Learning Rotational Features for Filament Detection

Germán González*
CVLab, EPFL
EPFL-I&C -CvLab
CH-1015 Lausanne,
Switzerland
german.gonzalez@epfl.ch

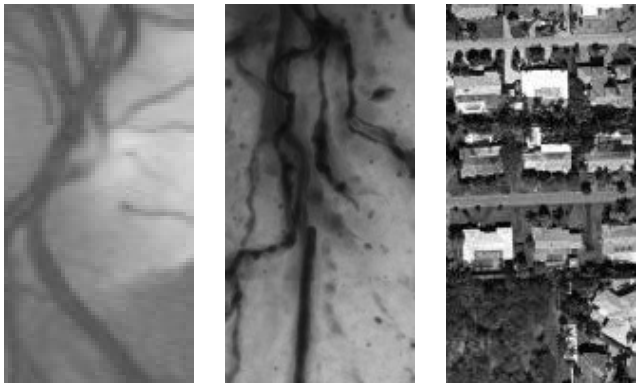
François Fleuret†
IDIAP Research Institute
Rue Marconi 19,
CH-1920 Martigny,
Switzerland
francois.fleuret@idiap.ch

Pascal Fua
CVLab, EPFL
EPFL-I&C -CvLab
CH-1015 Lausanne,
Switzerland
pascal.fua@epfl.ch

Abstract

State-of-the-art approaches for detecting filament-like structures in noisy images rely on filters optimized for signals of a particular shape, such as an ideal edge or ridge. While these approaches are optimal when the image conforms to these ideal shapes, their performance quickly degrades on many types of real data where the image deviates from the ideal model, and when noise processes violate a Gaussian assumption.

In this paper, we show that by learning rotational features, we can outperform state-of-the-art filament detection techniques on many different kinds of imagery. More specifically, we demonstrate superior performance for the detection of blood vessel in retinal scans, neurons in brightfield microscopy imagery, and streets in satellite imagery.



(a) (b) (c)

Figure 1. Examples of complicated filament structures. (a) Blood vessels in retinal scans. (b) Dendrites in brightfield microscopy. (c) Streets in a neighborhood.

1. Introduction

State-of-the-art approaches to detecting linear structures of significant width rely on ideal models of their appearance and of the noise processes. They are usually optimized to find ideal lines or tubular structures with smooth profiles. However, real linear structures such as those of Fig. 1 often do not conform to this model, which can drastically impact performance.

In this paper, we use the rotational properties of Gaussian derivatives to achieve rotational invariance as in [9, 11], However, instead of steering the filters, we rotate the image features to a common reference and replace the optimality criteria by a machine-learning algorithm that learns from training data. Because this data encompasses the deviations from the ideal model, the resulting algorithm is more robust than traditional ones and can be trained to detect not only simple linear-structures but also junctions and crossings. As a result, we obtain better performance than [7] and [9] for the detection of blood vessels in retinal scans, neurons in brightfield microscopy imagery, and streets in satellite imagery. We chose these two methods as our baseline because they are widely acknowledged as being among the best in their respective classes.

2. Related Work

Current approaches to finding linear structures that are not simple edges fall into two main categories. Some rely on models of an ideal ridge-like structure to derive optimal filters while others involve computing the Hessian matrix centered on individual pixels and relying on its eigenvalues to classify the pixel on the basis of how cylinder-like the local intensities are. We discuss these two classes below. For an extensive review on vessel extraction techniques, we refer the reader to [10].

*Supported by the European project PEGASE No AST5-CT-2006-030839.

†Supported by the Swiss National Science Foundation under the National Centre of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

2.1. Optimal Filtering

Optimal filters attempt to find linear image structures by following the criteria for optimal edge detection outlined in [4]. Methods following this approach include the Canny detector [4], and optimal convolution filters [12].

An elegant and computationally efficient approach to detect edges or ridges at any given orientation is the use of steerable filters [8]. The underlying principle is that the response of the filter at any orientation can be calculated as a linear combination of a basis of filters, thus avoiding the expensive convolutions with orientation-specific 2D kernels. A special case of steerable filters is the linear combination of Gaussian derivatives up to a given order [9, 11].

However, the criteria used to derive the steerable filters for ridge detection assumes ideal models of the ridges and noise. In theory, more realistic models could be used without changing the overall approach, but no generic strategy has been offered as to how this should be done.

2.2. Hessian-Based Approaches

Hessian-based approaches to filament detection model them as being elongated elliptical structures. This involves computing the eigen-decomposition of the Hessian matrix at individual pixels and relying on the eigenvalues of the Hessian to classify pixels as filament-like or not [14, 7, 18]. The Hessian matrix for a given pixel is constructed by convolving the local image patch with a set of Gaussian derivative filters. The Hessian can be modified to create an oriented filter in the direction of minimum variance, which should correspond to the direction of any existing filament [11]. To find filaments of various widths, a range of variances for the Gaussian is used and the most discriminant one is selected. The fact that intensity changes inside and outside the filaments has also been explicitly exploited by locally convolving the image with differential kernels [3], finding parallel edges [5], and fitting *superellipsoids* to the vessel based on its contour integral [19, 15]. All these methods, however, assume image regularities that are present in well-behaved images but not necessarily in noisier ones. Furthermore, they often require careful parameter tuning, which may change from one data-set to the next.

