TED: A Tolerant Edit Distance for Segmentation Evaluation

Jan Funke[a,b,c], Jonas Klein[c], Francesc Moreno-Noguer[a], Albert Cardona[b], Matthew Cook[c]

[a] *Institut de Robòtica i Informàtica Industrial, UPC/CSIC Barcelona*
[b] *Janelia Research Campus, VA, Ashburn*
[c] *Institute of Neuroinformatics, UZH/ETH Zurich*

Abstract

In this paper, we present a novel error measure to compare a computer-generated segmentation of images or volumes against ground truth. This measure, which we call Tolerant Edit Distance (TED), is motivated by two observations that we usually encounter in biomedical image processing: (1) Some errors, like small boundary shifts, are tolerable in practice. Which errors are tolerable is application dependent and should be explicitly expressible in the measure. (2) Non-tolerable errors have to be corrected manually. The effort needed to do so should be reflected by the error measure. Our measure is the minimal weighted sum of split and merge operations to apply to one segmentation such that it resembles another segmentation within specified tolerance bounds. This is in contrast to other commonly used measures like Rand index or variation of information, which integrate small, but tolerable, differences. Additionally, the TED provides intuitive numbers and allows the localization and classification of errors in images or volumes. We demonstrate the applicability of the TED on 3D segmentations of neurons in electron microscopy images where topological correctness is arguable more important than exact boundary locations. Furthermore, we show that the TED is not just limited to evaluation tasks. We use it as the loss function in a max-margin learning framework to find parameters of an automatic neuron segmentation algorithm. We show that training to minimize the TED, *i.e.*, to minimize crucial errors, leads to higher segmentation accuracy compared to other learning methods.

1. Introduction

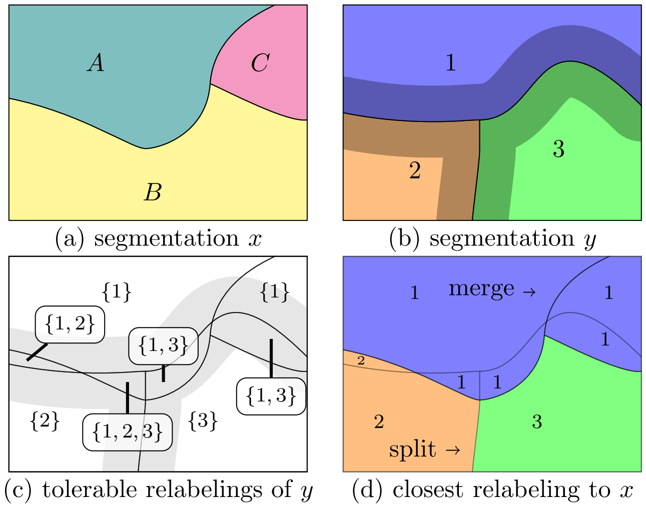
In the computer vision literature, several approaches to assess the quality of contour detection and segmentation algorithms can be found. Most of these measures have been designed to capture the intuition of what humans consider to be two similar results. In particular, these measures are supposed to be robust to certain tolerated deviations, like small shifts of contours. For the contour detection in the Berkeley segmentation dataset [1], for example, the precision and recall of detected boundary pixels within a threshold distance to the ground truth became the widely used standard [2, 3]. Contour error measures are, however, not a good fit for segmentations, since small errors in the detection of a contour can lead to the split or merge of segments. Therefore, alternatives like the Variation of Information (VOI), the Rand Index [4] (RI), the probabilistic Rand index [5, 6], and the segmentation covering measure [3], have been proposed.

However, these measures do not acknowledge that there are different criteria for segmentation comparison, and instead accumulate errors uniformly, even for many small differences that are irrelevant in practice. Especially in the field of biomedical image processing, we are often more interested in counting true topological errors like splits and merges of objects, instead of counting small deviations from the ground truth contours. This is in particular the case for imaging methods for which no unique "ground truth" labeling exists. In the imaging of neural tissue with Electron Microscopy (EM), for example, the preparation protocol can alter the volume of neural processes, such that it is hard to know where the true boundary was [7]. Further, the imaging resolution and data quality might just not be sufficient to clearly locate contours between objects [8], resulting in a high inter-observer variability.

