Steerable Features for Statistical 3D Dendrite Detection

Germán González[1,⋆], François Aguet[2], François Fleuret[1,3,⋆⋆], Michael Unser[2], and Pascal Fua[1]

[1] Computer Vision Lab, Ecole Polytechnique Fédérale de Lausanne, Switzerland
german.gonzalez@epfl.ch
[2] Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne, Switzerland
[3] Idiap Research Institute, Martigny, Switzerland

Abstract. Most state-of-the-art algorithms for filament detection in 3–D image-stacks rely on computing the Hessian matrix around individual pixels and labeling these pixels according to its eigenvalues. This approach, while very effective for clean data in which linear structures are nearly cylindrical, loses its effectiveness in the presence of noisy data and irregular structures.

In this paper, we show that using steerable filters to create rotationally invariant features that include higher-order derivatives and training a classifier based on these features lets us handle such irregular structures. This can be done reliably and at acceptable computational cost and yields better results than state-of-the-art methods.

1 Introduction

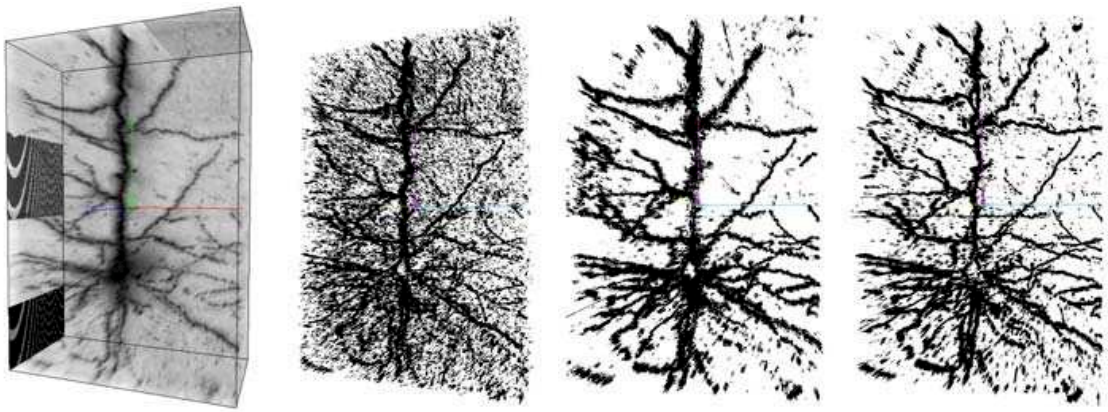
Most state-of-the-art approaches to filament detection in 3–D image-stacks rely on computing the Hessian matrix around individual voxels and labeling these voxels according to its eigenvalues. Some are optimized for ideal tubular structures, while others use statistical-learning techniques to improve detection results.

In this paper, we will show that the second-order derivatives used to compute the Hessian matrix do not provide a local description that is powerful enough to account for the fact that dendrites, such as those depicted by Fig. 1, are far from being regular tubular structures, which can drastically impact performance. To effectively account for such irregularities, one must use higher-order derivatives.

To this end, we rely on 3–D steerable filters [1] to create rotationally invariant features that include derivatives of order 2 to 4 that we use as input to a classifier trained to recognize voxels belonging to potentially irregular dendrites. Because the training data encompasses the deviations from the ideal model, the resulting algorithm has the potential to be more robust than traditional ones and can be trained to detect not only simple linear-structures but also junctions and crossings.

⋆ This work was funded in part by the Swiss National Science Foundation.
⋆⋆ Supported by the Swiss National Science Foundation under the National Centre of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

G.-Z. Yang et al. (Eds.): MICCAI 2009, Part II, LNCS 5762, pp. 625–632, 2009.
© Springer-Verlag Berlin Heidelberg 2009



(Original) (Frangi) (Second Order) (Fourth Order)

Fig. 1. Original image stack and its segmentation result for a false positive rate of 10^{-2} for three different methods: Frangi's non-linear relationship on the eigenvalues of the hessian [2], our methodology with second order features, and our methodology with fourth order features. As it will be demonstrated later, the fourth order method outperforms the other two ones.