Recently, probabilistic approaches able to learn whether a pixel belongs to a filament or not have been applied to the problem. Instead of assuming the filaments to be ellipsoidal, they aim at learning their appearance from the data. In [1], the eigenvalues of the structure tensor, are represented by a mixture model whose parameters are estimated via Expectation Maximization. Support Vector Machines operating on the Hessian's eigenvalues have also been used to discriminate between filament and non-filament pixels [13]. Probabilistic Boosting Trees with rotational features have also been used for vessel segmentation [16]. The main difference with our rotational features is that ours are estimated densely around the pixel under analysis, while theirs are sparsely sampled points.

The approach in [13] is closest to ours because it also relies on a learning paradigm. However, its ability to generalize is limited by the fact that it still relies on the eigenvalues of the Hessian, a low-dimensional descriptor, whereas we train our classifier directly on the space of gaussian derivatives, thereby making fewer assumptions and allowing our classifier to handle structures whose shape is more variable.

3. Methodology

Our goal is to devise an algorithm that detects filament-like structures of interest at any orientation or scale while rejecting the noise present in the images.

Our algorithm is as follows. For each pixel we compute multi-scale rotational features using Gaussian derivatives of various widths. Then, we train an SVM classifier on those feature vectors rotated to a canonical orientation. This classifier is used to classify filament-like structures, as depicted by Fig. 2.

3.1. Feature Vector

More precisely, let G^σ denote the symmetric centered Gaussian kernel of variance σ

$$\forall z \in \mathbb{R}^2,\ G^\sigma(z) = \frac{1}{2\pi\sigma^2}\ \exp\left(-\frac{\|z\|^2}{2\sigma^2}\right), \qquad (1)$$

This function and all its derivatives are separable in x and y due to their diagonal covariance matrix. Let $G^\sigma_{i,j}$ be the i^{th}derivative of the Gaussian kernel with respect to x and the j^{th} derivative with respect to y,

$$G^\sigma_{i,j} = \frac{\partial^{i+j} G^\sigma}{\partial x^i \partial y^j}. \qquad (2)$$

Then, the feature vector of dimension $d = \frac{1}{2}(M+3)M$ can be written as the value at z of the convolution of the image by the Gaussian derivatives:

$$v^\sigma(I,z) = \left(I * \left[\frac{G^\sigma_{0,1}}{E_{0,1}}, \frac{G^\sigma_{1,0}}{E_{1,0}}, \frac{G^\sigma_{2,0}}{E_{2,0}} \cdots \frac{G^\sigma_{0,M}}{E_{0,M}}\right]\right)(z) \quad (3)$$

where $E_{i,j}$ is the energy of the $G^\sigma_{i,j}$ function.

To achieve scale independence, we extend the feature vector of each point by adding features to it at S different scales. The full feature vector is of dimension $D = Sd$ and can be written as:

$$v(I,z) = [v^{\sigma_1}(I,z), \cdots, v^{\sigma_S}(I,z)] \qquad (4)$$

This feature vector is the projection of the image around pixel z into the space defined by the Gaussian derivatives.

