HAL open science

# SpineJ: A software tool for quantitative analysis of nanoscale spine morphology

Florian Levet, Jan Tønnesen, U. Valentin Nägerl, Jean-Baptiste Sibarita

## To cite this version:

Florian Levet, Jan Tønnesen, U. Valentin Nägerl, Jean-Baptiste Sibarita. SpineJ: A software tool for quantitative analysis of nanoscale spine morphology. Methods, 2020, 174, pp.49 - 55. 10.1016/j.ymeth.2020.01.020 . hal-03489915

**HAL Id: hal-03489915**

**https://hal.science/hal-03489915v1**

Submitted on 22 Aug 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Version of Record: https://www.sciencedirect.com/science/article/pii/S1046202319302373
Manuscript_ca375acbc1c838b9faf23bb0a4acd22b

# SpineJ: A software tool for quantitative analysis of nanoscale spine morphology

Florian Levet[1-5], Jan Tønnesen[1, 2#], U. Valentin Nägerl[1, 2*] and Jean-Baptiste Sibarita[1, 2*]

[1]Interdisciplinary Institute for Neuroscience, University of Bordeaux, Bordeaux, France
[2]Interdisciplinary Institute for Neuroscience, CNRS UMR 5297, Bordeaux, France
[3]Bordeaux Imaging Center, University of Bordeaux, Bordeaux, France
[4]Bordeaux Imaging Center, CNRS UMS 3420, Bordeaux, France
[5]Bordeaux Imaging Center, INSERM US04, Bordeaux, France
[#]Present address: Achucarro Basque Center for Neuroscience, Bilbao, Spain
[*]Corresponding authors

## Abstract

Super-resolution microscopy provides diffraction-unlimited optical access to the intricate morphology of neurons in living brain tissue, resolving their finest structural details, which are critical for neuronal function. However, as existing image analysis software tools have been developed for diffraction-limited images, they are generally not well suited for quantifying nanoscale structures like dendritic spines. We present SpineJ, a semi-automatic ImageJ plugin that is specifically designed for this purpose. SpineJ offers an intuitive and user-friendly graphical user interface, facilitating fast, accurate, and unbiased analysis of spine morphology.

## Introduction

Dendritic spines are the postsynaptic component of most excitatory synapses in the mammalian brain. Highly recognizable in light microscopic images as protrusions covering the membrane surface of dendrites, they serve as a convenient morphological proxy for synapses (Sala and Segal, 2014).

Spine structure and synapse function are closely related; large spine heads contain larger postsynaptic densities (PSD) and form synapses that have a higher conductance, while activity-dependent synaptic plasticity is associated with changes in the number, size and shape of dendritic spines. Hence, spine morphology and its dynamics can give important indications about the functional state and adaptations of synapses and neuronal circuits associated with brain development, learning and memory (Yuste and Bonhoeffer, 2004; Holtmaat and Svoboda, 2009) and brain disorders (Montagna et al., 2017).

However, dendritic spines feature nanoscale structural details, such as spine necks, which are functionally very important but cannot be properly resolved by conventional optical techniques like confocal and two-photon microscopy. Ranging in length between 0.2 and 2 micrometers with diameters smaller than 200 nm, spine necks have traditionally defied geometric analyses in live tissue.

Given this technical limitation, researchers have instead used calibrated fluorescence measurements to estimate spine size. This is in principle possible because the spatially integrated fluorescence signal scales with the volume of the source of the fluorescence, even if the underlying compartment cannot be spatially resolved (Svoboda et al., 1996; Matsuzaki et al., 2001). However, the problem is that this method is indirect and requires measurements of absolute fluorescence intensity, which is sensitive to potential confounders like fluorescence bleaching, variations in laser power, and the presence of organelles or other structures in the spines (Lenz and Tonnesen, 2019). While electron microscopy readily provides the required spatial resolution for direct geometric measurements, its workflow is very time-consuming, subject to fixation artifacts and incompatible with live-cell analysis.



© 2020 published by Elsevier. This manuscript is made available under the CC BY NC user license
https://creativecommons.org/licenses/by-nc/4.0/

By permitting live-cell imaging with nanoscale resolution, super-resolution fluorescence microscopy is capable of providing new insights into the functional properties of spines (Tønnesen and Nägerl, 2013). Among the new techniques increasingly adopted by neuroscientists, stimulated emission depletion (STED) microscopy is particularly well suited for imaging cellular morphology and has been successfully applied to study the structural dynamics of spines in brain slices (Nägerl et al., 2008; Tønnesen et al., 2014) and *in vivo* (Berning et al., 2012; Willig et al., 2014; Pfeiffer et al., 2018). It is a volumetric, laser-scanning imaging technique with high optical sectioning, just like confocal or 2-photon microscopy, which works with the same organic dyes and fluorescent proteins as these well-established techniques.

Several software packages are available for analyzing neuronal morphology, including dendritic spines. While these programs can determine spine densities and coarse spine size parameters, such as total length or volume, they are in general poorly suited for extracting geometric information about spine necks. A main drawback is that these programs mostly rely on data extraction after processing steps such as thresholding and binarization (Broser et al., 2004; Weaver et al., 2004; Bai et al., 2007; Rodriguez et al., 2008; Shen et al., 2008; Janoos et al., 2009; Jungblut et al., 2012; Blumer et al., 2015), which can compromise the morphological information contained in super-resolution images (Lenz and Tonnesen, 2019).

Given the wealth of morphological data available in super-resolution microscopy images, there is a need for dedicated analysis tools that are capable of faithfully extracting the relevant nanoscale information in units of length and area. Unfortunately, such tools are currently not available and manual analysis is the current working standard for the analysis of super-resolved images of spine morphology. This renders the quantitative analysis of large data sets time-consuming and subject to user fatigue and bias. Moreover, with the manual approach researchers tend to under-utilize their data, reporting only a single neck width measurement per spine, instead of carrying out a more comprehensive analysis to capture the variation of the diameter along the neck.

## Workflow

We introduce SpineJ, an advanced yet user-friendly ImageJ (Schneider et al., 2012) plugin for semi-automatic quantification of spine morphology. SpineJ allows for robust spine morphology analysis using a combination of filtering and segmentation techniques easily accessible through a graphical user interface (GUI). Its workflow is composed of three main steps (**Fig. 1**): (i) Interactive wavelet-based filtering to binarize the dendritic structures of interest from the background. (ii) Semi-automatic reconnection of spines that were erroneously separated from the dendritic shaft, which can happen with very thin and weakly fluorescent necks. (iii) Skeletonization-based segmentation allowing quantitative morphological analysis of spine neck width, length and head surface area.

SpineJ only requires basic ImageJ skills and its GUI allows for swift collection of spine geometry data in a reproducible and unbiased manner. The semi-automated analysis is insensitive to pixel size and gives consistent results with manual analysis of the same spines.

