

Neuromuscular fiber segmentation through particle filtering and discrete optimization

Thomas Dietenbeck^{a,b}, François Varray^a, Jan Kybic^b, Olivier Basset^a, Christian Cachard^a

^aUniversité de Lyon, CREATIS; CNRS UMR5220; INSERM U1044;
Université Lyon 1; INSA-LYON, France

^bCzech Technical University in Prague, Czech Republic

ABSTRACT

We present an algorithm to segment a set of parallel, intertwined and bifurcating fibers from 3D images, targeted for the identification of neuronal fibers in very large sets of 3D confocal microscopy images. The method consists of preprocessing, local calculation of fiber probabilities, seed detection, tracking by particle filtering, global supervised seed clustering and final voxel segmentation. The preprocessing uses a novel random local probability filtering (RLPF). The fiber probabilities computation is performed by means of SVM using steerable filters and the RLPF outputs as features. The global segmentation is solved by discrete optimization. The combination of global and local approaches makes the segmentation robust, yet the individual data blocks can be processed sequentially, limiting memory consumption. The method is automatic but efficient manual interactions are possible if needed. The method is validated on the Neuromuscular Projection Fibers dataset from the Diadem Challenge. On the 15 first blocks presented, our method has a 99.4% detection rate. We also compare our segmentation results to a state-of-the-art method. On average, the performances of our method are either higher or equivalent to that of the state-of-the-art method but less user interactions are needed in our approach.

Keywords: segmentation, confocal microscopy, neuromuscular fiber, neuron tracing, Diadem Challenge, SVM, particle filtering, N-cut

1. INTRODUCTION

In biomedical imaging, tubular structures appear frequently, representing for example blood vessels, bronchi, neuronal or muscular fibers. When analyzing neuromuscular projection fibers¹ (*cf.* Figure 1), finding the trajectories of individual fibers is crucial to answer a number of important biological questions contributing to understanding the function and structure of the neuronal system. This however requires to segment and distinguish the fibers from one another, which is made difficult by the fact that the fibers are usually intertwined. Manual or semi-automatic segmentation of such datasets can take weeks or months and there is thus a strong need for robust and fully automatic tracing methods. However, as demonstrated recently by the Diadem challenge*, despite a significant research effort, the problem is not yet fully solved.

Meijering² presents a comprehensive review of processing method, simulation tools and associated software related to neuron tracing. In particular, neuronal fibers were segmented by methods based on active contours³⁻⁵ and global optimization with geometric priors.⁶ While the results are promising, significant manual editing is needed to reach the quality required by the biologists.

In this paper, we propose a fiber segmentation and tracking method combining both global and local processing. The method is fully automatic but if needed, it allows manual interactions in a simple and efficient manner. It can distinguish several fibers and handle their bifurcations, circumventing the difficult problem of explicit branching point detection. It gains robustness from tracking only short distances from a number of initial seeds and combining these tracks together. It is designed to handle data consisting of a large number of slightly overlapping blocks with known relative geometrical positions, which is the case for large microscopy datasets. The blocks are processed sequentially, making the memory requirements manageable. This work is an extension of our previous work⁷ with the following differences: the computation of the fiber probability is made more robust by the use of an SVM classifier and a comparison is also made with another state-of-the-art method.⁴

*<http://diademchallenge.org/>

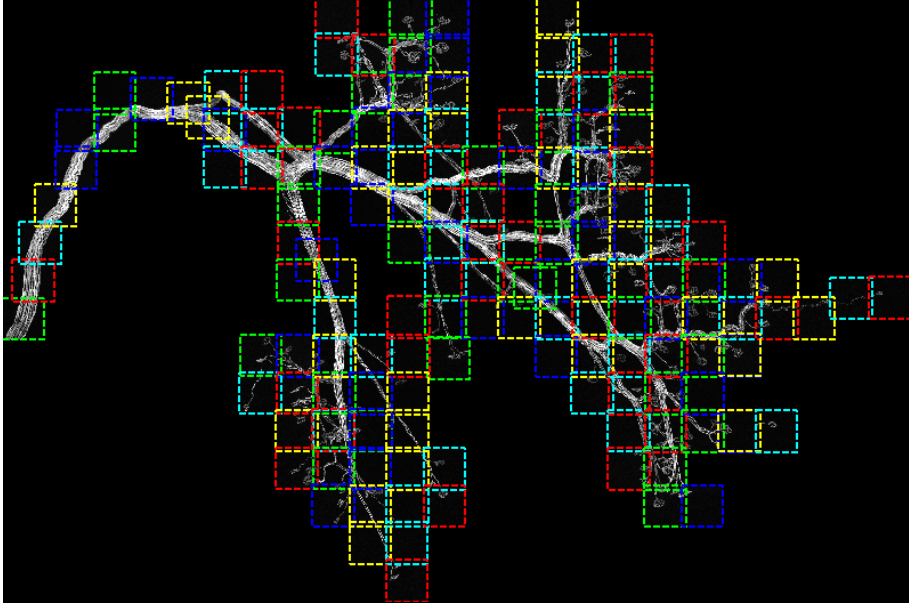


Figure 1: Maximum intensity projection of the neuromuscular fibers dataset with highlighted block boundaries in color. Each block is a grayscale 3D image stack.

