

# Anisotropic ssTEM Image Segmentation Using Dense Correspondence across Sections

Dmitry Laptev, Alexander Vezhnevets,  
Sarvesh Dwivedi, and Joachim M. Buhmann

Department of Computer Science, ETH Zurich, Switzerland  
{dlaptev,alexander.vezhnevets,sdwivedi,jbuhmann}@inf.ethz.ch

**Abstract.** Connectomics based on high resolution ssTEM imagery requires reconstruction of the neuron geometry from histological slides. We present an approach for the automatic membrane segmentation in anisotropic stacks of electron microscopy brain tissue sections. The ambiguities in neuronal segmentation of a section are resolved by using the context from the neighboring sections. We find the global dense correspondence between the sections by SIFT Flow algorithm, evaluate the features of the corresponding pixels and use them to perform the segmentation. Our method is 3.6 and 6.4% more accurate in two different accuracy metrics than the algorithm with no context from other sections.

**Keywords:** Membrane Segmentation, Anisotropic Data, Dense Correspondence, SIFT Flow.

## 1 Introduction

Neuroanatomists face the challenging task of reconstructing neuronal structure with synaptic resolution in order to gain insights into the functional connectivity of brain. Performing this geometry extraction manually has been demonstrated to be tedious, error prone and requires an impractical amount of time. Therefore, accurate algorithms for automatic neuronal segmentation are indispensable for large scale geometric reconstruction of densely interconnected neuronal tissue. In this paper we focus on the segmentation problem, i.e., to annotate neuronal structures in tissue as either membranes or the inside volume of neurons.

Currently serial section transmission electron microscopy (ssTEM) [5] is the only available technique which can provide sufficient resolution. ssTEM data depicts the observed volume as a stack of images (sections). This imaging technique visualizes the resulting volumes in a highly “anisotropic” way, i.e., the  $x$ - and  $y$ -directions<sup>1</sup> have a high resolution, whereas the  $z$ -direction has a low resolution, primarily dependent on the precision of serial cutting.

Local appearance around the pixel in a section may be insufficient to discriminate between the membrane or the inner area of a neuron. This ambiguity arises

---

<sup>1</sup> Here,  $x$  and  $y$  coordinates correspond to the dimensions of a section, and  $z$  corresponds to the vertical dimension of the stack.

from the fact that electron microscopy produces the images as a projection of the whole section, so some of the membranes that are not orthogonal to a cutting plane can appear very blurred.

To allow automatic methods to exploit information from neighboring sections we have to resolve the correspondence problem - finding a mapping from a neighboring section to the current one. We propose to solve this problem by finding global dense correspondence with SIFT flow algorithm [10] and to use the features from different sections to perform segmentation.

## 1.1 Related Work

There are three general approaches for anisotropic data segmentation. The first approach focuses on the detection of neuron membranes in each section independently [7]. The software package Fiji [1] implements this approach: first, in every pixel the vector of features is evaluated, and then this vectors are used to train Random Forest classifier. We use this package for feature extraction, described in details in Section 2.3.

The second approach incorporates context from different sections without correspondence alignment. In [8] the authors propose two terms for graph cut segmentation, one of them incorporates context from neighboring sections. In contrast to our algorithm, this term depends only on the feature vector evaluated in the pixel in a direct z-neighborhood, with no correspondence alignment. As the difference between the sections is usually quite significant, incorporating of this term doesn't lead to significant improvements.

The third approach [15] generates many, possibly contradictory, segmentation hypotheses in individual sections and combine them in order to optimize the global agreement functional defined on the whole stack. In contrast to this approach, we are not dealing with given segmentation hypotheses, but incorporate the context from neighboring sections to improve the segmentation of every single section.

The novel contribution describes how to exploit context from neighboring sections by solving the correspondence problem. We present it in the following sections.

