

Automated 3-D detection of dendritic spines from *in vivo* two-photon image stacks

P. K. Singh · P. Hernandez-Herrera · D. Labate · M. Papadakis

Received: date / Accepted: date

Abstract Despite the significant advances in the development of automated image analysis algorithms for the detection and extraction of neuronal structures, current software tools still have numerous limitations when it comes to the detection and analysis of dendritic spines. The problem is especially challenging in *in vivo* imaging, where the difficulty of extracting morphometric properties of spines is compounded by lower image resolution and contrast levels native to two-photon laser microscopy. To address this challenge, we introduce a new computational framework for the automated detection and quantitative analysis of dendritic spines in vivo multi-photon imaging. This framework includes: (i) a novel preprocessing algorithm enhancing spines in a way that they are included in the binarized volume produced during the segmentation of foreground from background; (ii) the mathematical foundation of this algorithm, and (iii) an algorithm for the detection of spine locations in reference to centerline trace and separating them from

the branches to whom spines are attached to. This framework enables the computation of a wide range of geometric features such as spine length, spatial distribution and spine volume in a high-throughput fashion. We illustrate our approach for the automated extraction of dendritic spine features in time-series multi-photon images of layer 5 cortical excitatory neurons from the mouse visual cortex.

Keywords Image processing · Automated neural image segmentation · Automated dendritic spine detection · two-photon microscopy

1 Introduction

Dendritic spines are micrometer-sized protrusions occurring on the surface of dendrites. They appear in a variety of shapes, and undergo activity-dependent structural remodeling so that they can also appear and disappear over time. Neurophysiological studies have shown that morphological changes at the level of dendritic spines are highly correlated with cognitive function, including mechanisms of learning and memory. Spines play a critical role in neuronal information processing, as electrical signals from other neurons are transmitted to dendrites via synaptic gateways located at spines. Dendritic spine morphology is closely correlated with synaptic strength and experimental results show that abnormal spine morphology is prevalent in many brain disorders Penzes et al (2011). Therefore, there is a crucial need to efficiently detect spines and accurately quantify their morphological properties. In the present paper, we propose an algorithm for *the automated, accurate spine detection and count applicable to both high and lower resolution fluorescence microscopy images, acquired, even in vivo*. The proposed algorithm works *genuinely in 3D and is suitable for high throughput image analysis acquired with two-photon laser microscopy or other higher resolution microscopes*. Our main

P. K. Singh
Department of Mathematics,
University of Houston
Houston, TX, 77204-3008, USA
E-mail: pankaj@math.uh.edu

P. Hernandez-Herrera
Laboratorio de Imagenes y Vision por Computadora Instituto de
Biotecnologia Universidad Nacional Autonoma de Mexico Cuernavaca, Morelos, Mexico
D. Labate
Department of Mathematics
University of Houston
Houston, TX, 77204-3008, USA
E-mail: dlabate@math.uh.edu

M. Papadakis
Department of Mathematics
University of Houston
Houston, TX, 77204-3008, USA
E-mail: mpapadak@math.uh.edu

novelty is the introduction of a preprocessing step which acts as a directional filter of the input image sensitive to 3-D orientations other than the main centerline and perpendicular to spine necks. Thus, this step enhances details of the dendritic surface, indicative of the presence of spines. As part of this study, we discuss a model for intensity distribution in the vicinity of spines and using this model we mathematically justify why our image preprocessing enhances spines. The automation of the method is inherited from the multiscale one-class classification segmentation algorithm of Hernandez-Herrera et al (2016). This algorithm contains an automated method for selecting the training set that does not require human intervention. User guidance is only needed to set the threshold for the minimum spine length, so that they can be separated from small dendritic branches. As part of this paper, we also include a method for extracting and separating spines from dendritic branches so that we can compute spine volume and potentially other shape characteristics.

The ultimate goal of spine analysis is the discovery of the specific structure-function relationship linking spine morphology to synaptic strength Donohue and Ascoli (2011). The problem is even more challenging in in-vivo imaging, where the difficulty of extracting morphometric properties of spines is compounded by lower image resolution and higher noise levels and tissue motion due to blood circulation. Under those constraints several other researchers made numerous contributions in neuronal imaging and in the imaging of spines.

Digital representation of neuronal morphology has traditionally relied on manual or semi-manual tracing, e.g., NeuroLucida Glaser and Glaser (1990), AutoNeuron Meijering (2010); Donohue and Ascoli (2011), Imaris. However, this approach is not only labor-intensive and prone to subjective errors, but also very impractical for the analysis of volumetric data. NeuroLucida's AutoSpine MBF Bioscience (2011) only provides qualitative or limited quantitative information in reference to sub-neuronal structures Scorcioni R. (2008) including spines, and requires a significant manual intervention. Recently published research limits the use of semi-automated tools for spine detection to single branch sections using head measurements made with simple thresholding filters, implemented on an Imaris or AutoNeuron platform Swanger et al (2011). Spine populations are tracked globally within a region of interest and not on a spine-to-spine basis. As a consequence, other methods for spine detection and analysis were proposed in the literature in recent years. Existing automated methods include 2D algorithms Bai et al (2007); Cheng et al (2007), where the analysis is conducted on maximally intensity projection (MIP) images, that is, microscopy image voxels with maximum intensity are projected onto a 2D plane perpendicular to the rays of illumination. This method is released as the Neu-

ronIQ freeware. However, 2-D methods may miss a significant amount of information since spines that are orthogonal or near-orthogonal to the imaging plane may be lost in the projection. Even for the spines that are detected, such methods cannot guarantee accurate morphological reconstruction and spine tracking in general (two-dimensional methods may be suitable for cultured neurons since their dendrites develop primarily horizontally and the available resolution is higher than in two-photon microscopy). To overcome these drawbacks, a number of 3D-based algorithms were proposed:

In Rodriguez et al (2008), dendrites are segmented and candidate spine voxels are identified using a classification approach based on voxel clustering; spines are detected by clustering candidate spine voxels based on connectivity properties. It includes a routine to reattach spine stems to spine heads when spine neck is missed during segmentation and a Rayburst method to classify spines. It is applied to confocal image stacks and implemented in Neuronstudio.

In Janoos et al (2009) a method based on skeletonization, which extracts the backbone of the neuron and identifies the spines as the shorter lines of the skeletonized data, is proposed. The method is time consuming and not very reliable in detecting spines.

In Zhang et al (2010), dendrites are extracted using a centerline extraction-based approach. Preprocessing is applied to remove noise and improve image quality; gradient vector tracking and feature point detection are used to find the neuronal backbone and detect candidate spine locations; a method based on eigen-analysis (eigen-analysis of the Hessian matrix) is applied to feature points to detect spines and spines are segmented using the fast marching approach. It is applied to confocal image stacks.

In Li and Deng (2012) data are preprocessed with median filter (to remove noise) and top-hat filter (to correct uneven illumination); fuzzy C-mean clustering is used to cluster image into background, weak spines, dendrites with strong spines; marching cubes algorithm is used to reconstruct 3D surface of neuron. Rayburst sampling algorithm is used to compute the dendrite backbone. Spines are detected using a combination of geometric features (distance to dendrite backbone, surface curvature and orientation). They also propose a method to separate touching spines. It is applied to confocal image stacks.

In Shi et al (2014) dendrites are segmented using an adaptive thresholding method (as in Cheng et al (2007)), followed by morphological operators to fill possible holes and smooth boundaries. The backbone of the dendrites is extracted using a Rayburst-type approach; spines are detected as in Rodriguez et al (2008) by clustering candidate voxels from the tip points and then segmented according to the response of a wavelet-based filter. Spines are classified using semi-supervised learning.

Boundary characteristics of dendritic spine structures are used in He et al (2012a,b) to detect spines by finding their tips. Such a distinctive tip feature is identified using the minimal cross-sectional curvatures on the surface of the spine as it significantly larger than those on other locations. Spines are then segmented using a region growing strategy. It is applied to confocal image stacks.

Curvilinear structure detection methods are used in Fan et al (2009) to detect the medial axis of the dendritic arbor, followed by spine detection based on adaptive local binary fitting. This work presents some of the highest spine detection rates reported in the literature. Input images have been acquired with a multiphoton microscope. This work in addition presents a very interesting method for spine tracking in time-lapse images.

