2-D unit matrix for double-echo MR images ($d = 2$). The detailed discussion of this distribution can be found in [73].

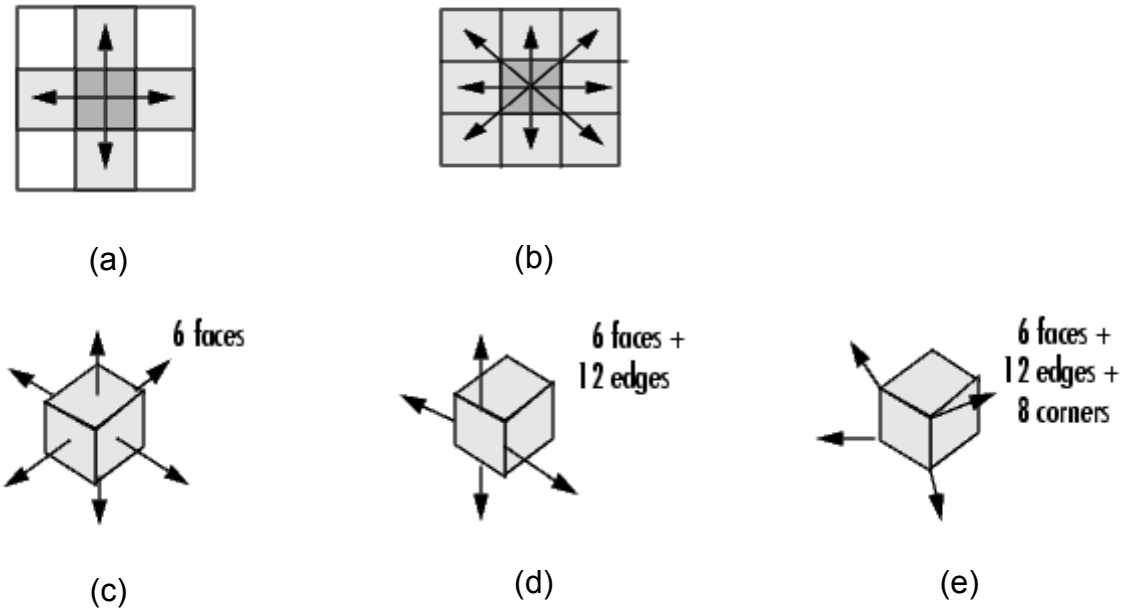


Figure 2.1 Common nearest neighbors applied in MRF: (a) 4-neighbor in 2-D (b) 8-neighbor in 2-D (c) 6-neighbor in 3-D, two voxles are neighbor if their faces touch (d) 18-neighbor in 3-D, two voxles are neighbor if their faces or edges touch (e) 26-neighbor in 3-D, two voxles are neighbor if their faces, edges or corners touch

2.1.4.2 Active contours (snakes)

Active contours are curves or surfaces defined within an image domain that can move under the influence of internal forces defined within the curve or surface, and external forces computed from the image itself [33]-[35]. With a priori knowledge about the model, internal forces hold the model together and keep it from bending too much. External forces are defined to move the model toward an object boundary or other desired features within the image. A common external force is the potential force, which is the negative gradient of a potential function. Other forces, such as pressure forces

and distance potentials, can also be applied to achieve different goals. The deformable model can be considered as a kind of boundary mapping, which constructs a mathematical representation of an object boundary from an image [36]-[39]. Deformable models have shown great success in medical image segmentation [40]-[49].

There are basically two types of deformable models: parametric deformable models and geometric deformable models. Parametric deformable models represent curves and surfaces explicitly in their parametric forms. Based on the theory of curve evolution and geometric flows, geometric deformable models represent curves and surfaces implicitly as a level set of an evolving scalar function. Their parameterizations are computed only after complete deformation, thus topological adaptiveness can easily be accommodated. Despite this fundamental difference, the underlying principles of both methods are very similar. A traditional snake is a curve given by $\mathbf{x}(s) = |\, x(s), y(s)\,|, s \in [0,1]$, that moves around in the image to minimize the energy

$$E = \int_0^1 \frac{1}{2}(\alpha \,|\, \mathbf{x}'(s)\,|^2 + \beta \,|\, \mathbf{x}''(s)\,|^2 + E_{ext}(\mathbf{x}(s))ds\,, \qquad (2.15)$$

where α and β are weighting parameters that control the deformable contour's tension and rigidity respectively [53].

The external force is written as the negative gradient of a potential function $E(x, y)$ which is derived from the image data and takes smaller values at object boundaries. For a grey-level image $I(x, y)$, typical external potential functions designed to lead to a deformable contour toward step edges are

$$E_{ext}^{(1)}(x,y) = -|\nabla I(x,y)|^2, \qquad (2.16)$$

or

$$E_{ext}^{(2)}(x,y) = -|\nabla(G_\sigma(x,y) * I(x,y))|^2, \qquad (2.17)$$

where $G_\sigma(x,y)$ is a 2D Gaussian function with standard deviation σ, and ∇ is the gradient operator. Figure 2.2 demonstrates an example of external force which will pull an active contour towards the object boundary.

Figure 2.2 The gradient vector flow (GVF) external force field for a U-shaped object [11].

The problem of finding a curve $\mathbf{x}(s)$ that minimizes the energy functional is known as a variational problem. It has been shown that the curve that minimizes E must satisfy the following Euler-Largrange equation [53]

$$\frac{\partial}{\partial s}(\alpha\frac{\partial \mathbf{x}}{\partial s}) - \frac{\partial^2}{\partial s^2}(\beta\frac{\partial^2 \mathbf{x}}{\partial s^2}) - \nabla E_{ext}(\mathbf{x}) = 0. \qquad (2.18)$$

[11] The original image is listed on http://iacl.ece.jhu.edu/projects/gvf/.

To find a solution, the snake is made dynamic by treating $\mathbf{x}(s)$ as a function of time t as well as s, i.e. $\mathbf{x}(s,t)$. The partial derivative of $\mathbf{x}$ with respect to t is then set equal to the left-hand side of above equation

$$\gamma \frac{\partial \mathbf{x}}{\partial t} - \frac{\partial}{\partial s}(\alpha \frac{\partial \mathbf{x}}{\partial s}) - \frac{\partial^2}{\partial s^2}(\beta \frac{\partial^2 \mathbf{x}}{\partial s^2}) - \nabla E_{ext}(\mathbf{x}). \qquad (2.19)$$

When the solution $\mathbf{x}(s,t)$ stabilizes, the left side vanishes and we achieve a solution. Thus, the minimization is solved by placing an initial contour on the image domain and allowing it to deform according to above equation.

One of the major drawbacks of the snake algorithms is that the performance is sensitive to the initial contour. Several methods have been proposed to make them more robust to the initialization [49]-[52]. Another problem for the standard deformable models is the poor convergence to concave boundaries. Xu et al. [53] have proposed a gradient vector flow (GVF) based algorithm which can effectively solve this problem.

2.2 Tracking

Basically there are two major approaches for a visual tracking system: Target Representation and Localization (TRL) and Filtering and Data Association (FDA) [54]. In a TRL tracking system, the objects may be segmented as blobs and then tracked by block based correlation or optical flow algorithms [55]-[56]. In some cases the object boundary is extracted and tracked by active contours or condensation algorithms [57]-[58]. Generally the objects are represented in the feature space. The localization is performed by the maximization of a similarity function of the target and the target

candidates. TRL is mostly a bottom-up process which needs to cope with the changes in the appearance of the target. The computational complexity is typically low. In contrast, FDA is mostly a top-down process, which involves incorporating prior information about the object via a prior model, defining object dynamics via a dynamic model, and evaluating different hypotheses via various optimization methods. This kind of approach is mostly built in a Bayesian framework, and the computational complexity is usually higher.

Tracking spines in time lapse image sequence can be described as a problem of estimation of the state of a system that changes over time using a sequence of noisy measurements made on the system (for dendritic spine tracking, the states describe the size and orientation information a spine.) The state-space approach to modeling dynamic systems is especially convenient for handling multivariate data and nonlinear/non-Gaussian process. In order to analyze dynamic system, the system model (dynamic model) and the measurement model (observation model) are required. The system model describes the evolution of the state with time, while the measurement model relates to the noisy measurements of the state.

Suppose the evolution of the state sequence $\{\mathbf{x}_k, k \in N\}$ is given by

$$\mathbf{x}_k = \mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1}), \qquad (2.20)$$

where $\mathbf{f}_k$ is a function which describes the change of the states; $\{\mathbf{v}_k, k \in N\}$ is an i.i.d. process noise sequence; N is the set of integers. The objective of tracking is to recursively estimate the state $\mathbf{x}_k$ based on the new observation

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k), \tag{2.21}$$

where $\mathbf{z}_k$ is a function of the state $\mathbf{x}_k$ and the i.i.d. measurement noise $\mathbf{n}_k$.

Here is a simple example of state space base tracking: suppose a car is moving along the road with a constant velocity subject to random perturbations $\mathbf{v}$ in its trajectory, the state vector can be expressed as

$$\mathbf{x}_k = [p_1(k)\ p_2(k)\ q_1(k)\ q_2(k)]^T, \tag{2.22}$$

where $[p_1(k)\ p_2(k)]$ and $[q_1(k)\ q_2(k)]$ are the actual position and velocity of the car at time k respectively, T is the transpose operator. If $\mathbf{z}_k = [z_1(k)\ z_2(k)]^{\mathrm{T}}$ is the observed position at time k and $\mathbf{n} = [n_1\ n_2]^T$ is the measurement error, equation (2.20) and (2.21) are then described as

$$\begin{bmatrix} p_1(k) \\ p_2(k) \\ q_1(k) \\ q_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_1(k-1) \\ p_2(k-1) \\ q_1(k-1) \\ q_2(k-1) \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}, \tag{2.23}$$

and

$$\begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_1(k) \\ p_2(k) \\ q_1(k) \\ q_2(k) \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}. \tag{2.24}$$

Tracking is basically composed with two steps: prediction and updating. Prediction is to estimate the prior probability of the state at time k by using the system model. From Eq. (2.14) we have [59]

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k-1}) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \ , \qquad (2.25)$$

which is based on the assumption that the system can be described by a Markov process of order one. By applying Chapman-Kolmogorov equation, the predicted state $\mathbf{x}_k$ based on all the previous observation data can be described as

$$p(\mathbf{x}_k \mid \mathbf{z}_{1::k-1}) = \int p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1} \ . \qquad (2.26)$$

At time k, a measurement $\mathbf{z}_k$ becomes available which can be used to update the prior estimation via Bayes' rule. This is performed in the step called updating

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1})}{p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})} , \qquad (2.27)$$

where $p(\mathbf{z}_k \mid \mathbf{x}_k)$ is the observation model at time step k; $p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1})$ can be obtained from

$$\begin{aligned} p(\mathbf{z}_k \mid \mathbf{z}_{1:k-1}) &= \int p(\mathbf{x}_k, \mathbf{z}_k \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_k \ , \\ &= \int p(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{z}_{1:k-1}) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_k \ , \\ &= \int p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) d\mathbf{x}_k \ . \qquad (2.28) \end{aligned}$$

Generally, the optimal solution of the posterior probability described by Eq. (2.27) is intractable. However, in a restrictive set of cases such as the Kalman filter, solutions do exist.

2.2.1 Kalman filter

There are some assumptions about the Kalman filter:

- The posterior density at every time step is Gaussian and hence can be parameterized by a mean and covariance.
- $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are drawn from zero mean Gaussian distributions of known parameters
- $\mathbf{f}_k(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$ and $\mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$ are known linear functions

Based on above assumptions, we have

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{v}_{k-1}, \tag{2.29}$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_k. \tag{2.30}$$

Here $\mathbf{F}_k$ and $\mathbf{H}_k$ are known matrices defining the linear functions. The covariances of $\mathbf{v}_{k-1}$ and $\mathbf{n}_k$ are $\mathbf{Q}_{k-1}$ and $\mathbf{R}_k$ respectively. The Kalman filter algorithm can be described as the following recursive relationship

$$p(\mathbf{x}_{k-1} \mid \mathbf{z}_{1:k-1}) = \mathrm{N}(\mathbf{x}_{k-1}; \mathbf{m}_{k-1|k-1}, \mathbf{P}_{k-1|k-1}), \tag{2.31}$$

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k-1}) = \mathrm{N}(\mathbf{x}_k; \mathbf{m}_{k|k-1}, \mathbf{P}_{k|k-1}), \tag{2.32}$$

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) = \mathrm{N}(\mathbf{x}_k; \mathbf{m}_{k|k}, \mathbf{P}_{k|k}), \tag{2.33}$$

where

$$\mathbf{m}_{k|k-1} = \mathbf{F}_k \mathbf{m}_{k-1|k-1}, \tag{2.34}$$

$$\mathbf{P}_{k|k-1} = \mathbf{Q}_{k-1} + \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^T, \tag{2.35}$$

$$\mathbf{m}_{k|k} = \mathbf{m}_{k|k-1} + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \mathbf{m}_{k|k-1}), \tag{2.36}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1}. \tag{2.37}$$

Here $\mathrm{N}(\mathbf{x}; \mathbf{m}, \mathbf{P})$ is a Gaussian density with argument $\mathbf{x}$, mean $\mathbf{m}$, and covariance $\mathbf{P}$

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k, \tag{2.38}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}_k^T\mathbf{S}_k^{-1}, \tag{2.39}$$

$\mathbf{S}_k$ and $\mathbf{K}_k$ Are the covariance of the innovation term $\mathbf{z}_k - \mathbf{H}_k\mathbf{m}_{k|k-1}$, and the Kalman gain respectively.

This is the optimal solution for the tracking problems if the highly restrictive assumptions can be satisfied. However, in most cases, the linear Gaussian assumptions cannot be hold. Particle filter is a widely used method to deal with the nonlinear non-Gaussian problems.

2.2.2 Particle filter

The particle filter is also known as the sequential Monte Carlo (SMC) [59], the bootstrap filtering [60], the condenstation algorithm [61], interacting particle approximations [62]-[63], and survival of the fittest [64]. It is a technique for implementing a recursive Bayesian filter without analytically calculating the posterior density function, which is intractable for nonlinear non-Gaussian problems. The posterior density function is estimated by a set of random samples with associated weights. As the number of samples becomes very large, this MC characterization of samples becomes an equivalent representation to the real posterior probability density. This process can be described by

$$p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i), \tag{2.40}$$

where $\{\mathbf{x}_{0:k}^i, i = 1,..., N_s\}$ is a set of support points with associated weights $\{\omega_k^i, i = 1,..., N_s\}$; and $\mathbf{x}_{0:k} = \{\mathbf{x}_j, j = 0,..., k\}$ is the set of all states up to time k. The states are estimated at time k, which are related to the states of N_s particles at time k; the weight for each particle can be sequentially estimated by different approaches such as sequential importance sampling (SIS) [65].

A common problem with the SIS particle filter is the degeneracy phenomenon: after a few iterations, all but one particle will have negligible weight. The degeneracy implies that a large part of computational effort is devoted to updating particles whose contribution to the approximation to $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ is almost zero. The effective sample size N_{eff} is a suitable measure of degeneracy of the algorithm introduced in [68]-[69]

$$N_{eff} = \frac{N_s}{1 + \mathrm{Var}(\omega_k^{*i})}. \tag{2.41}$$

Here $\omega_k^{*i} = p(\mathbf{x}_k^i \mid \mathbf{z}_{1:k}) / q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ is referred to as the "true weight", which can hardly be measured precisely. However, N_{eff} can be estimated by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N_s} (\omega_k^i)^2}, \tag{2.42}$$

where ω_k^i is the normalized weight obtained defined in Eq. (2.40)

To reduce the effect of degeneracy, we can select an importance density $q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_k)$ which minimizes $\mathrm{Var}(\omega_k^{*i})$ so that N_{eff} is maximized. The optimal

importance density function that minimizes the variance of the true weights ω_k^{*i} conditioned on $\mathbf{x}_{k-1}^i$ and $\mathbf{z}_k$ has shown to be [65]

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k),$$

$$= \frac{p(\mathbf{z}_k \mid \mathbf{x}_k \mid \mathbf{x}_{k-1}^i)\, p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^i)}. \qquad (2.43)$$

The weight can be calculated as

$$\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k \mid \mathbf{x}_{k-1}^i) = \omega_{k-1}^i \int p(\mathbf{z}_k \mid \mathbf{x}_k')\, p(\mathbf{x}_k' \mid \mathbf{x}_{k-1}^i) d\mathbf{x}_k' \ . \ (2.44)$$

Generally it is hard to sample[12] from $p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ and to evaluate the integral over the new state. The analytic evaluation is possible only in some special cases. However, it is often convenient to choose the importance density as

$$q(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k \mid \mathbf{x}_{k-1}^i), \qquad (2.45)$$

so that

$$\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k \mid \mathbf{x}_k^i). \qquad (2.46)$$

Resampling is another widely applied method to reduce the effect of degeneracy: Whenever a significant degeneracy is observed (i.e. when N_{eff} falls below a threshold), the particles are resampled. The basic idea of resampling is to remove particles with small weights by generating a new set $\{\mathbf{x}_k^{i*}\}_{i=1}^{N_s}$. The new particles are resampled N_s times from an approximate discrete representation of $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ given by

[12] There are different ways of sampling from a random distribution in different situations. For example if the random variable is with real values, one common approach is to compose the pseudo-inverse function of the cumulative distribution function (CDF) of the random variable $g(U)$. Then $g(U)$ is the acquired sample if U is a sample of a random variable with uniform distribution on (0,1). The detailed discussion can be found in [74]

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i), \qquad (2.47)$$

so that $\Pr(\mathbf{x}_k^{*i} = \mathbf{x}_k^j) = \omega_k^j$. Since the resulting sample is in fact an i.i.d. sample from the discrete density described by Eq. (2.47), the weights are now reset to $\omega_k^i = 1/N_s$.

It should be noticed that the resampling will introduce the problem of sample impoverishment although it can solve the degeneracy problem. This is because the particles with high weights are statistically selected many times, which leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. Algorithms have been proposed to solve this problem [60][66]-[67].

References

[1] P.K. Sahoo, S. Soltani, A.K.C. Wong, A survey of thresholding techniques, Comput.Vis. Graph. Image Proc. 41:233–60 (1988)

[2] H.R. Singleton, G.M. Pohost, Auto-matic cardiac MR image segmentation using edge detection by tissue classification in pixel neighborhoods. Magn. Reson. Med. 37:418-24 (1997)

[3] C.L. Gordon, C.E. Webber, J.D. Adachi, N. Christoforou, In vivo assessment of trabecular bone structure at the distal radius from high-resolution computed tomography images. Phys. Med. Biol. 41:495–508 (1996)

[4] W.E. Polakowski, D.A. Cournoyer, S.K. Rogers, M.P. DeSimio, D.W. Ruck, Computer-aided breast cancer detection and diagnosis of masses using difference of

Gaussians and derivative-based feature saliency. IEEE Trans. Med. Imaging 16:811–19 (1997)

[5] H. Cheng, Y.M. Lui, R.I. Freimanis, A novel approach to microcalcification detection using fuzzy logic technique. IEEE Trans. Med. Imaging 17:442–50 (1998)

[6] C. Lee, S. Hun, T.A. Ketter, M. Unser, Unsupervised connectivity-based thresholding segmentation of midsaggital brain MR images. Comput. Biol. Med. 28:309–38 (1998)

[7] G.B. Coleman, H.C. Andrews, Image segmentation by clustering, Proc. IEEE 5:773–85 (1979)

[8] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice Hall, (1988)

[9] L.A. Zadeh, Fuzzy sets. Inf. Control 8:338–53 (1965)

[10] D.L. Pham, J.L. Prince, An adaptive fuzzy c-means algorithm for image segmentation in the presence of intensity inhomogeneities. Pattern Recognit. Lett. 20:57–68 (1999)

[11] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. of the Royal Statist. Soci., Series B, 39(1):1–38 (1977)

[12] R. Neal, G. Hinton, A view of the EM algorithm that justifies incremental, sparse, and other variants, Learning in Graphical Models, pp 355-368. Cambridge, MIT Press (1999)

[13] M. Jamshidian, R.I. Jennrich, Acceleration of the EM Algorithm by Using Quasi-Newton Methods, J. of the Royal Statist.l Soci., Ser. B , 59: 569--587 (1997)

[14] J.C. Rajapakse, J.N. Giedd, J.L. Rapoport, Statistical approach to segmentation of single-channel Cerebral MR images, IEEE Trans. Med. Imaging 16:176–86 (1997)

[15] R.M. Haralick, L.G. Shapiro, Image segmentation techniques. Comput. Vis. Graph. Image Proc. 29:100–32 (1985)

[16] P. Gibbs, D.L. Buckley, S.J. Blackband, A. Horsman, Tumour volume detection from MR images by morphological segmentation, Phys. Med. Biol. 41:2437–46 (1996)

[17] S. Pohlman, K.A. Powell, N.A. Obuchowski, W.A. Chilcote, S.G. Broniatowski, Quantitative classification of breast tumors in digitized mammograms. Med. Phys. 23:1337–45 (1996)

[18] I.N. Manousakas, P.E. Undrill, G.G. Cameron, T.W. Redpath, Split-and-merge segmentation of magnetic resonance medical images: performance evaluation and extension to three dimensions. Comput. Biomed. Res. 31:393–412 (1998)

[19] J.F. Mangin, V. Frouin, I. Bloch, J. Regis, J. Krahe, From 3D magnetic resonance images to structural representations of the cortex topography using topology preserving deformations. J. Math. Imaging Vis. 5:297–318 (1995)

[20] J.K. Udupa, S. Samarasekera, Fuzzy connectedness and object definition: theory, algorithms and applications in image segmentation. Graph. Models Image Process. 58(3):246–61 (1996)

[21] S.Z. Li, Markov Random Field Modeling in Image Analysis, 2nd ed. Tokyo, Japan: Springer-Verlag (2001)

[22] G. Xiao, M. Brady, and J. A. Noble, Segmentation of ultrasound B-mode images with intensity inhomogeneity correction, IEEE Trans. on Med. Imag. Jan., 21(1): 48-57 (2002)

[23] A. Juslin, and J. Tohka, Unsupervised segmentation of cardiac PET transmission images for automatic heart volume extraction, Proc. of the 28th IEEE EMBS ann. int. conf., Aug., 1:1077-80 (2006)

[24] S. Ruan, and D. Bloyet, MRF models and multifractal analysis for MRI segmentation, Proc. of ICSP, pp. 1259-62 (2000)

[25] Y. Zhang, M. Brady and S. Smith, Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm, IEEE Trans. on Med. Imag., Jan., 20(1): 45-57 (2001)

[26] V.D. Calhoun, T. Adali, G.D. Pearlson, and J.J.Pekar, Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms, Hum. Brain Mapp., 13: 43-53 (2001)

[27] X. Descombes, F.Kruggel, and D.Y. von Cramon, Spatio-temporal fMRI analysis using Markov random fields, IEEE Trans. Med. Imag., Dec., 17(6): 1028-39 (1998)

[28] E. Salli, H.J. Aronen, S. Savolainen, A. Korvenoja, and A. Visa, Contextual clustering for analysis of functional MRI data, IEEE Trans. Med. Imag., May, 20(5): 403-14 (2001)

[29] J.C. Rajapakse, Y. Wang, X. Zheng, J. Zhou, Probabilistic Framework for Brain Connectivity From Functional MR Images, IEEE Trans. on Med. Imag., 27(6):825 - 833 (2008)

[30] S.P. Awate, R.T. Whitaker, Feature-Preserving MRI Denoising: A Nonparametric Empirical Bayes Approach, IEEE Trans. on Med. Imag., 26(9):1242 - 1255 (2007)

[31] C. D'Elia, C. Marrocco, M. Molinara, G. possi, G. Scarpa, and F. Tortorella, Detection of microcalcifications clusters in mammograms through TS-MRF

segmentation and SVM-based classification, Proc. of the 17th int. conf. on Patt. Recog. 3: 742-45 (2004)

[32] H.D. Li, M. Kallergi, L.P. Clarke, V.K. Jain, and R.A. Clark, Markov random field for tumor detection in digital mammography, IEEE Trans. on med. Imag. Sep., 14(3): 565-576 (1995)

[33] M. Kass, A. Witkin, D. Terzopoulos, Snakes: active contour models. Int. J. Comput. Vis. 1:321-331 (1987)

[34] C. Xu, J.L. Prince, Snakes, shapes, and gradient vector flow. IEEE Trans. Image Proc. 7(3) 359-369 (1998)

[35] V. Caselles, F. Catte, T. Coll, F. Dibos, A geometric model for active contours. Num. Math. 66:1-31 (1993)

[36] R. Ronfar, Region-based strtegies for active contour models. Int J. Comp. Vis. 13(2): 229-251 (1994)

[37] A. Blake, M. Isard, Active Contours, Springer, London, (1998)

[38] S. Lobregt, M.A. Viergever, A discrete dynamic contour model. IEEE Trans. Medical Imaging 14:12-24 (1995)

[39] L.D. Cohen, On active contour models and ballons, CVGIP: Image Understanding, 53: 211-218 (1991)

[40] A.J. Worth, N. Markris, V.S. Caviness, D.N. Kennedy, Neuroanatomical segmentation in MRI: technological objectives. Int. J. Patt. Recog. Artificial Intell. 11: 1161-1187 (1997)

[41] C.A. Davatzikos, J.L. Prince, An active contour model for mapping the cortex. IEEE trans. Med. Imag. 14: 65-80 (1995)

[42] C. Davatzikos, R.N. Bryan, Using a deformable surface model to obtain a shape representation of the cortex. IEEE Trans. Med. Imaging 15:785–95 (1996)

[43] T. McInerney, D. Terzopoulos, Medical image segmentation using topologically adaptable surfaces. Lect. Notes Comput. Sci. 1205:23–32 (1997)

[44] E. Bardinet, L.D. Cohen, N. Ayache, A parametric deformable model to fit unstructured 3D data. Comput. Vis. Image Underst. 71:39–54 (1998)

[45] A. Neumann, C. Lorenz, Statistical shape model based segmentation of medical images. Comput. Med. Image Graph. 22:133–43 (1998)

[46] I. Mikic, S. Krucinski, J.D. Thomas, Segmentation and tracking in echocardiographic sequences: active contours guided by optical flow estimates. IEEE Trans. Med. Imaging 17:274–84 (1998)

[47] J. Ho, W. Hwang, Automatic Microarray Spot Segmentation Using a Snake-Fisher Model, IEEE Trans. Med. Imag. 27(6):847 - 857 (2008)

[48] J. Hansegard, S. Urheim, K. Lunde, S.I. Rabben, Constrained Active Appearance Models for Segmentation of Triplane Echocardiograms, IEEE Trans. Med. Imag. 26(10):1391-400 (2007)

[49] I. Kompatsiaris, D. Tzovaras, V. Koutkias, M.G. Strintzis, Deformable boundary detection of stents in angiographic images, IEEE Trans. Med. Imag. 19(6):652-662 (2000)

[50] L.D. Cohen, On active contour models and balloons. CVGIP: Image Underst. 53:211–18 (1991)

[51] V. Caselles, F. Catte, T. Coll, F. Dibos, Ageometric model for active contours. Numer. Math. 66:1–31 (1993)