Most automated approaches to finding linear structures in image stacks assume them to be locally tubular and model them as generalized cylinders. The most popular one involves computing the Hessian matrix at individual voxels by convolution with Gaussian derivatives and relying on the eigenvalues of the Hessian to classify voxels as filament-like or not [3,2,4]. The Hessians can be modified to create an oriented filter in the direction of minimum variance, which should correspond to the direction of any existing filament [5,1]. To find filaments of various widths, these methods perform the computation using a range of variances for the Gaussian masks and select the most discriminant one. The fact that intensity changes inside and outside the filaments has also been explicitly exploited by locally convolving the image with differential kernels [6], finding parallel edges [7], and fitting *superellipsoids* or cylinders to the linear structure based on its surface integral [8,9].

All these methods, however, assume image regularities that are present in high-quality images but not necessarily in noisier ones. Furthermore, they often require careful parameter tuning, which may change from one data-set to the next. As a result, probabilistic approaches able to learn whether a voxel belongs to a filament or not have begun to be employed. Instead of assuming the filaments to be cylinders, they aim at learning their appearance from the data. In [10], the eigenvalues of the structure tensor, are represented by a mixture model whose parameters are estimated via E-M. Support Vector Machines that operate on the Hessian's eigenvalues have also been used to discriminate between filament and non-filament voxels [11].

The latter approach [11] is closest to ours in that it also relies on the statistical learning paradigm. However, its ability to generalize is limited by the fact that it still relies on the eigenvalues of the Hessian and therefore on second order derivatives, whereas using higher-order derivatives gives us access to a much richer descriptor.

As shown in Fig. 2, our method outperforms one of the very best Hessian-based methods [2]. Interestingly, this stops being true if we limit it to using only second-order derivatives as opposed to fourth-order ones and our results actually become worse. In other words, these higher-order derivatives are required to take full advantage of the statistical-learning framework that has been advocated in the literature [10,11]. Furthermore, steerable filters provide a very effective framework to do this robustly. We view this observation as the main contribution of the paper.

2 Method

To demonstrate that higher-order derivatives provide better descriptive power at an acceptable computational cost, we rely on 3–D steerable filters [1] to create rotationally invariant feature vectors that can be used to classify voxels as being part of a dendrite or not. In practice, to achieve rotation invariance, we compute a local orientation and use it to steer the filters and to create the feature vectors corresponding to a reference orientation. In other words, we rotate the feature vectors to a reference orientation.

In the remainder of this section, we first recall the basic theory of steerable filters. We then show how we use them to create feature vectors given a local orientation estimate. Finally, we discuss how we use these feature vectors to train the classifier we use at run time to detect filament-like voxels.

2.1 Steerable Filters in 2–D and 3–D

Steerable filters were introduced as an efficient means to compute filters that can be rotated to any orientation for a small computational cost [12]. In three dimensions, steerable filter based detection of a feature g in a volume f at a given orientation and position $\mathbf{u} = (x, y, z)$, is formulated as:

$$r = f(\mathbf{u}) * g(\mathbf{R}^{\theta,\phi}\mathbf{u}), \quad g(\mathbf{R}^{\theta,\phi}\mathbf{u}) = \sum_l b_l(\theta,\phi) g_l(\mathbf{u}), \tag{1}$$

where θ and ϕ parameterize the orientation of the feature template in spherical coordinates, $\mathbf{R}_{\theta,\phi}$ is the 3–D rotation matrix, and r is the response. The functions $b(\theta,\phi)$ are trigonometric polynomials that interpolate the templates $g_l(\mathbf{u})$. This decomposition decouples the rotation of the filters from the convolution in Eq. (1), which makes the estimation computationally efficient.