In summary, the main differences between these methods and ours is that we use binary segmentations of the dendritic volume including spines which is the basis of our inferences and essentially the elimination of the requirement for training which are particularly sensitive to variations of the dendritic surface. We entirely avoid denoising and intensity thresholding and instead introduce a spine enhancing preprocessing step. We contend that the special sensitivity of our algorithm can be attributed to the fact that the preprocessing step we apply acts as directional filtering which *automatically* sets the orientation sensitive to surface variations to be parallel to spines. Apart from this novel directionally sensitive preprocessing step, other contributions of our work are the development of:

1. An integrated algorithm which only requires minimal parameter tuning in order to automatically detect and count spines;
2. A mathematical model for fluorescent intensity distribution in the vicinity of spines. Although, our algorithm does not evaluate spine volume with high accuracy, it does give an estimate of it.
3. A natively 3-D method that works even for low resolution laser microscopy images, such as two-photon.

We envision that our work will provide a core tool for the automated, accurate 3-D analysis of fine structures in neurons that might facilitate the understanding of structure-function relationships at the base of synaptic plasticity.

2 Methods

In this paper we focus on data acquired *in vivo* using two-photon imaging.

Our spine detection and segmentation algorithm is organized according to the pipeline illustrated in Figure 1. After fluorescent image stacks are acquired, data sets are segmented to separate neurons from background after a preprocessing step which we discuss in detail in Subsection 2.1.

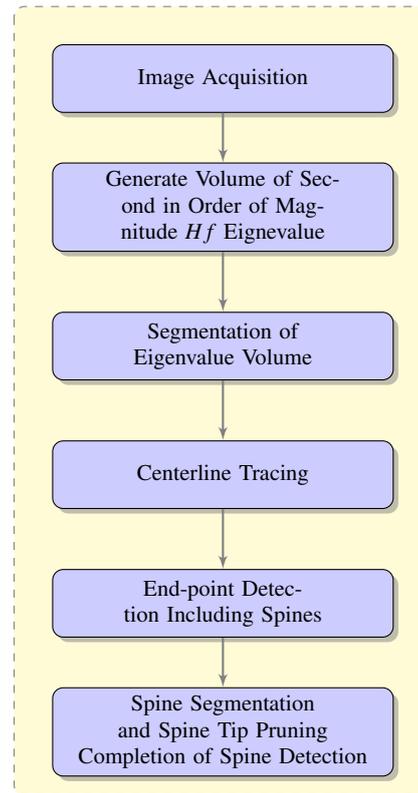


Fig. 1: Pipeline of the proposed automatic spine detection algorithm.

The segmented data are then processed to detect the end-points of the tubular structures and to extract the centerline. Using this information, the algorithm then proceeds to detect spines. A byproduct of the final step of spine detection is an estimation of their volume. In the following, we describe in detail each step of the algorithm and we discuss the mathematical justification of the image preprocessing step.

2.1 Image segmentation and preparation for spine detection

Several algorithms were proposed in the literature for the segmentation of fluorescent images of neuronal structures. These algorithms typically rely on the notion of tubular structures to represent the vessel-like systems found in the images of neurons. We must clarify that we do not refer here to centerline extraction algorithms for which it is sufficient only to detect the presence of a tubular structure (vesselness) but the the identification of the entire neuronal structure.

Neuronal segmentation algorithms use deterministic or probabilistic approaches to accomplish their goal. For example, Jiménez et al (2013); Jiménez et al (2015); Santamaria-Pang et al (2015) use a set of multiscale isotropic Laplacian filters to segment tubular structures. Other segmentation methods are based on probabilistic approaches, e.g. Schaap

et al (2007) where a Bayesian-based method for centerline extraction is proposed and, Agam and Wu (2005), where a probabilistic vessel enhancement method is developed; Tyrrell et al (2007) introduces a deformable framework based on super-ellipsoids for the segmentation of vessel-like structures.

The drawback of these tubular segmentation methods is that, while they effectively segment neurites and other vessel-like structures, they are usually not sufficiently sensitive to resolve fine scale structures on dendritic surfaces, such as dendritic spines. As mentioned above, fluorescent images of neurons and two-photon images especially are typically very noisy, and most neurite segmentation algorithms are not expected to resolve subdendritic structures and details of dendritic surfaces. To address this problem, we adopt a novel preprocessing step designed to enhance dendritic spines and other fine details of the dendritic surface as we explain below, before we even apply a neurite segmentation algorithm. The method on which this preprocessing step is based on is inspired by tubular structure detection eigenvalue-based methods originally proposed in Krisian et al (2000). This method was experimentally discovered and was first reported in the doctoral thesis of one us Herrera-Hernandez (2015). Here, we take this method one step further and provide a mathematical justification about why it works.

In a nutshell, the application of this method to a 3D image containing dendritic branches generates a new 3D image where small protrusions emanating from the dendritic surface are enhanced. As a first step, we filter the raw image with a suitable Gaussian 3D-isotropic filter in order to smooth it. We emphasize that the size of this Gaussian window must be fine enough to include the bandwidth in which spines live. Otherwise, spines and other fine surface structures will be truncated. The smoothness of the generated image enables the numerical computation of second-order partial derivatives which are needed to compute the Hessian of the input image. Our new derivative image is obtained by replacing the intensity of each voxel by the *second in order of magnitude eigenvalue of the Hessian matrix of the (smoothed) raw image*. Finally, on this new 3D derivative image we apply the segmentation algorithm proposed in Hernandez-Herrera et al (2014) to obtain the binary volume of the dendrite including its spines. This volume will be used for spine detection as described below in Subsection 2.2.

Next, we wish to explain the role in spine detection and spine enhancement of the second in order of magnitude eigenvalue of the Hessian matrix of the low-pass filtered raw image. To do so, we recall, that the Hessian matrix defines the second order differential of the filtered raw image f , hence it is the differential of ∇f . Consequently, if $Hf(x_0)$ is the

Hessian matrix of f at x_0 , then we must have

$$\lim_{v \rightarrow 0} \frac{\|\nabla f(x_0 + v) - \nabla f(x_0) - Hf(x_0)(v)\|}{\|v\|} = 0.$$

In this case $Hf(x_0)$ is a 3×3 matrix and $v \in \mathbb{R}^3$. To make everything work seamlessly, we need the assumption that all second order partial derivatives of f are continuous in a neighborhood of x_0 . Now, assume that the point x_0 is at the axis of a spine. Then it is reasonable to assume that the level surface of f containing x_0 is such that the axis of the spine coincides with the direction of the normal vector of the level surface at x_0 . Typical examples of such level surfaces arise from elliptical paraboloids. This model for intensity value distribution can also account for the anisotropy of the sampling grid known as z -smear. We assume that this is true as

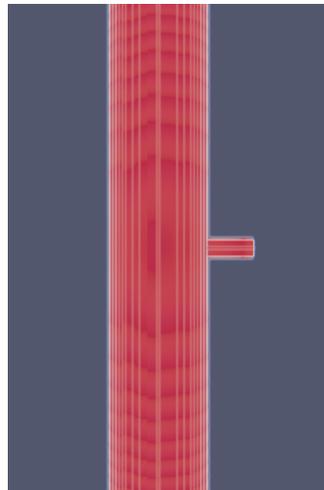


Fig. 2: A longitudinal view of the surface of the 3D synthetic dendritic branch and spine prototype model used to illustrate our theoretical analysis on the significance of the second in order of magnitude eigenvalue of Hf . The image has been produced by applying Gaussian blurring to the synthetic image of two cylinders, one representing a dendritic branch and the smaller one representing a spine neck.

x_0 traverses the axis of the spine, which we assume that it is a straight line segment. Recall that gradients are perpendicular to level surfaces. Our intensity spatial distribution model postulates that, when x_0 is close to the base of the spine neck, the gradient of the intensity f is relatively smaller than when x_0 has moved closer to the tip of the spine, because intensity drops gradually but faster when x_0 is closer to the tip of a spine. Figures 3 and 4 show how intensity varies inside a synthetic 3-D image of an idealized dendritic branch containing a spine .