2. METHOD

We propose a method (Algorithm 1) combining global and local approaches. First, our novel random local probability filtering (RLPF) is used to enhance the fiber location. Then, several features are computed based on steerable filters and a SVM classifier is used to obtain a fiber probability. A sparse set of seed points is detected by non-maxima suppression and a connection map (probabilities) between nearby seeds is estimated through particle filtering. Seeds are assigned to different fibers by a supervised discrete optimization graph clustering based on normalized cuts (N-cuts). Finally, fiber labels are assigned to all voxels. Let us now describe the method in detail.

2.1 Preprocessing

The aim of the preprocessing is to estimate a probability $p_{RLPF}(\mathbf{x})$ that a voxel \mathbf{x} belongs to a fiber. We will decompose it into two factors

$$p_{RLPF}(\mathbf{x}) = p_1(\mathbf{x})p_2(\mathbf{x}). \quad (1)$$

The first, p_1 , is based on the observation that fibers are bright and suppresses the dark background pixels:

$$p_1(\mathbf{x}) = \text{Sigmoid}(f(\mathbf{x}), a, b) = \frac{1}{1 + \exp\left(-\frac{f(\mathbf{x}) - a}{b}\right)}, \quad (2)$$

where f is the volume intensity and the parameters a and b are set conservatively to limit false negatives and are not critical. Typical values used are $a = 800$ and $b = 100$ for image intensities varying between 0 and 4095.

The second factor, p_2 , is calculated using random local probability filtering (RLPF), which estimates the probability that a voxel \mathbf{x} is brighter than voxels from its neighborhood $U(\mathbf{x})$:

$$p_2(\mathbf{x}) = \text{Prob}(f(\mathbf{x}) > f(\mathbf{y}); \mathbf{y} \in U(\mathbf{x})). \quad (3)$$

For efficiency, we approximate p_2 by sampling $f(\mathbf{y})$ at M normally distributed locations $\mathbf{h}_1, \dots, \mathbf{h}_M$ around \mathbf{x} and evaluating $p_2(\mathbf{x})$

$$p_2(\mathbf{x}) \approx \frac{1}{M} \|\{f(\mathbf{x}) > f(\mathbf{x} + \mathbf{h}_i); i = 1, \dots, M\}\|. \quad (4)$$

The parameter σ_l is related to the expected fiber diameter. If σ_l is too small, RLPF acts as a high pass filter, enhancing borders but suppressing fiber centers. For σ_l too high, fibers may not be sufficiently enhanced.

2.2 Fiber probability

The next step is to compute the probability that a voxel belongs to a fiber. Several approaches have been proposed to extract the fibers from the background such as tubularity measures,^{4,8} Optimally Oriented Flux,^{9–11} steerable filters and SVM classification.^{6,12} Following Gonzalez *et al.*,¹² we use a combination of the RLPF output and steerable filters as features for an SVM classifier. Steerable filters were introduced by Freeman and Adelson¹³ as an efficient way to rotate the filters to any orientation. Gaussian derivatives and their linear combinations are the best known class of such filters.^{14,15}

To classify a voxel as belonging either to a fiber or to the background, we first compute a local orientation through a modified Hessian method that relies on 2^{nd} -order steerable filters.¹⁵ The feature vector is then given by

$$\mathbf{V}(\mathbf{x}) = \left[f(\mathbf{x}), p_{\text{RLPF}}(\mathbf{x}), f(\mathbf{x}) * \left[G_{1,0,0}^\sigma, G_{0,1,0}^\sigma, G_{0,0,1}^\sigma, G_{2,0,0}^\sigma, G_{0,2,0}^\sigma, G_{0,0,2}^\sigma \right] \right] \quad (5)$$

where $G_{i,j,k}^\sigma$ corresponds to the Gaussian derivative kernel with normalized energy. It denotes the i^{th} derivative with respect to x , j^{th} derivative with respect to y and k^{th} derivative with respect to z and is steered to the estimated orientation. The feature vector is then used by an SVM classifier to estimate the likelihood p_{SVM} of the voxel being part of a fiber.

To train the SVM classifier, we selected 2500 voxels inside fibers as positive samples, while negative samples were taken both from voxels closed to fibers (2500 points) and far away (2500 points). The orientation for the steerable filters was computed using the same Hessian based method as used during the segmentation. Note that we also compared the performance of the classifier when using higher order derivatives as suggested in Gonzalez *et al.*¹² but found no differences in the classification and thus choose to only use the derivatives up to the second order.

