Neuron Segmentation With High-Level Biological Priors

N. E. Krasowski, T. Beier, G. W. Knott, U. Köthe, F. A. Hamprecht, and A. Kreshuk

***Abstract*— We present a novel approach to the problem of neuron segmentation in image volumes acquired by an electron microscopy. Existing methods, such as agglomerative or correlation clustering, rely solely on boundary evidence and have problems where such an evidence is lacking (e.g., incomplete staining) or ambiguous (e.g., co-located cell and mitochondria membranes). We investigate if these difficulties can be overcome by means of sparse region appearance cues that differentiate between pre- and postsynaptic neuron segments in mammalian neural tissue. We combine these cues with the traditional boundary evidence in the asymmetric multiway cut (AMWC) model, which simultaneously solves the partitioning and the semantic region labeling problems. We show that AMWC problems over superpixel graphs can be solved to global optimality with a cutting plane approach, and that the introduction of semantic class priors leads to significantly better segmentations.**

***Index Terms*— Segmentation, electron microscopy, probabilistic graphical model, automated tracing, connectomics.**

I. Introduction

CONNECTOMICS is an emerging domain of neuroscience, which seeks to establish the link between the physical configuration (connectivity) and the function of neural circuits. At nano-scale, the connectivity of individual neurons is reconstructed from high resolution Electron Microscope (EM) images of neural tissue. Analysis of such images is an extremely laborious, but unavoidable task: no method other than EM allows to resolve all connections at the necessary level of detail. In recent years enormous progress has been made on the instrumentation side, where entire compartments of the fruit fly brain and comparably large sections of mammalian cortex can now be imaged isotropically [1]–[3]. Automation of the image analysis has, however, lagged behind: large teams of expert annotators still have to invest months of effort into the manual reconstruction process [4]–[7].

The main difficulty for the automated segmentation lies in the fact that heavy metal staining, used for EM imaging, does not make the individual neural cells locally distinguishable from each other by color or texture. The algorithms therefore resort to separating the neurons by detecting the cell membranes. This approach presupposes an extremely high accuracy in the detection of membranes since a tiny local mistake in their segmentation can cut a neuron in two or merge two neurons together – a global error, which can substantially influence the reconstructed neural circuit.

Some of these errors can be pinpointed by analyzing the resulting circuit, where, for example, a segment which consists of two falsely merged neurons will exhibit contradictory biological properties of its constituents. However,

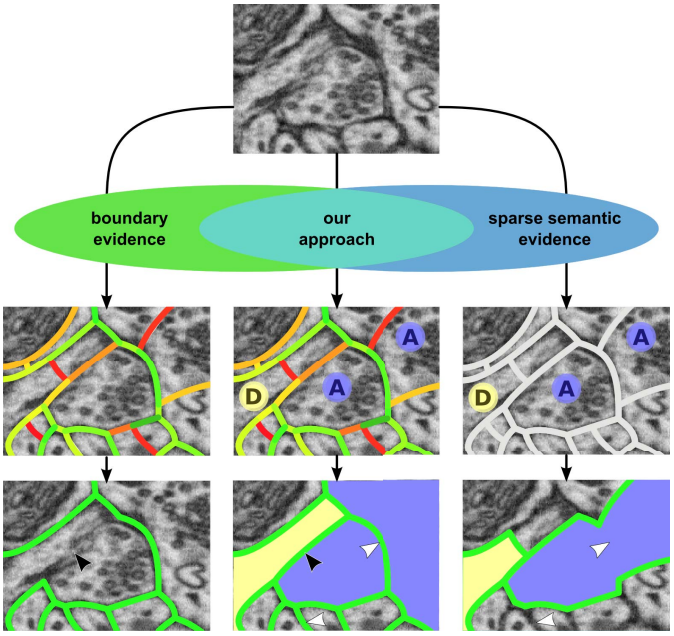


Fig. 1. Top: a tiny extract from a single image of the raw data. **Left** column: segmentation based purely on boundary evidence. The boundary evidence is shown by the color of the edges in the second image, with green edges the most likely and red edges the least likely to be "on" in the final segmentation (bottom image). Based on this evidence, the segmentation algorithm cannot reconstruct the edge marked by the black arrowhead. **Right** column: sparse semantic evidence for axon (A) and dendrite (D) classes. This evidence allows to reconstruct the edge missed by the boundary-only approach, since superpixels with different semantic priors have to be separated in the final segmentation (bottom image). However, sparse semantic evidence is not sufficient to separate superpixels with priors for the same class or with no prior information at all (white arrowheads). **Center** column: the proposed approach combines both sources of evidence and thus attains a more plausible segmentation.

Manuscript received March 27, 2017; revised May 29, 2017; accepted May 29, 2017. Date of publication June 6, 2017; date of current version April 2, 2018. *(Corresponding author: A. Kreshuk.)*

N. E. Krasowski, T. Beier, U. Köthe, F. A. Hamprecht, and A. Kreshuk are with HCI/IWR, University of Heidelberg, 69120 Heidelberg, Germany

G. W. Knott is with the Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2017.2712360

to the best of our knowledge, none of the existing approaches for automated segmentation can incorporate non-local information of this kind directly into the segmentation procedure. Such a combination of high-level biological priors with local membrane evidence is the main target of our contribution.

In more detail, the priors we take into account reflect the probability of a supervoxel to belong to an axon or a dendrite (distinguishable due to the limited field of view). While for most supervoxels in the image volume these priors are uninformative, some contain axon/dendrite indicators, which we can find automatically. These indicators include, for example, vesicle clouds for axons. As suggested by [8], Asymmetric Multiway Cut (AMWC) – a recently proposed algorithm for simultaneous graph partitioning and semantic labeling [9] – can take such information into account and ensure that no supervoxel with dendrite evidence is found in the same reconstructed neuron as a supervoxel that must, according to its appearance, belong to an axon. Enforcing a semantic decision in regions with no priors leads to propagation of the semantic information through the image volume. In case the image volume includes neuron somas, they can be introduced as an additional class, since AMWC can handle multiple semantic classes. AMWC uses two complementary sources of features: those describing the affinity (positive as well as negative) of touching segments, as well as those that capture distinct segment characteristics pointing to its semantic class.

The remainder of this article is organized as follows: section II gives an overview of existing approaches for the automated segmentation of neurons. In the following section III-A, the core algorithm of this work – the Asymmetric Multiway Cut (AMWC) – is explained in detail. Section III-B shows the variant of AMWC best suited for the inclusion of sparse biological priors and specifies our complete pipeline from the raw data to the final segmentation. Section IV contains the experimental results as well as a comparison of performance with established algorithms. We compare explicitly against the learned agglomeration approach of [10] and multicut-based segmentation of [11] on an isotropic dataset produced by a FIB/SEM microscope [1]. Additionally to the experiments introduced by [8], we extend the algorithm to anisotropic data and compare against the anisotropic modification of [11] from [12] on an ssTEM dataset made available by [13]. Finally, section V studies the distribution of sparse priors in the data and the effect of superixel projection on prior sparsity, and concludes by a discussion of all findings and their implications.

II. Related Work

Segmentation of neurons has proven to be the most challenging part of the EM image analysis automation and over the years multiple groups have proposed different approaches to tackle this problem. All of them start from pixelwise membrane segmentation – an all-important first step, which by itself is not sufficient at the current resolution and data quality [14]–[18]. The follow-up steps depend on the imaging method: for data with anisotropic resolution, 2D segmentation combined with linking across the stack images is the preferred approach [19]–[22]. In [23] and [24] the information of neighboring slices is used in the process of the 2D segmentation. Both publications show the significant benefit provided by the inclusion of the third dimension. ‘

For data with isotropic resolution, this advantage is inherently present and most segmentation algorithms operate directly in 3D. The stepwise segmentation approach, first proposed by [25], is currently used by many other 3D segmentation pipelines. The initial supervoxel over-segmentation introduces structure and drastically reduces the problem size, thus allowing more advanced algorithms to be used. In [26], a reinforcement learning approach is used to merge the supervoxels. Similarly, [10] uses reinforcement learning within a learning-based iterative hierarchical clustering algorithm. While very fast, this greedy approach can make incorrigible wrong merge decisions. In [27], the merging decisions are delayed by several steps, until some of their consequences are visible. Reference [28] proposed to greedily solve the segmentation problem within the framework of a merge-tree, while very recently the merge-tree approach has been further improved to the globally optimal level [29].