According to our remarks above, if n is the unit vector parallel to the axis of the spine, then $\nabla f(x_0) = \iota(x_0 \cdot n)n$,

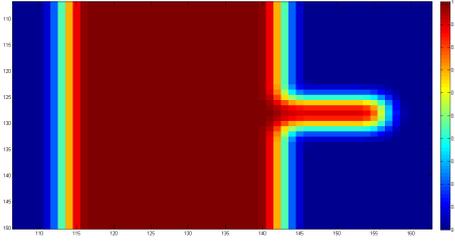


Fig. 3: Intensity values inside a longitudinal cross-section of the synthetic dendritic branch-spine 3-D image

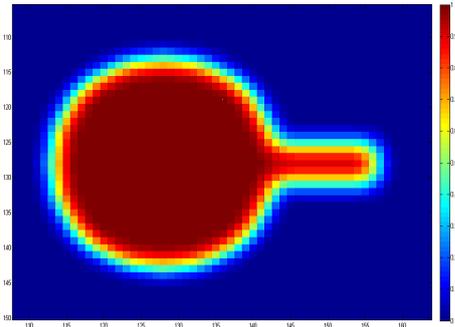


Fig. 4: Intensity values inside a coronal cross-section of the synthetic dendritic branch-spine 3-D image

where ι is a real-valued function defined exclusively on the spine axis. Here, $x_0 \cdot n$ is the coordinate of x_0 in reference to the spine axis (we can also use x_0 instead, but that should not obscure the fact that ι is an one-variable function only). Keep in mind that the gradient of f at x_0 and the normal vector n have opposite directions, because intensity is typically decreasing as x_0 traverses the spine axis toward its tip.

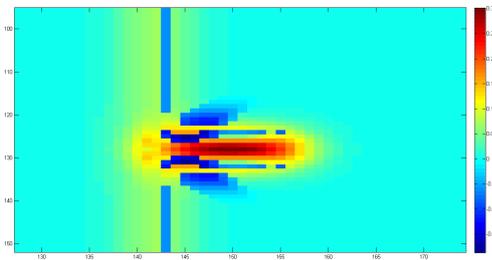


Fig. 5: A longitudinal cross-section of the the second in order of magnitude eigenvalue of Hf . Note how significant its values are along the axis of the spine just as our theoretical analysis predicts.

Now, if $v = hn$, where $h \in \mathbb{R}$, then

$$\begin{aligned} 0 &= \lim_{v \rightarrow 0} \frac{\|\nabla f(x_0 + v) - \nabla f(x_0) - Hf(x_0)(v)\|}{\|v\|} \\ &= \lim_{h \rightarrow 0} \left\| \frac{\nabla f(x_0 + hn) - \nabla f(x_0)}{h} - Hf(x_0)(n) \right\| \\ &= \left\| \lim_{h \rightarrow 0} \frac{\iota(x_0 \cdot n + h)n - \iota(x_0 \cdot n)n}{h} - Hf(x_0)(n) \right\|. \end{aligned}$$

Since f is twice differentiable, we have that ι is differentiable at $x_0 \cdot n$ and therefore

$$\begin{aligned} 0 &= \lim_{h \rightarrow 0} \left\| \frac{\nabla f(x_0 + hn) - \nabla f(x_0)}{h} - Hf(x_0)(n) \right\| \\ &= \|\iota'(x_0 \cdot n)n - Hf(x_0)(n)\|, \end{aligned}$$

which implies $\iota'(x_0 \cdot n)n = Hf(x_0)(n)$. Thus, n is an eigenvector of $Hf(x_0)$. So, the other two eigenvectors of $Hf(x_0)$ are both perpendicular to n . Our model postulates that when x_0 is close to the base of the spine neck, it is expected that intensity f around this point and along the spine axis decays slowly enough, so that the smallest eigenvalue of the Hessian of f at x_0 should correspond to the eigenvector n . The exact opposite should be true when x_0 is closer to the tip of a spine. Therefore, closer to the base of a spine neck or to the tip of a spine, the second in magnitude eigenvalue of $Hf(x_0)$ corresponds to an eigenvector which is perpendicular to the spine axis. This is also true at the branching points of the dendritic arbor. On the other hand, the 3D-image of the second in order of magnitude eigenvalue of Hf is the function defined by the matrix multiplication

$$x \rightarrow u(x)^T Hf(x)u(x),$$

where $u(x)$ is the unit eigenvector corresponding to the second in order of magnitude eigenvalue of $Hf(x)$. We remark that $u(\cdot)$ acts as a directional filter generating the 3D-image of the second in order of magnitude eigenvalue of Hf . At the neck or the tip of a spine, $u(\cdot)$ reorients itself automatically and perpendicularly to the spine neck, preferentially enhancing the spine neck and its tip. As x_0 slides on the spine axis in a direction opposite to the spine and inside the dendritic branch to which the spine neck is attached, intensity f level sets become cylindrical, oriented parallel to the branch axis. Thus, we can reasonably expect that the eigenvector corresponding to the smallest in magnitude eigenvalue of $Hf(x_0)$ (which is actually equal to zero) is parallel to the dendritic branch axis. However, n is always an eigenvector of $Hf(x_0)$ and as intensity decays faster inside the dendritic branch in an orientation other than n , we conclude, that the eigenvector corresponding to the second in order of magnitude eigenvalue of $Hf(x_0)$ is now parallel to n . In Fig. 5, the reader can see that close to the base of the spine the second in order of magnitude eigenvalue of $Hf(x_0)$ still gives some significant values, yet not as strong as on the spine axis which is inside

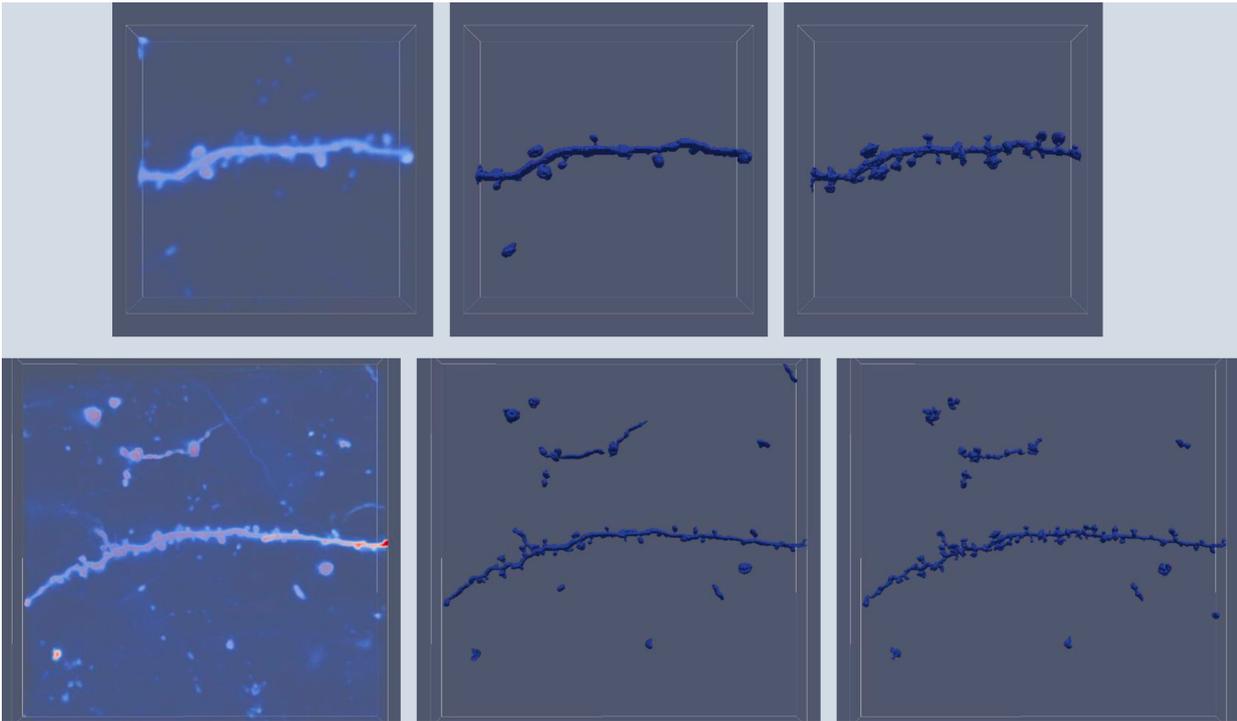


Fig. 6: A longitudinal cross-section of the the second in order of magnitude eigenvalue of Hf . Note how significant its values are along the axis of the spine just as our theoretical analysis predicts.

the spine. Any absolute value difference between the values of the second in magnitude eigenvalue of $Hf(x_0)$ in these neighboring locations of x_0 creates a contrast expected to enhance the detail of the spine neck. This type of enhancement cannot be achieved with the Laplacian of f . As it is equal to the sum of the eigenvalues of Hf , the Laplacian of f is always influenced by the eigenvalue of $Hf(x_0)$ with the highest absolute value, especially when the other eigenvalues are much smaller.