## 2 Proposed Method

Let  $\tau = \{I^k, Y^k\}_{k=1}^K$  be a training set, consisting of  $K$  images with a given labeling. Here  $I^k = \{x_p^k\}_{p=1}^N$  represents an input image of section  $k$ ,  $x_p^k$  corresponds to a pixel in section  $k$ .  $Y^k = \{y_p^k\}_{p=1}^N$  represents the labels of the corresponding pixels  $p$  for a section  $k$ .  $y_p^k$  equals 1 for the class "membrane" and 0 otherwise. Let  $\varphi(x_p^k)$  be a feature vector for the pixel  $x_p^k$ . Our goal is to build a segmentation algorithm that would automatically label new sets of images.

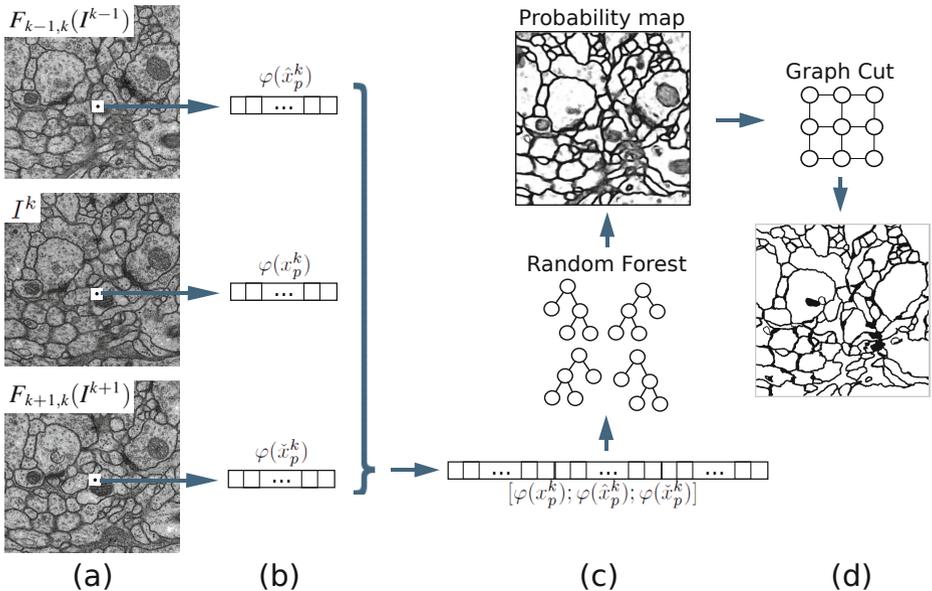
The proposed method constructs a dense correspondence between the neighboring sections and it uses features that are evaluated in all the corresponding

pixels for classification. Our workflow is illustrated in Figure 1. For a given section  $I^k$  we first find warpings from the neighboring sections  $I^{k+1}$  and  $I^{k-1}$ :  $F_{k+1,k}$  and  $F_{k-1,k}$ . Then, for every pixel  $x_p^k$  we find the corresponding pixels  $\hat{x}_p^k$  and  $\check{x}_p^k$ . Next, we calculate features in all three pixels  $\varphi(\hat{x}_p^k)$ ,  $\varphi(x_p^k)$ ,  $\varphi(\check{x}_p^k)$ , concatenate the feature vectors and use this extended feature vector as input to a Random Forest (RF) classifier. Finally, we use the probabilities returned by the RF for Graph Cut segmentation.

## 2.1 Framework

Suppose we are given a non-linear warping  $F_{k-1,k}$  that establishes the correspondence between the pixels in the image  $I^{k-1}$  and  $I^k$ . We discuss a method to obtain it in Section 2.2. We introduce two more images to the dataset:  $I^0 \equiv I^1$  and  $I^{K+1} \equiv I^K$  both for training and test sets, so that now there are two neighbors for every section from 1 to  $K$ . Every pixel  $x_p^k$  is then assigned to the corresponding pixels in the neighboring sections:

$$\hat{x}_p^k = F_{k-1,k}(x_p^k), \quad \check{x}_p^k = F_{k+1,k}(x_p^k). \quad (1)$$



**Fig. 1.** Based on the non-linear correspondings  $F_{k-1,k}$  and  $F_{k+1,k}$  the algorithm evaluates the warped images  $F_{k-1,k}(I^{k-1})$  and  $F_{k+1,k}(I^{k+1})$  (a). Then, feature vectors in the corresponding pixels are evaluated:  $\varphi(\hat{x}_p^k)$ ,  $\varphi(x_p^k)$ ,  $\varphi(\check{x}_p^k)$  (b). After that the method concatenates them and passes the concatenated feature vector to a RF (c). RF returns a probability map that is segmented by Graph Cut algorithm (d).

