Multiscale Centerline Detection by Learning a Scale-Space Distance Transform

Amos Sironi*[1], Vincent Lepetit[2], and Pascal Fua[1]

[1]Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne (EPFL)
[2]Institute for Computer Graphics and Vision, Graz University of Technology
{firstname.lastname}@epfl.ch, lepetit@icg.tugraz.at

Abstract

We propose a robust and accurate method to extract the centerlines and scale of tubular structures in 2D images and 3D volumes. Existing techniques rely either on filters designed to respond to ideal cylindrical structures, which lose accuracy when the linear structures become very irregular, or on classification, which is inaccurate because locations on centerlines and locations immediately next to them are extremely difficult to distinguish.

We solve this problem by reformulating centerline detection in terms of a regression *problem. We first train regressors to return the distances to the closest centerline in scale-space, and we apply them to the input images or volumes. The centerlines and the corresponding scale then correspond to the regressors local maxima, which can be easily identified. We show that our method outperforms state-of-the-art techniques for various 2D and 3D datasets.*

1. Introduction

Finding the centerline and estimating the width of linear structures is a critical first step in many applications, ranging from road delineation in 2D aerial images to modeling blood vessels, lung bronchi, and dendritic arbors in 3D biomedical image stacks. Most existing techniques rely on filters designed to respond to locally cylindrical structures [7, 23, 12, 17, 14, 28], optimized for specific profiles [11], or learnt [22, 8, 4]. They compute a scale-dependent measure that, ideally, should be maximal at the centerline of linear structures when computed for the correct scale.

Among these approaches, the learning-based ones tend to outperform the hand-designed ones when the linear structures become very irregular and deviate from the idealized models on which their design is based. Some works only aim at segmenting the linear structures from the background [4], and it is not clear how to reliably extract the centerlines from the segmentation. Others focus on the centerlines, but they typically rely on classification and this re-

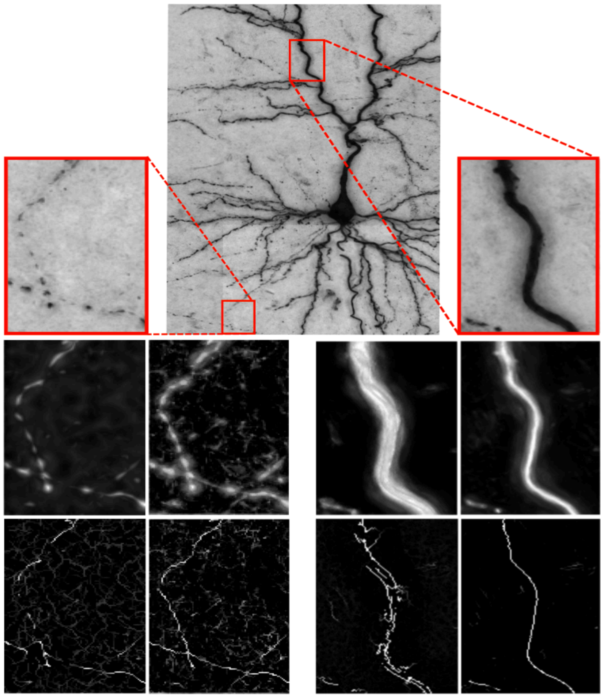


Figure 1. Detecting dendrites in a 3D brightfield image stack. **Top row:** Minimal intensity projection with two enlarged details. **Middle row:** Comparison of the responses of our method against a recent model based approach [25] and a classification based one [4]. **Bottom row:** Centerlines detected after performing Non-Maximum Suppression on the response images. Model based methods have trouble modeling highly irregular structures. Classification based approaches respond on the whole body of the tubular structure and do not guarantee maximal response at the centerline. Our method combines robustness against image artifacts and accurate centerline localization.

[*] This work was supported in part by the EU ERC project MicroNano.

sults in poor localization accuracy. This is because it is hard for the classifier to distinguish points on the centerline itself from those immediately next to it.

In this paper, we show that this problem can be solved by reformulating centerline detection in terms of a regression problem. More precisely, we train scale regressors to return distances to the closest centerline in scale-space. In this way, performing non-maximum suppression on their output yields both centerline locations and corresponding scales. We will show that, on very irregular structures, it outperforms the powerful OOF approach with and without antisymmetry term [13, 14] that is widely acknowledged as one of the best among those relying on hand-designed filters, a very recent extension of it [25] designed to improve its performance on irregular structures, and a similarly recent classification-based method [4].

In the remainder of the paper, we first review related work in section 2. Then, in section 3 we describe our method. Finally, in section 4 we present the results obtained on three challenging datasets and prove the superiority of our approach over the state-of-the-art.

2. Related Work

Centerline detection methods can be classified into two main categories, those that use hand-designed filters and those that learn them from training data. We briefly review both kinds below.

Hand-Designed Filters Such filters also fall into two main categories. The first is made of Hessian-based approaches [7, 23, 22, 17, 6] that combine the eigenvalues of the Hessian to estimate the probability that a pixel or voxel lies on a centerline. The main drawback of these approaches is that the required amount of Gaussian blur to compute the Hessian may result in confusion between adjacent structures, especially when they are thick.