[52] R. Malladi, J.A. Sethian, B.C. Vemuri, Shape modeling with front propagation: a level set approach. IEEE Trans. Pattern Anal. Mach. Intell. 17:158–75 (1995)

[53] C. Xu, J.L. Prince, Snakes, shapes, and gradient vector flow. IEEE Trans. Image Process. 7:359–69 (1998)

[54] D. Comaniciu, V. Ramesh, and P. Meer, Kernel-based object tracking, IEEE Trans. on PAMI, 25(5):564-77 (2003)

[55] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, Pfinder: Real-time tracking of the human body. IEEE Trans. PAMI, 19(7):780-85 (1997)

[56] S. Yamamoto, Y. Mae, Y. Shirai, J. Miura, Realtime multiple object tracking based on optical flows IEEE Int. Conf. on Robotics and Automation, Vol.3 2328-33 (1995)

[57] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, IEEE Trans. on PAMI, 24(7):932-946 (2002)

[58] M. Isard, A. Blake, CONDENSATION–Conditional Density Propagation for Visual Tracking, Int. J. of Computer Vision, 29(1):5-28 (1998)

[59] A. Doucet, S. Godsill, and C. Andrieu, An introduction to sequential Monte Carlo methods, Statistics and Computing 10: 197-208 (2000)

[60] W.R. Gilks and C. Berzuini, Following a moving target- Monte Carlo inference for dynamic Bayesian models, J.R. Statist. Soc. B, 63: 127-146 (2001)

[61] J. MacCormick and A. Blake, A probabilistic exclusion principle for tracking multiple objects, in Proc. Int. Conf. Comput. Vision, pp. 572-578 (1999)

[62] D. Crisan, P. Del Moral, and T.J. Lyons, No-linear filtering using branching and interacting particle systems, Markov Processed Related Fields, 5(3): 293-319 (1999)

[63] P. Del Moral, Non-linear filtering: interacting particle solution, Markov Processes related Fields, 2(4): 555-580 (1996)

[64] K. Kanazawa, D. Koller, and S.J. Russell, Stochastic simulation algorithms for dynamic probabilistic networks, in Proc. 11th Annu. Conf. Uncertainty AI, pp. 346-351 (1995)

[65] A. Doucet, On sequential Monte Carlo methods for Bayesian filtering, Dept. Eng., Univ. Cambride, Uk, Tck. Rep. (1998)

[66] B.P. Carlin, N.G. Polson, and D.S. Stoffer, A Monte Carlo approach to nonnormal and nonlinear state-space modeling, J. Amer. Statist. Assoc., 87(418): 493-500 (1992)

[67] S. Godsill, A. Doucet, and M. West, Methodology for Monte Carlo smoothing with application to time-varying autoregressions, in Proc. Int. Symp. Frontiers Time Series Modeling (2000)

[68] N. Bergman, Recursive Bayesian estimation: Navigation and tracking applications, Ph.D. dissertation, Linkopin Univ. Linkoping, Sweden (1999)

[69] J.S. Liu and M. West, Combined parameter and state estimation in simulation-based filtering, in Sequential Monter Carlo Methods in Practice, A. Doucet, J.F.G. de Freitas, and N.J. Gordon, Eds. NY: Springer-Verlag (2001)

[70] D.L. Pham, C. Xu, and J.L. Prince, Current Methods in Medical Image Segmentation", Annual Review of Biomedical Engineering, 2: 315-337 (2000)

[71] K. Held, E. R. Kops, B.J. Krause, W.M. Wells, R. Kikinis, H.W. Muller-Gartner, Markov random field segmentation of brain MR images, IEEE Trans. Med. Imag., 16(6): 878-86 (1997)

[72] R.O. Duda and P.E. Hart, pattern classification, John Wiley & sons (2001)

[73] R. Kikinis, M. Shenton, F.A. Jolesz, G. Gerig, J. Martin, M. Anderson et al. Quantitative analysis of brain and cerebrospinal fluid spaces with MR imaging, J. Magn. Res. Imag. Ppp. 619-629 (1992)

[74] J.S. Milton, J.C. Arnold, Introduction to probability and statistics: principles and applications for engineering and the computing sciences, McGraw-Hill (2002)

Chapter 3 NeuronIQ: a new informatics tool for automatic neuron dendritic spine detection and analysis

3.1 Introduction

Recently, several semi-automatic dendritic spine analysis approaches have been proposed [1-3]. Although these methods can greatly help neurobiologists with their work, shortcomings still exist. Some manually-determined globally-applied thresholds are used during segmentation. In general, the proper determination of this threshold is largely dependent on the experience of human operator. More importantly, the final results in terms of dendritic morphology are quite sensitive to the thresholds, thereby making the reliable extraction of information difficult. Also, unbiased and rapid analysis of a large set of images by other detection methods is not feasible because of the amount of user input required. To solve the above problems, many locally adaptive thresholding methods have been proposed in the literature. Some are based on the local variance of the image intensity. More specifically, the threshold for each pixel is calculated according to the local mean and variance in a window of a pre-determined size [4-6]. The formulation is simple and straight forward; however, the results are easily corrupted by the presence of spurious local intensity maxima. Other methods use surface-fitting procedures [7-9]. As gradient based methods, they are sensitive to background noise. Kriging is also widely used in the literature [2][10]. It is basically a two-pass algorithm using two global thresholds. All the pixels whose gray values are smaller than the lower threshold are segmented as the background. Those with gray

values greater than the higher threshold are segmented as the objects. The remaining undetermined pixels are left to the second pass, at which point these pixels are segmented by using the local covariance of the class indicators and the constrained linear regression technique called kriging. In this chapter, we propose a local contrast based segmentation algorithm. It is easily implemented for automatic and consistent analysis of neuron morphology.

In addition to segmentation, other problems also exist for the above mentioned dendritic spines analysis algorithms. Koh et al. propose a morphology-based algorithm for automatically detecting and quantifying the structure of dendritic spines [1]. However, this is only a simple distance-based algorithm for detached spine head detection. Also, this geometric approach is sensitive to noise and cannot detect spines of all morphologies. To separate spines from the shafts of dendrites, Xu et al. propose a new attached spine component detection algorithm by using two grassfire propagations [3]. Although their method is more robust to the noise and the irregular, rough surface of dendrites, it still has a problem similar to Koh's approach: the shape of the detected spine is not accurately described, i.e., the base of the spine protrudes into the dendrite, instead of stopping parallel to the surface of dendrite. Based on Koh's approach, Weaver et al. described a package which is capable of morphometry of an entire neuron, by combining the spine detection algorithms with dendritic tracing algorithms [2]. Their algorithm, however, does not accurately represent the attached spine component either. In this chapter we propose a signal-to-noise ratio (SNR) -based detached spine component detection algorithm that we show can considerably reduce

the poor detection of spine components with low intensity values. Our attached spine components detection algorithm, which is based on the morphological analysis of the local dendrite, is also presented. The method we describe is more robust to noise and rough dendrite boundaries, while also representing the shape of spines more accurately than existing detection methods.

3.2 Pre-processing

There are two primary phenomena which cause the degradation of images obtained by optical microscopy. One is the shot noise introduced by the imaging mechanism of the photomultiplier tubes (PMT), which has been discussed in detail in Chapter 1. The shot noise can generally be removed by median filtering without a loss in information concerning the neuronal spines and dendrites. The other mechanism for image degradation is the diffraction of light. The value of any voxel in the image actually is the convolution of intensities from its neighborhood. Numerous methods have been proposed to solve this problem with or without knowing the Point Spread Function (PSF) of the system [11]. Based on our experience, deblurring algorithms can greatly enhance the contrast of the image, which makes it easier to detect weak spines. Nevertheless, since such processing can also amplify the noise, deconvolution sometimes degrades the segmentation results; thereby complicating later spine detection. Since our adaptive thresholding method can effectively segment the weak spines, we do not apply any deblurring algorithms during image preprocessing. As a result, the falsely detected spines are largely reduced, while the number of missed spines does not increase.

3.3 Segmentation

The next step after denoising is to distinguish the spine or dendrite pixels from the background. The segmentation threshold can be set manually or automatically. Since there are no clear criteria to guide manual selection, the results are quite operator dependent. Furthermore, usually the manually set threshold is a global threshold, which causes problem in term of finding spine components with low intensities. To address these issues, an automatic local contrast based segmentation method, which is suitable for high-content bioimage analysis, is presented here.

In adaptive threshold methods, a threshold is set for each pixel. This threshold is then used to test against the pixel intensity value to produce binary image. The basic formulation of adaptive threshold for pixel p is given by

$$T_p = T\big(h(p), I(p)\big), \tag{3.1}$$

where $I(p)$ is the gray level of point $p(x, y)$ in the original image, and $h(p)$ is a certain local property of this point p. For local contrast based algorithms, usually the average or median value of intensity is used to calculate the local threshold. White and Rohrer [12] compare the gray value of the pixel with the average of the gray values of neighboring pixels. The pixel will be recognized as the foreground pixel if its intensity is greater than the average intensity of all the pixels around. In the method proposed by Bernsen [13], the median value is selected as the threshold. Both methods work well at their cases. The width of the averaging window is set according to the quality of images

and the general size of foreground objects. Basically, a larger window size would help suppress the noise in the image. However, it might also cause the loss of weak signals that are close to other strong signals. Based on equation (3.1), the adaptive thresholding algorithm is expressed as follows:

$$s(x,y)=\begin{cases}1, & \text{if } I(x,y) > avg\{I(x',y') \mid (x',y') \in \Omega(x,y)\} \\ 0, & \text{otherwise}\end{cases} \quad (3.2)$$

.

Here, '1' represents the foreground pixels, which are potentially spine or dendrite pixels and will be processed for later detection and analysis, '0' represents the background pixels, and I is the intensity value. The local region around a pixel (x,y), i.e., $\Omega(x,y)$ is defined as $\Omega(x,y)=\{(m,n)\big| \mid m-x \mid \leq (d-1)/2, \mid n-y \mid \leq (d-1)/2\}$. The quantity d is the width of the window which decides the size of local region. The value of d is selected by considering the width of most spines. Normally, d can be set about 1~2 times of the spine width value. For our test images in Section 3.7, optimal results can be obtained for any number between 15 and 20 pixels. Generally, the segmentation results of spines are not sensitive to the window size, as long as enough foreground and background pixels are included. However, problems may occur for those small weak spines which are very close to the dendrites (whose intensities are higher than those of nearby spines). We will propose an efficient solution to this problem in the later part of this section.

Another potential problem for the above adaptive threshold algorithm is that some background pixels with relatively high intensity are prone to be segmented as foreground pixels. More information should be applied in order to distinguish this kind of local maxima from the real spine pixels. Besides the intensity difference, the exact value of each pixel's intensity should also be considered. Only those pixels whose intensities satisfy the minimal value requirement can be segmented as foreground pixels. For example, a pixel with intensity values as 1 should not be segmented as a foreground pixel if all its neighboring pixels' intensities are 0. This idea is similar to the lower threshold introduced by the Kriging method [2][10]. The minimal intensity value of a possible foreground pixel is selected as follows:

$$T_{\min} = T_{low} + \alpha \quad . \tag{3.3}$$

Here, T_{low} is the global lowest intensity value; α is a control parameter. When α is zero, the segmentation results are totally decided by the local contrast. With a bigger α, local maximal intensity pixels are less likely to be segmented as foreground pixels. Generally, even a very small α can obviously improve the segmentation results. This is because the high intensity noise, which is the shot noise, has already been removed by the median filter in the preprocessing stage. The original images are 16-bit grey level image. During pre-processing, they are linearly transferred into 8-bit images. Based on the range of intensity of spine pixels in the transferred 2PLSM images (1~255), the value of the control parameter α is suggested as $\alpha \in [5,20]$. The validation and comparison results in Section 3.7 are obtained with setting α =15 for all images.

Besides the intensity, local region information should also be considered to improve the segment results. For example, a pixel has a higher probability to be a foreground pixel if most of its neighboring pixels are foreground pixels. This is based on the assumption that spines are blob-like objects. A simple approach for this idea is to only select those pixels whose most neighbors are segmented as foreground pixels. This post processing can be easily realized by implementing a median filter to the originally segmented image $s(x, y)$.

For neuron images, a common problem for local thresholding is that the weak signal might be suppressed by nearby strong signals. For example, for weak spines, low intensity pixels such as those at the spine neck are prone to be segmented as background pixels. This is because those pixels have lower intensity values compared with their neighboring average, which is caused by the nearby dendrite pixels with very high intensity. To solve this problem, the high intensity values of nearby dendrite pixels should be adjusted, i.e., substituting the dendrite pixels with low intensity values. This is realized by two-step segmentation: dendrites are found after the first adaptive thresholding (Figure 3.2). The dendrites are defined as those blobs whose areas are larger than the largest possible size of a spine; then all pixels in the dendrites are replaced with lower intensity values in the original image; the adaptive thresholding is performed the second time after the intensity substitution. Compared with single adaptive thresholding, the two-step adaptive thresholding method can better detect the low intensity pixels at spine neck. Thus, the shape of spines is better represented. However, the revised intensity of dendrite tissues cannot be set arbitrarily low,

otherwise some background pixels with relatively high intensity values will be wrongly segmented as foreground pixels. Therefore, the intensity adjustment should be carefully chosen with low bound set.

The intensity adjustment can be linear or nonlinear. In our case, we apply a simple linear transformation, i.e. $I'_d(x,y) = \varepsilon \cdot I_d(x,y)$. Here $I_d(x,y)$ is the original intensity of the pixel in dendrite, $I'_d(x,y)$ is the intensity value after adjustment, ε is the parameter which decides the suppression intensity. The upper bound for ε is 1, which means no suppression is performed. The lower bound for ε is calculated as follows:

$$\varepsilon_{\text{lower bound}} = \frac{\bar{I}_s}{\bar{I}_d} . \qquad (3.4)$$

Here $\bar{I}_s$ is the average intensity of all pixels in spines, and $\bar{I}_d$ is the average intensity of all pixels in dendrites.

After segmentation, basic morphological filtering (with size 3 disc as the structure element) is performed to remove the noise, fill the holes, and smooth the boundaries. Some morphological processing, such as filling holes, can distort dendrite structures. To prevent these from happening, local intensity information (e.g. only fill the holes in which the average intensity of the pixels is above a threshold so that only the holes *inside* the dendrites or spines are filled) is considered during the processing. Figure 3.1 shows the comparison of segment results between global thresholding and adaptive thresholding. The latter one is obviously better, with fewer missed spine components and better segmented dendrite structure.

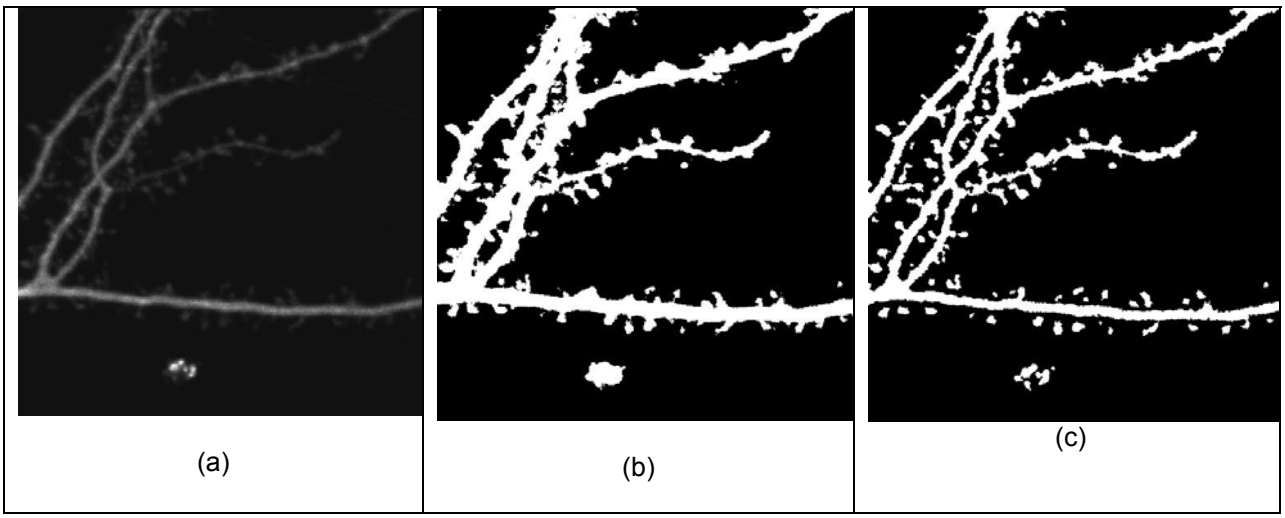


Figure 3.1: results of adaptive thresholding; (a) original images; (b) segmentation result with global threshold; and (c) segmentation result with adaptive threshold

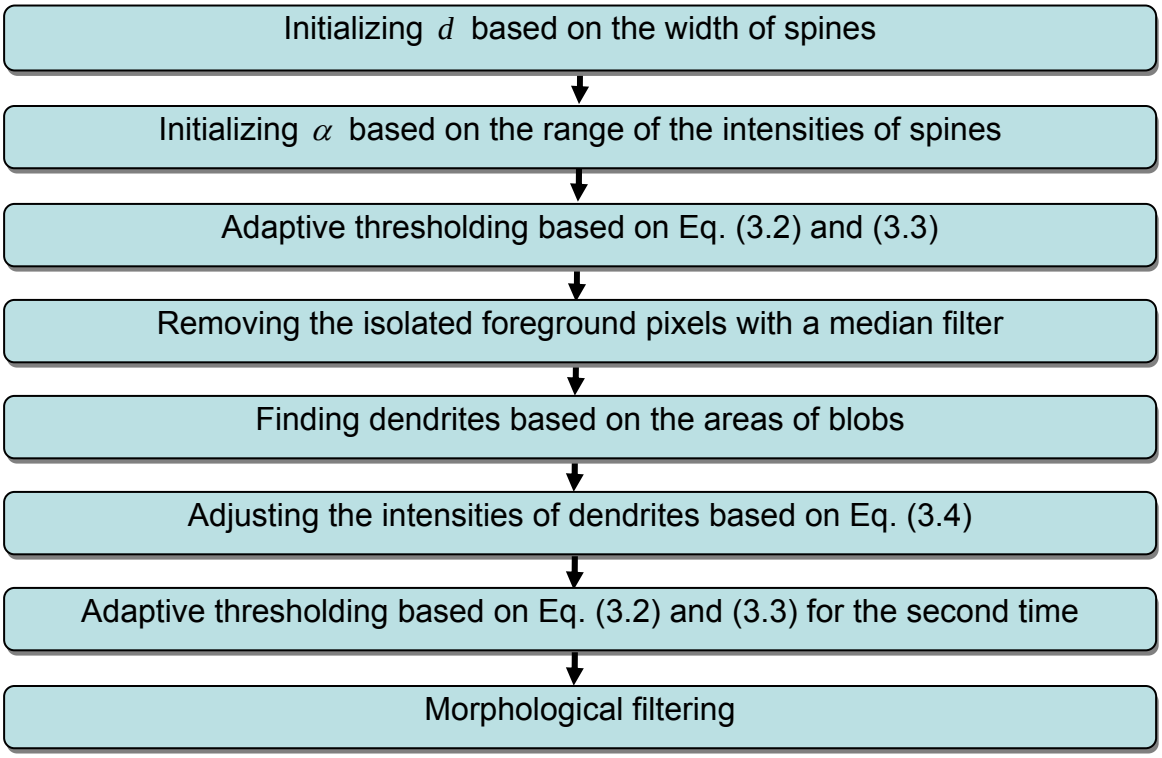
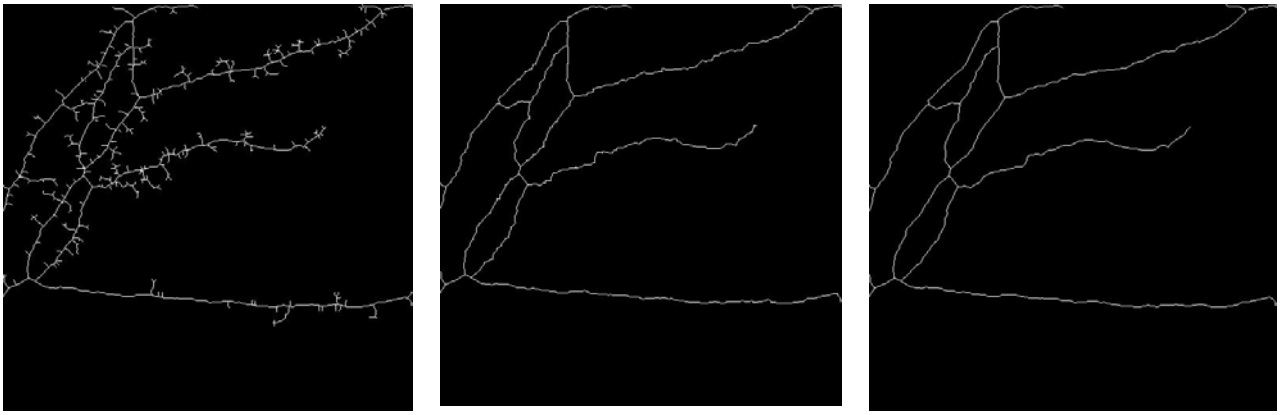


Figure 3.2: procedures of the segmentation algorithm

3.4 Backbone extraction

Dendrite structure is one of the most important geometric features in spine detection. Most existing spine detection algorithms are based on the successful extraction of backbones. There are two steps for backbone extraction: determining the medial axis by thinning algorithms and extracting the backbone from the medial axis by trimming branches. There are many thinning algorithms in the literature [14-19] to find the medial axis in 3D. Basically a set of deleting templates are designed, which preserve the topology and geometry of the object. The object voxel under consideration is checked against the templates and is removed if its spatial arrangement is compliant to one of them. The medial axis or skeleton obtained by such thinning algorithms is sensitive to noise. As we can observe from Figure 3.3, there are many spurs caused by noise (rough boundary). These spurs, in addition to the spurs caused by spines, should be removed to obtain a neater description of dendrite structures (backbones). The backbone is obtained by the trimming algorithms. In this section, we only discuss our trimming algorithm.



(a) (b) (c)

Figure 3.3: (a) medial axis of dendrites, (b) backbone after trimming, and (c) backbone after removing bumps.

In some ideal situations, trimming can be very easy. All we need to do is to set a threshold for the physical length of the spines, and remove all spines whose length does not satisfy the requirement. Or we can just pick up the longest pieces. This is the basic idea behind many of the skeleton pruning algorithms. An example of this kind of algorithm is proposed by Zikuan Chen et. al. [20]. They first try to find the starting point of the skeleton, by finding the line-end point with the maximal intensity value (or maximum diameter). From the starting pointing, the longest path P_{ij} is found. P_{ij} is then removed from the original skeleton image and another starting pointing is found. The procedures are repeated until the top N longest paths are found, which are assumed to be the dendrite backbone pieces. There are two potential problems for the above mentioned method. One concerns with the selection of the starting point. While one could consider all line-end points as starting points, this will however greatly increase the computation time. The other difficulty is how to 'automatically' set the value of N . As we can see from Figure 3.3, the medial axis of a dendrite image can be very complex.

This is caused by the complex structures in the big spines, or caused by the rough dendrite surface. If the spine density is high, the longest path based trimming algorithm can be very time consuming. In chapter, we introduce a recursive trimming algorithm, which can deal with the above problems without knowing the starting point and number of dendrite pieces. The algorithm is summarized as follows with the detailed listing of the steps provided in Table 3.1.

Table 3.1: recursive trimming algorithm

(1) Initialize $m = 1$

(2) Remove all end points (who only has one neighbor) in the image; keep doing for m times

(3) Check the length of all the removed pieces l_i; if $l_i = m$, restore the removed piece

(4) Let $m = m + s$, if $m < M$, go to step 1

(5) Finish trimming

The basic idea is to keep removing end pieces (curves with end points) whose length is less than m. The value of m is iteratively changed in step (4). The process keeps repeated until the threshold m reaches the upper bound M, which is usually the longest possible length of a spine. Notice this algorithm is not simply removing all end pieces whose length is smaller than the threshold M, a process that would not only remove the medial axis of spines, but also will remove the end pieces of dendrites. This can be a serious problem when the threshold M is big. In our approach, the dendrite pieces are restored at step (3). There are two situations in which the removed pieces cannot be restored, i.e., $l_i \neq m$. 1) The length of the end piece is less than the

temporary threshold m. In this case the short branches (medial axis of spines) are removed. 2) The length of the end piece is greater than the temporary threshold m. This happens when more than one connected branches are removed together. This is also the reason why the algorithm is implemented in an iterative way: if $m = M$ at the very beginning, the end pieces of dendrites won't be restored, because there might be some spines attached to the end pieces of dendrites and are removed together. In such situation, $l_i > m$ and the end pieces of dendrites cannot be restored. The trimming result is shown in Figure 3.3 (b).

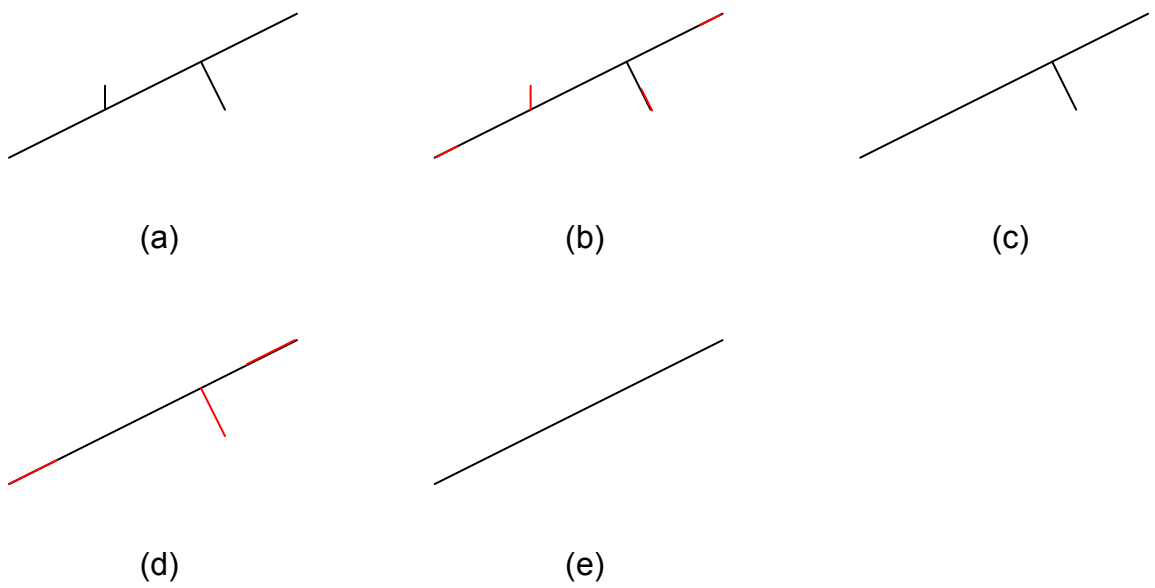


Figure 3.4: (a) backbone with small branches (b) end pieces with length $\leq m$ are removed (c) end pieces with length $= m$ are restored (d) end pieces with length $\leq m + s$ are removed (e) end pieces with length $= m + s$ are restored

Figure 3.4 illustrates the processing of removing small branch pieces with different length. The red line represents the removed pieces in the specific steps.