We remark that the 3D-image of the second in order of magnitude eigenvalue of Hf generates an input for the one-class classification segmentation algorithm in Hernandez-Herrera et al (2014) which considers the background as one class and the dendritic structure as the outlier (set-theoretic complement of the background class). Our experimental results verify that this combination of features and classification scheme can effectively incorporate in the binary volume fine-scale surface details, such as spines. We conjecture that the sensitivity of the system to spine presence relies on the fact that the 3D-image of the second in order of magnitude eigenvalue of Hf is the result of an essentially directional representation sensitive to the details of the dendritic surface.

We wish to emphasize that this preprocessing step is not intended for accurate spine segmentation. This can actually be verified from Fig. 5. The sole purpose of the preprocess-

ing step is to enable accurate and robust spine detection (see Subsection 2.2). In fact, our arguments predict that the second in order of magnitude eigenvalue of Hf does not necessarily have enough sensitivity to spine surfaces. The output of this preprocessing step is then used for volume segmentation, executed by a one-class classification segmentation algorithm proposed in Hernandez-Herrera et al (2014, 2016). This algorithm extracts global parameters for the segmentation. This forces the accuracy of the segmentation of a spine to not depend on local parameters but, instead, to depend on the global classification parameters derived for the entire volume by the segmentation algorithm of Hernandez-Herrera et. al. This algorithm is used twice, and since it provides automatic volume segmentation, we conclude that *our method for detecting spines inherits the same level of automatization*. In fact, in the experiment described in Table 3 below, we see how the automated extraction of the global classification parameters from the entire data volume required by the algorithm of Herrera-Hernandez *et.al*, Hernandez-Herrera et al (2016), influences the accuracy of spine detections.

We close this discussion by noting that the above mathematical model may seem restrictive since the spine axis is assumed to be perpendicular to the branch axis. We use this assumption to simplify the mathematical arguments of our analysis. The experimental results indicate that the applica-

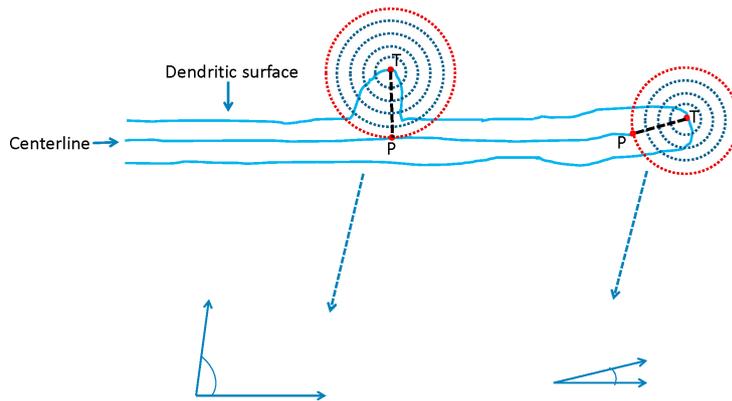


Fig. 7: Pictorial illustration of the final step of spine detection and identification: The right hand side of the figure is an example of dendritic endpoint and on the left hand side of the figure shows the way we extract the spine. This bounding sphere does not lead to an accurate spine segmentation, but it eliminates multiple spine tip designations on the surface of the same spine.

bility of the second in order of magnitude eigenvalue of Hf as a spine detector exceeds the domain of the above spine axis perpendicularity assumption, showing that this use of Hf gives a reliable spine detector.

In order to demonstrate pictorially why the second in order of magnitude eigenvalue of the Hessian of the filtered raw image enhances spines and visually illustrate the previous analysis, we generated a synthetic 3D image modeling a dendrite with a spine on it. One can easily see that the significant values of this eigenvalue (positive or negative) in a spine neighborhood occur close to the its neck or parallel to its axis (Fig. 5). With regards to the influence of scale of the smoothing Gaussian filter used to pretreat the raw image before computing the Hessian, we remark that, since spines can be considered as fine-scale image details, the use of multiple scales may not be effective to improve detection. In fact, we are not creating features to be used by a classification scheme for detecting spines, but instead we produce a new image in which spines are enhanced and we apply the binary segmentation code on that image in order to extract a binary volume of the dendritic arbor including its spines. In Fig. 6 we show how the second in order of magnitude eigenvalue of the Hessian of the filtered raw image enhances details of the dendritic surface including spines and helps the segmentation algorithm include them in the volume.

2.2 Dendrite centerline extraction and spine detection

Geometric analysis of 3D irregularly shaped objects is difficult. One of the widely used methods to analyze the geometry of objects is to determine and study their centerline. Several methods have been proposed for centerline extraction using the distance transform Bas and Erdogmus (2011); Koh (2001); Zhou and Toga (1999); Meijering (2010); Mor-

ison and Zou (2006). For this task, we use the algorithm developed by Jiménez et al (2015) to extract the centerline because it uses as input the binary volume of the dendritic arbor produced by the previous steps of our algorithm. Centerline extraction is important because it delivers the graph structure of the dendrite which serves as a natural coordinate system to track spines. First, we extract the centerline of the segmented volume and identify branching points and falsely identified minute branches. We use a simple length thresholding to prune unlikely branches and treat the rest of the centerline as the dendritic centerline.

Spine detection is carried out by identifying spine tip points after having extracted the dendritic centerline. Those points are identified by a voxel coding algorithm Zhou et al (1998); Zhou and Toga (1999) which, simply speaking, assigns a unique distance hierarchy rank from a given, manually selected, seed point for each voxel in the dendritic volume. Although this type of algorithm has the flavor of front-propagation or deformable model algorithms, voxel coding distance hierarchy assigns to each voxel an integer value indicating the discrete time step at which the algorithm reaches a certain voxel after initializing from a given seed point. Points with maximal rank whose every immediate neighbor has a smaller rank are classified as terminal points. Those points are either dendritic branch terminal points or spine tips. The distance of the latter from the closest centerline point is typically significantly smaller than the length of even small branches. Branch terminal points can be identified by their proximity to the end points of the extracted centerline. This criterion distinguishes spine from branch terminal points, thus completing the detection of spines. However, on each spine we may end up with more than one tip points, usually forming a cluster. This potential conflict is solved

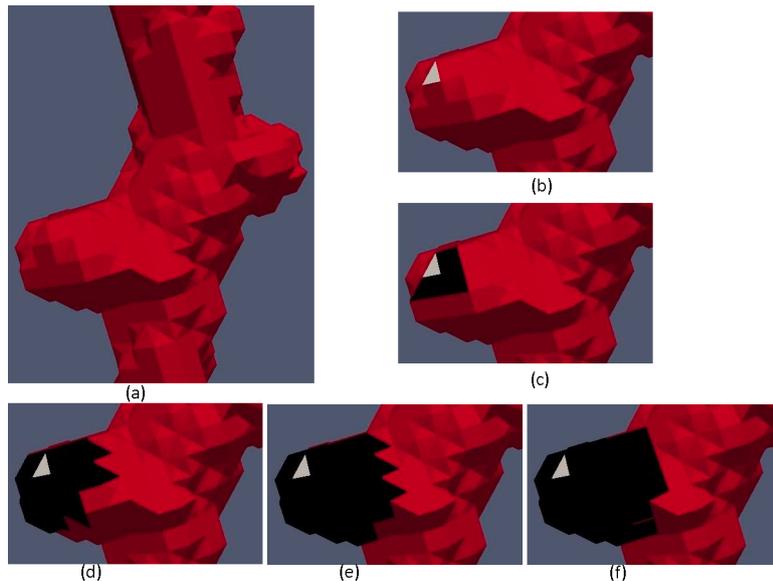


Fig. 8: 2D projected view of the steps of spine extraction: (a)detail of the region of interest; (b) tip voxel obtained shown on a particular spine; (c) region on the spine when the radius of the sphere is 2 units; (d) region on the spine when the radius of the sphere is 3 units; (e) region on the spine when the radius of the sphere is 4 units; (f) approximately extracted spine volume.

by using our spine segmentation algorithm described below which avoids multiple detections of the same spine.