## Integrated software components

### *Wavelet-based image filtering*

The segmentation of spine morphology requires efficient and reliable discrimination of the fluorescent structures of interest from the background. However, common segmentation techniques, such as thresholding or unsharpening, are not well suited because of the differences in fluorescence intensity between the thin spine neck (100 – 300 nm) and the much wider dendritic trunk (1 – 3 μm). To overcome this problem, we used wavelet filtering, which has

intrinsic multi-scale properties and allows to efficiently segment structures with widely varying sizes and intensities.

We used the fast "à-trous" algorithm (Holschneider et al., 1990), which computes a series of multi-scale wavelet coefficients by iterative convolutions of increasing kernels (**Supplementary Fig. 1A, Methods**). A-trous wavelets have several key features: (i) the noise variance $\hat{\sigma}_\epsilon$ can be robustly estimated (Donoho and Johnstone, 1995); (ii) the size of filtered objects is directly related to the wavelet scale, allowing segmenting structures of similar size by thresholding a given wavelet sub-band; (iii) wavelets are not sensitive to absolute image intensities, making it possible to quantify and compare different images.

All the STED images analyzed here were thresholded using the second and third wavelet coefficients with a threshold of $3\hat{\sigma}_\epsilon$ **(Supplementary Fig. 1B, Methods)**. It is important to note that the choice of wavelet coefficients and thresholds will mostly affect the number of spines that will have to be manually reconnected to the dendrite, but will not influence the measurements of spine neck widths, which are carried out on the raw images.

*Spine head reconnection*

Because of the low fluorescence intensity of spine necks relative to spine heads and dendritic trunks, some spine heads may appear to be disconnected from the parent dendrite (**Fig. 2A**). As spine necks are rarely straight and extend at variable angles from the dendrite, their reconnection using a straight line is a poor estimate of the true situation.

We adopted an approach originally developed in NeuronJ that allows the identification and tracing of dim structures (Meijering et al., 2004). A gradient field is directly computed from the original image, resulting in pair-pixel vectors reflecting their direction and magnitude (**Fig. 1**). It allows determining the best path between two points by capturing and following the orientation of bright structures present in the image. Reconnections are performed locally after manual identification of both the isolated spine heads and the parent dendrite (**Fig. 2B**).

*Automatic spine identification*

At this step, neurites are delimited from the background as a binary image and direct identification of spines remains difficult as individual pixels lack context. To overcome this problem, we used skeletons, which are 1D geometric descriptors that naturally contain information on the shape of the structures.

In order to properly account for spine neck geometries, it is essential to ensure smooth vectorial skeletons. This eliminates pixel-based skeletons resulting from segmentation mask thinning since they are usually jagged. We therefore used polygon-based skeletons, where the binarized neurite is represented as a polygon. We combined the $C^1$-continuous Catmull-Rom spline (Farin, 2002) with an arc-length parametrization, providing higher accuracy for curved and junctional parts of the dendrites (**Fig. 2C**), and resulting in segmenting the dendrite outline as a smooth vectorial line independent of pixel size.

The points of the spline are then used as seeds to compute a constrained Delaunay triangulation (**Fig. 2D**). This space-subdividing technique has two advantages: first, triangle connectivity can be used to extract a skeleton, with its branching and end points defining a graph $\boldsymbol{G}$ that accurately describes neuronal morphology (**Fig. 2E, Methods**). We used a pruning algorithm to remove insignificant small branches from the skeleton **(Methods)**. Second, a morphological compartment can be represented as a subgraph of $\boldsymbol{G}$ combined with a set of triangles, facilitating its geometric definition and analysis (**Fig. 3A**). In particular, we use three morphological compartments, namely dendrite, spine head, and spine neck (**Fig. 3B**). The automatic identification of spines is achieved by graph theory applied to $\boldsymbol{G}$ **(Fig. 3C, Methods)**. In

instances where spines are ill-defined, a correction can easily be manually applied using the *ROI tools* in ImageJ (**Methods**).

*Spine quantification*

Because the border between a spine head and its neck is not well defined, their separation is hard to automatize. We used the Delaunay triangulation to facilitate the separation. The neck base is defined as the shared edge between a spine and its parent dendrite (**Fig. 3D**), while its tip, which connects the spine head, is manually defined by selecting a triangle edge (**Fig. 3E**).

The software then automatically traces evenly spaced lines perpendicular to the neck skeleton (**Fig. 3F**). Neck widths are extracted from the full width at half maximum (FWHM) computed by Gaussian fits of intensity line profiles gathered from the raw images. The final neck width reported by SpineJ is determined as the minimal width along the spine neck. The spine head area is directly computed on the binarized image after separation from the neck. Most of the parameters can be selected automatically or adjusted manually (**Methods**).

*Robustness with respect to noise*

To assess the robustness of the wavelet segmentation with noise, we computationally regraded a STED image by adding Gaussian noise (**Supplementary Fig. 3, Methods**). In all scenarios, wavelet filtering managed to properly identify both the dendrite and the spines. However, low-SNR images required a threshold adjustment (**Methods**), resulting in small artefacts in the background which could easily be dismissed. Finally, low SNR also impacts the spine/dendrite reconnection and the neck width computation, since Gaussian fitting the intensity line profiles is less accurate with low-SNR images (Thompson et al., 2002) (**Supplementary Fig. 2**).

## Results

*Measurements of spine neck width*

Spine neck widths in STED images have usually been determined by fitting one-dimensional Gaussian (Ding et al., 2009; Tønnesen et al., 2011) or Lorentzian functions (Bethge et al., 2013; Takasaki et al., 2013) to intensity line profiles of spine neck cross sections and extracting the FWHM value (**Fig. 4A**). Usually, only a single measurement per spine is manually taken at the place where the neck looks the thinnest. By contrast, SpineJ automatically computes the minimal, maximal and average FWHM values all along the spine neck.

While both Gaussian and Lorentzian functions produced highly correlated measurements ($R^2$ = 0.83, y = 0.89x + 9.3; **Fig. 4B**), the Gaussian fit returned slightly larger values for the neck width than the Lorentzian fit ($\text{mean}_{\text{Gauss}}$ 144 ± 38 nm , $\text{mean}_{\text{Lorentz}}$ 137 ± 37 nm (SD), p = 0.0006, paired t-test, n = 69 spines from 6 dendritic segments; **Fig. 4C**). All subsequent measurements were performed by using Gaussian fits.

*Comparison of SpineJ performance with manual quantification*

We compared the performance of SpineJ with manual quantifications (**Fig. 4D-H**). For this, we asked 5 "naïve" persons to analyze 30 spines from 3 dendritic segments. In the case of manual analysis, the angle, position of the line across the necks and elliptical ROI representing the spine head were set by the experimenter. For the neck width, users placed the line ROI at the location they thought the neck was the thinnest. For the head, the area was measured from the ellipse ROI they placed to approximate its borders.

