Accurate and Efficient Linear Structure Segmentation by Leveraging Ad Hoc Features with Learned Filters

Roberto Rigamonti and Vincent Lepetit*

CVLab, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
{roberto.rigamonti,vincent.lepetit}@epfl.ch
http://cvlab.epfl.ch

Abstract. Extracting linear structures, such as blood vessels or dendrites, from images is crucial in many medical imagery applications, and many handcrafted features have been proposed to solve this problem. However, such features rely on assumptions that are never entirely true. Learned features, on the other hand, can capture image characteristics difficult to define analytically, but tend to be much slower to compute than handcrafted features. We propose to complement handcrafted methods with features found using very recent Machine Learning techniques, and we show that even few filters are sufficient to efficiently leverage handcrafted features. We demonstrate our approach on the STARE, DRIVE, and BF2D datasets, and on 2D projections of neural images from the DIADEM challenge. Our proposal outperforms handcrafted methods, and pairs up with learning-only approaches at a fraction of their computational cost.

1 Introduction

The extraction of linear or tubular structures has received a lot of attention from the medical imaging community, as it is the first step to recover the structure of blood vessels and neurons from images. Such extraction can be reliable if a human operator uses a semi-automated system [13]. However, imaging nowadays efficiently generates images with increasingly higher resolution, and the amount of data to analyse is overwhelming. Manually processing images thus becomes infeasible, even with very efficient semi-automated systems, and as such there is a need for automatic, reliable and fast, ways of extracting linear structures.

In order to fully automatize this extraction, many handcrafted approaches have been proposed. A common technique is to rely on the eigenvalues of the image Hessian matrix [11], which can be computed from the responses of a few separable filters [19,21], as in the multiscale vessel enhancement filtering (EF) method [7]. Other approaches rely on differential kernels [1], look for parallel

* This work has been supported in part by the Swiss National Science Foundation.The authors would like to thank C. Becker and F. Benmansour for their help in the experiments.

N. Ayache et al. (Eds.): MICCAI 2012, Part I, LNCS 7510, pp. 189–197, 2012.

edges [6], or fit superellipsoids to the image stack. A recent successful approach is the Optimally Oriented Flux (OOF) [12], computed by convolving the second derivatives of the image with the N-dimensional unit ball.

While these handcrafted methods are typically fast, the quality of their results is limited. This is because actual linear structures do not necessarily conform to the assumptions they make. For example, OOF sometimes provides weak responses, especially at bifurcations and crossovers, and yet these locations are crucial for the automated tracing of the tree structure underlying the input image. Moreover, its effectiveness on noisy data is rather poor.

Several authors used Machine Learning techniques to avoid making strong assumptions. [18,8] apply a Support Vector Machine to the responses of *ad hoc* filters. [18] considers the Hessian's eigenvalues, while the Rotational Features of [8] use steerable filters. More recently, [17] used a dictionary learning method to learn a set of linear filters on images of linear structures, by contrast with hard-coding them. In particular, it shows that convolving images with this filter bank gives responses that, when fed to an SVM, outperform state-of-the-art methods including EF, OOF, and the Rotational Features. Unfortunately, it requires a large number of filters—more than one hundred—which makes it impractical for large images.

In this paper we show that handcrafted methods and learned filters complement each other very well, as depicted in Fig. 1-right. We can therefore take advantage of both types of approaches to extract quickly and reliably linear structures. More precisely, we apply a classifier—we use Random Forests (RFs) [3] and ℓ_1-regularized logistic regression (ℓ_1-reg) [10] for efficiency—to the responses of several filters. For the handcrafted methods, we consider the EF and OOF methods. The other filters are learned with sparsity constraints, and by contrast with [17], we use a very small number of them, typically less than 10. Thanks to this small number, we save a great amount of time not only when extracting the features from the image, but also during training and testing, as the vectors to be classified are much more compact.

In short, our approach is significantly more accurate than handcrafted methods, and much faster than learning-based-only methods, bringing the accuracy advantage of learning to practical applications. In the remainder of the paper, we first describe our method, and then give a summary of our experiments comparing it against state-of-the-art approaches on challenging data.

2 Method

In this section we describe how we compute linear filters from training images, how we apply these filters to extract features from images, and how we use these features to extract linear structures.

