

Dendritic Spine Segmentation from 3D Confocal Microscopic Images

by

Hussein Sharafeddin

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the degree of

DOCTOR OF PHILOSOPHY

Major Subject: Computer and Systems Engineering

Approved by the
Examining Committee:

Badrinath Roysam, Thesis Adviser

Michael Wozny, Member

W. Randolph Franklin, Member

Deanna Thompson, Member

Rensselaer Polytechnic Institute
Troy, New York

September 2009

(For Graduation December 2009)

© Copyright 2009
by
Hussein Sharafeddin
All Rights Reserved

CONTENTS

Dendritic Spine Segmentation from 3D Confocal Microscopic Images	i
CONTENTS	iii
LIST OF TABLES.....	vi
LIST OF FIGURES	viii
DEDICATION.....	xiv
ACKNOWLEDGMENTS	xv
ABSTRACT	xvii
1. Introduction.....	1
1.1 Spine Basics	1
1.2 Dendritic Spines: Significance and 3D Imaging.....	3
2. Literature Review	5
2.1 Spine Types and Dimensions.....	5
2.2 Spine Detection Challenges	10
2.3 Previous Spine Detection Techniques.....	11
2.4 Summary of Contributions.....	15
3. Spine Detection Foundations.....	16
3.1 The Dendrite as a Primary Structure.....	16
3.2 Goals for Ideal Neuronal Tracing	17
3.3 The Dendrite Model.....	18
3.4 Simple Spine Detection Approaches.....	18
3.4.1 First Approach: Volume Difference and Residue Segmentation.....	19
3.4.2 Second Approach: Level Set Segmentation.....	23
3.4.3 Third Approach: Revolving Cylinders.....	26
3.5 Chapter Summary.....	33
4. Neuronal Tracing and Enhancement	35
4.1 The Dendrite Shape Model	35

4.2	Intensity Estimation	38
4.3	Model Validation	39
4.4	Tracing Fidelity Enhancement	40
4.4.1	Tracing Errors and Their Effect on Spine Segmentation	41
4.4.2	Fixing Tracing Errors	45
4.5	Chapter Summary.....	50
5.	Spine Ring Detection	51
5.1	Phase II: Spine Ring Detector	53
5.1.1	Cylindrical Ring Geometry	53
5.1.2	Super-Ellipsoidal Ring Geometry	55
5.1.3	Orientation and Detection	56
5.2	Spine Fragments or Candidates?	59
5.3	Intersecting Spine Candidates	60
5.3.1	Convex Hull Computation	60
5.3.2	Testing Spine Candidate Intersection.....	63
5.4	Results and Discussion.....	64
5.4.1	Detector Performance.....	64
5.4.2	Volume Merging	64
6.	Wave-front Propagation in Grayscale Images	67
6.1	Theoretical Background: On Level Sets	68
6.2	Theoretical Background: On Fast Marching Algorithms.....	69
6.2.1	Fast Marching 3-D Formulation	70
6.2.2	Fast Marching Implementation	70
6.3	Spine-Dendrite Geodesic Tractography	75
6.3.1	Wavefront Propagation	75
6.3.2	Speed Function.....	76
6.3.3	Starting and Ending Points.....	76

6.3.4	Path Extraction.....	77
7.	Experiments and Results.....	79
7.1	Images, Preprocessing and Tracing.....	79
7.2	Ring Detection	80
7.3	Wave Propagation and Medial Axis Extraction.....	80
7.4	Final Results.....	81
7.4.1	Cylindrical Ring.....	81
7.4.2	Super-ellipsoidal Ring.....	82
7.4.3	Path and Feature Extraction	82
7.4.4	Discussion	89
7.4.5	Validation Tools.....	94
8.	Summary.....	98
8.1	Main Contributions	99
8.2	Future Work	99
	Appendix A.....	100
	Maximum Likelihood Estimate of the center of the Laplacian distribution.....	100
	Appendix B.....	101
	Convex Hull Intersection.....	101
	Appendix C.....	102
	Implementation Issues	102
	Bibliography	104

LIST OF TABLES

Table 1 Data (rows 1-5) extracted from (Harris, Jensen et al. 1992). We added the last 3 rows. To distinguish between mushroom spines and the rest, they used the head volume to neck diameter ratios where it is much higher for the mushroom type. To distinguish between thin and stubby spines, the ratio of total length to neck diameter was calculated. Thin spines have the larger numbers. We added the length-to-dendrite-diameter ratio (last row) as a justification of our detector dimensions assumptions.	9
Table 2 Performance comparison on a typical image (Day4) between the volume difference and rotating cylinders approaches. Although the latter performs significantly better, the error rates suggest further enhancements are needed in both methods.	32
Table 3 Summary of common types of tracing errors, their effects on spine detection, and potential remedies.	42
Table 4 Comparison between NeuronStudio (NS) spine detection and our spine ring detector with convex hull merging (RPI). The images used in this experiment are provided by MBF-Bioscience. Due to the spine density in some images, only some dendrites were selected for comparison. The miss rate (false negatives) with the spine ring detector and convex hull merging (RPI FN) was better than that from NS. The same is not true for the false positives (RPI FP vs. NS FP). Improvements upon these results are discussed in chapter 7.	65
Table 5 Front Propagation Algorithm using the Fast Marching Algorithm	73
Table 6 Algorithm for solving the PDE and updating the propagation arrival times.....	74
Table 7 Anisotropy ratios in our experimental images.....	79
Table 8 Single-observer comparison between NeuronStudio spine segmentation (NS) and our cylindrical spine ring detector with paths extracted by Fast Marching (RPI). The false negative counts are still better, however, the false negative counts are much more improved than those done by convex hull intersection (Table 4). Further Improvements to this ring is achieved with the super-ellipsoid ring (Table 10).	83

Table 9 Multiple-observer validation: Up to 5 observers from MBF-Bioscience validated spine detection done by our method (spine ring detector and spine path merging by fast marching). The method improves on human observer detection as noted in the difference between the first two columns. There are still spine clusters being merged together and require further separation. “Manual” is the observers’ average initial marking; “Corrected” includes “Manual” and those detected by the method only; “TP” is automatic true positives; “FP” is false positives and over-counting; “FN” is false negatives; “Improv” and “Imp%” are detections improving on manual counting; “Clusters” and “Clust %” are correctly-detected spines but are clustered together.	85
Table 10 Results with the super-ellipsoidal ring validated with the new SeedEditor tool. There is significant decrease in clustered spines and false negatives as compared with the cylindrical ring (Table 8). However, the false positive average error is slightly improved. Improvement on human detection error is not reported for this experiment.	88
Table 11 Combined error rates anisotropy ratio (Table 7). The first column is the sum of the false positive and false negative errors in Table 10.....	90
Table 12 Pseudo code for testing whether a point is inside a convex hull.	101

LIST OF FIGURES

- Figure 1 Neurite Sub-Structures(adopted from (Fiala and Harris 1999)): The four commonly-referenced dendritic spine classes are Thin, Mushroom, Stubby and Branched. They are classified by head size and neck length. Crook Thorn appears less often in the literature. Filopodia are considered quasi-spines appearing during initial stages of spine formation. Axonal varicosities are deformations of the axon shaft and may confuse a spine detection algorithm. 3
- Figure 2 Spiny dendrite segment magnified to show spine head and neck. The dendrite shaft in this work denotes the dendrite proper..... 5
- Figure 3 Example of a bifurcated (branched) spine. (Cropped from image spine2.tif, courtesy MBFBioscience and Dr. T. Ballou, Northwestern U)..... 6
- Figure 4 Spine categories and defining features: Adopted from (Harris, Jensen et al. 1992). L =side length; n=neck; h=head; d=diameter. A spine is *thin* if the neck diameter is much smaller than the side length and slightly smaller or comparable to the head diameter. *Mushroom* spines have their neck diameter much smaller than that of the head while the side length is less relevant. If the length is almost similar to the neck diameter, the spine is labeled as *stubby*. A spine is *branched* if its head splits into two (or more); other measurements are then irrelevant. 7
- Figure 5. A gallery of dendrite and spine images rendered in 3D depicting subject variety. A-E: zoomed out dendrite views in different images where branching can influence spine identification. F-J: spines on dendrite segments showing variability. K-O: dendrite segment; zooming in: frontal view of a spine cluster with stubby and thin spines; dorsal view; another cluster of thin and mushroom spines; dorsal view. P-T: examples of branched(bifurcated) spines; notice 2 branched spines in P; a faintly identifiable branch in Q-R; clearer in S; T could be missed for other types. U-Y: examples of thin, stubby, and mushroom spines..... 14
- Figure 6. Day4 image –Courtesy Prof. Joshua Trachtenberg (UCLA). Maximum intensity projection. 20
- Figure 7 Day4 – reconstructed dendrite shafts using super ellipsoid fitting superimposed on the original grayscale projection (Figure 6). Residual fluorescence can be easily

seen; some being valid spines. Two spines have been erroneously traced and hence missed after subtraction.	21
Figure 8 Day4 – residual segmentation. Dendrite shafts (blue) subtracted from original followed by morphological operations to detect spines (green). Noise near the dendrite creates false positives. Thresholding involved in binarization causes actual spines to be missed.	23
Figure 9 Level set segmentation: Spine tips detected as sharp edges. However as seen in the next figure, this technique requires further validation.	25
Figure 10 Level set segmentation: Dendrite surface irregularities creating false corners.	26
Figure 11 Image illustrates how a cylindrical model (red) is revolved around the dendrite model (blue) in search of potential spines in the neighborhood. Two rotation axes are needed: one normal to the dendrite axis to create an “elevation”, and another one along the dendrite axis to create spin.	27
Figure 12 2D-projection of Day4 Image rendering, with detected spines superimposed in blue. See Figure 6 for original image projection. 47 spines were found by the rotating cylinders approach, 27 true positives, and 20 false positives (under triangle areas); there were 9 misses (some illustrated with arrows; others invisible in this projection).	32
Figure 13 2D projection of original image (left) and reconstructed image (right) with spines detected by rotating cylinders. Two cylinder types were used here: red cylinders rotate close to the dendrite surface while green ones are at an offset determined by dendritic width. Over- detection is easily seen with multiple cylinders detecting the same spine. The approach results in a large number of false positives and with several parameters requiring user setting, it is hard to generalize. Manual labeling accounted for 214 spines while automatic detection gave 1334. The over-count is too large and the image is too dense to discriminate them from correct counts.	33
Figure 14 Phantom image illustrating super-ellipsoid fitting and tracing. Courtesy Dr. Alex Tyrrell.	36

Figure 15 Model-fitting on a real image segment rendered in light green. A sequence of super-ellipsoid surface meshes is superimposed in purple. The super-ellipsoids are oriented to best fit the local direction. The major axes along the dendrite shaft centerline are truncated for easier visualization. Some over-fitting is noticeable near the dendrite surface (arrows). 37

Figure 16 Segment from spine1.tif (data courtesy MBF-Bioscience and Dr. Fulvia Gheorghita, DBCM, UNIL, Lausanne). A sample region $R(C) = \{x \mid F(x; \beta) < 1\}$ is framed in red to demonstrate hypothesis test described above. The closer the ensemble of pixel grayscale values is to the foreground than to the background, the more optimal is the fit. 40

Figure 17 Example of a Type A tracing error. Above: left: 3D rendering of spine1.tif, right: surface mesh of tracing models superimposed on rendered image. Below: Zooming in on selected region shows the purple trace diverted onto an attached spine and a new trace (yellow) started independently. Left as is, the trace will cause the spine to be missed. 43

Figure 18 Types A and B Tracing errors at different segments with the same image as Figure 17. The purple trace avoids the first spine (yellow) but is lead onto another spine where it erroneously terminates. The orange and pink traces start independently and terminate when intersecting with the purple trace. 44

Figure 19 Z-Projection of image 6297,312MLE Courtesy Prof. S. Potter (Georgia Tech). Traces are superimposed in color. The elongated white boundary identifies Type A error caused by intensity discontinuity making the dendrite chopped into 4 pieces. Some spine-induced Type A errors are highlighted by the ellipsoidal boundaries. Some Type B errors are marked by a star. 45

Figure 20 Type A tracing errors. Rectangular highlights show almost straight disconnections due to intensity variations. The trace endings are the same as their extremities; trace neighborhood is determined by those endpoints. Circular highlights show redirected trace extremities onto a spine. These trace endpoints are not suitable for establishing trace ending and neighborhood. We need to un-trace backwards until proper alignment with the opposite fragment is achieved leading to correct reattachment. 49

Figure 21 Spine Detection flowchart. Dendrite traces are given as input to the process which starts with creating surrounding ring detectors (A). Thresholding is followed with connected component labeling and a validity test to produce spine candidates (B). For each spine candidate, a wavefront is propagated using the fast marching technique and the spine axis is extracted (C). Neighboring axes are identified for merging spine fragments and finally validating the spine. 52

Figure 22 Cylindrical ring. Left: ring surrounding dendritic region. Top center: circular hollowness. Bottom center: inner and outer circumferences. 53

Figure 23 Dendrite segment rendering with ring detector dimensions. These dimensions are determined by the dendrite maximum lateral diameter as estimated in the tracing model. 55

Figure 24 Super-ellipsoidal ring. Left: ring surrounding rendered region. Middle: (almost) frontal view showing ellipsoidal rather than cylindrical hollowness. Right: Sagittal view showing rectangular cross section. 56

Figure 25 3-D rendering of dendrite segment with spine ring detector outline (purple) surrounding the super-ellipsoid tracing element (yellow). 57

Figure 26 Spine ring in motion. The top part shows a sequence of ring detectors in grey moving up the dendritic shaft (rendered in green). The detected voxels inside each ring are shown as projected onto its central axis. The bottom part highlights the 4 candidate spines that were detected inside the rings and separated by connected components (arrows). 58

Figure 27 Three spine candidates highlighted in yellow, pink and blue over dendrite rendering (green); viewed from different angles (arrows). 59

Figure 28 Demonstration of plane equation in Hessian normal form: $a \cdot x = b$; where a and x are respectively the normal vector and any point on the plane. The distance to the origin is $|b|$. The plane divides the space into two half-spaces. The halfspace in the same side as the normal vector satisfies the equation $a \cdot x - b > 0$ 61

Figure 29 Demonstration of polyhedron with facet normals pointing inside. A point x inside the polyhedron satisfies $A^t x - \beta > 0$ where the columns of $A_{3 \times n}$ are the facet normals a_i and the facet shifts from the origin are the elements of $\beta_{n \times 1}$ 62

Figure 30 Color coded 3D view of a typical distance map generated by wavefront propagation. The lighter colors correspond to faster arrivals (shorter geodesic distances). The extracted path appears in purple towards the top. 78

Figure 31. Wavefront propagation from spine head leaks towards a neighboring spine that is closer to the dendrite. However, because of the choice of multiple end points at the dendrite center, the error is avoided by choosing the closest dendrite center point geodetically. This produces a correct spine axis (yellow line)..... 81

Figure 32 Several views of Spine9 Image rendered and rotated with detected spine paths superimposed in white. Rendering by voxx software. Image courtesy MBF-Bioscience..... 84

Figure 33 Spine4 Image rendering with detected spine paths superimposed in white. Rendering by voxx software. Image courtesy MBF-Bioscience. 86

Figure 34 Spine7 Image rendering with detected spine paths superimposed in white and a selected region zoomed in. The bottom 2 spines are detected together as one spine. This accounts for one missed spine. Rendering by voxx software. Image courtesy MBF-Bioscience. 87

Figure 35 Left: minimum intensity projections of spine1 (top), spine2 and spine3. Right: zoomed-in regions demonstrating error from noise near the dendrite surface. 92

Figure 36 Minimum intensity projections of spine4 (top), spine5 and spine6. Right: zoomed-in regions demonstrating error from high spine density and low resolution. 93

Figure 37 Dendrite flattening demonstration. Top shows a rendering of a flattened and straightened dendrite with the spines appearing to protrude upward from the surface. Bottom shows a close up view. The tool involves Cartesian to polar transformation and hence the resulting image is affected by rounding. Validation using this tool has yet to be proven useful..... 95

Figure 38 Group edit-based validation tool of the FARSIGHT toolkit suite. Upper panel: rendering of the dendrite with spine axes. Lower right: spine features. Left:

confusion table of two features with a user-selected cluster highlighted in yellow and reflected synchronously with the feature and rendering windows.....	96
Figure 39 SeedEditor validation tool from FARSIGHT toolkit. The top panel shows a rendering of a dendrite segment with the original seeds in faint yellow. The newly added seeds in blue indicate false negatives and the dark orange seeds are false positives.	97
Figure 40 Spine Ring Detector implementation flowchart: This details boxes A and B of Figure 21. (1-3) Detectors are created and the contents thresholded. (4) A FIFO queue serves as a buffer for the latest sequence of detectors with hits. (5) When there is detector hit discontinuity, the queue is considered full and is emptied by merging the contents. (6) Connected components are labeled and validated for further processing.	103

DEDICATION

To my first lovers, nourishers and teachers, my beloved parents,
As-Sayyid Abdulkarim and As-Sayyida Batoul,
For their unlimited giving and sacrifice,
For the core values that they have instilled in me,
No thankfulness of mine can ever do you justice.

To my beloved children, Batoul, Sima and Mohammad,
For impatiently waiting for dad to come home and play together,
For the nights with unread bedtime stories,
And for keeping the hope that dad will be done soon,
Your patience has been worthwhile,
I love you; I thank you,
We'll chase the rainbow,
And we will climb other mounts together.

To my beloved wife As-Sayyida Zahra,
Without whom I would not have returned to academic ranks,
Nor would I have achieved this milestone,
For the sleepless nights and long years of waiting,
For her patience with my shortcomings,
And for all the love and joy that fills our lives,
I dedicate this work.

ACKNOWLEDGMENTS

"من لم يشكر المخلوق لم يشكر الخالق." - النبي الاكرم (ص)

"Whoever did not thank the created has not thanked the Creator."- The Holy Prophet (P)

My journey towards completing this work has been exceptional in many ways. Had it not been for the support and encouragement of many individuals, it would not have seen light. In particular, I wish to thank my adviser Professor Badri Roysam for his continued support, his exceptional kindness and understanding, his patience, assurance and everlasting optimism. His advice and direction have been essential in bringing this work to fruition; his attention to important details has proven crucial for its effective dissemination.

I would like to thank my committee members for their valuable time and discussions; I am grateful to Professor Michael Wozny for all his support and help during my stay at RPI, to Professor Deanna Thompson for her challenging questions and essential critique that enhanced my approaches and refined important parts of my dissertation and to Professor Randolph Franklin for his valuable suggestions and advice.

I wish to express my deep indebtedness and gratitude to my dear friend Dr. Sabir Azizi and to his family for being my second family in the absence of mine; their love, support and all the help they generously availed to me have been exceptionally relieving and direly needed during the course of this work.

My colleagues at Professor Roysam's lab have been an amazing group of dedicated researchers and very friendly team. I wish to thank them all and in particular Yousef Al-Kofahy, Arunachalam Narayanaswamy, Amit Mukherjee and Raghav Padmanabhan for the valuable discussions and collaborative work. Special thanks are due to my former colleagues, now doctors, Omar Al-Kofahi, Alex Tyrrell, Amri Abdulkarim, Xiaosong Yuan, Andrew Cohen, Nicolas Roussel and Dirk Padfield.

I wish to thank my former IBM colleagues and friends Dr. Ali El-Zein and Dr. Fadi Zaraket for their support. I also wish to thank IBM for partially supporting my graduate studies at RPI.

I thank the colleagues at MBFBioscience for their help in data verification, especially Dr. Sue Hendricks and Allisa Wilson. I also would like to thank the researchers who kindly gave us access and permission to use their images: professors Josh Trachtenberg (UCLA), Fulvia Gheorghita (DBCM, UNIL, Lausanne, Swiss), Ted Ballou (Northwestern U), Judson Chandler (Med.U. of S.C), Jakub Jedynek (U.Mich), Dr. Pilpel and Dr. Bruno. Special thanks go to Professor Steve Potter (Georgia Tech) for the image data as well as informative discussions and valuable collaboration.

This work has been partially supported by grants from MBFBioscience, Inc. and Gordon-CENSSIS to whom I extend my gratefulness.

ABSTRACT

We present a methodology for automated segmentation of dendritic spines in three-dimensional (3-D) fluorescence microscopy images.

Dendritic spines have been linked with brain cognition and memory as well as drug interaction effects. Their significance has increased in the past decade with advances in 3-D fluorescence microscopy imaging especially for time-lapse analysis (4-D image data) in the context of pre-clinical studies. Segmenting spines is necessary to achieve accurate quantification across several snapshots.

The proposed algorithm models the dendritic shaft as a tube-like structure of varying diameter in noise studded with spines of 4 possible types, filopodia, and/or varicosities. Automated tracing of the dendritic backbone is accomplished by a previously developed algorithm that fits superellipsoids robustly and estimates the local intensity surrounding them. Once this is done, the spines are initially detected by a multi-step algorithm that first extracts the surrounding regions that deviate from the tubular geometry. A generalized likelihood ratio test is used to determine whether the initially segmented region represents a spine or just imaging noise. Because spine sizes are close to the achievable image resolution, and the fluorescent marker fluctuates in intensity, many spines appear fragmented and/or disconnected from the dendrite and thus require further analysis. For this, features such as the spine convex hull and size are quantified. Then, the spine centerline is extracted using a Fast Marching approach and a curve distance measure is used to merge candidate spine fragments.

To verify the effectiveness of the proposed methodology, a total of 14 3-D confocal microscopy images are analyzed. To obtain a broad sampling of the field, the test images were obtained from multiple laboratories, and represent multiple choices of subject animals, experimental conditions, magnifications, and imaging quality. A fully automated processing was performed without the need for user intervention. The results are shown to be superior to published state-of-the-art methods, and provide an enhancement margin over human expert detection.

1. Introduction

In this chapter we present an introduction to dendritic spines. Their discovery, biological morphology, and importance of viewing them in three-dimensional microscopy are briefly described. More details are provided in chapter 22.

1.1 Spine Basics

History: Previously considered insignificant artifacts on the dendritic shaft, the thorn-like protrusions of neuronal membranes first gained attention with Ramon y Cajal's (1852-1934) novel interpretations of brain tissue images in 1888. Since then, not only the name – dendritic spines – has been retained, but also much of Cajal's neural spine hypotheses (DeFelipe 1999; Garcia-Lopez, Garcia-Marin et al. 2006). Despite the primitiveness of his microscope and techniques, Cajal's elucidations have laid the foundations of modern neurology (Hamburger 1980).

Occurrence: Spines arise from the neuronal membrane at the soma, dendrites, or the axon hillock in tiny structures and in different directions (Harris, Jensen et al. 1992; Nimchinsky, Sabatini et al. 2002). They appear on most principal neurons in the brain such as the pyramidal neurons of the neocortex and the cerebellar Purkinje cells with variable densities reaching up to 15 spines per μm of dendritic stretch (Fiala and Harris 1999). The spine count on the dendrites of a single neuron can range from an order of thousands to that of hundreds of thousands.

Development: The process of spine formation continues to be under investigation, nevertheless, they are believed to develop from longer and thin structures called filopodia that predominate the first postnatal week of early brain development (Hering and Sheng 2001). Filopodia rapidly sprout from dendrites and then retract to be replaced by dendritic shaft synapses and/or spines. Immature spines have necks but typically lack (or have very small) heads.

Morphology: Although their morphologies are quite diverse, most (mature) spines have a bulbous head and a thin neck that connects it to the dendritic shaft. They are broadly qualified by the head volume ($0.01\text{-}0.8\ \mu\text{m}^3$) and neck length ($0.03\text{-}0.9\ \mu\text{m}$). The most commonly used nomenclature categorizes them into the following four classes (Harris, Jensen et al. 1992; Nimchinsky, Sabatini et al. 2002):

1. *stubby*, closely attached to the dendrite with no obvious neck
2. *mushroom*, has a narrow neck and a large head
3. *thin*, narrow neck and elongated stem with no head
4. *bifurcated* or *branched*, similar to the *mushroom* but with a split head.

Fiala and Harris (1999) indicate the dominance of these four types, especially thin-shaped spines, but nevertheless describe a continuum of length and neck deformations. Moreover, specific neurons exhibit additional spine types such as crook thorn and gemmules (Figure 1). The significance of spine dimensions (surface area, head volume, neck length) and shape has been correlated with the synaptic strength and proliferation. The larger the spine size, the stronger and more efficacious is the synapse.

Plasticity: Spines change significantly in shape, size and number, usually, within minutes. They can spontaneously appear and disappear although the larger mushroom type and the more mature spines tend to be the most stable. Plasticity is a major focus of active dendritic spine research, not only for understanding spine functionality, but also for analyzing reactions to stimuli in relation to memory and cognition.

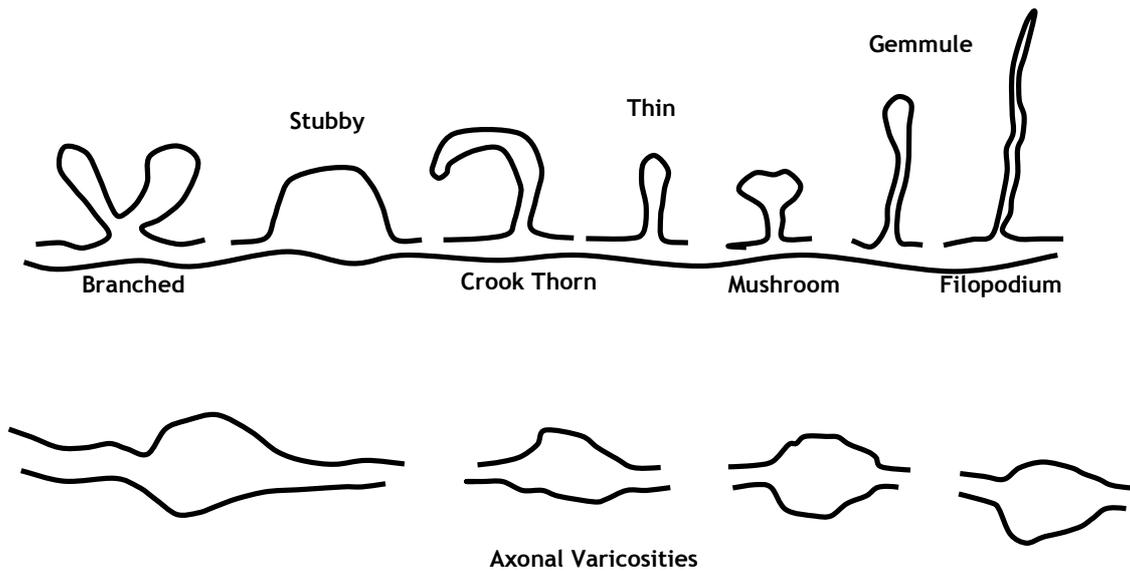


Figure 1 Neurite Sub-Structures(adopted from (Fiala and Harris 1999)): The four commonly-referenced dendritic spine classes are **Thin, Mushroom, Stubby and Branched**. They are classified by head size and neck length. **Crook Thorn** appears less often in the literature. **Filopodia** are considered **quasi-spines** appearing during initial stages of spine formation. **Axonal varicosities** are deformations of the axon shaft and may confuse a spine detection algorithm.

1.2 Dendritic Spines: Significance and 3D Imaging

Topics such as dendritic spine properties (shape, size, distribution, density, ...), plasticity, their relationship with the neuronal tree and their role within the synapse are the subject of ongoing research (Yuste and Bonhoeffer 2004; Garcia-Lopez, Garcia-Marin et al. 2006).

Although the complete functions of spines are still not well understood (Yuste and Bonhoeffer 2004), it is widely believed that spines are responsible for information exchange at the synapse. Numerous studies examine dendritic spine correlation with memory (Yuste and Bonhoeffer 2001), cognition and related disorders such as Alzheimer’s Disease (Akram, Christoffel et al. 2007) and fragile X syndrome (Comery, Harris et al. 1997). In recent years, advances in 3D imaging technologies have contributed to increased attention towards investigating spinal structure, development, as

well as drug interaction (Harris, Jensen et al. 1992; Nimchinsky, Sabatini et al. 2002; Trachtenberg, Chen et al. 2002; Potter 2005). For example, spine head volume, a significant attribute for spine classification and synaptic strength probing, requires 3D imaging for accurate measurement. Moreover, spine heads are now found to contain voltage-dependent ion channels (compartments). This leads to conclude that the spine is the smallest computational subunit of the dendrite (Sidiropoulou, Pissadaki et al. 2006).

Age-related spine changes were investigated using 3-D confocal images although spine counts were done on 1 μ m-separated 2-D slices (Duan, Wearne et al. 2003). A recent study on human brain tissue found a strong correlation between cognitive decline due to Alzheimer disease and dendritic spine loss (Akram, Christoffel et al. 2007).

In (Johnson and Ouimet 2004) 3D imaging was used for spine and filopodia length measurements at different time points in order to assess certain drug interactions (protein inhibitors) on spine growth. Although over 3000 spines were measured, there was no indication about whether the measurements were done manually or by automated software.

In the following chapter we present an overview of previous techniques and their advantages or disadvantages.

2. Literature Review

2.1 Spine Types and Dimensions

Harris et al. (Harris, Jensen et al. 1992) investigated different spine dimensions and their effects on dendritic synapse densities. It was found that spine classification could be deduced from geometric measurements of the spine volume, head, total length and neck diameter. An illustration of these spine parts is seen in Figure 2 and Figure 3 and their measurements are defined as depicted in Figure 4.