This has led to the development of a second class of methods based on Optimally Oriented Flux (OOF) [13]. They rely on the second order derivatives of an N-dimensional ball and are less sensitive to the presence of adjacent structures. Moreover, the radius of the ball provides a reliable estimate of the tubular structure scale. Remaining difficulties, however, are that OOF can also respond strongly to edges as opposed to centerlines and that its performance degrades when the structures become very irregular. A number of schemes have been proposed to solve the first problem [1, 24, 19, 14, 28]. For example, in [14], an Oriented Flux Antisymmetric (OFA) term was added and has proved effective. There has been less work on improving OOF's performance on truly irregular structures, except for the very recent approach of [25] that attempts to maximize the image gradient flux along multiple radii in different directions instead of only one as in [13].

The method proposed in [31] can be seen as a mixture of these two classes. Hessian computation implicitly assumes an ellipsoidal model whereas in [31] the ellipsoid is explicitly fitted to the data. Because this is harder to do than fitting OOF balls, it is achieved by a learning a regression model from image data to ellipse parameters. However, this has only been demonstrated in a tracking context.

Learned Filters Even if care is taken to add computational machinery to handle irregular structures [22, 25], the performance of hand-designed filters tends to suffer in severe cases such as the one depicted by Fig. 1. This is mostly because it is very difficult to explicitly model the great diversity of artifacts that may be present.

Some works therefore aim at segmenting linear structures in biomedical images [8, 29, 20, 4] or aerial ones [18, 27] by applying classification to label the pixels or voxels as belonging to the structure of interest or to the background. However, this is a problem simpler than the one we consider. It is not accurate to find the actual centerlines from the segmentation even with post-processing operations. In particular, there is no guarantee that the classifier responses will be maximal at the centers of the structures. By contrast, we recover the centerlines and the corresponding thickness of the linear structures to which they belong, and it is straightforward to generate a segmentation from this data.

Other techniques, [10, 26, 30, 5] aim at extracting the centerlines as we do, but still rely on binary classification to distinguish the image locations on centerlines from the rest. [30, 5] use Haar wavelets in conjunction with boosted trees to detect the centerlines of tubular structures at different scales. [10] uses spectral-structural features instead and SVMs to find road centerlines. In [26] co-occurrence features and the AdaBoost algorithm are used to detect the spinal column centerline.

These methods exhibit limited localization accuracy because points near the centerlines can easily be also classified as centerline points due to their similar appearance. As our experiments show, our approach based on regression rather than classification is more adapted to the problem at hand.

3. Method

Let $I(\mathbf{x})$, with $\mathbf{x} \in \mathbb{R}^N$, be an N-dimensional image containing curvilinear structures of various radii. A classification-based approach to finding their centerlines involves learning a function $y(\cdot)$, such that

$$y(f(\mathbf{x}, I)) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ is on a centerline,} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $f(\mathbf{x}, I)$ is a feature vector computed from a neighborhood surrounding $\mathbf{x}$ in image I. As discussed above,

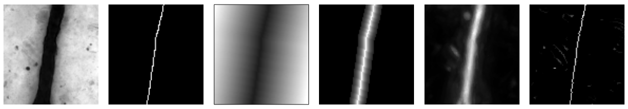

(a) Image I (b) Centerline C (c) Distance $\mathcal{D}_C$ (d) Function d (e) Function φ (f) NMS Image

Figure 2. Learning a regressor for centerline detection. (a) Raw image; (b) Ground truth centerline; (c) The distance transform to the centerline is used to discriminate points close to it; (d) The function we want to learn is maximal at the centerlines and it is thresholded to a constant value when the local window used to compute features does not contain any centerline points; (e) The function learned with our method; (f) Centerline detected after Non-Maxima Suppression (NMS) on function φ.

this is hard to do because points on the centerline itself for which $y(\cdot)$ should return 1 and their immediate neighbors for which it should return 0 look very similar. One way to solve this is to train $y(\cdot)$ to return 1 for all points within a given distance from the centerline. However, in practice, even if $y(\cdot)$ is allowed to return floating point values between zero and one, using for instance an SVM-style classifier, there is no guarantee that its value will be maximal at the centerline itself. This makes finding its accurate location, for example by using non-maximum suppression, problematic.

Our solution is to learn instead $y(\cdot)$ as a regressor whose values decrease monotonically as the distance of point $\mathbf{x}$ to the centerline increases. Then, as shown in Fig. 2, we can rely on simple non-maximum suppression to localize the centerlines. We will show in the next section that this solution is significantly more robust than both classification-based and filter-based methods.

In the remainder of this section, we first describe this process for structures whose scale is assumed to be known *a priori*. We then relax this constraint to handle structures of arbitrary scale and discuss the feature vectors we use as input to our regressors. Moreover, we will use the terms radius and scale interchangeably.

3.1. Learning a Regressor for Fixed Radius Structures

Let us momentarily assume that the linear structures have a known radius r. Let C be the set of centerline points and $\mathcal{D}_C$ the corresponding Euclidean distance transform, that is, $\mathcal{D}_C(\mathbf{x})$ is the metric distance from location $\mathbf{x}$ to the closest location in C.