2.3 Spine extraction

For each spine tip voxel T , a sequence $\{S_t\}_t$ of spheres with center T is generated (see figure 7 for a 2D illustration). Consider T fixed. The region of interest R_t^T around T is the intersection of the segmented volume V and the sphere S_t with center T . We find the radius of the sphere which intersects the main centerline for the very first time step t , say at the point P , and denote it by r_0^T . If the angle between the line segment PT and a small segment of the centerline containing P is relatively small, we classify such a tip voxel to be associated with a dendrite and not with a spine. Otherwise, the tip voxel T is classified as a spine tip point. In this case, we must identify the region of interest R_t containing all voxels belonging to the spine with T as its tip. Since the region of interest (ROI) is determined by the selection of S_t , we must find the optimal radius of the sphere centered at T which contains the minimum number of voxels not belonging to T . To this end, we estimate the local thickness of the dendrite r_1^T utilizing the centerline information. Thus, the ROI for the spine having T as a tip voxel is the intersection of the sphere with center T and radius $r^T = r_0^T - r_1^T$ and the spine is the connected component containing the ROI curved out using the voxel T as the center (see figure 8 for a projected view of the spine extraction procedure). Any other tip voxels be-

longing to this ROI will be ignored. So, the algorithm will proceed with spine tip voxels outside the previously identified ROI. The spine volume is roughly estimated by the number of voxels in this ROI. We do not claim that this is an accurate volumetric estimation process, since those ROIs are curved out by intersecting approximations of spheres in the digital domain with small radii with the binary volume of the dendritic arbor. However, it is reasonable to assume that each of these connected components corresponds to a single spine. Therefore, the number of the identified connected components is the total number of spines detected in the dendritic volume. Moreover, we can associate each spine with the point P on the centerline identified in the spine segmentation process and thus give a centerline coordinate to each spine which can be used for tracking spines. To avoid false detections, primarily due to noise we set a threshold for the minimum number of voxels in a spine. In our experiments we set this number equal to 10, which is a relatively small number given that spine axes range anywhere between 5 to 8 voxels minimum. So, below this threshold we anticipate not to have spines but imaging artifacts or random variations of the shape of dendritic branches.

3 Experiments and Results

Cranial windows are implanted in ketamine/xylazine anesthetized adult GFP-M transgenic mice (age at surgery, 80 – 100 days), which express enhanced GFP under the *thy-1* pro-

moter. The skull overlying the right visual cortex was removed and replaced with a cover-glass window, leaving the dura intact. Animals recovered from surgery for at least 30 days before imaging started. Live two-photon imaging was carried out using a custom-built microscope and a mode-locked Ti:sapphire laser (Mai Tai, Newport/Spectra Physics, Santa Clara, CA) at 912nm through a 40x water immersion objective (0.8 NA, Olympus, Tokyo, Japan). Scanning and image acquisition were controlled by ScanImage; the average power delivered to the brain was $< 50\text{ mW}$. Imaging was carried out at high resolution – 1024×1024 pixels, $0.08\ \mu\text{m}$ per pixel, $0.5\ \mu\text{m}$ z step size. We validated the performance of our method on the detection of spines using five data sets that we denote as Set 1, 2, 3, 4, 5. Sets 1 and 2 have sizes $469 \times 453 \times 38$, set 3 has size $512 \times 512 \times 96$. Volumes 1 through 3 were provided to us by Professor Tara Keck, of the MRC Center for Developmental Neurobiology of the King’s College, London, UK.

Data sets 4, 5 have also been acquired with a two-photon microscope. They are $512 \times 512 \times 90$ and $512 \times 512 \times 57$ image stacks respectively. Voxel sizes in both of them are $0.166 \times 0.166 \times 0.5\ \mu\text{m}^3$. These data sets were acquired by Dr. Mari Sajo of the Department of Psychiatry of the Mount Sinai Hospital in New York, NY with the purpose to compare spine turnover rate between WT mice and *Lynx1*KO mice Sajo et al (2016). Data sets 4 and 5 are from the WT mice. In this experiment Thy1-M line male mice (purchased from the Jackson Laboratory [B6.Cg-Tg (Thy1-EGFP) Mrs./J]), 7 months old. Image stacks are from cells in the visual cortex binocular zone, Layer 5 pyramidal neurons. Two-photon imaging was performed with a Prairie Technologies Ultima microscope and PrairieView software. All images were taken with $20\times$ water-immersion objective (Zeiss W Plan-APOCHROMAT, 1.0 numerical aperture). A mode-locked Ti:sapphire laser (Chameleon Ultra II; Coherent) was used to generate two-photon excitation. Dendritic spine images were acquired up to a depth of $100\ \mu\text{m}$ at a magnification of 6 zoom.

Evaluation of performance for the first four data sets differs from that of the fifth set. For all five sets automatically counted spine numbers were compared against manually counts. For volumes 1 through 4 we first performed a selection of subvolumes described in Subsection 3.2. Automatic spine counts were compared with manual spine counts in each of these subvolumes. The fifth volume was used in its entirety. Below, we discuss the details of spine counts.

3.1 Spine counts

Manual spine counts in a large 3D volume data set is highly impractical (in addition to being a tedious) task and is subjected to the non-uniform error propensity of the human expert who performs the count. To reduce the number of errors,

we decided to select randomly extracted non-overlapping sub-volumes from the entire set and perform the validation on each such subvolume. This trick statistically technically increases the number of data volumes we have, allowing us to extract estimates of accuracy which we cannot do if we use the five volumes we have in their entirety. A sample of five is not enough for estimation of detection accuracy. On the other hand, this estimation would be influenced by the inconsistent errors of the human operator. It is not hard to see that if there is an erroneous detection in a subvolume, the impact of this error in the detection accuracy is much higher when this error is counted in the subvolume instead of the entire volume. After algorithmic spine detections are completed, then we select the sub-volumes to carry out manual spine counts and compare them with those performed by our algorithm. We chose to perform the count of the fifth volume in its entirety to cross validate the performance estimates extracted from the subvolumes of data sets 1 through 4. For those volumes, we used the semi-automated tools Neuronstudio and Neurolucida to count spines and compare our spine detection rates with those of Neuronstudio and Neurolucida. We remark that the proposed algorithm’s accuracy rates (precision and recall) are the statistically the same with those obtained from the fifth volume. This implies that performance empirical success rates, although differently evaluated for the 5th volume, have not been influenced by the choice of using subvolumes to assess system performance.

3.2 Sub-volume selection

Here, we describe how we randomly selected the small sub-volumes from a given 3D image stack.

The goal is to select non-overlapping 3D boxes of the same size containing each a relatively reasonable number of spines, so that manual count is simpler to perform. In average, those 3D sub-volumes contain between 6-9 spines. In order to select such boxes, we first computed the 2D projection of the centerline perpendicularly to the z -direction. Then we randomly selected points on the centerline and computed boxes of size 70×70 with those points as their centers. We then sequentially discarded overlapping boxes until no more of such boxes existed. We eliminate overlapping boxes in order to maintain sampling independence. We also maintain equal subvolume sizes in order to avoid the variability of human count accuracy. By doing so, we selected a total of 63 boxes from the 4 data sets provided to us by Professor Tara Keck, of the MRC Center for Developmental Neurobiology of the King’s College, London, UK. Specifically the non-overlapping number of boxes are 21, 20, 11, 11 for the sets 1, 2, 3, 4, respectively. Figure 9 illustrates the process of the selection of these 3D sub-volumes. This figure shows the centerline and the randomly selected non-

overlapping square boundaries on the xy -plane projections of the selected sub-volumes from the four data sets.

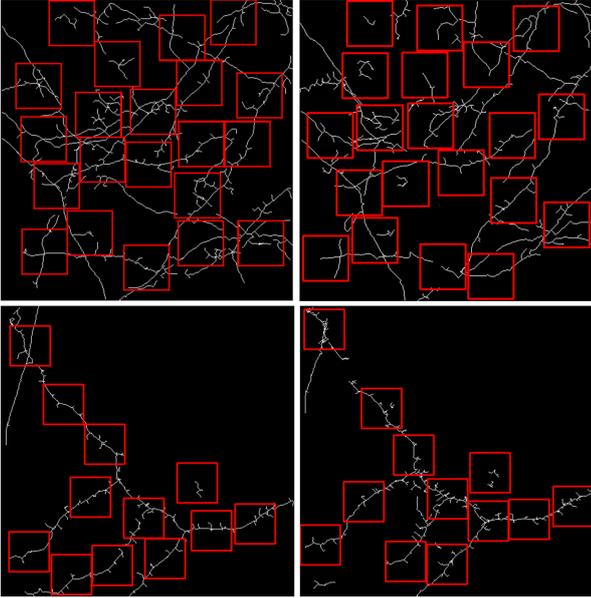


Fig. 9: **Non-overlapping boxes along centerline:** 2D projected view of the centerline and selection of sixty three random non-overlapping squares with their centers on the centerlines of the segmented volumes.