The best known class of such filters, and the ones used in this paper, are Gaussian derivatives and their linear combinations [13]. To preserve the separability of the resulting kernels, we limit ourselves to diagonal covariance matrices. Let G^σ denote the isotropic Gaussian kernel of variance σ centered at the origin. Let $G^\sigma_{m,n,p}$ denote it m^{th} derivative with respect to x, n^{th} derivative with respect to y and p^{th} derivative with respect to z.

$$\forall \mathbf{u} \in \mathbb{R}^3,\ G^\sigma(\mathbf{u}) = \frac{1}{(2\pi\sigma^2)^{3/2}}\ \exp\left(-\frac{\|\mathbf{u}\|^2}{2\sigma^2}\right),\ G^\sigma_{m,n,p} = \frac{\partial^{m+n+p} G^\sigma}{\partial x^m \partial y^n \partial z^p}. \tag{2}$$

The rotation equations for a filter that is formed by a linear combination of Gaussian derivatives is:

$$b_{m,n,p}(\theta,\phi) = \sum_{i=0}^{m}\sum_{k=0}^{n}\sum_{q=0}^{p}\sum_{j=0}^{i}\sum_{l=0}^{k} \frac{m!n!p!(-1)^{i-j+p-q}}{(m-i)!(i-j)!j!(n-k)!(k-l)!l!(p-q)!q!} \cos(\theta)^{m-i+j+k-l}\cos(\phi)^{m-i+n-k+q}\sin(\theta)^{i-j+n-k+l}\sin(\phi)^{j+l+p-q} a_{m-i+n-k+p-q,i-j+k-l,j+l+q} \quad (3)$$

where $a_{m,n,p}$ is the coefficient that multiplies $G^{\sigma}_{m,n,k}$ at the reference orientation.

2.2 Feature Vectors

We take the features vectors to be the convolution of the volume f with the set of templates $G^{\sigma}_{m,n,p}$ of normalized energy,

$$v_{\sigma}(f,\mathbf{u}) = \left(f * \left[\frac{G^{\sigma}_{1,0,0}}{E_{1,0,0}}, \frac{G^{\sigma}_{0,1,0}}{E_{0,1,0}}, \frac{G^{\sigma}_{0,0,1}}{E_{0,0,1}}\frac{G^{\sigma}_{2,0,0}}{E_{2,0,0}} \cdots \frac{G^{\sigma}_{0,0,M}}{E_{0,0,M}}\right]\right)(\mathbf{u}) \ , \quad (4)$$

where $E_{i,j,k}$ is the energy of the $G^{\sigma}_{i,j,k}$ function. These feature vector are equivalent to a steerable filter, and therefore can be steered to any orientation using Eq. (3).

2.3 Training and Detection

During a training phase, we use ground truth data for which the orientation is provided to train an SVM classifier. Then, to classify a voxel at run-time, we compute the local orientation to rotate the feature vectors to the reference orientation. Finally, the classifier is used to output the likelihood of the voxel belonging to the neuron.

The training data consists of quadruplets that include a 3–D location $\mathbf{u}$ in an image stack, two orientation angles θ and ϕ, and a single bit indicating whether it is a positive or negative sample. Formally, the training set can be written as

$$\mathcal{S} = \{(\mathbf{u}_1,\theta_1,\phi_1,1),\ldots,(\mathbf{u}_N,\theta_N,\phi_N,1), (\mathbf{u}_{N+1},\theta_{N+1},\phi_{N+1},0),\ldots,(\mathbf{u}_{2N},\theta_{2N},\phi_{2N},0)\} \quad (5)$$

where the first N quadruplets represent the positive samples and the following N the negative ones.