Our goal is to learn a function $y(\cdot)$ such that $y(f(\mathbf{x}, I))$ is maximal for $\mathbf{x}$ on the centerline and whose value decreases monotonically as $\mathbf{x}$ moves away of it. The function $d(\mathbf{x}) = -\mathcal{D}_C(\mathbf{x})$ has this property, see Fig. 3. In theory, given training data, we could learn $y(\cdot)$ as a regressor that takes $f(\mathbf{x}, I)$ as input and returns $-\mathcal{D}_C(\mathbf{x})$ as output. However, in practice, we learn a different function for the two following reasons.

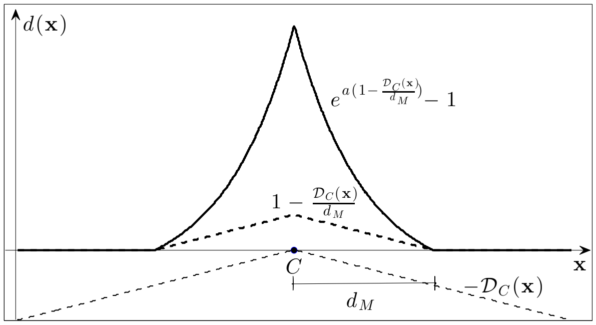


Figure 3. The function d in the case of $\mathbf{x} \in \mathbb{R}$. If a centerline point is located in C, the function we want to learn is obtained from the distance transform $\mathcal{D}_C$, after thresholding and scaling. The vertical axis has been scaled for visualization purposes.

First, because our feature vectors $f(\mathbf{x}, I)$ are computed using local neighborhoods of size s, a regressor could only learn it for points that are close enough to the centerlines for their neighborhood to be affected by it. For this reason, it makes sense to threshold d when $\mathcal{D}_C$ is greater than a given value d_M, which is a function of the neighborhood size s. This yields the modified function

$$d(\mathbf{x}) = \begin{cases} 1 - \frac{\mathcal{D}_C(\mathbf{x})}{d_M} & \text{if } \mathcal{D}_C(\mathbf{x}) < d_M \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

which takes values between 0 and 1, see Fig. 3. In our implementation, we set $d_M = s/2$, which means that d is uniformly 0 for points whose corresponding neighborhood does not overlap the centerline.

Second, a regressor trained to associate to a feature vector $f(\mathbf{x}, I)$ the value of $d(\mathbf{x})$ can only do so approximately. As a result, there is therefore no guarantee that its maximum is exactly on the centerline. To increase robustness to noise, we have therefore found it effective to train our regressor to reproduce a distance function whose extremum is better defined. In our actual implementation, we take it to be

$$d(\mathbf{x}) = \begin{cases} e^{a(1-\frac{\mathcal{D}_C(\mathbf{x})}{d_M})} - 1 & \text{if } \mathcal{D}_C(\mathbf{x}) < d_M \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $a > 0$ is a constant that control the exponential decrease rate of d close to the centerline, see Fig. 3. In all our experiments, we set $a = 6$.

The regression method we use to learn function d is the GradientBoost algorithm [9]. It can be viewed as a generalization of the AdaBoost algorithm and it can efficiently approximate very complex functions.

Given training samples $\{(f_i, y_i)\}_i$, where $f_i = f(\mathbf{x}_i, I_i) \in \mathbb{R}^M$ is the feature vector corresponding to a point $\mathbf{x}_i$ in image I_i and $y_i = d(\mathbf{x}_i)$, GradientBoost ap-

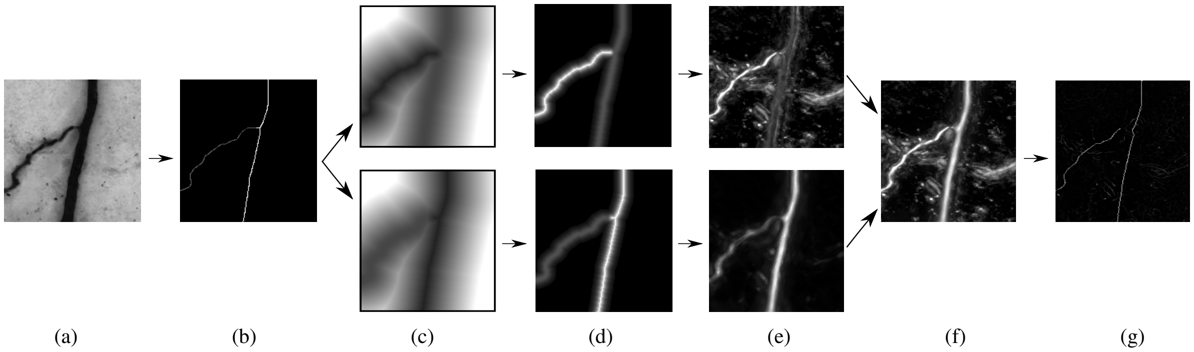


Figure 4. Training regressors for multiscale centerline detection. (a) Raw image; (b) Ground truth centerline; (c) Scale-space distance transform for two scale values; (d) Distance transform thresholded and rescaled. Local maxima of this function provide the centerline locations and the structure radii; (e) We approximate the function in (c) by learning different regressors for different radii; (f) The maximal intensity projection of functions (e) is used to extract the centerline by applying non-maximum suppression; (g) Detected centerline. For each centerline point we can recover the corresponding radius by considering the regressor that returned the maximal value for that point.