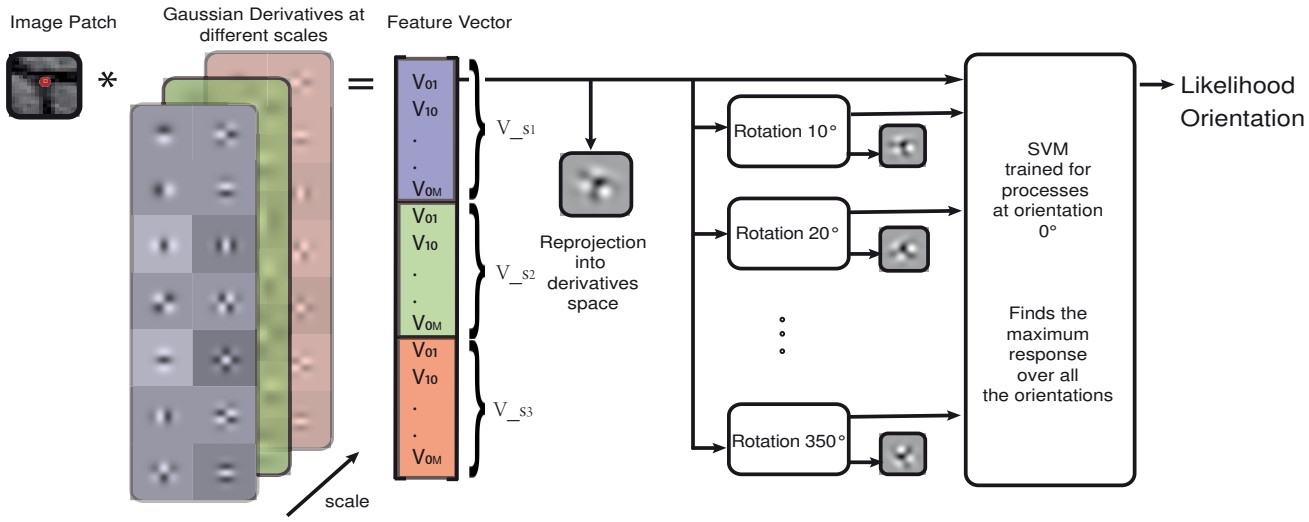


Figure 2. Our approach for filament detection is composed of two steps. First, we compute a feature vector for the image patch that can be rotated to any orientation. The second step is to classify the feature vector as a filament or non-filament using an SVM. This is done by rotating the feature vector to find the orientation with maximum classification score.

3.1.1 Rotating the Feature Vector

Following [9], any vector in the space defined by the Gaussian derivatives can be rotated or steered. By rotation, we mean that the shape of the function at any orientation can be defined as a linear combination of the Gaussian derivatives. The coefficients of that linear combination depend on the desired orientation. For a full explanation of this property we refer the reader to [9].

More precisely, if for an image I, angle θ, and location z in the image frame, I^θ denotes the image after rotation and z^θ the location in the rotated image frame, we have

$$\forall \sigma,\ \forall \theta,\ \exists R^{\theta,\sigma} \in \mathbb{R}^{d\times d},\ \text{such that,}$$
$$\forall I,\ \forall z,\ v^\sigma(I^\theta, z^\theta) = R^{\theta,\sigma}\, v^\sigma(I, z). \quad (5)$$

Hence, extraction of the feature vector at any orientation does not require the evaluation of new linear filter responses, but simply multiplying the vector $v^\sigma(I,z)$ by a $d \times d$ matrix. For each scale, the relation between v^θ and v is [9]:

$$v^\theta_{k,j} = \sum_{i=0}^{k} \sum_{\{l,m\}\in S} v_{k,i} \begin{pmatrix} k-i \\ l \end{pmatrix} \begin{pmatrix} i \\ m \end{pmatrix} (-1)^{k-i-l} cos(\theta)^{i+l-m} sin(\theta)^{k-i-l+m} \quad (6)$$

The set $\mathcal{S}$ can be defined as the set such that $\mathcal{S}(k,j,i) = \{l,m | 0 \le l \le k-i, 0 \le m \le i, k-(l+m) = j\}$.

This rotation of the feature vector is done for all scales. Let R^θ denote the $D \times D$ block matrix corresponding to the same linear operator for the multi-scale feature vector.

3.2. Learning the Shape of the Filaments

Because we can rotate all feature vectors to a canonical orientation, we can train a single general classifier. This classifier is trained as follows:

We select at random N triplets (image, location, orientation) corresponding to filament structures

$$\{(I_1, z_1, \theta_1), \ldots, (I_N, z_N, \theta_N)\}, \quad (7)$$

and N triplets corresponding to non-filaments

$$\{(I_{N+1}, z_{N+1}, \theta_{N+1}), \ldots, (I_{2N}, z_{2N}, \theta_{2N})\}. \quad (8)$$

Then, from the D-dimension feature vectors extracted at these points

$$\forall n,\ v_n = R^{\theta_n}\, v(I_n, z_n) \quad (9)$$

we define a training set, the first N samples of which are of class 1 and the last N of class 0

$$\{(v_1, 1), \ldots, (v_N, 1), (v_{N+1}, 0), \ldots (v_{2N}, 0)\}\ . \quad (10)$$

From that labelled sample set, we train an SVM of the form

$$f : \mathbb{R}^D \to \mathbb{R}$$
$$f(v) = \sum_{n=0}^{N} a_n\, \kappa(v_n, v) + b\ , \quad (11)$$

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

This strategy of training a classifier common to all orientations is a direct application of the idea of data aggregation

through stationary features [6]. We parametrize the features with a complex pose instead of training several classifiers dedicated to constrained subsets of samples. Doing so, we avoid both the computational overhead of training several classifiers and the over-fitting due to the fragmentation of the sample set.

3.3. Detecting Filaments

To build a predictor invariant to rotation, we take as a final prediction score the maximum over all possible orientations of the classifier f. More precisely, for any image I and any location z in the image, the final predicted score is

$$\psi(I, z) = \max_{\theta} f\left(R^{\theta}\, v(I, z)\right), \tag{12}$$

where f denotes the trained SVM, $v(I, z)$ is the D-dimensional feature vector computed in image I at location z and R^{θ} is the rotation linear operator for angle θ defined in Section 3.1.1.