The method of [29] is still, however, limited to the segmentations contained in the merge-tree. In [11], an overall globally optimal segmentation – for a given supervoxel oversegmentation and supervoxel boundary evidence – is found by introducing topological constraints on the supervoxel adjacency graph. Our work builds on this set of constraints and on the Multicut algorithm [30] in general. But, unlike [11] and all other methods listed above, we are not basing the merge/split decisions purely on the local boundary evidence. The main novelty of our contribution is a principled way to consider non-local prior information.

Computationally, we rely on the Asymmetric Multiway Cut algorithm, recently introduced by [9]. This algorithm extends the Multicut of [11] by a simultaneous semantic node labeling.

Our work needs synapse detections, which provide important clues about the biological priors we want to include. References [31]–[33] all present reliable approaches to perform this task.

III. Methods

The key part of our pipeline is the Asymmetric Multiway Cut (AMWC) segmentation algorithm. AMWC was recently presented in a conference submission [9]. Its application to the neuron segmentation problem with semantic priors was introduced in another conference submission [8], but a detailed examination of the algorithm is still lacking. We therefore offer an extensive explanation of the AMWC algorithm in the following section III-A. In III-B we show how we set the stage for AMWC to tackle the EM neuron segmentation problem and how exactly we include the biological prior information to improve the quality of the segmentation.

A. Asymmetric Cuts

The Asymmetric Cuts or Asymmetric Multiway Cut algorithm is a generic algorithm for joint *semantic labeling* and

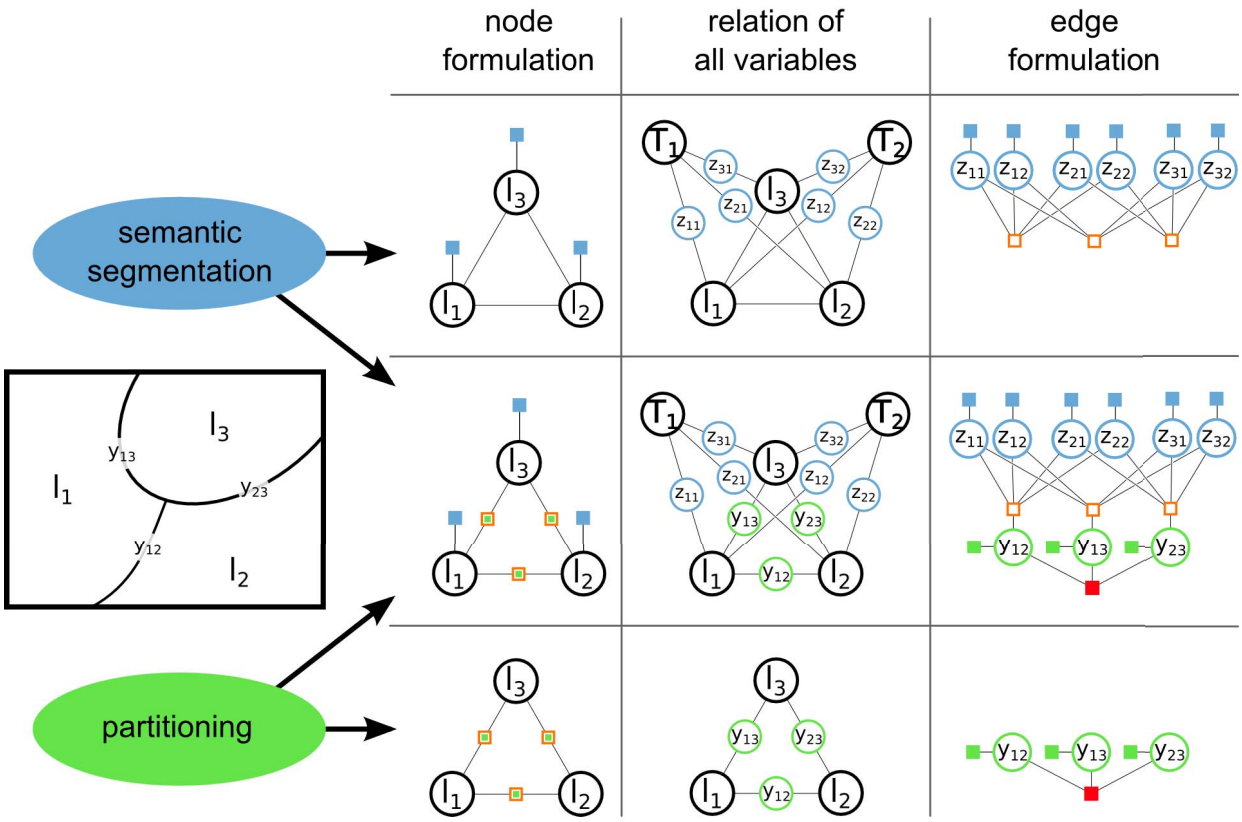


Fig. 2. Node- and edge-based formulations for the semantic segmentation problem (top row), partitioning problem (bottom row) and the AMWC problem, which combines both. Left: oversegmentation of an image into segments l_1, l_2 and l_3, separated by edges y_{12}, y_{13} and y_{23}. The green pairwise factors correspond to pairwise affinity in the node representation. In the edge representation they become unary factors. The blue semantic factors remain unary across representations. The consistency of semantic segmentation and partitioning is ensured via orange factors representing constraints. In the edge formulation the consistency of the partitioning expressed by y_{ij} is ensured via a global factor (red) encoding the Multicut constraints.

partitioning of a graph. In image analysis the graph is often given by an oversegmentation of an image. In this case, the nodes correspond to the superpixels of the oversegmentation. The partitioning is performed according to pairwise affinities of nodes. Simultaneously, a semantic labeling of the nodes based on local information is made consistent with respect to the partitioning.

For the neuron segmentation problem, the motivation for the inclusion of semantic labeling is that in difficult cases, where the edge evidence given by the membrane detection is inconclusive, semantic labels provide additional clues that support merge decisions of neighboring nodes (see Fig. 1).

The work of [9], where the AMWC problem was originally introduced, shows that the partitioning and the labeling can benefit from each other – solving the semantic segmentation problem first and following up with the partitioning problem or vice versa is not equivalent to the joint solution.

The model can be explained with respect to the regions in the oversegmentation (node representation) or the edges between them (edge representation). The first view is presented by a pairwise graphical model (an example is given in Fig. 2 (b)). Note, that the consistency between the labeling and the partitioning has to be ensured by the pairwise factors (see Eq. 5). Also note that the pairwise affinities can be both repulsive and attractive. By going to the edge domain, where the random variables are not the original nodes, but the edges between them, we attain a reduced (binary) label-space for all random variables that is more optimization friendly and a more explicit formulation of the constraints.

The following chapter will explain both formulations and their relationship. In principle, AMWC can be applied to fully connected graphs, but for the sake of clarity and with our applications in mind we choose to use a sparsely connected example.

1) Node Representation: The AMWC problem can be formulated as a node labeling problem for a graph $G = G(\mathcal{V}, \mathcal{E})$, where each node i has a partition label l_i^p as well as a semantic label $l_i^s \in T$.

Here T is the set of all semantic classes. We distinguish between two families of semantic classes: T_c and T_n with $T = T_c \cup T_n$ and $T_c \cap T_n = \emptyset$. For the first family T_c we say we allow "cuts within" for $l^s \in T_c$. This means that after joint labeling and partitioning, adjacent segments may both be of class $l^s \in T_c$, but a segment of class $l^s \in T_n$ can only have semantically different neighbors. For illustration, let's consider the problem of segmenting pedestrians in a street view. The "pedestrian" class then allows "cuts within" as different pedestrians should be able to touch without merging. The background does not need to be segmented into separate instances and thus belongs to "no cuts" T_n family. In the node

formulation of AMWC, each supervoxel i is associated with a label

$$l_i = (l_i^p, l_i^s), \quad l_i^p \in [1, \cdots, \|\mathcal{V}\|], \tag{1}$$

$$l_i^s \in [1, \cdots, \|T_n\| + \|T_c\|] \tag{2}$$

where l_i^p is an integer that indicates the affiliation of the node i in the partitioning problem and l_i^s denotes its semantic class which can be one of the $\|T_c\|$ classes with "cuts within" or one of the $\|T_n\|$ classes without. For the partitioning part of the problem, the number of final clusters is not known beforehand. To account for all possible partitions, one needs as many different partition labels as there are nodes: $\|\mathcal{V}\|$.