proximates $y(\cdot)$ by a function of the form

$$\varphi(f(\mathbf{x}, I)) = \sum_{k=1}^{K} \alpha_k h_k(f(\mathbf{x}, I)) , \quad (4)$$

where $h_k : \mathbb{R}^M \to \mathbb{R}$ are weak learners and $\alpha_k \in \mathbb{R}$ are weights. Function φ is built iteratively, selecting one weak learner and its weight at each iteration, to minimize a loss function $\mathcal{L}$ of the form $\mathcal{L} = \sum_i L(d_i, \varphi(f_i))$. We use the squared loss $L(d, \varphi(f(\mathbf{x}, I))) = \|d - \varphi(f(\mathbf{x}, I))\|_2^2$. As it is usually done with GradientBoost, we use regression trees as weak learners since they achieve state-of-the-art performance in many applications [9]. In all our experiments we used $K = 350$ trees of depth 2. Fig. 2 shows the output of the learned function for a sample image.

3.2. Handling Structures of Arbitrary Radius

In the previous section, we focused on structures of known radius. In general, however, structures of many different radii are present. To generalize our approach to this multi-scale situation, we redefine the function d of Eq. (3) once again as

$$d(\mathbf{x}; r) = \begin{cases} e^{a \cdot (1 - \frac{\mathcal{D}_C(\mathbf{x}; r)}{d_M})} - 1 & \text{if } \mathcal{D}_C(\mathbf{x}; r) < d_M, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

where C is now the set of centerline points and the corresponding radii. In other words, C now is a set of $(\mathbf{x}; r)$ $(N + 1)$-dimensional vectors and $\mathcal{D}_C(\mathbf{x}; r)$ is the *scale-space* distance transform of C

$$\mathcal{D}_C^2(\mathbf{x}; r) = \min_{(\mathbf{x}', r') \in C} \|\mathbf{x} - \mathbf{x}'\|_2^2 + k(r - r')^2 \ , \quad (6)$$

where k is used to weight the scale component differently from the space component. In practice k depends on the image resolution and the range of scales. In Section 4 we discuss the choice of k.

If we consider the maximum projection of $d(\mathbf{x}; r)$ along the radial component, we obtain a function of $\mathbf{x}$, whose local maxima are the centerline points for all the values of r. Therefore, if we train a regressor to output the values of d, the problem of multiscale centerline detection is reduced to the problem of finding local maxima in the projected image, see Fig. 4(f).

Moreover, function $d(\mathbf{x}; r)$ is defined so that points in C are local maxima of d not only along the spatial dimensions, but also along the radial component, as shown in Fig. 4(e). Therefore, we can easily find the scale corresponding to a centerline point as the one that gives the maximal value for that point.

We now want to learn a regressor $y(\cdot; r)$ that returns the values of this new d function. The simplest way would be to discretize the range of possible scales r into a finite set of scales and to use the fixed-radius method of Section 3.1 to learn one regressor φ_r for each scale in this set. This approach, however, decreases the number of training samples available to train each regressor, which in our experience severely impairs performance.

An alternative approach is to rely on scale-space theory [15] to train a single regressor φ_{r^0} for radius r^0. By properly scaling and normalizing the convolutional filters used to compute the feature vectors $f(\mathbf{x}, I)$, we can use φ_{r^0} to find the centerlines for all the other radii. The advantage of this approach is that we can exploit all training samples to train φ_{r^0} by rescaling them to have a radius equal to r^0. However, this assumes that the aspect of tubular structures is scale invariant. When this is not the case, the results are less accurate, especially for large differences between the actual radius of the structure and r^0.

We therefore adopt a hybrid approach. We learn a set of regressors $\{\varphi_{r_i}\}_{i \in S}$ for a small set of regularly sampled radii. We then apply the scale-space approach for intermediate radii and use the closest r_i to the scale we want to predict. In section 4 we discuss how these radii are selected.

3.3. Computing the Feature Vectors

Many options are available to compute the feature vectors $f(\mathbf{x}, I)$, such as Haar wavelets, steerable filters, or Gabor filters. Recent work [20] has shown that learning a set of convolutional filters via sparse coding techniques can produce expressive features that perform well on linear structures. We therefore take $f(\mathbf{x}, I)$ to be

$$f(\mathbf{x}, I) = [(\mathbf{f}_1 * I)(\mathbf{x}), \ldots, (\mathbf{f}_M * I)(\mathbf{x})]^\top , \qquad (7)$$

where the $\mathbf{f}_i$'s are convolutional filters learned in a unsupervised way as in [20] from a set of training images, and applied to image I.

In the case of 2D images, we used $M = 121$ filters. In the case of 3D volumes, the number of possible orientations of the tubular structures is significantly larger and therefore more filters should be used. We found it most effective to learn first a filter bank of $M = 121$ filters and then extend it by rotating the learned filters at different orientations, 14 in practice. To speed up the convolutions required to compute the descriptor in Eq. (7), we rely on the technique introduced in [21], which approximates the filters $\{\mathbf{f}_i\}$ with a set of separable ones.

The convolutional filters we used in our experiments and the separable ones used to approximate them are shown in the supplementary material. Moreover, we demonstrate their power by comparing the performances of our method using these learned filters against using it with other filters.