To incorporate the context from neighboring sections, an extended feature vector has to capture the contextual feature information associated with the pixel  $x_p^k$  itself, as well as with the pixels  $\hat{x}_p^k$  and  $\check{x}_p^k$ . The extended feature vectors form a training set for a RF classifier [4].

$$\tau = \{[\varphi(x_p^k); \varphi(\hat{x}_p^k); \varphi(\check{x}_p^k)], y_p^k, 1 \leq p \leq N, 1 \leq k \leq K\}. \quad (2)$$

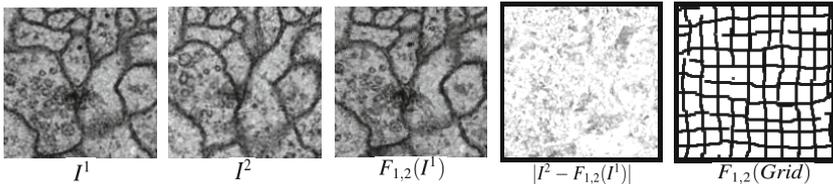
A trained RF returns the probability of every pixel of the image to belong to a membrane, i.e., a probability map. Afterwards, graph cut segmentation [3] with the probability map as unary potentials partitions the image into semantically meaningful segments.

## 2.2 Dense Correspondence

To find a dense correspondence between the sections we use the recently proposed method ‘‘SIFT Flow’’ [10]. SIFT Flow finds the non-linear warping  $F_{1,2}$  on the pixel grid  $x_p^1$  between the images  $I^1$  and  $I^2$  by minimizing the following energy:

$$E(F_{1,2}) = \sum_{p=1}^N \min(\|s(x_p^2) - s(F_{1,2}(x_p^1))\|, t) + \sum_{p=1}^N \gamma D(x_p^1, F_{1,2}(x_p^1)) + \sum_{(p,q) \in \epsilon} \min(\alpha D(F_{1,2}(x_p^1), F_{1,2}(x_q^1)), d). \quad (3)$$

$E(F_{1,2})$  is comprised of a data term, a small displacement term and a smoothness term. The first term constrains the SIFT descriptors  $s(x_p^2)$  [11] evaluated in pixel  $x_p^2$  to be matched along with the descriptors evaluated in pixel  $F_{1,2}(x_p^1)$ . The small displacement term constrains the changes between the original image and a wrapped one to be as small as possible.  $D$  is equal to the distance between the two pixels in a pixel grid. The smoothness term constrains the transformation of adjacent pixels to be similar. In this objective function, truncated  $L_1$  norms are used in both the data term and the smoothness term to account for matching outliers and discontinuities, with  $t$  and  $d$  as the threshold, respectively. Figure 2 shows the results of applying SIFT Flow algorithm to a drosophila larva data set. For further information we refer to [10].



**Fig. 2.** An example of non-linear warping between images  $I^1$  and  $I^2$  found by SIFT Flow. Image  $F_{1,2}(I^1)$  shows the warping applied to image  $I^1$  and image  $F_{1,2}(Grid)$  shows the warping applied to a grid image.

### 2.3 Features

We use 626 pixel features. When we incorporate the features from the neighboring section this number increases to 1878. RF performs well even in presence of lots of noisy features [4], therefore we need no feature selection procedure.