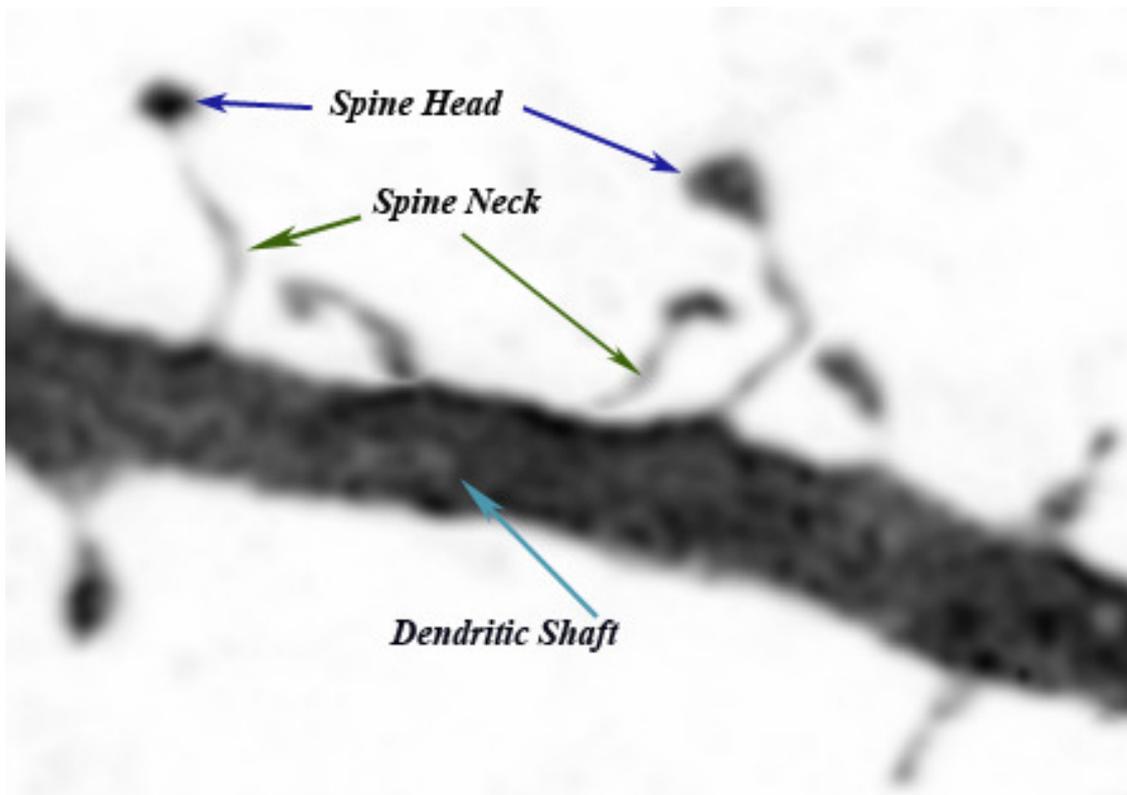


Figure 2 Spiny dendrite segment magnified to show spine head and neck. The dendrite shaft in this work denotes the dendrite proper.

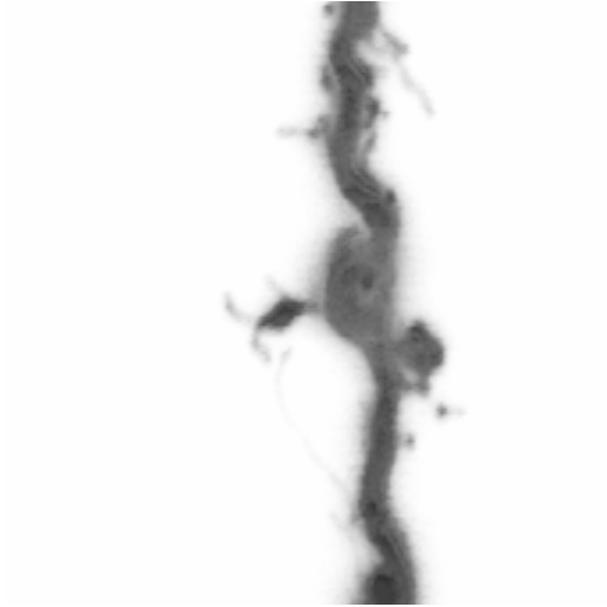


Figure 3 Example of a bifurcated (branched) spine. (Cropped from image spine2.tif, courtesy MBFBioscience and Dr. T. Ballou, Northwestern U).

Spine Type	Cross-section	Criteria
Thin		$d_n \leq d_h$ $d_n \ll L$
Mushroom		$d_n \ll d_h$
Stubby		$d_n \approx L$
Branched		> 1 head

Figure 4 Spine categories and defining features: Adopted from (Harris, Jensen et al. 1992). L =side length; n =neck; h =head; d =diameter. A spine is *thin* if the neck diameter is much smaller than the side length and slightly smaller or comparable to the head diameter. *Mushroom* spines have their neck diameter much smaller than that of the head while the side length is less relevant. If the length is almost similar to the neck diameter, the spine is labeled as *stubby*. A spine is *branched* if its head splits into two (or more); other measurements are then irrelevant.

It was found that the ratio of spine head volume to the neck diameter discriminates between mushroom spines and the rest. The total spine length to neck diameter ratio discriminates between stubby and thin spines. The first 5 rows in Table 1 show data extracted from the Harris et al. study. We added the last 3 rows to demonstrate typical ratios that are needed, as we shall see in Chapter 5, for setting the parameters of our proposed spine detector.

Harris et al.(Harris, Jensen et al. 1992) concluded that spine shape changes with time. Stubby spines were found to disappear with age; mushroom spines exhibit a wide range of size variability which may indicate a shape-volume continuum; thin spines appear to have more stability after formation. The study suggests different possibilities for branched spine formation (either splitting of a single head or merging of two or more individual synapses).

Filopodia, on the other hand, are characterized by their thin and up to $35\ \mu\text{m}$ -long protrusions (Arstikaitis, Gauthier-Campbell et al. 2008) with no head. This suggests that they are easily distinguishable from stubby, mushroom and branched spines but with less ease from thin spines. The role of filopodia remained unclear in the 1992 study, as they get lost between the early stages and maturation. However, as we stated in section 1.1, they appear more abundantly in neonatal brain development and rarely in mature dendrites. Interestingly, a more recent study has suggested that in addition to their transient appearance before spine or synaptic formations, filopodia may also form as a reverse transition from spines (Parnass, Tashiro et al. 2000). This indicates that the dynamics between filopodia and spines is more complex than originally believed. Nevertheless, Arstikaitis et al. (Arstikaitis, Gauthier-Campbell et al. 2008) indicate that no direct evidence has been found to illustrate the filopodia-spine transition.

Table 1 Data (rows 1-5) extracted from (Harris, Jensen et al. 1992). We added the last 3 rows. To distinguish between mushroom spines and the rest, they used the head volume to neck diameter ratios where it is much higher for the mushroom type. To distinguish between thin and stubby spines, the ratio of total length to neck diameter was calculated. Thin spines have the larger numbers. We added the length-to-dendrite-diameter ratio (last row) as a justification of our detector dimensions assumptions.

Adult Tissue	Stubby		Mushroom		Thin	
	Mean	STD	Mean	STD	Mean	STD
Total Volume	0.03	0.01	0.29	0.13	0.04	0.02
Head Volume	N/A		0.27	0.13	0.03	0.02
Total Length	0.44	0.15	1.50	0.25	0.98	0.42
Neck Diameter	0.32	0.13	0.20	0.07	0.10	0.03
Dendrite Diameter	0.78	0.36	0.81	0.24	0.62	0.26
Spine HeadVol/NeckDiam	N/A		1.35		0.30	
Spine Length/NeckDiam	1.38		7.50		9.80	
Spine Length/Dendrite Diam	0.56		1.85		1.58	
Post Natal Tissue	Stubby		Mushroom		Thin	
	Mean	STD	Mean	STD	Mean	STD
Total Volume	0.11	0.07	0.18	0.09	0.05	0.03
Head Volume	N/A		0.16	0.08	0.03	0.02
Total Length	0.65	0.38	0.95	0.30	1.40	0.39
Neck Diameter	0.43	0.07	0.21	0.12	0.13	0.05
Dendrite Diameter	0.81	0.29	0.83	0.37	0.70	0.33
Spine HeadVol/NeckDiam	N/A		0.76		0.23	
Spine Length/NeckDiam	1.51		4.52		10.77	
Spine Length/Dendrite Diam	0.80		1.14		2.00	

As seen in Table 1, spines may be less than $0.1 \mu m$ thick and would thus require good staining and high resolution imaging. Laser scanning confocal microscopes (LSCMs) can provide resolutions up to the diffraction limit of the radiation wavelength – typically $0.2 \mu m$ per voxel is close to the limit. This relation is formulated in the following equation:

$$d = 1.22\lambda N \approx \frac{0.61\lambda}{NA};$$

d is the pixel radius, λ is the radiation wavelength, N is the f-number and NA is the numeric aperture of the imaging system. Moreover, LSCMs provide easy 3-D imaging, and offer the ability to perform multi-channel imaging.

On the other hand, electron microscopes provide much higher resolutions because the wavelength of an electron is much smaller than that of a photon. However this is counterbalanced by price, maintenance requirements, and running complexities as compared to confocal microscopes. Hence image data sets are expected to be more readily available from LSCMs than electron microscopes. Typical volume images range from 512 to 1024 pixels in length and width, and from 30 to 150 slices in height (thickness) covering $50 \mu m$ to $100 \mu m$ (length and width) and $6 \mu m$ to $30 \mu m$ height.

2.2 Spine Detection Challenges

With increasing emphasis on quantitative measurements in biological research, accurate spine detection, modeling, classification and dynamic analysis has become crucial. Given the limitations of confocal microscopy with respect to spines, a dendritic volume image typically presents several challenges with respect to spine identification and segmentation. We illustrate this by a collection of images in Figure 5.

Spine formation, development and stability is reviewed by Yuste et al. (Yuste and Bonhoeffer 2001; Yuste and Bonhoeffer 2004). The studies survey different publications and underscore result discrepancies among researchers, some of which were ascribed to subjective factors in distinguishing & identifying filopodia and various types of spines.

Hypothesized roles of filopodia and the different types of spines were given in a recent 3-D quantitative study (Garcia-Lopez, Garcia-Marin et al. 2006).

Thus the necessity for objective modeling and classification of fluorescent structures in neuronal imaging has been established. The problem is that the state of the art solutions continue to be inadequate compared to the need. In particular, a successful spine detection technique must satisfy the following criteria:

- **Accuracy:** the number of false negatives (misses) and the number of false positives (alarms) should be as close to zero as possible.
- **Generalizability:** the algorithm should apply to a variety of 3D images possibly captured by different biologists and microscopy conditions.
- **Automation:** user intervention is undesired in order to allow for reliable batch processing.
- **Parameterization:** ideally, it is desired that no tunable parameters be required. However, in the event that user input is needed, setting algorithm parameters should be minimized. Moreover, parameters must be sufficiently intuitive for a non-technical user to the extent possible.

2.3 Previous Spine Detection Techniques

Developments in biomedical imaging technologies have encouraged analysis of dendritic spines and time-lapse dynamics. Manual spine detection was used by different groups (Capani, Martone et al. 2001), being presumably accurate and providing acceptable ground truths. However, looking closely at neuronal images, one easily discovers a great deal of variability in spine size, number and density. Moreover, dendritic thickness variability can easily cause the addition or deletion of a stubby spine. This makes manual inspection error-prone, non-repeatable and highly subjective. Manual counting is also slow, especially in 3D images, and fails to provide accurate morphologic measurements. Current automated spine detection methods mostly utilize thresholding and skeletonization as a basis to extract the dendrite backbone prior to additional image analysis. A grayscale threshold from the image histogram was utilized by (Rusakov and

Stewart 1995) to produce the medial axis in 2-D. The spine skeletons are manually separated from the dendrite, and spine length is estimated in 3-D using stereo images.

A parametric model of cylinders with hemispherical ends to fit the shape of dendrites and spines was used by (Herzog, Krell et al. 1997). The image is required to have no more than a single dendrite, and user intervention is needed for initialization. A hull is created around the modeled dendrite leaving a gap in-between. The authors indicate the need for higher-resolution images in order to obtain more accurate modeling. However, with current state-of-the-art 3-D imaging methods, higher resolution images are associated with magnified noise around the dendritic shaft making the dendrite-hull model quite inaccurate for spine detection. Moreover, short and detached spines are easily missed with this approach. In (Herzog, Niese et al. 2006) the cylindrical model is further developed into a growing cylindrical model where several directions for growth are considered. The mean intensity is used to determine the best next direction. As we shall see later in section 3.4.3, this exhaustive search approach is computationally expensive and is sensitive to initialization and quantization.

An automated 3D spine detection algorithm using geometric heuristics was proposed by (Koh, Lindquist et al. 2002). The algorithm starts with image skeletonization which, alone, does not discriminate between the dendrite and other sub-cellular structures (such as spines, filopodia, and cell debris). Therefore a post-processing trace, based on minimum deviation angle, is performed to eliminate extra skeletons and determine the final dendritic medial axes. Spine detection depends on connectivity with the dendrite surface. Therefore separate approaches were developed for each case. Verification of the detected spines was handled by geometric morphology analysis with respect to the dendrite shaft. A post-processing algorithm examines neighboring spines for possible merging. A reported drawback of the method is reduced performance with faint dendrites and spines. Also, dendrite discontinuity due to segmentation (thresholding) in 3-D images remained to be fixed (Weaver, Hof et al. 2004).

An automated 3D reconstruction tool for neurons was developed by (Evers, Schmitt et al. 2005); however, the user is required to define the spine origins on the dendrite. Wiseman et al. (Wiseman, Capani et al. 2002) used thresholding followed by intensity autocorrelation analysis to count dendritic spines in 2D images. Their results were

comparable with manual counting with an error tolerance of 15-20% (Wiseman, Capani et al. 2002).

Skeletonization of the medial axis followed by an implementation of a double-grassfire procedure to detect spines was proposed by (Xu, Cheng et al. 2006). Two waves were propagated in opposite directions, one from the dendrite shaft and another from candidate spine tips. The maximum distance covered provides an indication of the spine tip. The area differential between both wavefronts is calculated at each step and the maximum is found. The main advantage of this technique lies in the hypothesis that the spine is best separated from the dendrite shaft at the peak wave difference. However, using a global threshold to segment the fluorescent structures is always a coarse approximation for actual 3-D microscopic images and presents a generalization obstacle. Besides not showing evidence of thorough corroboration, this method has other limitations, such as assuming that “short” skeletons belong to spines while “longer” ones belong to dendrites, and that spines protrude at “sharp” angles from the dendrite process. As spine size decreases, more spines will be missed. Obviously image discontinuity artifacts may lead to erroneous spine/dendrite shaft separation. Moreover, there is no model for verifying detected spines; hence, the algorithm requires further development in order to achieve more accurate and more extensible results.

A recently developed technique has been devised by (Rodriguez, Ehlenberger et al. 2006) for 3-D volume representation using globular ray growth and assuming an underlying cylindrical model for the dendrite shaft. The algorithm is initiated by image skeletonization and a user-selected location on a dendrite. This becomes a central *core* for creating a large number of emanating rays (“*rayburst*”). As the rays of the rayburst are propagated, the local intensity is estimated by interpolation at each voxel-crossing. Deciding whether to extend the ray further is based on a dynamically-estimated intensity threshold. Once ray extension halts on all rays, the minimum distance between two opposite rays is chosen as the local dendritic diameter estimate. This technique is implemented as the basis for representing the entire 3-D dendrite structure and the associated spines. Among the limitations of this technique is the requirement of star-shaped convexity for 3-D structures prohibiting full segmentation of curvy spines. The

implementation in NeuronStudio is semiautomatic, requiring the user to select an initial dendrite for the rayburst core.

In (Rodriguez, Ehlenberger et al. 2008), the authors proposed further enhancements to their rayburst model. Detached spine fragments were re-attached to the dendrite by utilizing a Euclidean distance threshold. While this approach works well for a spine whose main axis is (almost) orthogonal to the dendrite shaft, it is unreliable for spines with oblique axes. The problem is further exacerbated if the axis is curvy.

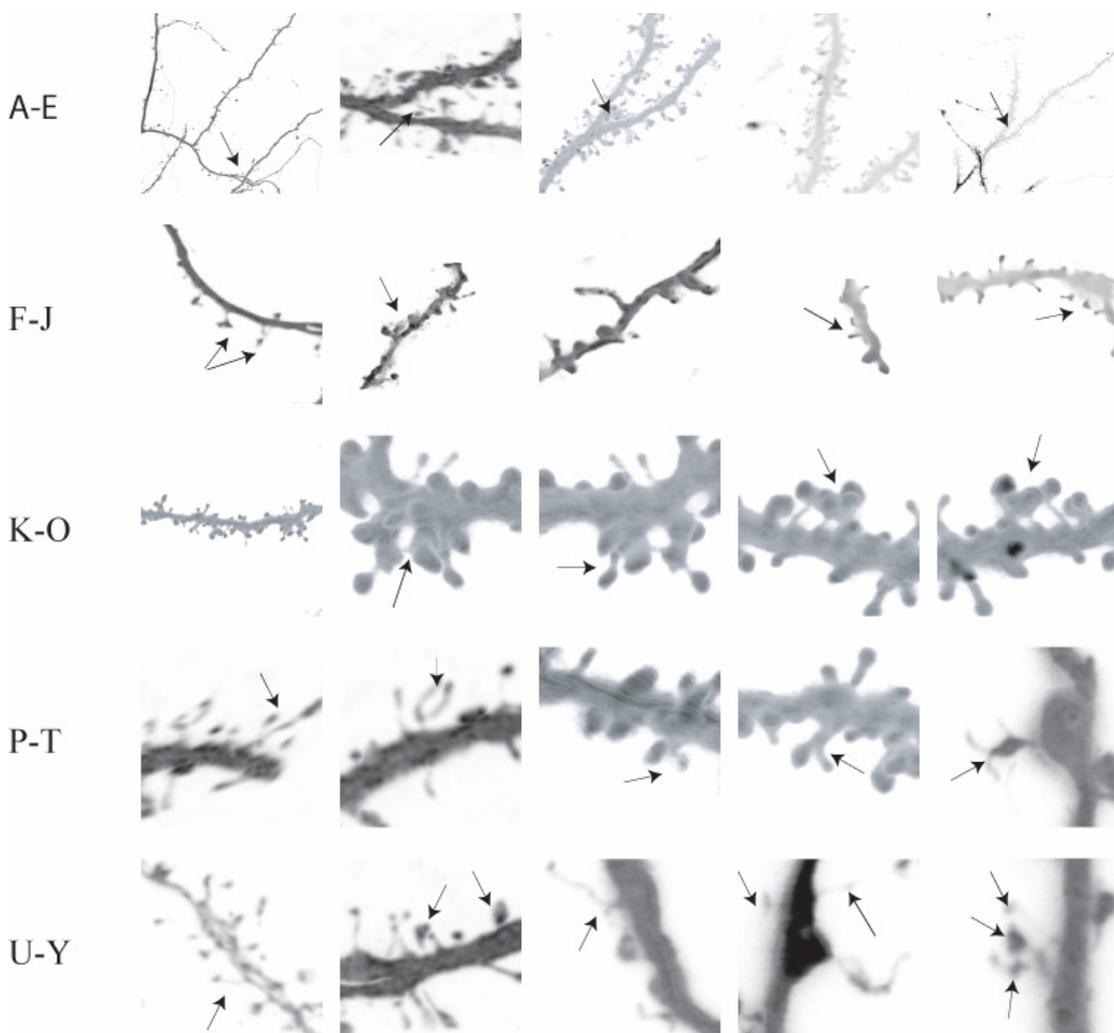


Figure 5. A gallery of dendrite and spine images rendered in 3D depicting subject variety. A-E: zoomed out dendrite views in different images where branching can influence spine identification. F-J: spines on dendrite segments showing variability. K-O: dendrite segment; zooming in: frontal view of a spine cluster with stubby and thin spines; dorsal view; another cluster of thin and mushroom spines; dorsal view.

P-T: examples of branched(bifurcated) spines; notice 2 branched spines in P; a faintly identifiable branch in Q-R; clearer in S; T could be missed for other types. U-Y: examples of thin, stubby, and mushroom spines.

2.4 Summary of Contributions

Motivated by the importance of dendritic spines and the challenges outlined above, this thesis focuses on automatic spine segmentation from 3D grayscale LSCM (Laser Scanning Confocal Microscopy) images.

The main contributions are:

- Fully automated spine segmentation algorithm: our method robust to feasible parameter changes and can be run in batch from start to finish. Current published techniques require some user settings and/or threshold determination specific to their approach.
- Our algorithm imposes no strong restrictions on image magnification, type of animal tissue nor does it require extreme imaging quality. It is demonstrably robust to image variability, noise, outliers and imaging artifacts.
- We propose a novel method that reconnects unattached spines back to dendrite shafts utilizing geodesic tractography. This method improves upon published state-of-the-art techniques that utilize Euclidean distances regardless of underlying intensities.
- Our technique is successful in finding spines that are hard to detect by mere visual inspection thus improving on human detection by as much as 30%.
- Our method introduces the usage of Fast Marching algorithms to extract spine features and help in segmentation validation.
- An automated graphical tool for a user to inspect and if needed, edit the tracing results for dendritic shafts.

Additional Contributions are:

- Open source implementation in C++ and ITK
- 3D verification tool for labeling results, extendible to other applications.

3. Spine Detection Foundations

As with most biological imaging applications, the problem of dendritic spine image analysis presents the challenge of generalizability. There are two issues in this regard: (i) the intrinsic design of the algorithm; and (ii) the problem of setting algorithm parameters. The success rate of an image analysis algorithm can vary considerably with the lighting conditions, signal-to-noise ratio and the imaging modality in question. Moreover, image analysis is almost always dependent on parameter settings thus hampering fully generalized automation over multiple images. From the literature survey on existing spine detection methods, it is easily seen that the challenge lies in overcoming a tradeoff between over-detection (false positives), under-detection (misses).

This chapter outlines the technical motivation behind our spine segmentation approach. We start with the need for dendrite shaft tracing followed by a brief description of existing techniques. Based on this survey, we propose to use the model-based tracing algorithm described by Tyrrell et al. (Tyrrell, di Tomaso et al. 2007). Then we describe existing spine detection approaches and explain their inadequacy.

Basic Assumption: Because of the highly specific nature of fluorescent staining used in the specimens, it is reasonable to assume that bright foreground pixels in the image do indicate the biological structures of interest, and vice versa. This assumption makes it safe to treat detected structures within the field of view as a *potential* candidate for the model being tested. Cell debris, tissue fragments and near-dendrite noise may slightly increase false positive errors; however, this issue is dealt with in post-processing by feature discrimination and edit-based validation.

3.1 The Dendrite as a Primary Structure

Since spines, sessile or detached, are sub-cellular structures depending on and surrounding the primary dendritic body, the (stubby, mushroom or branched) spine head or (thin) spine tip can only be present in the close vicinity of dendrite shafts.

Table 1 has an entry for the ratio of spine length to dendrite diameter. That entry indicates that a typical spine head or tip is usually less than five dendrite diameters away from the dendrite surface. Fluorescence detected sufficiently away from any dendrite should not be considered among spine candidate voxels or spine fragments. Therefore it becomes intuitive to rely on dendrite shaft segmentation and measurements before proceeding with spine detection.

Previous tracing methods for tubular structures have been proposed in the literature. Skeletonization for medial axis extraction was used by (Cohen, Roysam et al. 1994) for neuronal tracing. A graph-theoretic model of the neuron is generated from the medial axis in order to identify branch segments. An efficient tracing algorithm was successfully devised by (Al-Kofahi, Can et al. 2003) using a grid of seed points with variable-size templates for boundary-matching in 3-D. Importantly, this method has a high failure rate when tracing spiny dendrites. While their technique produces reliable results in many images, in general it requires user-intervention for parameter setting. Cohen and Deschamps utilized level sets and fast marching algorithms to segment 3D tubular structures and extract centerlines (Cohen and Deschamps 2007). At the time of this writing, their procedure could not be evaluated in terms of speed or accuracy; nevertheless, with level set methods it is not easy to exclude attached spines from the neurite segmentation as we have verified in our own experiments.

3.2 Goals for Ideal Neuronal Tracing

Since it is desirable to achieve accurate dendrite measurements, the following goals are required for a neuronal shaft tracing algorithm:

- Accurate estimate of local dendrite shaft orientation and thickness in three-dimensional space: this is important in order to provide a local frame of reference when evaluating the existence of spines within a spatial neighborhood of the dendrite surface.
- Avoidance of structures (spines, filopodia and cellular debris) that surround the dendrite surface: because of the fluorescent continuity between the dendrite surface and the connected spines, this presents a challenge to any tracing

methodology especially those based on level sets since they only rely on the image intensities.

- A reliable estimate for local foreground and background intensities: it is assumed that the local intensity profile is the same for dendrites and spines, i.e., the fluorescent stain labels dendrite backbones and spines equally.

3.3 The Dendrite Model

As mentioned above, a reliable tracing technique for the neuronal structures is required before attempting to detect spines and related structures. A geometric shape model, preferably with a closed-form equation, would best match our specifications since it would have built-in dimension (such as radii) and orientation estimates. However, this will require the shape model to be flexible enough to fit within the curvilinear geometry as well as the asymmetry of the dendritic shaft.

Our choice of dendrite tracing model is the one developed at RPI by Tyrrell et al. (Tyrrell, di Tomaso et al. 2007). It satisfies the above-mentioned criteria, but it is fully automatic to allow batch processing of images with no user intervention. Although initially implemented for microvasculature segmentation in 3-D images with high success, the algorithm has also proven successful for neuronal images. The main limitation of this algorithm is the fact that it often fragments dendrite traces and it can confuse spines for dendrites. These limitations are addressed in this work with some post-processing algorithms.

In chapter 4 the model-based tracing algorithm and our proposed corrective enhancements are described. The set of corrected traces is the basis for our proposed spine detection algorithm and we term this process (tracing and enhancement) **Phase I**.

3.4 Simple Spine Detection Approaches

As sub-cellular structures that are close to neuronal dendrite surfaces, spines are detected consequent to dendrite shaft segmentation. In order to motivate the proposed spine

detection approach, we briefly present three initial methodologies that have proven to be promising but nevertheless inadequate.

3.4.1 *First Approach: Volume Difference and Residue Segmentation*

Following the model-based trace, it is easy to reconstruct the dendrite shafts using their accurate model (Figure 6 and Figure 7). The reconstructed volume is then subtracted from the original image; henceforth theoretically, the residual voxels should belong to spines and/or other cell debris surrounding the dendrite (Figure 8).

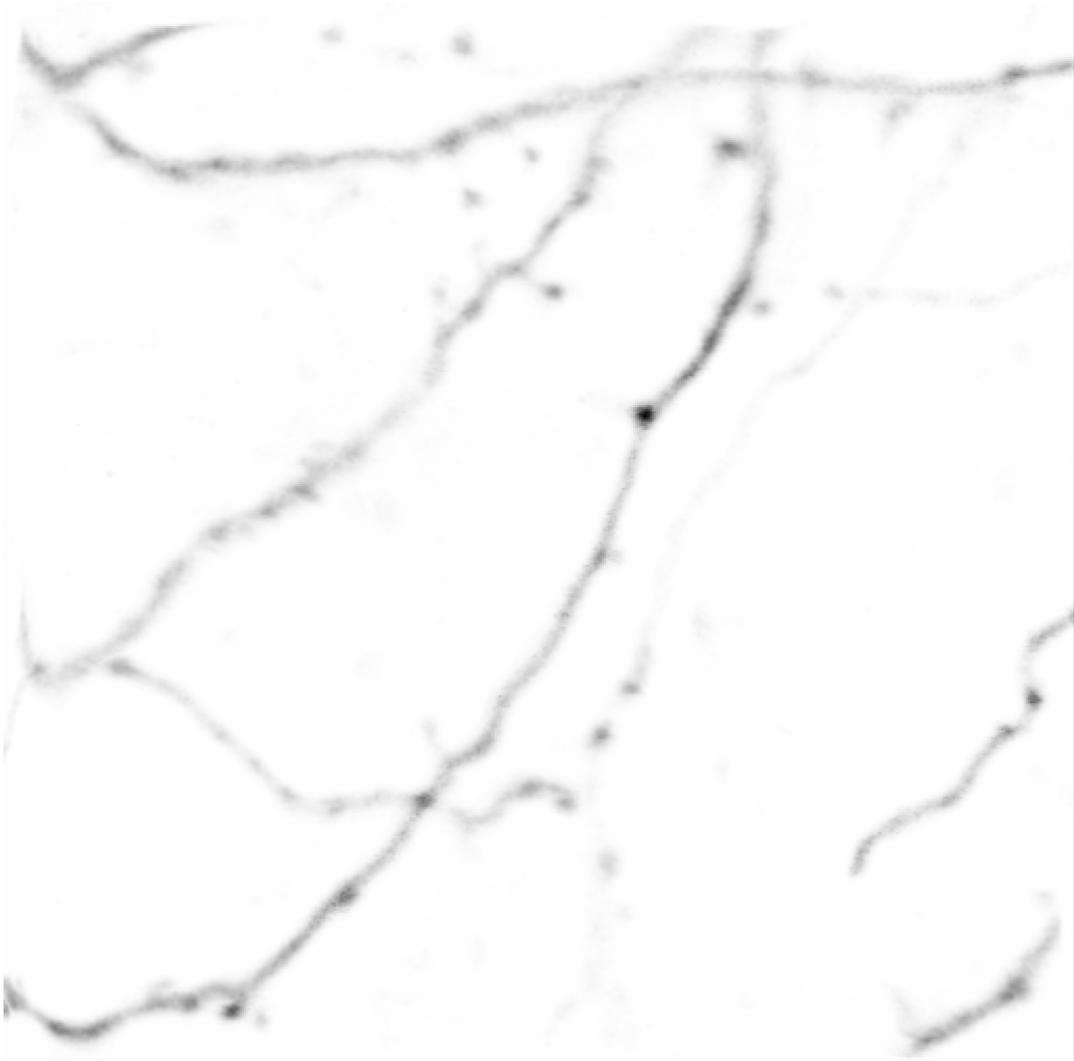


Figure 6. Day4 image –Courtesy Prof. Joshua Trachtenberg (UCLA). Maximum intensity projection.



Figure 7 Day4 – reconstructed dendrite shafts using super ellipsoid fitting superimposed on the original grayscale projection (Figure 6). Residual fluorescence can be easily seen; some being valid spines. Two spines have been erroneously traced and hence missed after subtraction.

Further binarization and morphological operations should ideally lead to good spine segmentation. However, our experiments uncovered several problems with this approach:

- Even though the tracing model we used is robust, the reconstruction cannot totally account for all voxels belonging to the 3D dendritic body mainly due to over-fitting. There are no guarantees that the model fits perfectly on the dendrite

surface at each tracing element due to imaging artifacts and model quantization. Thus, subtracting a reconstructed dendrite shaft from the original image creates a residual ghost effect.