3.4. Non-Maximum Suppression

Applying our method to an N-dimensional image, yields an $(N + 1)$-dimensional one, with N spatial dimensions and one scale dimension. Our method is designed to respond maximally at the centerlines in scale-space. To find these local maxima, we first compute a N-dimensional image by keeping for each location the maximum along the radii, and saving the radius corresponding to the maximum. We then perform a Canny-like non-maximum suppression by keeping only the locations that correspond to a local maximum along a line perpendicular to the local orientation, and within a neighborhood of width defined by the radius. We estimate the orientation using the eigenvectors of the oriented flux matrix [13], which we found to be more robust than using the Hessian matrix. Results from this non-maximum suppression step are shown in Fig. 5(e).

4. Results

In this section, we first introduce the datasets and the parameters used in our experiments. Then, we describe our evaluation methodology. Finally, we discuss our results.

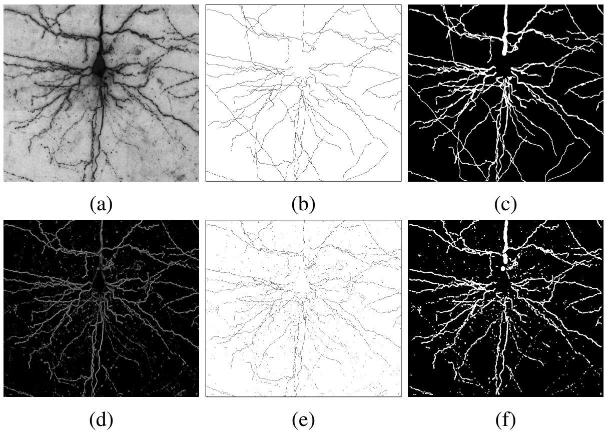


Figure 5. Segmentation of a 3D Brightfield image stack. (a) Minimum projected image; (b) Ground truth centerlines; (c) Ground truth segmentation . (d) Maximum projection of the function returned by our method; (e) Detected centerlines after non-maxima suppression; (f) Segmentation obtained from centerlines and radial estimation as described in Section 4.2.

4.1. Datasets and Parameters

Our method depends on few parameters, namely: the radial weight k in Eq (6); the size s of the filters used to extract the features; the range of sampled scales and the number of trained regressors.

The range of scales sampled for the different datasets is automatically determined from the ground truth data and were always sampled uniformly. We optimized the other parameters by a cross validation procedure on small volumes. We tested our method on the 2D road images and 3D biological image stacks depicted by Fig. 6. More specifically we used the following datasets:

- **Aerial**: Aerial images of road networks. We used a training set composed of 7 images and used 7 others for testing. We sampled 10 scales ranging from 5 to 14. We trained 4 regressors at scales 6, 8, 11 and 13 and learned filters of size $s = 21$. We set k to 1.

- **Brightfield**: A dataset of 3D image stacks acquired by brightfield microscopy from biocityne-dyed rat brains. We used 3 images for training and 2 for testing. We sample 12 scales corresponding to radii from 1 to 12 microns. We trained 2 regressors at scales 2 and 8. We learned filters of size $s = 21$ and used $k = 1$.

- **VC6**: Three dimensional brightfield micrographs of biocytin-labeled cat primary visual cortex layer 6 taken from the DIADEM challenge data [3]. We used 3 images for training and 2 for testing. We sampled 6 scales from 1 to 6, trained 3 regressors at scale 1, 3 and 5. We used $k = 7$ and $s = 11$.

- **Vivo2P**: Three dimensional in vivo two-photon images of a rat brain, capturing the evolution of neurons in the