3.3 Statistical analysis

We assess the performance of our spine detection algorithm using the standard statistical metrics of *precision* and *recall*.

Precision is defined as the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. *Recall* is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision and recall are usually expressed in terms of percentages. More formally:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

where

- TP: True Positive, that is, there is a spine and we detect it as a spine
- FP: False Positive, that is, there is not a spine, but we detect it as a spine
- FN: False Negative, that is, there is a spine, but we do not detect it as a spine

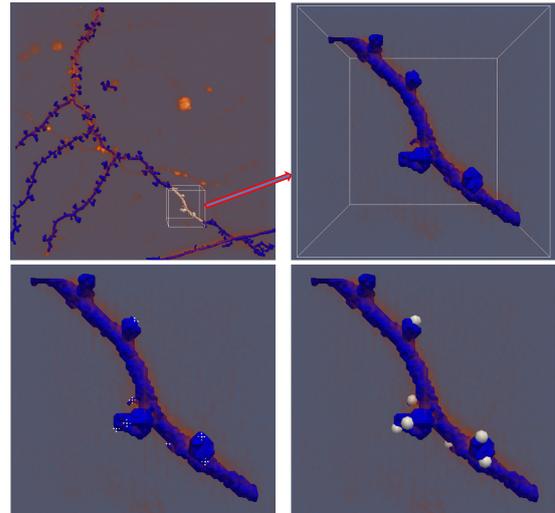


Fig. 10: **Sub-volume selection and spine detection in sub-volume:** A random point on the centerline is chosen and then a box of size $70 \times 70 \times n$ is drawn, where n is the number of z stacks. Top left, the entire volume along with a randomly selected sub-volume. Top right, zoom in the selected sub-volume. Bottom left: The detected terminal points by voxel scoring inside the sub-volume. Bottom right, Small spheres drawn at each of the detected spine associated endpoint. These small balls mark spines. **Note:** The sub-volume figures give the impression that there is a missed spine close to the top left corner. The terminal point associated with this spine is not shown because it belongs to part of the spine lying outside the selected sub-volume.

In the validation process, we ignore potential spine candidates that are very close to the boundaries of these sub-volumes. Usually, when we say ‘very close’ to the boundaries, we mean 1 to 3 voxels in length, but there are cases when we are 3 to 10 voxels away from the boundary and still cannot decide just by considering the sub-volume if a detected endpoint is on a spine or on a branch coming from outside the sub-volume. In the description of the method, we mention that dendritic protrusions whose axis forms a shallow angle with the dendritic centerline, below a certain angular threshold, should not be considered as spines. In our experiments, though, we did not apply this angular threshold.

The following table summarizes our experimental findings. For data sets 1-4, where we used the subvolume method to evaluate accuracy, we report the mean detection precision and recall from the randomly selected subvolumes from each data set.

To compare our method with other standard methods, we processed volume number 5 with NeuronStudio and NeuronStudio 360. Neuronstudio traces the centerline. When the

Data set No	Nr of Subvolumes	TP	FP	FN	Precision(%)	Recall(%)
1	21	139	13	14	91.45	90.85
2	20	97	15	12	86.60	88.99
3	11	82	4	6	94.95	94.00
4	11	94	5	6	95.35	93.18
5	1	64	8	3	88.9	95.50

Table 1: Accuracy of proposed algorithm on 63 small subvolumes and one entire image stack

voxel thickness and spine size range are provided (so the system knows the amount of z -smear) it finds 50 spines out of the 67 present in the volume. When the user chose isotropic voxels then Neuronstudio found only 29 spines. Neurolucida 360 performs in a satisfactory way and in par with our method, if the proper minimum and maximum spine heights are specified. Noise is another issue for Neurolucida. It also appears to be very sensitive to the specified range of length of spines, see Fig. 12. By varying those maximum and minimum lengths we detected without denoising a number of spines ranging from 42 up to 70, and after denoising of the raw data between 56 and 89 spines. We remark that volume number 5 has 67 spines per our expert manual analysis. In contrast to Neurolucida, our spine detection algorithm does not depend on the manual setting of their proper size range, but, only on the maximum spine length, above which a spine will be considered as a small dendritic branch. For our experiments, this threshold was set to 15 voxels.

We also performed one more experiment to compare the performance of Neurolucida with the proposed method for spine detection. In this experiment we used certain long dendrites with several spines obtained from the adult GFP-M transgenic mice. We used three dendritic segments from Volume 1, and 4 from Volume 3 above. The latter segments have no overlap with the 11 subvolumes of Volume 3 in Table 1. We stress that binary segmentation was performed using the herein proposed algorithm applied on each of the dendritic segments listed in Table 2. We selected 4 dendritic segments from Volume 3 above, from a mainly low intensity, small dynamic range region of this volume, in order to test the performance of Neurolucida 360 and of our algorithm in challenging conditions. Volume 6 in Table 2, comes from the dendrite imaged in Volume 2, in Table 1 above, but acquired at a different time instance. We chose this volume in order to enhance the variety of the experimental data set.

Stack 2, dendrites 1, 2, and Stack 3, dendrites 1, 2 show very low fluorescent intensity and small dynamic range (approximately 50). It seems that the low intensity yields a significant number of false positives for both NL 360 and the proposed method. However, the proposed method appears to be less prone to give big numbers of false positives in such extreme cases. In general, the average number of errors in spine counts is less for our method than for Neurolucida 360. In all of those counts we used for Neurolu-

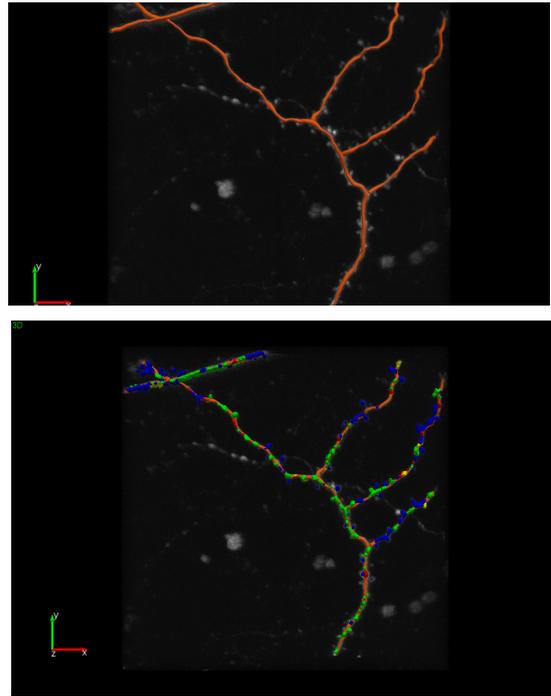


Fig. 11: Data set nr. 3 analyzed with Neurolucida. Upper image shows the intensity thresholded volume with the manually traced centerline. Lower image shows spine detections with different colors indicating different spine types.

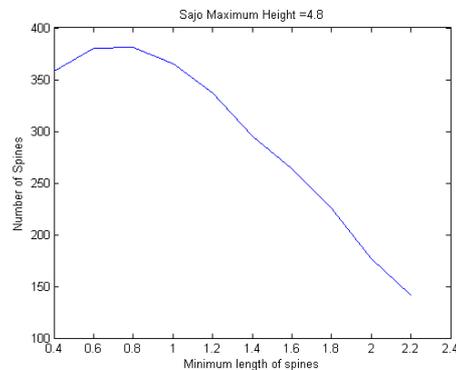


Fig. 12: Spine count monotonicity does not behave as anticipated. It initially increases as spine length range decreases

cida the same settings for spine head diameters (outer radius for spine detection, $2\mu\text{m}$, minimum spine detection radius $0.3\mu\text{m}$, minimum spine size 15 voxels; 100% sensitivity; image noise filter on). Those settings are dictated by the anatomy of the imaged neurons. Neurolucida spine detections were performed by our expert neurobiologist. These series of experiments revealed other very interesting aspects. First, Neurolucida 360 seems to be sensitive in the geometric settings required for spine detections. These need to be

Data set No	Expert's count	NL 360 count	PM	Errors PM	Errors NL 360
Stack 1, dend. 1	10	9	10	0	1
Stack 1, dend. 2	20	18	14	6	2
Stack 1, dend. 3	8	9	7	1	1
Stack 2, dend. 1	17	78	20	3	61
Stack 2, dend. 2	14	65	17	3	51
Stack 2, dend. 3	9	20	10	1	11
Stack 3, dend. 1	9	40	45	36	31
Stack 3, dend. 2	13	13	38	25	0
Stack 3, dend. 3	13	6	18	5	7
Stack 3, dend. 4	9	12	13	4	3

Table 2: Performance comparison of NeuroLucida 360 and proposed algorithm. *PM* abbreviates Proposed Method. Stack 2 dendrites 1, 2 and Stack 3, dendrites 1, 2 show very low fluorescent intensity and small dynamic range (approximately 50).