To enforce the desired behavior, we introduce pairwise factors between all neighboring nodes, which represent the inclination of a pair of neighbor nodes to end up in the same cluster and thus have the same partition label l_p (Fig. 2, bottom row). In addition to this, the pairwise factors enforce consistency between the semantic labels and the partition labels - e.g. if the semantic labels of an adjoining pair of nodes differ, the partition labels should differ as well (Fig. 2, center row). The semantic labels, in their turn, are influenced by unary factors that encode the local affinity of nodes to the semantic classes (see Fig. 2, top row).

Minimizing the following energy function results in the MAP state of the graphical model:

$$\underset{l}{\operatorname{argmin}} \left(w \sum_{i \in \mathcal{V}} E_i(l_i^s) + \sum_{(i,j) \in \mathcal{E}} E_{ij}(l_i, l_j) \right). \tag{3}$$

Note that the unary terms depend only on the semantic labels.

$$E_i(l_i^s) = -\log\left(\frac{p_i^s}{1 - p_i^s}\right) \tag{4}$$

Here p_i^s is the prior probability that node i belongs to the semantic class $s \in T$. With p_{ij} being the probability that node i and node j are in different partitions, the pairwise potentials are constructed in the following way:

$$E_{ij}\left(l_i, l_j\right) = \begin{cases} 0 & \text{if } l_i^p = l_j^p, l_i^s = l_j^s \\ -\log\left(\frac{p_{ij}}{1-p_{ij}}\right) & \text{if } l_i^p \neq l_j^p \\ \infty & \text{if } l_i^p = l_j^p, l_i^s \neq l_j^s \end{cases} \tag{5}$$

Solving the partitioning- and the semantic segmentation problem simultaneously to global optimality is hard. One reason is that the energy function is highly degenerate, as it is invariant under all permutations of the partition labels. Another reason is the non-submodular nature of the pairwise factors. On the one hand they may have positive as well as negative values and on the other hand they implicitly encode consistency constraints between the partition labels and the semantic labels (encoded in Fig. 2 by orange frames). Nevertheless, [9] shows that globally optimal solutions can be found for problems of relevant size by rewriting them as a cut problem in the edge domain.

2) Edge Representation: While the formulation presented in III-A.1 is intuitive and close to the formulation of common probabilistic graphical models, only the equivalent formulation as a cut problem can actually be solved in practice. We will see how it can be written as an integer linear program, to which a whole variety of solvers can be applied. In the general case, the problem is NP-hard. Since we apply the algorithm not on arbitrary graphs, but on graphs that originate from volumetric images, the graph architecture is highly structured. In [30] it is shown that for those cases only a small subset of the cycle constraints is really needed to find the globally optimal solution. A cutting-planes approach, where violated constraints are added iteratively, therefore stands to reason.

The cut formulation in the edge domain relies on binary random variables, which can be associated with edges in the adjacency graph, and binary random variables, which indicate the affiliation of the nodes to the so called terminal nodes, representing semantic classes. The price one has to pay for this clear representation is that not every possible edge labeling corresponds to a consistent node partitioning. Besides, the uniqueness of the affiliations to terminal nodes is not given naturally. These shortcomings must be addressed by the introduction of explicit constraints. One big advantage of this representation is that there is no intrinsic degeneracy of the objective.

More formally, the graph G from III-A.1 is now extended by terminal nodes T and terminal edges $\mathcal{E}_T$. Together with the internal nodes $\mathcal{V}$ and the internal edges $\mathcal{E}$ they describe all nodes $\mathcal{V}'$ and all edges $\mathcal{E}'$ in the augmented graph $G'(\mathcal{V}', \mathcal{E}')$. Fig. 2 (center column) shows the terminal nodes and edges introduced for the graph in Fig. 2 (left column). All terminal nodes are connected to all internal nodes in $\mathcal{V}$ via terminal edges $\mathcal{E}_T$.

$$\mathcal{V}' = \mathcal{V} \cup T, \mathcal{E}' = \mathcal{E} \cup \mathcal{E}_T, \text{where } \mathcal{E}_T = (i,t) | i \in \mathcal{V}, t \in T$$

We introduce binary variables y_{ij} and z_{it} for all edges in $\mathcal{E}$ and $\mathcal{E}_T$.

The optimization problem then reads:

$$\underset{\mathbf{y}' \in \{0,1\}^{|\mathcal{E}'|}}{\operatorname{argmin}} \left(\kappa \sum_{(i,t) \in \mathcal{E}_T} v_{it}(1 - z_{it}) + \sum_{(i,j) \in \mathcal{E}} w_{ij} y_{ij} \right) \tag{6}$$

$$\text{s.t. } \mathbf{y}' \in \text{AMWC}_{G'}. \tag{7}$$

where $\mathbf{y}' = (\mathbf{y}, \mathbf{z}) \in \mathcal{E}'$ and $\text{AMWC}_{G'}$ is the Asymmetric Multiway Cut polytope. Our convention is that $y_{ij} = 1$ stands for separated nodes i and j (the respective edge is cut) and $y_{ij} = 0$ stands for merged nodes i and j. $z_{it} = 0$ means that the internal node i is associated with the semantic class represented by T_t. $AMWC_{G'}$ is defined by the following linear constraints:

$$\sum_{(i,j) \in P} \mathbf{y}_{ij} \geq \mathbf{y}_{uv} \forall\ (u,v) \in \mathcal{E};\ \forall P \in \text{Path}(u,v) \subseteq \mathcal{E} \tag{8}$$

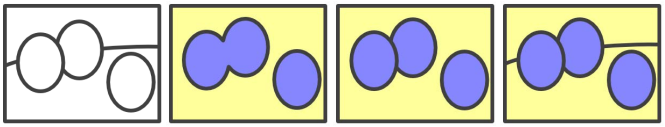
$$\sum_{t \in T} \mathbf{z}_{it} = |T| - 1 \quad \forall i \in \mathcal{V}; \tag{9}$$

$$\mathbf{y}_{uv} \geq \mathbf{z}_{vt} - \mathbf{z}_{ut} \quad \forall\ (u,v) \in \mathcal{E}; t \in T \tag{10}$$

$$\mathbf{y}_{uv} \geq \mathbf{z}_{ut} - \mathbf{z}_{vt} \quad \forall\ (u,v) \in \mathcal{E}; t \in T \tag{11}$$

$$\mathbf{y}_{uv} \leq \mathbf{z}_{ut} + \mathbf{z}_{vt} \quad \forall\ (u,v) \in \mathcal{E}; t \in T_n \tag{12}$$

The cycle inequalities (8), also used in [30], ensure the consistency of the partitioning. Eq. (9) ensures the unique affiliation of each node with a terminal node. The consistency



(a) MC partitioning | (b) GC/MWC sem. segmentation | (c) AMWC sem. seg. + partitioning | (d) SC sem. seg. + partitioning

Fig. 3. Possible results of the variants of the **A**symmetric **M**ulti**w**ay **C**ut algorithm. **(a)** partitioning Multicut algorithm. The image is divided into several segments with no semantic labels. **(b)** Pott's model, solved by **G**raphcut or **M**ulti**w**ay **C**ut. Blue and yellow indicate the semantic affiliation of the segments. It is not possible to have a subdivision of connected components of one semantic class. Unlike Graphcut, the Multiway Cut is able to handle multiple semantic classes and non-submodular potentials. **(c)** AMWC with "cuts within" only allowed in the foreground (blue) class. **(d)** **S**ymmetric **C**uts, with "cuts within" in all classes, used for neural segmentation later on.

between semantic labeling and partition labeling is ensured by eqs. (10) and (11). They state that whenever adjacent nodes are assigned to two different terminal nodes, there is an internal edge separating them and therefore they must belong to different partition elements. The constraints in eq. (12) ensure that "cuts within" are not allowed for all semantic classes $t \in T_n$.