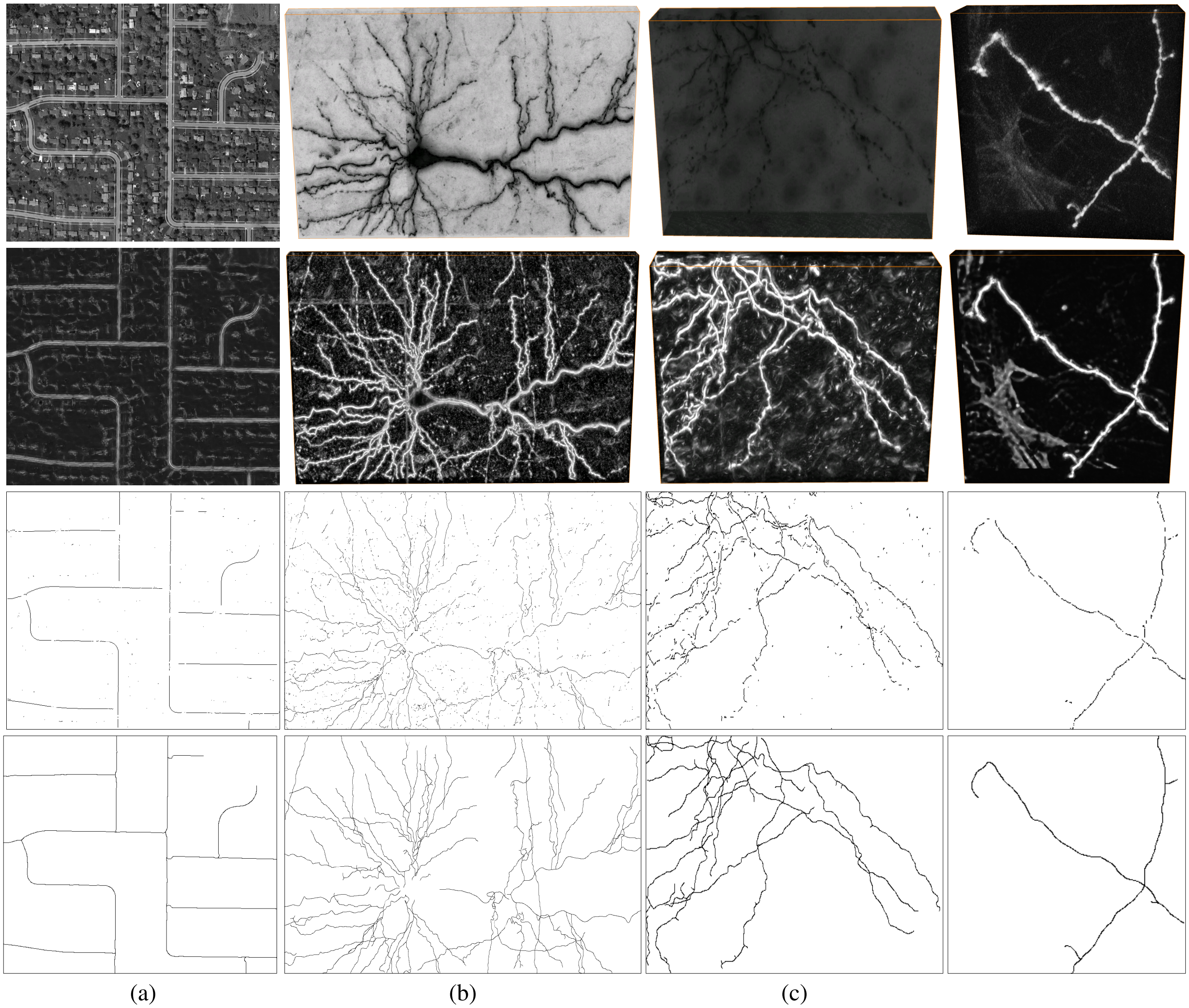


Figure 6. Centerline Detection Results. (a) Aerial image. (b) Brightfield image stack. (c) VC6 image stack. In each case, we show from top to bottom the original image, the maximum projection along the radial component of our regressor's output, centerlines detected by thresholding after non-maximum suppression, and ground truth centerlines.

neocortex. We used 2 images for training and 3 sequences of 3 images for testing. We sampled 3 scales, 0.6, 0.7, and 0.8 microns. We trained one regressor at scale 0.7, using $k = 1$ and $s = 21$.

For training, we randomly sampled 100 000 image locations within the distance d_M to the centerline and other 100 000 from points further than d_M to the centerline.

The size of the images ranges from $\sim 10^5$ pixels for the Aerial dataset to $\sim 10^8$ voxels for Brightfield. The running time in our Matlab implementation is of several hours for training and from few minutes to few hours for testing.

4.2. Evaluation

We compare our approach against three of the most powerful model-based methods for centerline detection. Optimally Oriented Flux (OOF) [13], Oriented Flux with Oriented Flux Antisymmetry [14] (OOF+OFA), and Multidirectional Oriented Flux [25] (MDOF). Moreover, to prove the importance of our regression approach compared to classification, we also train a GradientBoost classifier to segment the centerlines from the rest of the images, thus emulating the approach of [4].

As usually done to evaluate methods extracting one-pixel-wide curves [16, 18, 25], we introduce a tolerance factor ρ to perform plot precision-recall (PR) curves analysis.

A predicted centerline point is considered a true positive if it is at most ρ distant from a ground truth centerline point. We generate PR curves for all the methods for different value of ρ. The results for $\rho = 2$ are shown in Fig. 8(a) and show that our approach clearly outperforms all other datasets. Curves for other values of ρ are shown in the supplementary material and exhibit the same trend.

We also evaluate the accuracy of the radii we estimate. Again we follow the same evaluation methodology of [25]. We start by thresholding the image after non-maximum suppression at different values. Then, for each point in the thresholded image, we construct a sphere using the corresponding estimated radius. In this way we obtain for every threshold value a full segmentation of the tubular structures, which we can compare to the ground truth.

Since the ground truth data itself can be inaccurate, we introduce also in this case a tolerance factor δ, and eliminate from comparison points that are closer than $\delta\, r$ from the surface of a ground truth tube of radius r.

Fig. 8(b) shows the precision-recall results for $\delta = 0.4$ and graphs for other values of δ are given in the supplementary material. In this case also, our method outperforms all the others for all the relevant ranges of precision and recall.

We observe the biggest improvement for the Aerial dataset. There, model-based methods do worst because they respond strongly to bright polygonal objects such as houses, as can be seen in Fig. 7. Learning-based methods can be taught to discount them, and in this case, classification does better than hand-crafted methods, but still not as well as our approach. Classification is also competitive on the Vivo2P dataset. The reason is that there are mainly thin branches in this dataset. However, our method gives the higher accuracy, especially for the radial estimation.