After trimming, some correction to the backbone is performed. As we can see in Figure 3.3 (b), there are many small bumps along the backbone. These bumps are caused by the thinning algorithm during the process of obtaining the medial axis. The bumps may introduce the error for dendrite length estimation. They also cause trouble for spine length estimation and make length based attached spine detection algorithm more difficult. Usually, the bumps locate at the place where there is a protrusion at the dendrite surface. For each protrusion, there is a piece of small branch in the medial axis. Based on the above facts, here we propose a bump-removing algorithm as follows: (1) for each piece of branch along the backbone, find the branch point which is defined as the backbone pixel having more than two neighboring pixels (8-neighorhood) in the medial axis image; (2) find local backbone piece around the branch point inside a 5×5 window, where is the possible location of a bump; and (3) replacing the local backbone piece by a line connecting the two end points of the backbone piece.

3.5 Spine Detection

From Figure 3.1 we see that there are some blobs which are not in a reasonable distance to the dendrites, i.e., the distance is larger than the longest possible length of a spine. This kind of blobs should be removed as non-spine blobs. The regions around the backbone should be defined before the detection, which will help reduce both the processing time and false positives. A local-region-cutting algorithm is designed for the above purpose. Only blobs locate in the local regions are considered as potential spine

components. Those who are not in the local regions are removed before spine detection. Although the maximal distance idea of the off-cutting algorithm is identical to Koh's method [1], it is implemented in a more computation efficient way. Instead of calculating the distance from every detached blobs to the nearby dendrite backbone point, we define a local region around the backbone, which is simply obtained by dilating the backbone. The advantage becomes obvious if the spine density is high.

There are two different approaches for the detection of attached spine components and detached spine components. The two detection algorithms are discussed in the following sections.

3.5.1 Detection of detached spines

The detached spine component detection algorithm is based on the local Signal to Noise Ratio (SNR) analysis. The local region of each potential spine should be found before estimating the local SNR. Since the size of different spines may vary a lot, it is not feasible to choose a window with fixed size for local SNR analysis. The size of the window should be adjusted according to the size of each blob. The local region R_p for blob p is defined as follows:

$$R_p = \{(x, y) \mid x \in [x_0 - \lambda, x_1 + \lambda], y \in [y_0 - \lambda, y_1 + \lambda]\}. \qquad (3.5)$$

Here x_0, x_1, y_0, and y_1 are the least and largest values in x- and y- direction respectively for all pixels in the blob; λ is the extension parameter whose value will decide the ratio of background and foreground pixels in the local region. To get the size

of the local region R_p for local SNR analysis of each spine, we need to calculate the value of λ. Suppose the area of the blobs can be estimated by a rectangle $a \times b$, with $a = x_1 - x_0$ and $b = y_1 - y_0$ and the total area of local region is $n \cdot a \cdot b$, where n is a real number which satisfies $n > 1$, then the ratio between the number of background pixels and that of the foreground pixels is $1:(n-1)$. The value of λ can be obtained from

$$(a + 2\lambda)(b + 2\lambda) = nab, \tag{3.6}$$

$$\lambda = \frac{\sqrt{(a+b)^2 + 4ab(n-1)} - (a+b)}{4}, \tag{3.7}$$

For the purpose of implementation, λ can be rounded to the nearest integer.

Once the local region for a spot is determined, we can then define the local SNR. In image processing, the local SNR for spot P is usually estimated as [21]

$$SNR_p = (I_{sp} - I_{bp}) / \sigma_{bp}. \tag{3.8}$$

Here, I_{sp} is the average intensity of the detected signals, I_{bp} is the average intensity of the background pixels, σ_{bp} is the standard deviation of the background intensity of the local region. The local SNR analysis is applied to all detached blobs in the maximal intensity projection (MIP) image of a neuron image for spine detection. If the SNR is higher than a threshold, the certain blob is recognized as a detached spine. The threshold can be obtained by common clustering method such as k-nearest neighbor algorithm.

In some simple cases, this SNR based algorithm can detect detached spines well. However, sometimes some small non-spine blobs which arise from dying or imaging problems cannot be removed by simply increasing the SNR threshold. Nevertheless, the performance of SNR based detection algorithm will be improved if intensity distribution information among the neighboring slices can be used. For 3D neuron images, the resolution in z-direction is much lower compared with x- and y-direction. Most spines will only appear in 2 or 3 slices. Thus, the section area of a spine changes obviously in the neighboring slices. In addition, the intensity for spine voxels in the neighboring slices changes more obviously compared with the voxels in non-spine blobs. This is can be explained by the intensity difference among surface voxels and central voxels in a spine. Based on the above observations, a new SNR which considers the 3D intensity properties of a spine is proposed:

$$SNR_p = SNR_{normal} \bullet f_1 = \frac{I_{sp} - I_{bp}}{\sigma_{bp}} (1 + \frac{N_{FG}}{A_{local}})^{\eta} \qquad (3.9)$$

Here, A_{local} is the area of the blobs in the MIP (maximal intensity projection) image; N_{FG} is the number of foreground voxels. The foreground voxels are defined as: $P_{foreground} \in \{P \mid \text{voxels whose intensity difference in adjacent slices} \geq n_0\}$; n_0 is a threshold for intensity difference. If n_0 = 1, then all voxels with different intensity values among the neighboring slices are considered as foreground voxels. For each blob, three slices (central slice and two neighboring slices) are considered. The central slice of a blob is defined as the one with the highest average intensity. According to our tests, the intensities of voxels in the small non-spine blobs have few changes compared with those spine voxels. The variable η is the weight parameter with positive value. We can

observe that the newly defined SNR_p is equal to SNR_{normal} if $\eta = 0$. With a higher η, we emphasize more on the change of intensity distributions among the neighboring slices. In our case, η is set to 2 for all images used for validation. The value is manually set according to the testing results of several images.

The comparison of detection results based on normal SNR and revised SNR is shown in Figure 3.5. As we can observe from the SNR distribution maps, there is no threshold which can totally separate the spine components and the non-spine blobs by using normal SNR. However, these two different classes are finely separated if described by the revised SNR. Nearly all non-spine blobs have obviously lower values compared with the spine components, e.g., the values are lower than half of the value of the spine with the lowest revised SNR. We also notice that there are some non-spine blobs introduced by a crossing axon at the middle bottom of the image. This kind of non-spine blobs can also be recognized and discarded by using the revised SNR.

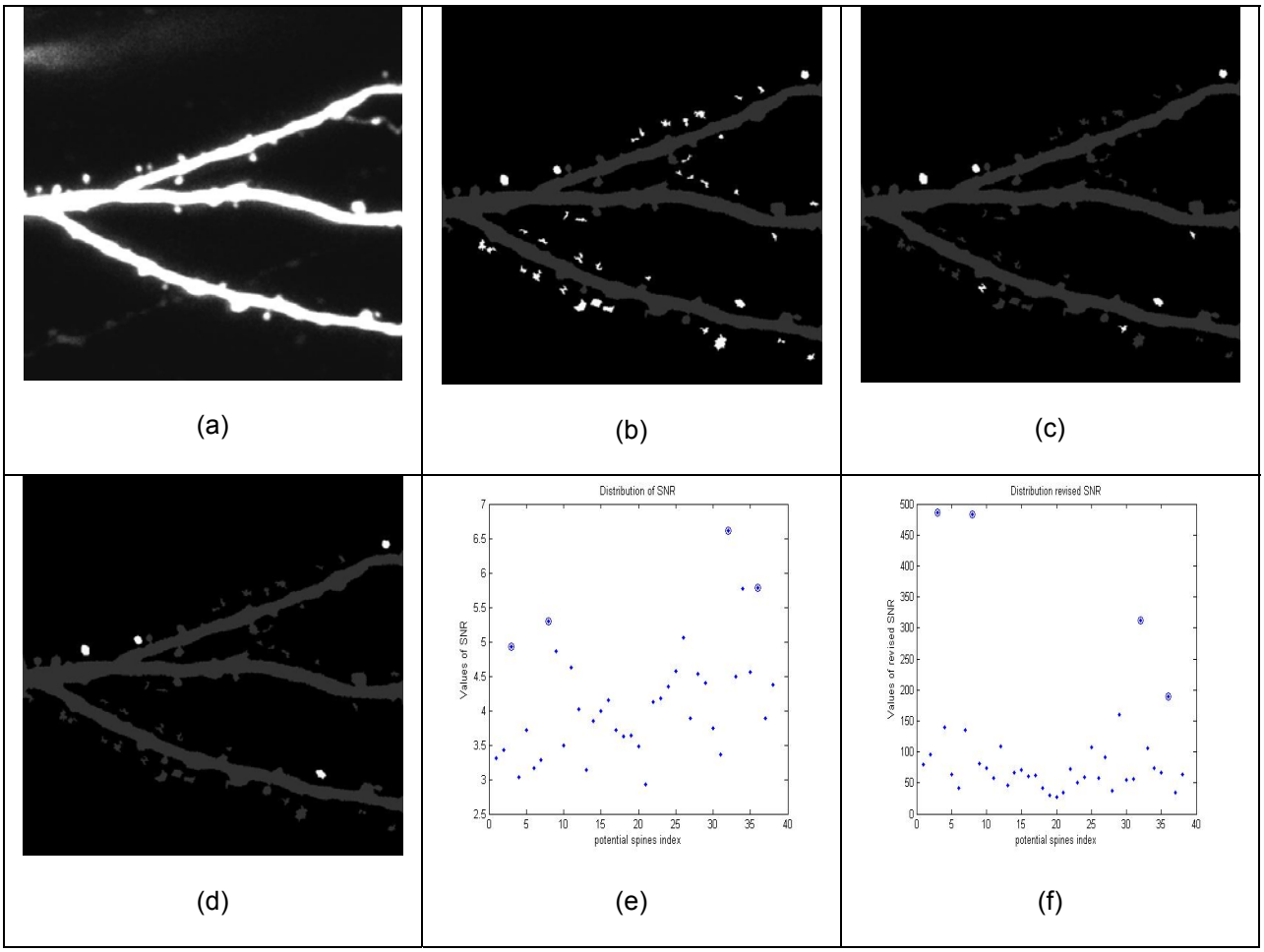
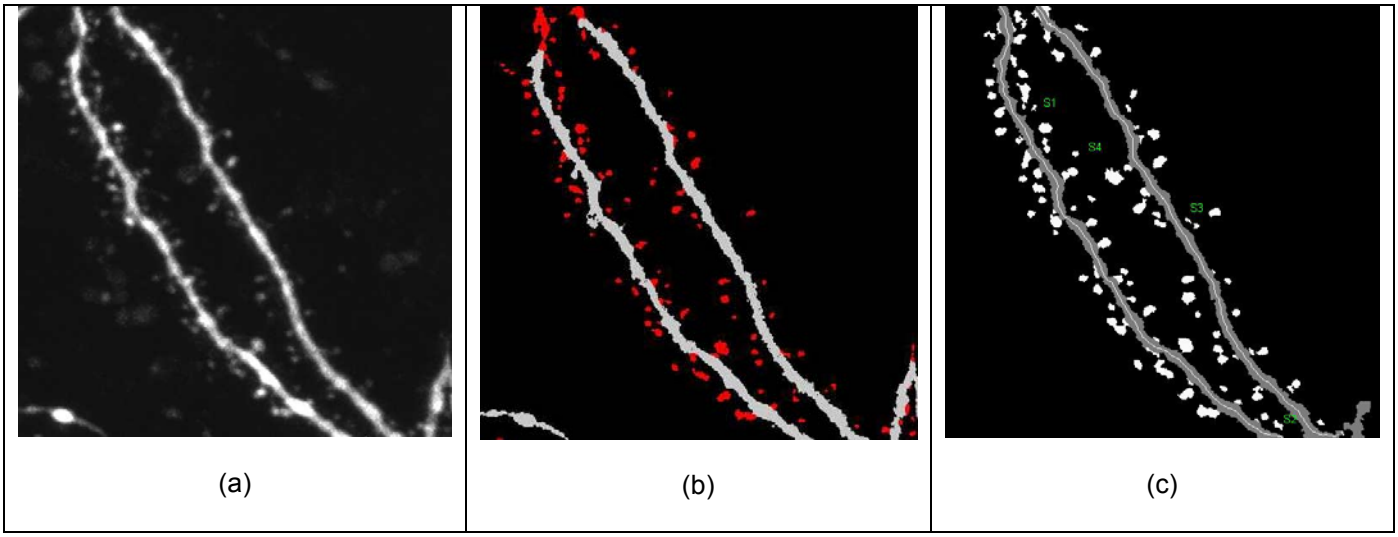


Figure 3.5: comparison of normal SNR and revised SNR for detached spine head detection. (a) original images; (b) segmented image obtained from adaptive threshold algorithm, with all detached blobs (possible detached spine head) marked; (c) detached spine head detected by using normal SNR; (d) detached spine head detected by using revised SNR; distribution of (e) SNR and (f) revised SNR for each detached blob. The circles represent the manually detected spine head in 3D image.

From Fig. 3.5 (c) and (d) we can see that all three false positives (non-spine blobs) are removed. Furthermore, our SNR analysis and adaptive thresholding based algorithm can also detect weak detached spine components, which are possibly missed by global thresholding. From Figure 3.6 we can see that four spines are missed

because of low intensity values by using global thresholding. For these spine components detected by both approaches, the shape of spine components are much better represented by using our method, e.g., the volume of spine will not apparently shrink because of a relatively high global threshold, which may be introduced to suppress strong noise. The quantitative comparison of spine detection results between our method and Koh's method [1] is presented in the next section.



(a) (b) (c)

Figure 3.6: comparison of spine detection results: (a) original images, (b) Koh's approach, and (c) our method.

3.5.2 Detection of attached spines

The approach to detecting attached spine components along the dendrite is quite different from that to detached spine components detection. Unlike the local contrast and SNR-based approach, it is mainly based on the local morphological analysis.

In the segmented image, there are some protrusions along the dendrite structures, which are potentially attached spine components. To find those spine components, one common approach is to find the tips for each protrusion, then locate each spine by local shape analysis [1]. The tips are the pixels along the boundary, which have local maximal distance to the backbone pixels.

There are several problems for this kind of detection method. The location of the tip is noise sensitive: with rough boundary of the dendrite, there will be too many existing local maximal distance pixels and it is hard to find the 'ideal' tip (the maximal distance pixel on the top of a protrusion). This problem can be partially solved by smoothing the dendrite with low-pass filters.

Another problem is that the shape of the spines is not well represented: the base of the spines is described as a piece of curve, instead of a straight line parallel to the nearby dendrite boundary. This is because the spine pixels are defined as the pixels those are within a certain range to the tip point of the protrusion. The local maximal distance based algorithms also have difficulties to detect some very small spines or spines with irregular shapes.

Here we propose a medial axis based attached spine detection algorithm, which can solve the above problems. In the medial axis image, there is a branch (small piece of line) at the location of each spine along the dendrite. Thus, the attached spine

component can be located by referring the position of the branches. However, there are also many branches at the place where no spine presents. Some criteria are needed to locate the real spines. The algorithm is described as follows:

(1) find all small branches according to the medial axis and backbone image, the small branches are found by removing backbone from the medial axis;

(2) estimate the local thickness of the dendrite in a neighboring region around each small branch;

(3) mark spine candidates based on the estimation of the local thickness and the distance map to backbone; all pixels in the local region that have a distance larger than the local thickness are marked as spine pixels;

(4) remove false spines based on the area and edge criteria.

The local thickness of the dendrite is estimated by calculating some local edge pixels' distance to backbone. Suppose p_i is the boundary pixel along the local region, and d_i is the distance to the nearest backbone pixel. The local thickness of the dendrite is estimated by:

$$Th = median(\{d_i \mid d_i \le \min(d_1, d_2, \cdots, d_k) + \beta\}) \quad (3.10)$$

Here, $d_{\min} = \min(d_1, d_2, \cdots, d_k)$ means the minimal distance among all k boundary pixels; β is a manually set threshold which decides the range of distance to be considered. To be more robust, the local thickness is not set equal to the local minimal distance. Instead, a set of local distance values are considered and the median value is picked as

the estimation of local dendrite thickness. So β should be greater than zero. Compared with the dendrite boundary pixels, the spine boundary pixels are much farther away from the backbone. For the purpose of the accurate estimation of local dendrite thickness, spine boundary pixels should be excluded to the largest possible extent. Thus, β should also have a higher bound. Usually, the value of parameter β is set according to the quality of the segmented results. If the quality of the image is very low, β should be set to a larger number to lower the probability of being trapped in the local minimums. For images with ordinary quality, small value of β can be chosen to exclude more spine boundary pixels. For all the images demonstrated here $\beta = 2$. The reason that we choose the median value of local distances is based on the observation: most of the local non-protrusion boundary pixels have the same distance to the backbone, which is a good estimation of the local thickness of a dendrite.

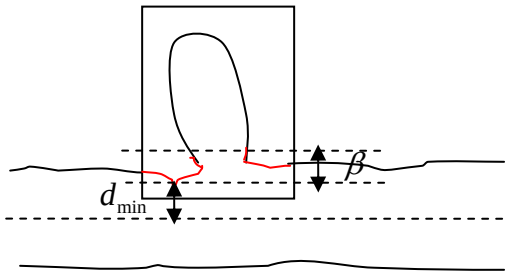


Figure 3.7: estimation of the local dendrite thickness. The local dendrite thickness is set as the median distance from qualified edge pixels to the backbone. The qualified edge pixels are marked red. The local region is represented by the rectangle.

As we mentioned before, not all spine candidates are actually a spine. Some criteria are set here to help to remove those false spines.

Area criterion:

Only those candidates whose area is larger than a threshold can be a real spine.

$$C_i = \begin{cases} \text{is a spine}, & \text{if } \mathrm{Area}(C_i) > \mathrm{A}_0 \\ \text{is not a spine}, & \text{if } \mathrm{Area}(C_i) \le \mathrm{A}_0 \end{cases}. \qquad (3.11)$$

Here C_i is the spine candidate found by the above detection algorithms; A_0 is the area threshold. This criterion is set to help remove the false positives along the rough dendrite boundary. Usually neurobiologists do not consider very small protrusions as spine components. This threshold can be manually set before the batch processing. Usually the value is related to the choice of different views. For example, in our experiments (where 0.084 micrometer/pixel), we ignore all small protrusions whose area is less than 5 pixels.

Edge criterion:

Only those candidates that locate outside the trunk of the dendrite can be a real spine.

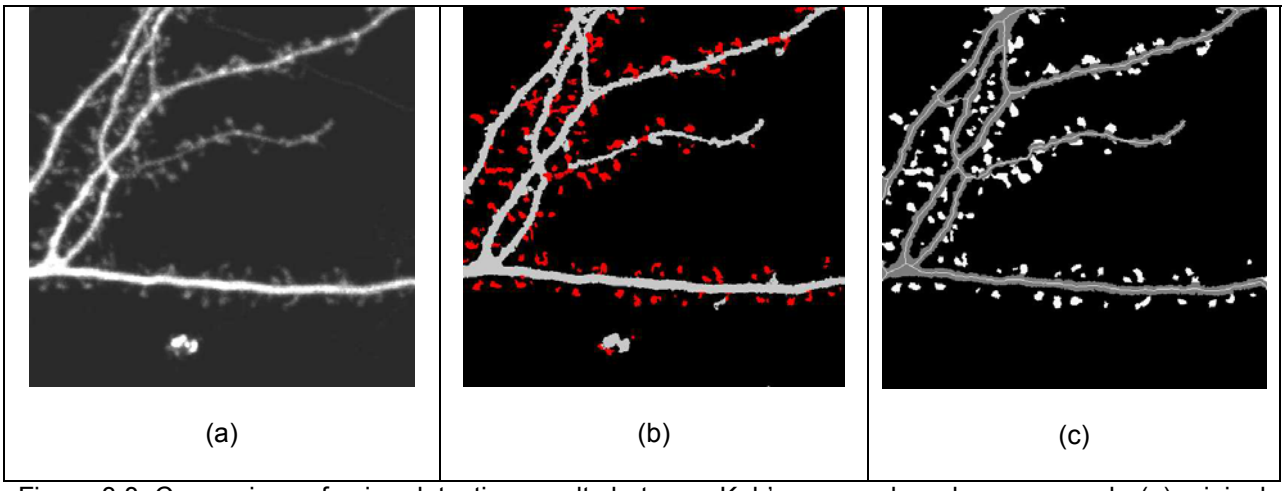
$$C_i = \begin{cases} \text{is a spine}, & \text{if } N'_{C_i} > N''_{C_i} \\ \text{is not a spine}, & \text{if } N'_{C_i} \le N''_{C_i} \end{cases}. \qquad (3.12)$$

Here N'_{C_i} is the number of boundary pixels of a spine candidate, which are also the boundary pixels of a local dendrite; N''_{C_i} is the number of boundary pixels of a spine candidate, which are not the boundary pixels of the local dendrite (pixels inside the dendrite). This criterion is introduced to remove the false detected spine components which protrude deep into the dendrite.

For the purpose of easy implementation, the algorithm for attached spine detection is based on the projection of backbone and the image of whole stack. However, the idea of estimation of the thickness of local dendrite pieces and the criteria to remove the false positives can also be implemented in 3D.

Besides the advantage of being suitable to high-content batch process, our approach also has better detection results compared with the existing approaches. Figure 3.7 shows the comparison of spine detection results between our method and Koh's method [1]. Compared with the existing spine detection algorithms, our method has the following advantages:

- Better detection results for very weak spines. Either for attached spine or detached spine components
- Fewer false positives caused by the broken dendrite part. This is due to our robust adaptive thresholding method.
- The shape of the spines is better represented, e.g., the bottom of the attached spine is a line close and parallel to the edge of dendrite. For Koh's approach, sometimes the position of the spine bottom cannot be found correctly, which means either only a part of the spine is detected or nearby dendrite tissue is segmented as the spine.



(a) (b) (c)

Figure 3.8: Comparison of spine detection results between Koh's approach and our approach: (a) original images; (b) detection results by using global thresholding and Koh's approach; and (c) detection results of our fully automatic approach.

3.6 Post-processing

There are mainly two purposes for post processing. One is to adjust the numbers of detected spines, and the other is to provide measurements about the spines and dendrites. The number of blobs detected as spines is not actually the real number of spines. In some cases, a spine can be broken into several parts in the image. One reason is that the signal around the neck of some thin spines is so weak that the spine is broken into the detached head and attached base components after segmentation.

A merging algorithm is performed to combine those detached and attached parts. The merging algorithm considers every attached spine or spine base, checking for possible merges with other detached components in the local region. It is possible that several detached components are recognized as the parts of a single spine. This

happens when the detached components are combined with the same attached spine base.

There are two criteria for the merging algorithm: the separated components should be close enough, and the separated components should satisfy certain relative orientation requirement. It is easy to implement the first criterion. For the second criterion, suppose P_1P_2 is the cutting line of the attached spine based on the dendrite; C_1C_2 is the connection line of the centers of attached component and detached component; θ is the angle between P_1P_2 and C_1C_2. If $\theta \geq \theta_0$, then the separated components are considered to be the parts of the same spine. A suggested range for the threshold θ_0 is: $\theta_0 \in [30^0, 45^0]$. In our case, $\theta_0 = 40^0$. Compared to Koh's method which requires the accurate position of the tip of each protrusion, this method is more robust to noise and bad segmentation.

During post processing, some important measurement of spines and dendrites are acquired. NeuronIQ provides the function to measure dendrite length, spine length, spine density, spine volume, spine section area, spine section perimeter, spine neck width, etc. For the purpose of comparison with manual results, only spine length and spine density are discussed here.

Spine length and spine density are two of the most important properties for neurons. The spine density is defined as: $d = n/l$, where l is the length of a dendrite, and n is the total number of spines on that dendrite.

The length of a dendrite is calculated based on the backbone image. One common approach for this problem is to first code the backbone by using some chain coding algorithms, e.g. Freeman Chain Coding [22] or Primitives Chain Coding [23]. The total length of the dendrites is calculated afterwards: for 2D images, when two pixels are in 4-neighborhood to each other, the total length is increased by 1; when two pixels are in 8-neighborhood but not in 4-neighborhood to each other, the total length is increased by 1.414.

The problem for the coding based algorithm is that the codes may become complex when the structure of the dendrites is complex. The coding becomes complex because it tries to store the exact location information for each pixel. However, we do not need to know the exact location of each pixel of the dendrite in order to figure out the total length of the dendrite. Based on this idea, we propose a fast and efficient algorithm to estimate the length of a dendrite:

(1) get the total number of pixels n_0;

(2) get the total number of pixels which have one (8-4)-neighbor, say n_1;

(3) get the total number of pixels which have two (8-4)-neighbors, say n_2;

(4) calculate the length of dendrite: $l = (n_0 - 1) + \frac{(\sqrt{2} - 1) \cdot (n_1 + 2n_2)}{2}$

A (8-4)-neighbor is defined as a pixel which is in the 8-neighborhood, but not in the 4-neighborhood at the same time. The number of (8-4)-neighbor for each backbone pixel can be easily obtained by filtering backbone image. The filtering window is defined as:

$$w = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Actually, n_1 and n_2 can be obtained by one time filtering. This method can be easily extended to 3D images. We just need to find the total number of (26-18) neighbors in the similar way.

By applying the same method, the length of each spine is obtained by calculating the length of branch relative to the spine. The branch is obtained from the medial axis image and backbone image.

3.7 Results and discussions

3.7.1 Results analysis based on single image

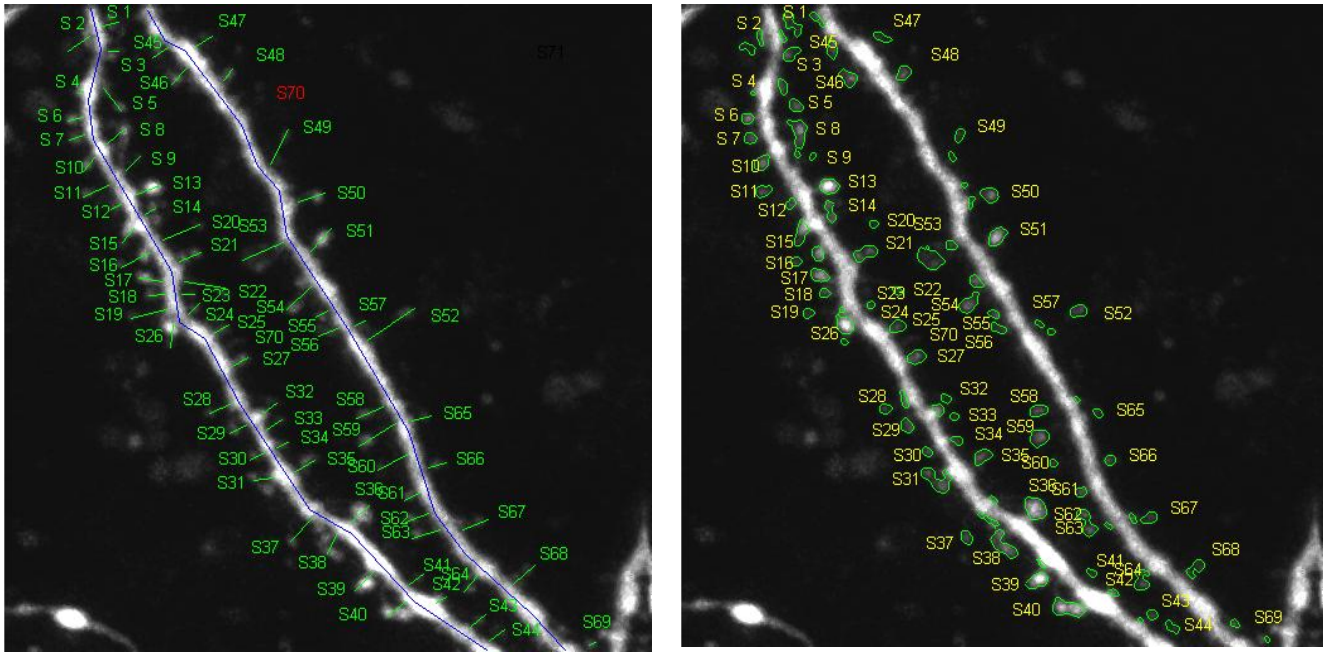
To validate our algorithms, the results obtained by NeuronIQ are compared with the manual results image by image. The results for spine numbers, spine density, and

spine length distribution are compared in detail in this section. The manual detection is performed without knowing the automated detection results. An expert will observe several neighboring slices before marking each spine. The whole image stack is divided into 4 regions. The expert can zoom into each region during the observation. The manual detection and measurement are performed in 3D. Figure 3.8 (a) is just a 2D demonstration of the manual results.