The weights v_{it} in Eq. (6) are equivalent to the unary energies in Eq. (5). Also the weights w_{ij}, which determine attraction or repulsion between the internal nodes i and j, are related to the energies in the node representation:

$$v_{it} = E_i(t);\ w_{ij} = -\log\left(\frac{p_{ij}}{1-p_{ij}}\right). \tag{13}$$

While the weights w_{ij} are purely encoding pairwise affinities, the pairwise energies E_{ij} in the node representation also have to ensure the consistency between partitioning and labeling.

3) Relation to Other Models: The AMWC framework of joint labeling and partitioning generalizes some well known algorithms.

For $\|T_c\| = 1$ and $\|T_n\| = 0$ we fall back to the case of the **Multicut** for partitioning/correlation clustering problems (see [30]). Here we omit all unary potentials/all semantic labels in the node formulation or equivalently we omit the constraints from eqs. (10) to (12) in the edge formulation. The omission of all terminal edges $\mathcal{E}_T$ considerably reduces the size of the problem. For pure partitioning/correlation clustering problems this model provides a globally optimal solution. Fig. 3 (a) shows an example Multicut partitioning.

The **Multiway Cut** (as introduced in [34]) is the special case of $\|T_c\| = 0$. All semantic classes are forbidden to have "cuts within". An exemplary problem from image analysis is an image labeling problem, where a single object is segmented from the background or where no instances of semantic classes are touching. While Fig. 3 (c) and (d) could not be produced by MWC, (b) is a possible output.

If $\|T_c\| = 0$ and $\|T_n\| = 2$ and in addition there are no repulsive pairwise potentials, this Pott's model can be solved by the popular **Graphcut** algorithm. Since in Fig 3 (b) only two semantic classes are used and no weights are specified, it could as well be produced by Graphcut.

If $\|T_c\| \neq 0$ and $\|T_n\| \neq 0$, only some semantic classes are allowed to have "cuts within", while the remaining semantic classes can not have inner boundaries. This case is studied in [9] and illustrated in Fig. 3 (c) where the (blue) foreground shows cuts within, the (yellow) background does not.

The special case of $\|T_n\| = 0$ – the antagonist of the Multiway Cut – has not been discussed in detail before. Here all semantic classes are allowed to have "cuts within", so we will refer to it as "Symmetric Cuts". This variant is needed for segmenting images with no specific background class, such as neural EM volumes. A possible output of Symmetric Cuts is shown in Fig. 3 (d).

4) Semantic Class Affinities: In the classic setup of semantic segmentation, establishing the semantic class of every pixel is the end goal of the algorithm. For AMWC this use case has been explored in [9], with simultaneous neuron and mitochondria segmentation serving as a motivating example. The unaries v_{it} of Eq. 6 can then simply be derived from an output of a semantic classifier.

Complementary to the work of [9] and more suiting to our immediate use case of neuron segmentation, we would like to explore the use of semantic affiliation of segments as a means to an end, as additional information for solving the partitioning problem. This information can be very sparse, with most nodes exhibiting no affinity to either semantic class. Conceptually, this problem is similar to seeded segmentation, where the information is propagated from the seeds to the segment boundaries. Our prior information is, however, much weaker: priors for the same semantic class are present in multiple objects, which need to be separated in the final segmentation. Besides, many objects do not include any semantic prior information at all. Another similar setup is a partitioning Multicut with additional long range repulsive edges between some non-neighbor segments [35]. This model is equivalent to AMWC, if not more than one segment with prior information is present for each semantic class, or if the costs of switching the semantic label of a node are infinite (hard constraints). Such constraints are easy to introduce if, for example, an expert user specifies that two nodes definitely belong to different objects in the final segmentation (in 3D, this task is very non-trivial). Our goal, however, is to derive the sparse semantic "seeds" and the corresponding costs of switching the node labels directly from the data, limiting the necessary user involvement to training a semantic classifier on a subset of data.

The predictions of this classifier do not need to represent the semantic class affinity directly. For example, in the neural segmentation use case, a classifier for vesicle clouds would indirectly predict that superpixels with a high response likely belong to an axon. The superpixels with a low response remain unassigned – as many axon superpixels are not covered by vesicle clouds, negative predictions of the classifier add no class affinity information. In the following we will refer to such predictions as "proxy probabilities". Consequently, we define p_i^s of [4] as

$$p_i^s = \max\left(0.5, \tilde{p}_i^s\right), \tag{14}$$

where $\tilde{p}_i^s$ is the proxy probability probability. With the unaries defined this way, the sparse semantic class information can