- The automatic model may over-fit the dendrite shaft onto an attached spine “concealing” the spine within the estimated dendrite surface. This will cause many stubby and short spines to be falsely eliminated by subtraction.
- Imaging artifacts around the dendrites create false positives, enlarged spines and spine clusters that are difficult to separate.
- Even if spines were correctly detected, it is hard to determine to which dendrite they belong near a branch. Such errors would carry over to further validation and dynamic tracking processes.

Nevertheless, there are advantages to the approach such as isolating most of the dendrite shaft and retaining most spines; however, these above disadvantages result in too many false positives thus rendering it inadequate for spine segmentation. Table 2 has a typical result of this approach in comparison with that of the next section.

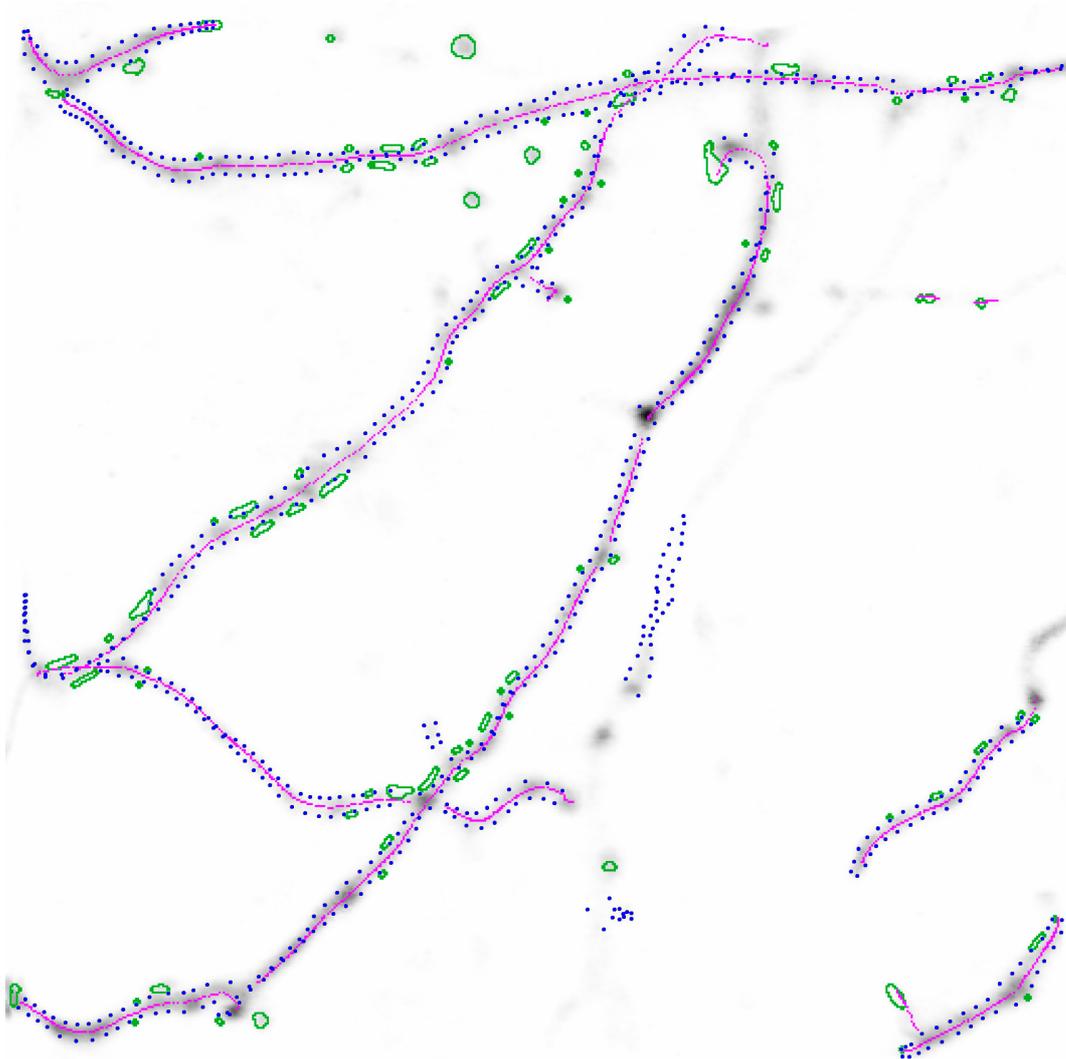


Figure 8 Day4 – residual segmentation. Dendrite shafts (blue) subtracted from original followed by morphological operations to detect spines (green). Noise near the dendrite creates false positives. Thresholding involved in binarization causes actual spines to be missed.

3.4.2 *Second Approach: Level Set Segmentation*

In order to overcome the above problems with over-fitting and noise, an attempt was made to fully segment the dendrite using level sets. In general, a level set formulation requires the solution of a partial differential equation of an energy function in order to propagate a wave-front through a particular medium. Because this entails exhaustive computation, and especially in large 3D images, the fast level set technique presented in

(Shi and Karl 2005) has been implemented. (Further detail of level set theory is presented in Chapter 6). The energy function (speed field) F used in this experiment has been calculated using the local foreground and background estimates \hat{f} and \hat{b} (see Section 4.2 for details):

$$F(x, y, z) = -\left|I(x, y, z) - \hat{f}\right| + \left|I(x, y, z) - \hat{b}\right|. \quad (3.1)$$

One point belonging to the centerline is chosen as the starting level set seed. What makes this method fast is that the level set function $\phi(x, y, z)$ is made discrete-valued as follows:

$$\phi(x, y, z) = \begin{cases} 3, & \text{if the point is outside the wave front} \\ 1, & \text{if the point is exterior to the boundary} \\ -3, & \text{if the point is inside the wave front} \\ -1, & \text{if the point is interior to the boundary} \end{cases}. \quad (3.2)$$

This allows us to avoid solving a PDE altogether because we only need to update the boundaries by checking the value of $F(x, y, z)$. Moreover, our tracing algorithm provides enough information to restrict the wavefront propagation along the dendrite surface.

As the wave propagates, it covers the dendrite shaft and its attached spines. The resulting distance map $\phi(x, y, z)$ is then passed through a Harris edge detector (Harris and Stephens 1988) in order to find sharp corners. With some empirical threshold, the latter are used to identify attached spines. The remaining unattached spines are obtained by morphological operations on the subtracted image. Figure 9 shows an example of a promising result while Figure 10 shows a challenging one.

The limitations of this approach are summarized as follows:

- The level set wave advances along the dendrite by the energy function estimated from tracing. Besides the additional computation this imposes on the tracing algorithm, the function is limited by the imaging artifacts. Particularly, dendritic discontinuities halt the propagation thus introducing false sharp corners.
- Due to limitations of imaging and/or staining, apparent holes on the dendritic surface cause another source of false edges and corners.

- In the best case, a correct detection marks a successful spine-localization, albeit devoid of proper segmentation for further validation.
- Even though Shi and Karl's method does not solve the PDEs, and is faster than conventional level set implementations, it remains too slow with 3D images.

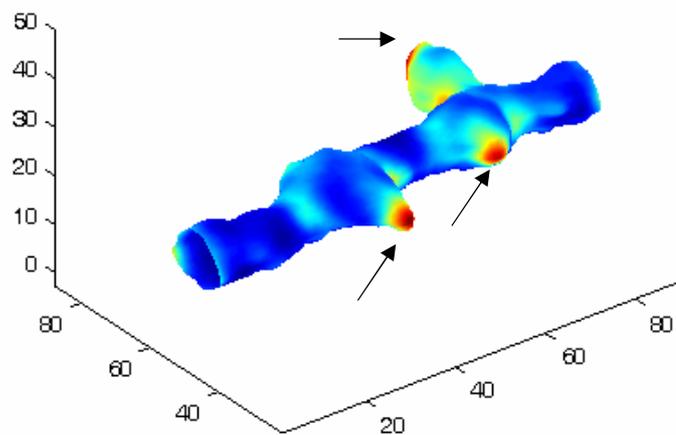


Figure 9 Level set segmentation: Spine tips detected as sharp edges. However as seen in the next figure, this technique requires further validation.

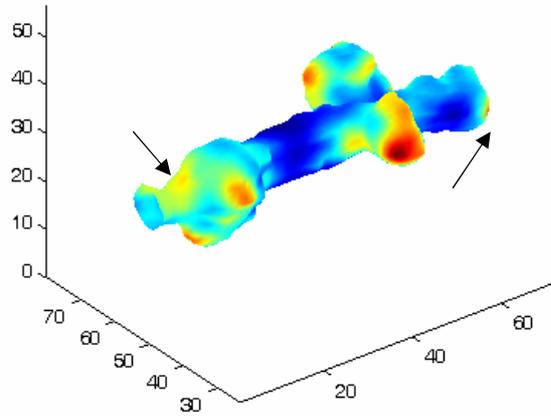


Figure 10 Level set segmentation: Dendrite surface irregularities creating false corners.

3.4.3 Third Approach: Revolving Cylinders

This is a model-based spine detection approach requiring initial model-based dendrite segmentation. Herzog et al. published their results using the same principle although with some differences (Herzog, Niese et al. 2006). Here we describe our experiments and discuss the results.

A tracing element from Phase I is a single super-ellipsoid fitted to a small sub-volume of the dendrite. These attributes (see the shape model in Section 4.1) are utilized to determine a suitable radius and height for the cylindrical detector based on the local dendrite geometry: super-ellipsoid center μ , axial directions $\mathbf{P}_{1,2,3}$ represented by quaternions \mathbf{q} , three radial lengths (σ_z in the dendrite shaft (major axis) direction and σ_y & σ_x in the dendrite shaft cross-sectional plane), the local foreground and background intensity estimates \hat{f} and \hat{b} .

The idea is to walk along the dendrite by steps equal to one or more elemental traces based on a user setting. At each step we create a cylinder C , position it on top and

elevate it parametrically by θ^{rad} from the dendrite surface, and spin it 360° around in search for possible spines of any type (Figure 11). This means 3 rotations are needed for each detector position.

3.4.3.1 Implementation Details

The following paragraphs describe the detector implementation in more detail.

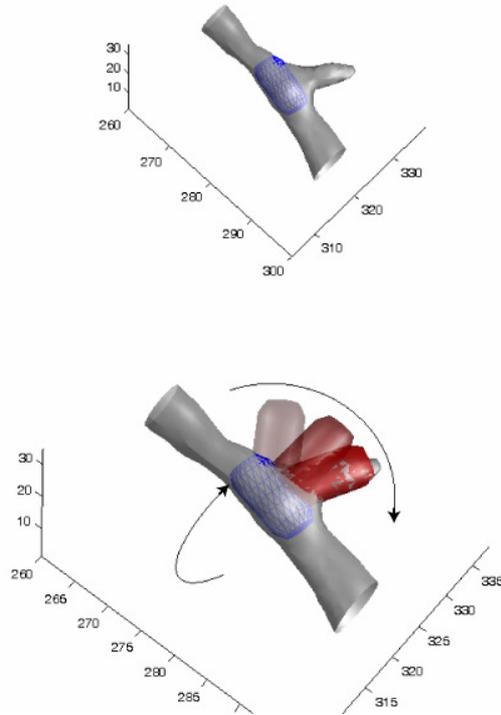


Figure 11 Image illustrates how a cylindrical model (red) is revolved around the dendrite model (blue) in search of potential spines in the neighborhood. Two rotation axes are needed: one normal to the dendrite axis to create an “elevation”, and another one along the dendrite axis to create spin.

1. The radius of C must not exceed the step regional width. If the region is defined to be one super-ellipsoidal segment of major axis length (along the dendrite shaft)

σ_z , the radius $r_C = \sigma_z \times 2$. If more than one ellipsoid defines a step region, the cylinder radius becomes $r_C = 2 \sum \sigma_z$.

2. The height of C is chosen to be a variable multiple of the maximum cross-sectional width of the ellipsoid: $h_C = \alpha \times \max(\sigma_x, \sigma_y)$ where α is either estimated empirically, or chosen = 1 if using split cylinders (below).
3. Orientating the cylinder along with the ellipsoid is the first step to place it in the correct detecting position. In order to avoid the *gimbal lock* problem associated with Euler rotations, we utilize quaternion rotation as described by (Shoemaker 1985) and derived in (Tyrrell, di Tomaso et al. 2007). A quaternion, defined as a 4-tuple $\mathbf{q} = \langle w, x, y, z \rangle$, provides for a useful rotational mapping, known as the axis/angle representation. We set $w = \cos(\phi/2)$ and $\langle x, y, z \rangle = \sin(\phi/2)\mathbf{n}$, where \mathbf{n} is the axis (unit vector) about which a rotation by ϕ can to be performed. This makes $\mathbf{q}(\phi, \mathbf{n}) = (\cos(\phi/2), \sin(\phi/2)\mathbf{n})$. Each tracing element has its own frame of reference described by three directional quaternions, indexed by 1,2 for the minor axes and 3 for the major axis, $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$. To align our detector with the dendrite model, three rotations are combined using a quaternion multiplication $\mathbf{q}_c = \mathbf{q}_1 \cdot \mathbf{q}_2 \cdot \mathbf{q}_3$. We extract a rotation matrix from each quaternion by

the equation:
$$\mathbf{R}(\mathbf{q}_c) = \begin{pmatrix} 1 - 2(y^2 + z^2) & 2(xy - wz) & 2(xz + wy) \\ 2(xy + wz) & 1 - 2(x^2 + z^2) & 2(yz - wx) \\ 2(xz - wy) & 2(yz + wx) & 1 - 2(x^2 + z^2) \end{pmatrix}.$$

(3.3)

Provided the composite quaternion is normalized to unit length, the resulting rotation is guaranteed to preserve orthogonality with no risk of gimbal lock. Hence, C is aligned with the major principal axis of the ellipsoid by:

$$C' = \mathbf{R}(\mathbf{q}_c) \times C. \quad (3.4)$$

4. The second rotation elevates the cylinder above the dendrite shaft. This is done by rotating C' about either one of the minor ellipsoid axes (it does not matter which

one because it will be spun 360° around it anyway): $C'' = \mathbf{R}(\mathbf{q}_{1,\theta}) \times C'$. The angle θ is varied in a step-wise fashion to cover 180° along the dendrite exterior.

5. The model is translated to the dendrite surface by offsetting it in the other minor axis direction: $C''' = C'' + h \times [\mathbf{q}_{2,xyz}]^t$.

6. Finally, the detector is step-wise spun around the ellipsoid major axis to cover all possible locations around the dendrite; the transformed cylinder is then:

$$C^T = \mathbf{R}(\mathbf{q}_{3,\phi}) \cdot C'''.$$

3.4.3.2 Verification

The voxel intensities within the cylinder are interpolated with the original image. A hypothesis test is performed to detect whether a spinal structure exists around the candidate region. Since the grayscale appearance and noise models are assumed to be similar to the model fitting algorithm, we found it natural to use a generalized hypothesis test that is similar to the one proposed by (Tyrrell, di Tomaso et al. 2007) based on the following log-likelihood ratio test:

$$\Lambda = \sum_{\mathbf{x}_k \in C^T} |I(\mathbf{x}_k) - \hat{b}| - |I(\mathbf{x}_k) - \hat{f}| \stackrel{H_0}{<} \tau \stackrel{H_1}{>} \quad (3.5)$$

where \mathbf{x}_k are the pixels inside the detector template, τ is a user-defined threshold, H_0 is the null hypothesis that no spine is detected.

We ran a number of heuristic tests in order to verify the legitimacy of the candidate spine within the context of the dendritic network .

- First, there is a possibility that the spinning cylinder might have hit a part of the current dendritic surface, especially when the elevation angle is tilted off of the normal direction, and the neighboring region is at a convenient curvature angle with the current one. Such an intersection leads to a *false alarm* since the candidate spine is in fact part of the primary structure in the image. A similar situation arises when the spinning cylinder “hits” another neurite in the 3D space

surrounding the current region. Hence, the detected foreground voxels are checked for intersection with the saved dendrite voxels (Phase I).

- Second, we assume that there can be at most one spine around a particular candidate region; therefore, a vote is held among the successful fits that passed the first verification pass. The candidate spine with largest volume is saved and the rest are discarded.
- Finally, from the nature of neural spines, we assume that a spine should not surround the tubular structure of the neurite over more than 180° . This eliminates further false hits when the dendrite local geometry is too irregular/jagged.

3.4.3.3 Discussion

3.4.3.3.1 Candidate Regions

A valid question is “what defines the candidate region around which a spine is to be sought?” Considering the dendrite model, each super-ellipsoid represents a small segment of arbitrary length at the major axis. Careful examination of the spine detection results reveals that these segments are mostly smaller than the spines; moreover, the segments overlap arbitrarily as well. If we know *a priori* the range of expected spine widths, and image scales, it would have been straightforward to combine an ensemble of dendrite model segments to create a cylindrical detector of suitable radius. However, this approach will require user intervention and would make the method difficult to generalize across specimens.

Similarly, the cylinder height is another source of segmentation inaccuracy. If the cylinder is too short, a portion of the spine is missed; if it is too long, the percentage of spine fit within the cylinder is diminished, possibly to the extent below a minimum acceptable threshold.

3.4.3.3.2 Robustness

In order to enhance robustness and circumvent the above problems, we made the following enhancement: we would start with a cylinder width as small as an elemental

trace length, and use maximum possible height (5 times dendrite width). If the detector hits, the model is iteratively transformed in height and width to produce optimal fit.

3.4.3.3.3 Results and Conclusion

The rotating cylinders approach has been tested on several images of variable magnification and spine density. In general it has been successful in detecting spines of different sizes; however, the major drawback remains in the number of false positives caused by fragmented spines, noise, and inexact matches between the dendrite model and its surface.

Even with the model deformation enhancement, the improvement was limited especially with detached spines due to the need for offsetting the detector a certain distance from the dendrite surface. This introduces an additional parameter to optimize. Moreover, since we need to orient the detector in steps of two angular arcs, the search for optimal parameters over all orientations (5 degrees of freedom) slows the algorithm down tremendously. Quantization introduces another source of error. Figure 12 shows spine detection on a typical low-density image and Table 2 compares the results between rotating cylinders and volume difference and residue segmentation (Section 3.4.1). Figure 13 shows results on a high spine-density image.

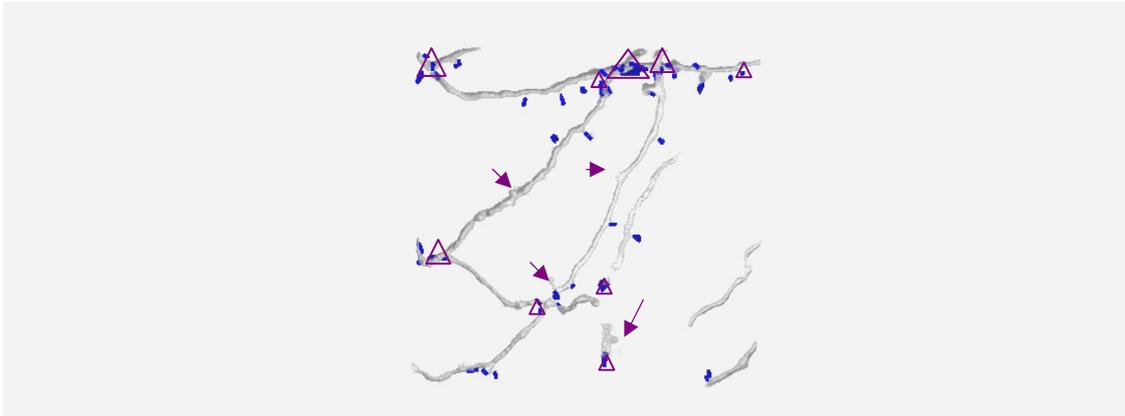


Figure 12 2D-projection of Day4 Image rendering, with detected spines superimposed in blue. See Figure 6 for original image projection. 47 spines were found by the rotating cylinders approach, 27 true positives, and 20 false positives (under triangle areas); there were 9 misses (some illustrated with arrows; others invisible in this projection).

Table 2 Performance comparison on a typical image (Day4) between the volume difference and rotating cylinders approaches. Although the latter performs significantly better, the error rates suggest further enhancements are needed in both methods.

Ground Truth:	True Positive	False Positive	False Negative	Total Error
36 Spines				
Volume Difference Seg.	26	63 (175%)	10 (28%)	203%
Rotating Cylinders	27	20 (55%)	9 (25%)	80%

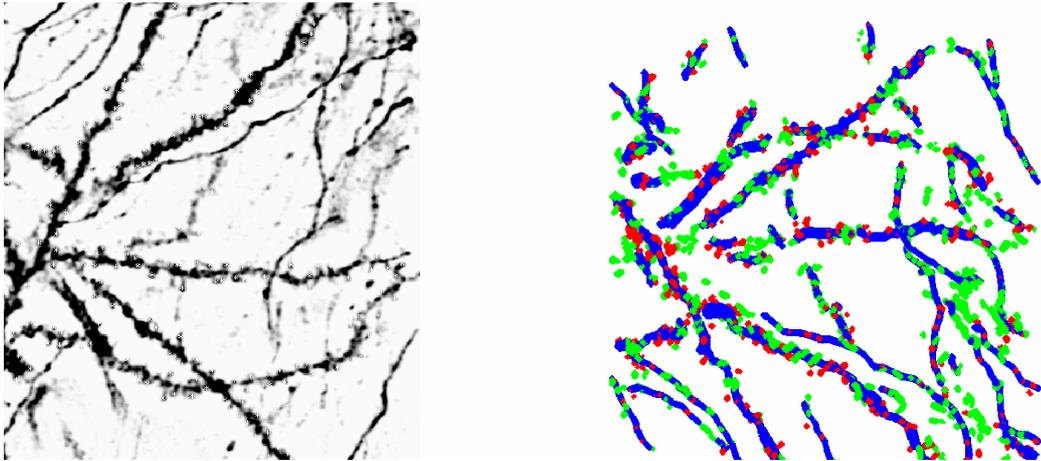


Figure 13 2D projection of original image (left) and reconstructed image (right) with spines detected by rotating cylinders. Two cylinder types were used here: red cylinders rotate close to the dendrite surface while green ones are at an offset determined by dendritic width. Over-detection is easily seen with multiple cylinders detecting the same spine. The approach results in a large number of false positives and with several parameters requiring user setting, it is hard to generalize. Manual labeling accounted for 214 spines while automatic detection gave 1334. The over-count is too large and the image is too dense to discriminate them from correct counts.

3.5 Chapter Summary

We described the spine segmentation problem as a hierarchical detection of primary image structures (the dendrite shaft) and secondary structures (spines). Spine segmentation quality depends on the accuracy of segmenting the dendrite shaft (Phase I) for which Tyrrell's (Tyrrell, di Tomaso et al. 2007) model has been chosen.

The limitations of previous spine detection work has been presented and a set of 3 experiments have been described in order to motivate the need for a method more robust, automatic and applicable to a wide range of grayscale 3D images. The experiments highlight the false positives problem that has limited every method so far. The most promising of the three methods was the revolving cylinders approach. However it was highly dependent on user settings, and even withstanding the slowdown of parameter optimization, it remains a challenge to validate the candidate spines.

These drawbacks prompted the development of an alternative method to detect the spines such that it favors full automation, less parameterization and reliable validation. We term this process Phase II which is the subject of chapter 5. In the next chapter we present the dendrite shaft tracing and enhancement algorithms (Phase I).

4. Neuronal Tracing and Enhancement

In this chapter we describe in some detail the dendrite tracing algorithm for Phase I of spine detection. The motivation and specifications desired for tracing were explained in chapter 3. The design of the tracing algorithm is based on the vessel tracing method of Tyrrel et al. (Tyrrel, di Tomaso et al. 2007). It is driven by three main concepts: (i) A super-ellipsoid based shape model for dendritic fragments, (ii) robust intensity estimation, and (iii) validation criteria. The algorithm is recursive - the shape model depends on the intensity estimates, and the intensity estimates are optimal when the shape model fits best. Therefore an iterative solution is devised such that an initial intensity is estimated and a super-ellipsoid is fit. The intensity estimates are updated using the pixels inside and outside the super-ellipsoid.

Although the model has proven to be generally reliable, it still suffers from tracing discontinuities and other errors. The following sections describe methods to correct such errors before using the tracing results for spine detection. We describe this tracing fidelity enhancement method in section 4.4.

4.1 The Dendrite Shape Model

The tubular dendritic structures of interest are modeled using super-ellipsoids. The tracing algorithm generates candidate tracing seeds from which the fitted model is grown iteratively in a direction that maximizes the fitting to the local image data. Each growth step estimates the next super-ellipsoidal parameters. The set of consecutive super-ellipsoids (tracing elements) comprises a neurite trace. This neurite is now the parent structure and the tracing elements are the children. A set of verification tests is performed to ensure certain structural criteria are met (such as minimum tube length and loop avoidance).

The following attributes are retained in each tracing element:

- Local foreground and background intensity estimates \hat{f} and \hat{b} ,
- super-ellipsoid model parameters:
 - a. centroid μ ,

- b. three axial directions $\mathbf{P}_{1,2,3}$ represented by quaternions \mathbf{q} ,
- c. three radial lengths (σ_z in the dendrite shaft (major axis) direction and σ_y & σ_x in the normal plane) and
- d. the super-ellipsoid east-west and north-south exponents e and n .

Whether or not a pixel belongs to the dendrite body is determined by its distance from the nearest super-ellipsoid model using the following distance equation(Gray 1997):

$$F = \left(\left| \frac{x - \mu_x}{\sigma_x} \right|^{\frac{2}{e}} + \left| \frac{y - \mu_y}{\sigma_y} \right|^{\frac{2}{e}} \right)^{\frac{e}{n}} + \left| \frac{z - \mu_z}{\sigma_z} \right|^{\frac{2}{n}}. \quad (4.1)$$

When $F < 1$, a pixel (x, y, z) is considered inside the super-ellipsoid. Our interest is to identify pixels closest to the ellipsoid surface in order to eliminate over-fitting of the neurite shaft estimates. This helps in reducing validation problems later when detecting spines (Phase II, chapter 5).

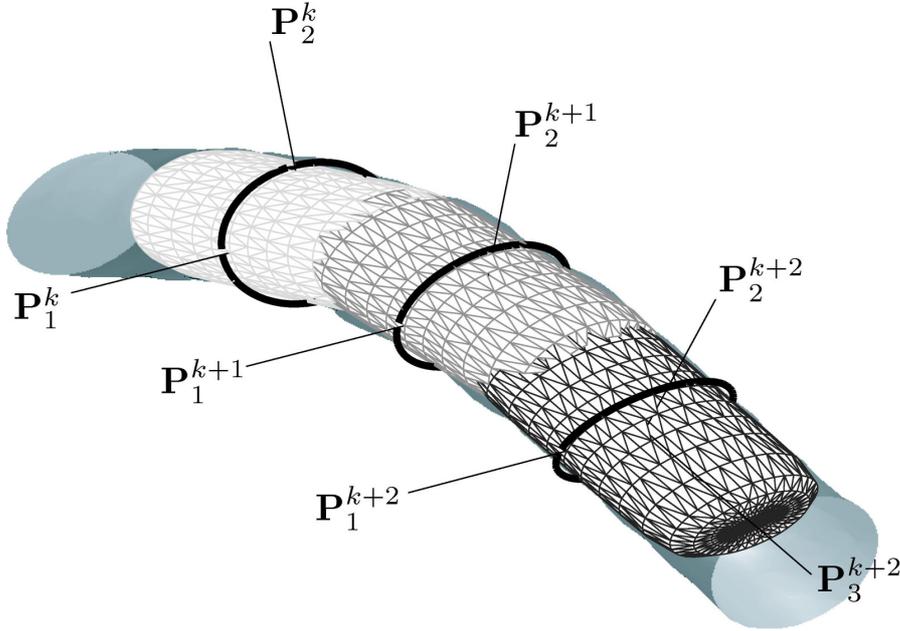


Figure 14 Phantom image illustrating super-ellipsoid fitting and tracing. Courtesy Dr. Alex Tyrrell.

The fitting algorithm assumes a homogenous grayscale intensity model, i.e. a homogenous foreground representing the fluorescence and a constant background. A maximum likelihood estimator is used to find the optimal super-ellipsoid model parameters. The corresponding log-likelihood equation is given by,

$$L(C; \hat{f}, \hat{b}) = \int_{R(C)} \log f(I(\mathbf{x}) - \hat{f}) d\mathbf{x} + \int_{\Omega/R(C)} \log f(I(\mathbf{x}) - \hat{b}) d\mathbf{x}, \quad (4.2)$$

where, given the current intensity estimates \hat{f} and \hat{b} , we need to search for the super-ellipsoid parameters $\beta = \{\mu, q_{1,2,3}, \sigma_{x,y,z}, e, n\}$ defining region $R(C)$ that maximize the function L . The log functions act as weighting terms that force the fitting towards the super-ellipsoid boundary. For the distribution function f , the Laplacian is chosen since it models best the robustness framework. This is explained further in the next section. The maximization process is formulated as a gradient ascent over the super-ellipsoid model parameters.

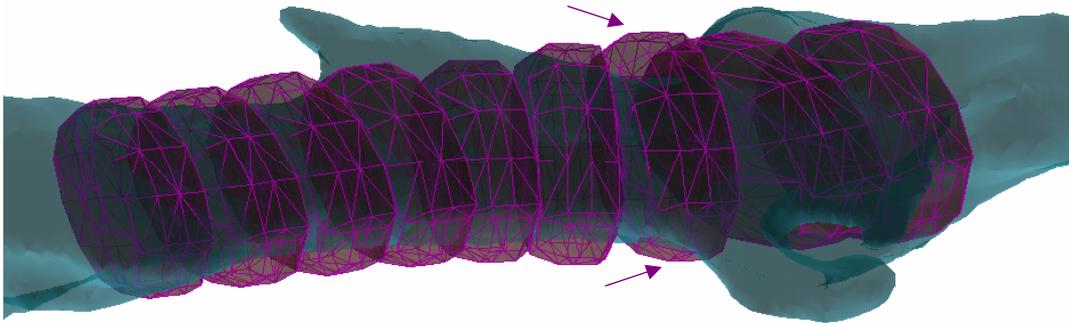


Figure 15 Model-fitting on a real image segment rendered in light green. A sequence of super-ellipsoid surface meshes is superimposed in purple. The super-ellipsoids are oriented to best fit the local direction. The major axes along the dendrite shaft centerline are truncated for easier visualization. Some over-fitting is noticeable near the dendrite surface (arrows).