2.1 Learning Linear Filters

We learn our linear filters by modeling the distribution of images representative of the problem at hand. This is done by assuming that there exists a sparse

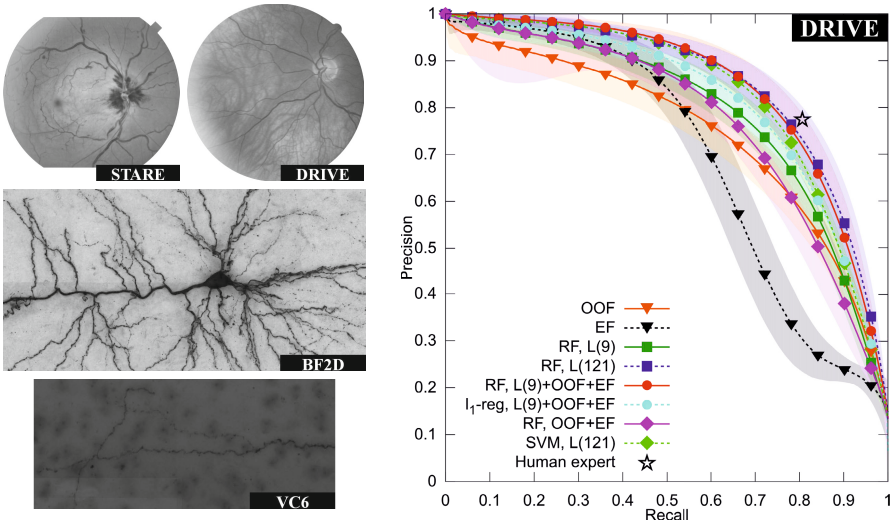

Fig. 1. Left: Sample images from the datasets we used to evaluate our approach. **Right:** Evaluation of different approaches on the DRIVE dataset. The state-of-the-art OOF and EF methods are significantly outperformed by learned features. However, this is only true when the number of learned features is very large, which makes this prohibitive in real medical imaging applications. We show that the handcrafted methods can be complemented by a very small number of learned features to obtain the same quality as learning only approaches, but at a fraction of their computational cost.

representation of the images, from which these images can be retrieved by applying a linear transformation. Such model was originally proposed in [14], and has been shown to be useful for image denoising and object recognition.

More exactly, we optimize the following objective function:

$$\underset{\{\mathbf{f}^j\},\{\mathbf{m}_i^j\}}{\operatorname{argmin}} \sum_i \left(\left\| \mathbf{x}_i - \sum_{j=1}^{N} \mathbf{f}^j * \mathbf{m}_i^j \right\|_2^2 + \lambda \sum_{j=1}^{N} \left\| \mathbf{m}_i^j \right\|_1 + \xi \sum_{j=1}^{N} \sum_{k \neq j} \left(\langle \mathbf{f}^j, \mathbf{f}^k \rangle \right)^2 \right). \quad (1)$$

The $\mathbf{x}_i$s are training images and the $\mathbf{f}^j$s are the learned filters. For each training image $\mathbf{x}_i$ the set $\{\mathbf{m}_i^j\}_{j=1\ldots N}$ is the corresponding representation. Each element $\mathbf{m}_i^j$ has the same size as $\mathbf{x}_i$. The $*$ symbol represents the convolution product. The second term in Eq. (1) forces the $\{\mathbf{m}_i^j\}$ representations to be sparse, while the third term was used in [17] to counteract the natural tendency of the filters to sometimes converge to similar solutions.

The minimization process alternates between the optimization with respect to the representations and the optimization with respect to the filters. For the former we adopt a proximal method [2], which in the case of ℓ_1-norm regularization simply consists in performing a step in the direction opposite to the gradient of the ℓ_2-regularized term, followed by a component-wise soft-thresholding of the

argument of the ℓ_1-penalized term. For the latter we use Stochastic Gradient Descent. The images are normalized to have zero mean and variance one, and the filters are constrained to have norm one to avoid trivial solutions [14].