For spine neck widths, even though the values obtained with SpineJ were smaller compared to manual analysis (SpineJ: 146 ± 48 nm SD; manual: 177 ± 64 nm SD), they correlated well (Linear fit: y = 0.799x, 95% confidence [0.759-0.836], no intercept; Correlation Kendall's tau: $\tau = 0.621$, $p < 0.0001$; **Fig 4.D**). SpineJ is expected to report a smaller minimal value because

it makes many more measurements along the spine neck, and thus has a much better chance to find the true minimal neck width. The variability between users (**Methods**) was almost two times lower with SpineJ compared to the manual analysis (SpineJ: 11 ± 11 nm SD; manual: 21 ± 18 nm SD; **Fig. 4.E**).

For spine heads, SpineJ and manual analysis returned similar areas (SpineJ: 0.37 ± 0.26 µm² SD; manual: 0.38 ± 0.34 µm² SD), which were highly correlated (Linear fit: $y = 0.863x$, 95% confidence [0.807-0.907], no intercept; Correlation Kendall's tau: $\tau = 0.81$, $p < 0.0001$; **Fig. 4.F**). The variability between users was more than five times lower with SpineJ compared to manual analysis (SpineJ: 0.014 ± 0.02 µm² SD; manual: 0.078 ± 0.07 µm² SD; **Fig 4.G**).

The mean analysis time per spine was less than half for SpineJ compared to manual analysis (SpineJ: 29 ± 12 s SD; manual: 70 ± 20 s SD; **Fig. 4.H**).

## Discussion

By breaking the diffraction barrier, super-resolution fluorescence microscopy gives optical access to micro-anatomical structures in live brain tissue. This has allowed geometric analysis of dendritic spines and axons, providing new insights into their biological function (Tønnesen et al., 2014; Chereau et al., 2017). However, geometric analysis of dendritic spines in super-resolution images currently still requires manual intervention, which inevitably introduces variability and bias, and is very time-consuming.

Here, we introduce SpineJ, a new software to quantify geometric information of nanoscale details of dendritic spines. The strength of SpineJ lies in its ability to analyze spine neck geometry in a fast and reproducible manner.

The software is based on a structured workflow design, where the user selects the population of spines to be analyzed in a first step, while automatic measurements are performed from the original image in a second step. This separation of spine selection and analysis steps has the effect of limiting user bias, as illustrated by the low variability in the measurements performed by different users.

We validated the performance of SpineJ by comparing it with manual analysis. SpineJ systematically reported a smaller minimal neck width than manual analysis, reflecting the fact that SpineJ measures neck width all along the entire neck and thus can precisely find the true minimal value.

The high variability associated with manual analysis of spine heads shows that elliptical shapes are a poor representation for spine heads. Users could use the hand-drawn ROI to properly account for discontinuous borders, but this would become even more complicated, time-consuming and an additional source of variability. By contrast, SpineJ can reliably estimate spine head size without making any assumptions about their shapes. Finally, the combination of robust image processing techniques with straightforward user interactions (a few mouse clicks) and instant visual feedback (preview, overlay, statistics) minimizes the time spent on analyzing each spine. While this gain in time was already significant during our tests with a small dataset (30 spines in 3 dendritic segments), it will surely increase with user fatigue when analyzing more than hundreds of spines.

Contrary to manual analysis, which is designed for a specific task and cannot be easily extended, the workflow used in SpineJ allows going beyond the more limited measurements done in the past. For example, the custom of measuring neck width at only a single place simply reflects the inadequacy of manual analysis. By contrast, by returning a multitude of systematic measurements all stored and accessible through a customized widget, SpineJ provides a more robust and reliable analysis of spine morphology (**Supplementary Fig. 3**).

We therefore encourage the adoption and use of SpineJ as a reference software for more meaningful and transparent quantitative analysis of spine morphology.

## Methods

*A-trous wavelet filtering*

The 'à trous' wavelet transform represents a discrete and translation-invariant approach to the classical continuous wavelet transform. We define $c_0(k)$ as the original fluorescent image. The smoothed data $c_i(k)$ at a given resolution level $i$ and at pixel $k$ are obtained by the convolution $c_i(k) = \sum_l h(l) c_{i-1}(k + 2^{i-1} l)$, where $h$ is a low-pass scaling function (usually a $B_3$ spline). The difference between two consecutive resolution levels $w_i(k) = c_{i-1}(k) - c_i(k)$ represents the wavelet coefficients (or subband) at level $i$. Segmentation is achieved by thresholding these wavelet coefficients independently.

In fluorescence microscopy, the noise is a mixture of Gaussian (electronic) and Poisson (photon) statistics. During the wavelet transform, while the noise $\varepsilon$ is decomposed in all of the wavelet sub-bands, more than 80% of its components are present in the first wavelet sub-band. For this reason, a robust estimation of the noise variance $\hat{\sigma}_\epsilon$ based on the median absolute value of the first wavelet coefficients and defined as $\hat{\sigma}_\epsilon = \frac{med(|w_1(x,y)|)}{0.06745}$ was proposed (Donoho and Johnstone, 1995). Since most of the noise $\epsilon$ is part of $w1$ while containing little useful signal, such estimator has become very popular and is widely used.

To identify pixels that are part of the actual dendrite, the wavelets sub-bands are thresholded with $w_i(x, y) > \alpha\hat{\sigma}_\epsilon$ where $\alpha$ is a coefficient set by the user. The final binary image is reconstructed by summing all the filtered wavelet coefficient sub-bands. We experimentally determined that a threshold of $3\hat{\sigma}_\epsilon$ applied on the second and third wavelet sub-bands provides a robust segmentation of our images. These values might slightly fluctuate depending on the acquisition parameters.

*Gradient field computation and reconnection*

In computer vision, ridges define a set of curves that represent local maxima of the image in at least one dimension. Since neurites are thin bright structures standing out from a darker background, they can readily be represented by ridges. Ridge detection is facilitated by computing a vector field on the image, with each pixel being associated with a vector direction and magnitude (Meijering et al., 2004).

Reconnection between spine head and dendrite is initiated by two user-defined clicks within the respective structures to be reconnected. The algorithm computes the optimal path connecting the two points as the path exhibiting the minimal cumulative cost when following the vector field main directions. The ridge connects the points of maximal magnitude.