3.4. Relationship with Steerable Filters and Hessian-Based Methods

The image features we use are the same as those proposed in [9] but, because the SVM is nonlinear, there is no analytical criterion to decide by how much the features should be rotated. As a result, we have to sample the space of possible orientations and find the one that produces the greatest SVM response. This, of course, results in additional computational complexity, which could be mitigated in several ways. The simplest would be to estimate the orientation at each pixel using a standard method and then to evaluate SVM response using that orientation only. A more sophisticated way would be to use the training data not only to learn the SVM parameters but also a specialized orientation estimator. A completely different, but compatible, approach would be to use a cascade of classifiers of increasing complexity to perform the full computation only at locations where it makes sense. These are issues we intend to pursue in future work.

Our method, as all those that involve steearable filters, has tight links with those that rely on the eigendecomposition of the Hessian. The eigenvalues of the Hessian can be interpreted as the response to a second order steerable filter in the direction of the underlying ridge [11]. The final result of this methods is a non-linear function applied on the eigenvalues of the Hessian [7, 14]. This function can actually be learned, as in [13], but because the Hessian is very specific, it does not have the kind of flexibility to learn the shape of very different kinds of filament-like structures, such as junctions and intersections, that our approach offers.

4. Results

We compared the detection performance of our algorithm against that of [9] and [7] on the three very different kinds of images depicted by Fig. 1. We have chosen these two algorithms as our baseline because they are state-of-the-art representatives of the two main classes of existing detection methods, as discussed in Section 2.

Images containing representative results of each method are shown in Figures 6, 8 and 10. Each figure contains the original image, the ground truth annotations, and the results of our method and the methods of [7] and [9]. The bottom row shows a detailed area of interest. To help visualizing these results, we threshold the images for a true positive rate and draw true positives in red, false positives in green, and false negatives in blue. The resulting figures are best seen in color.

The ROC curves in Figures 3,9 and 7 summarize the performance of each method. We will use the results to argue the following points:

1. On relatively clean images in which the linear structures truly conform closely the ideal model we obtain results very similar to those of [9] and [7] for very low false positive rates. When higher false positive rates are allowed we outperform the other methods.

2. On more difficult images, the performance of all methods degrade, but our methods degrades to a lesser extent.

We first discuss our experimental methodology and then the specific results obtained for blood vessels in retinal scans, dendrites in brightfield microscopy imagery, and streets in satellite images.

4.1. Experimental Methodology

Ground truth data sets were collected for each type of images, either by experts (blood vessels, neurons) or manual annotation (roads). The annotations denote linear structures and corresponding orientations. Negative samples were chosen randomly from non-filament structures. We compute a feature vector v_n for each sample ground-truth location z_n and its corresponding orientation θ_n. For each image type we collected a minimum of two fully annotated images. The annotated data was divided into disjoint training, validation and test sets, leaving at least one whole image for testing.

The training and validation sets for each image type contains 2500 positive and 2500 negative samples for each of the three kinds of linear structures we are interested in. The training set is used to train the SVM and the validation set is used to optimize its meta-parameters, namely the kernel variance ν and the regularization parameter C, as discussed

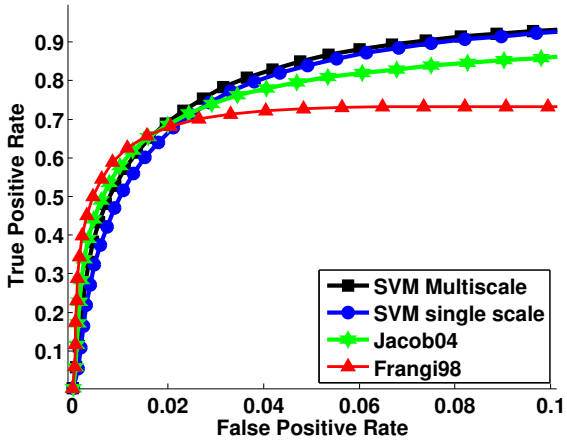


Figure 3. ROC for the blood vessel images. For high true positive rates, both our single-scale and mult-scale methods outperform those of [7, 9].

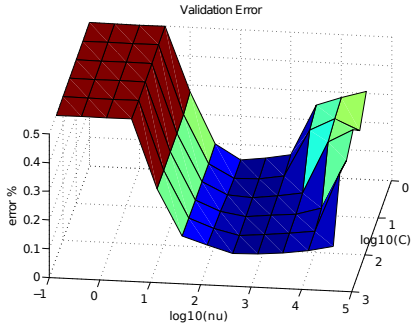


Figure 4. Landscape of the validation error with respect to the ν parameter of the kernel and the C parameter of the SVM for the blood vessels. The function is smooth and a simple optimization method yields good results.

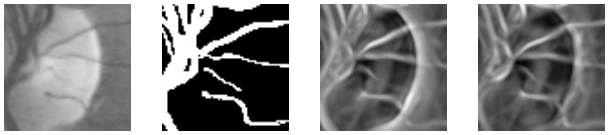
in Section 3.2. In Fig. 4 we plot the typical landscape of the validation error with respect to ν and C.