2.2 Computing Feature Maps with the Learned Filters

Once the filters have been learned, we can use them to compute feature maps. We simply compute our feature maps by plain convolution:

$$\mathbf{L}_j = \mathbf{f}^j * \mathbf{x}\,. \tag{2}$$

Another option is to use the sparse representation $\{\mathbf{m}^j\}$ of the image $\mathbf{x}$ as feature maps. However, while sparse representations are important for the learning procedure, their effectiveness for classification has been recently questioned [16], and [17] shows that accuracy is not improved by using them as feature maps at run-time. Unreported experiments yield the same conclusion for our approach.

2.3 Description Vector and Classification

For a given input image, we compute several feature maps, namely one for EF, one for OOF, and one for each learned filter. For each image location (u, v), we obtain the vector:

$$\Big[\mathrm{EF}[u,v], \mathrm{OOF}[u,v], \mathbf{L}_1[u,v] \ldots \mathbf{L}_N[u,v]\Big]^T \tag{3}$$

we call descriptor below. We then apply a Random Forest or a ℓ_1-regularized logistic regressor on such descriptors to classify each image location as lying on a linear structure or on the background. Compared to a traditional logistic regressor, the optimized functional for the ℓ_1-regularized logistic regressor includes a ℓ_1 penalty on the weights, forcing them to be sparse [10].

3 Results and Discussion

In this section we first introduce the datasets we have adopted for the evaluation of our method, we then describe our evaluation setup, and we finally present our results and how they compare to existing approaches [1]. Note that the Rotational Features [8] were shown to be outperformed by [17] and, as such, we do not compare against this method.

3.1 Datasets

We use four datasets in our evaluations (see Figure 1-left):

The STARE dataset [9] is composed of 20 RGB retinal fundus slides, along with two different ground truth sets traced by two different human experts.

[1] The code, the datasets, and the extensive experimental results are available on the website `http://cvlab.epfl.ch/~rigamont`

Half of the images come from healthy patients and are therefore rather clean, while the other half present pathologies which partly occlude the underlying vasculatures and alter their appearances. Moreover, some images are affected by severe illumination changes which challenge automated algorithms.

The DRIVE dataset [20] is a set of 40 retinal scans captured for the diagnosis of systematic diseases. It is simpler than the STARE dataset in that the pathologies affecting the patients compromise less the image quality. The dataset is splitted in 20 training images and 20 test images, and ground truth data is available for both sets.

The BF2D dataset is made by minimum intensity projections of bright-field micrographs that capture neurons. The images have a very high resolution but exhibit a low signal-to-noise ratio, because of irregularities in the staining process, and the dendrites often appear as point-like structures which can be easily mistaken for the structured and unstructured noise affecting the images. As a consequence, the quality of the annotations is poor. Also, only two images have been annotated by a human expert. For this reason we have selected the image with the best ground truth as test image, and used the other image for training.

We created the VC6 dataset from a subset of the images composing the publicly available Visual Cortical Layer 6 Neuron dataset [4], which consists of 25 separated dendritic and axonal subtrees from one primary visual cortical neuron, sectioned into five physical slices. We have taken three image stacks from this dataset and computed their minimum intensity projections. These projections exhibit numerous artifacts and have a poor contrast. Their segmentation therefore represents a challenging undertaking for automated systems. We selected the first two images for the training of the algorithms, and retained the third one for testing. Ground truth data has been reconstructed from the traces made by the experts.

3.2 Experimental Setup

We first pre-processed the images in the datasets, converting them to grayscale and rescaling pixel values to zero-mean, unit-variance. For the retinal scans we only considered the green channel, since it has been shown to present the highest contrast between vessels and background [15]. We then computed the multiscale OOF and EF responses.

We also learned several filter banks of different cardinalities, at a single scale. The size of each filter has been fixed to 21×21 pixels to be consistent with the filter banks used in [17] [2]. We have experimented with smaller filter sizes, and the results show little influence. The gradient step, the regularization parameter, and all the other parameters involved in the filter learning were manually tuned.

[2] Learning a bank of 121 filters posed some problems in the VC6 dataset case, as low contrast and high noise prevented the learning process to get more than few dozens of meaningful filters, leading to poor performances (only slightly superior to those of 16 filters). To make a fair comparison, and for the 121 filters/VC6 case only, we have weighted the training images inversely proportionally to the OOF response, easing the learning process by focusing only on the parts where OOF responds weakly.