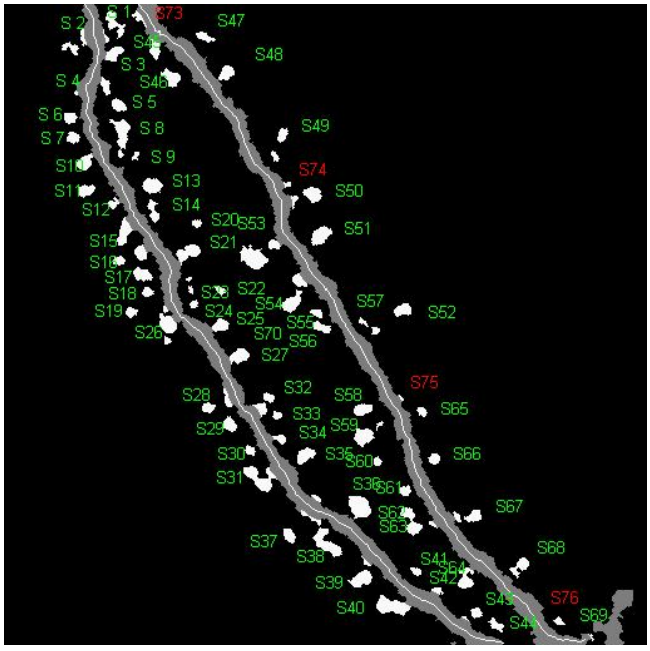
Table 3.2: comparison of manual and automatic results

Method	Num of spine detected	Spine density (number/micron)	False positive	False negative
Manual	70	70/104.675=0.669	0	4
Automatic	73	73/ 110.8388=0.659	0	1

Figure 3.6 shows the manually and automatically detected spines for one image sample. There are totally 69 spines detected both manually and automatically in this image. Compared with the manual results: there is one spine missed (spine number 70), and there are four small spines not manually detected (spine number 73, 74, 75, and 76). The detailed comparison between the automated and manual results is shown in Table 3.2. The reason of missing in automatic detection is mainly because the intensity of the spine is very low compared with the nearby dendrite. As for manual detection, those small size attached spine components are most likely to be ignored by human.

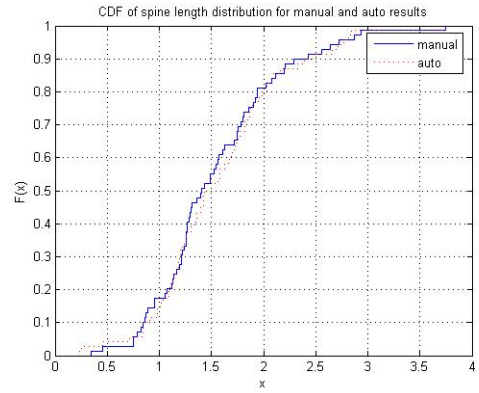


(a) (b)

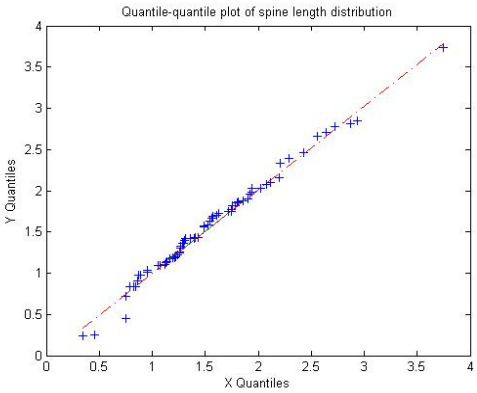


(c)

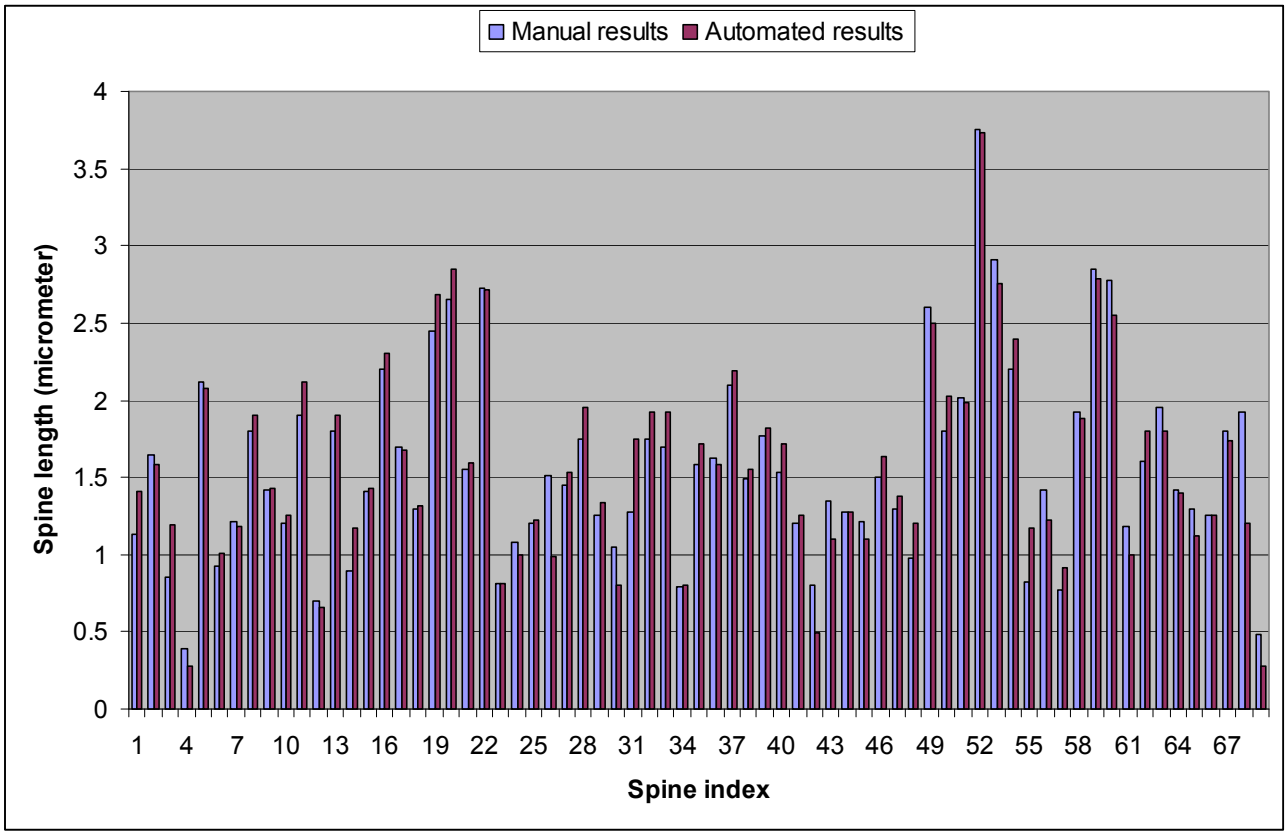
Figure 3.9: comparison of (a) manual results (b) automated spine detection results with edges marked on the original image (3) automated results on the segmented image



(a)



(b)



(c)

Figure 3.10: (a) CDF of spine length distribution for manual and automatic results; (b) empirical quantile-quantile plot (called QQplot) for two distributions; and (c) comparison of spine lengths.

The reason that the manually measured dendrite length is smaller than the automatic result is that the manual result is only an estimation of the actual dendrite length: estimate curve with several connected lines. The detailed spine length distribution can be found in Figure 3.9.

The mean square error (MSE) for the manually and automatically measured spine lengths is 0.0292. The length distributions for manual and automatic results are also tested by two-sample Kolmogorov-Smirnov test. The null hypothesis that the two distributions are the same is not rejected, which means that there is no obvious difference between the two distributions. The probability that the dendrite length distributions of manual and automatic results are the same is 99.13%. The biggest difference between these two distributions is 0.075.

3.7.2 Results analysis based on image set

In addition to the single image comparisons, the manual and automatic results are also compared in image set. We randomly select 16 images in our image dataset [8] for validation purpose. The spine densities are compared first. As we can see from the result shown in Figure 3.11, there is obvious linear relation between the manually and automatically measured spine densities. Two reasons cause the difference between manually measured spine density and the automatic measured results: one is the difference of dendrite length measured, and the other is that the very small spines are likely to be ignored manually.

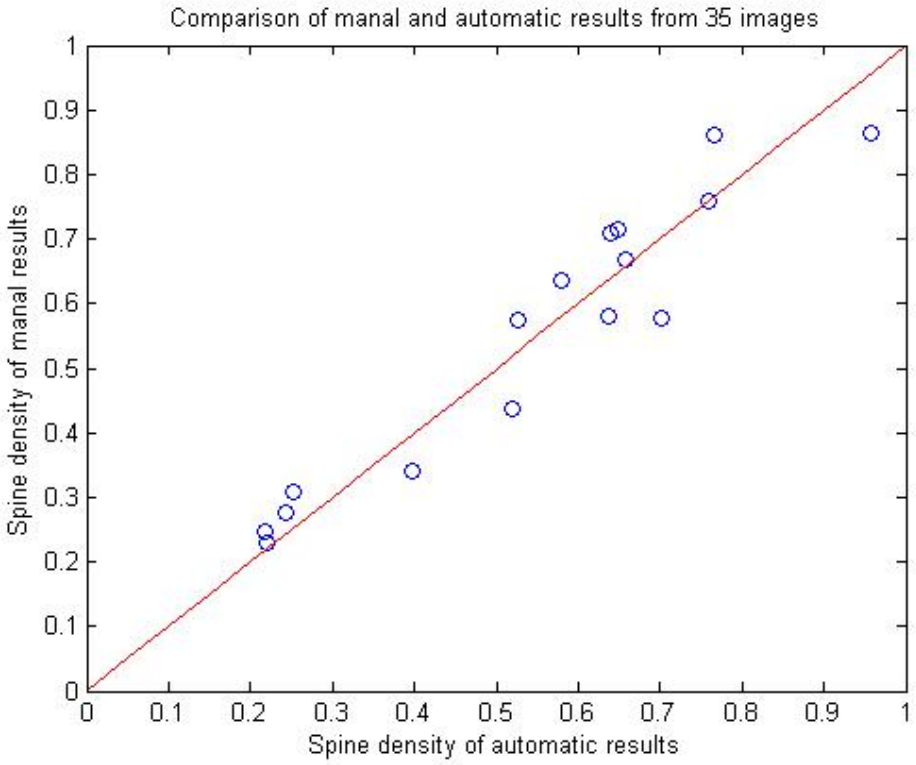


Figure 3.11: comparison of manual and automatic spine density results from 16 images.

Besides comparing with the manual results, we also compare the results obtained by our approach with those obtained by applying Koh's method [1]. Spine detection results of 5 neuron dendrite images are compared (Table 3.3). The 5 image stacks are 5 views from 5 different neurons. The false positives and false negatives are decided by comparing with the manual results. We can see clearly from the results that our method has less wrong detection and missing of spines compared with Koh's method. The missed spines are obviously reduced by using our method.

Table 3.3: comparison of spine detection results

		Our Method		Koh's Method	
Image index	Manually detected spine number	False positives (wrong detection)	False negatives (missing)	False positives (wrong detection)	False Negatives (missing)
1	70	4	1	5	8
2	32	3	1	4	4
3	28	3	0	2	4
4	63	1	0	1	4
5	36	1	3	6	2

3.7.3 Discussion

From the results shown above we can see clearly that our approach has better performance in spine detection compared with other existing methods. The spine length distribution and spine density obtained are also very similar to the manual results. Besides being automatic, our approach also has the following advantages:

- Robust local thresholding can reduce the possibility of missing spines. Spines with relatively low intensity values will not be segmented as background according to a globally set threshold. At the same time, the false positives are reduced because of the control parameter α stated in Equation (3.3).

- The measurement of spine components is more consistent. This is a very important property for the neuroscientists, who are interested in comparing the measurement in different conditions. Nevertheless, it is not the case for the global threshold based algorithms [1,3]: the size of the spines are sensitive to the global threshold.
- The shapes of both attached and detached spine parts are better represented. The base of attached spine components will not protrude into the dendrite. This can be observed from the comparison between our method and Koh's method, in both Figure 3.5 and Figure 3.6.
- For the detection of attached spine components, our method is more robust to noise and the rough dendrite boundary compared with local maximal distance (tips of a protrusion) based algorithms, such as Koh and Xu's approaches [1][3]. This is because these kinds of methods are prone to be trapped in the local maximums because of the noise. On the contrary, our local dendrite thickness estimation method, which relies on the median values of more than ten pixels, is much more robust.

However, there still more work can be done to improve the performance of the proposed algorithms. One interesting topic is how to automatically get the values of some manually set parameters. Table (3.4) is the list of all the manually set parameters. Among these parameters, some (such as M and A_0) are the prior knowledge about the neuron cells, while some (such as α, ε and β) are image dependent and need to be

input by the users (in a range) before the batch processing based on the quality of the images and different imaging conditions.

Table 3.4: Summary of the manually set parameters

Name	Description	Place used	Values
d	width of the window for adaptive thresholding	Eq. (3.2)	[15, 20]
α	control parameter to estimate the minimal intensity value of a possible foreground pixel	Eq. (3.3)	[5, 20]
ε	ratio which decides the suppression intensity	Eq. (3.4)	(0, 1]
M	the longest possible length of a spine	Table (3.1)	[35, 50]
η	weight parameter for SNR analysis	Eq. (3.9)	2
β	threshold for the range of distance to be considered	Eq. (3.10)	[0, 2]
A_0	the smallest possible area of a spine	Eq. (3.11)	5

For the remaining parameters, it is hard to automatically estimate the optimal value of η. The more feasible way is to test images acquired under different conditions and select a satisfying value based on the detection results. We are more interested in how to automatically find the optimal value of the local window size d for the adaptive thresholding. Although a manually set value of d works well for images acquired with the same imaging conditions (e.g., same resolution), there are potential problems if the imaging conditions are changed. Ideally, the size of the window should be a variable which changes its values according to the local texture and intensity distribution, so that

it can be small enough to detect the detail structures while at the same time contains adequate background *and* foreground pixels.

References

[1] I.Y.Y. Koh, W.B. Lindquist, K. Zito, E.A. Nimchinsky, K. Svoboda: An image analysis algorithm for dendritic spines, *Neural computation*, 14: 1284-1310 (2002)

[2] C.M. Weaver, P.R. Hof, S.L. Wearne, W.B. Lindquist, Automated algorithms for multiscale morphometry of neuronal dendrites, *Neural Computation,* 16: 1353-1383 (2004)

[3] X. Xu, J. Cheng, R.M. Witt, B.L. Sabatini, and S.T.C. Wong, A shape analysis method to detect dendritic spine in 3D optical microscopy image, Biomedical Imaging: Macro to Nano, 2006. 3rd IEEE International Symposium on April 6, pp. 554 – 557 (2006)

[4] O.D. Tier and A.K. Jain, Goal-directed evaluation of binarization methods, *IEEE Trans. Pattern Anal. Mach. Intell.* 17: 1191-1201 (1995)

[5] W.Niblack, An Introduction to Image Processing, Prentice-Hall, Englewood Cliffs, NJ (1986)

[6] J. Sauvola and M. Pietaksinen, Adaptive document image binarization, *Pattern Recogn*. 33: 225-236 (2000)

[7] S.D. Yanowitz and A.M. Bruckstein, A new method for image segmentation, Comput. Graph, *Image Process*. 46: 82-95 (1989)

[8] D. Shen and H.H.S. I, A Hopfield neural network for adaptive image segmentation: An active surface paradigm, *Pattern Recogn. Lett*. 18: 37-48 (1997)

[9] F.H.Y. Chan, F.K. Lam, and H. Zhu, Adaptive thresholding by variational method, *IEEE trans. Image Process*. IP-7: 468-473 (1991)

[10] W. Oh and B. Lindquist, Image thresholding by indicator kriging, *IEEE Trans. Pattern Anal. Mach. Intell.* 21: 590-602 (1999)

[11] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis, and Machine Vision, 2nd Edition, Brooks/Cole Publishing Company (1999)

[12] J.M. White and G.D. Rohrer, Image thresholding for optical character recognition and other applications requiring character image extraction, *IBM J. Res. Dev*. 27(4): 400-411 (1983)

[13] J. Bersen, Dynamic thresholding of gray level images, ICPR'86: Proc. Intl. Conf. Patt. Recog., pp. 1251-1255 (1986)

[14] K. Palagyi and A. Kuba, A 3D 6-subiteration thinning algorithm for extracting medial lines, *Pattern Recognition Letters*, 19: 613-627 (1998)

[15] C. Lohou, G. Bertrand, A new 3D 6-Subiteration thinning algorithm based on P-simple points, Discrte Geometry for Computer Imagery: 10 th International Conference, DGCI 2002, France, April (2002)

[16] C.M. Ma and M.Sonka, A fully parallel 3D thinning algorithm and its applications, *Computer Vision and Image Understanding*, 64(3): 420-433 (1996)

[17] C.M.Ma, S. Wan, A medial-surface oriented 3D two-subfield thinning algorithm, *Pattern Recogn. Lett*. 22(13): 1439-46 (2001)

[18] N. Nikolaidis, I. Pitas, 3D image processing algorithm, London: Wiley (2001)

[19] J.H. Chuang, C.H. Tsai, M.C. Ko, Skeletonization of three-dimensional object using generalized potential field, *IEEE Trans PAMI,* 22(11): 1241-51 (2000)

[20] Z. Chen, S. Molloi, Automatic 3D vascular tree construction in CT angiography, *Computerized Medical Imaging and Graphics*, 27: 469-479 (2003)

[21] Y. Zhang, X. Zhou, A. Degterev, M. Lipinski, D. Adjeroh, J. Yuan, S.T. Wong, A novel tracing algorithm for high throughput imaging screening of neuron based assays, J. of Neurosci. Methods, 160: 149-162 (2007)

[22] H. Freeman: Computer processing of line drawing images, *Comput. Surv,*. 6(1): 57-98 (1974)

[23] L. O'Gorman: Primitives chain code, in Progress in Computer Vision and Image Processing, A. Rosenfeld and L.G.Shapiro, eds., pp. 167-183 (1992)

Chapter 4 Orientated Markov Random Field Based Dendritic Spine Segmentation

4.1 Introduction

Manual analyses of neuronal images are extremely time-consuming and subject to investigator bias, i.e., results cannot be easily confirmed by other investigators. To deal with these problems, some automatic dendritic spines analysis algorithms have been provided. Koh et al. propose a geometric approach for automatically detecting and quantifying the structure of dendritic spines [1]. Xu et al. propose a new attached spine component detection algorithm by using two grassfire propagations [2]. Based on Koh's approach, Weaver et al. describe a package which is capable of morphometry on an entire neuron, by combining the spine detection algorithms with dendritic tracing algorithms [3]. Bai et al. [4] used an unsharp mask filter to partly correct the inhomogeneity of image intensity. A global threshold due to a histogram-based strategy is calculated to segment the neuron structures from the background. Zhang et al. [5] apply a curvilinear structure detector to extract the boundaries as well as the centerlines for the dendritic backbones and spines. A classifier using Linear Discriminate Analysis (LDA) is further built to help improve the accuracy of spine detection. In Chapter 3, we propose an automatic spine detection pipeline based on the adaptive thresholding and local dendrite morphology analysis which can effectively improve the detection performance compared with other existing algorithms. Although above mentioned algorithms can greatly reduce the human labor, problems still exist. For instance, the

efficiency of detection will degrade for images with lower resolutions, in which the spines occupy relatively few pixels. Also, these algorithms somewhat distort the shape of the detected spines: spine pixels with low intensity values, such as the pixels in the spine neck regions, are prone to be segmented as background, which cause the spines segmented into several broken components and further degrade spine detection performance. To solve these problems, here we propose a novel maximum a posteriori - orientated Markov random field (MAP-OMRF) framework for dendritic spine segmentation.

As what we have discussed in Chapter 2, the Markov random field (MRF) based segmentation methods have been widely studied for various types of medical images obtained by a wide range of modalities. These include, but are not limited to, cardiac imaging [6]-[7], brain imaging [8]-[12],[23]-[24], digital mammography [13]-[14] acquired by magnetic resonance imaging (MRI), ultrasound, positron emission tomography (PET), and electron microscopy.

Despite the successful applications in medical image segmentation, MRF methods have seldom been used in dendritic spine detection, partially because of the need to specialize these models to address the specific problems previously discussed concerning spine segmentation. To deal with these issues, the OMRF algorithm is proposed by including local orientation information of the spine and dendrite in the prior model. To further reduce the computation complexity and make the algorithm more robust to the background noise, the region of interest (ROI) estimation algorithm, which

is based on iterative spline background correction, is also proposed. The knowledge-based iterated conditional modes (KICM) algorithm is presented as well, which leads to an optimal solution with explicit biological meanings. The chapter is divided into the following parts: First, OMRF model is discussed in detail. How to obtain the orientation map and how to find the ROI are also included in the discussion. The optimization is discussed next, which includes the parameter estimation and a KICM algorithm. Last, the segmentation and detection results are compared with other existing methods and validated manually.

4.2 Oriented Markov Random Field

4.2.1 General Model

Being able to capture the intrinsic character of image by describing the natural spatial constraints in the prior model of images, MRF models are widely applied in various fields of image processing. Usually for MRF-based methods, segmentation is achieved through an optimization process. The optimization function is derived from probabilistic models of the data and the prior information concerning the spatial structure of the segments which is encoded in the MRF. Indeed, one of the most appealing aspects of the MRF formulation is its flexibility in mathematically capturing a wide range of structural prior information for a particular application. For the problem of interest here, the primary source of prior information concerning spines is related to the structure of the dendrite. Indeed knowing the location and orientation of the dendrite places

important geometric and morphological constraints on the associated spines. In this chapter we introduce a new MRF model which considers the inherent orientation information in the neuron cells, viz. OMRF model.

We model the observed grey scale image as a random field $\mathbf{y}$, with the intensity of a pixel at location i denoted by a random variable y_i. The distribution of different regions is denoted by a random field $\mathbf{x}$, where the variable $x_i \in \{1,2,...,M\}$ means that the pixel i belongs to one of the M region types. For the neuron images, M =3, which represents three types of regions of interest in this application: the dendrite, spine and background.

Given the observed data $\mathbf{y}$, the problem of segmentation amounts to the determination of $\mathbf{x}$, that is, the labels associated with each pixel. We pose this problem in a statistical framework as a maximum a posteriori estimation problem where $\mathbf{x}$ is chosen to maximize the posterior distribution of the label based on $\mathbf{y}$ [23]. Formally, by using Bayes theorem this distribution is described as:

$$p(\mathbf{x} \mid \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{x}) p(\mathbf{x}), \tag{4.1}$$

where $p(\mathbf{x})$ is the a priori probability density of the image; $p(\mathbf{y} \mid \mathbf{x})$ is the conditional probability density of the observed image given the distribution of regions. After removal of the shot noise by median filter, the likelihood probability $p(\mathbf{y} \mid \mathbf{x})$ for each pixel is assumed to be independent Gaussian [16]:

$$p(\mathbf{y} \mid \mathbf{x}) \propto \exp\left\{ \sum_{i \in S} -\frac{1}{2\sigma_i^{\,2}} (y_i - \mu_i)^2 \right\}. \tag{4.2}$$

Here S is the collection of all pixels; σ_i is the variance of variable y_i, μ_i is the mean value of y_i. The mean values and the variances are location dependent. This is based on the observation that the pixels of different spines, as well as different dendrite pieces, have different intensity distributions.