The whole set of features provided by [1] is used in this study: Gaussian blur, Sobel filter, Hessian, Difference of gaussians, Membrane projections, Variance, Mean, Minimum, Maximum, Median, Anisotropic diffusion, Bilateral, Lipschitz, Kuwahara, Gabor, Laplacian, Structure, Derivatives. Additionally, we incorporate newly developed features that proved to be informative for neuronal reconstruction: radon-like features [9], ray features [12] and line filter transform [13]. Also we use all the components of SIFT histogram [11] in the pixel.

### 2.4 Graph Cut Segmentation

We use graph cut segmentation to take into account the fact that the labels of the neighboring pixels are more likely to have the same label. For simplicity we drop the upper index in the following equations, as graph cut algorithm deals with one section at a time:  $y_p = y_p^k$ .

The segmentation task is formulated as an energy minimization problem  $\hat{Y} = \arg \min_Y E(Y)$ , where

$$E(Y) = \sum_{p=1}^N E_u(y_p) + \lambda_s \sum_{(p,q) \in \epsilon} E_s(y_p, y_q) + \lambda_{gf} \sum_{p=1}^N E_{gf}(y_p) + \lambda_{gc} \sum_{(p,q) \in \epsilon} E_{gc}(y_p, y_q). \quad (4)$$

Here the first term is a unary potential that equals to the negative log probabilities given by the RF in every pixel. Let  $i(x_p)$  be an intensity of the image in pixel  $x_p$ . Then the second term is a smoothness term:

$$E_s(y_p, y_q) = \exp\left(-\frac{(i(x_p) - i(x_q))^2}{2\sigma_s^2}\right) \frac{\delta(y_p, y_q)}{D(x_p, x_q)}, \quad (5)$$

where  $\delta(y_p, y_q)$  is a Kronecker function that equals 0 if  $y_p = y_q$  and 1 otherwise.

The gradient flux term [14] is defined as follows:

$$E_{gf}(y_p) = \begin{cases} \max(0, F(x_p)) & \text{if } y_p = 1 \\ -\min(0, F(x_p)) & \text{if } y_p = 0, \end{cases} \quad (6)$$

where  $F(x_p)$  denotes a gradient flux,  $F(x_p) = \sum_{x_q: (x_p, x_q) \in \epsilon} \langle u_{x_p, x_q}, v_{x_p} \rangle$ ,  $u_{x_p, x_q}$  represents a unit vector pointing from pixel  $x_p$  to the neighboring pixel  $x_q$  and vector  $v_{x_p}$  corresponds to the gradient vector at pixel  $x_p$ .

The good-continuation term [8] is defined as follows:

$$E_{gc}(y_p, y_q) = |\langle v_{x_p}, u_{x_p, x_q} \rangle| \exp\left(-\frac{(i(x_p) - i_m)^2}{2\sigma_c^2}\right) \frac{\delta_{\rightarrow}(y_p, y_q)}{D(x_p, x_q)}, \quad (7)$$

The variable  $i_m$  encodes the average gray value of membrane pixels and  $\sigma_{gc}$  is estimated as the variance of these gray values. The factor  $\delta_{\rightarrow}(y_p, y_q) = 1$  for  $y_p = 1, y_q = 0$  and equals 0 for all other cases.

The minimum of  $E(Y)$  is computed by max-flow/min-cut computation [3]. The cross-validation procedure determines the unknown parameters  $\lambda_s, \lambda_{gf}, \lambda_{gc}$  such that the results generalize in an optimal way.

As a post-processing procedure two steps are performed iteratively: region removing and line filter transform [13]. Region removing is performed by a series of thresholding operations based on region properties such as Area, Solidity, Euler Number and Eccentricity.

### 3 Experiments

**Data.** Our experiments are performed with the data provided for the ISBI 2012 challenge “Segmentation of neuronal structures in EM stacks” [2]. The dataset [5] is comprised of a training and a test set. Each set consists of 30 sections from a ssTEM of the *Drosophila* first instar larva ventral nerve cord (VNC), imaged at a resolution of 4x4x50 nm/pixel and cover a 2x2x1.5 micron cube of neural tissue. Training and test sets are taken from different volumes of the same VNC.