1.1. Contributions

The main contribution of this paper is a novel measure to evaluate segmentations on a clearly specified tolerance criterion to address the aforementioned issues. At the core of our measure, which we call *Tolerant Edit Distance* (TED)[1], is an explicit tolerance criterion (*e.g.*, boundary shifts within a certain range). Using integer linear programming, we find the minimal weighted sum of split and merge operations to transform one segmentation into another, which is tolerably close to the ground truth. By setting the weights of the split and merge operations to the expected effort to perform these operations, the TED reflects the total effort needed to manually fix a segmentation. Similar to VOI and RI, our measure does not require voxels of the same object to form a connected component,

[1] Source code available at http://github.com/funkey/ted.



(a) segmentation x (b) segmentation y

(c) tolerable relabelings of y (d) closest relabeling to x

Figure 1: Illustration of the Tolerant Edit Distance (TED) between two segmentations x and y. By tolerating boundary shifts to a certain extend, shown as shadow in (b), y is allowed to be changed to match x as closely as possible. For that, we consider regions obtained by combining x and y, illustrated in (c). For each of these regions, we enumerate a set of labels used by y that are within a threshold distance to all locations inside the region (shown in curly brackets). This threshold is the maximally allowed boundary shift. Note that in this example, the region obtained from intersecting A and 3 can change its label to 1 (or keep 3), but not to 2, since it contains points that are too far away from region 2. Regions with only one possible label are too large to be relabeled by shifting their boundary and have to keep their initial label. From all the possible ways to relabel y, the relabeling (d) minimizing the number of split and merge errors compared to x is chosen by solving an integer linear program.

and can thus be applied to volumes with missing data, known object connections via paths outside the volume, or on stitched volumes with registration artifacts. The reported numbers are intuitive (*e.g.*, time or cost effort to fix a segmentation), easy to interpret (splits and merges of objects), and errors can be localized in the volume. An illustration of the TED can be found in Figure 1.

1.2. Application to Neuron Segmentation.

To demonstrate the usefulness of our measure, we present our results in the context of automatic neuron segmentation from EM volumes, an active field of biomedical image processing (for recent advances, see [9, 10, 11, 12, 13]). We argue that especially in this field there is a need for explicit and intuitive error measures. Furthermore, we show how the TED can be used to train neuron segmentation algorithms. Our findings (based on our previous work [14]) show that training to minimize the TED leads to higher segmentation accuracy on a range of error measures, compared to other methods.

1.2.1. Evaluation

As it is the case in many biological applications, the criterion to assess the quality of a neuron segmentation depends on the biological question one would like to answer. On one hand, *skeletons* of neurons are sufficient to identify individual neurons [15], to study neuron types and their function [16], and to obtain the wiring diagram of a nervous system (the so-called *connectome*) [8]. In these cases, topological correctness is far more important than the diameter of a neural process or the exact location of its boundary (see Figure 2 for examples). On the other hand, for biophysically realistic neuron simulation, *volumetric* information is needed to model action potential time dynamics, and to understand and simulate information processing capabilities of single neurons [17]. In this case, the segmentation should be close to the true volume of the reconstructed neurons. Only small deviations in the boundary location might still be tolerable.

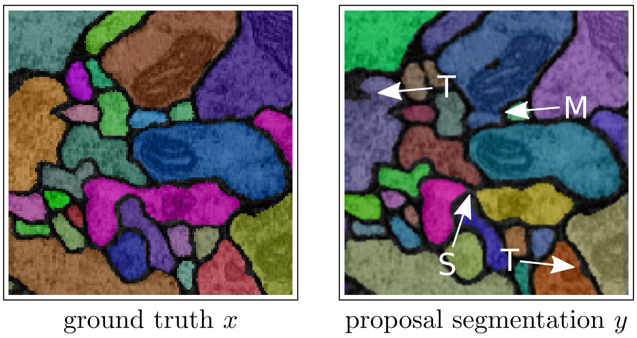


Figure 2: Example errors made by an automatic neuron segmentation algorithm. Errors like merges (M) and splits (S) dramatically change the reconstructed topology and should be avoided. Small disagreements in the boundary location (T) are however tolerable and should be ignored during evaluation.