Data set No	Expert's count	NL 360 count	PM	Errors PM	Errors NL 360
Stack 3, dend. 1	9	40	31	22	31
Stack 3, dend. 2	13	13	26	13	0
Stack 3, dend. 3	13	6	17	4	7
Stack 3, dend. 4	9	12	13	4	3

Table 3: Performance comparison of NeuroLucida 360 and proposed algorithm on Stack 3. In Stack 3, dendrites 1, 2 show very low fluorescent intensity and small dynamic range (approximately 50). This time the binary segmentation used by the proposed method was performed on the entire stack and not in subvolumes as in the experiment whose results are shown in Table 2.

adjusted for each dendrite. We saw this in Stack 3 dendr. 3, which is thicker than what NeuroLucida 360 anticipated. This resulted in false negatives. On the other hand, our algorithm seems to be more robust, as it was not influenced by the variability of dendritic branch thickness. However, the proposed method seems to require the binary segmentation step performed according Hernandez-Herrera et al (2016), to be carried out on the entire image stack generated by the second in order of magnitude eigenvalues of the entire filtered volume. This happens because our algorithm requires a variety of training examples for the correct automatic classification of the background voxels. This segmentation step in the experiment using Stacks 1,2,3 summarized in Table 2, was performed locally in each of these subvolumes. When we performed this particular binary segmentation step using the entire image stack nr. 3 our numbers of spine detections significantly improved. The results are shown on Table 3. Figs. 13 and 14 shown the segmentation of dendrite nr. 1 of Stack 3 when the entire stack is used to extract the global settings for the binary segmentation of the entire stack. Notice how well the segmented volume overlays on the low intensity low dynamic range raw volume.

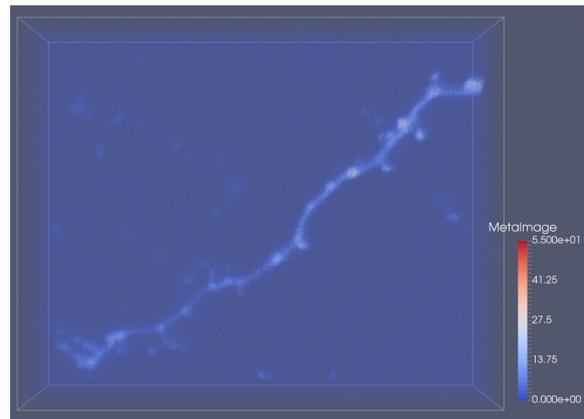


Fig. 13: Stack 3, Dendrite 1 in Table 2 shown as original max projection image along with color map shown the dynamic range of original fluorescent intensities

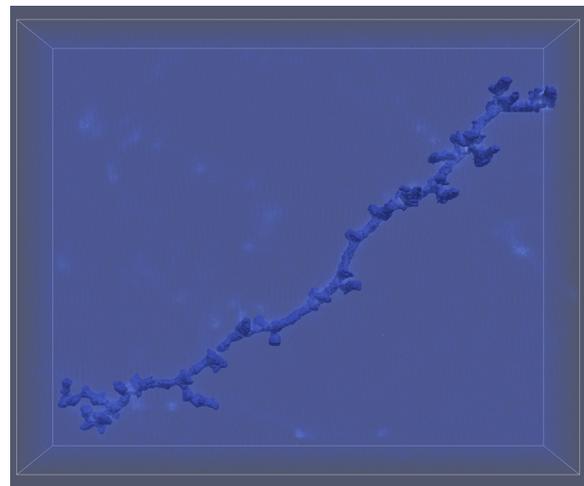


Fig. 14: Stack 3, Dendrite 1 in Table 2 shown with binary segmentation overlaid on the original image of the dendrite

4 Information Sharing Statement

We provide our entire source code written in Matlab at the link:
<https://github.com/pankajmath/SpineDetectionAndExtraction>

This link contains the binary segmentation, centerline tracing and spine detection source codes in Matlab. We also include a subvolume of test data set nr. 5 (in Table 1). The user can use it to test the code, as well as a trial demo and to verify that code is executed correctly. The reader should feel free to incorporate our code in their application/source code by giving the appropriate credits. Please, keep in mind that this source code has not been optimized for speed and use of RAM.

5 Discussion

We have introduced a novel automated algorithm for spine detection and 3D segmentation, and validated its performance on *in vivo* fluorescent images of neurons acquired using two-photon microscopy.

The numerical results reported in the paper show that the performance of our method is very competitive since precision and recall can average above 90%. The average precision and recall percentages are equal to 91.25 and 92.50, respectively, for all five test volumes considered (Table 1). These rates are among the highest reported in the literature for similar studies (see Blumer et al (2015); Zhang et al (2010); Fan et al (2009)). We also outperform two of the standard tools for spine detection (NeuronStudio and NeuroLucida 360, see Tables 1 and 2). However, this high average performance may drop when the fluorescent intensity is low with a relatively high dynamic range. We see this in Tables 2 and 3, where our average accuracy drops to 79%. In the same dataset NeuroLucida 360 becomes even more unstable. Below, we attempt to explain the drop of performance of our algorithm.

This proposed algorithm exploits the properties of the Hessian matrix to increase the prominence of structures emerging from the surface of tubular structures and on spine necks, in particular. Even though numerical evidence indicates that this property holds rather generally, we provide a theoretical justification of this enhancement approach only in the idealized case of spines which are perpendicular to the dendritic backbone. This initial detection step works very well unless the fluorescent intensity is low resulting in a low dynamic range. The extraction of the eigenvalues of the Hessian matrix requires that the fluorescent intensity function has some smoothness, otherwise the high local variability of intensity values may lead to false detections. This does not always happen as we see in Table 2. Moreover, a more detailed theoretical analysis, or even a different model, may be required to address the general case of fine-scale structures emerging from a tubular structure at a variety of angles, not only at 90 degrees and the cases where fluorescent intensity lacks smoothness. Perhaps, such a more general model may require other types of 3D filters beyond just the second in order of magnitude eigenvalue of the Hessian to perform feature extraction for spine detections.

With respect to other methods found in the literature, another advantage of our spine detection and segmentation algorithm is that it requires setting three parameters only: the minimum length of a dendritic branch, the minimum angle of a spine axis with respect to the axis of the dendritic branch to which the spine is attached to, and the minimum number of voxels in a spine. The latter threshold may affect the accuracy of detections but we as we see this happened only once in all of our experiments (Table 2, Stack 2, dendrite 2,

where the spine count increased by 2 when that threshold was dropped). The image segmentation step is fully automated as all training is carried out automatically. Hence, we can consider our method as an automatic method for spine detection, natively executed in 3-D. Our algorithm can process entire dendrites and not just selected segments as NeuroLucida 360, NeuronStudio and Imaris do, because in such software one needs to specify the anticipated size of spines and this quantity is not constant throughout a dendritic arbor. As a result, these three commonly used software tools require significant manual interventions to perform well, as also indicated by the numerical tests discussed previously. Among all three of them NeuroLucida 360, seems to require the least amount of human intervention. However, the requisite manual tracing of dendritic branches is an additional costly step which our algorithm does not require, making it possible to test images with several dendritic branches. In summary, the very small number of parameters the user must preset, the high level of automation of the entire process, make our algorithm make the proposed method suitable for high throughput 3D spine counts for entire dendrites and not just branches. If one wishes to observe spine populations in specific branches then, she (he) can use the metadata extracted by our algorithm which give locations of spines on each dendritic branch.