On the Brightfield and VC6 datasets, our approach still does best but classification does worst, especially in Brightfield case, due to the presence of very wide branches. As shown in Fig. 1, in such cases, the maximum response is not necessarily on the centerline and non-maxima suppression behaves badly. Our regression-based approach avoids this problem. As observed in [25], the antisymmetric term introduced by OFA degrades the results with respect to OOF for very irregular structures. However, with and without it, OOF is more sensitive than our algorithm to strong artifacts and image noise, which are hard to ignore for hand-crafted methods.

Only for the segmentation results on the Brightfield dataset and for very high recall values, does the precision of our approach significantly degrade. This is due to the sensitivity of our method to thin and faint structures. It is needed to detect the smallest branches but, inevitably, makes it also respond, albeit weakly, to noise. Moreover, in this range of recall values, the centerline localization accuracy of the other methods becomes very low, making their results essentially meaningless.

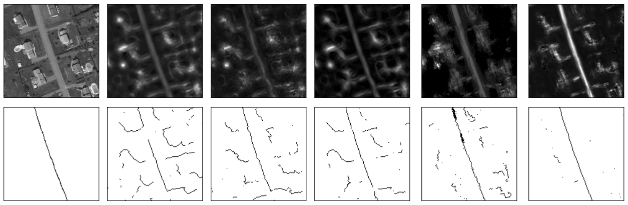

(a) Ground Truth (b) OOF (c) OOF+OFA (d) MDOF (e) Classification (f) Our Approach

Figure 7. Centerline detection on Aerial images. **Top row**: Raw image and responses returned by the different methods. **Bottom row**: Ground truth and extracted centerlines.

5. Conclusion

We have introduced an efficient regression-based approach to centerline detection, which we showed to outperform both methods based on hand-designed filters and classification-based approaches.

We believe our approach to be very general and applicable to other linear structure detection tasks when training data is available. For example, given a training set of natural images and the contours of the objects present in the images, our framework should be able to learn to detect such contours in new images as was done in [2]. This is a direction we will explore in future work.

References

[1] G. Agam and C. Wu. Probabilistic Modeling-Based Vessel Enhancement in Thoracic CT Scans. In *CVPR*, pages 684–689, 2005. 2

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 33(5):898–916, 2011. 7

[3] G. Ascoli, K. Svoboda, and Y. Liu. Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge, 2010. 5

[4] C. Becker, R. Rigamonti, V. Lepetit, and P. Fua. Supervised Feature Learning for Curvilinear Structure Segmentation. In *MICCAI*, September 2013. 1, 2, 6

[5] D. Breitenreicher, M. Sofka, S. Britzen, and S. Zhou. Hierarchical Discriminative Framework for Detecting Tubular Structures in 3D Images. In *International Conference on Information Processing in Medical Imaging*, 2013. 2

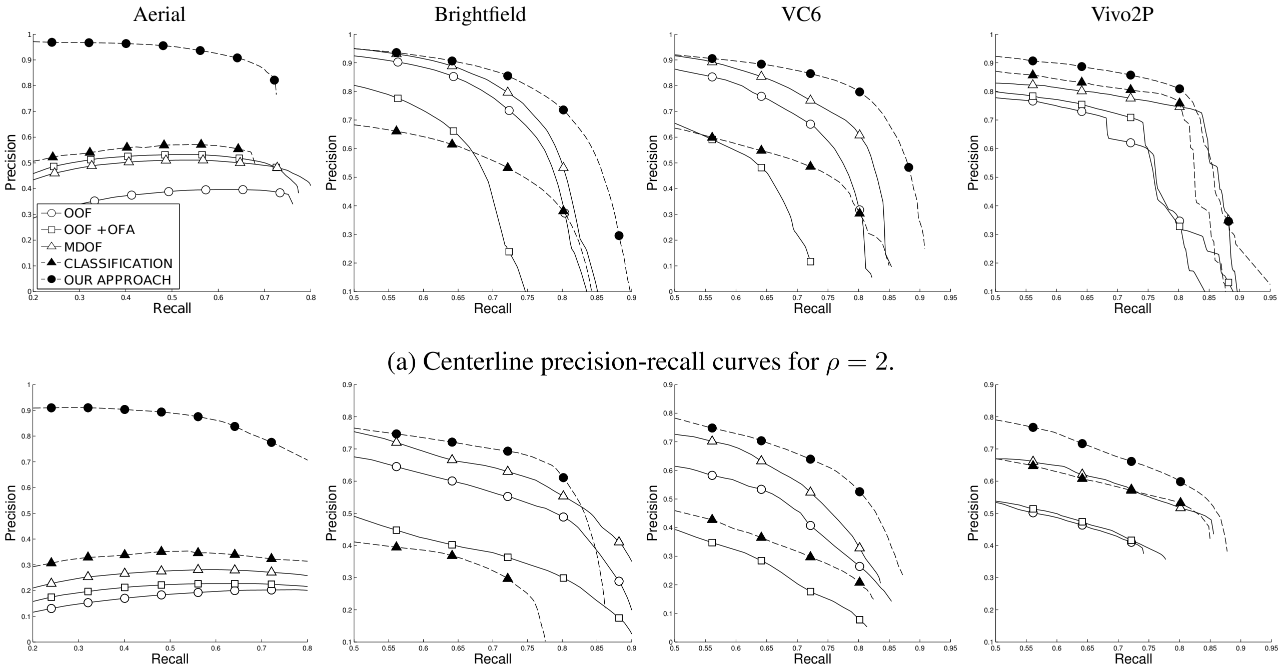
[6] A. H. Foruzan, R. A. Zoroofi, Y. Sato, and M. Hori. A Hessian-Based Filter for Vascular Segmentation of Noisy Hepatic CT Scans. *International Journal of Computer Assisted Radiology and Surgery*, 7(2):199–205, 2012. 2

