FUZZY LOGIC BASED DETECTION OF NEURON BIFURCATIONS IN MICROSCOPY IMAGES

Miroslav Radojević, Ihor Smal, Wiro Niessen and Erik Meijering

Biomedical Imaging Group Rotterdam
Erasmus MC – University Medical Center Rotterdam
Email: m.radojevic@erasmusmc.nl

ABSTRACT

Quantitative analysis of neuronal cell morphology from microscopic image data requires accurate reconstruction of the axonal and dendritic trees. The most critical points to be detected in this process are the bifurcations. Here we present a new method for fully automatic detection of bifurcations in microscopic images. The proposed method models the essential characteristics of bifurcations and employs fuzzy rule based reasoning to decide whether the extracted image features indicate the presence of a bifurcation. Algorithm tests on synthetic image data show high noise immunity and experiments with real fluorescence microscopy data exhibit average recall and precision of 90.4% and 90.5% respectively.

***Index Terms*—** Neuron reconstruction, bifurcation detection, fuzzy logic, fluorescence microscopy.

1. INTRODUCTION

Reconstruction of neuronal cell morphology from microscopic image data is an important task in many neuroscientific studies [1]. A key step in this process is the tracing of the axonal and dendritic trees emanating from the cell body. Generally these trees can be described as a graph consisting of hundreds to many thousands of branch points and segments. In principle, all branch points in a neuronal tree are bifurcations: locations where the network diverges, forming two new distinct threads in space. Accurate detection of bifurcation points is therefore of crucial importance in neuronal reconstruction. Despite being reasonably intuitive for human visual recognition, even in a single image neuronal bifurcations are far too numerous for manual annotation and require automated image analysis methods for their detection.

Computational methods for modeling and detecting bifurcation points in tree-like image structures vary depending on the type of image data, the tree complexity, and the application. Here we focus on fluorescence microscopy images of neurons (Fig. 1), where the detection is hampered by possible signal interruptions (due to imperfect staining), large differences between branch segments in terms of both intensity and diameter, and the often intricate local morphology. Detection may be performed either implicitly (after initial global image segmentation) or explicitly (as input for subsequent segmentation by local tracing). Several approaches have recently been proposed with reasonable success, such as deducing bifurcation points as locations where the active contours used for branch segment tracing happen to overlap [2] or where segmentation wavefronts split [3, 4], or applying hypothesis testing after initial tracing [5]. A wavelet based local model fitting scheme [6] enables explicit localisation, but does not model all relevant degrees of freedom in the data, such as the angles between branches. Some solutions use the spread of neighboring pixel intensities as a detection criterion [7]. Given the still limited performance of state-of-the-art neuron reconstruction algorithms [8], and the need to further reduce manual annotation and curation, improved methods for detection of bifurcations points are very much needed.

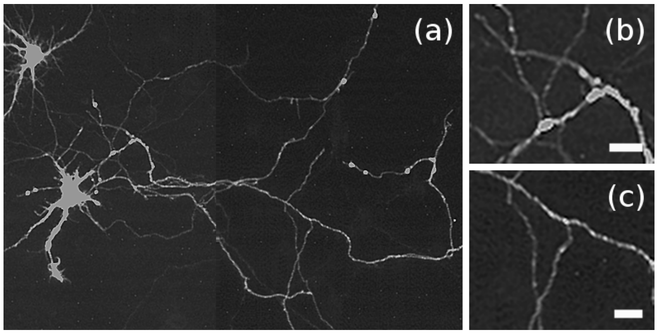


Fig. 1. (a) Fluorescence microscopy image of a neuron. (b,c) Detailed sections of the image (scale bar: 100 μm).

Here we propose a novel solution for fully automated detection of bifurcation loci in neuron microscopy images. Common signal discontinuities and bifurcation configuration diversity (branch intensities, diameters, angles) are tackled with a new filtering and profile analysis scheme. Uncertainty and nonlinearity of the final decision are treated with fuzzy logic and an appropriate set of IF-THEN rules.

This work was funded by the Netherlands Organization for Scientific Reseach (NWO grant 612.001.018 awarded to EM).

2. METHOD

The proposed method consists of three steps. First, a set of filters is applied to calculate an angular profile of local image structure. Next, peaks in the profile are detected, and associated follow-up points in the corresponding radial direction are analyzed. Finally, a fuzzy rule-based system is used to determine the most likely bifurcation points. We also discuss implementational aspects of our method.