2.3 Seed generation

Our strategy is to generate a sparse set of N_S points \mathbf{s}_k called seeds from which the fiber tracking is initiated. Compared to the strategy of starting from one initial point, the robustness is significantly increased, as failing to establish the correct connections of one seed will not ruin the whole tracking. A voxel \mathbf{x} is declared a seed if it is a local maximum¹⁶ in a neighborhood of size $[2k_x + 1; 2k_y + 1; 2k_z + 1]$ and if $p_{\text{SVM}}(\mathbf{x}) > 0.5$. Alternatively, seeds can be derived from the skeleton^{4–6} of a preliminary segmentation.

2.4 Particle Filtering

Standard particle filtering^{17,18} is typically used for parameter estimation even in the case of fiber tracking.^{19,20} Here however, we decided to use it to perform the tracking directly. It takes the local probabilities $p_{\text{SVM}}(\mathbf{x})$ and creates a connection field $\varphi(\mathbf{x})$, which approximates the probability density of a point \mathbf{x} being reachable from an initial point. This is achieved by launching N particles from the initial point and letting them evolve randomly.

The current state of a particle $k \in \{1, \dots, N\}$ at time $t \in \mathbb{N}$ is described by its position $\mathbf{x}_t^{(k)} \in \mathbb{R}^3$ and speed $\mathbf{v}_t^{(k)} \in \mathbb{R}^3$. The particles are moving inside the fibers with a constant speed and with a randomly altered direction at each iteration. More specifically, the state of the particles $(\mathbf{x}_t^{(k)}, \mathbf{v}_t^{(k)})$ is updated at each iteration as follows:

$$\mathbf{v}_t^{(k)} = \gamma \frac{\mathbf{v}_{t-1}^{(k)} + \mathbf{r}_t^{(k)}}{\|\mathbf{v}_{t-1}^{(k)} + \mathbf{r}_t^{(k)}\|} \quad (6)$$

$$\mathbf{x}_t^{(k)} = \mathbf{x}_{t-1}^{(k)} + \mathbf{v}_t^{(k)} \quad (7)$$

where $\mathbf{r}_t^{(k)}$ is a normally distributed zero mean random variable with a multidimensional covariance σ_r determining how curved the trajectories can be. The speed γ is set to 1 pixel per iteration. The initial state of each particle is $(\mathbf{x}_0, \mathbf{v}_0)$, where \mathbf{x}_0 are the seed coordinates and \mathbf{v}_0 optionally expresses direction preferences.

Each particle has an associated weight $w_t^{(k)}$. In the beginning, the weights are distributed uniformly, $w_0^{(k)} = 1/N$. At each iteration, the weights are updated according to the fiber probability field $p_{SVM}(\mathbf{x})$ and normalized:

$$\tilde{w}_t^{(k)} = w_{t-1}^{(k)} p_{SVM}(\mathbf{x}_t^{(k)}) \quad (8)$$

$$w_t^{(k)} = \frac{\tilde{w}_t^{(k)}}{\sum_{i=1}^N \tilde{w}_t^{(i)}}. \quad (9)$$

To avoid particle degeneration, *i.e.* very few particles having non-negligible weights, the particles are resampled as needed, cloning particles with large weight and suppressing low weight ones. The resampling is performed by creating a new set of N particles by drawing them randomly with repetition with probabilities equal to their weights $w_t^{(k)}$. Uniform weights are then assigned to the resampled particles. Resampling is performed when an effective number of particles

$$N_{\text{eff}} = \frac{1}{\sum_{i=1}^N (w_t^{(i)})^2} \quad (10)$$

is smaller than αN , with $\alpha = 0.1$.

The empirical probability

$$\varphi_t(\mathbf{x}) = \sum_{k=1}^N w_t^{(k)} \delta(\mathbf{x} - \mathbf{x}_t^{(k)}) \quad (11)$$

approximates the probability of a fiber being reachable from a given seed point \mathbf{x}_0 in time t . Its sum over time will be called a connection field and corresponds to the fiber being reachable at any time $t \leq N_{it}$. To obtain a useful estimate even for a small finite number of particles, the Dirac function $\delta(\cdot)$ in (11) is replaced with a Gaussian kernel with a covariance σ_φ :

$$\varphi_t(\mathbf{x}) = \sum_{k=1}^N w_t^{(k)} G(\mathbf{x} - \mathbf{x}_t^{(k)}) \quad (12)$$

with

$$G(\mathbf{x}) = (2\pi\sigma_\varphi)^{-\frac{3}{2}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma_\varphi}\right). \quad (13)$$

Two stopping criteria were used. First, the propagation is stopped after a fixed number of iteration N_{it} , which is chosen so that typical distance between seeds (determined by k_x, k_y, k_z) is covered multiple times. We choose $N_{it} = 5 \max(k_x, k_y, k_z)$. This way, a seed is usually connected to about 10 to 20 other seeds. Second, we stop if the sum $\sum_{i=1}^N \tilde{w}_t^{(i)}$ of unnormalized weights falls below a predetermined threshold ε (such as $\varepsilon = 10^{-3}$), assuming that all particles have left the fiber.