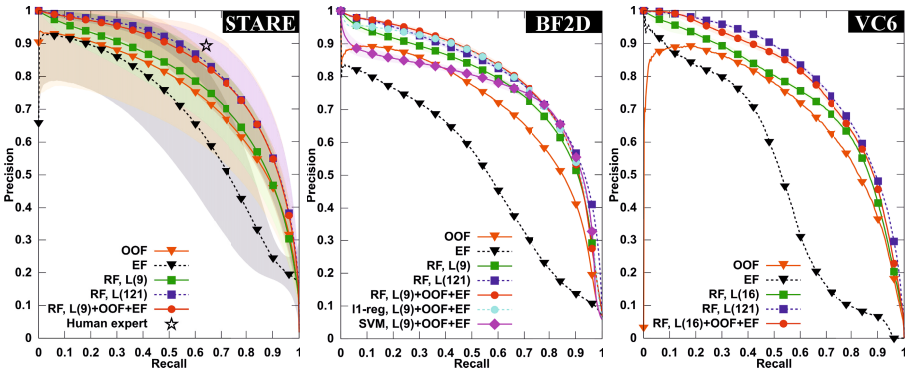

Fig. 2. Comparing EF,OOF, the method of [17] with few learned filters only, our approach, and the original method of [17] with 121 learned filters on the STARE, BF2D, and VC6 datasets (the results on DRIVE are given in Fig. 1-right). The cardinality of the learned filter banks (denoted as **L**) is given between parentheses. Note that the learning-based approaches outperform handcrafted methods even on the STARE dataset, where the used filters are not specifically tuned to the characteristics of the images, but are instead tuned to those of the DRIVE images. The results are averaged on 10 random trials and over the images of the different datasets. The shades represent 1 standard deviation around the mean value.

To test the generalization power of the learned filter banks, the results reported here for the STARE dataset were obtained with the filters learned on the DRIVE dataset, even though the former exhibits pathologies and illumination issues which are not present in the latter.

For classification, except when specifically noted, we have used 600 random trees learned on 10,000 positive and 10,000 negative samples. Comparisons between binary masks have been used as a metric in the evaluations.

3.3 Results and Discussion

The Precision/Recall curves [5] averaged over 10 random trials for different approaches are given in Figs 1-right and 2, while Fig. 3 depicts qualitative results for a randomly sampled region of a retinal scan. These figures show that our method and the method of [17] outperform the other methods, but ours is significantly faster: Table 1 details the average timings on the DRIVE dataset for the method we propose, and compares them with those of [17]. Because we use fewer filters and because the OOF and EF can be implemented in a very efficient, multi-threaded way, extracting the features is much faster in our approach. The gap widens considerably as higher-resolution images are considered.

Moreover, our descriptors are much more compact than [17]'s, as their sizes are divided by more than a factor of 10. This substantially speeds up the training and testing stages. While [17] considered only SVMs for classification, we found Random Forests and ℓ_1-regularized logistic regression well suited for the task at

Table 1. Average timings recorded for [17] and our approach on the 565×584 images of the DRIVE dataset. The time is expressed in seconds, except for the filter learning stage, and includes the time spent in reading/writing from disk ([17]: training 0.08s, testing 20s; our approach: training 0.01s, testing 2.8s). Although strongly parallelizable, implementations were restricted to use a single core to provide a fair evaluation. The OOF, which accounts for almost 50% of the time spent by our approach in the feature extraction phase, does not use an optimized implementation. The recorded SVM training timings do not include the time spent for the grid search on the parameters.

Method	Filter Learning	Feature Extraction	Training			Testing		
			RF	ℓ_1-reg	SVM	RF	ℓ_1-reg	SVM
[17]	several days	10.52	354.02	0.26	950.70	152.40	20.33	2568.53
our approach	several mins	2.12	55.91	0.05	210.56	86.70	2.84	455.97

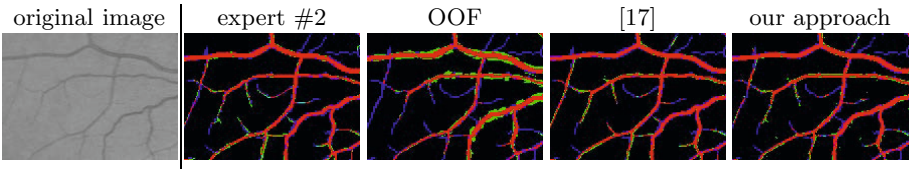

Fig. 3. Qualitative segmentation results on a randomly selected part of an image randomly chosen from the DRIVE dataset. True positives are outlined in red, false positives in green, and false negatives in blue. The segmentation accuracies for [17] and our approach are comparable, while our approach is much more efficient.

hand, obtaining comparable if not superior results in a fraction of the time (see Fig. 1-right), and without any need for a precise parameter tuning.