For each image type, we trained a single-scale and a multi-scale classifier. The single scale classifier is trained to allow for a fair comparison against the filter of [9]. The multi-scale one is trained using a range of scales that is adapted to the width the linear structures we are looking for.

The order of the derivatives in this paper is fixed to $M = 4$ to allow a fair comparison against the filters described in [9].

4.2. Blood Vessels

Blood vessel images obtained from the Drive data set [17] are the cleanest of the three image types. The background is mostly uniform and the vessels appear as dark structures. However, a circular clear structure, close to the root of the tree is a localized source of noise, as it can be seen in Fig. 6(a). The edges of this structure causes problems to both our classifier and to the algorithms of [9, 7].



(a) (b) (c) (d)

Figure 5. Re-training. (a) Original image. (b) Ground truth. (c) Detection obtained with our original detector. Note the high response to the bright circular structure. (d) Detection after retraining using negative examples obtained from similar circular structures. While not eliminated, the response to the distracting circular structure has been reduced.

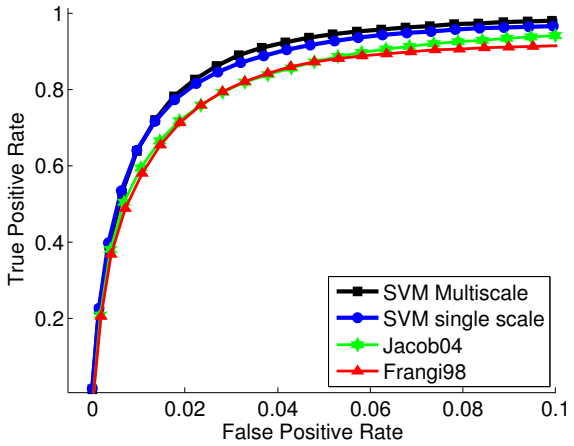


Figure 7. ROC for the neuron images. Our multi-scale and single scale methods outperform the methods of [9, 7] over the entire ROC curve.

The scales used for the training and detection are $\sigma = \{2, 5, 8\}$ in the multi-scale case and $\sigma = 2$ for the single scale . This single-scale σ was selected to give the best results in the method of [9].

The corresponding ROC curve and images can be seen in 3. At high true positive rates we clearly outperform [9, 7]. When the true positive rates falls below 60%, the methods of [9, 7] outperform ours. This is because our algorithm is more sensitive to edges of the clear structure than [9, 7]. During the training phase, very little points fell in this area, and thus we did not learn to reject it. Modifying the training and validation sets to include those points as negative samples makes our algorithm much less sensitive to this effect and improve our detection results. The response of this new classifier, together with the response of the original SVM classifier, is shown in Fig. 5.

This illustrates one of the strengths of our approach. We are not stuck with a rigid model, but can apply a bootstrapping approach that allows us to retrain the system until the desired results are obtained.

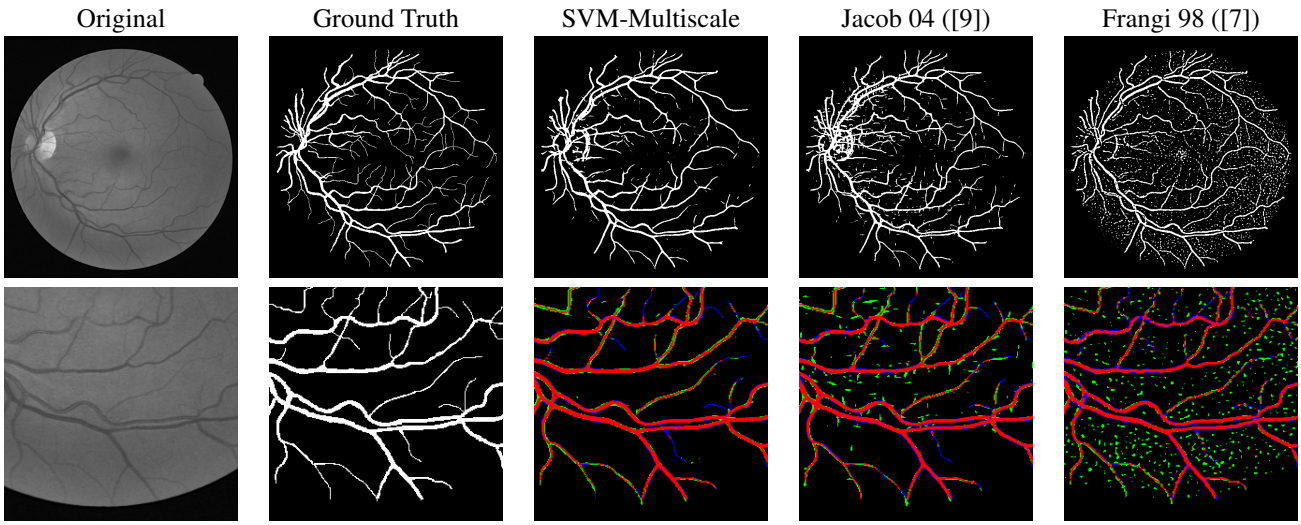


Figure 6. Images of blood vessels in retinal images. For each method, a threshold has been applied to achieve a true positive rate of 80%. Top row: full image. Bottom row: detail. In each result, red pixels indicate true positives, blue indicate false negatives and green indicate false positives. While all three methods are able to correctly recover the main structures, our approach is less sensitive to noise. Note the increase in density of green pixels in the two right-most columns.