Currently, reporting segmentation accuracy in terms of VOI or RI is the de-facto standard [11, 18, 10, 12, 13]. Less frequently used [9, 19] is the *Anisotropic Edit Distance* (AED) [9] and the *Warping Error* (WE) [20]. The AED is tailored to the specific error correction steps required for anisotropic volumes (splits and merges of 2D neuron slices within a section, connections and disconnections of slices between sections). The WE aims to measure the difference between ground truth and a proposal segmentation in terms of their topological differences. As such, the WE was the first error measure for neuron segmentation that deals with the delicate question of up to which point a boundary shift is not considered to be an error. However, since the WE assumes a foreground-background segmentation where connected foreground objects represent neurons, it is only applicable to volumes in which connectedness of neurons is preserved. Furthermore, only suboptimal solutions to the WE are found using a greedy, randomized heuristic, which makes it difficult to use for evaluation purposes. Consequently, the WE has found its main application in the training of neural networks for image classification [20].

In Section 2 we introduce the TED as an alternative to address some of the shortcomings of existing measures. Similar to the WE, the TED is designed to ignore small deviations from the ground truth and only count true topological errors, but is computed deterministically and to global optimality and does not impose constraints on the types of volumes being compared.

1.2.2. Training

Current state-of-the-art methods for automatic neuron segmentation can broadly be divided into isotropic [11, 18, 12, 13] and anisotropic methods [9, 10, 19]. Assignment models constitute the current state of the art for the segmentation of neurons from anisotropic volumes, as obtained by serial section EM [9, 10]. These models enumerate and price possible assignments of candidate segments across sections of EM stacks (see Figure 8 for an overview and Section 3 for details). A final segmentation is found by selecting a cost minimal and consistent subset of all assignments.

Learning in this kind of models consists of finding a function that maps from features of the candidate segments to a cost. Currently, this function is either set by hand [21, 10, 22], learned from a random forest classifier based on positive and negative assignment examples [9, 23], or found via grid-search by tuning weights of a small number of features [24]. Except for grid-search, which does not scale to larger sets of parameters, none of the currently used training methods implements real end-to-end learning. In Section 3, we show how to overcome these limitations by performing structured learning on a sensible loss function. For that, we solve two subproblems: (1) We show how to generate a *training sample* suitable for structured learning from human annotated ground truth. (2) We introduce a loss for structured learning, which minimizes the TED during learning.

We show that our learning framework leads to consistently higher segmentation accuracy compared to other learning methods. Furthermore, we show that our learning framework can be used to train on skeleton annotations without big sacrifices in segmentation accuracy. Skeleton annotations are non-volumetric centerlines of neurons, which are in practice much faster to obtain.

2. Tolerant Edit Distance

In this section, we formally introduce the TED and its associated optimization problem. We will show how to compute the TED for a specific class of tolerance criteria, of which the boundary shift is an example. Finally, we will analyze some of the properties of the TED in the context of neuron segmentation and contrast them with conventional error measures used in this field.

2.1. Definition of the TED

The TED measures the distance[2] between two segmentations $x : \Omega \mapsto K_x$ and $y : \Omega \mapsto K_y$, where Ω is a discrete set of voxel (or supervoxel) locations in a volume, and K_x and K_y are sets of labels used by x and y, respectively. The distance is reported in terms of the minimal number of splits and merges appearing in a relabeling of y, as compared with x. How y is allowed to be relabeled is defined on a tolerance criterion, *e.g.*, the maximal displacement of an object boundary.

We say that a label $k \in K_x$ overlaps with a label $l \in K_y$, if there exists at least one location $i \in \Omega$ such that $x(i) = k$ and $y(i) = l$. If x and y represent the same segmentation, each label l overlaps with exactly one label k, and vice versa. Consequently, if a label $k \in K_x$ overlaps with n labels from K_y, we count it as $n - 1$ splits. Analogously, if a label $l \in K_y$ overlaps with n labels from K_x, we count it as $n-1$ merges. For two labelings x and y, we denote as $\mathrm{s}(x, y)$ and $\mathrm{m}(x, y)$ the sum of splits and merges over all labels.