2.5 Supervised seed clustering

There is a known small number of fibers (classes) in the image. The class of these seeds is given by the user, and at least one seed is known for each fiber. The objective is then to assign each seed to its fiber, such that well connected seeds rest together. We calculate the connection field φ_k for all N_S seeds \mathbf{s}_k by launching the particles in state $\mathbf{x}_0 = \mathbf{s}_k, \mathbf{v}_k = 0$. A transition matrix is formed, storing the strength (estimation of probability) of pairwise connections between seeds

$$\tilde{\Phi} = \begin{pmatrix} \varphi_1(\mathbf{s}_1) & \varphi_2(\mathbf{s}_1) & \cdots & \varphi_{N_S}(\mathbf{s}_1) \\ \varphi_1(\mathbf{s}_2) & \varphi_2(\mathbf{s}_2) & \cdots & \varphi_{N_S}(\mathbf{s}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_1(\mathbf{s}_{N_S}) & \varphi_2(\mathbf{s}_{N_S}) & \cdots & \varphi_{N_S}(\mathbf{s}_{N_S}) \end{pmatrix} \quad (14)$$

and a symmetry is enforced

$$\Phi = \tilde{\Phi} + \tilde{\Phi}^T \quad (15)$$

Note that because the propagation length N_{it} is limited, only close seeds are connected, making the matrix Φ sparse.

The classification is performed by supervised graph clustering based on discrete optimization where the edge nodes V correspond to seeds and edge weights w_{ij} are given by the elements of the matrix Φ . Since the size of the fibers is about the same, we use an approach based on normalized cuts (N-cut) with constraints.^{21,22} We want to find an optimal classification $c : V \mapsto \{1, \dots, K\}$. The criterion to maximize is:

$$\sum_{k=1}^K \frac{\text{assoc}(C_k, C_k)}{\text{assoc}(C_k, V)} \quad (16)$$

where $C_k = \{i \in V, c(i) = k\}$ are nodes of class k and

$$\text{assoc}(A, B) = \sum_{i \in A, j \in B} w_{ij}. \quad (17)$$

The constraints are given by the user-provided attribution of initial seeds to classes. The problem can be reformulated as:

$$c^* = \operatorname{argmin} \sum_{l=1}^K \frac{X_l^T (D - W) X_l}{X_l^T D X_l} \quad (18)$$

$$\text{subject to } X = \{0, 1\}^{N \times K}, X 1_K = 1_N \text{ and } B^T X = b. \quad (19)$$

with $D = \text{diag}(W 1_N)$ a diagonal matrix and B representing the constraints. We use a fast algorithm²³ for solving two class normalized cut problems with linear constraints. To handle multi-class segmentation, an $\alpha\beta$ -swap method has been employed.²⁴

To decrease the impact of long paths, which may lead to incorrect segmentation of nodes far from the initial seeds, we replace weights w_{ij} by w_{ij}^p . This corresponds to measuring the link quality in equation (17) not by an l_1 -norm but some higher order l_p -norm. For very high p , we are approaching the l_∞ -norm, where the path length is not important at all and only the weakest link counts. In practice, we found $p = 3$ to work well.

Once the seeds have been classified, it is easy to propagate the classes to all voxels to obtain the final segmentation S : a voxel gets the class of a seed to which it is most strongly connected.

$$S(\mathbf{x}) = c^*(\operatorname{argmax}_i(\varphi_i(\mathbf{x}))). \quad (20)$$

2.6 Multiple blocks segmentation

So far, we have described how to treat a single block. However, in practice the dataset will consist of a large number of blocks, and the challenge is that all blocks will not fit into memory simultaneously.

Our strategy (Algorithm 1) is to perform the seed finding for each block separately. Then, the seeds are merged using the known relative positions of the blocks. Third, particle filtering is performed on each block, using all seeds falling into this block. The individual transition matrices are combined into a global transition matrix Φ which is then used to assign fiber classes to seeds. Finally, classes of each voxel are determined by equation 20.

To give the user a possibility to correct the results, he/she can force the class of the wrongly classified seeds. This is easy to accomplish by adding a new prior constraint. The discrete optimization algorithm is then re-run with the new constraints.

3. RESULTS

To evaluate the performances of our method, we used the Neuromuscular Projection Fiber dataset from the Diadem Challenge.¹ It consists of a set of 152 partially overlapping confocal microscopy 3D blocks, corresponding

Algorithm 1 Final multi-blocks fiber tracking algorithm in pseudo-code.