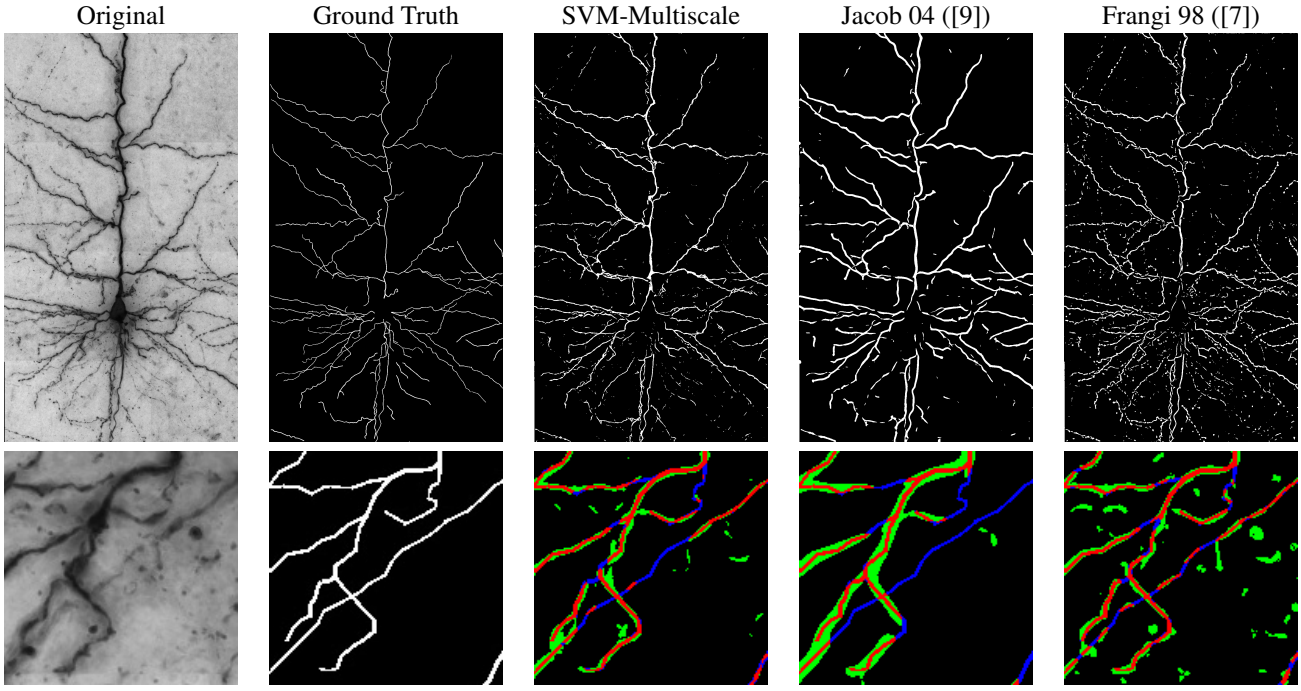


Figure 8. Images of neurons thresholded for a true positive rate of 80% for each method. Top row: full image. Bottom row: detail. The detections of our algorithm and [7] have a tighter fit around the filament than [9]. This can be explained by the fact that our method and the method of [7] employ non-linear techniques, whereas [9] uses a linear filter with a smoother profile. Note that our method is less sensitive to noise from blob structures than [7] because we have learned to reject such structures.

4.3. Neurons in Brightfield Microscopy

Images of neuron dendritic trees, shown in Fig. 8, are obtained from brain sections of rats. The neuron is dyed and then imaged with a brightfield microscope at different tissue depths. The images we analyze are the minimum intensity projection of the 3D stacks.