*Polygon-based skeleton generation*

Triangles of the constrained Delaunay triangulation are divided into three categories (**Supplementary Fig. 4A**): extremal (E, one neighboring triangle), transitional (T, two neighboring triangles), and junctional (J, more than two neighboring triangles). The skeleton is obtained by connecting the midpoints of the edges shared by transitional triangles (**Supplementary Fig. 4B**). The graph is easily generated by creating graph nodes originating from all the junctional and extremal triangles, and connecting them by following the respective skeleton branches (**Supplementary Fig. 4B**). To avoid insignificant and unstable branches, a pruning algorithm is applied on the skeleton extremal branches (**Supplementary Fig. 4C**), resulting in a more stable skeleton and graph (**Supplementary Fig. 4D**).

*Pruning of the skeleton*

To avoid insignificant and unstable branches, extremal triangles are subject to a pruning algorithm designed to merge extremal regions with adjacent transitional triangles. If all the outline points of the extremal region are inside the semicircle, whose diameter is the edge between the extremal region and its adjacent transitional triangle, this edge is removed and the transitional triangle is added to the extremal region (**Supplementary Fig. 4C,** *top*). This process is repeated until at least one outline point is outside the semicircle or if a junctional triangle is reached (**Supplementary Fig. 4C,** *bottom*). After pruning, the skeleton is more stable and still embeds the topology of the neurite.

This pruning process affects both the graph and the skeleton. If the pruning translates from an extremal triangle to a junctional one, the whole graph branch (and corresponding skeleton) is discarded (**Supplementary Fig. 4D**). In addition, the graph node originating from the junctional triangle is also discarded. If the pruning does not reach a junctional triangle, the graph node originating from the extremal triangle is unchanged, while the skeleton stops at the first pruned triangle it encounters (**Supplementary Fig. 4D**).

*Automatic spine identification*

Since the graph $\boldsymbol{G}$ is extracted from the Delaunay triangulation, any of its vertex or edge is linked to a set of triangles (**Supplementary Fig. 5A-B**). The degree $d(v_i)$ counts the number of vertices connected to $v_i$. First, the graph vertices are classified in 2 categories: leaf ($d(v_i) = 1$) or junctional ($d(v_i) = 3$) nodes (**Supplementary Fig. 5C**). Since they are part of the dendritic trunk, leaf nodes touching the image borders are modified to junctional nodes. All leaf nodes are tagged as spine nodes. Spines, on the other hand, start from a leaf node and can be composed of several nodes (extremal and junctional). Junctional nodes are then tagged as spine nodes if they are connected to two leaf nodes within a distance $d_{merge}$ of 800 nm (**Supplementary Fig. 5D**), corresponding to a common length for spines. This value can be modified by users. By adding all the edges of $\boldsymbol{G}$ that connect two spine nodes plus the connected edges with only one dendrite node, we define a subgraph $\boldsymbol{G_{spines}}$ that represents all the spines (**Supplementary Fig. 5E**), with individual spines as the connected components of $\boldsymbol{G_{spines}}$. Each spine is then defined as the set of triangles of all its vertices and edges (**Supplementary Fig. 5F**).

*Manual correction of missing or ill-defined spines*

Since spines can exhibit very diverse shapes, the automatic spine classification can misidentify some spines, especially if $d_{merge}$ is poorly defined (**Supplementary Fig. 6A-B, $d_{merge}$ = 200 nm**). Manual definition of a spine is possible by tracing a ROI on the image (**Supplementary Fig. 6C**), using one of the ROI (rectangular, oval, freehand, etc.) provided by ImageJ. First, any existing spine intersecting this ROI is discarded, with all its graph nodes and edges tagged as dendrite (**Supplementary Fig. 6AD**). Then, all the graph nodes and edges intersecting the ROI are tagged as spine (**Supplementary Fig. 6E**), and finally the corresponding set of triangles is used to compute spine shape (**Supplementary Fig. 6F**).

*Automatic determination of the neck width*

To quantify the width along a given spine neck, SpineJ automatically traces several evenly spaced lines perpendicularly through the spine neck skeleton. Each line is defined by a width and a thickness, describing a rectangle under which SpineJ computes the intensity profile of the spine neck in the raw image, thus corresponding to line intensity profiles in previous manual analyses. In SpineJ the line thickness can be chosen by the users, and here we used a fixed value of 100 nm for all our images. The number of lines $nb$ is defined as $nb = {}^{l}/_{0.75 * t}$, with $l$ and $t$ being respectively the neck length and line thickness. This ensures optimal sampling with some overlap between the lines. Each line represents the width at a specific location of the neck skeleton. This width is determined by Gaussian fitting the intensity profile with spatial extent

ranging between 400 nm to 800 nm, with 50 nm steps. The fit having the highest $R^2$ is selected as the optimal width for this skeleton position.

In our study, the final neck width reported is defined as the minimal width of all the lines computed alongside the neck skeleton.

*Assessment of the wavelet filtering robustness*

To test the robustness of the method with the noise, we added Gaussian noise using the "Add Specified Noise" function in ImageJ, of amplitudes 5, 15 and 25 (**Supplementary Fig. 3B-D**). For each structure of interest (dendrite, spine head and neck), SNR was quantified as the amplitude of the signal, subtracted from the background, divided by the standard deviation of the signal. For the two high SNR images (**Supplementary Fig. 3A-B**), a threshold of $w_3(x,y) > 3\hat{\sigma}_\epsilon$ was used to generate the binary image. For the two low SNR images (**Supplementary Fig. 3C-D**), this threshold was adjusted to $w_3(x,y) > \hat{\sigma}_\epsilon$. The minimum goodness of fit of the Gaussian fitting was also adjusted depending on the SNR: we used 0.8 for the original image (**Supplementary Fig. 3A**), and 0.6 to 0.4 for the noisiest images (**Supplementary Fig. 3B-D**). Even with a smaller goodness of fit value, several fits either failed or underestimated the neck width (black arrows in **Supplementary Fig. 3B-D**).

*Gaussian and Lorentzian fits*

Gaussian and Lorentzian fits were performed within ImageJ and Prism, respectively.

*Protocol for analysis and computation of variability between users*

A set of 30 spines from 3 cells were preselected for the analysis, in a well-defined order. We asked each user to follow this order and checked for it, allowing us to compute variabilities on each measurement. A measurement is defined as the quantification of one parameter, e.g. neck width, head area, etc…, for a single spine. Each measurement was performed by the 5 users, both using SpineJ and manually. The variability per measurement is defined as the standard deviation between the 5 values obtained by each user.

*Availability of SpineJ*

Plugin and source-code of SpineJ are available at https://github.com/flevet/SpineJ.

## References


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

Berning S, Willig KI, Steffens H, Dibaj P, Hell SW (2012) Nanoscopy in a living mouse brain. Science 335:551.

Bethge P, Chereau R, Avignone E, Marsicano G, Nägerl UV (2013) Two-photon excitation STED microscopy in two colors in acute brain slices. Biophys J 104:778-785.