Input data: Set of 3D image blocks with known positions.

Output: An assignment of each voxel to one of the fibers or background.

for each 3D block **do**

 Compute the feature vector — RLPF volume p_{RLPF} and output of steerable filters (Section 2.2 and 2.1).

 Calculate the fiber probability p_{SVM} of each voxel.

 Extract seeds by non-maximum suppression.

end for

Merge all seeds into a common coordinate system.

for each 3D block **do**

 Find seeds located in this block (Section 2.3).

 Compute the sparse local transition matrix by particle filtering (section 2.4).

end for

Merge transition matrices into a global transition matrix.

while user not happy **do**

 Assign classes to all seeds by N-cut algorithm (Section 2.5).

for each 3D block **do**

 Compute final segmentation (equation 20).

end for

 Add class constraints manually if needed.

end while

Parameter	Value
k_x, k_y	8
k_z	4
N	500
σ_l	0.2
σ_φ	5
γ	1

Table 1: Parameter values used for the experiments.

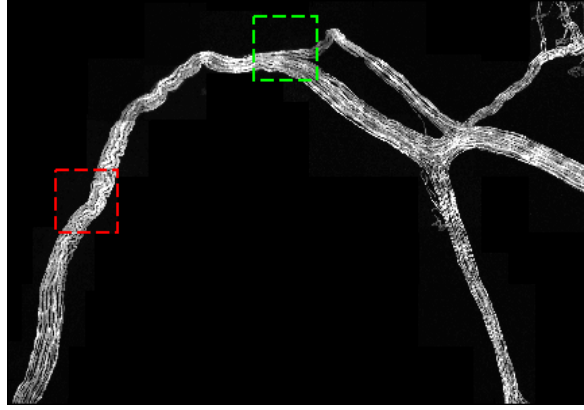
to about 36 GB of uncompressed data. The x, y position of the blocks was provided and the z offsets were determined by an automatic algorithm. A 2D maximum intensity projection (MIP) of the blocks is shown in Figure 1. A 2D projection of the fiber centerlines is also provided and used as ground truth (GT). The fibers are allowed to branch.

The parameters used are summarized in Table 1. Twelve initial fibers (classes) were identified in the first block. For computational and verification reasons, only the 31 first blocks of the volume have been individually segmented.

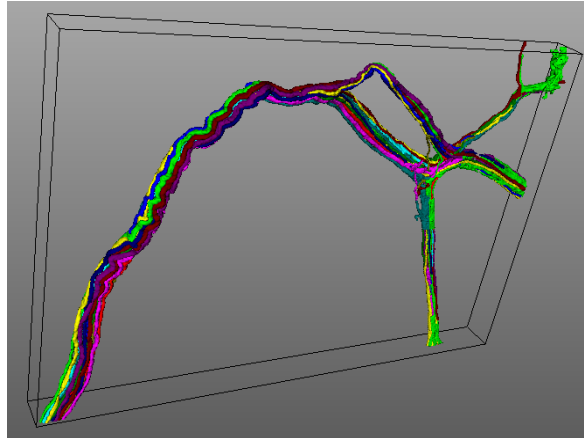
The segmentation is evaluated by counting the number of points of the GT that are correctly assigned. Figure 2 shows the segmentation result on the 31 first blocks of the dataset while Figure 3 and 4 show the 3D rendering [†] and orthogonal cuts through the red and green blocks in Figure 2(a) respectively. On the 15 first blocks our method has a 99.4% detection rate. When considering the more difficult set of 31 first blocks with more branching, the success rate drops to 70%, as several fibers are lost at the second branching.

We also compared our segmentation results to a state of the art method proposed recently by Wang *et al.*⁴ The comparison is based on 3 criteria: Precision, Recall and the number of manual interactions. Following the definition used in Wang *et al.*,⁴ the precision and recall are given by $\text{Precision} = l_{TP}/l_{Tot}$ and $\text{Recall} = l_{TP}/l_{GT}$,

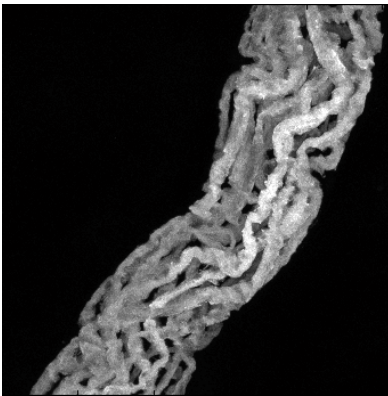
[†]The 3D rendering has been created using the CreaTools software http://www.creatis.insa-lyon.fr/site/en/CreaTools_home²⁵