Several artifacts appear in these images due to irregularities of the staining process, non-gaussian blur attributed to the image acquisition technique and the 3D to 2D projection. As a result, many interesting filaments appear as

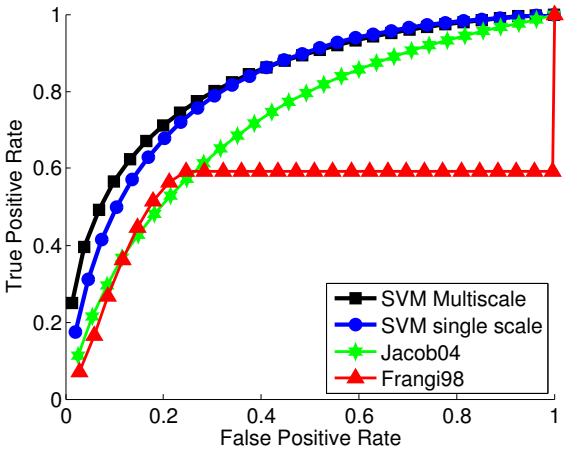


Figure 9. ROC for the street images. All three methods performed noticeably worse on street images, however, our algorithm is the most robust to structured noise and outperforms [7, 9] by a greater margin.

faint structures, and filaments with abrupt changes in width severely blurred. The image in Figure 1(b) contains some of these challenging artifacts.

We compare the performance of our single and multi-scale detectors and those of [9, 7] in Fig. 7. We clearly outperform [9, 7] over the entire ROC curve.

4.4. Streets in Satellite Images

Detecting streets in satellite images was the most challenging task for the filament detectors. As seen in Fig. 10, the streets are often occluded by trees. The intensities of the streets vary according to the quality of the concrete. There are many other structures that can be mistaken for roads, such as houses, swimming pools and parking lots. Furthermore, the areas surrounding some streets have the same intensity as the street. In this case the street no longer resembles a filament.

We detected filaments in the street images in the same manner as the vessel or and neuron images. As one might expect, the results are worse in this dataset than in the previous ones. Nevertheless, our algorithm outperforms [9] and [7], and even more convincingly in this case. This is because our algorithm is much less sensitive to the structured noise present in the images (e.g. from houses, parking lots), as we have learned to reject it in the training phase. This robustness to structured noise is demonstrated by the white houses that appear in the detail of Fig. 10. While [9, 7] react strongly to them, our algorithm rejects them almost completely.

5. Conclusion

We have presented an algorithm to detect filament-like structures in complicated imagery. Instead of explicitly modeling the filaments and the noise present in the image, we learn a model of the filaments from the data itself. By doing this we are able to not only detect filaments, but reject structured noise. Our approach is also able to detect non-ideal filament structures, which cannot be easily explicitly modelled, such as junctions or filaments of non-uniform width.

Our main contribution is to combine a machine learning approach with the decomposition of the image into a multi-scale rotational basis to achieve rotation and scale invariance. This allows us to train a single classifier that learns on data in a canonical orientation. Our classifier is able to detect filaments at any orientation by applying a simple rotation transformation to the extracted feature vector of the pixel under analysis.

We have shown the generality of our approach by evaluating it on three different types of images: blood vessels in retinal images, neurons in brightfield microscopy and streets in satellite imagery. Our results demonstrate that our approach outperforms state-of-the-art methods in all three scenarios and, that the margin of improvement increase with the difficulty of the image.

6. Future Work

Thus far, we have not discussed optimal selection of the set of scales in which the feature vector is computed, or the order of the filters. For some type of images, only Gaussian derivatives of even order might be relevant. The automatic selection of the scales and orders could be done through the inclusion of an L^1 regularization term in the computation of the SVM.

Finally, a future line of research might be to extend our algorithm to 3D image stacks using the rotation properties of steerable filters in 3D [8, 2]. The computational cost of adding a new dimension would have to be addressed, as the number of features needed to achieve rotation increases drastically.

References

[1] G. Agam and C. Wu. Probabilistic modeling-based vessel enhancement in thoracic ct scans. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 684–689, Washington, DC, USA, 2005. IEEE Computer Society.

[2] F. Aguet, M. Jacob, and M. Unser. Three-dimensional feature detection using optimal steerable filters. In *Proceedings of the 2005 IEEE International Conference on Image Processing (ICIP'05)*, volume II, pages 1158–1161, Genova, Italy, September 11-14, 2005.