The a priori density $p(\mathbf{x})$ is described by the OMRF model. As what we mentioned in Chapter 2, any Markov random field can be described by a probability distribution of the Gibbs form:

$$P(\mathbf{x}) = \frac{1}{Z} e^{-U(\mathbf{x})}, \tag{4.3}$$

where $P(\mathbf{x})$ denotes the probability of configurations of the random field $\mathbf{x}$ and $U(\mathbf{x})$ is the energy function. We notice that the neuron dendrites are orientated structures: the growing direction of each spine is normal to the orientation of local dendrite pieces. This natural orientation information can be combined into the energy function first presented in Section 2.1.4.1:

$$U(\mathbf{x}) = \beta \cdot \sum_{i \in S} \sum_{j \in N_i} D(x_i, x_j)[1 + \alpha \cdot (1 - |\vec{u}_{ij}{}^T \vec{v}_i|)], \tag{4.4}$$

where $\vec{v}_i$ is the orientation of the dendrite in the vicinity of pixel i and will be described in detail in the following section; $\vec{u}_{ij}$ is the direction of the line connecting two neighboring pixels of i and j; α is the weight factor of the orientation information; β is a parameter which controls how strong the spatial regularization is; S is the collection of all pixels; N_i is the 8-neighbor of pixel i; x_i is the segment associated with pixel i; x_j is the segment of neighboring pixels of i; $D(x_i, x_j)$ is the similarity function. If $x_i = x_j$, i.e. pixel

i and j are of the same type, the value of $D(x_i, x_j)$ is set as 0, otherwise the value is set as 1. Thus, pixel i is intended to be in the same region as most of its neighboring pixels, which will reduce the energy $U(\mathbf{x})$. Since logarithm function is monotonous, which does not change the position of the minima, logarithm operation is applied on both sides of Eq. (4.1) to simplify the computation. From Eq. (4.1) - (4.4), we have

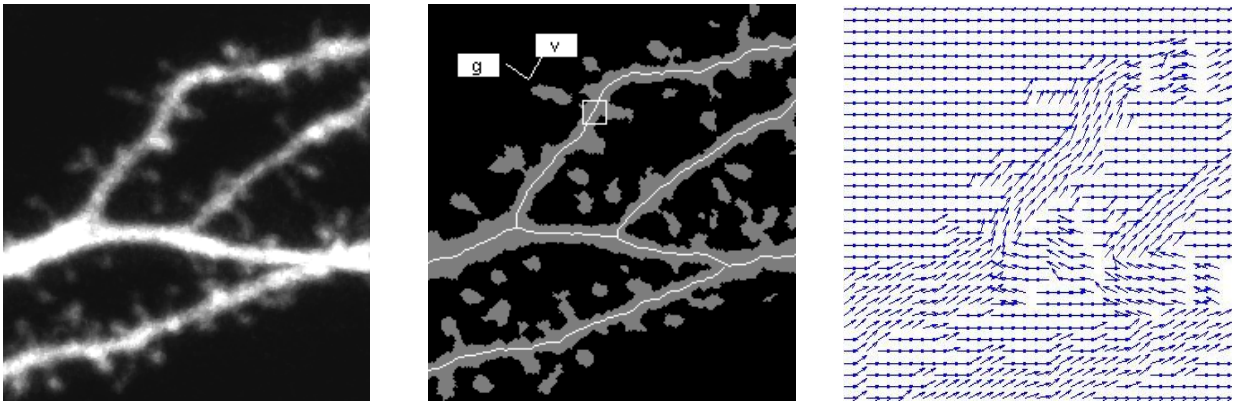
$$\lg p(\mathbf{x} \mid \mathbf{y}) \propto \lg p(\mathbf{y} \mid \mathbf{x}) + \lg(\mathbf{x}),$$

$$\propto \sum_{i \in S} -\{\frac{1}{2\sigma_i^{\,2}}(y_i - \mu_i)^2 + \beta \sum_{j \in N_i} D(x_i, x_j)[1 + \alpha \cdot (1 - |\vec{u}_{ij}{}^T \vec{v}_i|)]\}. \quad (4.5)$$

4.2.2 Orientation map

The use of orientation is motivated by the intuition that the probability of being segmented as an spine pixel should be related to the directions of both $\vec{v}_i$ and $\vec{u}_{ij}$. Suppose the backbone of the dendrite has already been obtained by the thinning and trimming algorithms described in Chapter 3. Assume c_k is a backbone pixel, $R_{c_k} = \{c_{k-2}, c_{k-1}, c_k, c_{k+1}, c_{k+2}\}$ is the local backbone pixels based on which its orientation is estimated by applying linear least square estimation method; γ_{c_k} is the slope of the line estimated, then the value of $\vec{v}_i$ is defined as

$$\vec{v}_i = \{\text{unit vector with slope } \gamma_{c_k} \mid c_k \text{ is the nearest backbone pixel to } i\}. \quad (4.6)$$



(a) (b) (c)

Figure 4.1: (a) original image; (b) initial segment result of the neuron without spine detection; the white curves are the extracted backbone pieces; $\vec{v}$ and $\vec{g}$ are the estimated local dendrite orientation and the growing direction of the spine respectively; (c) sampled orientation map

Fig. 4.1 demonstrates how the orientation map is obtained. The orientation of a pixel is defined as that of the neighboring backbone pixel with the least Euclidian distance, which is estimated in a local window as shown in Fig. 4.1 (b). Fig. 4.1 (c) shows the sampled orientation map. For spine processing, we draw our attention to a region of interest (ROI) surrounding the dendrites, whose estimation will be discussed in the following section. The orientation of pixels outside the ROI is constantly set zero.

4.2.3 Region of interest

In existing spine detection algorithms, physical constraints are applied to improve the detection results. For example, only those detected blobs within a certain distance to the dendrite are considered to be spines. However, many background pixels are still contained in the local regions. These noisy background pixels not only increase the

processing time, but also potentially increase the risk of misdetection. To deal with this issue, we propose a local region estimation method based on iterative spline based background correction algorithm (ISBC) [18]. The ISBC algorithm is initially proposed for solving the problem of large scale intensity variations and shading effects in the image caused by uneven illumination, however, it can also be applied to acquire ROI with minimal modification. Here we want to point out that although local region estimation is quite useful for the MRF based algorithm in the sense of reducing both the processing time and the risk of misclassification, it is not necessary for the adaptive thresholding method discussed in Chapter 3, mainly because of the low computation complexity.

The background is estimated and modeled by a cubic B-spline composed by several patches $S(u,v)$, which is modeled by a tensor product of spline functions [24]

$$S(u,v) = \sum_{kl} B_k(u) B_l(v) c_{kl} \ , \qquad (4.7)$$

where B_k and B_l are the B-spline blending polynomials and c_{kl} are the control points of the surface. 5×5 evenly spread control points are applied here. The spline surface is initially estimated by minimizing the distance between the spline surface and the original image using least squares regressions. Since the much brighter pixels in dendrites and spines are also possible to be included in the spline surface, the initially estimated background should be adjusted. This is done by iteratively masking out the foreground pixels and applying the spline surface fitting in refined regions. The foreground pixels are obtained in the background compensated image. All pixels that are brighter than a threshold, which is related to the standard deviations of the pixel values in the compensated image, are recognized as foreground pixels. The iteration will continue

until the average change of pixel values between two consecutively estimated background is small enough, e.g. less than the original quantization step of the image. As we can see in Fig. 4.2, the foreground pixels contain all dendrite and spine pixels as well as some (but not too many) neighboring background pixels. The possible neck regions of the dendritic spines, which are missed by the existing detection algorithms, are also included.

The ROI are obtained by thresholding the final background compensated image, which is obtained by subtracting the estimated background from the original image. Suppose the standard deviation of the pixel values in the compensated image is σ_c, then the region of interest R is defined by

$$R = \{\text{all pixel } i \mid I_c(i) > k \cdot \sigma_c\}, \tag{4.8}$$

where $I_c(i)$ is the pixel value at i in the background compensated image; k is a factor which is set as 10 during our validation.

Fig. 4.2 shows how the image of ROI is obtained. Fig. 4.2 (b) is the obtained background compensation image. For the purpose of demonstration, a simple linear transformation is applied to transfer it into a grey-scale image with 256 intensity levels. Fig. 4.3 (c) shows the segmented result, i.e., ROI. The ROI is further refined after denoising and removing the far away isolated blobs. The denoising is performed by applying simply morphology operations such as opening. As we can observe from Fig. 4.2 (d), all potential spines, i.e. protrusions along the dendrite boundary, are contained

in the ROI. The obtained ROI is composed of the potential spines together with their surroundings, without including too many background pixels.

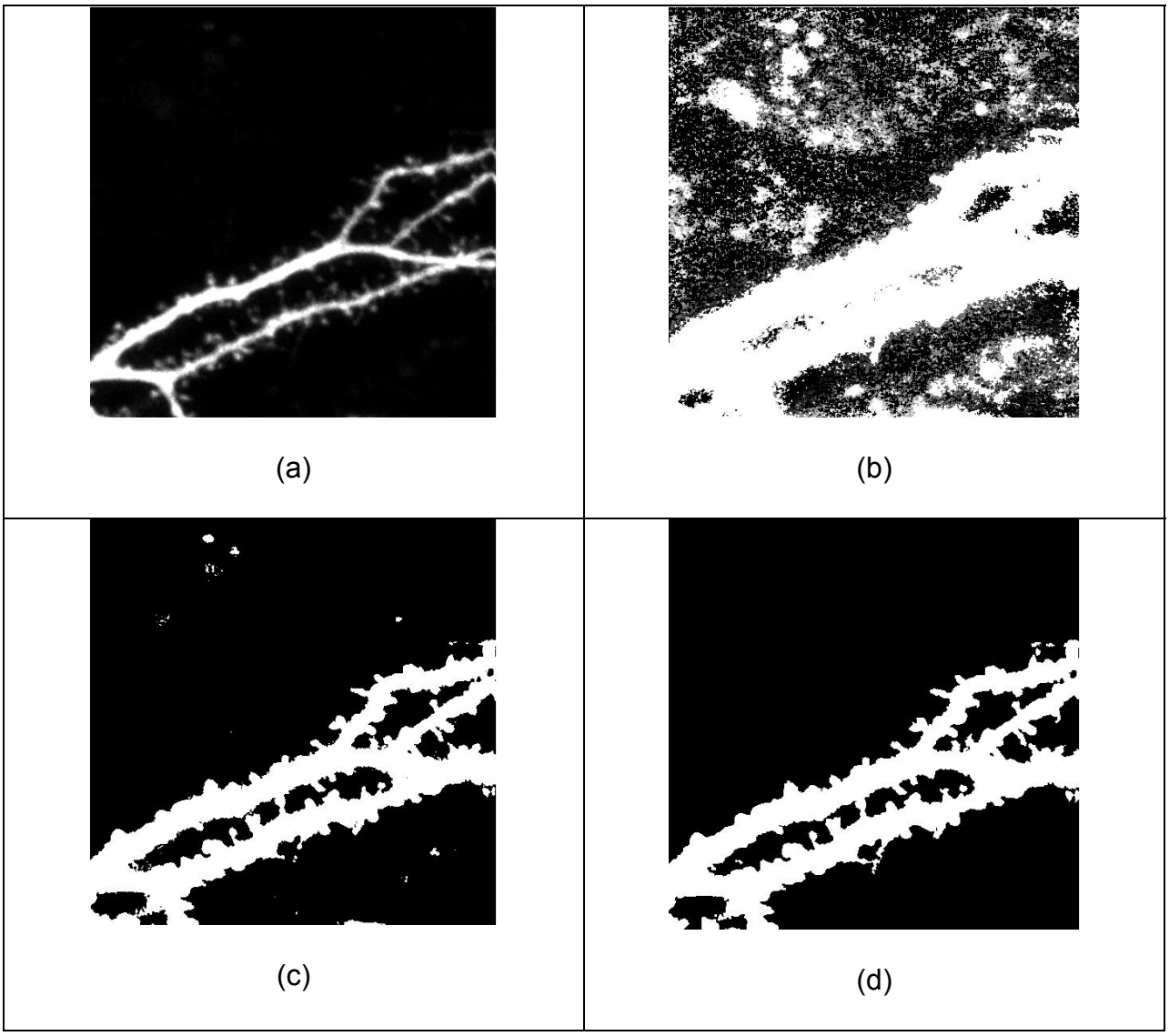


Figure 4.2: local regions of neuron images (a) original image; (b) background compensated image after linear transformation (c) local regions obtained after segmenting the background compensated image; (d) ROI obtained after denoising and removing spurious blobs

4.3 Optimizing Algorithm

In neuron images, there are three different region types, i.e. spine, dendrite and background. For our approach, the segmentation of neuron images is designed as a maximum a posteriori estimation (MAP) problem. Suppose $\mathbf{y}$ is the observed data, the optimal segment results $\hat{\mathbf{x}}$ is found by solving the following MAP estimation problem:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} P(\mathbf{x} \mid \mathbf{y}) . \tag{4.9}$$

To acquire the optimal solution, an extended iterated conditional modes (ICM) algorithm, viz. knowledge-based iterated conditional modes (KICM) algorithm is proposed in this chapter. The interleaved ICM and local parameters updating scheme is applied [22], i.e., estimation of the optimal segmentation $\hat{\mathbf{x}}$ and parameter updating are alternatively performed.

4.3.1 Parameter estimation

Suppose $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ are the optimal parameter estimations for the prior model and the likelihood model respectively. Here $\boldsymbol{\theta}_1 = (\alpha, \beta)$, where the meanings of α and β are described in Eq. (4.4); $\boldsymbol{\theta}_2 = (\varphi_1\ \varphi_2 \ldots \varphi_N)$, where N is the number of total pixels in a neuron image. The parameter vector $\varphi_i = (\mu_i, \sigma_i^2)$ is estimated locally for every pixel i, where $i \in [1,2,\ldots,N]$. As mentioned in Eq. (4.2), μ_i and σ_i stand for the mean value and variance of the intensity distribution of pixel i respectively. For the purpose of simplifying the computation complexity, here $\boldsymbol{\theta}_1$ is set constant. For all images being tested, we

have $\alpha = 0.5$ and $\beta = 0.3$. Assuming the initial segmentation of different regions $\mathbf{x}^{(0)}$ has been obtained by existing algorithms [1]-[5] or the methods described in Chapter 3, the estimation of parameter $\boldsymbol{\theta}_2$ can be acquired based on the likelihood model described by Eq. (4.2): knowing the segmentation results, the intensity of each pixel is assumed to be described by a Gaussian distribution. The parameters of the distributions are segmentation dependent, viz. the intensity distribution of spines, dendrites and the background are different. Moreover, they are also position dependent. This is based on the observation that even in the same image the intensities vary largely for spines or dendrite pieces at different locations. Thus, the mean and variance should be estimated adaptively. Suppose the intensities of the same type pixels in a local window are i.i.d. Gaussians, the posterior probability of $\varphi_i^{(1)}$ at position i for the initial segment $\mathbf{x}^{(0)}$ and observed data $\mathbf{y}$ can be described as:

$$p(\varphi_i^{(1)} \mid \mathbf{x}^{(0)}, \mathbf{y}) = \frac{1}{(\sqrt{2\pi\sigma_i^2})^M} e^{-\sum_{j \in S_i} (y_j - \mu_i)^2 / 2\sigma_i^2}, \qquad (4.10)$$

where S_i is defined as the set of same type pixels in a local 15×15 window around pixel i. The size of the window is chosen based on the normal size of the dendritic spines in the image; M is the total number of pixels in the set. A necessary condition for maximizing $p(\varphi_i^{(1)} \mid \mathbf{x}^{(0)}, \mathbf{y})$, or equivalently maximizing $\ln p(\varphi_i^{(1)} \mid \mathbf{x}^{(0)}, \mathbf{y})$, is $\frac{\partial \ln p}{\partial \mu_i} = 0$ and $\frac{\partial \ln p}{\partial \sigma_i} = 0$. Solving this, we can find the adaptive ML estimates:

$$\hat{\mu}_i = \frac{1}{M} \sum_{j=1}^{M} y_j, \qquad (4.11)$$

$$\hat{\sigma}_i^2 = \frac{1}{M}\sum_{j=1}^{M}(y_j - \hat{\mu}_i)^2 . \quad (4.12)$$

Since the estimation of segmentation results $\mathbf{x}$ is updated iteratively, $\hat{\boldsymbol{\theta}}_2$ is also iteratively estimated.

4.3.2 Estimation with KICM

The optimal segment result $\hat{\mathbf{x}}$ is further estimated after the parameters $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ are obtained. Here an extended iterated conditional modes (ICM) algorithm, viz. knowledge-based iterated conditional modes (KICM) algorithm, is applied to acquire the optimal solution. ICM [20] is based on two assumptions: 1) The observation components $y_1,...y_m$ are conditionally independent given $\mathbf{x}$, and each y_i has the same known conditional density function $p(y_i \mid x_i)$ which depends only on x_i, i.e., $p(\mathbf{y} \mid \mathbf{x}) = \prod_{i \in S} p(y_i \mid x_i)$; 2) The labeling results satisfy the Markovianity: $\mathbf{x}$ depends on the labels in the local neighborhood.

The algorithm sequentially updates the segment result of every point in the image $x_i^{(k)}$ into $x_i^{(k+1)}$ by maximizing the conditional probability $P(x_i^{(k+1)} \mid \mathbf{y}, \mathbf{x}_{S-\{i\}}^{(k)})$, which is performed in a raster scan:

$$x_i^{(k+1)} = \arg\max_{x_i \in L} P(x_i \mid \mathbf{y}, \mathbf{x}_{S-\{i\}}^{(k)}) . \quad (4.13)$$

Here $L = \{1,2,3\}$ represents the three different region types in neuron images. The iteration stops when the number of pixels that change during a cycle is less than a

threshold. The initial estimate $\mathbf{x}^{(0)}$ can be obtained by any spine detection algorithms [1]-[5].

For real applications such as dendritic spine detection, a pure statistically optimized solution is not always the best choice for the purpose of image analysis. For example, for neuron images, an intensity distribution based model is likely to segment a dendrite pixel as a spine pixel because of the similar intensity values. To deal with this issue, more constraints with explicit biological meanings must be incorporated into the processing. In our case, these new constraints are not used to modify the cost function. Instead, they are employed to develop a new strategy of searching the ideal solutions. Unlike the ICM algorithm, the proposed KICM algorithm will stop at some point before reaching the statistically optimal solution. The 'sub-optimized' solution, however, is supposed to possess more biological meaning. For neuron image processing, the local morphology constraints are usually considered. For instance, spines are protrusions along the boundary of a dendrite. Thus, all pixels inside the dendrite should not be segmented as spine pixels.

To better describe the KICM algorithm, we first rewrite the normal ICM algorithm in a matrix form with the transition matrix A being defined. For each pixel i, suppose that $\mathbf{s}^{(k)} = [s_1^{(k)}\ s_2^{(k)}\ s_3^{(k)}]^{\mathrm{T}}$ is the segment result of the k-th step. Here $\mathbf{s}^{(k)} = [1\,0\,0]^{\mathrm{T}}$, $[0\,1\,0]^{\mathrm{T}}$, or $[0\,0\,1]^{\mathrm{T}}$ means the pixel is segment as the spine, dendrite or background respectively. Then we have

$$\mathbf{s}^{(k+1)} = \mathrm{A} \cdot \mathbf{s}^{(k)} = \begin{bmatrix} a(1) & a(1) & a(1) \\ a(2) & a(2) & a(2) \\ a(3) & a(3) & a(3) \end{bmatrix} \cdot \mathbf{s}^{(k)}. \quad (4.14)$$

Here A is the transition matrix, $a(m) = \begin{cases} 1, \text{ if } e_m = \min\{e_1, e_2, e_3\} \\ 0, \text{ if } e_m \neq \min\{e_1, e_2, e_3\} \end{cases}$ and e_1, e_2, e_3 represent the energy if the pixel is segmented as spine, dendrite or background respectively. The energy can be calculated by Eq. (4.5):

$$e_1 = E_s = -\log p(y_i \mid x_i \in spine) + U(x_i \in spine), \quad (4.15)$$

$$e_2 = E_d = -\log p(y_i \mid i \in dendrite) + U(x_i \in dendrite), \quad (4.16)$$

$$e_3 = E_b = -\log p(y_i \mid i \in background) + U(x_i \in background), \quad (4.17)$$

With some modification of the transition matrix A, the KICM algorithm is described as

$$\mathbf{s}^{(k+1)} = \begin{cases} \mathrm{A} \cdot \mathbf{s}^{(k)} = \begin{bmatrix} a(1) & a_{12} & a(1) \\ a_{21} & a(2) & a(2) \\ a(3) & a_{32} & a(3) \end{bmatrix} \cdot \mathbf{s}^{(k)}, & \text{if } \mathrm{A} \cdot \mathbf{s}^{(k)} \neq [0\,0\,0]^{\mathrm{T}}. \\ \mathbf{s}^{(k)}, & \text{otherwise} \end{cases} \quad (4.18)$$

Here the last column of the transition matrix is unchanged, which means if at the k-th step a pixel is segmented as background, at the next step it will be segmented only based on the lowest energy criterion. However, it is not the case if a pixel is formerly segmented as dendrite or spine.

The value of a_{12}, a_{21}, and a_{32} are calculated based on the prior knowledge (assumption) of the neuron image and the initial segmentation results:

a. The incorrectly segmented background pixels are some ‘detached’ noise blobs, which have relatively high intensity values and are likely to be segmented as the

spine. Thus we assume in the initial segmentation results, no background pixels are segmented as dendrite. Based on this assumption we have $a_{32} = 0$, which means that initially segmented dendrite pixels cannot be segmented as the background later.

b. The spines are the protrusions along the dendrite boundary. If we can estimate the width of local dendrite piece, then the regions of potential spines can be obtained: spine pixels can only be those pixels whose distance to the center of the dendrite (the backbone) is greater than half of the local dendrite width. The detailed algorithm to estimate the width of local dendrite piece is discussed in Chapter 3. Based on this assumption, the value of a_{12} and a_{21} can be defined:

$$a_{12} = \begin{cases} 1, \text{ if } e_1 = \min\{e_1, e_2\} \text{ and } d_i > \dfrac{t_i}{2} \\ 0, \text{ otherwise} \end{cases}, \tag{4.19}$$

$$a_{21} = \begin{cases} 1, \text{ if } e_2 = \min\{e_1, e_2\} \text{ and } d_i < \dfrac{t_i}{2} \\ 0, \text{ otherwise} \end{cases}. \tag{4.20}$$

Here d_i is the distance between pixel i and backbone, t_i is the estimation of the width of local dendrite

Let x_i be the segment result for pixel i, $x_{S\setminus i}$ be the current segment result elsewhere, $x_{\partial i}$ is the segment result for the 8-neighbor ∂i of pixel i. For MRF models, we have

$$P(\mathbf{x} \mid \mathbf{y}) = P(x_i \mid y, x_{s\setminus i}) P(x_{s\setminus i} \mid y) = P(x_i \mid y, x_{\partial i}) P(x_{s\setminus i} \mid y). \tag{4.21}$$

We notice that $P(x_i \mid y, x_{\partial i})$ never decreases: x_i is either updated when lower energy is found, or is unchanged when either (a) lower energy can not be found or (b) when $\mathrm{A} \cdot \mathbf{s}^{(k)} = [0\,0\,0]^{\mathrm{T}}$ as described in Eq. (4.18). Besides, $P(x_{s\setminus i} \mid y)$ keeps constant no matter what value x_i is. Thus just like the traditional ICM algorithm, the KICM algorithm performs in a hill-climbing manner and eventual convergence is assured.

(a) (b) (c)

Figure 4.3: comparison between MRF-ICM and OMRF-KICM, (a) original image; (b) segment result with normal MRF-ICM algorithm; (c) segment result with OMRF-KICM algorithm

The KICM algorithm is actually a good complement for the MRF based algorithms, which are based on the spatial constraints of local neighborhood. Normally, the bigger the neighborhood, the better results can be obtained with more local information included. However, larger neighborhood makes the model more complex and greatly increases the computation complex. That is the reason why 4- or 8- neighborhood is normally applied for MRF models in most cases. The KICM algorithm provides a novel perspective on how to combine more local information in a much bigger neighborhood without largely increasing the computation complexity. Together with the orientation

information included in the prior models of the neuron images, the segmentation results can be noticeably improved. Fig. 4.3 shows the comparison of segment results between normal MRF-ICM algorithm and the proposed OMRF-KICM algorithm. We can observe from the results that the shapes of the dendritic spines are much better represented with the "weak" components (spine neck) being enhanced. The wrongly detected spines are also obviously reduced.

As a summary, the main frame of the proposed OMRF-KICM algorithm is described below:

1. Obtain region of interest by background compensation algorithm described in Section 4.2.3
2. Obtain the orientation map described in Section 4.2.2
3. Set the value of $\boldsymbol{\theta}_1$
4. Compute the likelihood probability $p(y_i \mid x_i)$ at position i
5. **Initialize** the algorithm with the segment result $\mathbf{x}^{(0)}$, $k = 0$
6. **Repeat**
7. **Estimate** $\hat{\boldsymbol{\theta}}_2$ with equation (4.11) and (4.12)
8. **Repeat** with $\mathbf{x}^{(k)}$ known
9. **For** each pixel i in the image
10. calculate energy e_1, e_2, e_3 using Eq. (4.15)-(4.17)
11. $\mathbf{s}^{(k+1)} = \mathrm{A} \cdot \mathbf{s}^{(k)}$ using Eq. (4.18)
12. **End**

13. $k = k + 1$

14. **Until** the difference between $\mathbf{x}^{(k)}$ and $\mathbf{x}^{(k-1)}$ is below a threshold

15. **Until** N times of iteration

16. **Return** $\mathbf{x}^{(k)}$ as the optimal segmentation results

4.4 Results and discussion

4.4.1 Validation with existing algorithms

The proposed algorithm is first applied to the initial detection results obtained by existing detection algorithms based on both global thresholding [1] and adaptive thresholding (Chapter 3). As we can see from Fig. 4 and 5, the OMRF based method can efficiently improve the spine detection results for neuron images acquired at different resolutions. For the purpose of demonstration, we only show the maximal intensity projection (MIP) image of the original 3D neuron images.

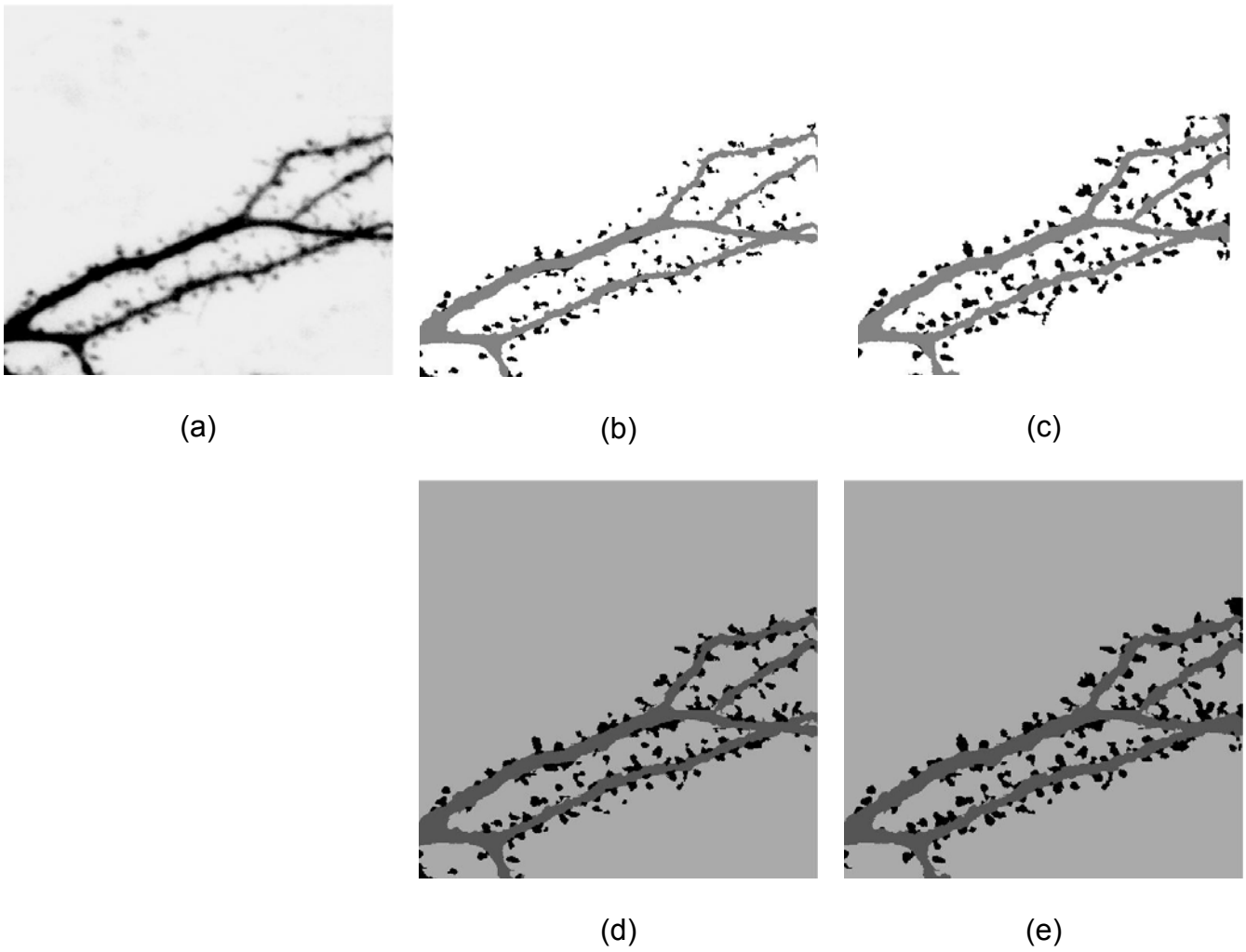


Figure 4.4: Detection results for images on with resolution of 0.08 micron/pixel. (a) MIP images; (b), (d) initial segment results using global thresholding based algorithms and results after applying OMRF model respectively; (c), (e) initial segment results using adaptive thresholding based algorithms and results after applying OMRF model

Fig. 4.4 shows the detections results for neuron images acquired with resolution of 0.08 μm/pixel. From Fig. 4.4 (b) and (d), we can observe that the initial detection results obtained by global thresholding based algorithm are obviously improved: there are

much fewer missing spines after the processing. Also, the false positives are reduced by combining the broken spine components. Generally, with ‘weak’ regions such as the spine neck areas being enhanced, broken spine components are largely reduced, which will potentially increase the accuracy of spine detection and better measurement of spine numbers and spine density. Compared with the global thresholding based methods, the detection results acquired by adaptive thresholding based algorithm is much better with fewer missing spines. However, there are still some false positives caused by the local maximums. The OMRF based method can efficiently remove those positives as shown in Fig. 4.4 (c) and (e).