[7] A. Frangi, W. Niessen, K. Vincken, and M. Viergever. Multiscale Vessel Enhancement Filtering. *Lecture Notes in Computer Science*, 1496:130–137, 1998. 1, 2

[8] G. Gonzalez, F. Aguet, F. Fleuret, M. Unser, and P. Fua. Steerable Features for Statistical 3D Dendrite Detection. In *MICCAI*, pages 625–32, September 2009. 1, 2

[9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001. 3, 4

[10] X. Huang and L. Zhang. Road Centreline Extraction from High-Resolution Imagery Based on Multiscale Structural Features and Support Vector Machines. *International Journal of Remote Sensing*, 30:1977–1987, 2009. 2



(a) Centerline precision-recall curves for $\rho = 2$.

(b) Segmentation precision-recall curves for $\delta = 0.4$.

Figure 8. Precision Recall curves. Our method outperforms the others on all the datasets we considered, both for centerline detection and joint centerline and radius estimation.

[11] M. Jacob and M. Unser. Design of Steerable Filters for Feature Detection Using Canny-Like Criteria. *PAMI*, 26(8):1007–1019, August 2004. 1

[12] K. Krissian, G. Malandain, N. Ayache, R. Vaillant, and Y. Trousset. Model-Based Detection of Tubular Structures in 3D Images. *CVIU*, 80(1):130–171, November 2000. 1

[13] M. Law and A. Chung. Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In *ECCV*, 2008. 2, 5, 6

[14] M. Law and A. Chung. An Oriented Flux Symmetry Based Active Contour Model for Three Dimensional Vessel Segmentation. In *ECCV*, pages 720–734, 2010. 1, 2, 6

[15] T. Lindeberg. Scale-Space for Discrete Signals. *PAMI*, 12(3):234–254, 1990. 4

[16] D. Martin, C. Fowlkes, and J. Malik. Learning to Detect Natural Image Boundaries Using Local Brightness, Color and Texture Cues. *PAMI*, 26(5), 2004. 6

[17] 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. 1, 2

[18] V. Mnih and G. Hinton. Learning to Label Aerial Images from Noisy Data. In *ICML*, 2012. 2, 6

[19] M. Pechaud, G. Peyré, and R. Keriven. Extraction of Tubular Structures over an Orientation Domain. In *CVPR*, pages 336–342, 2009. 2

[20] R. Rigamonti and V. Lepetit. Accurate and Efficient Linear Structure Segmentation by Leveraging Ad Hoc Features with Learned Filters. In *MICCAI*, 2012. 2, 5

[21] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua. Learning Separable Filters. In *CVPR*, 2013. 5

[22] A. Santamaría-Pang, T. Bildea, C. M. Colbert, P. Saggau, and I. Kakadiaris. Towards Segmentation of Irregular Tubular Structures in 3D Confocal Microscope Images. In *MICCAI Workshop in Microscopic Image Analysis and Applications in Biology*, 2006. 1, 2

[23] 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. *MIA*, 2:143–168, June 1998. 1, 2

[24] M. Sofka and C. Stewart. Retinal Vessel Centerline Extraction Using Multiscale Matched Filters, Confidence and Edge Measures. *TMI*, 2006. 2

[25] E. Turetken, C. Becker, P. Glowacki, F. Benmansour, and P. Fua. Detecting Irregular Curvilinear Structures in Gray Scale and Color Imagery Using Multi-Directional Oriented Flux. In *ICCV*, December 2013. 1, 2, 6, 7

[26] C. Wang, Y. Li, W. Ito, K. Shimura, and K. Abe. A Machine Learning Approach to Extract Spinal Column Centerline from Three-Dimensional CT Data. In *SPIE*, 2009. 2

[27] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. A Higher-Order CRF Model for Road Network Extraction. In *CVPR*, 2013. 2

[28] C. Xiao, M. Staring, Y. Wang, D. Shamonin, and B. Stoel. Multiscale Bi-Gaussian Filter for Adjacent Curvilinear Structures Detection with Application to Vasculature Images. *TIP*, 22(1):174–188, 2013. 1, 2

[29] Y. Zheng, M. Loziczonek, B. Georgescu, S. K. Zhou, F. Vega-Higuera, and D. Comaniciu. Machine Learning Based Vesselness Measurement for Coronary Artery Segmentation in Cardiac CT Volumes. *SPIE*, 7962(1):79621–7962112, 2011. 2

[30] S. K. Zhou, C. Tietjen, G. Soza, A. Wimmer, C. Lu, Z. Puskas, D. Liu, and D. Wu. A Learning Based Deformable Template Matching Method for Automatic Rib Centerline Extraction and Labeling in CT Images. In *CVPR*, 2012. 2

[31] Q. Zhu, D. Zheng, and H. Xiong. 3D Tubular Structure Extraction Using Kernel-Based Superellipsoid Model with Gaussian Process Regression. In *VCIP*, pages 1–6, 2012. 2