# Automated Segmentation of Synapses in 3D EM Data

A. Kreshuk*, C.N. Straehle*, C. Sommer*, U. Koethe*, G. Knott†, F.A. Hamprecht*,[1]

*Heidelberg Collaboratory for Image Processing (HCI), Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg, Germany, †Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland

[1] Corresponding author, fred.hamprecht@iwr.uni-heidelberg.de.

**Abstract**

This contribution presents a method for automatic detection of exitatory, asymmetric synapses and segmentation of synaptic junctional complexes in stacks of serial electron microscopy images with nearly isotropic resolution. The method uses a Random Forest classifier in the space of generic image features, computed directly in the 3D neighborhoods of each pixel, and an additional step of interactive probability maps thresholding. On the test dataset, the algorithm missed considerably less synapses than the human expert during the ground truth creation, while maintaining an equivalent false positive rate. The algorithm is implemented as an extension to the Interactive Learning and Segmentation Toolkit "ilastik" and is freely available on our website.

## 1 INTRODUCTION

Segmentation of neuronal structures by image analysis is an essential technique for understanding their form and function. Automated routines for segmenting the synaptic contacts will present useful tools for characterising large volumes of brain tissue, providing the ability to make important comparisons between different regions or effects of activity.

Despite the latest improvements in the resolution of fluorescent light microscopy, electron microscopy remains the only tool capable of imaging synapses and analyzing different aspects of their morphology. The introduction of the FIB/SEM microscopy technique [1], with isotropic resolution reaching 5nm, allowed for synapse counting directly in 3D volumes, without the need to resort to stereological extrapolations from individual 2D scans [2]. But, even for image stacks with high

z-resolution, synapse detection still has to be performed manually and remains a tedious and time consuming task. In this contribution, we propose to alleviate this problem by introducing an interactive tool for semi-automatic detection of asymmetric synapses in 3D data.

It is implemented as an extension of the "ilastik" Interactive Learning and Segmentation Toolkit [3], is open source and is freely available on the web. The 2-step segmentation procedure is based on adaptive thresholding of Random Forest [4] classification results. In the first step, the classifier is trained on the labels provided by the user on a small subset of data in the interactive mode. The classifier can give real time feedback, which helps the user to concentrate on the areas of high uncertainty and therefore reduces the necessary labeling time. Classification is based on generic (non-linear) features of each pixel neighborhood, computed directly in 3D. In the second step, a user-defined threshold is applied on the predicted class probability maps and the pixels which have a high probability of belonging to a synapse are joined into synapse candidate objects. Additional size-based filtering is performed on the candidates to eliminate clearly non-synapse false positive detections and the remaining synapse candidates are expanded by all the adjacent pixels with synapse class probability greater than background class probability. The segmentation results can be saved into many popular image formats as well as into an hdf5 file for subsequent analysis.

## 2 RELATED WORK

For fluorescent light microscopy data, an approach similar to ours has recently been proposed by [5]. It uses a linear kernel SVM and the prediction is based on the intensity values in the 2D pixel neighborhoods, which limits the use case to data with synapse-specific staining. Similarly, [6] proposes a technique of synaptic Brainbow to label and detect synapses in light microscopy images.

Specifically for EM images, the main body of related work is concerned with automatic segmentation of neurons for the reconstruction of neural wiring circuits [7, 8, 9]. [10] aims to solve the problem of mitochondria detection and proposes geometric image features well suited to this task. In [11], an automatic method of synapse detection is used in the course of a large-scale connectivity study. It is based on neural network predictions from several labeled images and achieves a sufficiently good error rate. However, as the search is only performed along the cell membranes, it requires a prior segmentation of the entire volume into cells. A very recent plug-in to the ImageJ framework [12] offers interactive annotation and segmentation of neural images, but it treats image stacks as collections of independent images and does not directly exploit the 3D connectivity of the neural data.

# 3 WORKFLOW

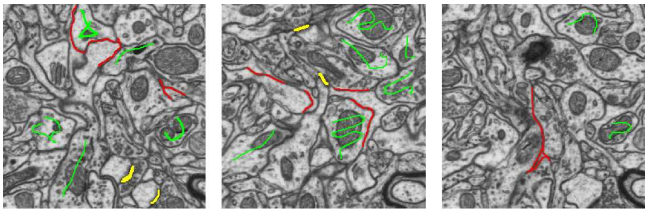

Figure 1: All user labels used during the algorithm evaluation. Synapses are marked as yellow, membranes as red and the rest as green.