4.2 Intensity Estimation

The foreground pixels are those within the fitted super-ellipsoids; the rest are background. Therefore, ideally, the best foreground estimate would be the mean intensity internal to the super-ellipsoid boundary. However, the boundary itself is based on an intensity estimate (Eq.4.2). Hence there is a need to iterate between the two processes, i.e., estimation of the super-ellipsoid fitting parameters and estimation of the foreground intensity. The initial background estimate is computed globally since it accounts for the majority of pixels, and is nearly constant throughout the image. Estimating the foreground is complicated by embedded noise with an as yet of unknown distribution. Mahadevan et al. demonstrated the inadequacy of matched filters for non-Gaussian noise in retinal images (Mahadevan, Narasimha-Iyer et al. 2004). Noting that the generalized hypothesis test suffers from great sensitivity to outliers when independent pixel intensity is assumed, they verified the efficacy of an alternative test due to (Huber 1965). The authors showed that the Huber test produces robust model parameter estimates. However, it is computationally expensive. Tyrrell et al. found that the Huber model can be simplified to a generalized hypothesis test using a Laplacian model for the noise distribution (Tyrrell, di Tomaso et al. 2007). This model offers a practical compromise between the Gaussian and Poisson distributions that are best applicable at high- and low-photon counts, respectively. The computational advantage of the Laplacian is compensated for by the slight degradation in the quality of the estimate since it relies on the ensemble of pixels within the detector rather than the individual intensities as in the Huber model. Since this tradeoff is reasonably worthwhile, we have adopted it in this work as well.

Because of the nature of the Laplacian distribution (also known as double exponential) which is “heavy”-tailed rather than a Gaussian, a possibly large number of outliers can appear within the model due to noise and background pixels near the surface. For a random variable x centered around the estimate μ with scale parameter λ , the Laplacian distribution is given by:

$$f(x | \mu, \lambda) = \frac{1}{2\lambda} e^{-|x-\mu|/\lambda}, \lambda > 0. \quad (4.3)$$

The maximum likelihood estimator of μ is the sample median which further justifies our choice. A proof of the estimate is given in the appendix.

Therefore, the foreground estimate is chosen to be the median intensity within the super-ellipsoid, keeping in mind the need to validate the model fit afterwards. This choice lends the foreground estimate its robustness against outlier pixels around the dendrite surface (noise, background clutter and imaging artifacts).

4.3 Model Validation

In order to ensure a reliably-automated tracing procedure, an additional step is required to verify that the model being fit is suitable for a tubular fluorescent structure, such as a neurite. A generalized hypothesis test is performed over the fitted region such that

$$\begin{aligned} H_0 : I(\mathbf{x}) &= \hat{f} + N ; \\ H_1 : I(\mathbf{x}) &= \hat{b} + (\hat{f} - \hat{b}) \times 1_{[F(\mathbf{x};\beta) < 1]} + N , \end{aligned} \quad (4.4)$$

where H_0 denotes the hypothesis that the data is part of the background and H_1 is the alternate hypothesis indicating that the data within the super-ellipsoid F (Eq.4.1) with parameter set β is part of the foreground. The indicator function $1_{[F(\mathbf{x};\beta) < 1]}$ evaluates to 1 for image data within the model, and to 0 otherwise.

Since no prior information about the model parameters or intensities is assumed, the optimal detector is a generalized likelihood ratio test:

$$\frac{\max_{\theta_1} p(I(\mathbf{x}) | H_1)}{\max_{\theta_0} p(I(\mathbf{x}) | H_0)} \underset{H_0}{\overset{H_1}{>}} \tau . \quad (4.5)$$

A Laplacian distribution is used as explained above,

$$p(I(\mathbf{x}) | H_0) = \prod_{k=1}^{|\Omega|} \frac{1}{2\lambda} e^{-|I(\mathbf{x}_k) - \hat{b}|/\lambda} . \quad (4.6)$$

The log-likelihood ratio test for the fitted region $R(C)$ is therefore

$$\Lambda = \sum_{\mathbf{x}_k \in R(C)} |I(\mathbf{x}_k) - \hat{b}| - |I(\mathbf{x}_k) - \hat{f}| \stackrel{H_0}{<} \tau \stackrel{H_1}{>} \quad (4.7)$$

The optimum choice of the threshold τ is determined empirically. However, the model is still insensitive to small perturbations around the optimal value. Typically we set $\tau = 1$.



Figure 16 Segment from spine1.tif (data courtesy MBF-Bioscience and Dr. Fulvia Gheorghita, DBCM, UNIL, Lausanne). A sample region $R(C) = \{\mathbf{x} \mid F(\mathbf{x}; \beta) < 1\}$ is framed in red to demonstrate hypothesis test described above. The closer the ensemble of pixel grayscale values is to the foreground than to the background, the more optimal is the fit.

4.4 Tracing Fidelity Enhancement

Our experiments have proven the capability of the above model to trace 3D neuronal dendrite images from diverse biological experiments. If we quantify the outcome based on detected/traced fluorescence, the tracing results are reliable enough to account for most, if not all, foreground of interest. However, for our purposes the tracing quality should be based on its effectiveness for achieving reliable spine segmentation.

Reevaluating the traces uncovers errors that eventually are carried over to the next phase of spine detection.

4.4.1 *Tracing Errors and Their Effect on Spine Segmentation*

Fragmented dendrite traces are usually caused by imaging noise, faded contrast between successive image slices, or absence of fluorescence between axonal varicosities. In these cases, the incremental tracing algorithm using the super ellipsoid model, together with the automatic validation test, fail to locate enough foreground voxels connecting the previous fit. This results in a premature stopping of the tracing procedure for a given dendrite, and leads to a gap. Identifying such false gaps in traces is important to complete the search for spines in the surrounding neighborhood. Failing to do so correctly will increase the spine miss rate.

On the other hand, since the tracing method is driven by the fluorescence intensity, a spine with sufficient intensity and size so as to favor steering the super-ellipsoid towards it can cause the tracing process to overlook the remaining dendrite and trace the spine itself. Identifying a spine as part of the dendrite shaft will cause that spine to be missed when looking for spines. Not only will this error fragment the dendrite, it will possibly lead to more spine misses in-between the fragments.

Another source of error is the tracing of detached spines, filopodia, long spines, or other sub-cellular debris in the image field that are long enough to fit a sequence of super-ellipsoid models independently, i.e., these structures can be traced by the algorithm. Obviously classifying these as neurites would falsely eliminate them from being detected as spines/filopodia, leading to miss errors. A similar problem is falsely tracing regions that are too noisy or uninteresting. This kind of error potentially contributes to false spine detections.

We can summarize the above observations by classifying the tracing errors into two major categories (Table 3) according to their effect on tracing, especially given that we do not have any spine information at this time. The next section details our method for identifying the errors and fixing them.

Table 3 Summary of common types of tracing errors, their effects on spine detection, and potential remedies.

Error Source	Error Category	Tracing Result	Spine Detection Effect	Remedy
Attached spine	Type A	Fragmented dendrite	Spine Miss	Un-trace spine; reattach dendrite fragments
Intensity Discontinuity	Type A	Fragmented dendrite	Potential spine miss or false positive	Reattach dendrite fragments
Tracing a long or detached spine, or filopodium	Type B	Invalid trace	Spine miss	Delete trace
Dense noise clutter or uninteresting region	Type B	Invalid trace	Potential false positives	Delete trace

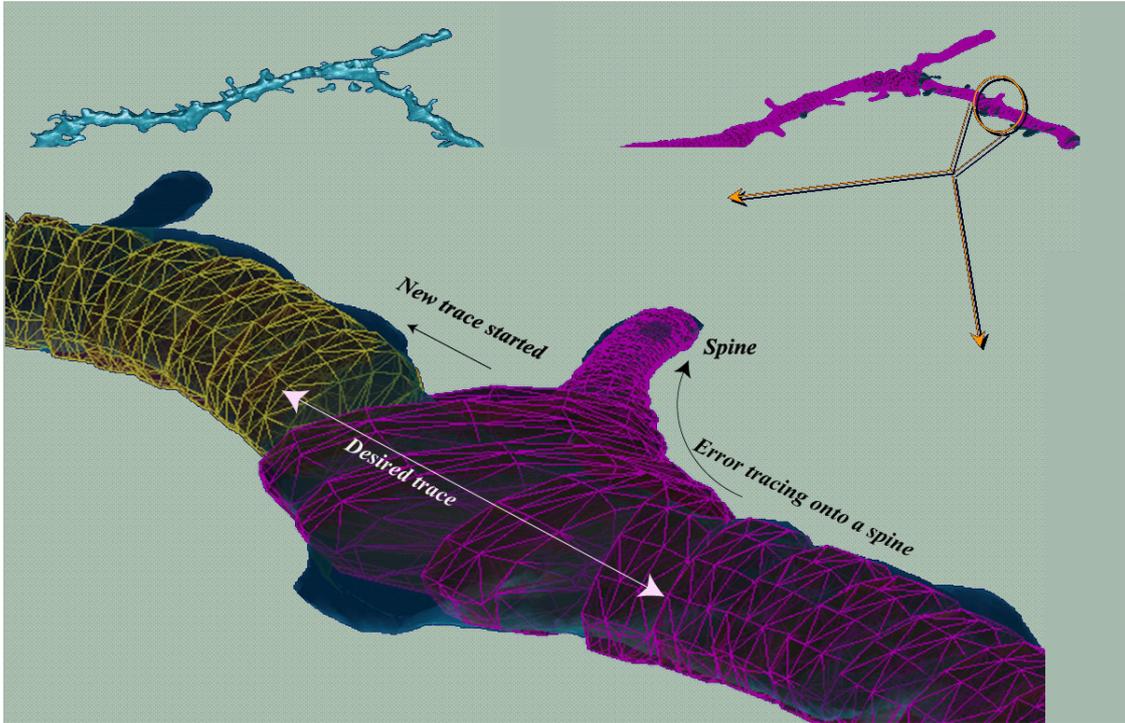


Figure 17 Example of a Type A tracing error. Above: left: 3D rendering of spine1.tif, right: surface mesh of tracing models superimposed on rendered image. Below: Zooming in on selected region shows the purple trace diverted onto an attached spine and a new trace (yellow) started independently. Left as is, the trace will cause the spine to be missed.

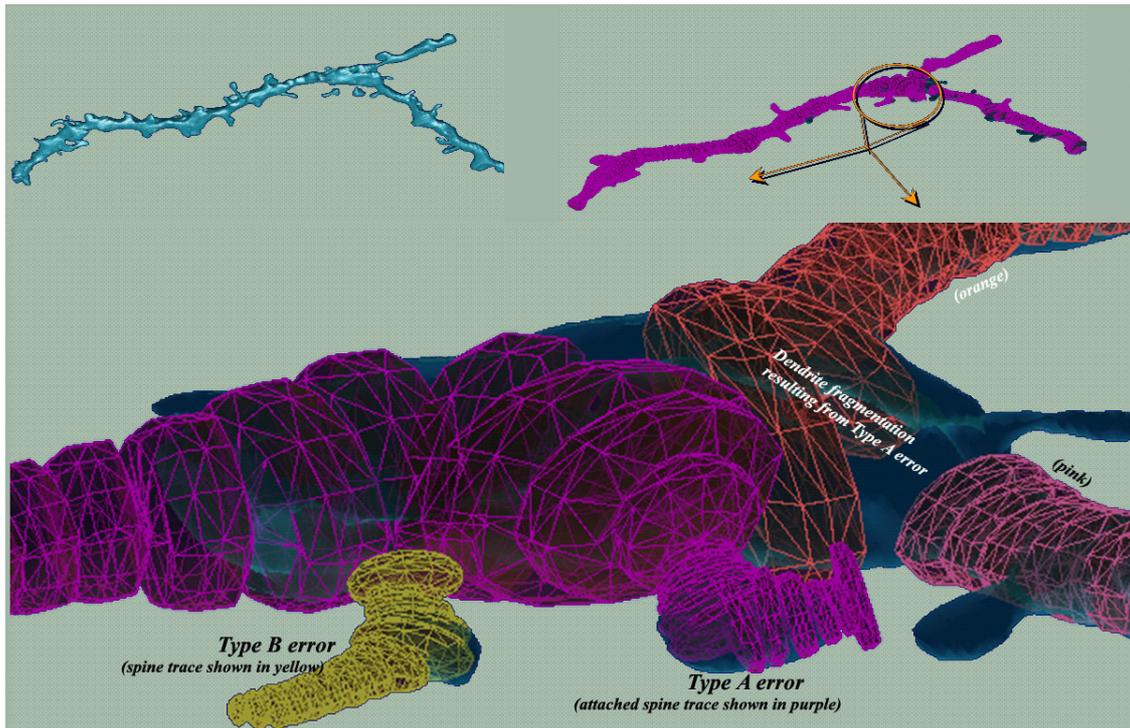


Figure 18 Types A and B Tracing errors at different segments with the same image as Figure 17. The purple trace avoids the first spine (yellow) but is lead onto another spine where it erroneously terminates. The orange and pink traces start independently and terminate when intersecting with the purple trace.

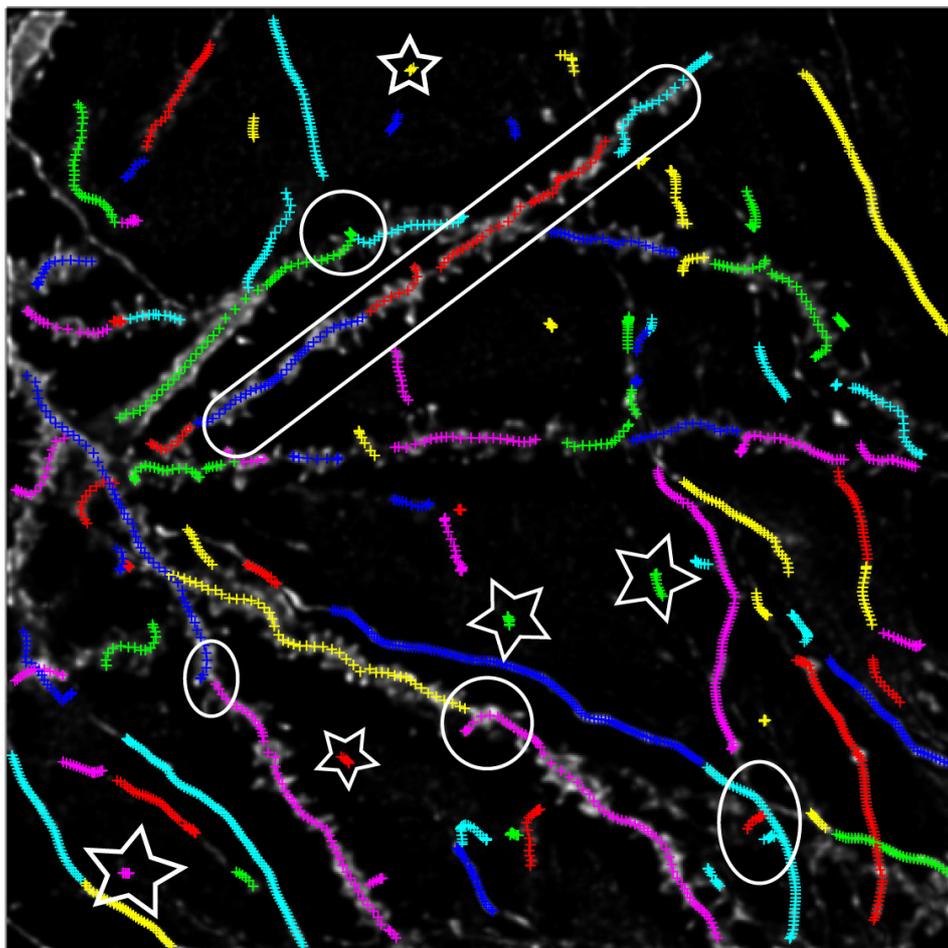


Figure 19 Z-Projection of image 6297,312MLE Courtesy Prof. S. Potter (Georgia Tech). Traces are superimposed in color. The elongated white boundary identifies Type A error caused by intensity discontinuity making the dendrite chopped into 4 pieces. Some spine-induced Type A errors are highlighted by the ellipsoidal boundaries. Some Type B errors are marked by a star.

4.4.2 *Fixing Tracing Errors*

In this section, we describe an automated procedure that enhances tracing fidelity by searching for tracing errors, identifying their type and correcting them. This approach is rule-based relying on dendrite continuity and on the primary/secondary relation between dendrite and spine.

We describe the Phase I results as a set Ω of k traces φ_i ; each trace φ_i consists of a number n_i of directional center points:

$$\Omega = \{\varphi_1, \varphi_2, \dots, \varphi_k\}; \quad (4.8)$$

$$\varphi_i = \{v_{i_1}, v_{i_2}, \dots, v_{i_{n_i}}\}; \quad (4.9)$$

$$v_{i_s} = \{x, y, z, q_{1,2,3}, \sigma_{x,y,z}\}_{i_s}. \quad (4.10)$$

The elements of each set v_{i_s} are extracted from the super-ellipsoid parameters as detailed in Section 4.1. As described in Table 3, the type of correction for a particular trace line is dictated by the particular type of error that is detected. The corrective remedy would either be reattaching the broken dendrite, possibly after un-tracing a spine in between, or deleting the trace entirely. The order of these detections is important: in order to mark a small fragment for deletion, we must first attempt to reconnect it to a larger fragment if possible. Therefore, all Type A error fixes must be attempted before Type B errors.

4.4.2.1 Detecting a breakage (Type A)

With no prior information, we assume that every trace is a potential dendrite fragment. We first look for neighboring *ending parts* E_i (discussed below) from separate traces. With $\|\cdot\|$ denoting Euclidean distance, we define the distance between two trace endings as:

$$d(v_{i_s}, v_{j_t}) = \|(x, y, z)_{i_s} - (x, y, z)_{j_t}\|; \quad (4.11)$$

$$d_{E_i E_j} = d(E_i - E_j) = \min_{v_{i_s} \in E_i, v_{j_t} \in E_j} d(v_{i_s} - v_{j_t}). \quad (4.12)$$

Two trace endings E_i and E_j are considered to be neighbors if

$$d_{E_i E_j} \leq \delta_d, \quad (4.13)$$

where δ_d is a global distance threshold. To formulate the set E_i requires some discussion. At first glance, it would seem intuitive to define an ending E_i of trace φ_i as the set

containing v_{i_1} and $v_{i_{n_i}}$ (the extreme tracing elements). Consequently, two traces φ_i and φ_j are neighbors if v_{i_1} or $v_{i_{n_i}}$ is close in Euclidean distance to v_{j_1} or $v_{j_{n_j}}$. This definition works well when only the intensity variation within the dendrite has caused the breakage and neither resultant trace fragment has terminated onto a spine. Here, the fragments are cut almost abruptly without a significant change in directions, as highlighted by rectangular boxes in Figure 20.

However, as explained previously, when the dendrite fitting model favors a spine, the trace terminates at the spine head thereby creating an unaligned fragmentation of the dendrite (see the circular highlights in Figure 20) because the spine protrudes from the dendrite shaft at an angle. In these cases $E_i = \{v_{i_1}, v_{i_{n_i}}\}$ is inadequate for establishing the neighborhood. The erroneous terminal point of the trace could easily have followed the spine to a distance $> \delta_d$ from the would-be neighboring fragment. Therefore, in order to establish the neighborhood, the spine should be excluded from E_i . However, at this stage of the algorithm, the spines have not yet been segmented. This is potentially a ‘‘catch-22’’ situation.

This problem can be alleviated by defining the trace ending parts as the set of endpoints covering a fraction $\rho < 1$ of the dendrite shaft length on either extremity:

$$E_i = \{v_{i_1}, \dots, v_{i_{\lfloor \rho n_i \rfloor}}, v_{i_{\lfloor (1-\rho)n_i \rfloor}}, \dots, v_{i_{n_i}}\}. \quad (4.14)$$

In this case the trace points potentially representing a spine would still be included in the set, but they are not the only ones. At the very least, we can guarantee that there is a better candidate ending point that fits Eq.4.12 than the trace point at the spine head. Our empirical studies have shown that 0.2 is a reliable value for the parameter ρ . Larger values do not degrade correctness but would cost redundant computation. In order to estimate δ_d , we rely on the intra-tracing distances $d(v_{i_k}, v_{i_{k+1}})$, the gap between any two tracing elements. These distances vary from one dendrite to another based on the dendrite tracing model estimation. When a dendrite has high contrast it tends to create longer tracing elements. In order to cover all possibilities, we take twice the maximum of all distances:

$$\delta_d = 2 \times \max_{i,k} d(v_{i_k}, v_{i_{k+1}}).$$

The factor 2 was determined empirically.

The above approach for establishing neighborhoods among traces works well in general but still suffers from a rare deficiency. A situation can arise when two *really* separate neurites are in close proximity of one another but do not naturally form a single dendrite. Connecting their traces unnecessarily would potentially increase false positives when detecting spines. To deal with this situation we add a smoothness constraint that prevents abrupt orientation changes at the connection site. This calls for exploiting the dendrite model directions q_{i_3} and examining the alignment of the neighboring fragments. If their relative directions are within a tolerable deviation δ_θ from a straight angle (180°), then they should be reattached:

$$\theta_{ij} = \angle q_{i_3}, q_{j_3} = \cos^{-1}(q_{i_3} \cdot q'_{j_3}) \leq \delta_\theta. \quad (4.15)$$

We found $\delta_\theta = 30^\circ$ empirically as a reliable parameter setting.

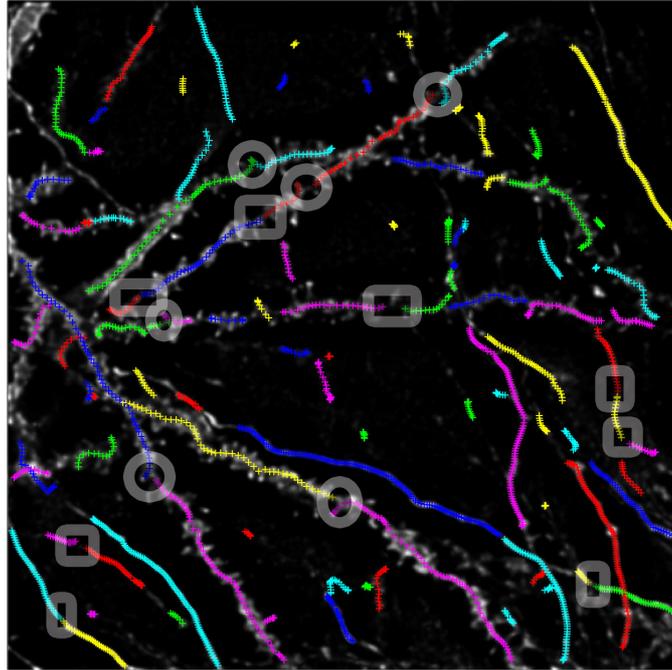


Figure 20 Type A tracing errors. Rectangular highlights show almost straight disconnections due to intensity variations. The trace endings are the same as their extremities; trace neighborhood is determined by those endpoints. Circular highlights show redirected trace extremities onto a spine. These trace endpoints are not suitable for establishing trace ending and neighborhood. We need to un-trace backwards until proper alignment with the opposite fragment is achieved leading to correct reattachment.

4.4.2.2 Tracing Model Interpolation

Gaps separating the tracing segments of each dendrite are not uniform. In particular, gaps that have been identified as tracing errors and reconnected may be too wide. Thus the entire dendrite trace requires model-regularization. This is done by interpolating between the adjacent super-ellipsoid elements with respect to all their relevant fields; i.e. position, orientation, radii and intensity estimation. As described later in chapter 5, the interpolated elements are nearly as important as the original traces with respect to the spine detector dimensions and orientation.

4.4.2.3 Fixing Type B Errors

After fixing Type A errors, we are confident that there are no more connectable traces to attach. The remaining errors are the short traces caused by either tracing a detached spine or tracing an insignificant or noisy region. A possible remedy for these cases is to eliminate them from the trace set Ω altogether. Various approaches can be used to classify a short trace as a Type A error. One way is to cluster the traces based on length and proximity to another dendrite shaft excluding the extremities. A small trace that is close to a dendrite surface is most likely a spine. Small and distant traces can be selectively negligible because any possibly detectable surroundings are less likely verifiable. As a proximity threshold, we accept dendrite lengths exceeding 2 standard deviations from the mean.

4.5 Chapter Summary

In this chapter we described the tracing approach used for the spine detection to be described in Chapter 5. It is based on a recursion that alternately fits a super-ellipsoid and estimates the local foreground and background intensity levels. We then described our model for automatically correct the tracing errors in order to increase the tracing accuracy which is essential for our ultimate purpose.

5. Spine Ring Detection

As mentioned in chapter 3, we view the spine as a structure secondary to the dendrite. Thus our proposed spine detection approach is subsequent to the model-based neuronal tracing Tyrrell(Tyrrell, di Tomaso et al. 2007) which, unlike skeletonization-based methods, is robust against image intensity variability, image artifacts, dendrite branch points and crossover. The revolving cylinder detector (section 3.4.3) proved to be a promising model for initial spine detection, albeit with the need for computationally-expensive detector optimization. This motivated us to propose a simpler, faster and more reliable detector that avoids the major drawbacks of its predecessor such as quantization errors of the elevation and spinning angles, sensitivity to the cylinder dimensions; i.e., radius and height, as well as its position above the dendrite surface. Furthermore, those parameters depend on a single spine, which makes fitting a cylinder not only computationally costly, but also shape biased. For example, a spine with a curved central axis would not match well with a cylinder.

Our novel detector alleviates these problems by parallelizing detection without initial presumptions for a shape model. The proposed algorithm creates a series of hollow annuli surrounding and along the dendrite surface. Each annulus serves as a local detector and is part of a dynamic queue that is *binarized*¹ to create spine candidates from the connected components. The central axes are extracted to merge spine fragments and link the detached spine back to its parent dendrite. These steps are summarized as follows:

- Initial Ring detector: adjacent hollow annuli surrounding the dendrite shaft are created. The interpolated intensities are thresholded resulting in foreground patches inside the annuli (Figure 21 –A).
- Spine Candidate Designation: adjacent annuli are merged based on detector hit continuity. Extracted connected components are now called spine candidates (Figure 21 –B).

¹ To *Binarize* an image is to present it in a binary form replacing all pixel intensities above a pre-determined threshold with 1 and the rest with 0.

- Fragment Merging: central axes from spine candidates are extracted. Based on inter-axial distances merging spine candidates is decided (Figure 21 –C).
- Feature Extraction: (length, volume, tip, root, head, neck¹) (Figure 21 –D).

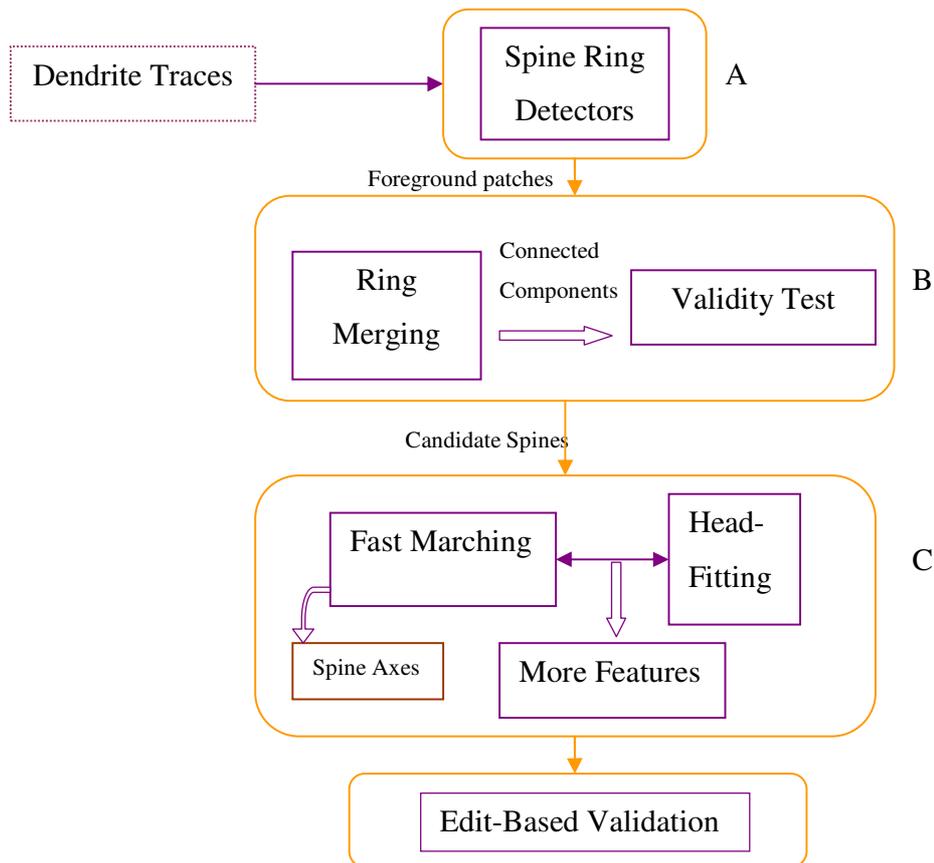


Figure 21 Spine Detection flowchart. Dendrite traces are given as input to the process which starts with creating surrounding ring detectors (A). Thresholding is followed with connected component labeling and a validity test to produce spine candidates (B). For each spine candidate, a wavefront is propagated using the fast marching technique and the spine axis is extracted (C). Neighboring axes are identified for merging spine fragments and finally validating the spine.

In this chapter we describe these modules in more detail.

¹ Future work.

5.1 Phase II: Spine Ring Detector

The detector is a 3-dimensional annulus that is constructed around a local region of the dendrite. It is a *rectangular toroid* in the sense that the sagittal cross section is rectangular. The inner hollowness is dictated by the underlying assumption of the dendritic shape. In this work we tested two such ring detectors: one with a circular inner cross section and the other with a super-ellipsoidal one. This makes our detector generalizable to suit different dendrite tracing approaches.