While the performance of logistic regression is inferior to that of Random Forests, it is still interesting to keep it into account: Its execution time is several orders of magnitude smaller, which can be appealing in practical applications.

Reducing the number of learned filters yields important speedups not only at run-time, but also during the learning of the filter banks themselves. A few minutes are typically required to learn a bank of 9 filters, which have to be compared with days for the larger filter banks in [17]. This, together with the reduced computational times, allows the practitioner to set up the segmentation pipeline from scratch and get state-of-the-art results within few dozens of minutes.

4 Conclusion

Through extensive experiments we showed that handcrafted and learned features can complement each other very well for the extraction of linear structures. This results in an efficient implementation, useful for practical applications. Our approach is general and could be used in domains where handcrafted methods exist, as is the case for the initial steps of many medical image processing algorithms, benefitting from improvements in the latter.

References

1. 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. ITB (2002)
2. Bach, F., Jenatton, R., Mairal, J., Obozinski, G.: Convex Optimization with Sparsity-Inducing Norms. MIT Press (2011)
3. Breiman, L.: Random Forests. Machine Learning (2001)
4. Brown, K., Barrionuevo, G., Canty, A., Paola, V.D., Hirsch, J., Jefferis, G., Lu, J., Snippe, M., Sugihara, I., Ascoli, G.: The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. Neuroinformatics (2011)
5. Davis, J., Goadrich, M.: The Relationship Between Precision-Recall and ROC Curves. In: ICML (2006)
6. Dima, A., Scholz, M., Obermayer, K.: Automatic Segmentation and Skeletonization of Neurons from Confocal Microscopy Images Based on the 3D Wavelet Transform. TIP (2002)
7. 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)
8. González, G., Fleuret, F., Fua, P.: Learning Rotational Features for Filament Detection. In: CVPR (2009)
9. Hoover, A., Kouznetsova, V., Goldbaum, M.: Location Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response. TMI (2000)
10. Koh, K., Kim, S.J., Boyd, S.: An Interior-Point Method for Large-Scale ℓ_1-Regularized Logistic Regression. JMLR (2007)
11. Krissian, K., Malandain, G., Ayache, N., Vaillant, R., Trousset, Y.: Model Based Detection of Tubular Structures in 3D Images. CVIU (2000)
12. Law, M.W.K., Chung, A.C.S.: Three Dimensional Curvilinear Structure Detection Using Optimally Oriented Flux. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 368–382. Springer, Heidelberg (2008)
13. Meijering, E., Jacob, M., Sarria, J.C., Steiner, P., Hirling, H., Unser, M.: Design and Validation of a Tool for Neurite Tracing and Analysis in Fluorescence Microscopy Images. Cytometry A (2004)
14. Olshausen, B., Field, D.: Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? Vision Res. (1997)
15. Patasius, M., Marozas, V., Jegelevicius, D., Lukoševičius, A.: Ranking of Color Space Components for Detection of Blood Vessels in Eye Fundus Images. In: EMBEC (2009)
16. Rigamonti, R., Brown, M., Lepetit, V.: Are Sparse Representations Really Relevant for Image Classification? In: CVPR (2011)
17. Rigamonti, R., Türetken, E., González, G., Fua, P., Lepetit, V.: Filter Learning for Linear Structure Segmentation. Tech. rep., EPFL (2011)
18. 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)

19. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikinis, R.: 3D Multi-Scale Line Filter for Segmentation and Visualization of Curvilinear Structures in Medical Images. Med. Image Anal. (1998)
20. Staal, J., Abràmoff, M., Niemeijer, M., Viergever, M., van Ginneken, B.: Ridge-Based Vessel Segmentation in Color Images of the Retina. TMI (2004)
21. Streekstra, G., van Pelt, J.: Analysis of Tubular Structures in Three-Dimensional Confocal Images. Network-Comp. Neural (2002)