## 3.1 Classification

For the first step of the procedure, the user has to label a representative data subvolume. While the labeling can be very sparse, the pixels selected for training have to reflect the underlying properties of the data. For this reason we recommend to label the membranes separately, as they tend to get under-represented when only brush strokes across the intracellular space are used. An example labeling is shown in Fig.1.

Ilastik provides a rich set of features to describe different properties of pixel neighborhoods at various scales. The feature browser allows the user to study the feature response on the data and to select only the features best suited for the task at hand (see Fig.2 for an example). For synapse detection, texture features, such as the eigenvalues of the structure tensor computed at scale of approximate synapse width, seem to be the most discriminative, but not sufficiently so to limit the feature set to these features only. In our experience, the best classification results are achieved when they are computed at several scale values and combined with Gaussian smoothing, magnitude of Gaussian gradient, Laplacian of Gaussian and the difference of Gaussians, but the users are free to further explore the provided feature set, try additional combinations and, using the ilastik plug-in system, to implement features of their own. Based on the user labels and the selected features, the classifier computes the probabilities of class membership for every pixel.

Figure 2: Two features, found to be very informative during training. Left: the raw data. Center: first eigenvalue of the structure tensor. Right: first eigenvalue of the Hessian matrix.

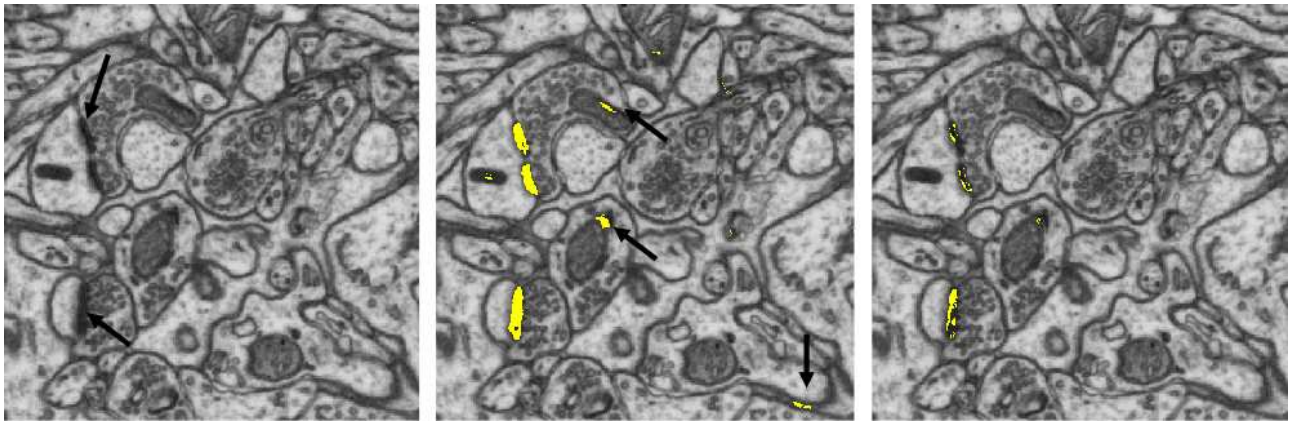

Figure 3: The effect of thresholding. Left: raw data, arrows point to synapses. Center: Predictions thresholded with threshold 0.66, arrows point to false positive detections. Right: Predictions thresholded with 7.0 threshold

### 3.2 Detection and Segmentation

To avoid pixel-wise discontinuities, the class probability maps are smoothed by convolution with a Gaussian. In our experiments we used the standard deviation value of 5 pixels, as it is close to the synapse thickness at 5nm resolution, but the user can easily adjust this value to better suit the data. Afterwards, probability maps of the non-synapse classes are combined into a single background class and thresholding is performed so that only the pixels, for which the ratio between the probability of the synapse class and of the background class is greater than a user-defined threshold, remain in the selection (see Fig.3). Synapse object candidates are then constructed as connected components of the remaining pixels. But, even with a high threshold value, many false positive candidates are still present in the dataset. As further increasing the threshold would lead to the loss of algorithm sensitivity, we employ an additional biologically motivated size-based filter. After computing the size of the synapses, which were labeled in the training dataset, we discard all the candidates more than 10 times smaller than the smallest labeled synapse or more than 10 times larger than the largest synapse. The detection procedure only returns the clusters of highest synapse probability. To correct the estimate of the synapse size, we enlarge these clusters to include all those adjacent pixels for which the probability of the synapse class is higher than that of the background.

## 4 IMPLEMENTATION

The freely available ilastik toolkit provides an interactive and user-friendly interface for multi-class labeling and classification in up to three spatial and one spectral dimension. We have developed an extension of ilastik, which includes interactively adjustable multivariate thresholding, finding of connected components and the possibility to display the selected objects in 3D using the Mayavi2

visualization tool [13].