Blumer C, Vivien C, Genoud C, Perez-Alvarez A, Wiegert JS, Vetter T, Oertner TG (2015) Automated analysis of spine dynamics on live CA1 pyramidal cells. Med Image Anal 19:87-97.

Broser PJ, Schulte R, Lang S, Roth A, Helmchen F, Waters J, Sakmann B, Wittum G (2004) Nonlinear anisotropic diffusion filtering of three-dimensional image data from two-photon microscopy. J Biomed Opt 9:1253-1264.

Chereau R, Saraceno GE, Angibaud J, Cattaert D, Nägerl UV (2017) Superresolution imaging reveals activity-dependent plasticity of axon morphology linked to changes in action potential conduction velocity. Proc Natl Acad Sci U S A 114:1401-1406.

Ding JB, Takasaki KT, Sabatini BL (2009) Supraresolution imaging in brain slices using stimulated-emission depletion two-photon laser scanning microscopy. Neuron 63:429-437.

Donoho DL, Johnstone IM (1995) Adapting to Unknown Smoothness via Wavelet Shrinkage. Journal of the American Statistical Association 90:1200-1224.

Farin G (2002) Curves and Surfaces for Computer Aided Geometric Design, 5th Edition. San Diego: Academic Press.

Holschneider M, Kronland-Martinet R, Morlet J, Tchamitchian P (1990) A real-time algorithm for signal analysis with the help of the wavelet transform. In: Wavelets. Inverse problems and theoretical imaging (Combes J.M. GA, Tchamitchian P., ed), pp 289–297: Springer-Verlag.

Holtmaat A, Svoboda K (2009) Experience-dependent structural synaptic plasticity in the mammalian brain. Nature reviews Neuroscience 10:647-658.

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. Med Image Anal 13:167-179.

Jungblut D, Vlachos A, Schuldt G, Zahn N, Deller T, Wittum G (2012) SpineLab: tool for three-dimensional reconstruction of neuronal cell morphology. J Biomed Opt 17:076007.

Lenz MO, Tonnesen J (2019) Considerations for Imaging and Analyzing Neural Structures by STED Microscopy. Methods Mol Biol 1941:29-46.

Matsuzaki M, Ellis-Davies GC, Nemoto T, Miyashita Y, Iino M, Kasai H (2001) Dendritic spine geometry is critical for AMPA receptor expression in hippocampal CA1 pyramidal neurons. Nat Neurosci 4:1086-1092.

Meijering E, Jacob M, Sarria JC, Steiner P, Hirling H, Unser M (2004) Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry A 58:167-176.

Montagna E, Dorostkar MM, Herms J (2017) The Role of APP in Structural Spine Plasticity. Frontiers in molecular neuroscience 10:136.

Nägerl UV, Willig KI, Hein B, Hell SW, Bonhoeffer T (2008) Live-cell imaging of dendritic spines by STED microscopy. Proc Natl Acad Sci U S A 105:18982-18987.

Pfeiffer T, Poll S, Bancelin S, Angibaud J, Inavalli VK, Keppler K, Mittag M, Fuhrmann M, Nägerl UV (2018) Chronic 2P-STED imaging reveals high turnover of dendritic spines in the hippocampus in vivo. Elife 7.

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:e1997.

Sala C, Segal M (2014) Dendritic spines: the locus of structural and functional plasticity. Physiol Rev 94:141-188.

Schneider CA, Rasband WS, Eliceiri KW (2012) NIH Image to ImageJ: 25 years of image analysis. Nat Methods 9:671-675.

Shen H, Sesack SR, Toda S, Kalivas PW (2008) Automated quantification of dendritic spine density and spine head diameter in medium spiny neurons of the nucleus accumbens. Brain Struct Funct 213:149-157.

Svoboda K, Tank DW, Denk W (1996) Direct measurement of coupling between dendritic spines and shafts. Science 272:716-719.

Takasaki KT, Ding JB, Sabatini BL (2013) Live-cell superresolution imaging by pulsed STED two-photon excitation microscopy. Biophys J 104:770-777.

Thompson RE, Larson DR, Webb WW (2002) Precise nanometer localization analysis for individual fluorescent probes. Biophys J 82:2775-2783.

Tønnesen J, Nägerl UV (2013) Superresolution imaging for neuroscience. Exp Neurol 242:33-40.

Tønnesen J, Katona G, Rozsa B, Nägerl UV (2014) Spine neck plasticity regulates compartmentalization of synapses. Nat Neurosci 17:678-685.

Tønnesen J, Nadrigny F, Willig KI, Wedlich-Soldner R, Nägerl UV (2011) Two-Color STED Microscopy of Living Synapses Using A Single Laser-Beam Pair. Biophys J 101:2545-2552.

Weaver CM, Hof PR, Wearne SL, Lindquist WB (2004) Automated algorithms for multiscale morphometry of neuronal dendrites. Neural Comput 16:1353-1383.

Willig KI, Steffens H, Gregor C, Herholt A, Rossner MJ, Hell SW (2014) Nanoscopy of filamentous actin in cortical dendrites of a living mouse. Biophys J 106:L01-03.

Yuste R, Bonhoeffer T (2004) Genesis of dendritic spines: insights from ultrastructural and imaging studies. Nature reviews Neuroscience 5:24-34.

## Figure Legends

**Figure 1: SpineJ workflow.** The workflow is composed of three steps. First, the dendrites of interest are identified by using a wavelet filtering process. Spines separated from the dendrite because of weak fluorescent necks can be reconnected by following a gradient field computed on the original image. Finally, the morphological analysis of spines is performed by computing a constrained Delaunay triangulation on the dendrite outlines.

**Figure 2: Abstracting dendrites as graphs.** (A) Original STED image (left) and binarized dendrite after à-trous wavelet filtering (right). Scale bar 1 µm for the complete image and 500 nm for the magnification. (B) Reconnection of separated dendritic spines (green). Location where the user had to click are identified by the red arrow tips. (C) Vectorial outline of the dendrite (red). (D) Constrained Delaunay triangulation computed on the vectorial outline (red). (E) Skeleton (magenta) and graph (green) extracted from the triangulation.

**Figure 3: Spine neck quantification.** (A) Triangle sets (left) and corresponding graph (right) of a spine (green) connected to its dendrite (orange), scale bar: 300 nm. (B) Definition of the three morphological compartments: the spine head (magenta) and neck (green) and the dendrite (orange). (C) Automatic identification of 24 spines (scale bar: 1 µm). (D) Basis of the spine neck is automatically defined as the edge connecting the triangle sets of the spine and the dendrite (red arrow). (E) The user defines the neck end by clicking on a triangle edge (tip of the red arrow). (F) SpineJ computes the three compartments and automatically traces several perpendicular lines to analyze the neck morphology (blue).