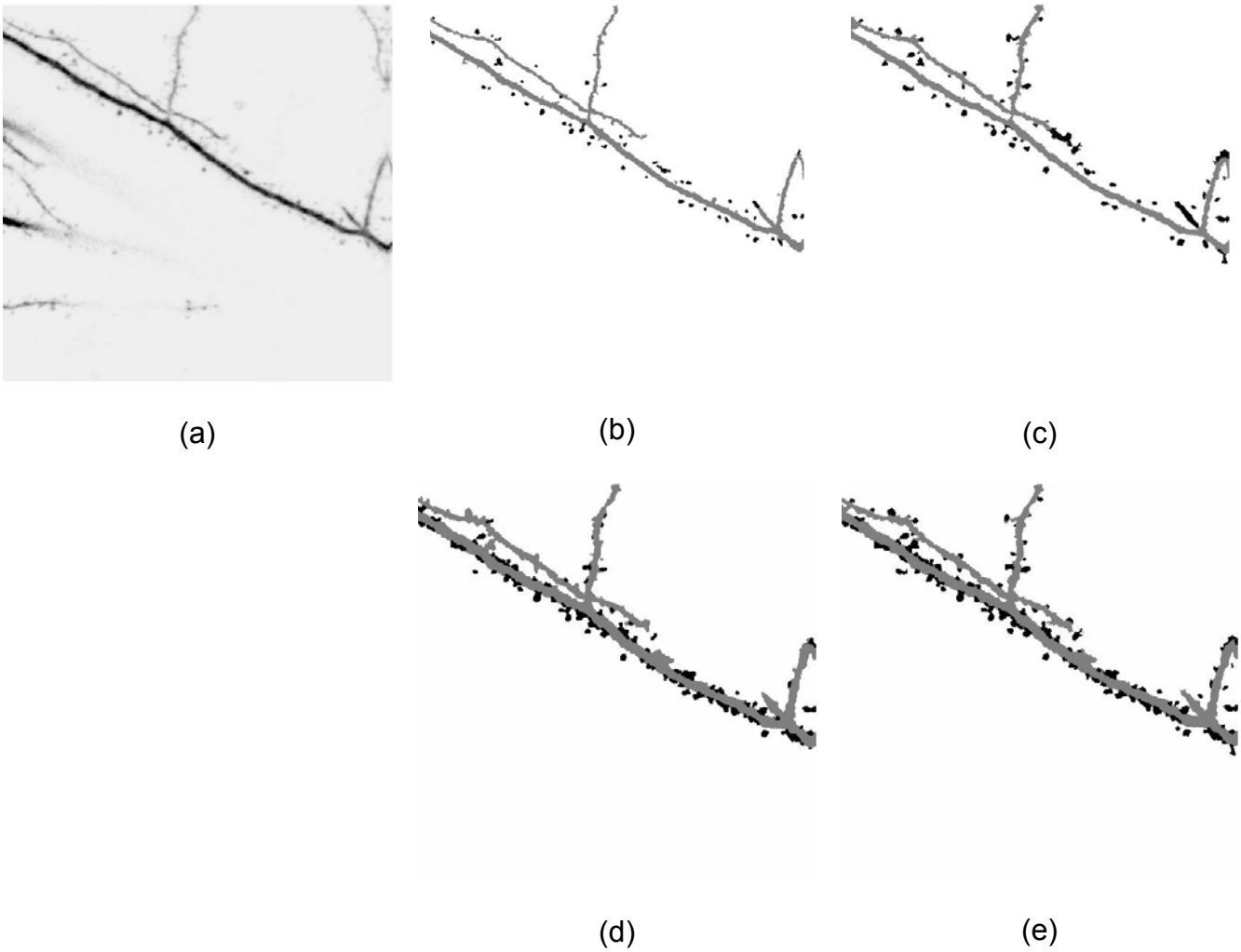


Figure 4.5: Detection results for images with resolution of 0.125 micron/pixel. (a) original MIP images; (b), (d) initial segment results using global thresholding based algorithms and results after applying OMRF model respectively; (c), (e) initial segment results using adaptive thresholding based algorithms and results after applying OMRF model

For existing automatic dendritic spine detection algorithms, the accuracy of detection might noticeably degrade for images acquired at lower resolutions, where the size of the spines will become much smaller. There are mainly two considerations to improve the detection performance for lower resolution images. First, larger observation view can be

obtained for a lower resolution image. Second, there are large amounts of experimental data acquired at low resolution, it would be very time consuming and a waste of resource if all the images have to be acquired again at higher resolutions. As shown in Fig. 4.5, for some neuron images acquired at lower resolution, both global and adaptive thresholding based algorithms cannot efficiently detect the 'weak' spines with relative lower intensity. This problem can be partially solved by interpolation. However, the processing time will be tremendously increased with much bigger image size. The proposed OMRF based detection algorithm can nevertheless solve the missing problem well, as shown in Fig. 4.5 (d) and (e).

4.4.2 Validation with manual results

Automatic dendritic spine detection results, with and without OMRF processing, are first compared with the manual results. Initial results are acquired by both global thresholding and adaptive thresholding based methods. Five neuron images with different resolutions (0.125 micron/ pixel and 0.08 micron/pixel) are compared. As we can see from Table 4.1, both FP (false positives, i.e., wrong detections) and FN (false negatives, i.e. missings) are obviously decreased after the processing of the proposed method.

Table 4.1: comparison of spine detection results

	Image 1	Image 2	Image 3	Image 4	Image 5
Manually detected spine number	82	67	86	93	36
FP (Global thresholding)	8	1	12	4	2
FN (Global thresholding)	19	15	22	11	4
FP (Global thresholding + OMRF)	3	0	5	3	0
FN (Global thresholding+OMRF)	9	7	8	6	2
FP (Adaptive thresholding)	3	2	6	5	0
FN (Adaptive thresholding)	10	13	15	6	3
FP (Adapt. thresholding +OMRF)	2	0	3	2	0
FN (Adapt. thresholding +OMRF)	6	6	5	2	2

Further validation is performed by comparing the spine length distribution of the proposed method with the manual results. Spines in different section of the same neuron cell under the same condition (shLUCI in hippocampal pyramidal neurons in rat organotypic slice cultures) are tested. There are altogether 235 spines in three different images. The two distributions are tested by two-sample Kolmogorov-Smirnov test. The null hypothesis that the two distributions are the same is not rejected, which means that there is no obvious difference between the two distributions. The probability that the dendrite length distributions of manual and automate results are the same is 99.13%. The biggest difference between these two distributions is 0.075.

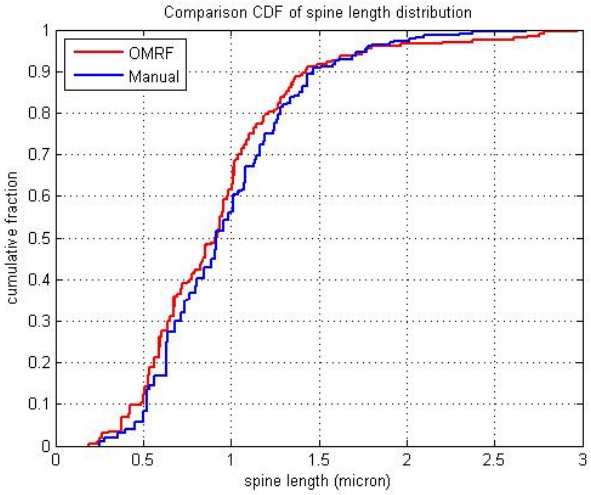


Figure 4.6: Comparison of spine length distribution of manual results and results obtained by OMRF method.

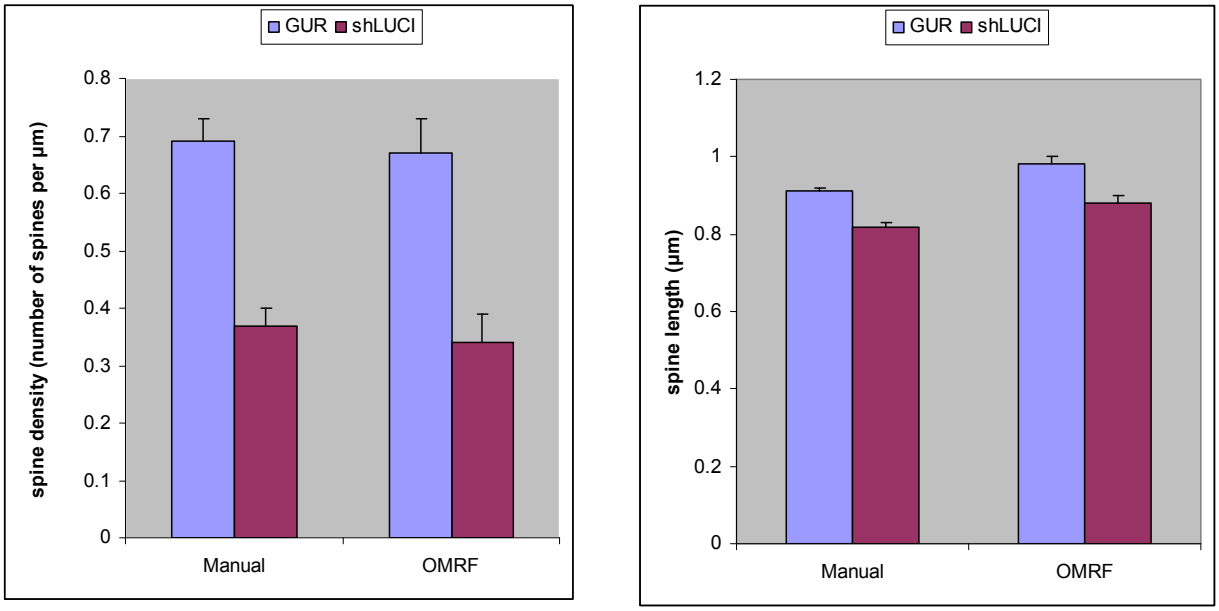


Figure 4.7: shRNA expression induces retraction of dendrites. Method 1 and 2 are manual and OMRF results respectively. a). Summary of data of average spine density at 4DPT for GUR and shLUCI neurons. b). Summary of data of average spine length at 4DPT for GUR and shLUCI neurons.

The proposed method is also validated by the published manually obtained data. This recently published paper [15] examines the off-target effects of expressing short hairpin RNAs (shRNA) in neurons. The study shows that expression of shRNA against luciferase, whose coding region is not found in the rat genome, triggers dramatic loss of dendritic spines and simplification of dendritic arbors. Figure 4.7 shows the difference of average spine densities and average spine length for GUR and shLUCI neurons at 4DPT. OMRF analysis detects a similar reduction of spine density and length by shLUCI as those identified by the manual analysis. These results, therefore, validate the proposed algorithm as a valuable tool for automatic analysis of neuronal morphology and the identification of biologically relevant changes in dendritic spine morphology and number

4.4.3 Conclusion and Discussion

In this chapter we propose a novel OMRF-KICM based method which can efficiently improve the automatic detection results of existing algorithms. With the intensity distribution information of background, dendrite and spine pixels, as well as the context and orientation information of the neurons, OMRF-KICM method can obviously improve the detection performance for image acquired at different resolutions. It is shown that the false positives and the false natives of spine detection can be decreased by more than 50% after applying the proposed method. Furthermore, it also has the promising potentials in the fidelity of shape representation, i.e., the shape of detected spines

resembles more to their real shapes. The weak parts of the spines, especially the neck regions which are hard to detect, can now be captured.

The proposed algorithms are 2D based and all the images being tested are MIP images. This is based on the following considerations: 1) the resolution in z- direction is much lower compared with those in x- and y- directions. For spines which are big enough, the 3D information in neighboring slices is helpful for the detection. However, there are many small spines which are so small that they only appear in one slice. Not enough 3D information can be obtained about them. The same is true with the spine necks which are much thinner compared with the head components. Since the purpose of our algorithm is to solve the detection problem of small spines and the spine necks, we only propose 2D algorithms. 2) Our neurobiology collaborators are more interested in the comparison between the measurements under difference conditions. As long as the measurements can clearly state the changes, no precise measurements are needed.

References

[1] I. Y. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, and K. Svoboda, An image analysis algorithm for dendritic spines, *Neur. Comput*. 14: 1283–310 (2002)

[2] X. Xu, J. Cheng, R. M. Witt, B. L. Sabatini, S. T.C. Wong, A shape analysis method to detect dendritic spine in 3D optical microscopy image, *Biomedal Imaging: Macro to Nano*,. 3rd IEEE International Symposium, Apr. 6, pp554 – 557 (2006)

[3] C. M. Weaver, P. R. Hof, S. L. Wearne, W. Lindquist, Automated algorithms for multiscale morphometry of neuronal dendrites, *Neur. Comput.* 16: 1353-83 (2004)

[4] W. Bai, X. Zhou, L. Ji, J. Cheng, and S. T. Wong, Automatic dendritic spine analysis in twophoton laser scanning microscopy images, *Cytometry A* 71: 818-826 (2007)

[5] Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjeroh, and S. T. Wong, Dendritic spine detection using curvilinear structure detector and LDA classifier. *Neuroimag*. 36: 346–360 (2007)

[6] G. Xiao, M. Brady, and J. A. Noble, Segmentation of ultrasound B-mode images with intensity inhomogeneity correction*, IEEE Trans. on Med. Imag*. Jan., 21(1): 48-57 (2002)

[7] A. Juslin, and J. Tohka, Unsupervised segmentation of cardiac PET transmission images for automatic heart volume extraction, *Proc. of the 28th IEEE EMBS ann. int. conf.*, Aug., 1:1077-80 (2006)

[8] S. Ruan, and D. Bloyet, MRF models and multifractal analysis for MRI segmentation, *Proc. of ICSP*, pp. 1259-62 (2000)

[9] Y. Zhang, M. Brady and S. Smith, Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm, *IEEE Trans. on Med. Imag.*, Jan., 20(1): 45-57 (2001)

[10] V.D. Calhoun, T. Adali, G.D. Pearlson, and J.J.Pekar, Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms, *Hum. Brain Mapp.*, 13: 43-53 (2001)

[11] X. Descombes, F.Kruggel, and D.Y. von Cramon, Spatio-temporal fMRI analysis using Markov random fields, *IEEE Trans. Med. Imag.*, Dec., 17(6): 1028-39 (1998)

[12] E. Salli, H.J. Aronen, S. Savolainen, A. Korvenoja, and A. Visa, Contextual clustering for analysis of functional MRI data, *IEEE Trans. Med. Imag.*, May, 20(5): 403-14 (2001)

[13] C. D'Elia, C. Marrocco, M. Molinara, G. possi, G. Scarpa, and F. Tortorella, Detection of microcalcifications clusters in mammograms through TS-MRF segmentation and SVM-based classification, *Proc. of the 17th int. conf. on Patt. Recog*. 3: 742-45 (2004)

[14] H.D. Li, M. Kallergi, L.P. Clarke, V.K. Jain, and R.A. Clark, Markov random field for tumor detection in digital mammography, *IEEE Trans. on med. Imag.* Sep., 14(3): 565-576 (1995)

[15] Veronica A. Alvarez, Dennis A. Ridenour, and Bernardo L. Sabatini, Retraction of synapses and dendritic spines induced by off-target effects of RNA Interference, *J. of Neuroscience*, 26(30): 7820-25 (2006)

[16] T.J. Hebert, Fast iterative segmentation of high resolution medical images, *IEEE Trans. on Nuc. Sci.*, 44(3): 1362-67 (1997)

[17] S.Z. Li, Markov Random Field Modeling in Image Analysis, 2nd ed. Tokyo, Japan: *Springer-Verlag* (2001)

[18] C. Wahlby, J. Lindblad, M. Vondrus, E. Bengtsson and L. Bjorkesten, Algorithms for cytoplasm segmentation of fluorescence labeled cells, *Analytical Cellular Pathology*, 24(2-3): 101-11 (2002)

[19] S. Geman and D. Geman, Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images, *IEEE Trans. on PAMI*, 6(6):721-41 (1984)

[20] J. Besag, On the Statistical Analysis of Dirty Pictures, *J. of the Royal Statistical Society*, Series B, 48(3):259-302 (1986)

[21] J. Besag, Towards Bayesian Image Analysis, J. of Applied Statistics, 16(3): 395-407 (1989)

[22] T. N. Pappas, An adaptive clustering algorithm for image segmentation, *IEEE Trans. on Signal Processing*, 40(4):901-914 (1992)

[23] J.C. Rajapakse, Y. Wang, X. Zheng, J. Zhou, Probabilistic Framework for Brain Connectivity From Functional MR Images, *IEEE Trans. on Med. Imag.*, 27(6):825 - 833 (2008)

[24] S.P. Awate, R.T. Whitaker, Feature-Preserving MRI Denoising: A Nonparametric Empirical Bayes Approach, *IEEE Trans. on Med. Imag.*, 26(9):1242 - 1255 (2007)

Chapter 5 Tracking in time-lapse neuron images

5.1 Introduction

Dendritic spines have a specialized cytoskeleton composed of a dense network of actin filaments. They are highly dynamic and change their shapes constantly [1]-[3]. The rapid and spontaneous changes of dendritic spines have been widely observed in vivo and cultured neurons. Experiments have revealed that the morphologic plasticity of dendritic spines is correlated with synaptic activity, sensory experience, learning, memory, and drug addiction [4]-[10]. With the development of high-sensitivity charge-coupled device (CCD) video cameras and automated image capture and analysis facilities, the scope and practicality of high-resolution time-lapse microscopy have been enormously extended. It has become possible to study aspects of plasticity relating to the growth or retraction of individual spines and the dynamics that underlie spine plasticity.

Currently, the analysis of dendritic spines morphology in time lapse images remains largely manual. Parameters defining the geometry of each spine, such as the length and width, are manually measured with the aid of commercial or in-house developed software through a point-and-click interface [4]-[10]. In recent years, several approaches have been proposed to detect and measure dendritic spines automatically [11]-[18]. However, the time-lapse analysis functions are provided in only a few of these approaches. For example, Koh et al propose a simple spine tracing algorithm by finding

the least absolute image difference summation [13]; two spines scanned at different times are considered to be the same if they overlap sufficiently. Mosaliganti et al devise a more sophistic temporal matching method to track the evolution of dendritic spines [16]. In their method, a graph model spatially aligned with the dendrite is described, where the nodes represent the spine branches, and then a maximum a posteriori (MAP) – maximum likelihood estimator (MLE) matching framework is employed to match graph models from different time points.

Both of the aforementioned algorithms are global matching methods based on the spine detection result of time-lapse images. These methods have the following shortcomings: first, matching results heavily depend on the quality of the images and the performance of the implemented detection algorithms. It is infeasible to match poorly segmented spines in low quality images. Second, detection and matching have to be performed for all spines in the whole image because of the global matching process. Obviously, it is not an efficient way for tracking if we are more interested in analyzing the morphological changes for part of the spines over time [40][41].

To address the above problems, in this chapter we propose a novel tracking algorithm which can effectively trace the morphologic changes of specific dendritic spines in time lapse microscope, and is served as the supplementary function of NeuronIQ. The algorithm is composed of two parts to respectively deal with two types of movement in time-lapse neuron images: the global displacement of the observation view, which is introduced during image acquisition; and the deformation of the highly

dynamic dendritic spines, whose changes in the size, shape, or orientation can be observed even between images acquired within a short time interval.

The global displacement can be adjusted by rigid registration, which is usually implemented by matching the dendrites. The skeleton of the dendrite, i.e., the backbone, has been commonly used to represent the dendrite structure [11]-[17], which is extracted by pruning the small branches from the medial axis of the dendrite. ICP (iterative closest points) algorithms have been well studied and widely applied for the registration of free curves. Many variables of the algorithm have been proposed since it was firstly introduced by Besl et al [30] and Chen [19]. A thorough review can be found in Rusinkiewicz and Levoy's paper [20]. These algorithms either focus on selecting the sampling and matching points [21]-[24] or trying to refine the search scheme [25]-[29]. However, although these variables can improve the basic ICP with a faster convergence speed and better registration results, they all require that the motion between two successive frames be in some sense "small," which is not the case for neuron images. Based on the prior knowledge that we are interested specifically in matching neuronal dendrite images, we propose a revised ICP algorithm which can effectively relax the small range requirement.

This chapter is divided into the following parts: First, we propose the ICP algorithm with feature selection for global registration. Then, the particle filter based algorithm which can track and measure the dendritic spines simultaneously is described in detail.

Last, the tracking results are demonstrated and compared with existent methods. The measurements of the spine morphologic changes are also manually validated.

5.2 Algorithms

The ICP based algorithm can be easily trapped in local minima. Thus a small displacement between two images is required for the satisfying registration results. In Section 5.2.1, we propose a feature point sets selection algorithm specially designed for neuron images registration, which can efficiently relax the small displacement requirement.

Compared with the relatively stable dendrites, the deformation of the dendritic spines is much more severe. Changes in shape, size, or orientation of the spines can be observed in neuron images acquired in consecutive time frames. To deal with this issue, in Section 5.2.2 a particle filter based algorithm is proposed to track these highly dynamic spines after the global dendrite registration. The measurements of the spine shape, such as the length, width, area, and orientation, can be conveniently obtained from the dynamic models.

5.2.1 Global registration

5.2.1.1 Iterative closest point algorithm

The details of the basic ICP algorithm can be found in a paper by Besl et al [18]. Suppose the backbones in consecutively acquired images correspond to two different data point sets, $Y = \{\vec{y}_i\}$ and $X = \{\vec{x}_i\}$, to be aligned. The least squares registration can be found by minimizing the mean square objective function

$$f(\mathbf{R}, \vec{q}) = \frac{1}{N_y} \sum_{i=1}^{N_y} \| \vec{x}_i - \mathbf{R} \cdot \vec{y}_i - \vec{q}_T \|^2 \qquad (5.1)$$

.

In Eq. (5.1), N_y is the number of points in Y and the closest points in X. The point $\vec{x}_i$ is defined as the closest point to $\vec{y}_i$ if $D(\vec{x}_i, \vec{y}_i) = \min_{\vec{x}_j \in X} D(\vec{x}_j, \vec{y}_i)$, where D represents the Euclidean distance between two points. Below we will briefly describe a quaternion-based algorithm to find the rotation matrix $\mathbf{R}$ and the translation vector $\vec{q}_T = [q_4 q_5 q_6]^t$, where t is the vector transpose operator.

The 3×3 rotation matrix $\mathbf{R}$ can be calculated by a unit rotation quaternion $\vec{q}_R = [q_0 q_1 q_2 q_3]^t$:

$$\mathbf{R} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 + q_2^2 - q_1^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 + q_3^2 - q_1^2 - q_2^2 \end{bmatrix}. \qquad (5.2)$$

The unit quaternion is a vector which satisfies $q_0 \geq 0$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. It can be obtained as the eigenvector corresponding to the largest eigenvalue of a symmetric 4×4 matrix:

$$\mathbf{Q}(\mathbf{\Sigma}_{yx}) = \begin{bmatrix} tr(\mathbf{\Sigma}_{yx}) & \mathbf{\Delta}^T \\ \mathbf{\Delta} & \mathbf{\Sigma}_{yx} + \mathbf{\Sigma}_{yx}^T - tr(\mathbf{\Sigma}_{yx}) \mathbf{I}_3 \end{bmatrix}. \qquad (5.3)$$

The cross-covariance matrix $\mathbf{\Sigma}_{yx}$ of $\mathbf{X}$ and $\mathbf{Y}$ is given by

$$\mathbf{\Sigma}_{yx} = \frac{1}{N_y}\sum_{i=1}^{N_y}[(\vec{y}_i - \vec{\mu}_y)(\vec{x}_i - \vec{\mu}_x)^t] = \frac{1}{N_y}\sum_{i=1}^{N_y}[\vec{y}_i\vec{x}_i^t] - \vec{\mu}_y\vec{\mu}_x^t \qquad (5.4)$$

,

where $\vec{\mu}_X$ and $\vec{\mu}_Y$ are the centroids of the point set $\mathbf{X}$ and $\mathbf{Y}$ respectively:

$$\vec{\mu}_X = \frac{1}{N_x}\sum_{i=1}^{N_x}\vec{x}_i \text{ and } \vec{\mu}_Y = \frac{1}{N_y}\sum_{i=1}^{N_y}\vec{y}_i \qquad (5.5)$$

.

The column vector $\mathbf{\Delta}$ is defined as $\mathbf{\Delta} = [\mathbf{A}_{23} \ \mathbf{A}_{31} \ \mathbf{A}_{12}]^T$, where the matrix component is calculated from the cross-covariance matrix: $\mathbf{A}_{ij} = (\mathbf{\Sigma}_{yx} - \mathbf{\Sigma}_{yx}^T)_{ij}$.

After the rotation matrix $\mathbf{R}$ is obtained, the translation vector can be calculated by

$$\vec{q}_T = \vec{\mu}_X - \mathbf{R}\cdot\vec{\mu}_Y \qquad (5.6)$$

.

5.2.1.2 Feature points selection

The ICP algorithm has been proved to be capable of converging monotonically to a local minimum of Eq. (5.1) [18]. One possible method for obtaining the "best" local minimum would be to perform ICP over a range of initial guesses for the rotation matrix and translation vector using some type of coarse-grid search. The best result from all the local minima found is then selected as the final solution. Certainly, it is not an efficient way to randomly search in a high dimensional space. To solve this problem, we redesign the sampling step of the ICP algorithm based on the prior knowledge of the neuron images. The sampling is based on the assumption that not all the backbone pixels are equally important for the registration. Thus, instead of choosing all or

randomly picking a subset of backbone pixels, only the feature points (to be defined below) are picked for use in Eq. (5.1). Properly selected feature points can greatly reduce the probability of being trapped in a local minimum, just like it is much easier to match the corner points than to match the whole surface. The feature points are chosen based on the following criteria.