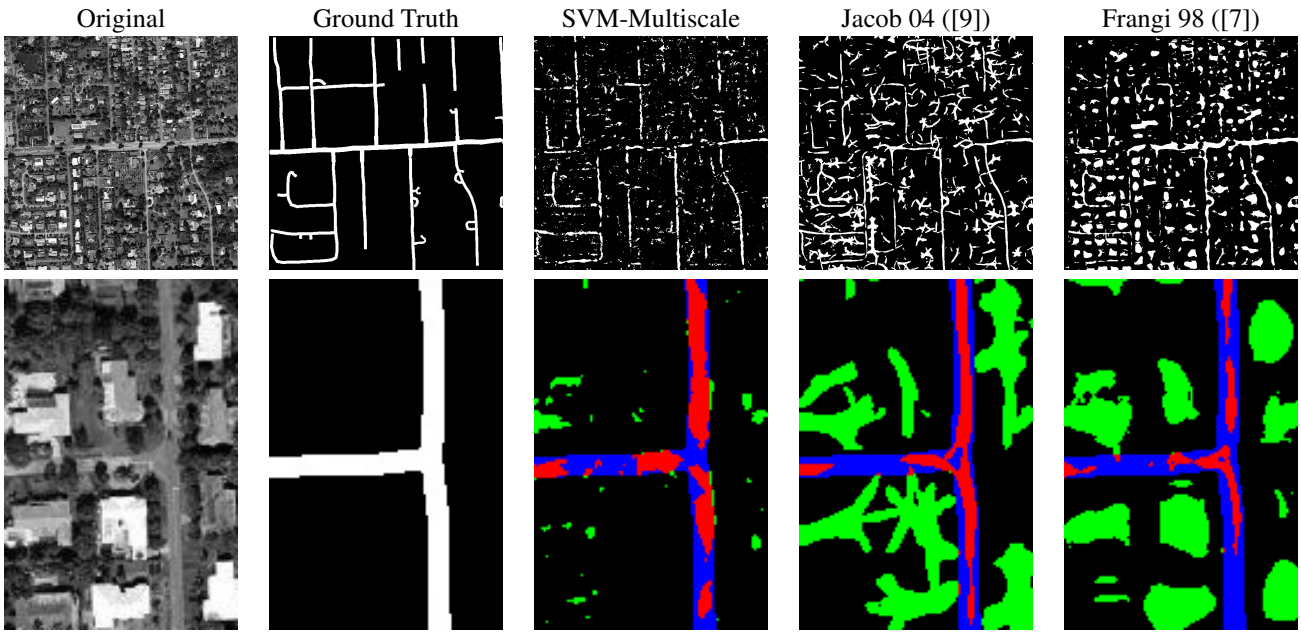


Figure 10. Images of roads thresholded for a true positive rate of 40% for each method. Top row: full image. Bottom row: detail. The performance of every method is noticeably worse than on other image modalities. Nevertheless, our approach is clearly better at reducing noise from structures such as houses. This is because our algorithm is not only trained to detect the structures of interest, but also to reject irrelevant ones.

[3] K. Al-Kofahi, S. Lasek, D. Szarowski, C. Pace, G. Nagy, J. Turner, and B. Roysam. Rapid automated three-dimensional tracing of neurons from confocal image stacks. *IEEE Transactions on Information Technology in Biomedicine*, 2002.

[4] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986.

[5] A. Dima, M. Scholz, and K. Obermayer. 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.

[6] F. Fleuret and D. Geman. Stationary features and cat detection. *Journal of Machine Learning Research (JMLR)*, 2008.

[7] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Lecture Notes in Computer Science*, 1496:130–137, 1998.

[8] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:891–906, 1991.

[9] M. Jacob and M. Unser. Design of steerable filters for feature detection using canny-like criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1007–1019, August 2004.

[10] C. Kirbas and F. Quek. Vessel extraction techniques and algorithms: A survey. In *Procc. of the Third IEEE Symposium on BioInformatics and BioEngineering*, page 238, 2003.

[11] E. Meijering, M. Jacob, J.-C. F. Sarria, P. Steiner, H. Hirling, and M. Unser. Design and validation of a tool for neurite tracing and analysis in fluorescence microscopy images. *Cytometry Part A*, 58A(2):167–176, April 2004.

[12] M. Petrou. Optimal convolution filters and an algorithm for the detection of wide linear features. *Communications, Speech and Vision, IEE Proc. I*, 140(5):331–339, Oct 1993.

[13] A. Santamaría-Pang, C. M. Colbert, P. Saggau, and I. A. Kakadiaris. Automatic centerline extraction of irregular tubular structures using probability volumes from multiphoton imaging. In *MICCAI (2)*, pages 486–494, 2007.

[14] Y. Sato, S. Nakajima, H. Atsumi, T. Koller, G. Gerig, S. Yoshida, and R. Kikinis. 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Medical Image Analysis*, 2:143–168, June 1998.

[15] S. Schmitt, J.-F. Evers, C. Duch, M. Scholz, and K. Obermayer. New methods for the computer-assisted 3d reconstruction of neurons from confocal image stacks. *NeuroImage*, 23:1283–1298, 2004.

[16] R. Socher, A. Barbu, and D. Comaniciu. A learning based hierarchical model for vessel segmentation. *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 1055–1058, May 2008.

[17] J. Staal, M. Abramoff, M. Niemeijer, M. Viergever, and B. van Ginneken. Ridge based vessel segmentation in color images of the retina. *IEEE Trans. on Medical Imaging*, 23:501–509, 2004.

[18] G. Streekstra and J. van Pelt. Analysis of tubular structures in three-dimensional confocal images. *Network: Computation in Neural Systems*, 13(3):381–395, August 2002.

[19] J. Tyrrell, E. di Tomaso, D. Fuja, R. Tong, K. Kozak, R. Jain, and B. Roysam. Robust 3-d modeling of vasculature imagery using superellipsoids. *Medical Imaging*, 26(2):223–237, February 2007.