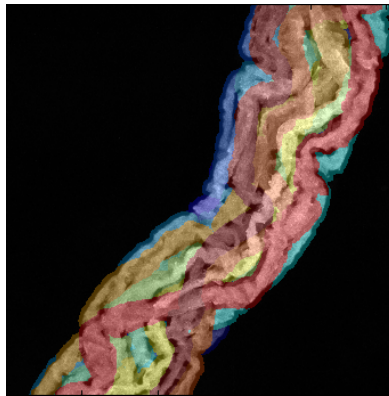
(a)



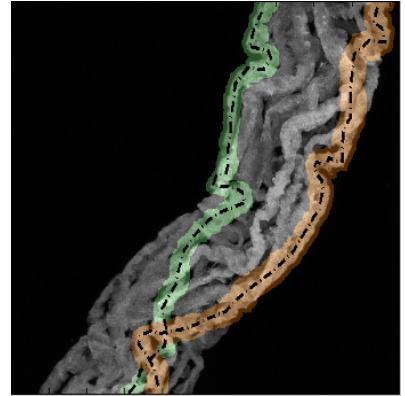
(b)



(c)



(d)



(e)

Figure 2: Results of the segmentation of the 31 first blocks of the dataset. (a) MIP image of the input blocks. (b) Final segmentation in 3D. (c) MIP image of the red square in (a). (d) Projection of the segmentation result on the MIP image. (e) Segmentation result for 2 fibers with the corresponding 2D ground truth in dashed lines (for clarity).

where l_{TP} is the total length of fibers detected correctly, l_{Tot} the total length of the detected fibers and l_{GT} the total length of the fibers given in the ground truth.

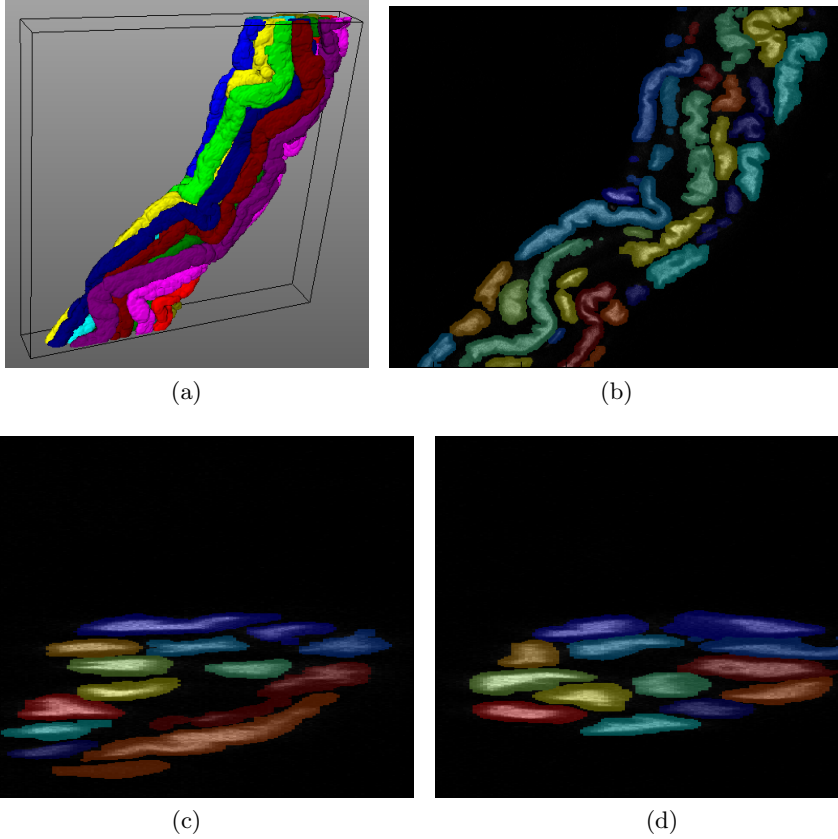


Figure 3: Results of the segmentation of the block 144 (red square in Figure 2). (a) 3D rendering of the labeling, Segmentation result superimposed on the data for (b) an XY plane, (c) an XZ plane and (d) a YZ plane.

Block	12	13	14	23	26	50	56	68	120	142	143	144	148	Average
PM	Precision	1	1	1	1	0.988	0.975	0.966	0.961	0.978	1	1	1	0.989
	Recall	1	1	0.997	1	0.916	0.941	0.946	0.94	0.962	0.989	1	1	0.978
	# MI	0	5	3	2	1	0	12	10	0	12	2	2	3.923
W ⁴	Precision	0.993	1	0.996	0.972	0.923	0.995	0.995	1	1	1	0.993	1	0.989
	Recall	1	0.971	1	1	1	0.956	0.994	0.934	0.955	0.882	1	1	0.976
	# MI	3	2	3	13	4	6	7	1	0	8	6	3	4.538

Table 2: Results of the proposed method (PM) and Wang *et al.*⁴ (W) in terms of precision, recall and the number of manual edit operations needed (# MI). The results of Wang *et al.* are extracted from their article.