The positive samples are taken from ground truth data. Negative samples are taken from two populations. The first one includes points closer than a given radius to the dendrites but not belonging to them, the second one are points taken at random in the whole volume, but not belonging to the neuron. The local orientation of negative points is given by the same algorithm as the one used during detection, which in our case is the steerable filters optimized using Canny criteria for filament detection of [1].

For each of the training points, the feature vector is computed and rotated back from its labeled orientation. Let $v^{\theta,\phi}$ be the feature vector rotated by angles θ and ϕ. The set of samples used for training is

$$\mathcal{V} = \{(v_1^{-\theta_1,-\phi_1}, 1), \ldots, (v_N^{-\theta_N,-\phi_N}, 1), (v_{N+1}^{-\theta_{N+1},-\phi_{N+1}}, 0), \ldots, (v_{2N}^{-\theta_{2N},-\phi_{2N}}, 0)\} \ . \tag{6}$$

After training an SVM, the detection score for a voxel $\mathbf{u}$ at orientation θ, ϕ becomes

$$\Psi : \mathbb{R}^D \rightarrow \mathbb{R} \ , \quad \psi(\mathbf{u}, \theta, \phi) = \sum_{n=0}^{N} a_n \, \kappa \left(v_n, v^{-\theta,-\phi}(\mathbf{u})\right) + b \ , \tag{7}$$

where κ is the standard Gaussian kernel, the variance ν of which is obtained by minimizing the error on a validation set.

At run-time, to classify a voxel as belonging to a filament or not, we need to estimate the orientation of that filament if it exists. In standard Hessian methods, this is done by computation the eigenvectors of the Hessian matrix. However, there is no obvious method to do the same using our feature vectors. To derive the orientations we need, we therefore use a modified linear Hessian method that relies on second-order steerable filters optimized according to the Canny criterion [1]. We have found empirically that the orientations it returns allow us to achieve better performance than when using other methods.

3 Results

In this section, we show that using fourth order steerable features allow us to detect dendrites more accurately in brightfield microscopy images than second order methods. We compare our method to both [2], which we believe to be one of the best Hessian-based methods, and to our own algorithm constrained to use only second-order derivatives. The ROC curve of Fig. 2 summarizes our findings.

The dataset used for these comparisons consists of two image stacks of neurons imaged using standard brightfield microscopy and the associated ground truth data. The first is used for training and validation and the second for testing. Fig. 1(Original) is a 3–D minimum intensity projection of the test stack. Cross-sections in the XY and XZ planes are shown in Figs. 3 and 4. In Fig. 4, please note the cone of shadow cast by the dendrites, which causes problems to second order filament detectors.

In the remainder of this section, we first describe implementation details of the training and detection procedures, and offer a more in-depth analysis of our results.

3.1 Training and Detection

For training we used as positive samples 2500 hand-labeled voxels and their associated orientations, and 2500 more for validation purposes. In addition, we collected 1250 negative samples around the neuron and 1250 chosen at random,