Suppose Φ is the collection of all the backbone pixels in an image. An ideal feature point set ϕ should satisfy the following criteria:

1. Robust: the feature points should be robust to the noise and possible segmentation and detection errors.
2. Localized: any feature points of current image should not be "too" far from the feature point set of the previous image. Formally, let ϕ_k be the feature point set of image acquired at time k and $D(i, j)$ be the Euclidean distance between pixel i and j. For every pixel $i \in \phi_{k+1}$, we require

$$\min_{j \in \phi_k} D(i, j) \leq \beta \quad . \tag{5.7}$$

   The threshold β can be set as the largest possible displacement between two neighboring image frames. Here $\beta = 20$ (pixel).
3. Evenly sampled: to well represent the location and structure of the dendrites, the feature points should be evenly distributed along the backbone. This is measured by the distance between the centroids of the feature point set ϕ and backbone set Φ. If the distance is within a threshold, we believe that the feature points are evenly distributed along the backbone. Thus, we have

$$D(C_\phi, C_\Phi) \le \alpha \max_{i,j \in \Phi} D(i,j), \qquad (5.8)$$

where C_ϕ is the centroid of the feature points, C_Φ is the centroid of the backbone, and α is a ratio parameter which is set as 30%. Note that in theory this criterion could be satisfied if all of the feature points were selected close to the backbone; however, this is unlikely to happen since the (stable) spines usually distribute along the whole dendrite piece being observed.

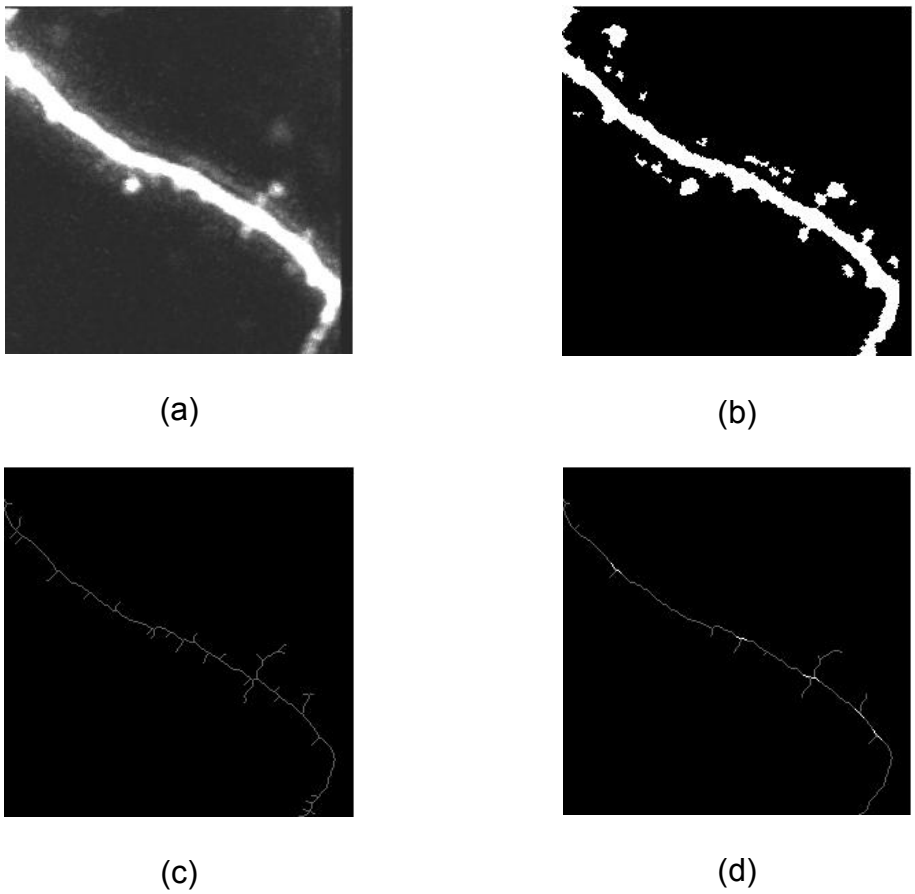


Fig. 5.1 (a) original image, (b) results after adaptive segmentation, (c) medial axis obtained without filtering, and (d) medial axis obtained after low-pass filtering; extracted feature points are the highlighted backbone pieces.

We observe that the locations where the spines attached to the dendrite are relatively stable regardless of dendrite deformation and morphologic changes of the spines. Nearby backbone pieces around the attached spines seem to be good candidates for the feature points. More specifically, possible location of these points can be associated with small branches in the medial axis of the dendrite, which can be obtained by the method described in Chapter 3. A sample image of acquired medial axis of the dendrite is shown in Fig. 5.1 (c), which is obtained from the adaptive segmentation results shown in Fig. 5.1 (b). The details of the segmentation method can be found in Chapter 3. As we can observe from Fig. 5.1 (c), many of those small branches are caused by the rough dendrite boundary, instead of attached spines. A low-pass filter can be applied to smooth the image before the processing to help remove this kind of small branches. Here a 5×5 median filter is applied. The potential feature points are selected as the backbone pixels in a local 9×9 windows around the branch points. The branch point is defined as the conjunction point of the small branch and the backbone obtained from a medial axis computation applied to the post-filtered image data. One example of the extracted feature points is demonstrated in Fig. 5.1 (d). It is possible that some weak dendritic spines are also removed after the smoothing. Nevertheless it is not necessary to find all the attached spines for the purpose of feature point selection. The selected backbone pieces are the feature point set as long as they satisfy all the three criteria.

As the central line of the dendrites, the backbone is insensitive to the rough dendrite boundary caused by the noise. However, sometimes large spines can be wrongly

segmented as small dendrite branches, and this would cause problem during the dendrite registration. This problem can be solved if only the backbone pieces around an attached spine are selected. The pixels in the central line of the large spine, which is wrongly recognized as part of the dendrite backbone, will not be selected as the feature points since no other spines are attached to it.

To satisfy the second criterion, the selected backbone pieces around the attached spines are further processed. All the backbone pixels that are not within a certain distance from the feature points in the previous image are removed. For the first image, all the selected backbone pixels are assumed as the feature points. During this step, only the backbone pixels around the stable spines are selected as the feature points. For example, backbone pixels near the newly born or disappeared spines are removed.

The evenly distributed criterion can be easily satisfied during our tests because of the evenly distribution of spines along the dendrite. In case that the requirement cannot be satisfied, the size of the smoothing window can be reduced so that fewer small branches are removed and more backbone pixels can be potentially selected as the feature points.

Suppose $Y = \{\vec{y}_i\} \equiv \phi_k$ is the data set, $X = \{\vec{x}_i\} \equiv \phi_{k-1}$ is the model set to be registered, and P is a subset of X, which is composed of the nearest neighbors of Y. The pixel $\vec{p}$ is defined as the nearest neighbor of pixel $\vec{y} \in Y$ if

$$D(\vec{p}, \vec{y}) = \min_{\vec{x} \in X} \| \vec{x} - \vec{y} \| \quad (5.9)$$

.

The brief structure of the ICP algorithm with feature point selection can be stated as follows:

For all images at time $k = 1,..., K$

1. **Feature point selection**
    a. Segmentation and acquire the medial axis of the dendrites using methods in Chapter 3
    b. Obtain backbone pieces in the local region of the small dendrite branches, which are the remaining pieces after removing the backbone from the medial axis, using methods in Chapter 3
    c. Refine the results by criterion 2, according to Eq. (5.7)
    d. Test the results by criterion 3, according to Eq. (5.8)

    If satisfied

    Obtain the feature point set ϕ_k

    If not

    Reduce the size of the smooth window; go to 1(b)

    End

2. **Registration**

    If $k = 1,$ no need for registration: $k = k + 1,$ go to 1(a)

    a. Initialization:

    $X \equiv \phi_{k-1},\ Y_0 \equiv \phi_k,\ \vec{q}_0 = [1, 0, 0, 0, 0, 0, 0]^{\mathrm{T}}$

    b. **Repeat if** change in mean square error falls below a preset threshold $\tau > 0$

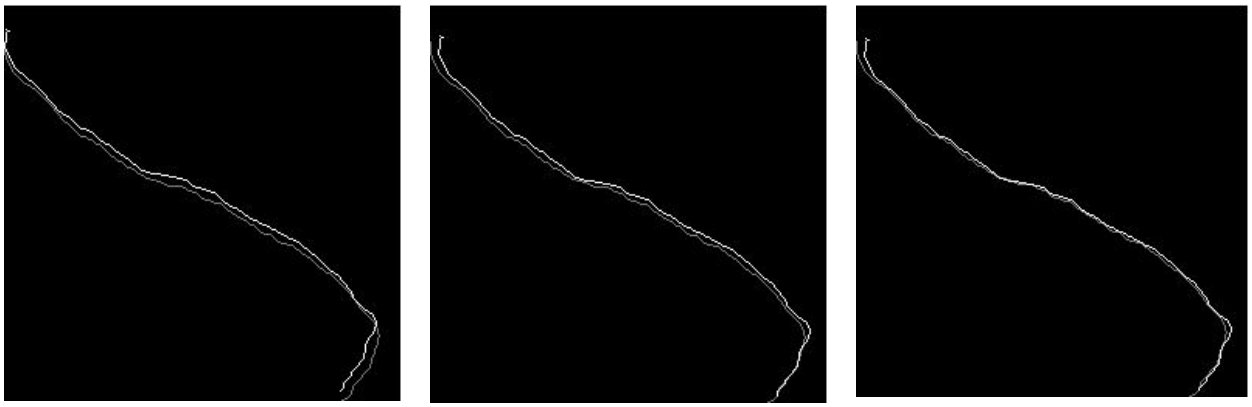
    1). Compute the closest points P_t to Y_t according to Eq. (5.9)

2). Based on P_t and Y_t, compute the matrix $\mathbf{R}_t$ and the translation vector $\vec{q}_T^{(t)}$, according to Eq. (5.2) - (5.6)

3). $Y_{t+1} = \mathbf{R}_t \cdot \vec{y}_i^{(t)} - \vec{q}_T^{(t)}$, $t = t+1$

End

End



(a) (b) (c)

Fig. 5.2 (a) backbones of neuron images acquired at time k (darker one) and $k+1$ (brighter one), (b) registration results by ICP without feature point set selection, and (c) registration results by ICP with feature point set selection.

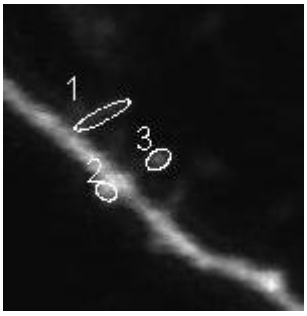
Fig. 5.2 shows the comparison of the registration results of ICP with and without feature point selection. As we can see from the results, the performance of ICP algorithm has been obviously improved for dendrite registration with feature point selection.

5.2.2 Tracking

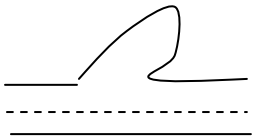
In this section, the tracking problem and the models being used will be discussed first. Then, how to apply the particle filters in tracking the deformable dendritic spines will be discussed in detail.

5.2.2.1 General models

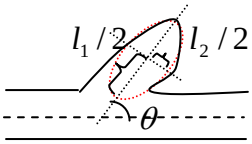
Dendritic spines can be roughly categorized into three major classes including stubby, thin, and mushroom spines [39]. Stubby spines have a constant or continually decreasing diameter. A thin spine has a bulbous head and a thin neck that connects the head of the spine to the stalk of dendrite while a mushroom spine is composed of a thin neck topped off by a large bulbous head. The intensity distribution of a dendritic spine can be roughly described by a 2D-Gaussian distribution: the pixels around the centroid of the spine have the highest intensity values; the intensity keeps decreasing as the pixel moves away from the centroid, with the pixels on the boundary having the lowest intensity. In addition, since the equidensity contours of a non-singular multivariate Gaussian distribution are ellipsoids with different sizes and orientations, we can also use a 2D-Gaussian distribution to simulate the shape of the dendritic spines.



(a)



(b)



(c)

Figure 5.3 (a) different types of dendritic spines represented by 2D Gaussian distributions: 1. thin 2. stubby 3. mushroom; (b) typical shape of a dendritic spine; (c) a dendritic spine simulated by an 2D Gaussian distribution, with the equidensity contour being illustrated red.

From Fig. 5.3 we can observe that the shape of a dendritic spine can be well described by an orientated ellipse (contour of a 2D Gaussian distribution). Although the dendritic spine cannot exactly fit into an ellipse, the most important geometric information such as the spine length, width, and orientation, can be well estimated. The 2D Gaussian model for a spine at time k can be described as:

$$\hat{I}_j(k) = I_c(k) \cdot \exp[-\frac{\mathbf{d}_j^T(k)\boldsymbol{\Sigma}^{-1}(k)\mathbf{d}_j(k)}{2}]. \qquad (5.10)$$

Here, $I_c(k)$ is the intensity of the spine centroid, $\hat{I}_j(k)$ is the intensity value of the spine pixel j, and $\boldsymbol{\Sigma}$ is the covariance matrix. The vector $\mathbf{d}_j(k) = (r_{x_j}(k) - r_{x_c}(k), r_{y_j}(k) - r_{y_c}(k))$ describes the position difference between the j-th pixel and the center of spine.

The shape and orientation of the dendritic spine are specifically defined by the covariance matrix Σ: the directions of the principal axes are given by the eigenvectors of the covariance matrix. The squared relative lengths of the principal axes are given by the corresponding eigenvalues. The eigen decomposition of matrix Σ is

$$\Sigma = U\Lambda U^T = U\Lambda^{1/2}(U\Lambda^{1/2})^T. \quad (5.11)$$

Here, $\Lambda = \begin{bmatrix} l_1^2/4 & 0 \\ 0 & l_2^2/4 \end{bmatrix}$ is the scaling matrix, and $U = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$ is the rotation matrix.

With the dendritic spines being described by 2D-Gaussians models, tracking a deforming dendritic spine amounts to estimating the parameters of a time varying 2D-Gaussian, based on registered data from one time point to the next. The tracking is performed in a state space framework. Suppose $\mathbf{x}$ and $\mathbf{z}$ are the state vector and the observed data respectively, at time k, we have:

$$\mathbf{x}_k = (\theta(k), c_x(k), c_y(k), l_1(k), l_2(k)), \quad (5.12)$$

$$\mathbf{z}_k = \{I_j(k) \mid j \in S(k)\}, \quad (5.13)$$

where θ, l_1, l_2 represents the angle, the length and the width of the estimated spine blob, $\vec{c}(k) = (c_x(k), c_y(k))$ is the location of the centroid of the spine; $I_j(k)$ is the intensity value of pixel j in a time variant local region $S(k)$. With $\mathbf{x}_k$ and $\mathbf{z}_k$ defined, the dynamic model $p(\mathbf{x}_k \mid \mathbf{x}_{k-1})$ is

$$p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) \propto \exp[-\frac{(\mathbf{x}_k - \mathbf{x}_{k-1})^T \Sigma_x^{-1} (\mathbf{x}_k - \mathbf{x}_{k-1})}{2}]. \quad (5.14)$$

We assume that all the five states change independently under the Gaussian distribution, i.e. Σ_x is a diagonal matrix.

Suppose the intensity distribution of a dendritic spine can be described by a time variant 2D-Gaussian, the observation model $p(\mathbf{z}_k \mid \mathbf{x}_k)$ for each spine at time k can be defined as

$$p(\mathbf{z}_k \mid \mathbf{x}_k) = \frac{1}{C}\exp[-\sum_{j\in S_k}\frac{(I_j(k)-\hat{I}_j(k))^2}{\sigma_z(k)}], \quad (5.15)$$

if the noise is additive white Gaussian. Here, C is a constant number for normalization; the local region S_k is composed of the previously tracked spine (at time $k-1$) and its surrounding region. Combined with Eq. (5.14), we have

$$p(\mathbf{z}_k \mid \mathbf{x}_k) = \frac{1}{C}\exp[-\sum_{j\in S_k}\frac{\left(I_j(k)-I_c(k)\cdot\exp[-\frac{\mathbf{d}_j^T(k)\boldsymbol{\Sigma}^{-1}(k)\mathbf{d}_j(k)}{2}]\right)^2}{\sigma_z(k)}]. \quad (5.16)$$

After all the vectors and models being defined, we will present how to efficiently track the dynamic and deforming spines by implementing a particle filter.

5.2.2.2 Tracking deforming spines by particle filter

Particle filtering is a technique for implementing a recursive Bayesian filter without analytically calculating the posterior density function, which is intractable for nonlinear non-Gaussian problems. The posterior density function is estimated by a set of random

samples with associated weights. As the number of samples becomes very large, the Monte Carlo (MC) characterization of samples becomes an equivalent representation to the real posterior probability density. The posterior density function can be estimated by the summation of the weighted samples, as shown in Eq. (2.40). The weight for each particle can be sequentially estimated by different approaches such as sequential importance sampling (SIS) [31], which is described hereafter.

Suppose $p(x)$ is a probability density from which drawing samples is difficult, and $q(x \mid z)$ is the importance density from which the samples can be easily generated. If the samples $\mathbf{x}_{0:k}^{i}$ are drawn from an importance density $q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$, then the weights can be defined as

$$\omega_k^i \propto \frac{p(\mathbf{x}_{0:k}^i \mid \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^i \mid \mathbf{z}_{1:k})} . \tag{5.17}$$

Suppose the *importance density* can be factorized as

$$q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) = q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1,} \mathbf{z}_{1:k}) q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1}) , \tag{5.18}$$

which means that one can obtain samples $\mathbf{x}_{0:k}^{i} \sim q(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k})$ by augmenting each of the existing samples $\mathbf{x}_{0:k-1}^{i} \sim q(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1})$ with the new state $\mathbf{x}_k^i \sim q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k})$. It has also been shown in [32] that

$$p(\mathbf{x}_{0:k} \mid \mathbf{z}_{1:k}) \propto p(\mathbf{z}_k \mid \mathbf{x}_k) p(\mathbf{x}_k \mid \mathbf{x}_{k-1}) p(\mathbf{x}_{0:k-1} \mid \mathbf{z}_{1:k-1}) . \tag{5.19}$$

From Eq. (5.17-5.19), we can obtain the following recursive expression of the weights

$$\omega_k^i \propto \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)\, p(\mathbf{x}_{0:k-1}^i \mid \mathbf{z}_{1:k-1})}{q(\mathbf{x}_k^i \mid \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})\, q(\mathbf{x}_{0:k-1}^i \mid \mathbf{z}_{1:k-1})},$$

$$= \omega_{k-1}^i \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{0:k-1}^i, \mathbf{z}_{1:k})}. \tag{5.20}$$

Furthermore, if the values of the current states are only decided by the values of the immediately previous states:

$$q(\mathbf{x}_k \mid \mathbf{x}_{0:k-1}, \mathbf{z}_{1:k}) = q(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{z}_{1:k}), \tag{5.21}$$

then Eq. (5.20) can be simplified to

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(\mathbf{z}_k \mid \mathbf{x}_k^i)\, p(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i \mid \mathbf{x}_{k-1}^i, \mathbf{z}_{1:k})}, \tag{5.22}$$

and the posterior filtered density $p(\mathbf{x}_k \mid \mathbf{z}_{1:k})$ can be approximated as

$$p(\mathbf{x}_k \mid \mathbf{z}_{1:k}) \approx \sum_{i=1}^{N_s} \omega_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i). \tag{5.23}$$

The initial values of the state $\mathbf{x}_0 = (\theta(0), x(0), y(0), l_1(0), l_2(0))$ can be either set manually, or automatically measured from the segmentation results. Here, location $(x(0), y(0))$ is set as the position of the centroid of the spine to be tracked. Orientation $\theta(0)$ is set as the angle of the line L_1 connecting the centroid and the base center of the spine. Semi-major axis l_1 and semi-minor axis l_2 are set as the parts of L_1 and L_2 inside the spine respectively. The line L_2 perpendicularly intersects with L_1 at the centroid of the spine. To deal with the degeneracy problem of the sequential

importance sampling (i.e., all but one particle will have negligible weight after a few iterations), resampling is performed to remove particles that have small weights.

The pseudo-codes for the tracking algorithms

1. Initialization: $k = 0$,

   For $i = 1,..., N$

   - Set the initial state vectors $\mathbf{x}_0^{(i)}$ with the same value
   - Let $\omega_0^{(i)} = \frac{1}{N}$

   End

2. Tracking with updated particles

   For $k = 1,2,...$ (different time)

   1) Sequential importance sampling: estimate the state values and relative weights; here we suppose the transition prior is the importance density, i.e. $q(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})$

   For $i = 1,..., N$ (different particles)

   - Sample $\hat{\mathbf{x}}_k^{(i)} \sim p(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)})$ and set $\hat{\mathbf{x}}_{0:k}^{(i)} \equiv (\hat{\mathbf{x}}_{0:k-1}^{(i)}, \hat{\mathbf{x}}_k^{(i)})$
   - Evaluate the weight for each particle $\omega_k^{(i)} = \omega_{k-1}^{(i)} p(\mathbf{z}_k \mid \mathbf{x}_k^{(i)})$
   - Normalize the importance weights $\tilde{\omega}_k^{(i)} = \omega_k^{(i)} [\sum_{j=1}^{N} \omega_k^{(i)}]^{-1}$

   End

   2) Resampling to reduce the effects of degeneracy

- Remove samples with low importance weight from the whole sample set $\hat{\mathbf{x}}_k^{(i)}$; samples $\hat{\mathbf{x}}_k^{(i)}$ with high importance weight $\tilde{\omega}_k^{(i)}$ are used to estimate the posterior density function $p(\mathbf{x}_k \mid \mathbf{z}_k)$, based on which N new particles are randomly sampled $\mathbf{x}_k^{(i)}$, representing the N updated state vectors at time k .
- For $i = 1,..., N$, let all new particles have the same weights $\omega_k^{(i)} = \tilde{\omega}_k^{(i)} = \frac{1}{N}$

3) State estimation

The estimated state vector at time k , $\hat{\mathbf{x}}_k$ can be calculated as the marginal conditional mean of $\mathbf{x}_{0:k}$, which is also the optimal MMSE estimate of the current state of the system

$$\hat{\mathbf{x}}_k = \frac{1}{N} \sum_{j=1}^{N} \mathbf{x}_k^{(i)}$$

End

5.3 Results

5.3.1 Tracking Analysis

To validate the tracking performance, time-lapse image series are processed by both the particle filter based algorithm and an existing segmentation based algorithm. A time lapse analysis algorithm proposed by Koh et al in [13] is cited for comparison purpose here. In their method, two images are co-registered by comparing the detected spines in each image. The optimal global offset is found by minimizing the absolute value sum

difference between the two images. Two spines in different images are considered the same if at least 25% percent of the volumes are overlapped after registration.

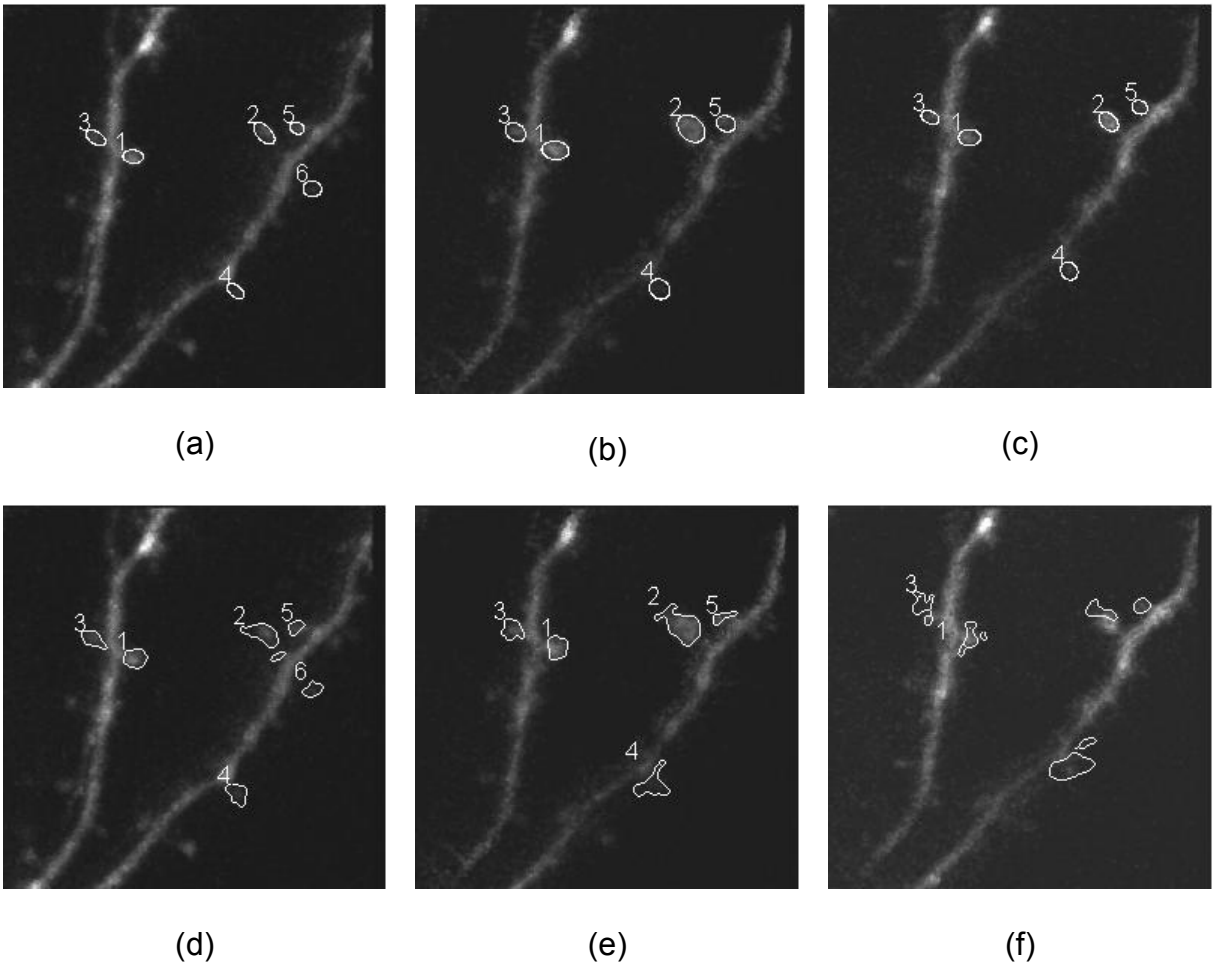


Figure 5.4: (a)-(c): tracking result of particle filter based algorithm, at time point 1, 3, 7; (d)-(f): matching based tracking results by Koh's method, at time point 1, 3, 7.

From the results in Fig. 5.4 we can see that the particle filter based algorithm can successfully track 5 of the 6 randomly selected spines. Spine 6 is considered vanished or dead. A spine is defined as a vanished spine if the maximal intensity of the pixels

around the estimated centroid (3×3 window) is below a predefined threshold. The threshold is set as 50 for 8-bit grey level images during our testing. Spine 1 and 2 are the two spines being stimulated. Obvious size changes can be observed after the stimulation.