2.1. Local filtering and profiling

To estimate the presence of curvilinear threads we use a set of oriented filters $T(x', y')$ distributed around a given location (x, y) (Fig. 2a) by rotation over angle $\alpha \in [0, 2\pi)$ and translation over a distance kD:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} kD \\ kD \end{bmatrix}$$

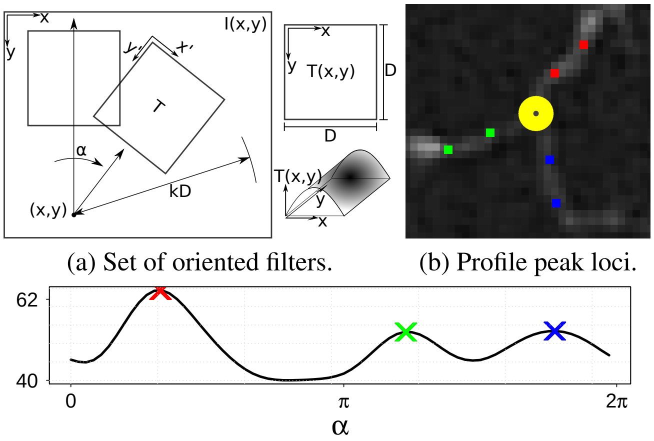
where k is a scale parameter and D defines the filter kernel patch size, the latter of which relates to the expected diameter of the traced neuron structure. The kernels are profiled with normalized Gaussian weights (Fig. 2a):

$$T(x,y) = C\,\mathrm{e}^{-x^2/2\sigma^2}, \quad (x,y) \in \left[-\frac{D}{2}, \frac{D}{2}\right], \quad \sum_{x,y} T = 1$$

and convolved with image intensity values at corresponding image locations. The use of a Gaussian cross-sectional profile is justified by the fact that the diameter of neuronal branches is often smaller than the imaging resolution, resulting in image structures reflecting the point-spread function of the microscope, which can be accurately modeled by a Gaussian. For each location, application of the set of filters with different angles results in an angular response profile $p(\alpha; x, y, k, D)$ characterizing the local image structure (Fig. 2c).

2.2. Peak detection and association

Rotational invariance and the ability to adapt to different configurations are accomplished by detecting peaks in the profile at each image location. The peaks extracted from $p(\alpha; x, y, k, D)$ represent radial directions (red, green, blue clusters in Fig. 2b) in which relatively high intensities are found in the image. Peak detection is accomplished using a line search optimization scheme [9] that iteratively moves a set of abscissa points until each of them converges to one of the profile extrema. Profiles are generally sufficiently smooth for the line searching not to be affected by outliers or noise. Iterative procedures such as line searching are convenient since the number of points converging to clustered peaks provides quantitative information that can be used to rank and identify the strongest peaks for further analysis.



(a) Set of oriented filters. (b) Profile peak loci.

(c) Typical $p(\alpha; x, y, k, D)$ profile with peak detections.

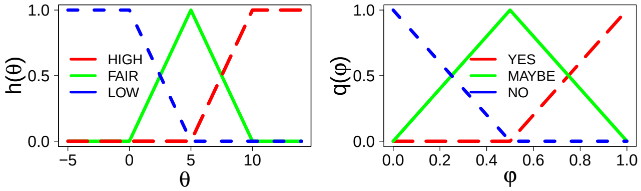
Fig. 2. Profile extraction and analysis.

Each profile peak corresponds to an angle α_p that can be used to calculate a follow-up location $[x_p, y_p] = [x, y] + kD[\cos\alpha_p, \sin\alpha_p]$ in the image where the described profiling and peak detection scheme is repeated. Tracking initiated this way is recursively expanded by extracting the next generation of peak locations starting from already obtained ones. In the case of bifurcations, the majority of expanded peak locations group together in clusters, along elongated image structures (Fig. 2b). Up to four tracks are considered in order to cover both bifurcations and cross-overs.