**Error Metrics.** There are two metrics used for the task of membrane segmentation: *Pixel error* and *Splits and Mergers Warping error*. Given the estimated labeling  $\hat{Y}$  and ground truth  $Y^*$ , the pixel error is defined as the Hamming distance between the two labelings  $\sum_p \delta(\hat{Y}_p, Y_p^*)$ .

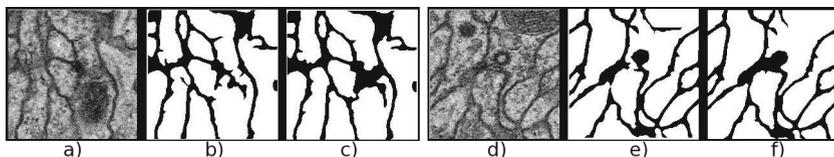
Splits and Mergers Warping error is a segmentation metric that penalizes topological disagreements between the two labelings [6]. The warping error is the squared Euclidean distance between  $Y^*$  and the “best warping”  $L$  of  $\hat{Y}$  onto  $Y^*$  such that the warping  $L$  is from the class  $\mathcal{A}$  that preserve topological structure:  $\min_{L \in \mathcal{A}} \sum_p \delta(L(\hat{Y})_p, Y_p^*)$ .

Both types of errors are evaluated automatically on the test set when the results are submitted to the testing server. The challenge also provides the error value caused by discrepancy in human labeling.

#### 3.1 Results

Our experiments are conducted with the default parameters of the SIFT flow algorithm:  $\gamma = 0.05, t = 0.1, \alpha = 2, d = 40$ . We compare the results of three different versions of our algorithm: with no context from neighboring sections (*one slice*), with direct correspondence (we incorporate the context from the pixels being a direct z-neighbors, with no warping procedure), and with dense correspondence found by SIFT flow algorithm.

Results are presented in Table 1. Some examples of the resulting images are presented in Figure 3. Incorporating the context from the neighboring sections with direct correspondence leads to improvement in terms of pixel error, but it performs worse in terms of warping error. On the other hand, using dense



**Fig. 3.** Original images: (a, d), one slice results: (b, e), dense correspondence: (c, f)

**Table 1.** Comparison of error results on a testing set for different versions of the algorithm and the results of other teams. **Human** denotes the error of human annotators.

Method	Pixel error	Warping error	Method	Pixel error	Warping error
<b>Human</b>	$6.7 * 10^{-2}$	$3.4 * 10^{-4}$	<i>IDSIA</i>	$6.0 * 10^{-2}$	$4.3 * 10^{-4}$
Dense <i>ETH</i>	$7.9 * 10^{-2}$	$6.2 * 10^{-4}$	<i>CSIRO</i>	$8.7 * 10^{-2}$	$6.8 * 10^{-4}$
Direct <i>ETH</i>	$8.0 * 10^{-2}$	$6.5 * 10^{-4}$	<i>Utah</i>	$1.3 * 10^{-1}$	$1.6 * 10^{-2}$
One slice <i>ETH</i>	$8.5 * 10^{-2}$	$6.4 * 10^{-4}$	<i>NIST</i>	$1.5 * 10^{-1}$	$1.6 * 10^{-2}$

correspondence leads to improvement in both objectives: 3.6% improvement in warping error and 6.4% for pixel error.

Most of other algorithms applied in the ISBI challenge exploited the context of only one single slice. S. Iftikhar & A. Godil (NIST) and X. Tan & C. Sun of CSIRO Enquiries employed Support Vector Machine (SVM) as a classifier. A team from Scientific Computing and Imaging Institute, University of Utah<sup>2</sup> designed Series of Classifiers and Watershed Tree. The Swiss AI Lab IDSIA team<sup>3</sup> trained Deep Neural Networks which appeared to be competitive to ours and their solution was slightly better in quantitative terms. This approach, however, requires almost a week of training time with specialized hardware, and it is therefore much more difficult to apply in real-world scenarios.