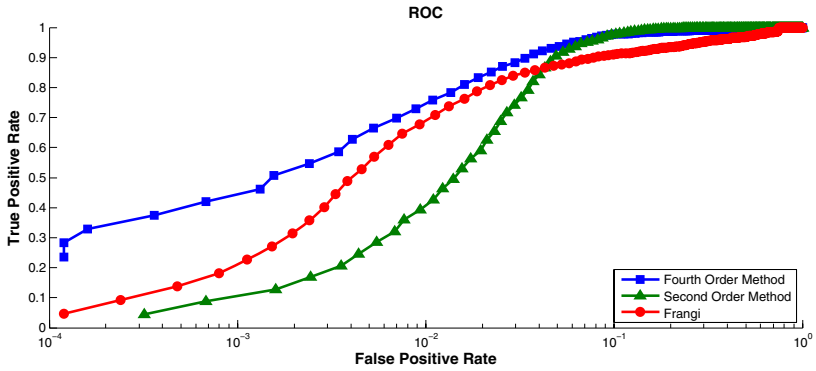
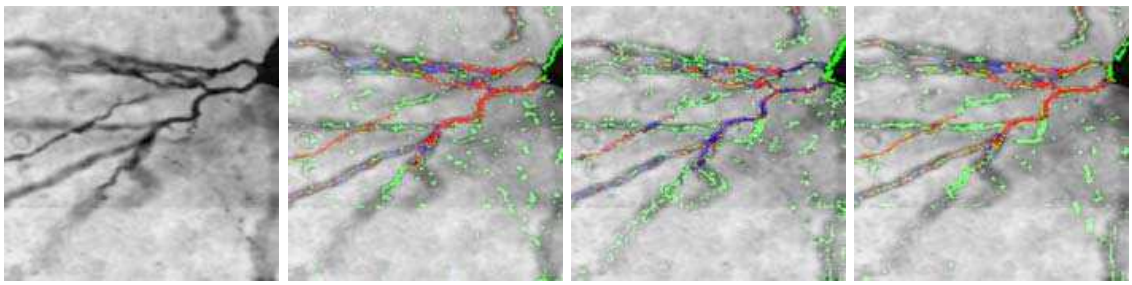


Fig. 2. ROC curve for several methods. The presented method of order four outperforms the method of Frangi [2] and our algorithm constrained to second order features. The improvement is due to the use of fourth order features, which allows us to emcompass a higher frequency signals. Please not the logarithmic scale in the false positive rate.



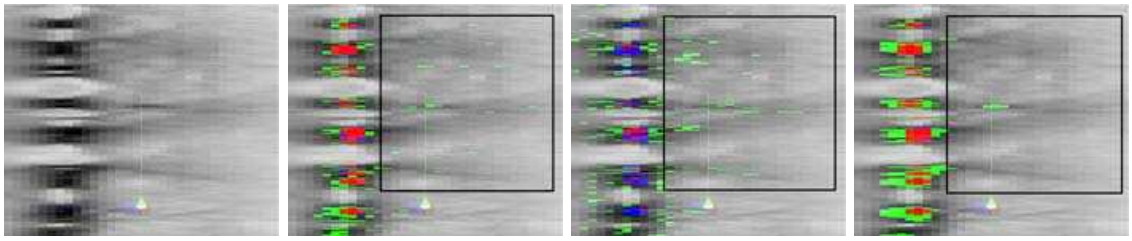
(Original) (Frangi) (Second Order) (Fourth Order)

Fig. 3. Detail of one of the images in the stack. We compare our detector using features of order four and two against that of Frangi [2]. In red we show true positives, in green false positives and in blue false negatives. The false positive rate is fixed to 10^{-2}. Our second-order detector fails to detect some filaments and is more sensitive to the shadow casted by the dendrites. However, our fourth order filter outperforms the second order Hessian-based method of Frangi. False positives for our method are clustered around the true dendrite locations, and the true positive rate is incremented from 67.7% to 74.5%.

but not belonging to the dendrites. The orientation of the negative samples is taken from the output of the orientation predictor [1], using same scale as the one used to compute the feature vectors. We use the method of [1] to compute the orientation as it is a more elongated template than the Hessian and provides a more accurate orientation estimation. During detection, the orientation is estimated using the same method as for assigning orientation to negative points.

3.2 Discussion

The ROC curve of Fig. 2 indicates that our fourth order filter outperform the second order methods over the entire range of false positive rates. This comes from the fact that fourth order derivatives can encompass higher frequency



(Original) (Frangi) (Second Order) (Fourth Order)

Fig. 4. Detail of the YZ projection of several filaments in parallel. As in Fig. 3, we show true positives in red, false positives in green and false negatives in blue. Our method produces ellipsoids centered in the true filament but bigger in size. This is due to uncertainties in the training data. Frangi's method [2] detects the real dendrites accurately but produces many false-positives away from them. Our second order method responds mainly to the shadow of the dendrite in this cross-section. Please note the noise in the different images highlighted by the rectangles.

signals. For example, this is what explains that our fourth order method avoids confusion between the cone of shadow of the signal and the actual dendrite in the difficult case of Fig. 4.