Once the user is satisfied with the classification results on the training dataset, the rest of the data can be processed offline using the automation capabilities of ilastik and an additional post-processing script. The synapse detection results are saved into an hdf5-based ilastik project file along with the raw data. This file can then be loaded into ilastik for result verification, 3D displaying, additional processing or saving in a different format (see Fig.4).

# 5 EVALUATION

The test dataset consisted of 140 FIB/SEM serial scans of size 2048x1532, taken at 5nm resolution and downsampled to 6.83nm per pixel in x and y, and 9nm resolution in z. These images were produced as a part of a larger stack, and the training subvolume was taken from a different part of the same stack. Before processing, the images were registered using the TrakEM2 plug-in of the Fiji image processing framework [14]. The synapse locations in the images were manually annotated using TrakEM2 in the course of a different on-going study and additionally manually re-verified to obtain an estimate of a human expert error on the test dataset. Only the asymmetric, presumed glutamate type synapses were considered for evaluation.

In ilastik, only the labels, shown on Fig.1 were provided for training, and the features described in Section 3.1 were computed. Segmentation of the resulting probability maps was performed for different values of the probability threshold. The best performance is reached for the threshold values around 7.0: for 118 synapses in the ground truth, 5 are not found by the algorithm and 7 false positives are found at threshold 7.0, 10 false negatives and 4 false positives are returned for threshold 9.0., and 4 false negatives and 18 false positives for threshold 5.5. Our results show that with an appropriate selection of the probability threshold the algorithm can achieve an error rate better than that of a human expert (5 false negatives and 7 false positives for the algorithm vs. 16 false negatives and 16 false positives for the human out of 118 total synapses found). It is also interesting to note, that many of the false positive synapse detections, erroneously produced by the algorithm, can easily be filtered out by a human, as they are usually caused by a mitochondrion with a pronounced membrane located very close to a cell membrane. Since the algorithm predictions are pixel-based, it lacks the larger context which the human uses to discard such cases. We are currently working on an extension of the algorithm which will be able to incorporate more context information.

The advantage of 3D processing can be clearly seen by comparing the number of false negative detections between the algorithm and the human expert. Since the human judgments are based

on the frontal 2D projections of the volume, it is hard for the expert to detect synapses which are parallel to the imaging plane. The automatic procedure considers the complete 3D neighborhood of each pixel and does not suffer from such errors. However, as it relies on the training dataset to be a good representative of the test dataset, it can erroneously filter out the synapses which are too different from the ones labeled by the user.

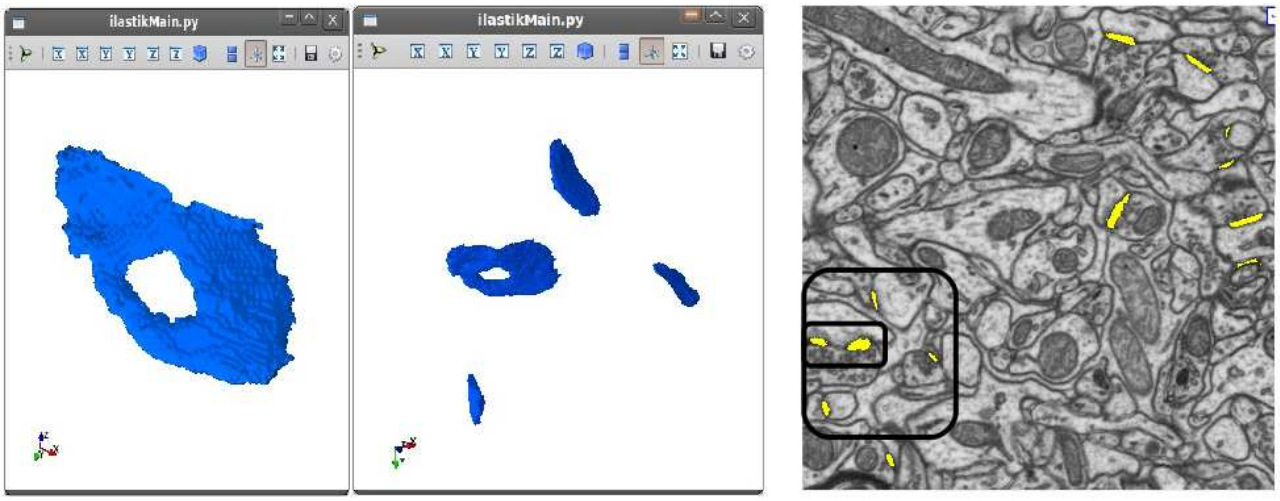


Figure 4: Synapses, detected by the algorithm. Left: 3D view of the perforated synapse, marked by the small rectangle in the right figure, where it appears as two blobs, because the imaging plane is cutting through the hole. Center: 3D view of the four synapses, marked by the larger rectangle in the right figure. Right: synapse detection results