5.1.1 Cylindrical Ring Geometry

The cylindrical annulus T is characterized by an inner radius r_i , an outer radius r_o , and a thickness h (Figure 23). A point $p(x, y, z)$ belongs to a Toroid T_o centered at the origin according to the following equation:

$$p(x, y, z) \in T_o \Leftrightarrow r_i^2 \leq x^2 + y^2 \leq r_o^2, -h/2 \leq z \leq h/2. \quad (5.1)$$

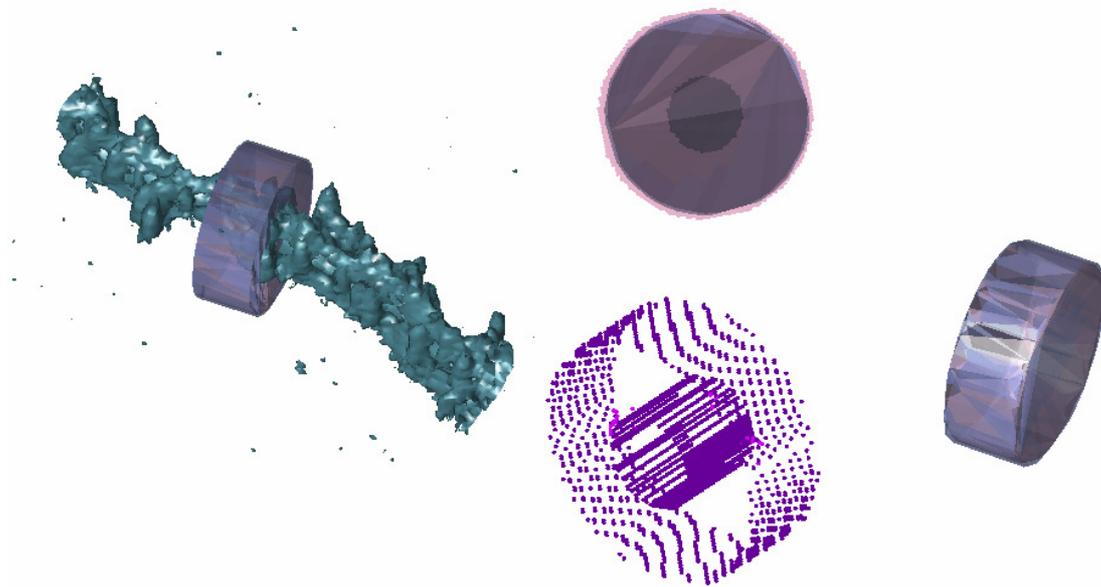


Figure 22 Cylindrical ring. Left: ring surrounding dendritic region. Top center: circular hollowness. Bottom center: inner and outer circumferences.

We utilize the direction and size of the tracing model to create the most optimal detector at each detection site. In order for the annulus to avoid missing stubby spines as well as falsely detecting part of the dendrite shaft, its inner circle must touch the dendrite surface. In our tracing model there are two super-ellipsoid lateral radii σ_x and σ_y . We choose

$$r_i = \max(\sigma_x + \sigma_y). \quad (5.2)$$

Because spine lengths vary, the requirement for r_o is to maximize the full capture of dendrite surface protrusions. However, an excessive radius may possibly intersect the annulus with a neighboring neurite or its spines. We dealt with this tradeoff using prior biological information that the spine length is typically between 1 and 3 dendrite diameters (Table 1). In order to cover all types of images without the need for prior information about magnification or resolution, we chose

$$r_o = 5r_i. \quad (5.3)$$

The annulus thickness h is designed to cover the entire length of the tracing element in the dendrite shaft direction ($h = 2\sigma_z$). Sub-pixel accuracy is achieved by varying the resolution within the annulus. This can be user-specified if desired.

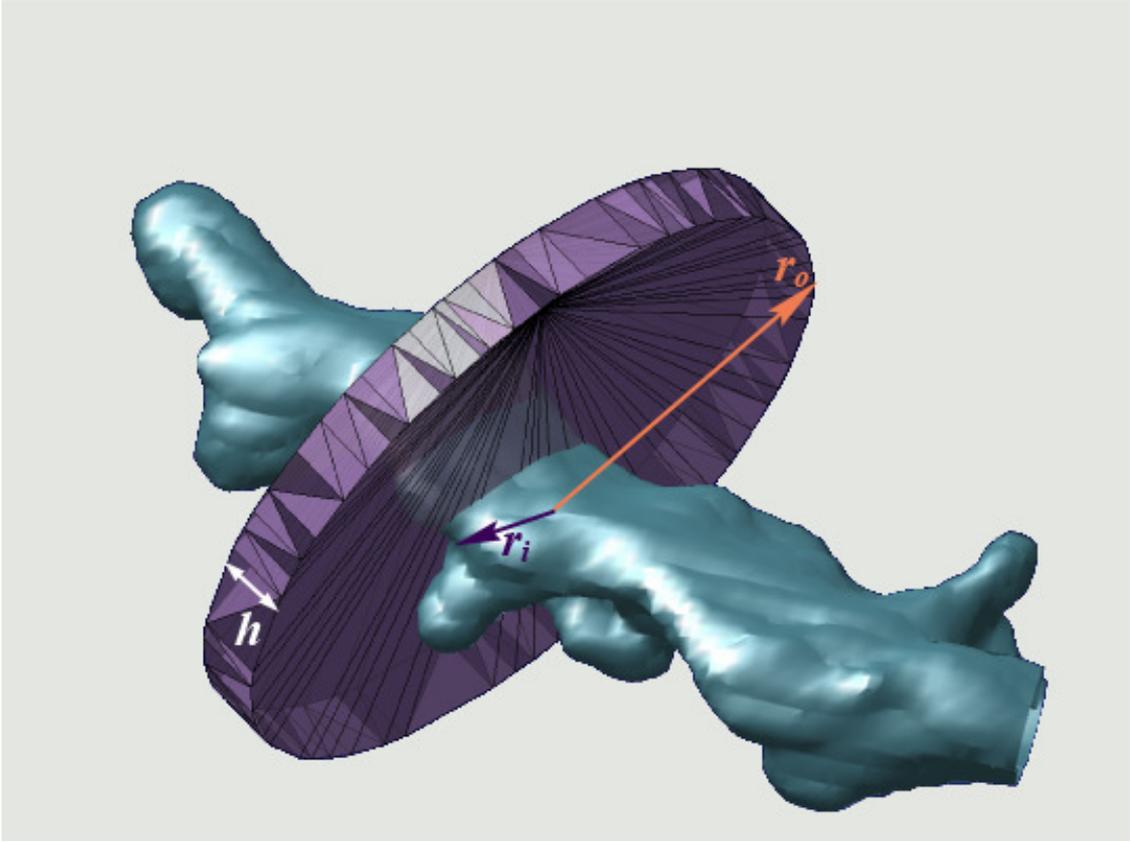


Figure 23 Dendrite segment rendering with ring detector dimensions. These dimensions are determined by the dendrite maximum lateral diameter as estimated in the tracing model.

5.1.2 Super-Ellipsoidal Ring Geometry

This ring differs from the above cylindrical ring only in the inner hollowness. Because the underlying tracing algorithm estimates the dendrite as a sequence of super-ellipsoidal segments, the inner ring dimensions are specifically the same as those of the dendritic segments, bearing a slight magnification in order to ensure exclusion of dendritic boundary from detection. The details were discussed in Section 4.1; here we reproduce the fundamental super-ellipsoid equation:

$$F = \left(\left| \frac{x - \mu_x}{\sigma_x} \right|^{\frac{2}{\epsilon}} + \left| \frac{y - \mu_y}{\sigma_y} \right|^{\frac{2}{\epsilon}} \right)^{\frac{\epsilon}{n}} + \left| \frac{z - \mu_z}{\sigma_z} \right|^{\frac{2}{n}}, \quad (5.4)$$

where now $F > 1$ is desired for the region external to the dendrite.

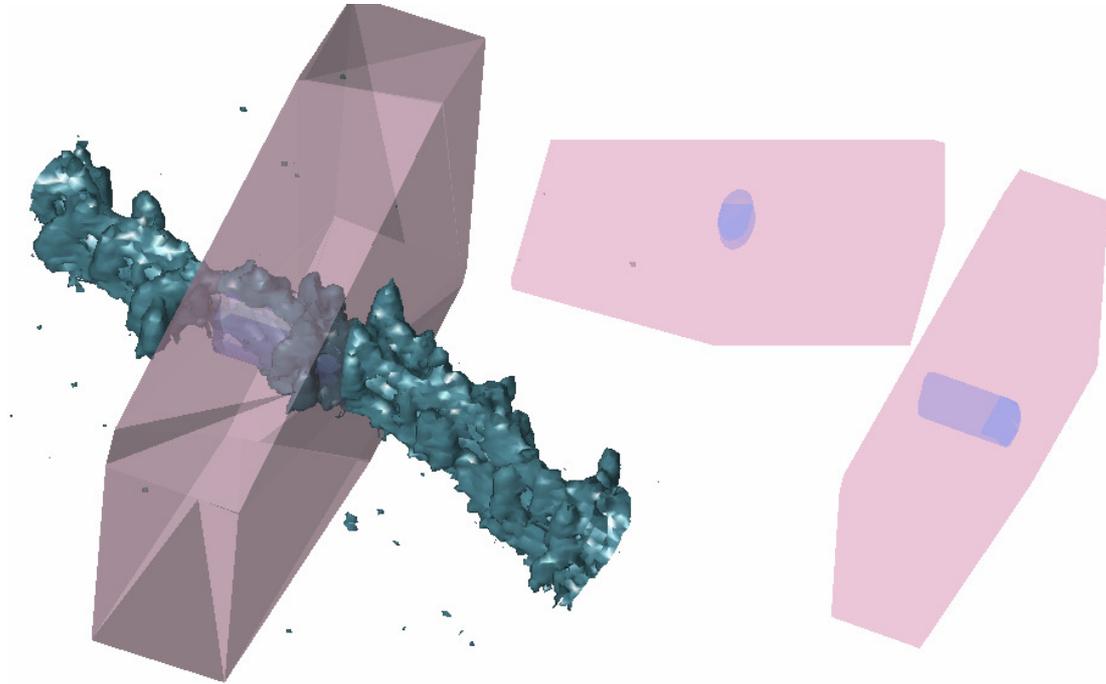


Figure 24 Super-ellipsoidal ring. Left: ring surrounding rendered region. Middle: (almost) frontal view showing ellipsoidal rather than cylindrical hollowness. Right: Sagittal view showing rectangular cross section.

5.1.3 Orientation and Detection

We orient the detector around the dendrite shaft (Figure 23, Figure 25) utilizing the quaternion rotations described in section 3.4.3.1. Hence, a detector T is aligned with the major principal axis of an ellipsoid $F(\mu_{xyz}, \sigma_{xyz}, q_c)$ by a rotation and translation:

$$T' = R(q_c) \times T + \mu. \quad (5.5)$$

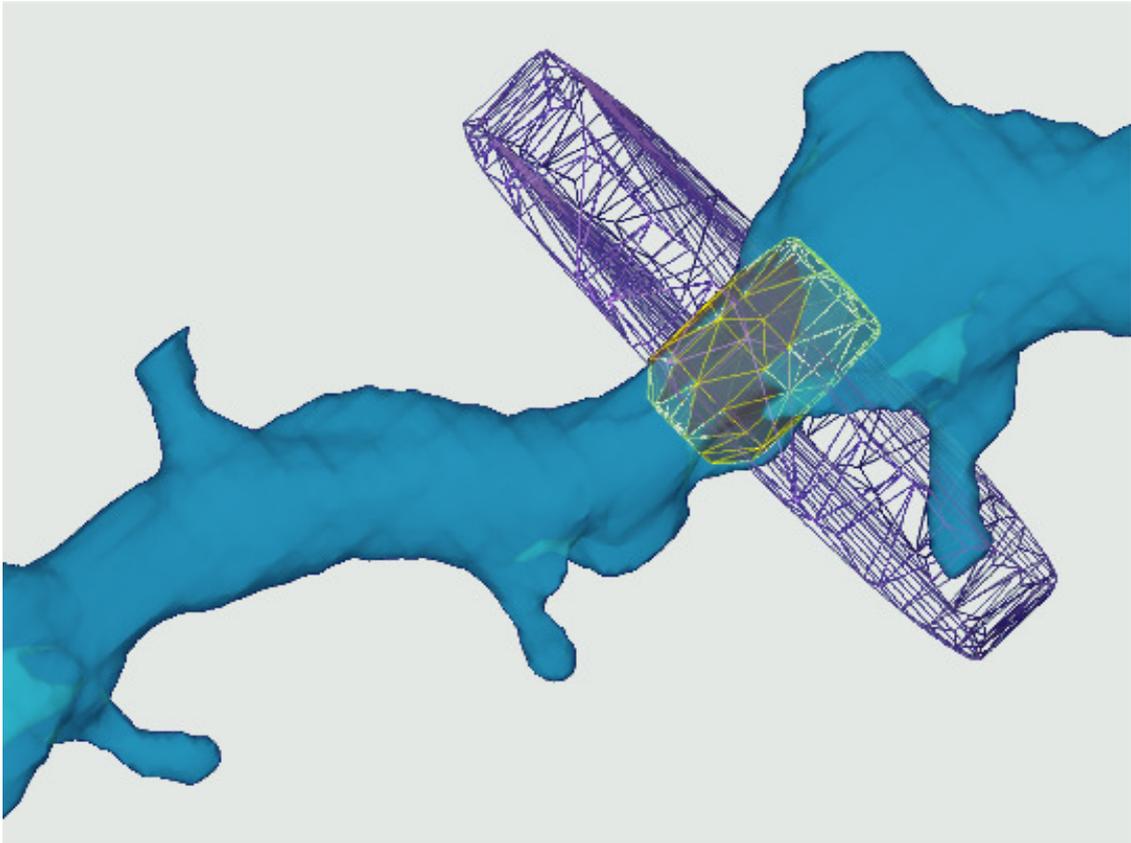


Figure 25 3-D rendering of dendrite segment with spine ring detector outline (purple) surrounding the super-ellipsoid tracing element (yellow).

The internal space of the detector is populated with a pixel mesh whose intensities are interpolated from the image. The foreground and background intensity estimates of the dendrite are used as a basis to threshold the region. Since the detector is moved in small increments, a spine would be partially detected within a single detector. Therefore, adjacent detectors with foreground continuity in between need to be merged.

Next, connected components are labeled and filtered through a correlator detector as a preliminary validity test. Components that pass the test are called spine candidates.

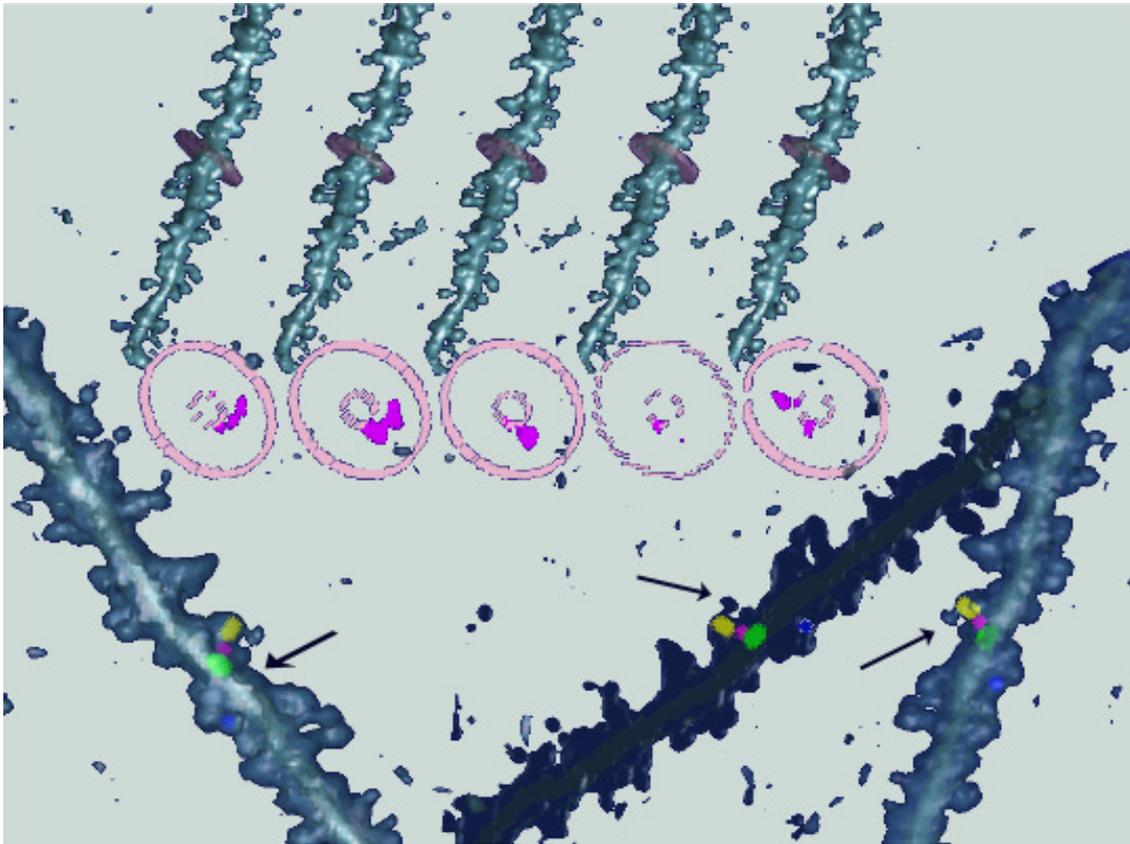


Figure 26 Spine ring in motion. The top part shows a sequence of ring detectors in grey moving up the dendritic shaft (rendered in green). The detected voxels inside each ring are shown as projected onto its central axis. The bottom part highlights the 4 candidate spines that were detected inside the rings and separated by connected components (arrows).

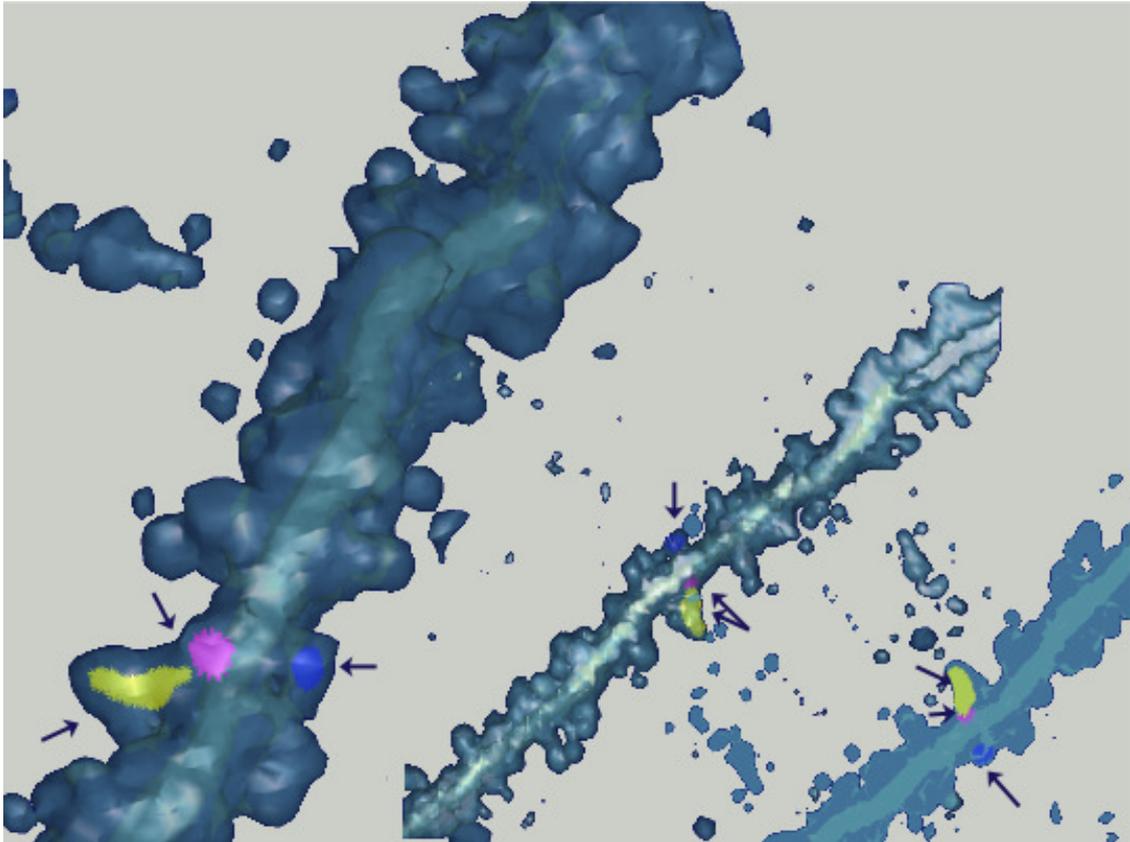


Figure 27 Three spine candidates highlighted in yellow, pink and blue over dendrite rendering (green); viewed from different angles (arrows).

5.2 Spine Fragments or Candidates?

The individual structures segmented using the spine ring detector are a mixture of complete spines, spine fragments, noise, and, possibly, parts of neighboring dendrites. This mixture underscores the challenge embedded in correct spine segmentation. For example, by definition, a *stubby* spine body is mostly close to the dendritic surface and can be confused with image blur. Moreover, depending on the biomarker staining and microscopy conditions, some spines – mostly of the *mushroom* and *branched* types – appear broken into 2 or more fragments and often totally detached from the dendrite surface. Since our proposed spine ring detector relies eventually on connected component labeling, a fragment is counted as a candidate spine. A remedial process is therefore necessary to merge fragments in order to correct the spine count and reduce false

positives. Moreover, for detached spines, a “reattachment path” back to the dendrite surface is required for further spine analysis.

We evaluate two fragment merging criteria. The first is described in section 5.3 and is based on volume intersection. We describe the geometrical background and the algorithms we used for the test. The second is described in section 6.3 and is based on the minimal geodesic path to the dendrite surface. We use fast marching for propagating a wavefront on the original grayscale image. The theoretical background and implementation are presented in the first part of chapter 6.

5.3 Intersecting Spine Candidates

As we have seen earlier, the proposed spine ring detector identifies a region of connected voxels as a spine candidate, or fragment, using geometric transformations and sub-voxel interpolation. Euclidean proximity of two such candidates is but one criterion for deciding whether they are fragments of the same spine. In the absence of a shape model the volume enclosed by the fragment voxels can be approximated using their convex hull. Thus proximity between two fragments can be estimated by geometric queries between one fragment and the convex hull of a neighboring one. In particular we are interested in detecting overlap or in measuring minimal separation between a point set and a convex polyhedron. The remaining part of this section describes the mathematical background and the algorithms used.

5.3.1 Convex Hull Computation

In general, given a set of points $S \subset \mathbb{R}^d$, the convex hull $P(S)$ is the minimal convex set containing S (Goodman and O'Rourke 2004). Each facet bounding the convex hull has d vertices belonging to the same (hyper)plane. In 1D the convex hull is a line segment bounded by two point-facets. In 2D it is a polygon bounded by line-facets connecting 2 vertices each. In 3D it is an intersection of plane-facets defined by 3 or more vertices each and bounded by line segments. Below we will describe the 3D formulation that pertains to the problem of interest, namely, point location (also known as membership) with respect to the 3D polyhedron (3-polytope).

A plane can be specified by the so- called *Hessian normal form*:

$$a^t \cdot x = b ; \tag{5.6}$$

where a is the vector normal, x is any point on the plane and b is the plane shift from the origin. If $b < 0$, the origin is located at the same side from the plane as its normal a . Figure 28 demonstrates this notation.

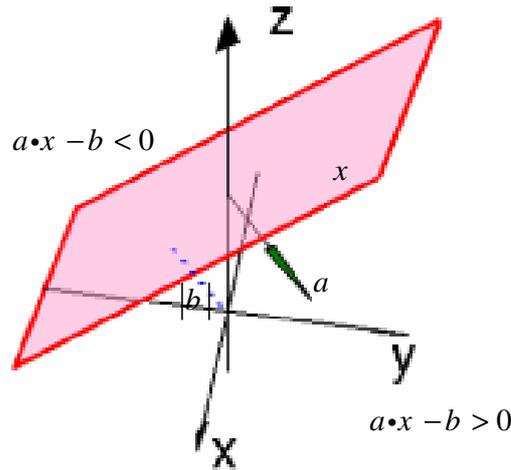


Figure 28 Demonstration of plane equation in Hessian normal form: $a \cdot x = b$; where a and x are respectively the normal vector and any point on the plane. The distance to the origin is $|b|$. The plane divides the space into two half-spaces. The halfspace in the same side as the normal vector satisfies the equation $a \cdot x - b > 0$.

Thus for a 3D polyhedron P , each of its facets divides the entire space into two regions, only one of which contains P . Thus P is the intersection of all half-spaces that contain it (which is an alternative definition of a convex hull). Consequently, a point x is in the interior of P if it belongs to the same half-spaces as P . Assuming P has n facets and the facet normals a_i make up the columns of matrix $A_{3 \times n}$, and $\beta_{n \times 1}$ constitutes all facet shifts b_i , then x satisfies

$$A^t x - \beta > 0. \tag{5.7}$$

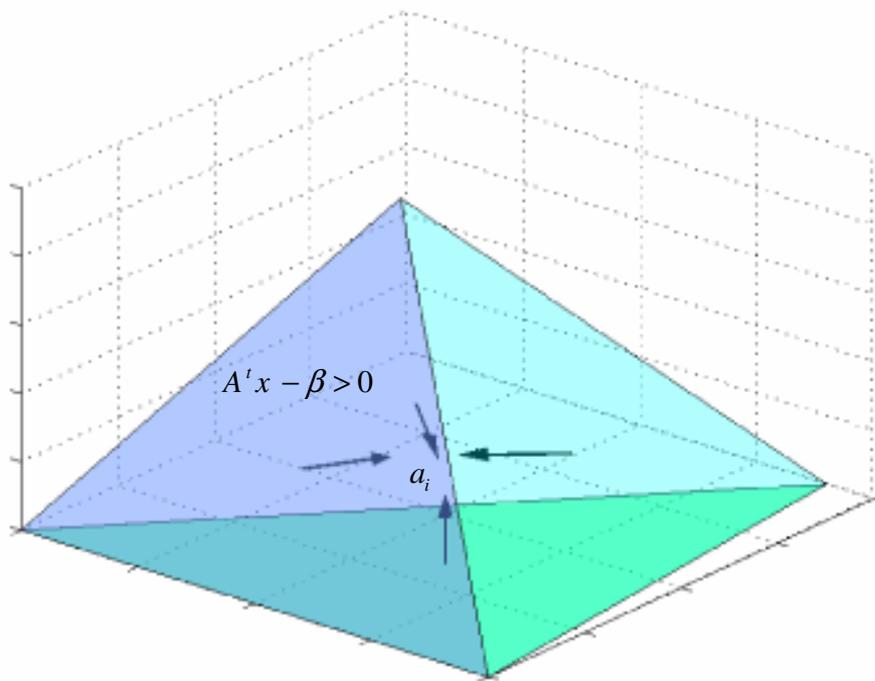


Figure 29 Demonstration of polyhedron with facet normals pointing inside. A point x inside the polyhedron satisfies $A^T x - \beta > 0$ where the columns of $A_{3 \times n}$ are the facet normals a_i and the facet shifts from the origin are the elements of $\beta_{n \times 1}$.

An efficient approach is to utilize the sign of a dot product as an inside/outside indicator. The algorithm is described in the following pseudo code.

If their convex hulls P_1 and P_2 , and check whether any vertex x of P_2 lies inside P_1 .

Several convex hull computation algorithms have been developed with the primary goal of reducing computational cost especially in higher dimensional spaces. Avis et al. (Avis, Bremner et al. 1997) compare several approaches on high dimensional data. Of those the Quickhull algorithm(Barber, Dobkin et al. 1996) was shown to be fast and efficient especially for low-dimensional space, as is the case with our application.

Quickhull is an *incremental* computation method; i.e., it starts with a $d + 1$ subset of points to create an initial convex hull. The other points are selected incrementally and tested for being inside or outside the convex hull. If a point x is inside all of the current facets; i.e., inside the convex hull, it is discarded. Otherwise, the convex hull is “grown” by including x as a vertex and eliminating the facet it faces from outside the hull. Incremental algorithms differ in the order of selecting the new points. The *Quickhull* algorithm chooses the farthest point to the outside of each facet in the current iteration as a candidate new point.

5.3.2 Testing Spine Candidate Intersection

Each detected spine candidate is saved as a set of points in 3D space. The question we try to answer is, given two sets of points, S_1 and S_2 , whether the volumes they constitute overlap. If they don’t overlap, we need to measure their Euclidean separation.

Our approach is to calculate the convex hull P_1 of S_1 and query the point membership of the elements of S_2 with respect to P_1 . In order to optimize the computation, we only test neighboring spine candidates. This query is based on the convex hull equation described above (Eq. 5.6). One implementation utilizes the Quickhull output (matlab `qhull` function) and the sign of dot products between the normal and the vector joining a facet vertex to the test point. The interested reader may refer to Appendix B for implementation details and the pseudo code.

A by-product of the calculation gives the distance between the test point and the nearest facet on the convex hull in case it lies outside. This measurement is useful to determine the minimum separation distance τ_{CH} between the two sets S_1 and S_2 in case no intersection is detected.

5.4 Results and Discussion

5.4.1 Detector Performance

The performance of the proposed spine ring detector is best appreciated in comparison with the revolving cylinders approach (section 3.4.3). By surrounding the dendrite shaft completely around a particular tracing element, the spine ring is posed to capture the entire “envelope” region; hence, it avoids cylindrical parameterization and angular quantization. This results in two main advantages of the proposed approach:

1. minimizing spine misses: the ring uses optimal parameters that are large enough to capture the surrounding fluorescence.
2. detection efficiency: by avoiding elevation, rotation and cylinder parameter optimization, the ring captures the entire surroundings in one shot. Although there is a cost associated with this speedup, namely, connected components labeling, it remains much less than the gain. On a typical image the revolving cylinders method would typically require a few hours on a 2GHz Windows machine, whereas the spine ring detector would finish in an order of 10 to 15 minutes.