The final output at this stage are scores ψ that quantify the strength and connectivity of every track step. Here, ψ is defined as the median image intensity calculated along the line connecting the peak locations of a given track. A set of six ψ values is assigned to each image location (x, y), since the three highest-scoring tracks are selected and supplied to the detection system: $\psi_A^{1,2}, \psi_B^{1,2}, \psi_C^{1,2}$, where A, B, C denote the tracks (marked with red, green, blue in Fig. 2b) and $1, 2$ denote track steps counted from the center. Each score is associated with its corresponding peak location, for example $\psi_A^1 \to (x_A^1, y_A^1)$ and $\psi_B^2 \to (x_B^2, y_B^2)$.

2.3. Fuzzy rule based detection

The last stage of the detection algorithm uses the computed scores ψ of each image location to decide whether or not there is a bifurcation at that location. The aim is to deal with uncertainty about the scores. Artificial intelligence theory models uncertainty using probabilities or logical reasoning. Ontologically, both concepts treat events as crisp true or false, but assign a level of (un)certainty. Fuzzy logic systems (FLS) [10], on the other hand, introduce vagueness, allowing events to be partially true or false. Here we employ the FLS concept to grade the likelihood of each analyzed location to be a bifurcation region. We define three fuzzy sets to express an outcome in linguistic terms: YES (Y) if it belongs to a bifurcation region; MAYBE (M) if it possibly belongs to a bifurcation region; NO (N) if it does not belong to a bifurcation region.



(a) Fuzzification, $i_{\text{Diff}} = 10$. (b) Defuzzification.

Fig. 3. $h(\theta)$ and $q(\varphi)$ denote the degree of membership to input (a) and output (b) linguistic variable.

The decision making in this case involves non-linear reasoning. Hence we adopt a rule-based system approach similar to the one applied to solving motion correspondence in biomedical image analysis [11].

The inputs of our FLS are calculated by subtracting the estimated local background value $\beta(x, y)$ from the scores ψ obtained in the previous stage:

$$\theta_s^t = \psi_s^t - \beta(x_s^t, y_s^t), \quad s \in \{A, B, C\}, \quad t \in \{1, 2\}$$

where $\beta(x, y)$ at the given image location (x, y) is estimated as the median of the image intensities extracted from a local circular neighborhood, with radius equal to three times the value of the expected neuron diameter D to capture enough context. Numerical values θ express how different the median intensity along a given track segment is compared to the local background. Values higher than a predefined threshold level i_{Diff} indicate the existence of relevant image structure. θ is assigned a membership value for each linguistic variable, ω, with possible values $\{$HIGH, FAIR, LOW$\}$, abbreviated as $\{$H, F, L$\}$. FLS takes as input linguistic variables and outputs linguistic variable $\upsilon \in \{$Y, M, N$\}$ grading the bifurcation detection outcome. The system allows for tolerance on i_{Diff} and avoids strict mathematical formulation of the nonlinear input/output relationships by using a set of essential rules (Table 1) to associate fuzzy linguistic variables.

Input numerical values are assigned to linguistic variables (the "fuzzification" step) using common triangular membership functions (Fig. 3a). A set of IF-THEN rules is then used to determine the value of the output variable (Table 1). Rules are employed using fuzzy set operations [10] and are combined to yield output fuzzy values through an inference process that accumulates the results of individual rules using the maximum algorithm [10]. The membership function of the output value (Fig. 3b) together with the commonly used centroid "defuzzification" procedure [10] are used to convert the produced fuzzy value into a crisp output φ. The latter output is a real number, $\varphi \in [0, 1]$, corresponding to the abscissa range in the defuzzification membership function (Fig. 3b). This crisp output is what we will refer to as *bifurcationess*: the degree of membership to the fuzzy set of bifurcation points denoted with the linguistic term YES. Our method ultimately looks for connected regions with $\varphi > 0.6$.

Table 1. IF-THEN rules used in the fuzzy logic system emulate the decision-making of a human expert. Fuzzy set operations union ($\cup$) and intersection ($\cap$) operating on the membership values, are defined as *max* and *min* respectively. $s, s_1, s_2 \in \{A, B, C\}$ and $t, t_1, t_2 \in \{1, 2\}$.