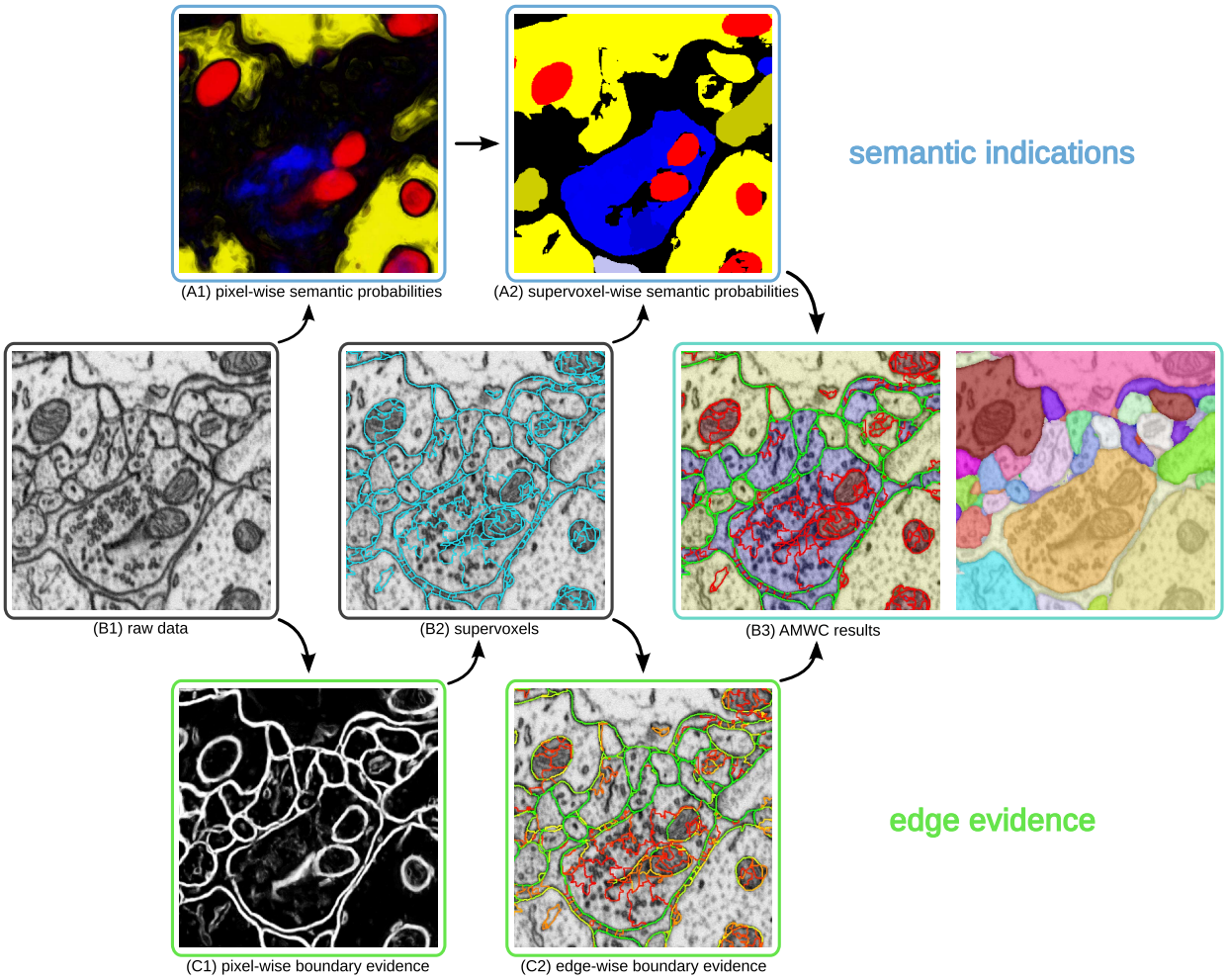


Fig. 4. The flow diagram for the presented segmentation pipeline, with consecutive steps going from left to right. The starting point (B1) in the central row to the left is the raw EM volume. The intermediate image volumes (A1), (C1) and (C2) rely on predictions of a Random Forest classifier: (A1) and (C1) show the voxelwise semantic and membrane probability. (C2) shows the pairwise split probability projected on the touching faces of the g-supervoxel oversegmentation (see III-B.2 for details). The edge color represents how likely the edge is to remain in the final segmentation, with green the most likely and red the most unlikely. (B2) shows the GALA supervoxels as described in III-B.1. The big GALA supervoxels allow for expressive features and therefore good classification of the faces. The mapping from the pixelwise semantic probabilities to the supervoxel level is described in III-B.3. The core AMWC algorithm is described in full detail in III-A. Its result, a jointly optimized semantic labeling and a partitioning is represented in (B3). For reference, (B3) also shows the final segmentation, where each object is assigned a different color.

propagate through the nodes with no class affinity (as no cost is payed to switch their semantic label to either class) and influence the cut decisions for an uncertain edge far away from the original "seeds". This property is especially important in our 3D application.

B. AMWC Workflow for Neuron Segmentation

This section describes the setup of the AMWC algorithm for the task of neural segmentation: the creation of supervoxels for the graph, the estimation of supervoxel edge affinities, the definition of semantic classes and their proxies and the necessary post-processing.

1) Supervoxel Generation: The oversegmentation of the image volume into supervoxels is based on a pixelwise membrane probability map. This probability map is de-noised by the non-local means algorithm [36]. Local minima on the smoothed map serve as seeds for a watershed algorithm. The supervoxels, resulting from the watershed algorithm, are termed w-supervoxels. For our test block, we constructed 111 164 w-superpixels in total.

The w-supervoxels are coarse enough to compute the first set of pairwise features and to train a Random Forest classifier for the respective affinities. The GALA hierarchical clustering algorithm [10] does this in a convenient fashion and merges the supervoxel pairs iteratively, while updating the probabilities for the newly emerged pairs. Since this greedy strategy cannot provide any guarantees that the solution will be close to the globally optimal one, we do not want to use it to attain the final segmentation (cluster until there are no pairs of neighboring supervoxels with a positive affinity left). Instead, we only apply it for easy merges, where the algorithm can utilize its advantage of adapting edge-weights, and stop at a conservative threshold to avoid undersegmentation. After running GALA

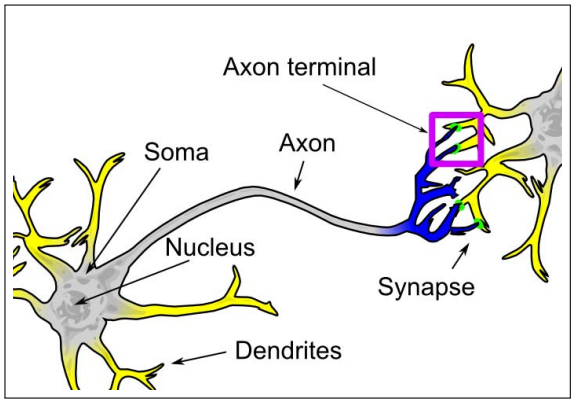


Fig. 5. Two mammalian neurons and their synaptic connections (green). The pink rectangle sketches qualitatively our field of view. The yellow/blue parts are the two semantic classes used to distinguish neurons. (Figure based on [37])

on our test block, there are 11 672 supervoxels left. In the following, we will refer to them as g-supervoxels.

The early stopping at a conservative threshold has multiple advantages: if we want to follow up with a globally optimal procedure such as AMWC, we profit from a smaller adjacency graph in terms of runtime. Also, features accumulated on supervoxel surfaces become more expressive the larger the supervoxels get.

2) Probabilities: Both the GALA algorithm and the subsequent AMWC base their merge/split decisions on the pairwise affinities of supervoxels. We compute these by a Random Forest classifier, which determines whether a touching face of two supervoxels is representing a membrane or not. The classification is based on statistics of pixel feature values in the direct proximity of the faces and on sorted statistics of the whole volumes of the two supervoxel neighbors.

The pixelwise features are composed of the following filter responses: Gaussian smoothing, Gaussian gradient magnitude, the Hessian of Gaussian eigenvalues and Laplacian of Gaussian. They are applied on scales 1.6, 4.2 and 8.3. The following statistics are computed along the faces (all voxels in neuron i/j, which have a voxel in neuron j/i in their 6-neighborhood): mean, variance, quantiles (25%, 75%), kurtosis median, skewness and size of the faces. Besides those, histogram of grayvalues, voxel count, kurtosis, maximum, minimum, quantiles (10%, 25%, 50%, 75%, 90%), radii of supervoxel, skewness, sum and variance are computed on each of the volumes of two adjacent supervoxels. All feature pairs can be translated to features of the touching faces by applying: min(., .), max(., .) and ‖ difference (., .) ‖. This avoids an ambiguity in ordering of the plain feature pairs of the two touching supervoxels.

3) Semantic Classes: Axon, Dendrite and Mitochondria: In mammalian cortex, a clear separation exists between the axon and the dendrite of a neuron (see Fig. 5). Therefore, and since most current high-resolution image volumes do not encompass the soma, axonic and dendritic neurites inside the volume must be disjoint.[1] Consequently, if for every supervoxel in the volume we knew whether it belongs to an axon or a dendrite, edges between axons and dendrites would be preserved regardless of the strength of the edge indicator. For most supervoxels this information is, however, not available and their axonic/dendritic affiliation is not clear even for expert neuroscientists. Nevertheless, sparse affiliation indicators do exist and the AMWC framework allows us to benefit even from a small number of supervoxels, where these indicators can be found. The particular indicators we use are listed below:

- **Synapses**. Most chemical synapses in the mammalian cortex are between an axon and a dendrite or dendritic spine. Spines are small protrusions from the dendrite and receive the majority of excitatory contacts. Given that synapses in FIB/SEM data can be found automatically along with their direction ([31], [32]), we can assign supervoxels, touching the synapse on the presynaptic side, to axons and on the post-synaptic side to dendrites.
- **Vesicle clouds**. These are mostly found in axons, as they contain neurotransmitters for synaptic transmission. We train a pixel classifier to detect vesicles and discard detections that are not part of a cloud.
- **Dendritic cytoplasm**. Dendritic shafts can become significantly thicker than axons. We train another pixel classifier for the cytoplasm of the dendritic shafts and discard all small connected components of the resulting segmentation.

The mapping of proxy probabilities from pixels to g-supervoxels is performed in two steps. In the first stage we take the mean pixel value for each w-supervoxel. In the second step we take the maximum of all the values from the w-supervoxels that compose the respective g-supervoxel. The mean operation results in lower vulnerability to noise and the maximum operation expands the volume for which we have the class information available.

Along with axonic and dendritic priors described above, we found it beneficial to separate the mitochondria in a semantic class of their own. False merging errors often occur at locations, where mitochondria are situated very close to the neuron membrane (examples for such problematic situations can be found in Fig. 9, second and third row). Introduction of the "mitochondrion" semantic class helps the algorithm resolve the cases, where a mitochondrion membrane is not distinguishable from the cell membrane. The prior for mitochondria is also derived from the pixelwise mitochondria probability map, but, since mitochondria are directly observable, a simple averaging over the g-supervoxels is sufficient. Mitochondria are merged back with their respective neurons by a post-processing step, described in the next section.