**Figure 4: Comparison of Lorentzian and Gaussian fits and measurement of spines.** (A-C) Comparison of Lorentzian and Gaussian fits. (A) Lorentzian (FWHM 92 nm) and Gaussian (FWHM 102 nm) fits of the same spine neck width. (B) Gaussian fit (144 ± 38 nm SD) identifies a slightly wider width than the Lorentzian one (137 ± 37 nm SD), n = 69 spines. (C) Correlation between the Lorentzian and the Gaussian FWHM ($R^2$ = 0.83). (D-H) Comparison of manual and SpineJ analysis. (D) Correlation between the measurements obtained manually or by SpineJ for the neck width. (E) User variability of the manual and SpineJ analysis for the neck width. (F) Correlation between the measurements obtained manually or by SpineJ for the head area. (G) User variability of the manual and SpineJ analysis for the head area. (H) Comparison of the time needed to analyze one spine manually and with SpineJ. N = 150 spines.

**Supplementary Figure 1: Wavelet filtering.** Wavelets (A) and coefficients (B) sub-bands (scale bar: 1 µm). (B) Filtered sub-bands resulting from applying a threshold of $3\widehat{\sigma_\varepsilon}$ (top), with $\widehat{\sigma_\varepsilon}$ the noise variance determined automatically form the first wavelet sub-band. The final filtered image is obtained by adding the 2$^{nd}$ and 3$^{rd}$ filtered sub-bands (bottom).

**Supplementary Figure 2: Robustness to noise.** (A) Fluorescence image (scale bar: 1 µm) with magnification of a spine (scale bar: 500 nm), SNR values, reconnected binary image and quantification of the magnified spine. Addition of Gaussian noise on the original image (A) with mean 0 and standard deviation 5 (B), 15 (C) and 25 (D) grey levels. Black arrows identification location where the Gaussian fitting either failed or underestimated the neck width.

**Supplementary Figure 3: Modeling spine necks.** (A) Measurements of the neck width at several locations (green) on the original image (left). (Right) Corresponding unfolding and modeling of the spine neck, with its length, maximal and minimal width (red). (B-C) Modeling of the neck for 3 spines. Scale bar 400 nm. (E) Customized widget storing every measurement performed by SpineJ.

**Supplementary Figure 4: From triangulation to spine neck quantification.** (A) Delaunay triangulation of a spine. Triangles are classified into three categories: extremal (E), junctional (J) and transitional (T). (B) Original skeleton (green) extracted from the triangulation, and corresponding graph (magenta). (C) Pruning algorithm: if all points of an extremal region are inside the circumcircle, the current triangle is merged to its neighbor. This process is stopped when encountering a junctional triangle or if a point is outside the circumcircle. (D) Skeleton and graph after pruning.

**Supplementary Figure 5: Automatic spine identification.** (A) Vectorial outline of a spine and a portion of the dendrite (scale bar: 1 µm). (B) Constrained Delaunay triangulation of the outline. The graph is computed after pruning (cyan). (C) Two classes organization of the graph vertices. (D) Leaf nodes are tagged as spine nodes. A junctional node is tagged as spine node (arrow) since it is connected to two spine nodes within a distance of 800 nm. (E) Subgraph representing the spines (green). It is composed of five connected components. (F) Corresponding identified spines. The spine/dendrite separations are defined by the red edges.

**Supplementary Figure 6: Manual correction of ill-defined spines.** (A-B) Poorly defined $d_{merge}$ distance ($d_{merge}$ = 200 nm) results in misidentification of a spine as two spines, with its neck being identified as part of the dendritic trunk (scale bar: 500 nm). (C) Definition of a ROI directly on the image. (D) The two misidentified spines are discarded, with their graph nodes and edges tagged as dendrite. (E) All the graph nodes and edges intersecting the ROI are tagged as spine. (F) Corrected spine.

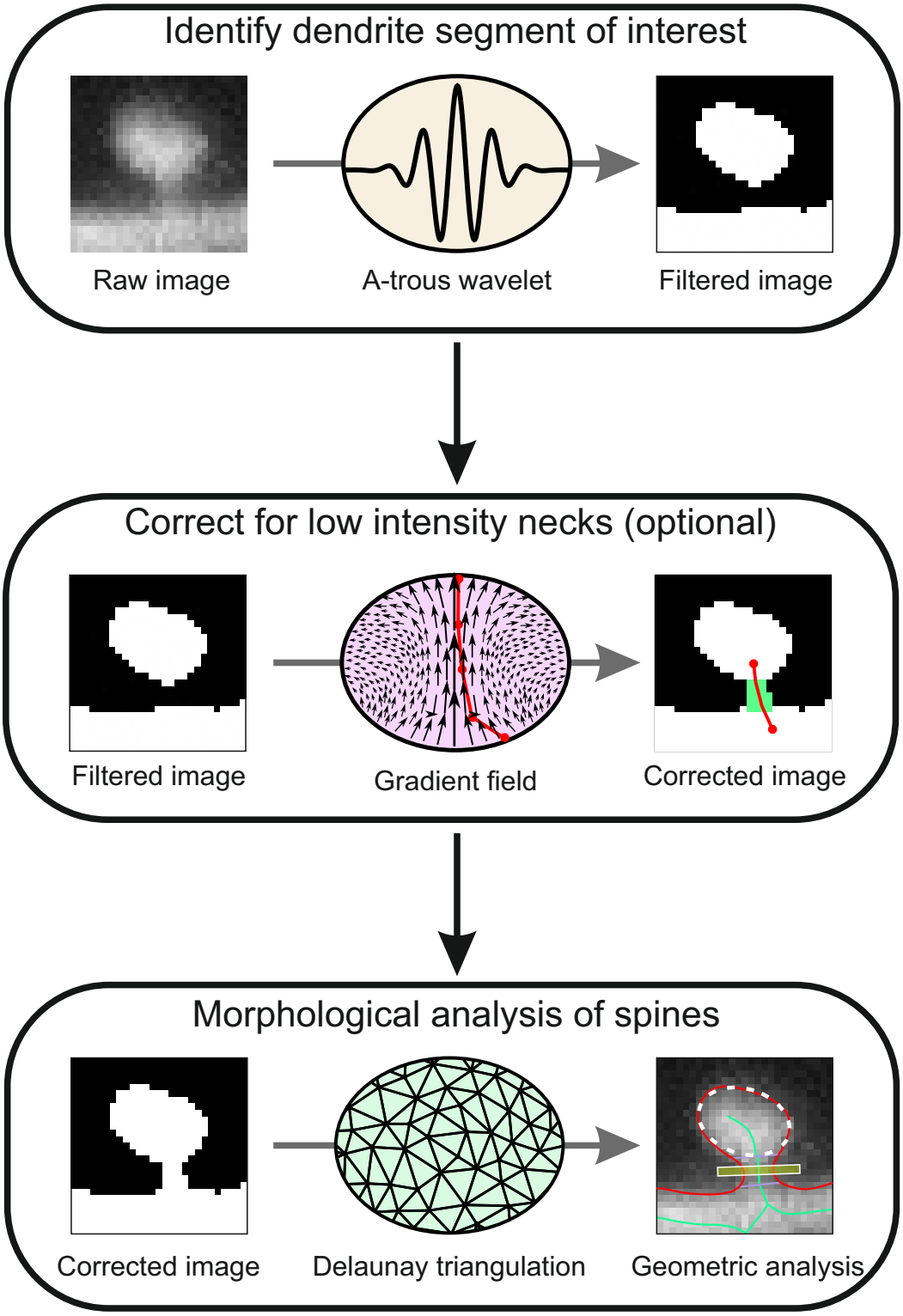
Identify dendrite segment of interest
Raw image
A-trous wavelet
Filtered image
Correct for low intensity necks (optional)
Filtered image
Gradient field
Corrected image
Morphological analysis of spines
Corrected image
Delaunay triangulation
Geometric analysis