Further enhancements are still possible with the proposed detector. For example, the fact that the ellipsoid models overlap can be utilized to optimize the number of annuli used. Information about neighboring branches can be utilized to further optimize the ring outer radius r_o . Moreover, multi-resolution detection can be implemented; the ring is initiated with a coarse resolution and in the event of any detection, the resolution is incrementally increased. This speeds up the intensity interpolation inside the detector.

5.4.2 Volume Merging

Two intersecting spine candidates are considered fragments of the same spine and are therefore merged into one. This seems plausible since volume intersection implies connectedness and, consequently, foreground continuity. However, our experiments showed that the intersection test, as described above, is too restrictive. A large number of neighboring false positives would have been merged together, except that they were up to

a few voxels apart. This motivated a relaxation of strict intersection in favor of a tolerance level for the minimum separating distance τ_{CH} . Now the fragments are merged if they are either intersecting or within τ_{CH} voxels apart. The value of τ_{CH} was decided empirically to be 3 voxels.

The convex hull merging technique has shown considerable improvement on the initial spine counts and did not exacerbate the false negative errors. The results are far better than the revolving cylinders method (Section 3.4.3). However, compared to the NeuronStudio method (Rodriguez, Ehlenberger et al. 2006) the results are mixed: they are still better in terms of spine misses (false negative errors) but not so for false positives. A comparative result between the two methods is shown in the table below.

Table 4 Comparison between NeuronStudio (NS) spine detection and our spine ring detector with convex hull merging (RPI). The images used in this experiment are provided by MBF-Bioscience. Due to the spine density in some images, only some dendrites were selected for comparison. The miss rate (false negatives) with the spine ring detector and convex hull merging (RPI FN) was better than that from NS. The same is not true for the false positives (RPI FP vs. NS FP). Improvements upon these results are discussed in chapter 7.

Image	Manual Count	NS Tru	RPI Tru	NS FP	RPI FP	NS FN	RPI FN	NS FN%	RPI FN%	NS FP%	RPI FP%
spine1	39	25	36	16	53	14	3	35.9	7.7	41	136
spine2	49	34	25	3	4	15	24	30.6	49	6.1	8.2
spine3	34	14	32	8	40	20	2	58.8	5.9	23.5	117.6
spine4	254	174	177	2	4	82	77	32.3	30.3	0.8	1.6
spine5	227	73	176	15	70	151	51	66.5	22.5	6.6	30.8
spine6	149	44	80	7	54	105	69	70.5	46.3	4.7	36.2
spine7	158	135	105	28	10	23	53	14.6	33.5	17.7	6.3
spine8	120	98	114	36	74	22	6	18.3	5.0	30.0	61.7
spine9	40	19	21	18	18	21	19	52.5	47.5	45.0	45.0
Mean								42.2	27.5	19.5	49.3

The lack of specificity demonstrated in this approach is mainly due to intensity variability within the spine images. As the image magnification decreases, the spine size tends to decrease causing more misses (spine2,5,6,7). Image quality appears to affect the false positive (FP) count. The worst FP performers in the above table (spine1,3,8) were the noisiest. Spine1 and spine3 contained too much blur around the dendrite surface. Spine8 was a low resolution image making some fragmented spines appear more separated than the tolerance distance τ_{CH} .

As with any image analysis threshold, this distance creates a tradeoff situation. Increasing τ_{CH} too much would falsely enlarge a spine or merge several spines into one big cluster counted as one spine; on the other hand, reducing τ_{CH} may leave many spines fragmented thus increasing the false positives. In the next section we propose a novel merging approach based on tracing the spine fragment back to the dendrite surface. The method provides us with a strong criterion for merger as well as allows for further spine shape characterization.

6. Wave-front Propagation in Grayscale Images

The neuronal tracing method (Phase I) described in chapter 4 paved the way for the proposed spine segmentation approach described in chapter 5 (Phase II). However, as we saw in the discussion 5.4, the problem of false positives is not satisfactorily solved by merging fragments based on volume intersection. Moreover, a re-attachment path from “detached” spines or fragments to the dendrite surface is needed for further spine validation and possibly classification.

Cheng et al (Cheng, Zhou et al. 2007) identified detached spines by calculating and thresholding the signal-to-noise ratio of globally detached blobs. Detached spine fragments are connected to an attached spine base using distance and orientation criteria. Koh et al. (Koh, Lindquist et al. 2002) used geometric heuristics to perform the attachment. Rodriguez et al. (Rodriguez, Ehlenberger et al. 2008) modeled the detached part as a bell-shaped curve and located its attached part by calculating its perpendicular distance to the model central axis. None of those methods attempt a non-linear connector between the detached spine and the dendrite surface.

We propose a novel approach to solve the re-attachment problem and enhance the spine validation by utilizing wavefront propagation. Using the fast marching technique, we propagate a directed wavefront from the spine to the dendrite surface. This creates a geodesic distance map for the region under examination. A minimal geodesic path is extracted from the map to reconnect the detached spine with the dendrite surface. The path is further used to validate spine candidates or induce spine fragment merger. These details are described in section 6.3 after reviewing the theoretical background in sections 6.1 and 6.2. Although we do not claim completeness in coverage, the mathematical formulations and some literature relevant to the usage of wave-front propagation in grayscale image analysis are presented. Since fast marching is a special case of level set methods, we start described with the latter.

6.1 Theoretical Background: On Level Sets

Level set methods are numerical techniques for computing the position of propagating interfaces (curves in 2D or surfaces in 3D) (Sethian 1996) in a variety of settings. Applications utilizing such surfaces are found in diverse engineering disciplines including fluid mechanics, computational geometry, image processing and computer vision (Sethian 1999; Osher 2002).

The level set concept is based on a closed interface $\Gamma(t)$ moving in a direction normal to itself with a known speed function $F(L, G, I)$, where L, G and I are parameters determined by local properties such as curvature and normal direction, global properties such as shape and position, and independent medium properties such as the underlying speed $F(L, G, I)$. The interface motion is parameterized in the level sets of $\phi(x(t), t)$, such that

$$\phi(x, t = 0) : R^N \rightarrow R; \quad (6.1)$$

$$\Gamma(t = 0) = \{x \mid \phi(x, t = 0) = 0\}, \quad (6.2)$$

where point¹ $x \in R^N$. This makes $\phi(x, t) = \pm d$ represent the level set of points at distance $\pm d$ (negative is to the inside of the closed curve) from $\Gamma(t)$. Since the evolving function ϕ always matches the propagating (hyper)surface Γ , we have

$$\phi(x(t), t) = 0.$$

Also, limiting the motion to be always normal to the surface, the following relation holds:

$$x_t \cdot n = F(x(t)),$$

where $x_t = x'(t)$ and the normal to the curve is $n = \frac{\nabla \phi}{|\nabla \phi|}$. Combining the above two relations and using the chain rule, we get the level set first-order partial differential equation (PDE)

$$\phi_t + F |\nabla \phi| = 0, \text{ given } \phi(x, t = 0). \quad (6.3)$$

¹ The vector notation is implicit throughout this brief description.

The above is a time-dependent Hamilton-Jacobi equation whose solution may lead to discontinuities causing the evolving curve to break or form sharp corners. However, using a weak solution borrowed from viscous fluid formulations, good numeric approximations are attainable with finite-difference schemes.

6.2 Theoretical Background: On Fast Marching Algorithms

In the special case that the speed F of the evolving curve is always positive (or negative), the level set formulation can be converted from a time-dependent PDE to a stationary one without a time variable. Let $T(x)$ represent the time at which the evolving curve crosses the point $x \in R^N$; i.e., $T(x) = t$ is the level set of points requiring a time lapse t to cross the curve. Because speed is inversely proportional to time difference, we get a simplified stationary Hamilton-Jacobi equation

$$|\nabla T| F = 1, \text{ given } \Gamma(t) = \{x \mid T(x) = t\}. \quad (6.4)$$

If F depends on position only, the above becomes the well-known Eikonal equation which is the basis for the Fast Marching level set method. As with the time-dependent general case, this equation may also lead to singularities, and requiring the same “viscosity” principle to be embedded in the solution. The numeric gradient approximation suggested by Sethian results in a fast solution scheme for this special case of monotonically advancing (or retreating) fronts. The efficiency of the technique has gained it widespread attention and made it applicable to a variety of engineering problems, including shape-from-shading problems, lithographic development calculations in microchip manufacturing, and arrival time problems in control theory (Sethian 1996), biomedical applications (Deschamps and Cohen 2001; Deschamps, Malladi et al. 2004; Yan, Zhuang et al. 2004; Bazin and Pham 2005; Sermesant, Coudiere et al. 2005; Cardinal, Meunier et al. 2006; Padfield, Rittscher et al. 2006; Tizon, Lin et al. 2007), image and video processing (Sharma and Reilly 2002; Moenning and Dodgson 2003)...

In the remaining part of this section, we describe the 3D mathematical formulation and the implementation details.

6.2.1 Fast Marching 3-D Formulation

Following the notation of (Deschamps and Cohen 2001; Sethian 2001), let $T(x, y, z)$ be the arrival time integrated along a path of travel between points $p_0(x_0, y_0, z_0)$ and $p(x, y, z)$ with a positive speed $F(x, y, z)$. It is required to compute the distance map (action map) which is the surface of minimal arrival times (action) going from p_0 towards p . This is done by solving the Eikonal equation

$$|\nabla T(x, y, z)|F(x, y, z) = 1. \quad (6.5)$$

As proposed by Sethian, the discrete approximation avoids over-shooting by implementing a finite difference in the upwind direction of the moving front giving the correct viscosity solution:

$$\begin{aligned} & (\max\{t - T_{i-1,j,k}, t - T_{i+1,j,k}, 0\})^2 + \\ & (\max\{t - T_{i,j-1,k}, t - T_{i,j+1,k}, 0\})^2 + \\ & (\max\{t - T_{i,j,k-1}, t - T_{i,j,k+1}, 0\})^2 = \frac{1}{F_{i,j,k}^2}. \end{aligned} \quad (6.6)$$

This upwind difference approximation implies that information propagates only from small values to large values of T . That is, the arrival time of grid point $p(x, y, z)$ gets updated only by neighboring points with smaller T . This monotonic property allows for the maintenance of a narrow band of candidate points around the front representing its outward motion. The property can further be exploited for the design of a simple and efficient algorithm by freezing the T values of existing points and subsequently inserting neighboring ones into the narrow band thereby marching the band forward (Moenning and Dodgson 2003). The solution guarantees stability (no singularities) by choosing the maximal upwind difference estimate.

6.2.2 Fast Marching Implementation

The basic fast marching algorithm is detailed in Table 5 and Table 6 following (Deschamps and Cohen 2001). Three lists are maintained: a list of “alive” points with

finalized arrival times and located to the inside of the propagating surface, a list of points “*far*” from the surface, and a list of border points- “*trial*”.

Starting from a pre-chosen initial point p_0 , update the arrivals of its neighboring points $A_1, A_2, B_1, B_2, C_1, C_2$ marking them *trial*. Choose the *trial* point with the smallest arrival time, update its 6 neighbors and move them to *trial*, then save it in the *alive* list. Choose the next *trial* point and repeat.

Updating the arrival times requires a numeric solution for Eq.6.6. Each of the three elements on the right hand side needs to be maximized. Knowing that only the starting point will have a zero arrival time, there remains to pick the maximum differential on each side of opposite neighbors on the grid:

$$\begin{aligned} & (\max\{t - T_{i-1,j,k}, t - T_{i+1,j,k}\})^2 + \\ & (\max\{t - T_{i,j-1,k}, t - T_{i,j+1,k}\})^2 + \\ & (\max\{t - T_{i,j,k-1}, t - T_{i,j,k+1}\})^2. \end{aligned}$$

This is equivalent to choosing the neighbor with the minimum arrival time for each pair in solving Eq.6.6. Therefore, without loss of generality, we assume the neighboring arrivals of every *live* point in consideration satisfy

$$T_{A_1} \leq T_{A_2}, T_{B_1} \leq T_{B_2}, T_{C_1} \leq T_{C_2}; \text{ and } T_{C_1} \geq T_{B_1} \geq T_{A_1}. \quad (6.7)$$

Combining Eq.6.6 and 6.7 leads to

$$(t \geq T_{A_1})^2 + (t \geq T_{B_1})^2 + (t \geq T_{C_1})^2 = \frac{1}{F^2}, \quad (6.8)$$

where t is the new arrival time to be found and F is the local speed. Since it is assumed that the wavefront will propagate with monotonic speed, the new arrival time t values must be larger than the latest frozen value. This is a key point in the algorithm because it guarantees a solution as well as stability.

The algorithm is efficient because of the unidirectional speed of motion (always towards the outside or the inside of the surface). In addition, each point in the domain is selected once from among the *trial* list and never visited again. A heap sort storage of the bordering *trial* points makes the algorithm become even more efficient. With a total of $M = n \times p \times q$ points in 3D, the algorithm is $O(M \log(M))$ (Sethian 1996). Moreover,

the distance map T is computed directly across the domain (3D image in our case) and a separate binary image providing the source points will not be required. Finally, since the arrival time information of a grid point is only propagated in the direction of increasing distance, the size of the narrow band remains small eliminating the need for large storage or slower sorting.

Table 5 Front Propagation Algorithm using the Fast Marching Algorithm

<ul style="list-style-type: none"> • Define the following three types of points:
<ul style="list-style-type: none"> ○ <i>Alive</i> is the set of all grid points at which the action value t has been reached and will not be changed;
<ul style="list-style-type: none"> ○ <i>Trial</i> is the set of next grid points (6-connectivity neighbors) to be examined and for which an estimate of T has been computed using the above equation.
<ul style="list-style-type: none"> ○ <i>Far</i> is the set of all other grid points, for which there is not yet an estimate for T
<ul style="list-style-type: none"> • Initialization:
<ul style="list-style-type: none"> ○ <i>Alive</i> set is confined to the starting point p, with $T(p_0) = 0$
<ul style="list-style-type: none"> ○ <i>Trial</i> is confined to the six neighbors p of p_0 with initial value $T(p) = \frac{1}{F(p)}$
<ul style="list-style-type: none"> ○ <i>Far</i> is the set of all other grid points p with $T(p) = \infty$ (Numerically, this is substituted by a number much larger than $\max_p \frac{1}{F(p)}$)
<ul style="list-style-type: none"> • Loop:
<ul style="list-style-type: none"> ○ Let $(i_{\min}, j_{\min}, k_{\min})$ be the <i>Trial</i> point with the smallest action $T_{i_{\min}, j_{\min}, k_{\min}}$
<ul style="list-style-type: none"> ○ Move it from the <i>Trial</i> to the <i>Alive</i> set (i.e. $T_{i_{\min}, j_{\min}, k_{\min}}$ is frozen)
<ul style="list-style-type: none"> ○ For each neighbor $p(i, j, k) \in \{A_1, A_2, B_1, B_2, C_1, C_2\}$ (6-connectivity in 3D) of $(i_{\min}, j_{\min}, k_{\min})$ <ul style="list-style-type: none"> ▪ If $p(i, j, k)$ is <i>Far</i>, add it to the <i>Trial</i> set and compute $T_{i, j, k}$ using the gradient approximation of the above Eikonal equation as shown in Table 6. ▪ If $p(i, j, k)$ is <i>Trial</i>, re-compute the action $T_{i, j, k}$ and update it. ▪ If $p(i, j, k)$ is <i>Alive</i>, do not update its arrival time $T_{i, j, k}$

Table 6 Algorithm for solving the PDE and updating the propagation arrival times

Solving locally the upwind scheme in 3D	
<ul style="list-style-type: none"> We need to solve for t in the quadratic equation 	
$(t \geq T_{A_1})^2 + (t \geq T_{B_1})^2 + (t \geq T_{C_1})^2 = \frac{1}{F^2} \quad (6.9)$	
<ul style="list-style-type: none"> Assumption 1: $t \geq T_{C_1} \geq T_{B_1} \geq T_{A_1}$ (without loss of generality in ordering the neighboring arrival times). Let Δ_c be the discriminant of Eq.6.9. 	
<ul style="list-style-type: none"> o If $\Delta_c \geq 0$, pick t' to be the larger of the two solutions of Eq.6.9. <ul style="list-style-type: none"> ▪ If $t' \geq T_{C_1}$, return t' ▪ Otherwise, go to Assumption2 	
<ul style="list-style-type: none"> o If $\Delta_c < 0$, at least one of the neighboring arrival times T_{C_1}, T_{B_1} or T_{A_1} is too large to influence the solution. This makes the assumption $t \geq T_{C_1} \geq T_{B_1} \geq T_{A_1}$ invalid. Go to Assumption2. 	
<ul style="list-style-type: none"> Assumption 2: $t \geq T_{B_1} \geq T_{A_1}$ and $t < T_c$. Let Δ_b be the discriminant of 	
$(t \geq T_{A_1})^2 + (t \geq T_{B_1})^2 = \frac{1}{F^2} \quad (6.10)$	
<ul style="list-style-type: none"> o If $\Delta_b \geq 0$, pick t' to be the larger of the two solutions of Eq.6.10. <ul style="list-style-type: none"> ▪ If $t' \geq T_{B_1}$, return t' ▪ Otherwise, go to Default. 	
<ul style="list-style-type: none"> o If $\Delta_b < 0$, the neighboring arrival time T_{B_1} is too large to influence the solution. This makes the assumption $t \geq T_{B_1} \geq T_{A_1}$ invalid. Go to Default.. 	
<p>Default: having $t < T_c$ and $t < T_b$, we must have $t \geq T_a$. Return $t' = T_a + \frac{1}{F(p)}$</p>	

6.3 Spine-Dendrite Geodesic Tractography

In this section we describe our proposal for a spine candidate validation model that utilizes the grayscale appearance as well as the relative position between the spine and the dendrite surface. For each spine candidate (or spine fragment), we propagate a wavefront powered by the pixel intensity differential. Solving the wavefront propagation equation results in computing a unidirectional geodesic distance map from a seed point on the spine to the dendrite surface; i.e. the wavefront velocity is unidirectional with respect to propagation. Our purpose of the distance map is to extract the shortest geodesic path that connects the seed point to the dendrite surface. A convenient way to achieve this is to use the efficient fast marching technique to solve the Eikonal Equation, a special partial differential equation (PDE). The theoretical background is detailed in the previous section.

Several groups have used fast marching for extracting curvilinear structures in biological or medical images. For example, Deschamps and Cohen (Deschamps and Cohen 2001) used it to extract minimal endoscopy paths in 3D CT and MRI imagery. Geodesic tractography from fast marching propagation has been used by Jbabdi et al.(Jbabdi, Bellec et al. 2008) to extract and connect brain tracts from MRI images. At the time of this writing, this technique has not been utilized in spine segmentation approaches.

6.3.1 Wavefront Propagation

The fast marching approach introduced by Sethian(Sethian 1996) (section 6.2) propagates a wavefront, starting from a designated seed point(s), and to the entire region of interest using an weight or speed function. The wavefront arrival times at the region pixels make up the geodesic distance map from which an optimal path to an end point can be deduced (see Figure 30). Since spine candidates are independent from one another, we deal with each one separately; i.e., a suitable volume around the spine candidate is masked in order to minimize wavefront leakage. In the following sections we discuss the wavefront elements: speed function, starting and end points and path extraction.

6.3.2 Speed Function

As per the discussion in section 6.2.2, the propagation favors the stronger weights (speed values) at the wavefront boundary. In other words, the higher the speed is in a certain sub-region, the faster the wave propagates through it. A pixel of 0-speed is a propagation barrier that halts the wavefront.

Ideally, the speed function is fastest inside the desired region and slowest (or zero) outside the region boundary. In order to achieve this property, the region may be smoothed and the edges enhanced. However, given the tiny size of a spine in an image, smoothing a spine might blur its head with surrounding spines, dendrite, or noise, and might erase the thin neck completely. Edge enhancement on thin structures would also leave the spine with little internal structure of non-zero speed.

The above reasons, verified empirically, led us to choose the original grayscale intensity as the basis for the weight function albeit with a slight modification. As mentioned previously propagation freezes at 0-speed pixels (background in this choice of speed function). However, this is undesirable with detached spines because the wavefront would never leave the spine head. Therefore, we modify the above energy function in such a way that the speed is never zero, even for background pixels. This modification has proven to work mostly well for all our experiments.

6.3.3 Starting and Ending Points

A common characteristic of all spines is that they emanate from some dendrite surface regardless of whether this is apparent in the image. This gives us a clue that the propagation must start either from the spine tip towards the dendrite surface or *vice versa*. The question is how to determine these locations. As an initial approach we chose the pixel that is furthest, in Euclidean sense, from the dendrite surface as the spine “tip”. This works well in most cases. Multiple starting points can also be used at the same time in order to eliminate the potential errors with curvy spines that get closer to the dendrite at the head than in the middle.

The spine root at the dendrite surface is more challenging because the spine neck might not be attached to the dendrite, or it might be too faint to be detected fully. Our solution to this problem is to let the wavefront propagate all the way to the dendrite central axis rather than stop at a pre-determined point. Thus we choose several dendrite center points as candidate points with zero speed that would force the wavefront to stop. The first such point reached is the closest to the spine tip geodetically and hence allows for optimal path extraction (Figure 31). The spine root at the dendrite surface is then picked from this path by applying the super-ellipsoid distance function (Eq. 5.4). The first point outside the super-ellipsoid ($F > 1$) is considered the spine root at the surface. It is important to note that this solution does not rely on Euclidean measures and is driven primarily by the intensity values in the region.

6.3.4 Path Extraction

The solution of the Eikonal equation at a given pixel is the wavefront arrival time t which is proportional to the geodesic distance traveled from the starting point. Hence these solutions are called the geodesic distance map. Because the distance at any given location is optimal by design, the local direction of the shortest path back to the starting point is along the negative gradient of the distance map. Thus given an end point, backtracking through gradient descent towards the starting point gives the optimal path. This can be formulated mathematically by the following equation:

$$\frac{dP}{ds} = -\overline{\nabla T} \text{ with } P(0) = x_1, \quad (6.11)$$

where $P(s)$ is a parametric curve representing the path, T is the distance map and x_1 is the path end point.

Because such path traverses the spine body based on the biomarker fluorescence along the path to the dendrite, it is more accurate than straight line estimations as calculated by (Koh, Lindquist et al. 2002) and (Rodriguez, Ehlenberger et al. 2008). It potentially serves three purposes:

- Reconnects an apparently detached spine (or fragment) to its dendrite

- Provides a basis for comparison with a path from a neighboring spine fragment in order to decide on merging the fragments.
- Eliminates invalid spines that cannot produce a central axis. This is useful for future edit-based validation tools as in FARSIGHT (Roysam et al.).

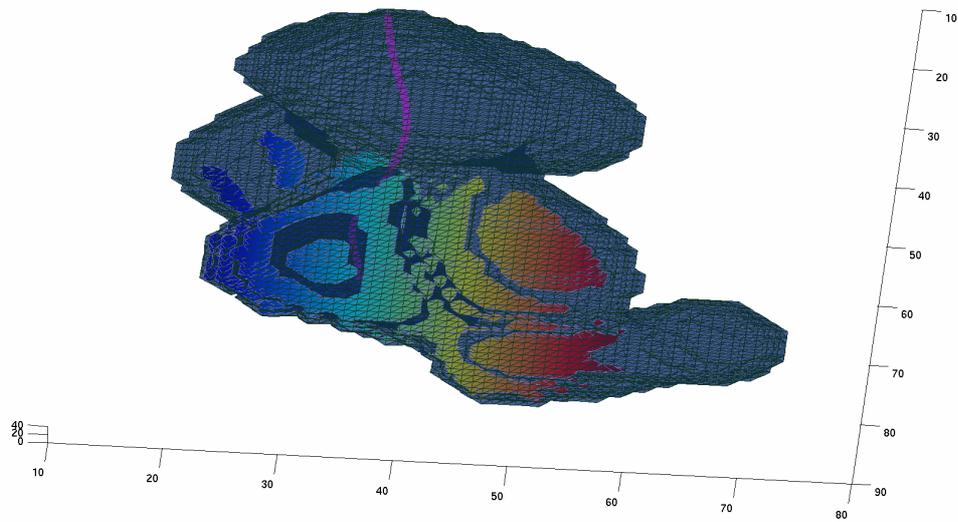


Figure 30 Color coded 3D view of a typical distance map generated by wavefront propagation. The lighter colors correspond to faster arrivals (shorter geodesic distances). The extracted path appears in purple towards the top.

7. Experiments and Results

7.1 Images, Preprocessing and Tracing

The data sets used are from various research collaborators using different animals. The image quality varies in terms of resolution (x, y and z), anisotropy ratio, noise, biomarker expression and spine density. The following table lists the spatial resolutions and anisotropy ratios in the images used in this study.

Table 7 Anisotropy ratios in our experimental images.

image name	Anisotropy Ratio (Z/X)
spine1	4.42
spine2	5.11
spine3	10.12
spine4	0.79
spine5	1.34
spine6	1.10
spine7	1.21
spine8	1.45
spine9	7.60
spine10	0.87
312tiff	4.00
648tiff	4.00

As the spine size decreases, the similarity between spine and noise increases. Due to the minuteness of spines in most images, image preprocessing is rarely done in order to preserve maximal spine information. Blind deconvolution (for deblurring) is performed only in cases where noise near the dendrite surface is strong and the spine intensities contrast well with the background. Tracing is performed in batch mode using Tyrrell's (Tyrrell, di Tomaso et al. 2007) code. The traces are then automatically fixed by our algorithm discussed in section 4.4.14.4, and then validated by manual inspection.

7.2 Ring Detection

Two sets of experiments have been conducted for comparing the cylindrical to the super-ellipsoidal ring detector. The input to this step is the corrected traces. The experiments were run with the default parameter settings for both detectors. The cylindrical detector critical parameters are inner radius, outer radius and up-sampling scale. The thickness is determined dynamically from the gap separating the dendrite centerlines. On the other hand, the super-ellipsoidal ring utilizes the radii of the dendrite super ellipsoidal segment as the basis for the inner ring dimensions. The inner ring is driven away from the dendrite surface by scaling those radii by a small factor of 1.5. This is a default value that can be used for all experiments even though it can be user-customizable. The outer radii are set to a scale of 5 times the inner ones to ensure adequate detection at the farther end of the ring. In both experiments up-sampling proved more useful as the image resolution decreases. The output of this step is a set of spine candidates associated with their corresponding dendrites. The subsequent steps of the algorithm are common to both ring types.

7.3 Wave Propagation and Medial Axis Extraction

After the initial spine ring detection, wavefront propagation and spine-to-dendrite paths are extracted by gradient descent as explained in section 6.3. This requires setting the descent direction step size as the desired path resolution. The default step size is at a sub-pixel value of 0.5. Regardless of their size, most detected spines (or spine fragments) carry enough information to generate such a path. However, in rare cases the path may be invalid either because its length is too short or because its distance to the dendrite surface is less than a minimal acceptable threshold. These thresholds are either set empirically or left for edit-based validation to be group-selected (see section 7.4.5.2).

For the path computation we find the nearest dendrite center point and its n neighbors on both sides where n is chosen such that it is large enough to cover the spine projection on the dendrite. Larger values do not affect the results but may compromise the performance as the spine region extracted for wave propagation becomes larger. Typically a value in $[5, 10]$ is adequate.

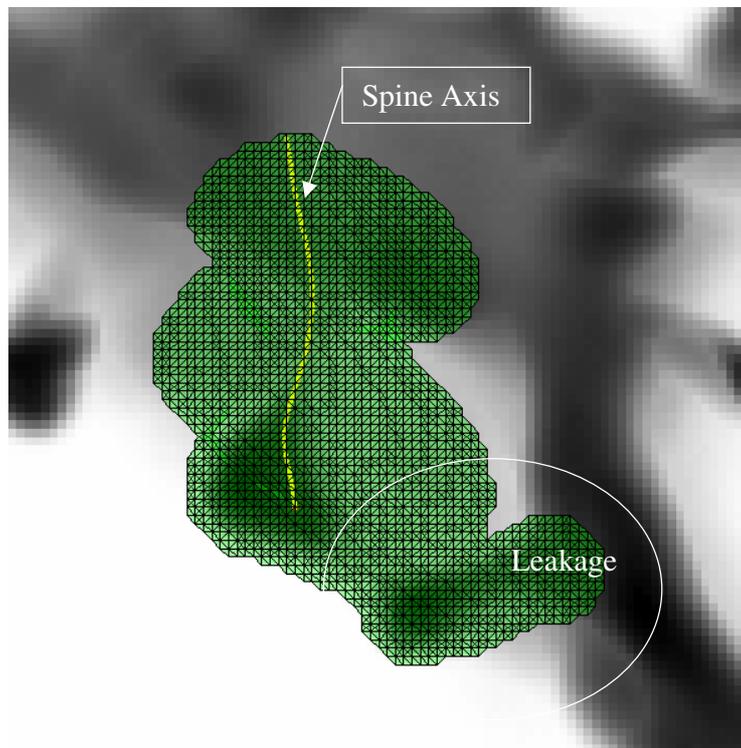


Figure 31. Wavefront propagation from spine head leaks towards a neighboring spine that is closer to the dendrite. However, because of the choice of multiple end points at the dendrite center, the error is avoided by choosing the closest dendrite center point geodetically. This produces a correct spine axis (yellow line).

7.4 Final Results

7.4.1 Cylindrical Ring

In the first experiment set the cylindrical ring is used although the underlying dendrite model was assumed to be super-ellipsoidal. This is the initial ring detector proof of concept study and is kept here for comparison with the second experiment set using the super-ellipsoidal ring.

In the first study, the results were compared to those of the state-of-the-art tool NeuronStudio of (Rodriguez, Ehlenberger et al. 2006; Rodriguez, Ehlenberger et al. 2008). Our algorithm is superior in terms of detection and is comparable in terms of false positives (Table 8). In the same study, our results have been compared with up to 5 expert

human observers from MBF-Bioscience. We demonstrate that our tool improves on human detection by up to 30% and an average of 12%. This is mainly due to the 3D volume visualization as well as the spine size minuteness that can easily escape the human eye at first examination. Small, mainly stubby, spines in the z-direction could only be verified by 3-D rendering with rotation. These auto-corrections are summarized in Table 9 below.