1. $\bigcap_{s,t}(\theta_s^t = \text{H}) \Rightarrow \upsilon = \text{Y}$
2. $(\theta_s^t = \text{F}) \cap \Big(\bigcap_{(s_1,t_1) \neq (s,t)} (\theta_{s_1}^{t_1} = \text{H}) \Big) \Rightarrow \upsilon = \text{M}$
3. $(\theta_s^t = \text{F}) \cap (\theta_{s_1 \neq s}^t = \text{F}) \cap \Big(\bigcap_{s_2 \neq s, s_1} (\theta_{s_2}^t = \text{H}) \Big) \Rightarrow \upsilon = \text{M}$
4. $\bigcup_{s,t} \Big((\theta_s^t = \text{F}) \cap (\theta_s^{t_1 \neq t} = \text{F}) \Big) \Rightarrow \upsilon = \text{N}$
5. $\bigcup_{s,t} \Big((\theta_s^t = \text{F}) \cap (\theta_{s_1 \neq s}^{t_1 \neq t} = \text{F}) \cap (\theta_{s_2 \neq s, s_1}^{t_2 \neq t, t_1} = \text{F}) \Big) \Rightarrow \upsilon = \text{N}$
6. $\bigcup_{s,t}(\theta_s^t = \text{L}) \Rightarrow \upsilon = \text{N}$

2.4. Implementational aspects

Microscopy images of neuronal cells are usually very sparse. Depending on the experiment, the relevant image structures may cover only 5%-50% of the image area, suggesting a computational speed-up by ignoring regions that evidently belong to the background. In our implementation of the method all image locations (x, y) with intensity lower than $\beta(x, y) + i_{\text{Diff}}$ are excluded from the analysis and assigned $\upsilon = 0$. The method was implemented in Java as a plugin for ImageJ [12]. On a standard PC configuration with Intel Core i7 8 × 2 GHz CPU, 6 GB RAM, it requires about 8 seconds to process a 512 × 512 pixel image with 10% of foreground. Execution time is proportional to the amount of foreground in image. Computationally demanding tasks such as kernel convolution and iterative peak search are implemented in parallel.

3. RESULTS

The performance of the proposed method was evaluated using both synthetic images and manually annotated real neuron images. Parameters D and i_{Diff} were set to 4 and 10 respectively. Synthetic data contained artificial bifurcations generated as intersections of linear segments and consisted of seven subsets, Ci, $i = 1 \ldots 7$ (Table 2), each representing a different combination of branch diameters and random splitting angles between branch segments. Each subset contained 1,000 bifurcations with known ground-truth positions and branch diameters ranging from 3 to 7 pixels. Three different signal-

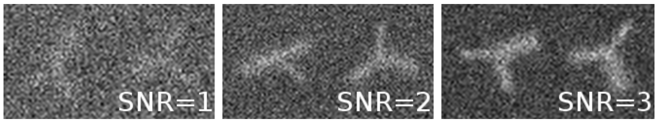


Fig. 4. Synthetic bifurcations for different SNR.

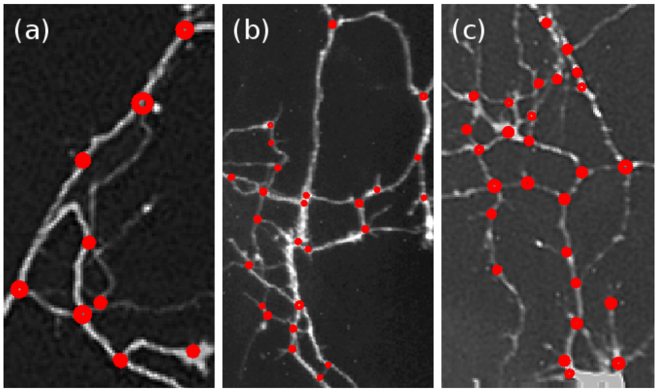


Fig. 5. (a, b, c) Bifurcation detection on different neurons.

to-noise ratios (SNR) were considered (Fig. 4) to evaluate the robustness of the method to (Poisson) noise.

Bifurcation detection performance was assessed in terms of the amount of true-positive (TP), false-positive (FP), and false-negative (FN) detections for different subsets and SNR level (Table 2). From these we computed the recall, $\mathrm{R} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, and precision, $\mathrm{P} = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$. We observed that the detection performance is very high for SNR $\geq$ 2 but drops significantly for SNR ≈ 1 (Table 2). Since real image data usually contains less noise we conclude that the method is practically robust to noise.

The method was also tested on a selection of nine fluorescence microscopy neuron images with a total of 724 annotated bifurcations. Performance was measured in the same manner as with the synthetic data and showed consistently high recall and precision with average of 90.4% and 90.5% respectively (Table 2). Illustrative examples of detection results are shown in Fig. 5. The neuron tracing literature, which often lacks information on bifurcation detection performance, reports a detection rate of 86% in a recent study [5].