Figure 1

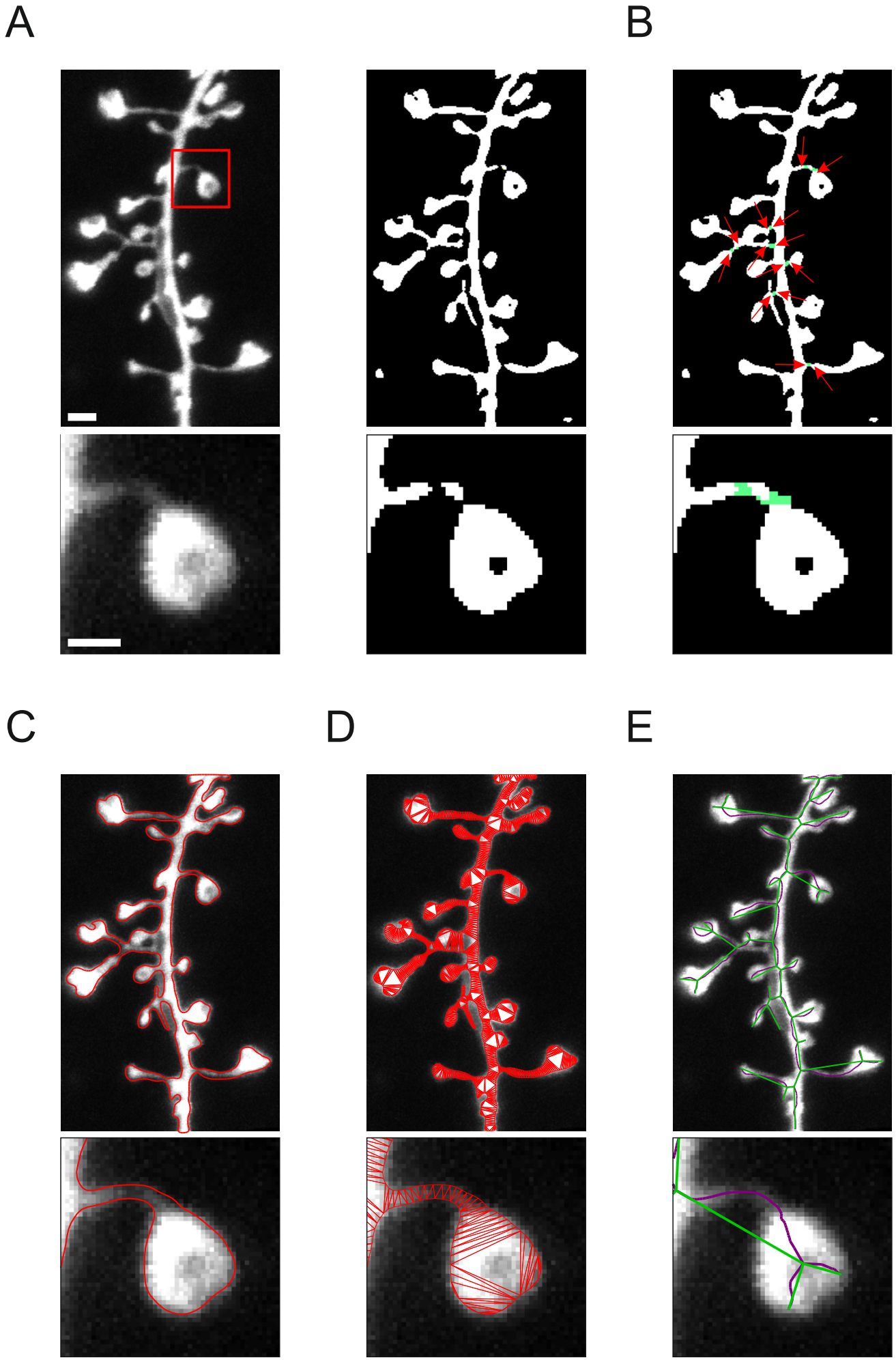


Figure 2

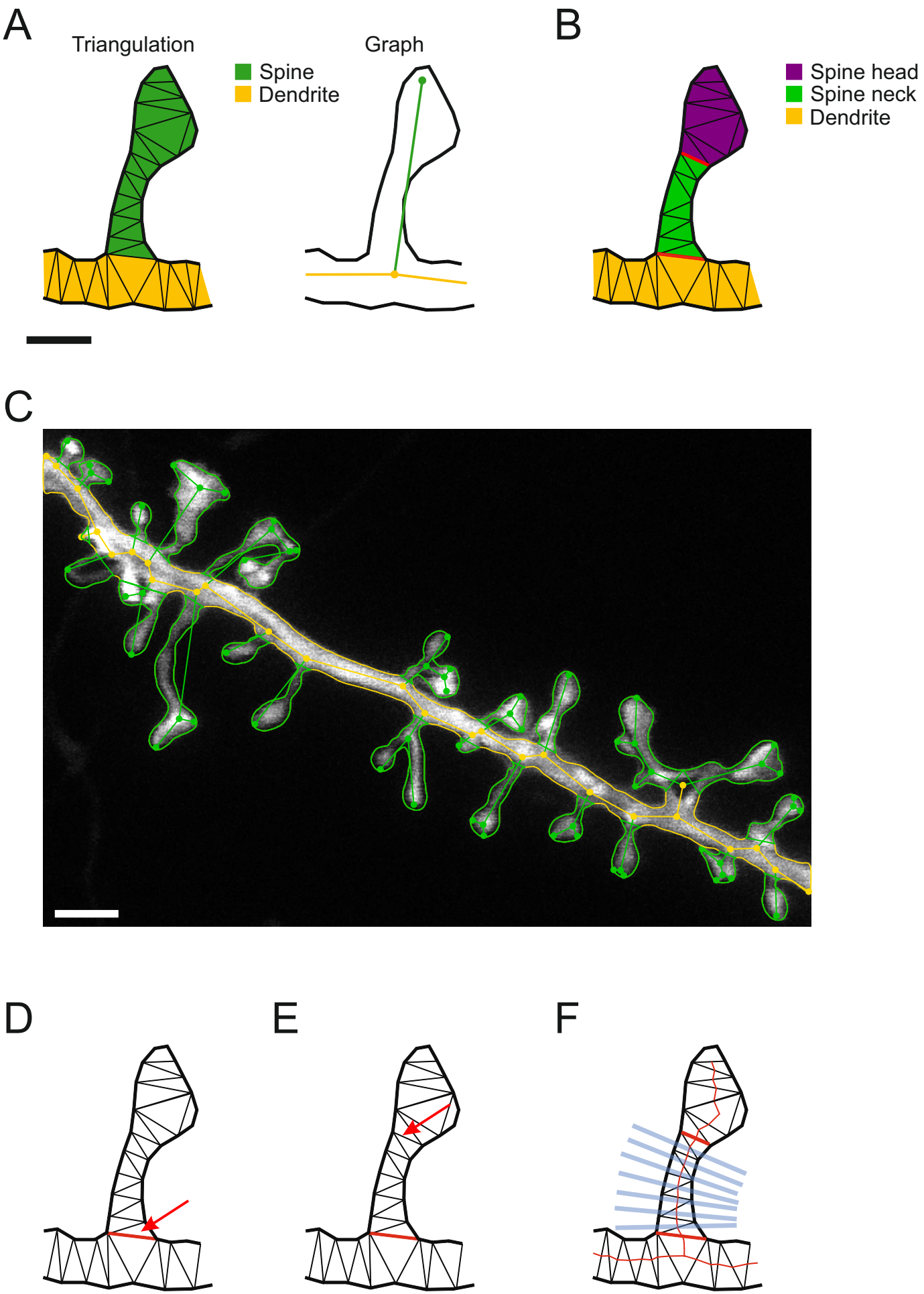
A
Triangulation
Graph
Spine
Dendrite
B
Spine head
Spine neck
Dendrite
C
D
E
F

Figure 3

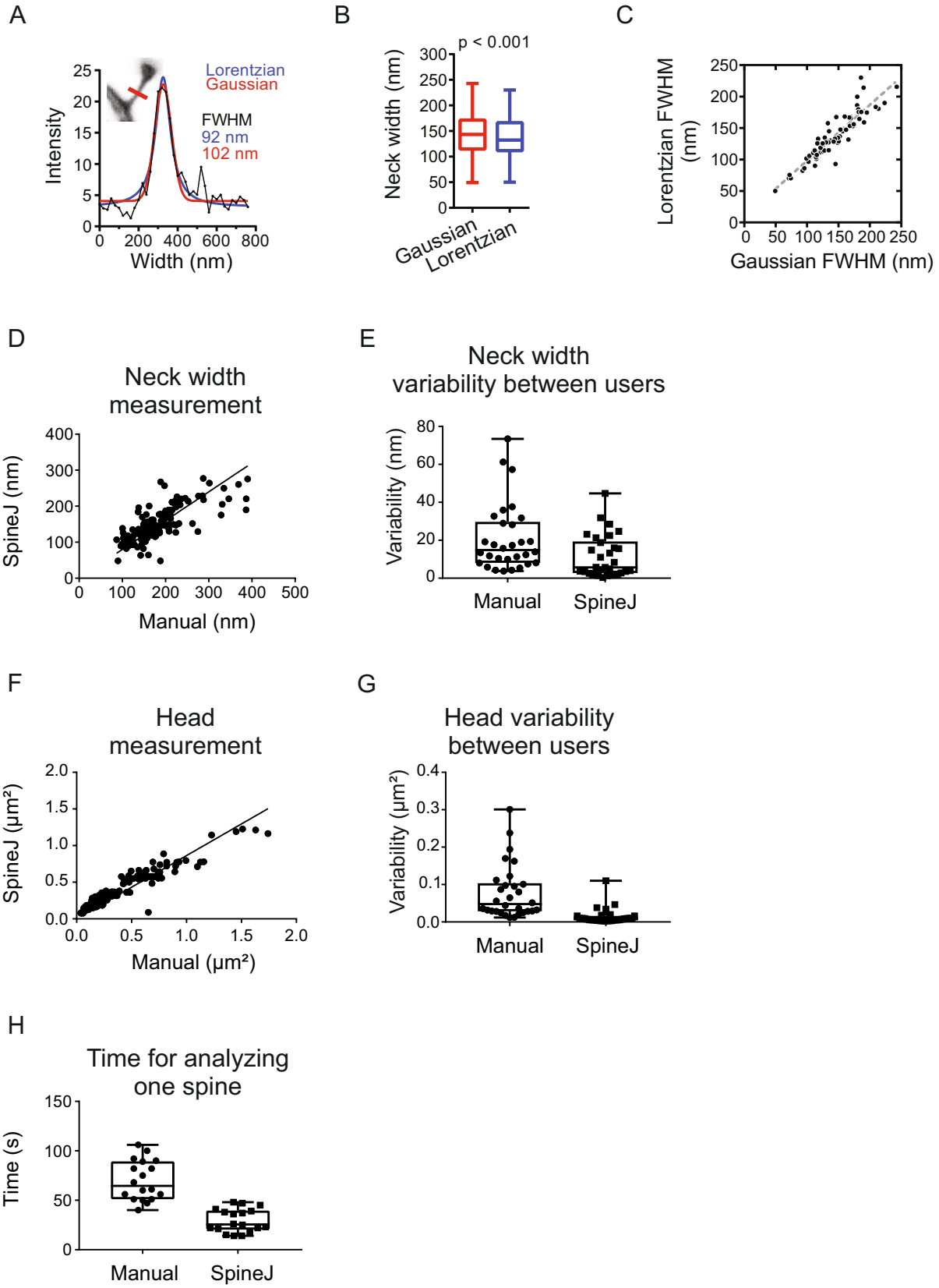
A
Lorentzian
Gaussian
FWHM
92 nm
102 nm
Intensity
Width (nm)
B
p < 0.001
Neck width (nm)
Gaussian
Lorentzian
C
Lorentzian FWHM (nm)
Gaussian FWHM (nm)
D
Neck width measurement
SpineJ (nm)
Manual (nm)
E
Neck width variability between users
Variability (nm)
Manual
SpineJ
F
Head measurement
SpineJ (µm²)
Manual (µm²)
G
Head variability between users
Variability (µm²)
Manual
SpineJ
H
Time for analyzing one spine
Time (s)
Manual
SpineJ

Figure 4