7.4.2 Super-ellipsoidal Ring

The second experiment set exploited the super-ellipsoidal ring detector in order to examine the significance of inner ring similarity to the dendrite shape model. The study utilized all 10 images from MBF-Bioscience (spine1-10) as well as two images from Professor Trachtenberg's and two from Professor Potter's time-lapse data sets. Taking into account the same images only (Spine1-10) the average false positive error decreased to 15% (a 23% improvement) while the average false negative error decreased to 5% (80% improvement). Moreover, the clustered spines detected together have also been reduced to below 1%. These results clearly indicate the validity of matching the inner ring shape to that of the dendrite.

7.4.3 Path and Feature Extraction

In general, the spine paths extracted using the fast marching technique contributed to the accuracy of the results due to the merging reliability. This also showed significant improvement over the previous convex hull merging technique of section 5.4.2. Moreover, obtaining the path allows us to quantify the spine neck (if it exists) as it can be distinguished from the head. Measuring the width or volume of the spine head can be done either by fitting a sphere or a super-ellipsoid around the path starting point. We used the super ellipsoid fitting algorithm of (Tyrrell, di Tomaso et al. 2007) and the maximum between the two radii is used. Other features such as spine length, volume, distance from the dendrite (for detached spines) and average intensity can easily be computed using the segmented spine. Such features are useful for group-based validation where a group of spines can be selected from the output based on a visual inspection. A pilot effort is being

developed in the FARSIGHT toolkit (Roysam et al.) framework for such capability (Figure 40).

Table 8 Single-observer comparison between NeuronStudio spine segmentation (NS) and our cylindrical spine ring detector with paths extracted by Fast Marching (RPI). The false negative counts are still better, however, the false negative counts are much more improved than those done by convex hull intersection (Table 4). Further Improvements to this ring is achieved with the super-ellipsoid ring (Table 10).

Image	Manual Count	NS Tru	RPI Tru	NS FP	RPI FP	NS FN	RPI FN	NS FN %	RPI FN %	NS FP %	RPI FP %
spine1	40	25	36	16	10	14	4	35.0	10.0	40	25.0
spine2	37	34	28	3	5	3	9	8.1	24.3	8.1	13.5
spine3	24	14	24	8	16	20	0	83.3	0.0	33.3	66.7
spine4	86	59	58	6	0	27	26	31.4	30.2	6.9	0.0
spine5	103	49	53	19	7	54	50	52.4	48.5	18.4	6.8
spine6	94	26	38	8	12	68	56	72.3	59.6	8.5	12.8
spine7	115	107	91	23	4	8	24	7	20.9	20.0	3.5
spine8	34	31	32	8	12	3	2	8.8	5.9	23.5	35.3
spine9	39	33	28	8	5	4	9	10.3	23.0	20.5	12.8
Av								34.3	24.7	19.9	19.6

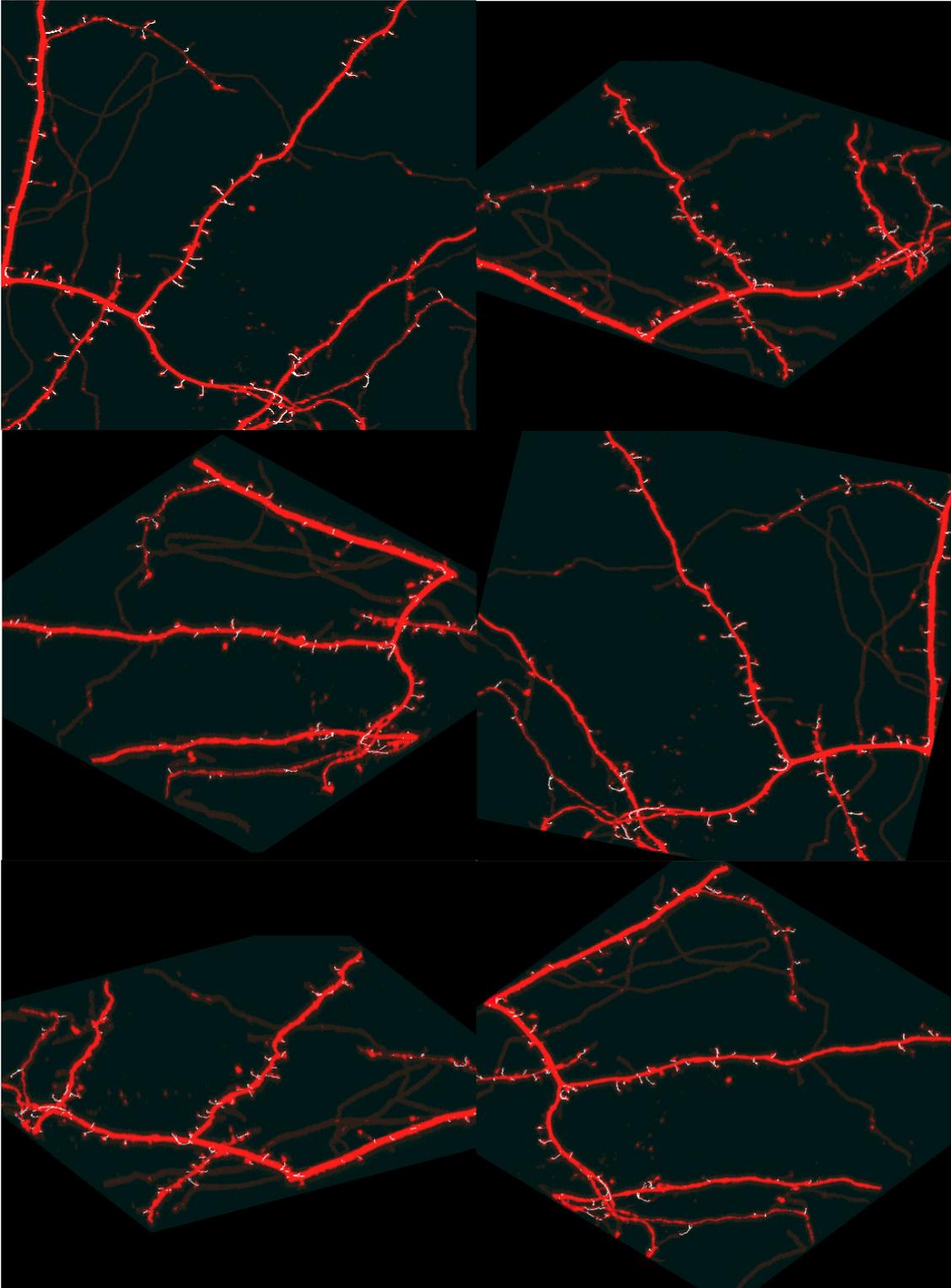


Figure 32 Several views of Spine9 Image rendered and rotated with detected spine paths superimposed in white. Rendering by voxx software. Image courtesy MBF-Bioscience.

Table 9 Multiple-observer validation: Up to 5 observers from MBF-Bioscience validated spine detection done by our method (spine ring detector and spine path merging by fast marching). The method improves on human observer detection as noted in the difference between the first two columns. There are still spine clusters being merged together and require further separation. “Manual” is the observers’ average initial marking; “Corrected” includes “Manual” and those detected by the method only; “TP” is automatic true positives; “FP” is false positives and over-counting; “FN” is false negatives; “Imp” and “Imp%” are detections improving on manual counting; “Clusters” and “Clust %” are correctly-detected spines but are clustered together.

Image	Man- ual	Corre- cted	TP	FP	FN	Imp	Clust- ers	TP %	FP %	FN %	Imp %	Clust %
Spine1	56	63	46	15	9	7	8	73.0	23.8	14.3	12.5	12.7
Spine7	259	268	166	3	38	9	64	61.9	1.1	14.2	3.5	23.9
Spine8	68	71	45	0	12	3	14	63.4	0.0	16.9	4.4	19.7
Spine9	104	124	101	10	21	20	2	81.5	8.1	16.9	19.2	1.6
Spine10	66	86	68	22	10	20	8	79.1	25.6	11.6	30.3	9.3
AVERAGE									11.7	14.8	12.6	13.4

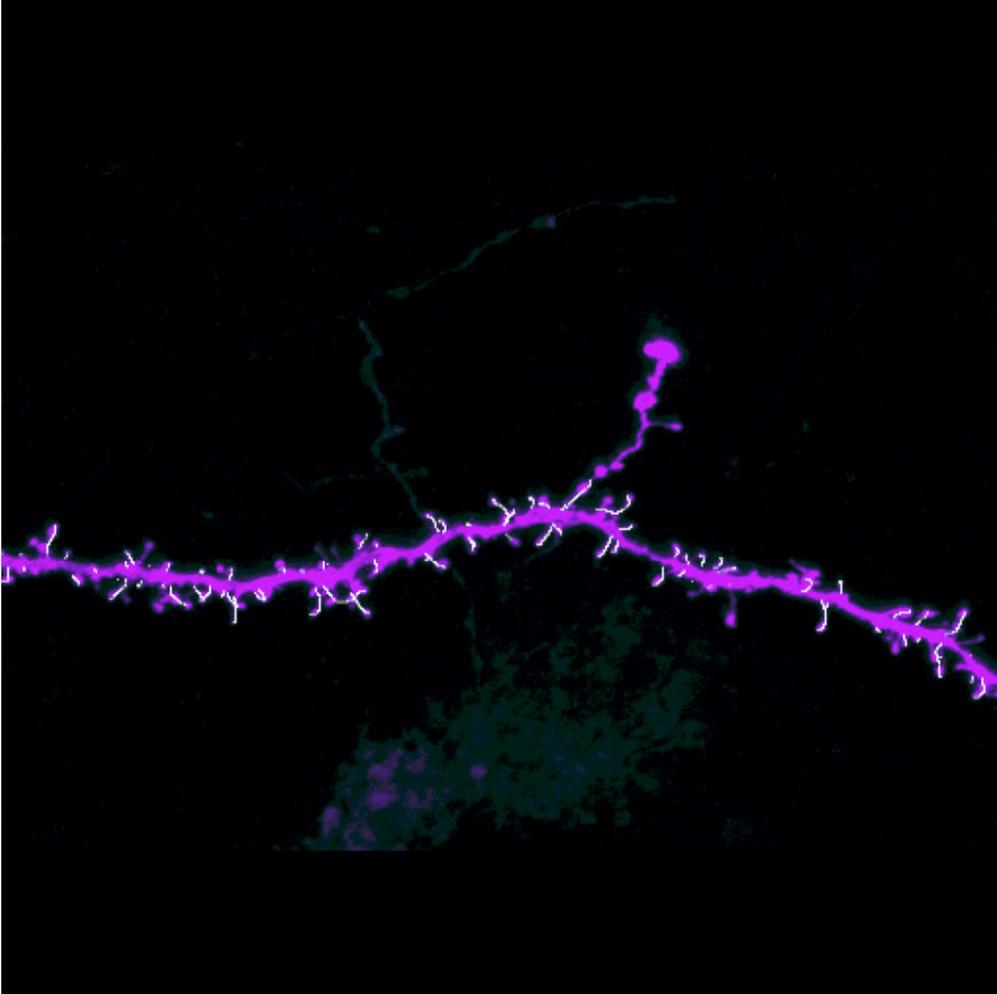


Figure 33 Spine4 Image rendering with detected spine paths superimposed in white. Rendering by voxx software. Image courtesy MBF-Bioscience.

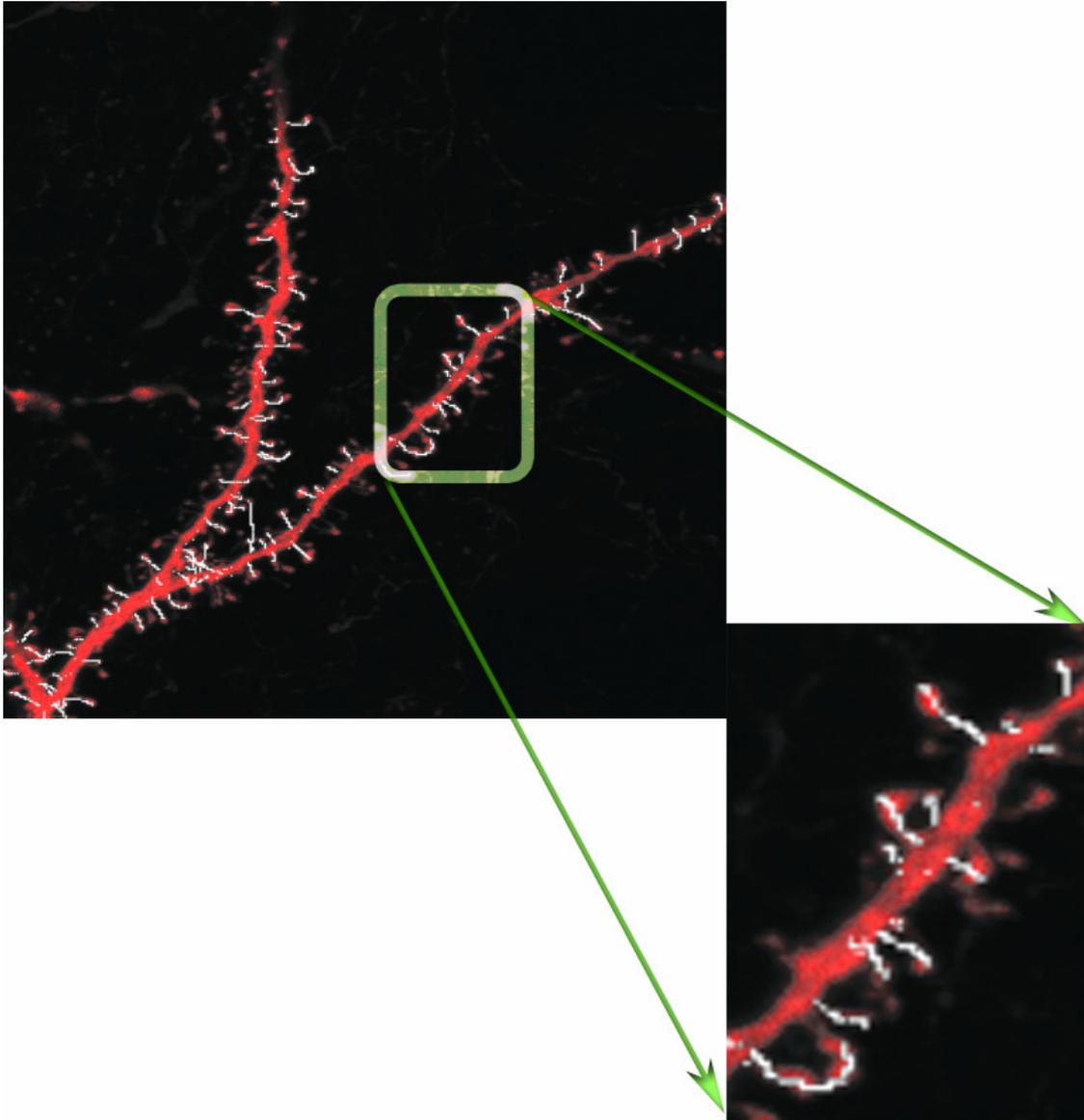


Figure 34 Spine7 Image rendering with detected spine paths superimposed in white and a selected region zoomed in. The bottom 2 spines are detected together as one spine. This accounts for one missed spine. Rendering by voxx software. Image courtesy MBF-Bioscience.

Table 10 Results with the super-ellipsoidal ring validated with the new SeedEditor tool. There is significant decrease in clustered spines and false negatives as compared with the cylindrical ring (Table 8). However, the false positive average error is slightly improved. Improvement on human detection error is not reported for this experiment.

Image Name	TP	FP	FN	Clusters	FP%	FN%	Clusters%
spine1	50	10	5		20.00%	10.00%	0.00%
spine2	50	3	4		6.00%	8.00%	0.00%
spine3	24	14	2		58.33%	8.33%	0.00%
spine4	197	13	1	4	6.60%	0.51%	2.03%
spine5	206	23	3		11.17%	1.46%	0.00%
spine6	73	10			13.70%	0.00%	0.00%
spine7	139	2	20	2	1.44%	14.39%	1.44%
spine8	99	1	4		1.01%	4.04%	0.00%
spine9	230	60	5	1	26.09%	2.17%	0.43%
spine10	99	8	5		8.08%	5.05%	0.00%
312tiff	90	18	3		20.00%	3.33%	0.00%
648tiff	78	23	4		29.49%	5.13%	0.00%
11A	33	4	1		12.12%	3.03%	0.00%
11B	23	7	1		30.43%	4.35%	0.00%
Average					15.24%	5.4%	0.39%
Spine1-Spine10							
Total Average					17.46%	4.99%	0.28%
Overall Median					12.12%	4.04%	0.3%

7.4.4 Discussion

In this section we try to analyze the error rates as they contrast from one method to the other and as they correlate with the image characteristics.

7.4.4.1 Comparing Results

We ran confidence measure tests on the results of Table 8 and Table 10. Table 8 compares the results between our cylindrical ring detector and the method of NeuronStudio. The results were taken from the same set of dendrite segments with their spines; therefore, a Wilcoxon signed-rank test has been used. However, the results were insignificant even for a 90% confidence interval.

On the other hand the super-ellipsoidal ring results of Table 10 come from a different sample set of dendrite segments and spines from those shown in Table 8; therefore, the Mann-Whitney-Wilcoxon test is used. For the same reason the same test is used to compare the NeuronStudio results with the super-ellipsoidal ring results. For each pair of methods we compared 3 results: false negatives together, false positives together and combined total errors together. For all 3 pairs of methods, comparing false positive rates was insignificant for 90% of confidence. However, when comparing false negative rates as well as combined error rates, the tests were significant with at least 95% of confidence. This result suggests that the super-ellipsoidal ring did produce a significant improvement over the previous two methods.

7.4.4.2 Error Sources

As we noted in the beginning of this chapter, the images we used vary from different perspectives: resolution, anisotropy ratio, noise, biomarker expression and spine density. Here we briefly discuss the causes that would generally lead to the errors we encountered.

Table 11 Combined error rates anisotropy ratio (Table 7). The first column is the sum of the false positive and false negative errors in Table 10.

Image Name	Combined Error%	Anisotropy Ratio Z/X
spine1	30.00%	4.42
spine2	14.00%	5.11
spine3	66.67%	10.12
spine4	7.11%	0.79
spine5	12.62%	1.34
spine6	13.70%	1.10
Spine7	15.83%	1.21
Spine8	5.05%	1.45
Spine9	28.26%	7.60
Spine10	13.13%	0.87
312tiffs	23.33%	4.00
648tiffs	34.62%	4.00
11A	15.15%	
11B	34.78%	

Anisotropy ratio: Images of high ratio (4.0 and above: spine1, spine2, spine3, spine9, 312tiff, 648tiff) resulted in the largest combined errors (30%, 14%, 67%, 28%, 23% and 34% respectively). The best images in terms of error rates are spine4 and spine8 that are characterized by low anisotropy ratio and adequate resolution. If higher anisotropy is inevitable, it should not increase by more than 1.5.

Noise: This is not too much of a concern for the dendrite tracing method as its robustness against noise has been shown. The effect of noise on our spine segmentation algorithm is primarily in the ring detector and particularly near the dendrite surface. Such noise results in a large number of false positives (spine3 and spine1) regardless of resolution. The cause of this imaging artifact has been discussed in classic literature and is usually take care of with de-blurring or median filtering. These methods have been of

limited effect, and sometimes even counterproductive, with spine images. The spine head or neck diameter may be so small that the noise filter would effectively delete it.

Spine density: this is the number of spines per dendrite unit length. As this number increases, spines tend to appear too close to each other that the human eye can hardly discriminate between them (spine5,6,7 for example). Moreover, the biomarker tends to cause extra fluorescence between them and the dendrite surface leading to more false positives and clusters. Higher imaging resolution would help especially if it enlarges the spine image beyond the effects of the nearby noise (*unlike* the case of spine1).

Spine Size: this depends on the image resolution. The larger the spine appears in the image, the more likely it is to contrast with the noise near the dendrite surface. Moreover, the ring detector favors large contiguous structures for counting as a single connected component, and hence, lowering the potential false positives. Obviously, the more the spine gets filled with the fluorescent marker, the larger it will be and the more likely it is to have a neck.

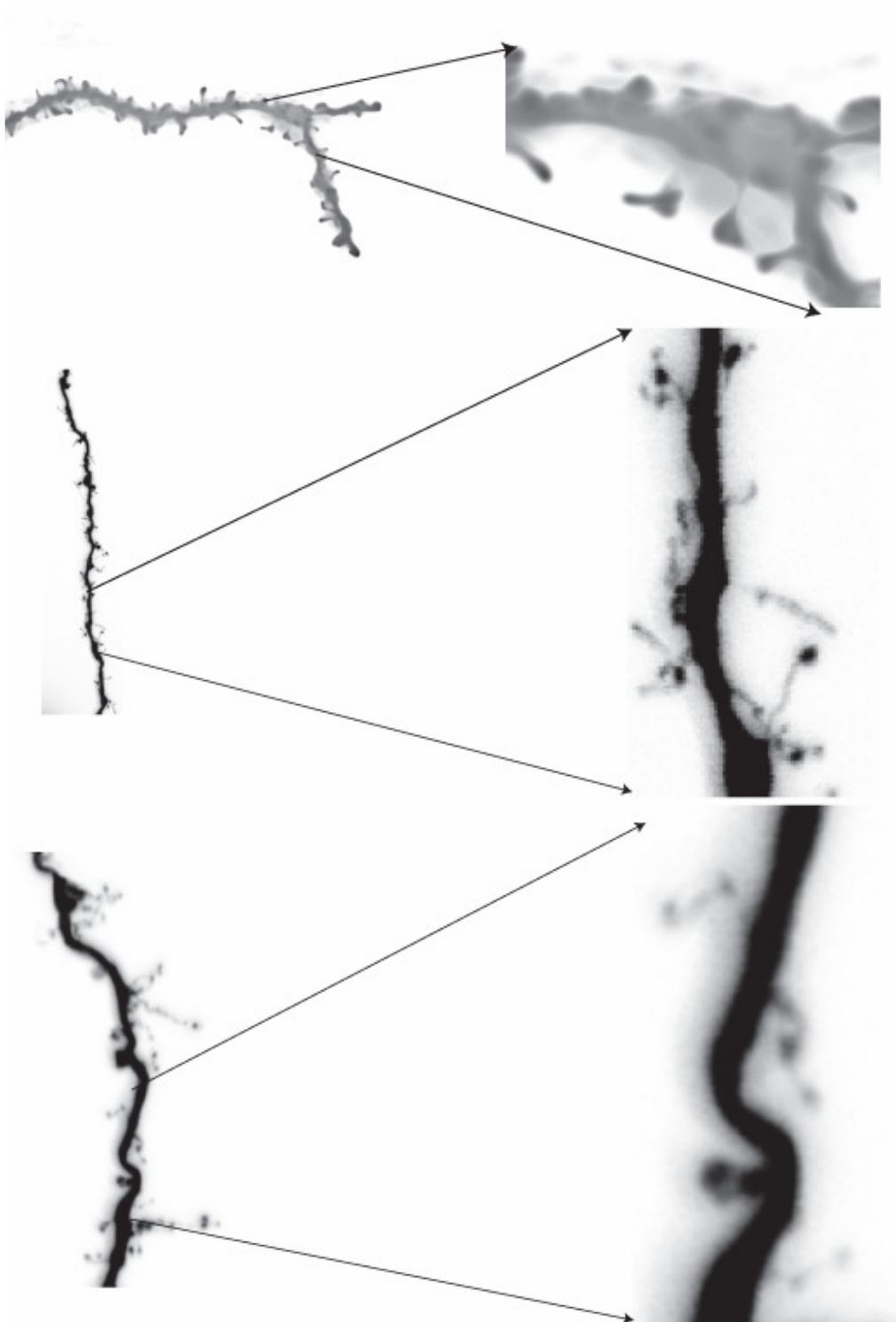


Figure 35 Left: minimum intensity projections of spine1 (top), spine2 and spine3. Right: zoomed-in regions demonstrating error from noise near the dendrite surface.

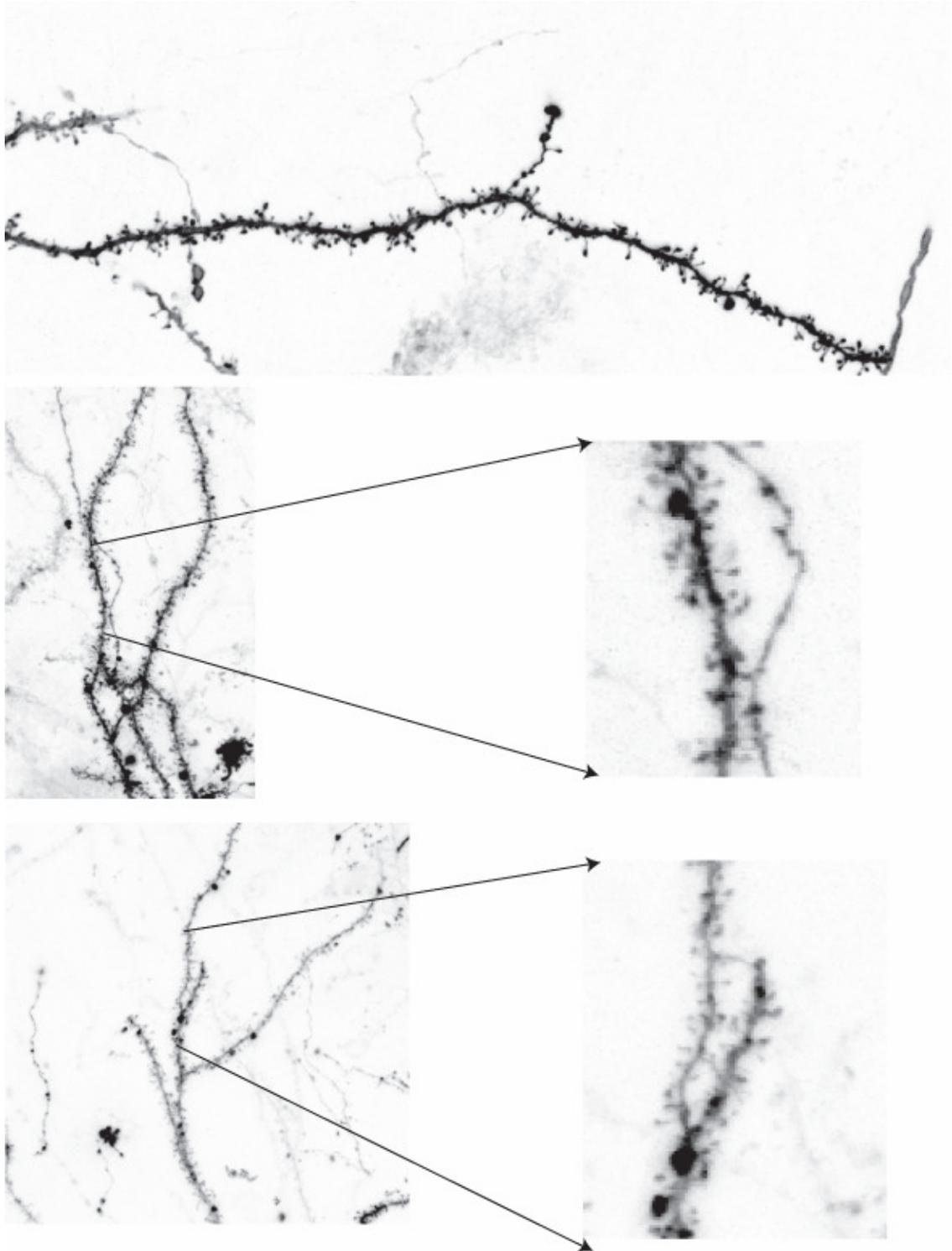


Figure 36 Minimum intensity projections of spine4 (top), spine5 and spine6. Right: zoomed-in regions demonstrating error from high spine density and low resolution.

7.4.5 Validation Tools

Due to the difficulty in visualizing the 3D volume and especially given the minuteness of the spines, a number of tools have been used to achieve satisfactory validation. NeuroLucida, a proprietary software from MBFBioscience, Inc., allows for inserting markers on any slice within a sliding 2D view. Among its features is a 3D rendering view including the markers that can be loaded, edited and saved. This is useful for images with a small number of spines and a few slices. However, as the slice number increases, navigating across the slices becomes inconvenient especially when many spines need to be validated. Voxx is a freely-available software providing fast 3D rendering but no marker capability. It is useful for its speed, rendering quality and automatic movie generation.

We also developed a dendrite flattening routine to alleviate the need for repetitive rotations while validation. However, due to the rounding involved in Cartesian to polar coordinate transformation, the image quality has not been useful enough for our purposes (Figure 37).

7.4.5.1 FARSIGHT SeedEditor

Due to the limitations of the above tools, we contributed to the development of a new validation tool within FARSIGHT that can render a volume image on which superimposed are seed points read from a text file for validation. The tool can edit the seeds and modify their colors according to the user judgment of false positive, cluster, deleted or newly added status (Figure 39). This tool has proven very useful in the latter part of this study. It renders rotations faster than the above tools and is user-friendly. It can also read a ground truth marker file simultaneously for comparison with the detected seeds. It is also open source and easily extendible. Nevertheless, the tool's main limitation is in the nature of 3D rendering that sometimes compromises small or faint spines and lead the user to label a correctly detected spine as a false positive. This necessitates occasional consultation through a more original view such as that in NeuroLucida.