Koh's method has difficulty in tracking the small spine (spine 5). In addition, for the segmentation based algorithms, the tracking performance is largely restricted by the efficiency of the segmentation and detection algorithms. Spine 4 cannot be tracked because of the poor segmentation result. Spine 2 is lost during the tracking because of the detection error. It is first detected as a detached spine as shown in Fig. 4 (d), (e). Then as shown in Fig. 4 (f), its major part is detected as a dendrite branch while only a small portion is detected as the spine. This causes the failure of tracking. In contrast to the segmentation based tracking algorithm, our algorithm is nevertheless much more robust to the image quality, and the performance is not restricted by the implemented segmentation and detection algorithms.

5.3.2 Measurement Analysis

To analyze how spines change in time-lapse, it is important to provide the user with a wide range of morphological statistics. Certain key geometric features of the spine, such as the spine orientation and head width, are difficult to be automatically measured by existing spine detection algorithms. However, they can be easily obtained within the framework of particle filter based tracking. The information about spine length, width,

area, and orientation can be directly obtained from the values of parameters θ, l_1, and l_2. The spine orientation is described by θ. The head width is described by the length of minor axis l_2. The spine length $L = \frac{l_1}{2} + l_0$, where l_1 is the length of the major axis; l_0 is the distance between the spine centroid and the center of the base. The spine area is defined as $S = \frac{\pi}{4} \cdot l_1 \cdot l_2$.

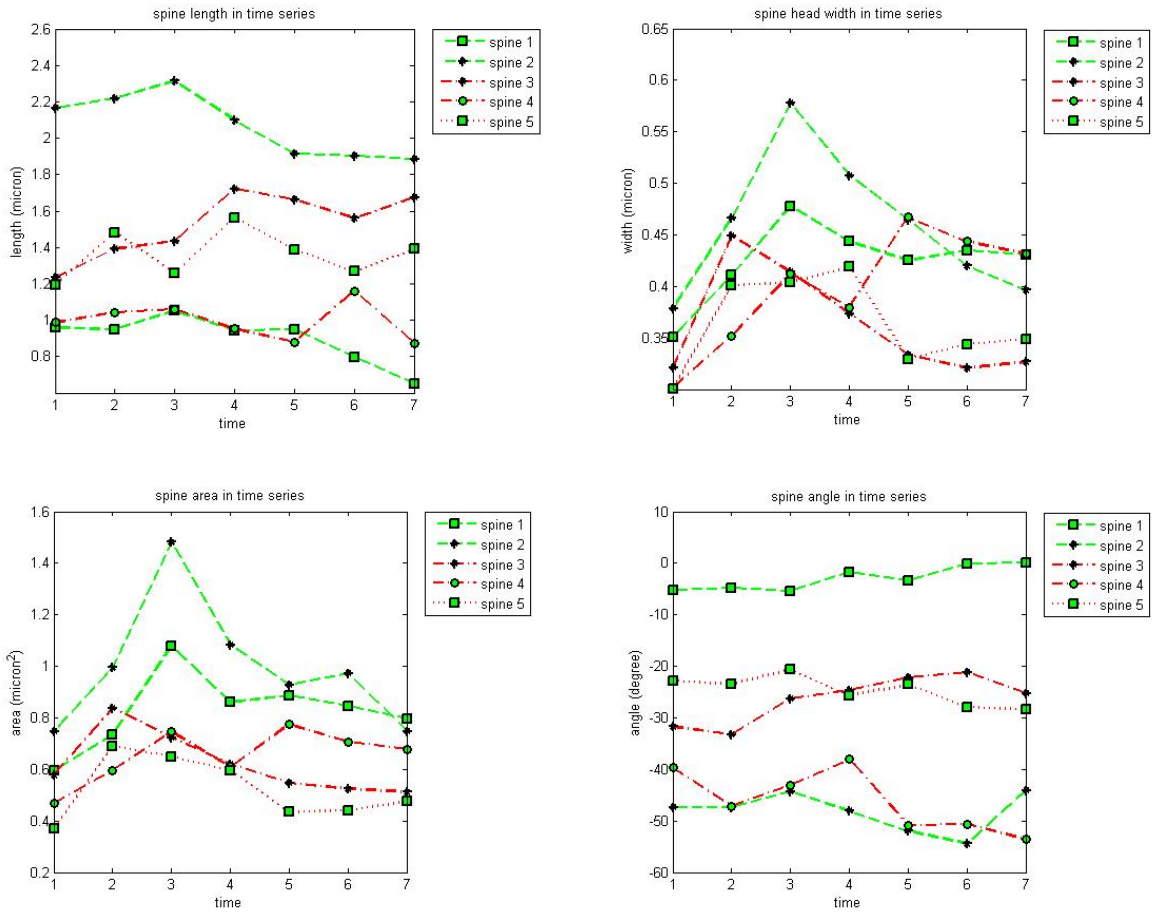


Figure 5.5: Time series plot for the geometric features of spine 1, 2, 3, and 5 in Fig. 4: (a) spine length; (b) spine head width; (c) spine area; and (d) spine orientation.

For all the image sequences processed here, the images are acquired every five minutes, with 1 or 2 spines being stimulated at time point 3. For the images demonstrated in Fig. 5.4, two of the spines (1, 2) are stimulated. We can clearly observe from the curves in Fig. 5.5 that the size of the stimulated spines turns bigger immediately after the stimulation and shrinks a little afterwards. However, the change of spine orientations is rather random.

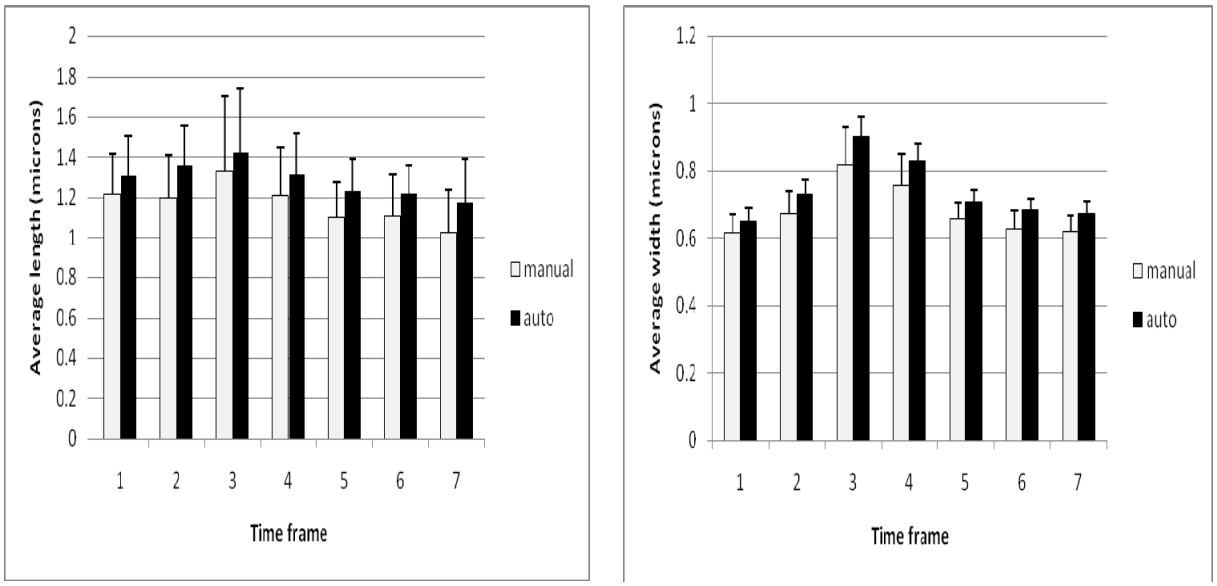


Figure 5.6: Comparison between automated and manual measurement of spines in time series: (a) average spine length and (b) average spine head width.

To validate the spine morphologic measurement automatically obtained during the tracking process, average spine length and width of 30 spines in time series are compared. These 30 spines are randomly selected from 6 sets of time-lapse images

acquired at 7 different time points. In all of the 6 sets image sequence, 1 or 2 spines are stimulated at time point 3. As shown in Fig. 5.6, manual and automatic results agree to within one standard deviation. However, obvious bias can be observed for the automatically obtained geometric features. The automatically obtained results are constantly larger than the manual results. Pearson correlation is therefore used to test whether the automatic and manual results are correlated at each time point. The correlation values r at 7 time points range from 0:527 to 0:971, which suggest significant linear correlations between the automatic and the manual results in time series.

Table 5.1: Pearson correlation for spine lengths and width in time series

	Time 1	Time 2	Time 3	Time 4	Time 5	Time 6	Time 7
Length	0.7734	0.875	0.907	0.851	0.527	0.792	0.637
Width	0.827	0.785	0.971	0.703	0.782	0.675	0.573

5.4 Discussions

In this chapter, we propose a particle filter based tracking algorithm. Compared with the existing algorithms, the proposed algorithm can better track the highly dynamic dendritic spines and is more robust to the noise and the segmentation and detection errors. Since the spine parameters are naturally included in the dynamic model, shape information of the spines can be obtained at the same time during tracking. In

comparison with the existing match based algorithms, more information about the spine morphology such as the width and orientation can be easily acquired.

In the proposed algorithm, the spines are independently tracked such that the tracking results for different spines will not affect each other. In some algorithms, such as Mosaliganti' method [16], all the dendritic spines are tracked as a whole set. Thus, the tracking error of one spine may risk degrading the tracking results of other spines. Another advantage of separately tracking spines is that it is much more time efficient by only concentrating on certain interesting spines, rather than trying to detect over a hundred of spines in an image. The proposed algorithm is also user friendly, and, if necessary, human interference can be conveniently incorporated into the tracking framework. For instance, the particle parameters can be initialized based on the manual selection of the local region, if the image quality is too low to be correctly segmented by the existing algorithms.

References

[1]. A. Matus, Actin-based plasticity in dendritic spines, *Science*, Oct. 27, 290: 754-758 (2000)

[2]. S. Halpain, Actin and the agile spine: how and why do dendritic spines dance, *Trends Neurosci.* 23: 141–146, (2000)

[3]. M. Fischer, S. Kaech, D. Knutti and A. Matus, Rapid actin-based plasticity in dendritic spines, *Neuron*, 20(5): 847–54 (1998)

[4]. S. Cho, J. Ko, J. Lee, H. W. Lee, K. Kim, H. S. Chung, H. Kim, and E. Kim, ARF6 and EFA6A regulate the development and maintenance of dendritic spines, *J. Neurosci.*, May 3, 26(18):4811–19 (2006)

[5]. A. Dunaevsky, A. Tashiro, A. Majewska, C. Mason, and R. Yuste, Developmental regulation of spine motility in the mammalian central nervous system, *PNAS*, Nov. 9, 96(23): 13438–43 (1999)

[6]. L. Zhou, S. J. Martinez, M. Haber, E. V. Jones, D. Bouvier, G. Doucet, A. T. Corera, E. A. Fon, A. H. Zisch, and K. K. Murai, EphA4 signaling regulates phospholipase C□1 activation, cofilin membrane association, and dendritic spine morphology, *J. Neurosci.*, May 9, 27(19):5127-38 (2007)

[7]. Michael E. Dailey and Stephen J Smith, The Dynamics of Dendritic Structure in Developing Hippocampal Slices, *J. Neurosci.*, May 1, 16(9):2983–94 (1996)

[8]. J. Hao, P. R. Rapp, A. E. Leffler, S. R. Leffler, W. G. M. Janssen, W. Lou, H. McKay, J. A. Roberts, S. L. Wearne, P. R. Hof and J. H. Morrison1, Estrogen alters spine number and morphology in prefrontal cortex of aged female rhesus monkeys, *J. Neurosci.*, March 1, 26(9):2571–78 (2006)

[9]. B. Lendvai, E. A. Stern, B. Chen and K. Svoboda, Experience-dependent plasticity of dendritic spines in the developing rat barrel cortex in vivo, *Nature,* Apr. 20, 404: 876-81 (2000)

[10]. A. Dunaevsky, A. Tashiro, A. Majewska, C. Mason, R. Yuste, Developmental regulation of spine motility in the mammalian central nervous system, *PNAS,* Nov. 9, 96(23): 13438-43 (1999)

[11]. J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, S. T.C. Wong, A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy, *J. of Neurosci. Methods*, Sept. 15, 165(1):122-34 (2007)

[12]. D. A. Rusakov, M. G. Stewart, Quantification of dendritic spine populations using image-analysis and a tilting dissector. *J Neurosci Methods* 60(1-2):11–21 (1995)

[13]. I. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, K. Svoboda, An image analysis algorithm for dendritic spines. *Neural Comput.,* 14:1283–310 (2002)

[14]. C. M. Weaver, P. R. Hof, S. L. Wearne, W. B. Lindquist, Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Comput.,* 16:1353–383 (2004)

[15]. Y. Zhang, X. Zhou, R. M. Witt, B. L. Sabatini, D. Adjeroh, S. T. Wong, Dendritic spine detection using curvilinear structure detector and LDA classifier, *NeuroImage*, June , 36(2): 346-360 (2007)

[16]. K.R. Mosaliganti, F. Janoos, X. Xu, R. Machiraju, K. Huang, S.T. Wong, Temporal matching of dendritic spines in confocal microscopy images of neuronal tissue sections, In Proceedings of the MICCAI Workshop on Medical Image Analysis with Applications in Biology, Copenhagen, Denmark, Oct. 5, pp 106-113 (2006)

[17] A.R. Cohen, B. Roysam, and J.N. Turner, Automated tracing and volume measurements of neurons from 3-D confocal fluorescence microscopy data, *J. Microscopy*, 173(2): 103-114 (1994)

[18]. W. Bai, X. Zhou, L. Ji, J. Cheng, S. T. Wong, Automatic dendritic spine analysis in two-photon laser scanning microscopy images, *Cytometry* A, 71A:818-826 (2007)

[19] Y. Chen and G. Medioni, Object Modeling by Registration of Multiple Range Images, *Proc. IEEE Conf. on Robotics and Automation* (1991)

[20] S. Rusinkiewicz, M. Levoy, Efficient variants of the ICP algorithm, *Proc. 3rd Intl. conf. on 3-D digital imaging and modeling (3DIM)* (2001)

[21] S. Weik, Registration of 3-D Partial Surface Models Using Luminance and Depth Information, *Proc. 3DIM* (1997)

[22] G. Godin, M. Rioux, and R. Baribeau, Three-dimensional registration using range and intensity information, *Proc. SPIE: Videometrics III*, Vol. 2350 (1994)

[23] D. Simon, Fast and accurate shape-based registration, Ph. D. Dissertation, Carnegie Mellon University (1996)

[24] R. Benjemaa, and F. Schmitt, Fast global registration of 3D sampled surfaces using a multi-Z-buffer technique, *Proc. 3DIM*, (1997)

[25] K. Pulli, Surface reconstruction and display from range and color data, Ph. D. Dissertation, University of Washington (1997)

[26] T. Masuda, K. Sakaue, and N. Yokoya, Registration and integration of multiple range images for 3-D model construction, *Proc. CVPR* (1996)

[27] C. Dorai, J. Weng, and A. Jain, Registration and integration of multiple object views for 3D model constrution*, IEEE Trans. PAMI*, 20(1): 83-89 (1998)

[28] G. Blais, and M. Levine, Registering multiview range data to create 3D computer objects, *IEEE Trans. PAMI*, 17(8): 820-24 (1995)

[29] G. Turk, and M. Levoy, Zippered Polygon Meshes from Range Images, *Proc. SIGGRAPH*, pp. 311-318 (1994)

[30] P. J. Besl and N. D. McKay, A method for registration of 3-D shapes, *IEEE Trans. PAMI*, 14(2): 239-56 (1992)

[31] A. Doucet, On sequential Monte Carlo sampling for Bayesian filtering, Dept. Eng., Univ. Cambride, Uk, Tck. Rep. (1998)

[32] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Trans. Sig. Proc.* 50(2): 174-88 (2002)

[33] V. Alvarez, D. A. Ridenour and B. L. Sabatini, Retraction of synapses and dendritic spines induced by off-target effects of RNA Interference, *J. of Neurosci.*, 26(30):7820-25 (2006)

[34] D. Comaniciu, V. Ramesh, and P. Meer, Kernel-based object tracking, *IEEE Trans. on PAMI*, 25(5):564-77 (2003)

[35] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, Pfinder: Real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780-85 (1997)

[36] S. Yamamoto, Y. Mae, Y. Shirai, J. Miura, Realtime multiple object tracking based on optical flows *IEEE Int. Conf. on Robotics and Automation*, Vol.3 2328-33 (1995)

[37] T. Drummond, R. Cipolla, Real-time visual tracking of complex structures, *IEEE Trans. on PAMI*, 24(7):932-946 (2002)

[38] M. Isard, A. Blake, CONDENSATION–Conditional Density Propagation for Visual Tracking, *Int. J. of Computer Vision,* 29(1):5-28 (1998)

[39] K. M. Harris, F. E. Jensen, and B. Tsao, Three-dimensional structure of dendritic spines and synapses in rat hippocampus (CA1) at postnatal day 15 and adult ages: Implications for the maturation of synaptic physiology and long-term potentiation, *J. Neurosci*. 12: 2685-705 (1992)

[40] S. Kaech, H. Brinkhaus, and A. Matus, Volatile anesthetics block actin-based motility in dendritic spines, *PNAS*, 96(18): 10433–10437 (1999)

[41] A. Majewska and M. Sur, Motility of dendritic spines in visual cortex in vivo: Changes during the critical period and effects of visual deprivation, *PNAS*, 100(26): 16024-16029 (2003)

Chapter 6 Conclusions and future work

To release the biologists from heavy manual burden for quantitative analysis of dendritic spines, we propose a pipeline for automatic dendritic spine detection in Chapter 3, which is especially suitable to batch process a large image dataset. First, an adaptive thresholding method is presented. A signal to noise ratio (SNR) based detached spine detection algorithm is presented afterwards. An effective attached spine detection algorithm based on local morphological information is also discussed in detail. Compared with the existing algorithms, the detection performance is improved in the sense of both reducing the false negatives (missed spines) and false positives (wrongly detected results). However, besides the problem of automatic estimation of the window size as we mentioned in Chapter 3, other problem still exists. For example, although we can successfully solve the problem to combine an attached spine component with its detached components, in cases that there are only two or more detached spine components in the image and not any relative attached spine component, we cannot find an efficient way to combine those detached spine parts without wrongly combining the parts from different spines. This problem also needs to be addressed in the future work.

Furthermore, as we mentioned in Chapter 4, the efficiency of detection will degrade for images with lower resolutions in which the spines occupy relatively few pixels. Also, the algorithm somewhat distorts the shape of the detected spines: spine pixels with low intensity values, such as the pixels in the spine neck regions, are prone to be

segmented as background, which cause the spines segmented into several broken components and further degrade spine detection performance.

To deal with these issues, in Chapter 4 we propose a novel maximum a posteriori - orientated Markov random field (MAP-OMRF) framework for dendritic spine segmentation. A knowledge guided iterative conditional modes (KICM) method is also proposed to find the optimal solutions. By combining the natural spatial constraints in the prior model of images, the proposed method can obviously improve the detection performance for image acquired at different resolution. With the weak parts of the spines, especially the neck regions being enhanced, the shape of detected spines resembles more to their real shapes.

Despite of the advantages of the OMRF based neuron image segmentation method mentioned above, more work can be done which will potentially improve the detection results. For example, as described in Section 4.3.1 the parameter of the prior model $\boldsymbol{\theta}_1$ is set constant for all testing images for the purpose of simplifying the computation complexity. However, intuitionally the optimal values of $\boldsymbol{\theta}_1$ should be image dependent. Thus, the value of $\boldsymbol{\theta}_1$ should be estimated online instead of being set constant.

Suppose $\mathbf{y}$ is the observed data, $\hat{\boldsymbol{\theta}}_2$ is the maximal likelihood estimation as described in Section 4.1, the optimal segment results $\hat{\mathbf{x}}$ and parameter $\hat{\boldsymbol{\theta}}_1$ can be found by solving the following MAP estimation problem:

$$(\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}}_1) = \arg\max_{\mathbf{x},\boldsymbol{\theta}_1} P(\mathbf{x}, \boldsymbol{\theta}_1 \mid \hat{\boldsymbol{\theta}}_2, \mathbf{y}). \quad (6.1)$$

Assume that $\boldsymbol{\theta}_1$ is uniformly distributed, using the laws of conditional probabilities, Equation (6.1) can be changed to

$$(\hat{\mathbf{x}}, \hat{\boldsymbol{\theta}}_1) = \arg\max_{\mathbf{x},\boldsymbol{\theta}_1} P(\mathbf{y} \mid \mathbf{x}, \hat{\boldsymbol{\theta}}_2) P(\mathbf{x} \mid \boldsymbol{\theta}_1) P(\boldsymbol{\theta}_1),$$

$$= \arg\max_{\mathbf{x},\boldsymbol{\theta}_1} P(\mathbf{y} \mid \mathbf{x}, \hat{\boldsymbol{\theta}}_2) P(\mathbf{x} \mid \boldsymbol{\theta}_1). \quad (6.2)$$

It can be further divided into two sub-problems which can be estimated iteratively.

$$\hat{\boldsymbol{\theta}}_1^{(k)} = \arg\max_{\boldsymbol{\theta}_1} P(\mathbf{y} \mid \hat{\mathbf{x}}^{(k-1)}, \hat{\boldsymbol{\theta}}_2) P(\hat{\mathbf{x}}^{(k-1)} \mid \boldsymbol{\theta}_1), \quad (6.3)$$

$$\hat{\mathbf{x}}^{(k+1)} = \arg\max_{\mathbf{x}} P(\mathbf{y} \mid \mathbf{x}, \hat{\boldsymbol{\theta}}_2) P(\mathbf{x} \mid \hat{\boldsymbol{\theta}}_1^{(k)}). \quad (6.4)$$

Many algorithms have been proposed to solve this problem, such as maximum likelihood, coding, pseudo-likelihood, expectation-maximization, mean field approximations, and least squares fit etc [1]. However, how to find a computational efficient algorithm, which combines the prior knowledge of the neuron images, remains a further question to be investigated.

In addition, in Chapter 4 a knowledge-based iterated conditional modes algorithm (KICM) is proposed to find the optimal segmentation result. The basic idea of KICM algorithm is to develop a new strategy of searching the ideal solutions (though statistically sub-optimal) by incorporating some constraints with explicit biological meaning. The experimental results are satisfying. However, ideally those constraints should be combined in the energy function so that a statistically optimal solution can by found by common optimization algorithms such as ICM. For example, in Section 4.3.2

the distance constraint is mentioned: spine pixels can only be those pixels whose distance to the center of the dendrite (the backbone) is greater than half of the local dendrite width. Suppose the label of the dendrite and spines are “1” and “2” respectively, after considering the distance constraints (Eq. 4.19, 4.20), the energy function expressed in Eq. (4.4) can be rewritten as:

$$U(\mathbf{x}) = \beta \cdot \sum_{i \in S} \sum_{j \in N_i} D(x_i, x_j)[1 + \alpha \cdot (1 - |\vec{u}_{ij}{}^T \vec{v}_i|)] + \beta_d \cdot [\delta_{x_i,1} H(d_i - \frac{t_i}{2}) + \delta_{x_i,2} H(\frac{t_i}{2} - d_i)], \quad (6.5)$$

where $\delta_{x_i,1}$ and $\delta_{x_i,2}$ are Kronecker delta function, $H()$ is the Heaviside step function, β_d is the weight of the distance constraint, d_i is the distance between pixel i and backbone, t_i is the estimation of the width of local dendrite. How to obtain the best value of β_d and how to combine other constraints into the energy function would be an interesting topic for us future work.

To automatically analyze the morphological changes of dendritic spines in time lapse images, we propose a particle filter based tracking algorithm in Chapter 5. The algorithm is shown to be able to efficiently track and measure the dendritic spines simultaneously. However, there are still some unsolved problems. For example, the resampling is applied to solve the degeneracy problem. However, it will also induce the problem of sample impoverishment because the particles with high weights are statistically selected many times, which leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. In some bad cases, e.g. if the process noise is very small, all particles will collapse to a single point within a few iterations. To deal with this issue, Carpenter et al. [4] propose a method which can

monitor the efficiency of the particle filter by providing a simple quantitative assessment of sample impoverishment. A computation efficient particle filters which can prevent the collapse of the particle system is also presented. Zhu el al. [5] propose an improved particle filter by integrating support vector regression (SVR) into sequential Monte Carlo framework. By using an SVR based re-weighting scheme, the posterior density is re-approximated and sample impoverishment can be effectively avoided.

Another unsolved problem for the proposed particle filter based algorithm is: although it is easy to detect stable and vanished dendritic spines, it is much more difficult to detect a newly born spine. How to apply the particle filter to detect a newly born spine is also a potential topic for us.

For Monte Carlo methods (particle filter is sequential Monte Carlo), the basic idea is to estimate the probability density functions (PDF) according to a set of randomly chosen samples with different weights. Obviously, the more samples are provided, the better the estimation is. However, the number of the samples cannot be arbitrarily large because of the computation cost. One possible solution for this problem is to compose the PDF in a way so that the samples are generated in a space with reduced dimensions. For our proposed method, the samples space is a 5-D: orientation, location of the spine centroid, the length and width of the spines. An interesting future topic is that if it is feasible to track a spine in space(s) with smaller dimensions? For example, can we first track the location of the spines in a 2-D space and then adjust the morphology of the spines in a 3-D space?

Lastly, in this thesis the dendritic spine detection and tracking algorithms are based on the maximal intensity projection (MIP) image of the original 3D laser scanning microscopy (LSM) image stacks. However, to quantify 3D spine shapes with sufficient precision can greatly help us understand how the spine morphology controls learning and memory. Compared with the 2D projection based algorithms, 3D based methods can more accurately describe the dendritic spines and potentially more robust to track the morphological changes of the spines. Recently, several 3D dendritic spine reconstruction and detection algorithms have been proposed [2][3]. However, one major disadvantage of algorithms operating fully in 3D is the complex computation. How to model in 3D and design a computation efficient tracking algorithm is an interesting future topic.

References

[1] S.Z. Li, Markov Random Field Modeling in Image Analysis, 2nd ed. Tokyo, Japan: *Springer-Verlag* (2001)

[2] F. Janoos, K. Mosaliganti, X. Xu, R. Machiraju, K. Huang, S.T. Wong, Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging, *Med. Imag. Analy.* 13:167–179 (2009) published in advance online

[3] A. Rodriguez, D. Ehlenberger, D. L. Dickstein, P. R. Hof, S. L. Wearne, Automated Three-Dimensional Detection and Shape Classification of Dendritic Spines from Fluorescence Microscopy Images, *PLoS ONE*, Apr. 3(4) (2008)

[4] J. Carpenter, P. Clifford, P. Fearnhead, Improved particle filter for nonlinear problems, IEE Proc. Radar, sonar and navigation, 146(1): 2-7 (1999)

[5] G. Zhu, D. Liang, Y. Liu, Q. Huang, W. Gao, Improving particle filter with support vector regression for efficient visual tracking, IEEE inter. conf. on imag. proc. (2005)