Our features are linear combinations of SVM kernel functions evaluated at the support vectors. As we are using Gaussian kernels, these functions are smooth. By contrast, [2] uses ratios between features. This creates a sharper detection profile that makes the method more discriminative at low false positive rates than ours when using only second-order derivatives.

4 Conclusion

In this paper we have presented an approach to detecting dendrites in 3–D image stacks that outperforms state-of-the-art Hessian based methods in brightfield image stacks. The performance gain is due to the use of a rich feature set made of higher-order image derivatives. At the heart of our implementation are steerable filters that let us rotate the feature vectors to a reference orientation and train a classifier to recognize which ones correspond to dendrite voxels.

This approach is very generic because, instead of postulating *a priori* models for the filaments we are looking for, our algorithm can learn specific appearance models for each new situation. In future work, we will therefore extend our approach to other imaging modalities in which the filaments break the perfectly tubular structure assumption.

References

1. Aguet, F., Jacob, M., Unser, M.: Three-dimensional feature detection using optimal steerable filters. In: Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP 2005), Genova, Italy, September 11-14, vol. II, pp. 1158–1161 (2005)

2. Frangi, A.F., Niessen, W.J., Vincken, K.L., Viergever, M.A.: Multiscale vessel enhancement filtering. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 130–137. Springer, Heidelberg (1998)
3. Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R.: 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. Medical Image Analysis 2, 143–168 (1998)
4. Streekstra, G., van Pelt, J.: Analysis of tubular structures in three-dimensional confocal images. Network: Computation in Neural Systems 13(3), 381–395 (2002)
5. Meijering, E., Jacob, M., Sarria, J.C.F., Steiner, P., Hirling, H., Unser, M.: Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. Cytometry Part A 58A(2), 167–176 (2004)
6. Al-Kofahi, K., Lasek, S., Szarowski, D., Pace, C., Nagy, G., Turner, J., Roysam, B.: Rapid automated three-dimensional tracing of neurons from confocal image stacks. IEEE Transactions on Information Technology in Biomedicine (2002)
7. Dima, A., Scholz, M., Obermayer, K.: Automatic segmentation and skeletonization of neurons from confocal microscopy images based on the 3-d wavelet transform. IEEE Transactions on Image Processing 11(7), 790–801 (2002)
8. Tyrrell, J., di Tomaso, E., Fuja, D., Tong, R., Kozak, K., Jain, R., Roysam, B.: Robust 3D modeling of vasculature imagery using superellipsoids. Medical Imaging 26(2), 223–237 (2007)
9. Schmitt, S., Evers, J.F., Duch, C., Scholz, M., Obermayer, K.: New methods for the computer-assisted 3D reconstruction of neurons from confocal image stacks. NeuroImage 23, 1283–1298 (2004)
10. Agam, G., Wu, C.: Probabilistic modeling-based vessel enhancement in thoracic ct scans. In: CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Washington, DC, USA, vol. 2, pp. 684–689. IEEE Computer Society, Los Alamitos (2005)
11. Santamaría-Pang, A., Colbert, C.M., Saggau, P., Kakadiaris, I.A.: Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging. In: Ayache, N., Ourselin, S., Maeder, A. (eds.) MICCAI 2007, Part II. LNCS, vol. 4792, pp. 486–494. Springer, Heidelberg (2007)
12. Freeman, W., Adelson, E.: The design and use of steerable filters. Pattern Analysis and Machine Intelligence 13, 891–906 (1991)
13. Jacob, M., Unser, M.: Design of steerable filters for feature detection using canny-like criteria. Pattern Analysis and Machine Intelligence 26(8), 1007–1019 (2004)