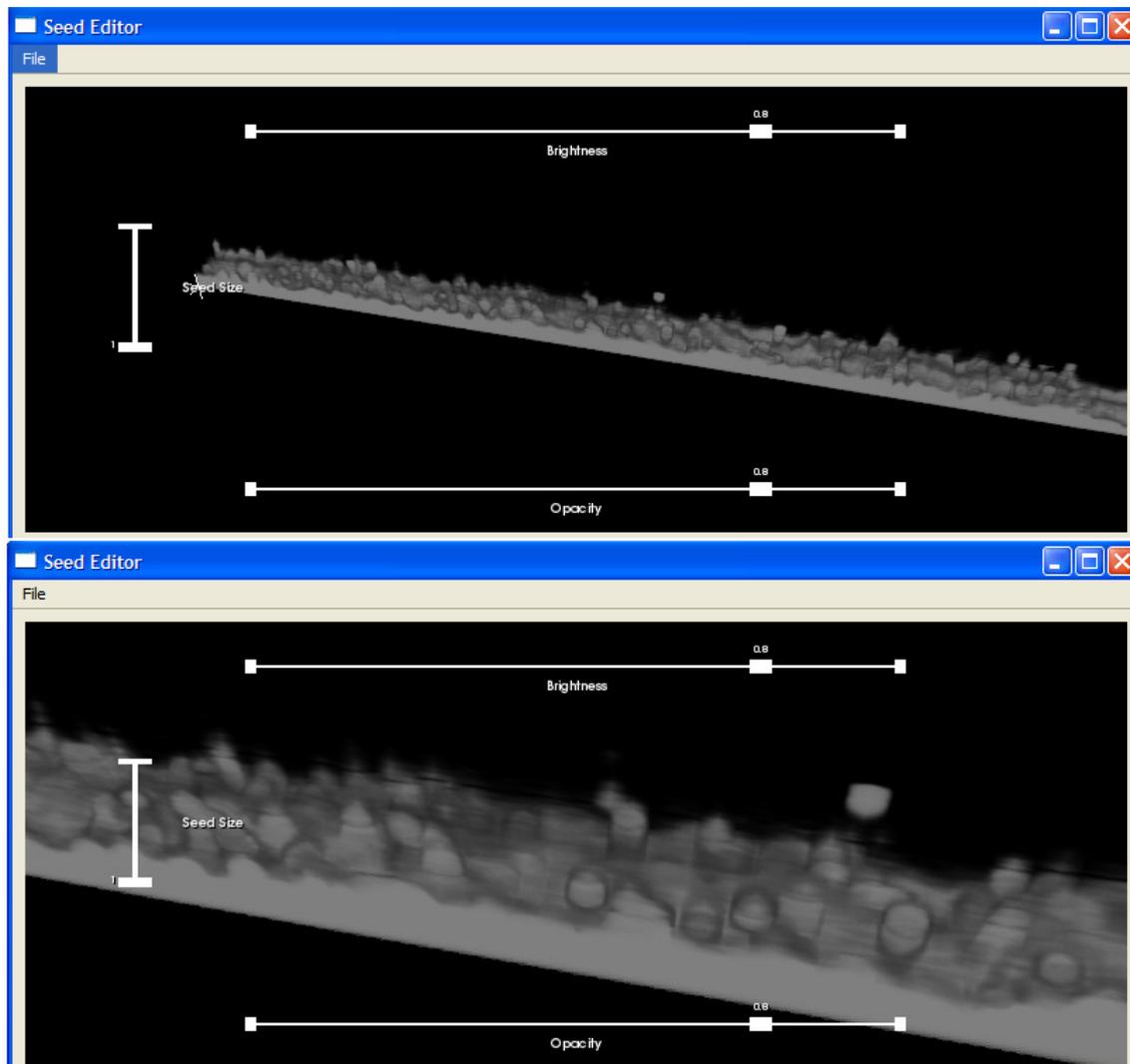


Figure 37 Dendrite flattening demonstration. Top shows a rendering of a flattened and straightened dendrite with the spines appearing to protrude upward from the surface. Bottom shows a close up view. The tool involves Cartesian to polar transformation and hence the resulting image is affected by rounding. Validation using this tool has yet to be proven useful.

7.4.5.2 FARSIGHT Edit-Based Validation System

In the making is a new tool within the FARSIGHT toolkit (Roysam) suite that inputs together with the 3D image a variable number of features for user inspection and validation. A desired number of confusion tables of feature couples can be opened at a time for selective editing of the detected objects. This will prove especially useful for false positive elimination when apparent correlation between features and image can be deduced from the synchronized views. A demonstration is shown below.

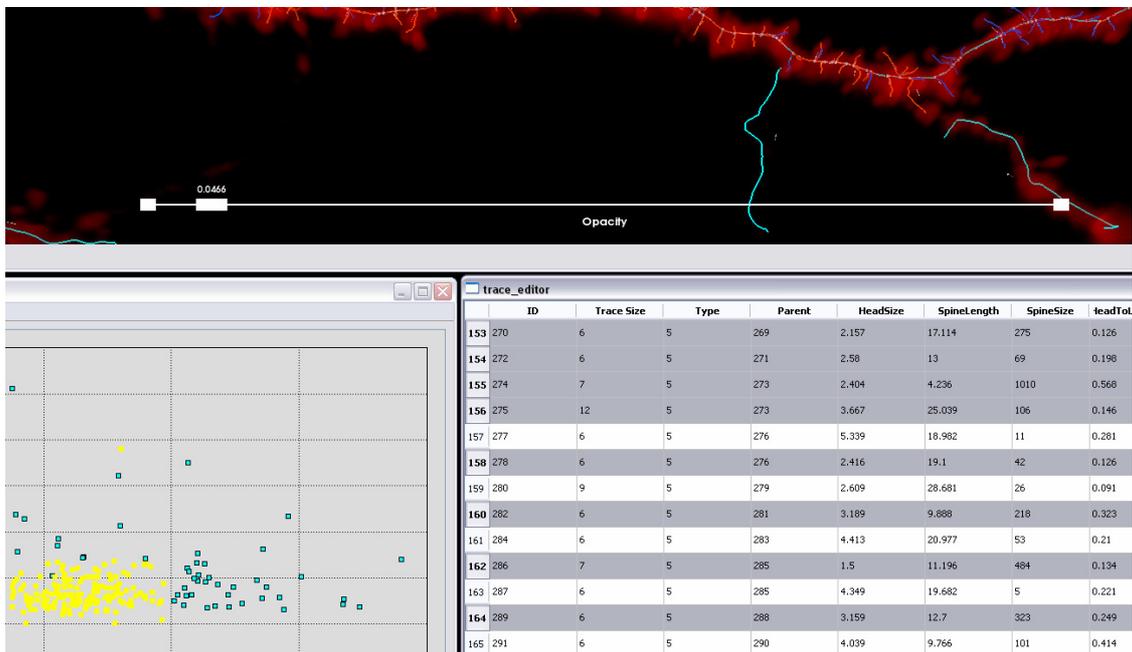


Figure 38 Group edit-based validation tool of the FARSIGHT toolkit suite. Upper panel: rendering of the dendrite with spine axes. Lower right: spine features. Left: confusion table of two features with a user-selected cluster highlighted in yellow and reflected synchronously with the feature and rendering windows.

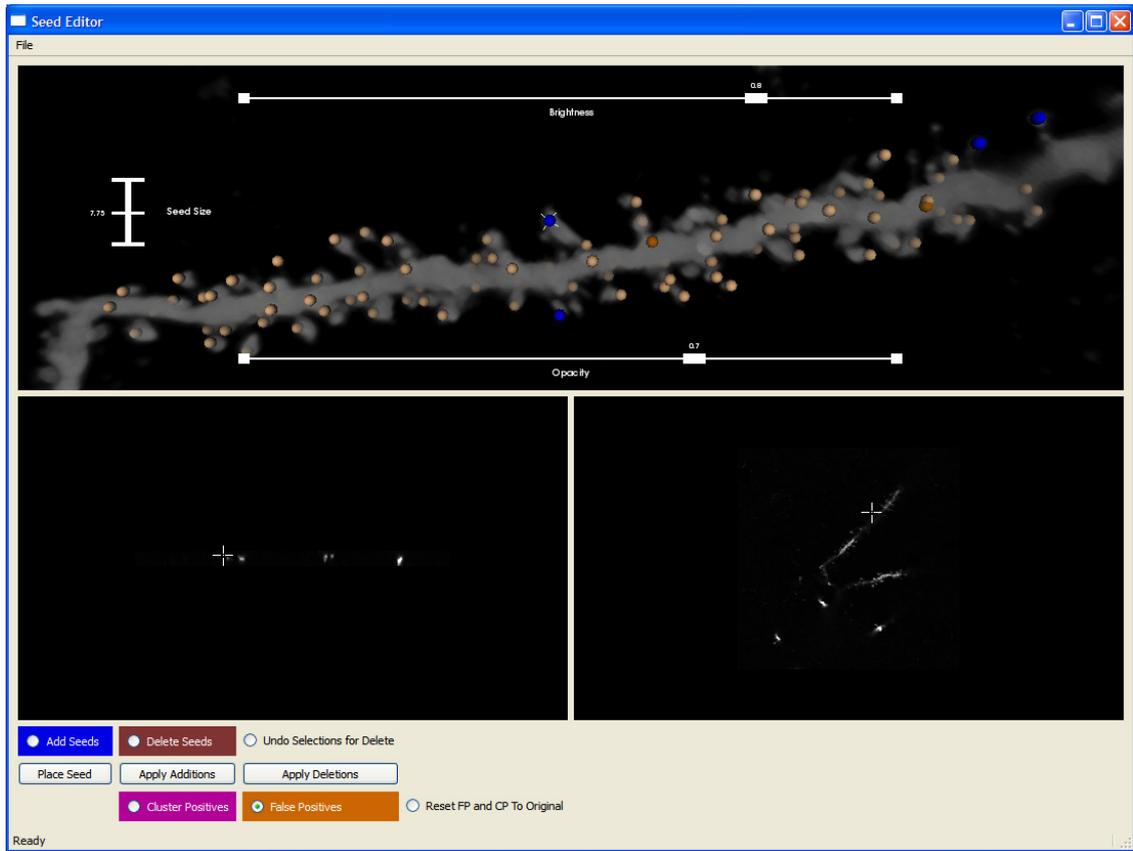


Figure 39 SeedEditor validation tool from FARSIGHT toolkit. The top panel shows a rendering of a dendrite segment with the original seeds in faint yellow. The newly added seeds in blue indicate false negatives and the dark orange seeds are false positives.

8. Summary

This work has been concerned with 3-dimensional grayscale confocal images obtained from diverse biological sources without restrictions on image resolution, quality or size. The imaging technique is not limited to the animal species, type of tissue, sample thickness, fluorescent marker or other conditions. Thus the image quality encountered herein is diverse and no prior information has been required for analysis.

Of the challenges we have encountered is the fact that spines are much smaller than dendrites causing imaging resolution to often restrict spine volume to a few voxels. Moreover, among the four mentioned spine types, there is a continuum of sizes, shapes, and warping. Therefore, complete spine segmentation from the original grayscale image is needed for classification.

In this thesis we presented a novel spine detection algorithm based on a robust model-based dendrite tracing. The tracing model was utilized to generate a series of annuli for thresholding the volume surrounding the dendrite surface. We called this the spine ring detector. Following a spine candidate generation from the detected foreground regions, we proposed two merging approaches in order to eliminate false positives. The more promising one was that based on spine-to-dendrite path extraction utilizing Fast Marching wavefront propagation. Spine candidates are validated and possibly merged together allowing for further feature extraction for classification. The results showed median error rates of 12% (false positive) and 4% (false negative) as well as robustness to image variability. The discussion highlighted different factors leading to spine segmentation errors. These include image dimensions (resolution and anisotropy ratio), imaging artifacts, biomarker expression, spine density and spine size.

A complete C++ open source code together with a 3D visualization and validation tool has been contributed to the FARSIGHT toolkit.

8.1 Main Contributions

- Fully automated spine segmentation algorithm
 - Non parametric detection, hypothesis test segmentation
 - Robust to image variability, noise, outliers and imaging artifacts
 - Connects unattached spines back to dendrite shaft
 - Finds spines that are hard to detect visually
 - Applying fast marching algorithm to spine validation and merging
 - Dendrite trace enhancement: An automated tool to fix the tracing results for neuronal dendrites with optional user-interaction.
-
- **Additional Contributions:**
 - 3D visualization and result verification tool extendible to other applications.
 - Open-Source C++ code using ITK as well as matlab version.

8.2 Future Work

Although the above results are very promising, some improvements are still needed, mainly in two areas: reducing false positives and de-clustering multiple spines. To that effect, further investigation is needed in spine feature extraction and validation. Although shape fitting is not easy to achieve because of spine variability, sub-accurate shape fitting or statistical spine shape models would help in validation and possibly classification.

How this tool compares with an MDL-based segmentation of Yuan (Yuan 2009) remains to be investigated.

We plan to work on a novel interactive validation approach based on multiple feature visualization. This enables the biologist to validate and group spine candidates according to common features. The FARSIGHT tool is a cornerstone of this effort because of its visualization versatility between the image and the feature spaces. We will develop the required specialized routines for validation and result-editing.

Appendix A

Maximum Likelihood Estimate of the center of the Laplacian distribution

Consider the i.i.d. random variable x with Laplacian distribution

$$f(x | \mu, \lambda) = \frac{1}{2\lambda} e^{-|x-\mu|/\lambda}, \lambda > 0, \quad (\text{A.1})$$

with center μ and scale λ . For n samples of x the likelihood function is the product of the individual probabilities:

$$\phi = \frac{1}{(2\lambda)^n} \exp\left(-\sum_{i=1}^n |x_i - \mu|/\lambda\right). \quad (\text{A.2})$$

To maximize ϕ with respect to μ ,

$$\frac{\partial \phi}{\partial \mu} = \frac{1}{(2\lambda)^n} \exp\left(-\sum_{i=1}^n |x_i - \mu|/\lambda\right) \cdot \frac{d\left(-\sum_{i=1}^n |x_i - \mu|/\lambda\right)}{d\mu} = 0.$$

Since the exponential is non-zero,

$$\frac{d\left(-\sum_{i=1}^n |x_i - \mu|/\lambda\right)}{d\mu} = \frac{\left(-\sum_{i=1}^n d|x_i - \mu|/\lambda\right)}{d\mu} = 0;$$

$$\frac{\sum_{i=1}^n d|x_i - \mu|}{d\mu} = \sum_j d(x_j - \mu)/d\mu - \sum_k d(x_k - \mu)/d\mu; x_j > \mu, x_k \leq \mu, j+k = n.$$

$$\sum_j 1 - \sum_k 1 = j - k = 0 \Rightarrow j = k = \left\lfloor \frac{n}{2} \right\rfloor, \text{ truncating the division to an integer.}$$

$\therefore \mu < x_{\lfloor n/2 \rfloor}$ and $\mu \geq x_{\lfloor n/2 \rfloor}$; hence μ is the sample median.

Appendix B

Convex Hull Intersection

This can be done by dividing the entire space into two regions: one inside P and the other outside P . In 3D each facet of P is a polygon on a plane that separates between two half-spaces, only one of which contains P . Thus P is the intersection of all half-spaces that contain it (which is an alternative definition of a convex hull). Consequently, a point x is in the interior of P if it belongs to the same half-spaces as P . An efficient approach is to utilize the sign of a dot product as an inside/outside indicator. The algorithm is described in the following pseudo code.

Table 12 Pseudo code for testing whether a point is inside a convex hull.

- Given a set of points S
- Compute its convex hull P using Quickhull (above).
- Compute the centroid s_0
- For every facet f_i with vertices $v_{i,1,2,3}$:
 - Compute the facet normal $n_i = v_{i,1} \times v_{i,2}$
 - Ensure n_i is oriented towards the interior of P :
 - Consider a ray from any point on facet to the hull centroid, say $(s_0 - v_{i,1})$
 - If $n_i \cdot (s_0 - v_{i,1}) < 0$ then $n_i := -n_i$
 - Check if x and P are on the same side of f_i :

If $n_i \cdot (x - v_{i,1}) < 0$ then x is outside P ; Stop

Appendix C

Implementation Issues

In the matlab implementation, performance is more critical than the C++ version. The connected components algorithm is slow for large 3D regions. Therefore we adopted a divide-and-conquer approach for detector merging such that it is done as soon as the first foreground discontinuity is encountered or when there becomes a long sequence of detectors waiting. This process is implemented using a dynamic FIFO queue. The queue is filled with the foreground pixels and is emptied at the breakage by merging its contents. The flowchart below explains the details.

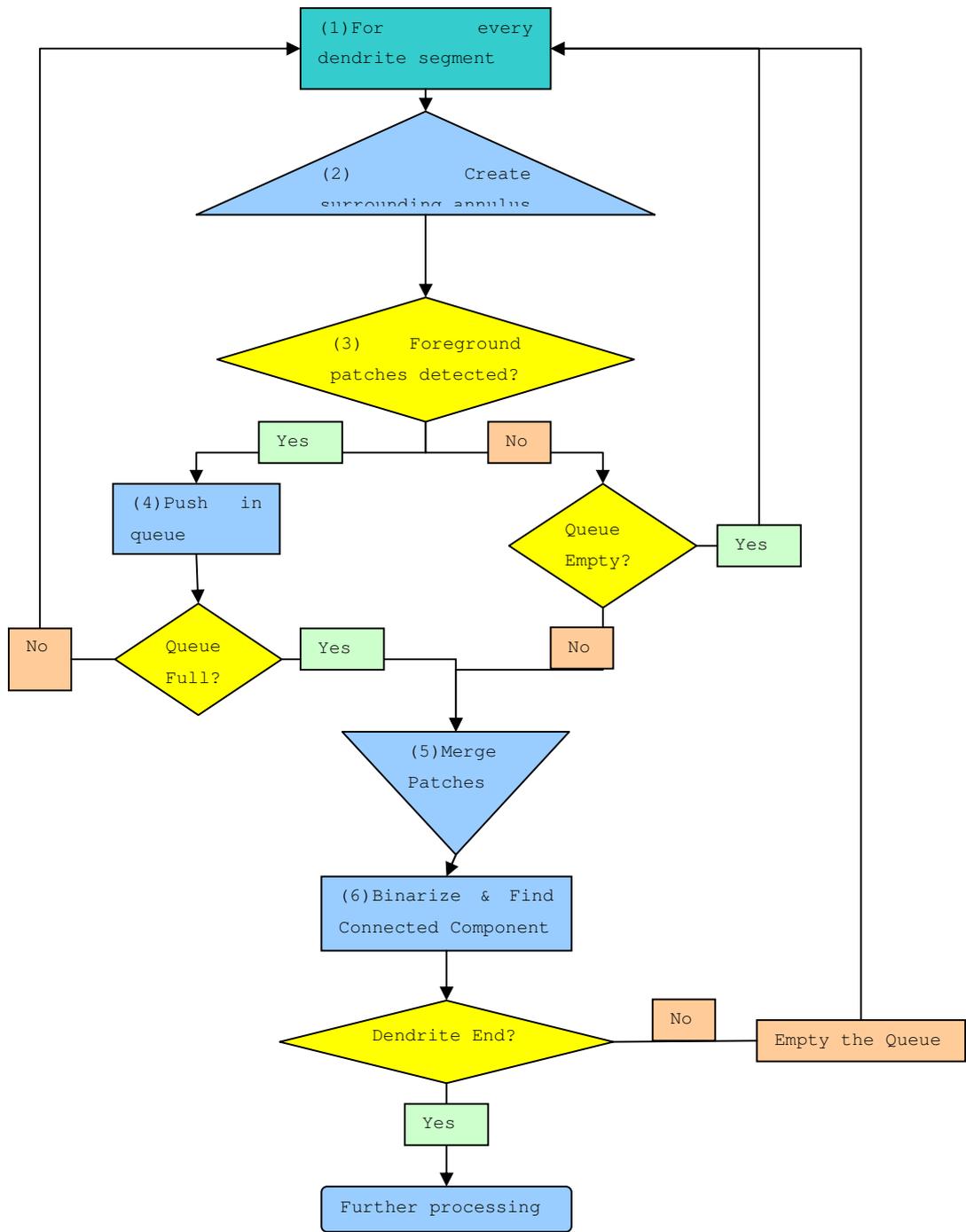


Figure 40 Spine Ring Detector implementation flowchart: This details boxes A and B of Figure 21. (1-3) Detectors are created and the contents thresholded. (4) A FIFO queue serves as a buffer for the latest sequence of detectors with hits. (5) When there is detector hit discontinuity, the queue is considered full and is emptied by merging the contents. (6) Connected components are labeled and validated for further processing.

Bibliography

- Akram, A., D. Christoffel, et al. (2007). "Stereologic estimates of total spinophilin-immunoreactive spine number in area 9 and the CA1 field: Relationship with the progression of Alzheimer's disease." Neurobiol Aging.
- Al-Kofahi, K. A., A. Can, et al. (2003). "Median-based robust algorithms for tracing neurons from noisy confocal microscope images." IEEE Trans Inf Technol Biomed **7**(4): 302-17.
- Arstikaitis, P., C. Gauthier-Campbell, et al. (2008). "Paralemmin-1, a modulator of filopodia induction is required for spine maturation." Mol Biol Cell **19**(5): 2026-38.
- Avis, D., D. Bremner, et al. (1997). "How good are convex hull algorithms?" Computational Geometry **7**(5-6): 265-301.
- Barber, C. B., D. P. Dobkin, et al. (1996). "The quickhull algorithm for convex hulls " ACM Transactions on Mathematical Software **22**(4): 469-483
- Bazin, P. L. and D. L. Pham (2005). "Topology correction using fast marching methods and its application to brain segmentation." Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv **8**(Pt 2): 484-91.
- Capani, F., M. E. Martone, et al. (2001). "Selective localization of high concentrations of F-actin in subpopulations of dendritic spines in rat central nervous system: a three-dimensional electron microscopic study." J Comp Neurol **435**(2): 156-70.
- Cardinal, M. H., J. Meunier, et al. (2006). "Intravascular ultrasound image segmentation: a three-dimensional fast-marching method based on gray level distributions." IEEE Trans Med Imaging **25**(5): 590-601.
- Cheng, J., X. Zhou, et al. (2007). "A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy." J Neurosci Methods **165**(1): 122-34.
- Cohen, A. R., B. Roysam, et al. (1994). "Automated tracing and volume measurements of neurons from 3-D confocal fluorescence microscopy data." J Microsc **173**(Pt 2): 103-14.
- Cohen, L. D. and T. Deschamps (2007). "Segmentation of 3D tubular objects with adaptive front propagation and minimal tree extraction for 3D medical imaging." Comput Methods Biomech Biomed Engin **10**(4): 289-305.
- Comery, T. A., J. B. Harris, et al. (1997). "Abnormal dendritic spines in fragile X knockout mice: maturation and pruning deficits." Proc Natl Acad Sci U S A **94**(10): 5401-4.
- DeFelipe, J., Ed. (1999). Cajal. The MIT encyclopedia of the cognitive sciences. Cambridge, Mass., MIT Press.
- Deschamps, T. and L. D. Cohen (2001). "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy." Med Image Anal **5**(4): 281-99.
- Deschamps, T., R. Malladi, et al. (2004). "Fast evolution of image manifolds and application to filtering and segmentation in 3D medical images." IEEE Trans Vis Comput Graph **10**(5): 525-35.

- Duan, H., S. L. Wearne, et al. (2003). "Age-related dendritic and spine changes in corticocortically projecting neurons in macaque monkeys." Cereb Cortex **13**(9): 950-61.
- Evers, J. F., S. Schmitt, et al. (2005). "Progress in functional neuroanatomy: precise automatic geometric reconstruction of neuronal morphology from confocal image stacks." J Neurophysiol **93**(4): 2331-42.
- Fiala, J. C. and K. M. Harris (1999). Dendrite Structure. Dendrites. G. Stuart, N. Spruston and M. Häusser. Oxford, UK, Oxford University Press.
- Garcia-Lopez, P., V. Garcia-Marin, et al. (2006). "Three-dimensional reconstruction and quantitative study of a pyramidal cell of a Cajal histological preparation." J Neurosci **26**(44): 11249-52.
- Goodman, J. E. and J. O'Rourke (2004). Handbook of discrete and computational geometry. Boca Raton, Chapman & Hall/CRC.
- Gray, A. (1997). **Modern Differential Geometry of Curves and Surfaces with Mathematica**. Boca Raton, FL, CRC Press: 292.
- Hamburger, V. (1980). "S. Ramon y Cajal, R. G. Harrison, and the beginnings of neuroembryology." Perspect Biol Med **23**(4): 600-16.
- Harris, C. and M. Stephens (1988). A combined corner and edge detector. 4th Alvey Vision Conference.
- Harris, K. M., F. E. Jensen, et al. (1992). "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**(7): 2685-705.
- Hering, H. and M. Sheng (2001). "Dendritic spines: structure, dynamics and regulation." Nat Rev Neurosci **2**(12): 880-8.
- Herzog, A., G. Krell, et al. (1997). Restoration of three-dimensional quasi-binary images from confocal microscopy and its application to dendritic trees. Three-Dimensional Microscopy: Image Acquisition and Processing IV, San Jose, CA, USA, SPIE.
- Herzog, A., R. Niese, et al. (2006). Detection of presynaptic terminals on dendritic spines in double labeling confocal images
Detection of presynaptic terminals on dendritic spines in double labeling confocal images. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on.
- Huber, P. J. (1965). "A robust version of the probability ratio test." Ann. Math.Statist. **36**(6): 1753-1758.
- Jbabdi, S., P. Bellec, et al. (2008). "Accurate anisotropic fast marching for diffusion-based geodesic tractography." Int J Biomed Imaging **2008**: 320195.
- Johnson, O. L. and C. C. Ouimet (2004). "Protein synthesis is necessary for dendritic spine proliferation in adult brain slices." Brain Res **996**(1): 89-96.
- Koh, I. Y., W. B. Lindquist, et al. (2002). "An image analysis algorithm for dendritic spines." Neural Comput **14**(6): 1283-310.
- Mahadevan, V., H. Narasimha-Iyer, et al. (2004). "Robust model-based vasculature detection in noisy biomedical images." IEEE Trans Inf Technol Biomed **8**(3): 360-76.
- Moening, C. and N. Dodgson (2003). Fast Marching Farthest Point Sampling. Computer Laboratory Technical Report 562, University of Cambridge.

- Nimchinsky, E. A., B. L. Sabatini, et al. (2002). "Structure and function of dendritic spines." Annu Rev Physiol **64**: 313-53.
- Osher, S. J. a. F., R.P. (2002). Level Set Methods and Dynamic Implicit Surfaces, Springer.
- Padfield, D. R., J. Rittscher, et al. (2006). Spatio-temporal cell cycle analysis using 3D level set segmentation of unstained nuclei in line scan confocal fluorescence images. Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on.
- Parnass, Z., A. Tashiro, et al. (2000). "Analysis of spine morphological plasticity in developing hippocampal pyramidal neurons." Hippocampus **10**(5): 561-8.
- Potter, S. M. (2005). Two-photon microscopy for 4D imaging of living neurons. Imaging in Neuroscience and Development: A Laboratory Manual. K. A. Yuste R, Cold Spring Harbor Laboratory Press: 59-70.
- Rodriguez, A., D. B. Ehlenberger, et al. (2008). "Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images." PLoS ONE **3**(4): e1997.
- Rodriguez, A., D. B. Ehlenberger, et al. (2006). "Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images." Nat Protoc **1**(4): 2152-61.
- Roysam, B. "Edit-based Validation System, FARSIGHT toolkit." from <http://farsight-toolkit.org/wiki/EVS>. (Last accessed 09/28/2009)
- Roysam, B. The FARSIGHT toolkit.
- Rusakov, D. A. and M. G. Stewart (1995). "Quantification of dendritic spine populations using image analysis and a tilting disector." J Neurosci Methods **60**(1-2): 11-21.
- Sermesant, M., Y. Coudiere, et al. (2005). "A fast-marching approach to cardiac electrophysiology simulation for XMR interventional imaging." Med Image Comput Comput Assist Interv Int Conf Med Image Comput Comput Assist Interv **8**(Pt 2): 607-15.
- Sethian, J. A. (1996). "A fast marching level set method for monotonically advancing fronts." Proc Natl Acad Sci U S A **93**(4): 1591-5.
- Sethian, J. A. (1999). Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science Cambridge University Press.
- Sethian, J. A. (2001). "Evolution, implementation, and application of level set and fast marching methods for advancing fronts " J. Comput. Phys. **169** (2): 503-555.
- Sharma, P. and R. Reilly (2002). Fast marching methods applied to face location in videophone applications using colour information. Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on.
- Shi, Y. and W. Karl (2005). A Fast Level Set Method Without Solving PDEs. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on.
- Shoemake, K. (1985). Animating rotation with quaternion curves, ACM. **19**: 245-254.
- Sidiropoulou, K., E. K. Pissadaki, et al. (2006). "Inside the brain of a neuron." EMBO Rep **7**(9): 886-92.

- Tizon, X., Q. Lin, et al. (2007). "Identification of the main arterial branches by whole-body contrast-enhanced MRA in elderly subjects using limited user interaction and fast marching." J Magn Reson Imaging **25**(4): 806-14.
- Trachtenberg, J. T., B. E. Chen, et al. (2002). "Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex." Nature **420**(6917): 788-94.
- Tyrrell, J. A., E. di Tomaso, et al. (2007). "Robust 3-D modeling of vasculature imagery using superellipsoids." IEEE Trans Med Imaging **26**(2): 223-37.
- Weaver, C. M., P. R. Hof, et al. (2004). "Automated algorithms for multiscale morphometry of neuronal dendrites." Neural Comput **16**(7): 1353-83.
- Wiseman, P. W., F. Capani, et al. (2002). "Counting dendritic spines in brain tissue slices by image correlation spectroscopy analysis." J Microsc **205**(Pt 2): 177-86.
- Xu, X., J. Cheng, et al. (2006). A shape analysis method to detect dendritic spine in 3D optical microscopy image. Biomedical Imaging: Macro to Nano, 2006, 3rd IEEE International Symposium on
- Yan, J., T. G. Zhuang, et al. (2004). "Lymph node segmentation from CT images using fast marching method." Comput Med Imaging Graph **28**(1-2): 33-8.
- Yuan, X. (2009). Complexity constrained 3D skeletonization algorithms for automated extraction of dendrites and spines from fluorescence confocal images. Dept. of Electrical, Computer, and Systems Engineering. Troy, Rensselaer Polytechnic Institute. **PhD**.
- Yuste, R. and T. Bonhoeffer (2001). "Morphological changes in dendritic spines associated with long-term synaptic plasticity." Annu Rev Neurosci **24**: 1071-89.
- Yuste, R. and T. Bonhoeffer (2004). "Genesis of dendritic spines: insights from ultrastructural and imaging studies." Nat Rev Neurosci **5**(1): 24-34.