4. DISCUSSION

We have presented a new method for bifurcation detection in fluorescence microscopy images of neuronal cells. The method finds the number of peaks in local directional filter responses and identifies the bifurcation configuration. Peak detection and weighting is implemented using an efficient iterative line search method. In addition, a fuzzy rule-based decision scheme is used to assign a fuzzy bifurcationess score to each location. Experimental results with synthetic data show that the method can cope with SNR as low as 2 and that it has high recall and precision of around 90%. Although in this paper we have focused on 2D, the proposed method is expandable to 3D. Also, while we have focused here on neuron image analysis, we expect the method to be useful for other applications too, including retinal and vascular image analysis. This will be the subject of future work.

Table 2. Detection performance for synthetic and real neuron images. Synthetic bifurcations are random angular configurations, Ci, $i = 1 \dots 7$, with different combinations of branch diameters, $D = 3, 5, 7$ pixels, evaluated for SNR $= 1, 2, 3$. Listed are recall (R) and precision (P) in percent.

SYNTHETIC IMAGE DATA (1,000 bifurcations)

		D	D D	D	DDD	D D	D D	D
SNR		C1	C2	C3	C4	C5	C6	C7
1	R	23	22	20	14.3	10.1	11.5	7
	P	100	100	100	100	100	100	100
2	R	96.2	94	96.3	90	89.5	94.7	94.4
	P	100	100	100	100	100	100	100
3	R	99.1	97.5	97.8	92.5	90.8	93.2	94
	P	100	100	100	100	100	100	100

REAL IMAGE DATA (♯ = number of bifurcations)

	I	II	III	IV	V	VI	VII	VIII	IX
R	92	91	88	91	87.5	94.2	97.2	82.3	90
P	95	91	83	90	97.2	89.1	92.2	86.7	90
♯	43	80	106	100	41	39	159	134	22

5. REFERENCES

[1] E. Meijering, "Neuron tracing in perspective," *Cytometry Part A*, vol. 77, no. 7, pp. 693–704, 2010.

[2] Y. Wang, A. Narayanaswamy, C. Tsai, and B. Roysam, "A broadly applicable 3-D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, no. 2-3, pp. 193–217, 2011.

[3] P. Chothani, V. Mehta, and A. Stepanyants, "Automated tracing of neurites from light microscopy stacks of images," *Neuroinformatics*, vol. 9, no. 2-3, pp. 263–278, 2011.

[4] Z. Vasilkoski and A. Stepanyants, "Detection of the optimal neuron traces in confocal microscopy images," *Journal of Neuroscience Methods*, vol. 178, no. 1, pp. 197–204, 2009.

[5] Y. Al-Kofahi, N. Dowell-Mesfin, C. Pace, W. Shain, J. N. Turner, and B. Roysam, "Improved detection of branching points in algorithms for automated neuron tracing from 3D confocal images," *Cytometry Part A*, vol. 73, no. 1, pp. 36–43, 2008.

[6] Z. Püspöki, C. Vonesch, and M. Unser, "Detection of symmetric junctions in biological images using 2-D steerable wavelet transforms," in *Proceedings of the IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2013, pp. 1488–1491.

[7] A. Choromanska, S. Chang, and R. Yuste, "Automatic reconstruction of neural morphologies with multi-scale tracking," *Frontiers in Neural Circuits*, vol. 6, no. 25, 2012.

[8] Y. Liu, "The DIADEM and beyond," *Neuroinformatics*, vol. 9, no. 2-3, pp. 99–102, 2011.

[9] W. H. Press, S. A. Teukolsky, W. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, NY, 1992.

[10] J. M. Mendel, "Fuzzy logic systems for engineering: a tutorial," *Proceedings of the IEEE*, vol. 83, no. 3, pp. 345–377, 1995.

[11] S. Jiang, X. Zhou, T. Kirchhausen, and S. T. C. Wong, "Tracking molecular particles in live cells using fuzzy rule-based system," *Cytometry Part A*, vol. 71, no. 8, pp. 576–584, 2007.

[12] M. D. Abràmoff, P. J. Magalhães, and S. J. Ram, "Image processing with ImageJ," *Biophotonics International*, vol. 11, no. 7, pp. 36–42, 2004.