Obviously, in a volume of neural tissue an axon can touch another axon and a dendrite can touch another dendrite. Therefore, we choose the Symmetric Cuts AMWC variant, where "cuts within" are allowed for all semantic classes.

4) Optimization: We use the open source library OpenGM [38] together with the commercial optimization software package CPLEX from IBM for the actual inference. OpenGM allows for the simple formulation of the problem as a graphical model and performs the transformation into

[1]Note that this does not restrict this model from scaling up to bigger volumes. If somas are present in the data, they just have to be included in the modeling as a separate semantic class

an explicit integer linear program formulation automatically. Technical details of the optimization procedure can be found in [39].

5) Post-Processing: In most cases, we can simply merge the mitochondria segments with the neuron segments, with which they have the longest common boundary. However, occasionally the mitochondria are (wrongly) connected over the neuron boundary and we need to apply the merging stepwise. The strategy is as follows: first we find the "connector" supervoxels, which keep the two mitochondria joined. Then we remove the connectors and merge the remaining pieces by the simple procedure above. The connectors themselves are merged in the last step. The connectors are found by running a shortest path algorithm on the adjacency graph of the g-supervoxels for all pairs of nodes for each connected component of mitochondria class. Each node in this adjacency graph counts how often it was used for crossing. The local maxima of this estimate are taken as connectors.

If more than one connector is found in a mitochondria segment, the procedure is iterated until all connectors are merged. Using this approach, all mitochondria of our test block have been correctly merged with their containing neuron.

IV. Experiments

A. Training- and Test-Set Specifications and Setup

The performance of the algorithm has been evaluated both on isotropic (FIB/SEM) and anisotropic (ssTEM) data.

1) Isotropic Data: we used a FIB/SEM dataset from adult mouse somatosensory cortex with approximately isotropic resolution of $5 \times 5 \times 6$ nm. The dense volumetric groundtruth segmentation of the dataset was created by an expert neuroscientist using the carving algorithm from ilastik [40], [41] and the Connectome Annotation Tool Mojo 2.0 [22].

The first block of $900 \times 900 \times 200$ px is used to train the random forests. The pixelwise classifiers for the semantic classes and for the membrane/non-membrane predictions are trained on sparse user labels in ilastik [41]. The random forests that predict the merge probability of neighboring supervoxels are trained using a projection of the dense, pixelwise groundtruth on the respective supervoxels. The projection is done in a way that the projected groundtruth respects the supervoxel boundaries, by an argmax operation on all voxelwise neuron ids within the voxel composing a supervoxel. All the tests have been performed on a dataset of $700 \times 700 \times 700$ px, adjacent to the training set, but not overlapping with it.

2) Anisotropic Data: we used the apical dendrite dataset from [13]. This dataset comes with extensive groundtruth annotations which include not only the segmentation, but also classification of objects into axons, dendrites and astrocytes. Additionally, we densely annotated the mitochondria in the substack we used for experiments. We used the anisotropic variant of the multicut pipeline as described in [12] and compared the performance of the multicut algorithm with and without high-level priors. From the apical dendrite dataset we cut out a subvolume of $900 \times 860 \times 100$ pixels (so that, accounting for the anisotropy in resolution, it is roughly the same size as the isotropic volume above). The in-plane resolution of the images is approximately 2 nm, with section thickness of 49 nm. We cut this dataset in two and first do all the training on slices 0-50 and test on slices 50-100. Then we switch, train on slices 50-100 and test on 0-50. Besides, to estimate the variability of the priors' effect, we have divided the dataset into slices of 5, ran the segmentation algorithms and computed the mean and the standard deviation of the performance improvement. The axon, dendrite and mitochondria are learned the same way as in the isotropic case. Following [12], we compute the supervoxels in each slice separately using agglomerative clustering of wateshed results, where the watershed in each slice is seeded by the minima of the distance transform on the membrane probability map. The graphical model is then constructed for all the slices at the same time, with additional edges introduced between superpixels which touch across the z-axis. Besides comparison to the Multicut baseline, we have also run the AMWC algorithm with dense groundtruth semantic priors provided by [13] to estimate the upper limit of the potential performance improvement.

B. Results

Our experiments aim to establish if the globally optimal approach of the Multicut algorithm is beneficial despite the increased CPU-load and if, in their turn, the semantic classes, introduced for the AMWC algorithm, carry enough additional information to improve on the standard partitioning Multicut. For the first point, we compare to the popular GALA hierarchical clustering algorithm. To estimate the upper bound of the potential improvement due to semantic classes, for anisotropic data we also evaluate the performance of the Asymmetric Multiway Cut algorithm with dense semantic priors as provided by the groundtruth in [13]. Additionally, we estimate the variability of the performance improvement by running the algorithm over sub-stacks of 5 slices and averaging the results.

Table I summarizes the results of three experiments on isotropic data: (i) GALA segmentation on w-supervoxels (w-GALA), (ii) MC on g-supervoxels (g-MC), (iii) AMWC on g-supervoxels (g-AMWC). It is infeasible to run MC/AMWC on small w-supervoxels and running GALA on g-supervoxels is equivalent to a single run from w-supervoxels to a higher threshold. To better estimate the effect of AMWC's additional priors, we have also computed the split and merge Variation of Information separately. For g-MC, these stood at 0.57. For g-AMWC, VI split was measured at 0.65 and VI merge at 0.40. These results support our expectation that additional high-level priors help to avoid false merge errors, but, due to inaccuracies in the pixel classification for the priors we use, they increase the number of split errors.

Anisotropic data experiments are summarized in Table II and Fig. 6. Here we compare MC, AMWC with sparse learned semantic priors and AMWC with dense semantic priors from the groundtruth annotations.

V. Discussion

The comparison of GALA with both MC and AMWC in Table I demonstrates the advantage of global optimal

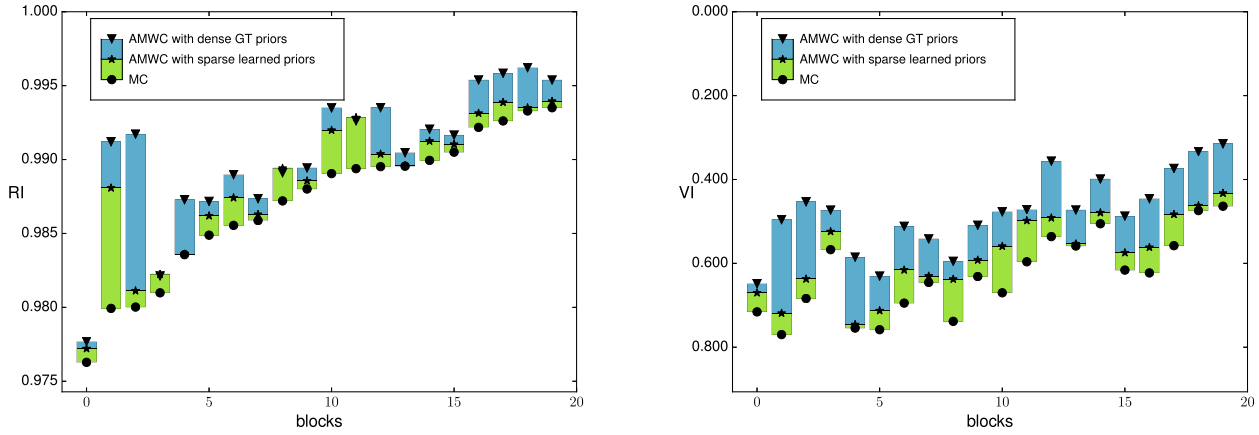


Fig. 6. Rand Index (RI, left) and Variation of Information (VI, right) comparison for the Multicut baseline, for the presented algorithm – AMWC with sparse semantic priors learned from the data, and for the AMWC with dense semantic priors taken from the manual groundtruth annotations. AMWC with dense priors shows the upper limit of performance improvement achievable from axon/dendrite/mitochondria semantic priors. The blocks are ordered by the RI of the Multicut algorithm. Note that the scale of VI is inverted for easier comparison.