# 6 CONCLUSIONS

We have presented a method for automatic detection of asymmetric synapses and segmentation of their junctional complexes in nearly isotropic 3D EM data. The proposed approach is based on interactive thresholding of class probability maps, produced by a Random Forest classifier from the generic geometric features of 3D pixel neighborhoods. The interactive processing of a small subvolume of data is followed by a fully automatic classification and segmentation of the rest of the volume. The error rate of the method is equivalent to that of a human expert for false positive detections and is considerably lower for false negative detections.

By making this method freely available we hope to significantly simplify and speed up the analysis of the many neuroscience experiments, that entail the counting of synapses, the synapse density estimation or the estimation of synapse-to-neuron ratio.

# References


[1] Graham Knott, Herschel Marchman, David Wall, and Ben Lich, "Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling," *J. Neurosci.*, vol. 28, no. 12, pp. 2959–2964, Mar. 2008.

[2] Angel Merchan-Perez, Jose-Rodrigo Rodriguez, Lidia Alonso-Nanclares, Andreas Schertel, and Javier Defelipe, "Counting synapses using FIB/SEM microscopy: A true revolution for ultrastructural volume reconstruction," *Frontiers in Neuroanatomy*, vol. 3, pp. 18, 2009.

[3] Christoph Sommer, Christoph N. Straehle, Ullrich Koethe, and Fred A. Hamprecht, "ilastik," October 2010, http://www.ilastik.org for the general tool, http://www.ilastik.org/synapse-detection for this contribution.

[4] Leo Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[5] Julia Herold, Walter Schubert, and Tim W Nattkemper, "Automated detection and quantification of fluorescently labeled synapses in murine brain tissue sections for high throughput applications," *Journal of Biotechnology*, vol. 149, no. 4, pp. 299–309, Sept. 2010.

[6] Yuriy Mishchenko, "On optical detection of densely labeled synapses in neuropil and mapping connectivity with combinatorially multiplexed fluorescent synaptic markers," *PloS One*, vol. 5, no. 1, pp. e8853, 2010.

[7] Dmitri B Chklovskii, Shiv Vitaladevuni, and Louis K Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Current Opinion in Neurobiology*, vol. 20, no. 5, pp. 667–675, Oct. 2010.

[8] James R Anderson, Bryan W Jones, Jia-Hui Yang, Marguerite V Shaw, Carl B Watt, Pavel Koshevoy, Joel Spaltenstein, Elizabeth Jurrus, Kannan UV, Ross T Whitaker, David Mastronarde, Tolga Tasdizen, and Robert E Marc, "A computational framework for ultrastructural mapping of neural circuitry," *PLoS Biology*, vol. 7, no. 3, Mar. 2009.

[9] Bjoern Andres, Ullrich Koethe, Moritz Helmstaedter, Winfried Denk, and Fred A. Hamprecht, "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification," in *Pattern Recognition*, Gerhard Rigoll, Ed. 2008, vol. 5096 of *LNCS*, p. 142152, Springer.

[10] A. Lucchi, K.Smith, R. Achanta, V. Lepetit, and P. Fua, "A fully automated approach to segmentation of irregularly shaped cellular structures in EM images," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, Beijing, China, 2010.

[11] Yuriy Mishchenko, Tao Hu, Josef Spacek, John Mendenhall, Kristen M Harris, and Dmitri B Chklovskii, "Ultrastructural analysis of hippocampal neuropil from the connectomics perspective," *Neuron*, vol. 67, no. 6, pp. 1009–1020, Sept. 2010.

[12] Verena Kaynig, Ignacio Arganda-Carreras, and Albert Cardona, "Trainable segmentation plugin," October 2010, http://pacific.mpi-cbg.de/wiki/index.php/Trainable_Segmentation_Plugin.

[13] Prabhu Ramachandran and Gal Varoquaux, "Mayavi: Making 3D data visualization reusable," in *Proceedings of the 7th Python in Science Conference*, Gal Varoquaux, Travis Vaught, and Jarrod Millman, Eds., Pasadena, CA USA, 2008, pp. 51 – 56.

[14] Albert Cardona, "TrakEm2: an ImageJ-based program for morphological data mining and 3D modeling," in *Proceedings of the 1st ImageJ User and Developer Conference*, Luxembourg, 2006, pp. 18–19.