In contrast to the three competing algorithms mentioned above, our algorithm does not provide spine classification into stubby, filopodia and mushroom types. From conversations with domain experts we found that many neuroscientists disagree with classification into rigid spine type classes. The segmented spine volumes computed by our algorithm can be easily processed to generate parameters associated to their shape properties.

6 Acknowledgments

The authors want to thank Professor Tara Keck of the MRC Centre for Developmental Neurobiology, King's College, London, UK and Dr. Mari Sajo, of the Icahn School of Medicine at Mount Sinai for providing us with data sets and guidance for the experimental verification of our method. Without their help this work would never have come to realization. Many thanks to Dr. Ryan Ash, MD, of the Baylor College of Medicine for this help with manual validations and NeuroLucida 360 and for sharing with us his insight on spines. Also many thanks have to go to our doctoral student Mr. Nikolaos Karantzas for helping us with the validation experiments. Finally, we thank Professor Ioannis Kakadiaris of the University of Houston for several helpful discussions on 3-D volume segmentation.

This work was partially supported by NSF-DMS 1320910 and by a GEAR 2015 grant awarded by the Division of Research of the University of Houston and by a CONACYT

graduate scholarship. Last, but, by no means least, we express our warm thanks to the reviewers who gave us insightful comments and suggestions to improve the original manuscript.

References

- Agam G, Wu C (2005) Probabilistic modeling based vessel enhancement in thoracic ct scans. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, IEEE, vol 2, pp 649–654
- Bai W, Zhou X, Ji L, Cheng J, Wong ST (2007) Automatic dendritic spine analysis in two-photon laser scanning microscopy images. *Cytometry Part A* 71(10):818–826
- Bas E, Erdogmus D (2011) Principal curves as skeletons of tubular objects. *Neuroinformatics* 9(2-3):181–191
- Blumer C, Vivien C, Genoud C, Perez-Alvarez A, Wiegert JS, Vetter T, Oertner TG (2015) Automated analysis of spine dynamics on live cal pyramidal cells. *Medical image analysis* 19(1):87–97
- Cheng J, Zhou X, Miller E, Witt RM, Zhu J, Sabatini BL, Wong ST (2007) A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *Journal of neuroscience methods* 165(1):122–134
- Donohue DE, Ascoli GA (2011) Automated reconstruction of neuronal morphology: An overview. *Brain Research Reviews* 67(1–2):94 – 102, DOI <http://dx.doi.org/10.1016/j.brainresrev.2010.11.003>, URL <http://www.sciencedirect.com/science/article/pii/S0165017310001293>
- Fan J, Zhou X, Dy JG, Zhang Y, Wong ST (2009) An automated pipeline for dendrite spine detection and tracking of 3d optical microscopy neuron images of in vivo mouse models. *Neuroinformatics* 7(2):113–130
- Glaser JR, Glaser EM (1990) Neuron imaging with neuroLucida — a pc-based system for image combining microscopy. *Computerized Medical Imaging and Graphics* 14(5):307 – 317, DOI [http://dx.doi.org/10.1016/0895-6111\(90\)90105-K](http://dx.doi.org/10.1016/0895-6111(90)90105-K), URL <http://www.sciencedirect.com/science/article/pii/089561119090105K>, progress in Imaging in the Neurosciences Using Microcomputers and Workstations
- He T, Xue Z, Kim Y, Wong ST (2012a) Three-dimensional dendritic spine detection based on minimal cross-sectional curvature. In: 2012 9th IEEE International Symposium on Biomedical Imaging (ISBI), IEEE, pp 1639–1642
- He T, Xue Z, Wong ST (2012b) A novel approach for three dimensional dendrite spine segmentation and classification. In: SPIE Medical Imaging, International Society for Optics and Photonics, pp 831,437–831,437
- Hernandez-Herrera P, Papadakis M, Kakadiaris IA (2014) Segmentation of neurons based on one-class classification. In: Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on, pp 1316–1319, DOI 10.1109/ISBI.2014.6868119
- Hernandez-Herrera P, Papadakis M, Kakadiaris IA (2016) Multi-scale segmentation of neurons based on one-class classification. *J Neurosci Methods* 266:94–106, DOI 10.1016/j.jneumeth.2016.03.019, URL <http://dx.doi.org/10.1016/j.jneumeth.2016.03.019>
- Herrera-Hernandez P (2015) 3-d morphology of neurons. Phd, University of Houston
- Janoos F, Mosaliganti K, Xu X, Machiraju R, Huang K, Wong ST (2009) Robust 3d reconstruction and identification of dendritic spines from optical microscopy imaging. *Medical image analysis* 13(1):167–179
- Jiménez D, Papadakis M, Labate D, Kakadiaris IA (2013) Improved automatic centerline tracing for dendritic structures. In: Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on, IEEE, pp 1050–1053
- Jiménez D, Labate D, Kakadiaris IA, Papadakis M (2015) Improved automatic centerline tracing for dendritic and axonal structures. *Neuroinformatics* 13(2):227–244, DOI 10.1007/s12021-014-9256-z, URL <http://dx.doi.org/10.1007/s12021-014-9256-z>
- Koh YY (2001) Automated recognition algorithms for neural studies. PhD thesis, Citeseer
- Krissian K, Malandain G, Ayache N, Vaillant R, Trousslet Y (2000) Model based detection of tubular structures in 3d images. *Computer Vision and Image Understanding* 80(2):130–171
- Li Q, Deng Z (2012) A surface-based 3-d dendritic spine detection approach from confocal microscopy images. *Image Processing, IEEE Transactions on* 21(3):1223–1230, DOI 10.1109/TIP.2011.2166973
- MBF Bioscience (2011) Autospine. URL <http://www.mbfbioscience.com/autospine>
- Meijering E (2010) Neuron tracing in perspective. *Cytometry Part A* 77(7):693–704
- Morrison P, Zou JJ (2006) Skeletonization based on error reduction. *Pattern recognition* 39(6):1099–1109
- Penzes P, Cahill ME, Jones KA, VanLeeuwen JE, and KMW (2011) Dendritic spine pathology in neuropsychiatric disorders. *Neural Neuroscience* 14:285–293
- Rodriguez A, Ehlenberger DB, Dickstein DL, Hof PR, Wearne SL (2008) Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PloS one* 3(4):e1997
- Sajo M, Ellis-Davies G, Morishita H (2016) Lynx1 limits dendritic spine turnover in the adult visual cortex. *Journal of Neuroscience* 36(36):9472–9478
- Santamaria-Pang A, Hernandez-Herrera P, Papadakis M, Saggau P, Kakadiaris IA (2015) Automatic mor-

- phological reconstruction of neurons from multiphoton and confocal microscopy images using 3d tubular models. *Neuroinformatics* 13(3):297, DOI 10.1007/s12021-014-9253-2, URL <http://dx.doi.org/10.1007/s12021-014-9253-2>
- Schaap M, Manniesing R, Smal I, Van Walsum T, Van Der Lugt A, Niessen W (2007) Bayesian tracking of tubular structures and its application to carotid arteries in cta. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, Springer, pp 562–570
- Scorcioni R, AG Polavaram S (2008) L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nature Protocols* 3(5):866–876, DOI doi:10.1038/nprot.2008.51
- Shi P, Huang Y, Hong J (2014) Automated three-dimensional reconstruction and morphological analysis of dendritic spines based on semi-supervised learning. *Biomed Opt Express* 5(5):1541–1553
- Swanger SA, Yao X, Gross C, Bassell GJ (2011) Automated 4d analysis of dendritic spine morphology: applications to stimulus-induced spine remodeling and pharmacological rescue in a disease model. *Molecular brain* 4(1):1
- Tyrrell JA, di Tomaso E, Fuja D, Tong R, Kozak K, Jain RK, Roysam B (2007) Robust 3-d modeling of vasculature imagery using superellipsoids. *Medical Imaging, IEEE Transactions on* 26(2):223–237
- Zhang Y, Chen K, Baron M, Teylan MA, Kim Y, Song Z, Greengard P, Wong ST (2010) A neurocomputational method for fully automated 3d dendritic spine detection and segmentation of medium-sized spiny neurons. *Neuroimage* 50(4):1472–1484
- Zhou Y, Toga AW (1999) Efficient skeletonization of volumetric objects. *Visualization and Computer Graphics, IEEE Transactions on* 5(3):196–209
- Zhou Y, Kaufman A, Toga AW (1998) Three-dimensional skeleton and centerline generation based on an approximate minimum distance field. *The Visual Computer* 14(7):303–314