TABLE I
ALGORITHM PERFORMANCE FOR ISOTROPIC DATA

	w-GALA	g-MC	g-AMWC	GT
RI	0.9907	0.9956	0.9965	1.0
RI_w	0.9918	0.9967	0.9977	0.9989
RI_g	0.9920	0.9972	0.9981	0.9987
VI	0.6971	0.5745	0.5421	0.0
VI_w	0.4693	0.3494	0.3105	0.2177
VI_g	0.4282	0.2874	0.2490	0.2702

TABLE II
ALGORITHM PERFORMANCE FOR ANISOTROPIC DATA

	MC	AMWC, learned	AMWC, GT priors
Mean RI	0.9871	0.9885	0.9904
RI improvement		0.0015±0.0018	0.0033±0.0030
Mean VI	0.6280	0.5791	0.4787
VI improvement		0.05±0.03	0.15±0.05

merge decisions. In the commonly used quality measures for segmentation, Rand Index and Variation of Information, Multicut clearly outperforms the greedy GALA algorithm on our test data.

As for the additional semantic classes, the comparison between Multicut and AMWC, using the same supervoxels and the same merge probabilities, shows that the semantic information is indeed complementary to our boundary evidence and brings a significant benefit even when the priors are sparse. This effect is demonstrated both on isotropic (Table I) and anisotropic data (Table II). Fig. 9 illustrates this conclusion by three difficult examples in isotropic data, correctly solved by AMWC, but not by the Multicut. Fig. 6 compares the performance of Multicut and AMWC on 20 blocks of anisotropic data and shows that semantic priors improve segmentation quality on all of them. Like the segmentation quality itself, the amount of improvement varies a lot between the blocks. This variability can probably be explained by the non-uniform distribution of sparse priors across blocks as well as by the non-uniform distribution of difficult locations where semantic priors could help.

Since [13] provides dense groundtruth labels for axons, dendrites and astrocites, we have used these annotations to estimate the upper limit on the improvement brought by the semantic priors. It is interesting to note that for some blocks sparse learned priors already give the maximal improvement of the Rand Index (although not of the Variation of Informaion) and for all blocks significant advantage can be gained by using semantic priors. While it is not possible to provide such dense annotations for existing datasets unless a segmentation is performed in the first place, we hope our results will provide additional motivation for research into new staining methods and correlative setups [4], [42], which will allow for selective staining of different cell types visible either in EM directly or in co-registered LM image volumes.

RI_w, VI_w, RI_g and VI_g rows of Table I compare all three algorithms with the groundtruth projection on w- and g-supervoxels. Since both the segmentation and the groundtruth respect the boundaries given by the supervoxels, this comparison minimizes the effect of insignificant boundary shifts and shows more of the serious, topology-changing errors. For the AMWC algorithm, such errors are roughly of the same magnitude as inaccuracies of the boundaries.

Now that the benefit of the introduction of semantic class priors is clear, let us take a more detailed look at their distribution in the data. As described in III-B.3, for "axon" and "dendrite" classes the priors are sparse and indirect: the absence of vesicles, for example, says nothing about the affinity of a supervoxel to the axonic or dendritic class. Fig. 7 demonstrates, just how sparse our prior knowledge is on the pixel level and how the volume coverage of the priors increases as we propagate them to w- and g-supervoxels. It is interesting to note, that the volumetric information gain which comes from the max-projection of the priors from w- to g-supervoxels is not reflected on the node level (compare the gray/purple bars in Fig. 7). In fact, we observe a percentage decrease of semantically conclusive supervoxels after GALA clustering.

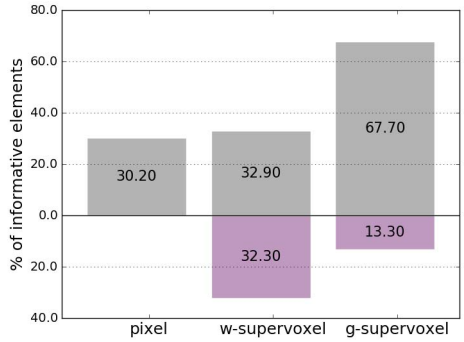


Fig. 7. Percentage of elements with prior knowledge ($p > 0.5$). The gray bars show the *volumetric* coverage in the isotropic test block (how many pixels have prior information): original pixelwise priors (left), projected on w-supervoxels (center), projected on g-supervoxels (right). The purple bars show the *nodewise* coverage in percent (supervoxels with prior information): for w-supervoxels (center) and for g-supervoxels (right).

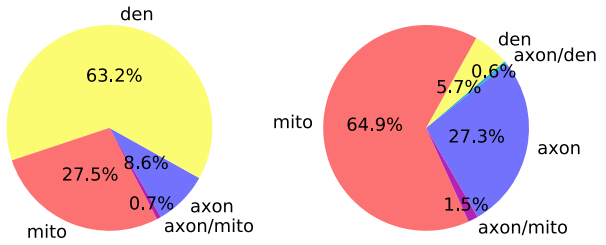


Fig. 8. Classwise composition of elements with prior knowledge from Fig. 7: pixelwise (left), supervoxelwise (right, for g-supervoxels used in AMWC). Elements with conflicting information are marked as overlap between the main classes in red, blue and yellow. For instance, in (b) 29.4% of all informative nodes vote for the axon class (blue) and 27.3% do so without local contradiction.

Seeking to interpret this decrease, we studied the classwise composition of the priors (Fig. 8). The dendrite class contains most of the informative pixels (Fig. 8 (a)). However, most of the easy merge decisions, which we delegate to GALA, are made in the "empty" dendritic shafts, and since the dendrite supervoxels are relatively large, their absolute number (Fig. 8 (b)) is small.

Fig. 8 also shows that co-occurrence of priors is not very common. In the rare cases where it happens it does not necessarily do harm. If all priors had maximal amplitude, all assignments to semantic classes would have been of the same cost and we would fall back to a prior-less situation.

The fact that AMWC can make use of sparse priors, where most of the data exhibits no information on class affinity, suggests that the algorithm could be applied to other problems, where the individuals to be segmented differ only by a small part or by a manually introduced seed. Another example from the bioimaging domain could be the segmentation of cells with different nuclei appearance or, for real world images, segmentation of airplanes with different tail logos or birds with different beaks. If, however, the prior information is not exceedingly sparse and the superpixels in the over-segmentation are sufficiently large, other merge-based segmentation algorithms could also benefit from the introduction of high-level priors. While this would be a difficult problem

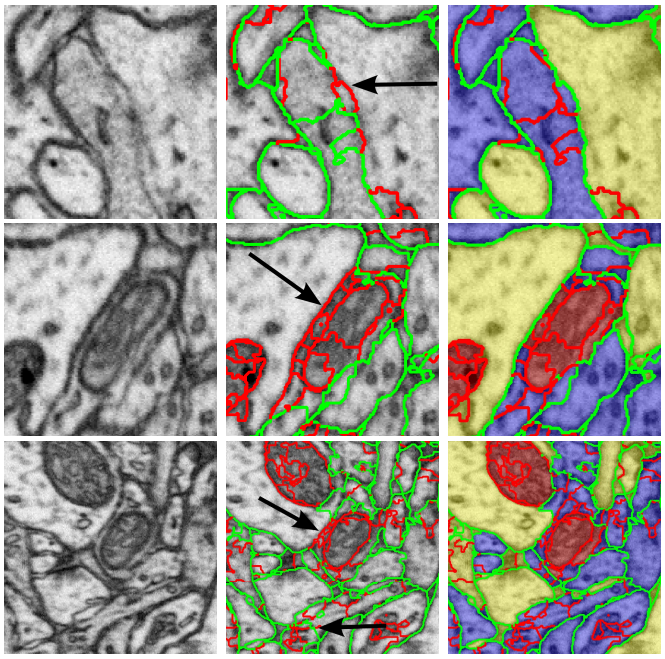


Fig. 9. Left: difficult locations in the raw data. Center: Multicut fails to correctly close the edge, pointed out by the arrows. Right: Asymmetric Multiway Cut overcomes this problem. The red supervoxel edges are merged in the final segmentation, while the green edges remain "on". In the AMWC result colors of the final segments denote their semantic class membership: blue for axon, yellow for dendrite, red for mitochondria.

for algorithms which merge superpixels pair by pair, delayed agglomeration approaches as in [27] or optimal cuts in the merge tree as in [29] could probably be extended to take such priors into account.