Table 2 gives the error measures and the number of manual interactions needed by the expert to correct the algorithm’s output for each block and on average. From these figures we can see that both methods have approximately the same precision, while our method is able to provide a slightly higher recall; it performs better in 9 out of 13 cases both in terms of precision and recall. In addition, our method need less manual interactions.

4. CONCLUSIONS

We have presented a promising algorithm to segment fiber structure from microscopical images, especially for situations when the fiber can be relatively well separated from the background, when there is a number of nearby intertwined fibers which must be distinguished and when the dataset is too large to fit into memory. The SVM classification and the particle tracking, which are currently the computational bottleneck, are easily parallelisable. The strength of our technique is that the final segmentation is based on solving a global discrete optimization problem (labeling problem) for which proven solution methods exist. And while the algorithm is

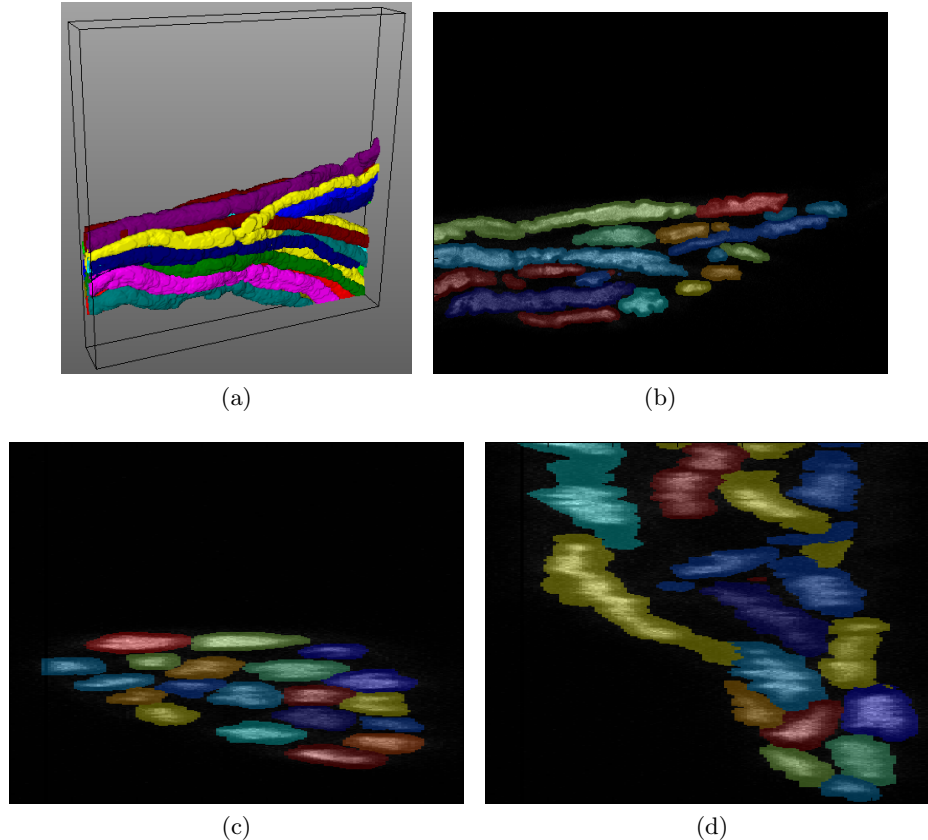


Figure 4: Results of the segmentation of the block 14 (green square in Figure 2). (a) 3D rendering of the labeling, Segmentation result superimposed on the data for (b) an XY plane, (c) an XZ plane and (d) a YZ plane.

not yet capable of segmenting these challenging datasets fully automatically, the amount of necessary manual interaction is quite low.

The method is validated on the Neuromuscular Projection Fibers dataset from the Diadem Challenge and compared to a state of the art method. On average, the performances of our method are either higher or equivalent to those of the state of the art method but less user interactions are required to correct the segmentation.

5. ACKNOWLEDGEMENT

The data was provided by J.Lichtman for the DIADEM Challenge, with precise 3D block alignment by F. Benmansour. The project was partially supported by Centre Lyonnais d’Acoustique (CeLyA), ANR grant n° 2011-LABX-014 and the Czech Science Foundation under project P202/11/0111.