## 4 Conclusion

This paper addresses the problem of automatic membrane segmentation in stacks of electron microscopy brain tissue sections. Since the image stacks in our applications are anisotropic, we are not able to exploit information from neighboring sections by exploring direct  $z$ -neighborhood. This paper demonstrates, for the first time, how to exploit context information from neighboring sections by robustly solving the correspondence problem.

We show that this problem can be effectively solved with the SIFT Flow algorithm. Our method calculates features in all the corresponding pixels, concatenates their feature vectors and uses this extended feature vector for a RF classifier. Finally Graph Cut segmentation is performed. The proposed method is 3.6% more accurate for warping error, and 6.4% for pixel error.

<sup>2</sup> T. Liu, M. Seyedhosseini, E. Jurrus, & T. Tasdizen.

<sup>3</sup> D. Ciresan, A. Giusti, L. Gambardella, & J. Schmidhuber.

**Acknowledgement.** We like to thank Verena Kaynig-Fittkau, GVI Group Harvard, for valuable discussions. This work was partially supported by the SNF grant Sinergia CRSII3\_130470/1.

## References

1. Advanced Weka Segmentation (Fiji), <http://bit.ly/MdCr0v>
2. ISBI 2012 challenge, <http://bit.ly/riGDUm>
3. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Trans. Pattern Anal. Mach. Intell.* 26, 1124–1137 (2004)
4. Breiman, L.: Random forests. *Machine Learning* 45(1), 5–32 (2001)
5. Cardona, A., Saalfeld, S., Preibisch, S., Schmid, B., Pulokas, A.C.J., Tomancak, P., Hartenstein, V.: An integrated micro- and macroarchitectural analysis of the drosophila brain by computer-assisted serial section electron microscopy. *PLoS Biol.* 10 (2010)
6. Jain, V., Bollmann, B., Richardson, M., Berger, D.R., Helmstaedter, M., Briggman, K.L., Denk, W., Bowden, J.B., Mendenhall, J.M., Abraham, W.C., Harris, K.M., Kasthuri, N., Hayworth, K.J., Schalek, R., Tapia, J.C., Lichtman, J.W., Seung, H.S.: Boundary learning by optimization with topological constraints. In: *CVPR*, pp. 2488–2495 (2010)
7. Kaynig, V., Fuchs, T.J., Buhmann, J.M.: Geometrical Consistent 3D Tracing of Neuronal Processes in ssTEM Data. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) *MICCAI 2010, Part II*. LNCS, vol. 6362, pp. 209–216. Springer, Heidelberg (2010)
8. Kaynig, V., Fuchs, T.J., Buhmann, J.M.: Neuron geometry extraction by perceptual grouping in sstem images. In: *CVPR*, pp. 2902–2909. IEEE (2010)
9. Kumar, R., Reina, A.V., Pfister, H.: Radon-like features and their application to connectomics. In: *MMBIA*. IEEE (2010)
10. Liu, C., Yuen, J., Torralba, A.: Sift flow: Dense correspondence across scenes and its applications. *Trans. Pattern Anal. Mach. Intell.* 33(5), 978–994 (2011)
11. Lowe, D.G.: Object recognition from local scale-invariant features. In: *ICCV*, p. 1150. IEEE (1999)
12. Lucchi, A., Smith, K., Achanta, R., Lepetit, V., Fua, P.: A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) *MICCAI 2010, Part II*. LNCS, vol. 6362, pp. 463–471. Springer, Heidelberg (2010)
13. Sandberg, K., Brega, M.: Segmentation of thin structures in electron micrographs using orientation fields. *Journal of Structural Biology* 157(2), 403–415 (2007)
14. Vasilevskiy, A., Siddiqi, K.: Flux maximizing geometric flows. *Trans. Pattern Anal. Mach. Intell.* 24, 1565–1578 (2001)
15. Vazquez-Reina, A., Huang, D., Gelbart, M., Lichtman, J., Miller, E., Pfister, H.: Segmentation fusion for connectomics. In: *ICCV*. IEEE (2011)