VI. Conclusions

We have introduced a principled approach to incorporate high-level biological priors into the Multicut neuron segmentation procedure of [11]. While the Multicut algorithm is already superior to greedy approaches due to its globally optimal nature, we show that it can be further improved by considering sparse global priors of neuron type in addition to local boundary evidence. The formulation of the partitioning problem in the edge domain allows us to find – by a cutting-planes optimization procedure – a globally optimal segmentation with respect to both the semantic and the transition probabilities. The respective increase of the computation time is counteracted by introduction of very large supervoxels, which also serve to propagate the sparse voxelwise prior information to larger parts of the image volume.

Segmentation of neurons with semantic differentiation between axons and dendrites is only the first of many possible areas of AMWC application. In future, we plan to extend it to other biological classes or other proxies for existing semantic classes. This is especially important to enable the use of the algorithm for non-mammalian neural tissue without strong axon/dendrite differentiation.

We hope that the improvement we demonstrated will lead other researchers to consider more high-level biological

information and finally serve as another step to fully automated connectome reconstruction.

REFERENCES

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

[2] S.-Y. Takemura *et al.*, "Synaptic circuits and their variations within different columns in the visual system of Drosophila," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 44, pp. 13711-–13716, 2015.

[3] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168–174, 2013.

[4] J. Anderson *et al.*, "Exploring the retinal connectome," *Molecular Vis.*, vol. 17, pp. 355–379, Feb. 2011.

[5] J. S. Kim *et al.*, "Space-time wiring specificity supports direction selectivity in the retina," *Nature*, vol. 509, no. 7500, pp. 331–336, 2014.

[6] S. Takemura *et al.*, "A visual motion detection circuit suggested by *Drosophila* connectomics," *Nature*, vol. 500, no. 7461, pp. 175–181, 2013.

[7] N. Kasthuri *et al.*, "Saturated reconstruction of a volume of neocortex," *Cell*, vol. 162, no. 3, pp. 648–661, 2015.

[8] N. Krasowski, T. Beier, G. Knott, U. Koethe, F. Hamprecht, and A. Kreshuk, "Improving 3d em data segmentation by joint optimization over boundary evidence and biological priors," in *Proc. IEEE 12th Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2015, pp. 536–539.

[9] T. Kroeger, J. Kappes, T. Beier, U. Koethe, and F. Hamprecht, "Joint supervised-unsupervised segmentation," in *Proc. GCPR*, 2014, pp. 1276–1286.

[10] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii, "Machine learning of hierarchical clustering to segment 2D and 3D images," *PLoS ONE*, vol. 8, no. 8, pp. 1–11, 2013.

[11] B. Andres *et al.*, "Globally optimal closed-surface segmentation for connectomics," in *Proc. ECCV*, 2012, pp. 778–791.

[12] T. Beier *et al.*, "Multicut brings automated neurite segmentation closer to human performance," *Nature Methods*, vol. 14, pp. 101–102, Jan. 2017.

[13] K. M. Harris *et al.*, "A resource from 3D electron microscopy of hippocampal neuropil for user training and tool development," *Sci. Data*, vol. 2, Sep. 2015, Art. no. 150046.

[14] V. Jain *et al.*, "Supervised learning of image restoration with convolutional networks," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[15] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2843–2851.

[16] E. Jurrus *et al.*, "Detection of neuron membranes in electron microscopy images using a serial neural network architecture," *Med. Image Anal.*, vol. 14, no. 6, pp. 770–783, 2010.

[17] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *Proc. ICCV*, 2013, pp. 2168–2175.

[18] G. Huang and V. Jain, "Deep and wide multiscale recursive networks for robust image labeling," in *Proc. ICLR*, 2014, pp. 1–2.

[19] J. Funke *et al.*, "Candidate sampling for neuron reconstruction from anisotropic electron microscopy volumes," in *Proc. MICCAI*, 2014, pp. 17–24.

[20] H.-F. Yang and Y. Choe, "Cell tracking and segmentation in electron microscopy images using graph cuts," in *Proc. IEEE Int. Symp. Biomed. Imag., Nano Macro (ISBI)*, Jun. 2009, pp. 306–309.

[21] S. N. Vitaladevuni and R. Basri, "Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2203–2210.

[22] V. Kaynig *et al.*, "Large-scale automatic reconstruction of neuronal processes from electron microscopy images," *Med. Image Anal.*, vol. 22, no. 1, pp. 77–88, 2015.

[23] D. Laptev, A. Vezhnevets, S. Dwivedi, and J. M. Buhmann, "Anisotropic ssTEM image segmentation using dense correspondence across sections," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI*. Berlin, Germany: Springer-Verlag, 2012, pp. 323–330.

[24] A. Vazquez-Reina, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister, "Segmentation fusion for connectomics," in *Proc. ICCV*, 2011, pp. 177–184.

[25] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F. A. Hamprecht, "Segmentation of SBFSEM volume data, of neural tissue by hierarchical classification," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2008, pp. 142–152.

[26] V. Jain *et al.*, "Learning to Agglomerate Superpixel Hierarchies," in *Proc. NIPS*, 2011, pp. 648–656.

[27] T. Parag, A. Chakraborty, S. Plaza, and L. Scheffer, "A context-aware delayed agglomeration framework for electron microscopy segmentation," *PLoS ONE*, vol. 10, no. 5, p. e0125825, 2015.

[28] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen, "A modular hierarchical approach to 3D electron microscopy image segmentation," *J. Neurosci. Methods*, vol. 226, pp. 88–102, Apr. 2014.

[29] M. G. Uzunbas, C. Chen, and M. Dimitris, "An efficient conditional random field approach for automatic and interactive neuron segmentation," *Med. Image Anal.*, vol. 27, pp. 31–44, Jan. 2016.

[30] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schn, "Globally optimal image partitioning by multicuts," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2011, pp. 31–44.

[31] A. Kreshuk *et al.*, "Automated detection and segmentation of synaptic contacts in nearly isotropic serial electron microscopy images," *PLoS ONE*, vol. 6, no. 10, p. e24899, 2011.

[32] C. Becker, K. Ali, G. Knott, and P. Fua, "Learning context cues for synapse segmentation," *IEEE Trans. Med. Imag.*, vol. 32, no. 10, pp. 1864–1877, Oct. 2013.

[33] P. Fua and G. W. Knott, "Modeling brain circuitry over a wide range of scales," *Frontiers Neuroanatomy*, vol. 9, p. 42, Apr. 2015.

[34] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1998, pp. 648–655.

[35] B. Andres *et al.*, "Segmenting planar superpixel adjacency graphs wrt non-planar superpixel affinity graphs," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2013, pp. 266–279.

[36] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. CVPR*, 2005, pp. 60–65.

[37] (2013). *Wikimedia Commons, Multipolar Neuron*. [Online]. Available: https://commons.wikimedia.org/wiki/File:Blausen_0657_MultipolarNeuron.png

[38] B. Andres, T. Beier, and J. H. Kappes. (2012). *OpenGM: A C++ Library for Discrete Graphical Models*. [Online]. Available: http://hci.iwr.uni-heidelberg.de/opengm2/

[39] J. H. Kappes, M. Speth, G. Reinelt, and C. Schnorr. (2013)."Higher-order segmentation via multicuts." [Online]. Available: https://arxiv.org/abs/1305.6387

[40] C. Straehle, U. Köethe, G. Knott, and F. Hamprecht, "Carving: scalable interactive segmentation of neural volume electron microscopy images," in *Proc. MICCAI*, 2011, pp. 653–660.

[41] C. Sommer, C. Straehle, U. Koethe, and F. Hamprecht, "Ilastik: Interactive learning and segmentation toolkit," in *Proc. ISBI*, Apr. 2011, pp. 230–233.

[42] S. Adams *et al.*, "Multicolor electron microscopy for simultaneous visualization of multiple molecular species," *Cell Chem. Biol.*, vol. 23, no. 11, pp. 1417–1427, 2016.