REFERENCES

- [1] Lu, J., Tapia, J., White, O., and Lichtman, J., “The interscutularis muscle connectome,” *PLoS Biology* **7** (2009).
- [2] Meijering, E., “Neuron tracing in perspective,” *Cytometry Part A* **77A**, 693–704 (2010).
- [3] Rodriguez, A., Ehlenberger, D., Hof, P., and Wearne, S., “Three-dimensional neuron tracing by voxel scooping,” *J. Neuroscience Methods* **184**, 169–175 (2009).
- [4] Wang, Y., Narayanaswamy, A., Tsai, C.-L., and Roysam, B., “A Broadly Applicable 3-D Neuron Tracing Method Based on Open-Curve Snake,” *Neuroinformatics* **9**, 193–217 (2011).

- [5] Wang, Y., Narayanaswamy, A., and Roysam, B., “Novel 4-D open-curve active contour and curve completion approach for automated tree structure extraction,” in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR’2011)*], 1105–1112 (2011).
- [6] Türetken, E., González, G., Blum, C., and Fua, P., “Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors,” *Neuroinformatics* **9**, 279–302 (2011).
- [7] Varray, F., Kybic, J., Basset, O., and Cachard, C., “Neuromuscular fiber segmentation using particle filtering and discrete optimization,” in [*MICCAI Workshop: Histopathology Image Analysis (HIMA), Nice, France*], (2012).
- [8] Frangi, A., Niessen, W., Vincken, K., and Viergever, M., “Multiscale vessel enhancement filtering,” in [*Medical Image Computing and Computer Assisted Intervention (MICCAI’98)*], **1496**, 130–137 (1998).
- [9] Law, M. and Chung, A., “Three dimensional curvilinear structure detection using optimally oriented flux,” in [*European Conference on Computer Vision (ECCV’08)*], 368–382 (2008).
- [10] Benmansour, F. and Cohen, L. D., “Tubular structure segmentation based on minimal path method and anisotropic enhancement,” *Int. J. of Computer Vision* **92**(2), 192–210 (2011).
- [11] Türetken, E., Benmansour, F., and Fua, P., “Automated reconstruction of tree structures using path classifiers and mixed integer programming,” in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR’12)*], 566–573 (2012).
- [12] González, G., Aguet, F., Fleuret, F., Unser, M., and Fua, P., “Steerable features for statistical 3D dendrite detection,” in [*Medical Image Computing and Computer Assisted Intervention (MICCAI’09)*], 625–632 (2009).
- [13] Freeman, W. and Adelson, E., “The design and use of steerable filters,” *IEEE Trans. Pattern Anal. Machine Intell.* **13**, 891–906 (1991).
- [14] Jacob, M. and Unser, M., “Design of steerable filters for feature detection using canny-like criteria,” *IEEE Trans. Pattern Anal. Machine Intell.* **26**, 1007–1019 (2004).
- [15] Aguet, F., Jacob, M., and Unser, M., “Three-dimensional feature detection using optimal steerable filters,” in [*IEEE International Conference on Image Processing (ICIP’05), Genova, Italy*], 158–1161 (2005).
- [16] González, G., Fleuret, F., and Fua, P., “Automated delineation of dendritic networks in noisy image stacks,” in [*European Conference on Computer Vision (ECCV’2008)*], (2008).
- [17] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. Signal Process.* **50**(2), 174–188 (2002).
- [18] Blanco, J., Gonzalez, J., and Fernandez-Madrigal, J., “Optimal filtering for nonparametric observation models: Applications to localization and slam,” *The International J. of Robotics Research* **29**(14), 1726–1742 (2010).
- [19] Magee, D., Bulpitt, A., and Berry, E., “3D automated segmentation and structural analysis of vascular trees using deformable models,” in [*IEEE Workshop on Variational and Level Set Methods in Computer Vision*], 119–126 (2001).
- [20] Florin, C., Paragios, N., and Williams, J., “Particle filters, a quasi-monte-carlo solution for segmentation of coronaries,” in [*Medical Image Computing and Computer Assisted Intervention (MICCAI’05)*], 246–253 (2005).
- [21] Shi, J. and Malik, J., “Normalized cuts and image segmentation,” *IEEE Trans. Pattern Anal. Machine Intell.* **22**(8), 888–905 (2000).
- [22] Yu, S. and Shi, J., “Segmentation given partial grouping constraints,” *IEEE Trans. Pattern Anal. Machine Intell.* **26**(2), 173–183 (2004).
- [23] Xu, L., Li, W., and Schuurmans, D., “Fast normalized cut with linear constraints,” in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR’09)*], 2866–2873 (2009).
- [24] Eriksson, A., Olsson, C., and Kahl, F., “Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints,” in [*IEEE International Conference on Computer Vision (ICCV’07)*], 1–8 (2007).
- [25] Dávila, E., Guigues, L., Roux, J.-P., Cervenansky, F., Pop, S., Reyes, J., Flórez-Valencia, L., Hoyos, M. H., and Orkisz, M., “Creatools: applications and development framework for medical image-processing software,” in [*ISBI Workshop on Open Source Medical Image Analysis Software*], (2012).