At the core of the TED lies a to-be-defined tolerance criterion T, which is meant to formalize our intuition about how a segmentation y is allowed to be relabeled. More formally, $T(y, y')$ is supposed to evaluate to $\top$ if y' is a tolerated relabeling, and to $\bot$ otherwise. With $\mathcal{Y}$ being the set of all labeling functions $y' : \Omega \mapsto K_y$, (*i.e.*, all possible labelings of Ω using the labels of y), we call the subset $\mathcal{Y}^+(y) = \{y' \in \mathcal{Y} \mid T(y, y') = \top\}$ the set of all tolerated relabelings of y. The TED is the minimal weighted sum of splits and merges over all tolerable relabelings $\mathcal{Y}^+(y)$:

$$\mathrm{TED}(x, y) = \min_{y' \in \mathcal{Y}^+(y)} \alpha\, \mathrm{s}(x, y') + \beta\, \mathrm{m}(x, y'), \qquad (1)$$

where the weights α and β represent the time or effort needed to fix a split or merge, respectively.

Without imposing restrictions on the tolerance criteria T, the optimization in (1) is intractable in general. Therefore, we restrict ourselves to what we call *local tolerance criteria* in the following. A local tolerance criterion is completely defined by providing relabel alternatives $A_i \subseteq K_y$ for each location i, such that each relabeling y' using any label $y'(i) \in A_i$ is tolerated. More formally,

$$T_{\mathrm{local}}(y, y') = \bigwedge_{i \in \Omega} y'(i) \in A_i. \qquad (2)$$

One example of such a local tolerance criterion is the boundary shift up to a distance threshold θ, which we illustrate in Figure 1 (c). For this tolerance criterion, A_i of a location i comprises the union of labels of all other locations that are within a θ distance from i.

For local tolerance criteria, (1) can be solved with the following integer linear program (ILP):

$$\min_{\mathbf{v}} \quad \alpha s + \beta m \qquad (3)$$

$$\text{s.t.} \quad \sum_{l \in A_i} v_{i \leftarrow l} = 1 \qquad \forall i \in \Omega \qquad (4)$$

$$\sum_{i \in \Omega} v_{i \leftarrow l} \geq 1 \qquad \forall l \in K_y \qquad (5)$$

[2] Note that, due to the intended tolerance to small deviations, the TED is not a proper metric on the space of segmentations. In slight abuse of nomenclature we use the term distance here anyway, which is sometimes used synonymous for metric.

$$a_{kl} - v_{i \leftarrow l} \geq 0 \qquad \forall i \in \Omega : x(i) = k \tag{6}$$

$$a_{kl} - \sum_{i \in \Omega : x(i) = k} v_{i \leftarrow l} \leq 0 \qquad \forall k \in K_x \;\; \forall l \in K_y \tag{7}$$

$$s_k - \sum_{l \in K_y} a_{kl} = -1 \qquad \forall k \in K_x \tag{8}$$

$$m_l - \sum_{k \in K_x} a_{kl} = -1 \qquad \forall l \in K_y \tag{9}$$

$$s - \sum_{k \in K_x} s_k = 0 \tag{10}$$

$$m - \sum_{l \in K_y} m_l = 0 \tag{11}$$

At the core of this ILP are binary indicator variables $\mathbf{v} = (v_{i \leftarrow l} \in \{0, 1\} \mid i \in \Omega,\ l \in A_i)$ to indicate the assignment of label l to location i. Constraints (4) and (5) ensure that exactly one of the labels gets chosen for each location and that each label of y has to appear at least once. Further, we introduce binary variables a_{kl} that indicate the presence of a joint assignment of label k from x and label l from y' at at least one location. With constraints (6) and (7) we make sure that each $a_{kl} = 1$ if and only if there is at least one location $i \in \Omega$ such that $x(i) = k$ and $y'(i) = l$. To count the number of times a label $k \in K_x$ is split in y', we further introduce integers $s_k \in \mathbb{N}$. These counts equal the number of times k was matched with any other label minus one, which we ensure with constraints (8). Analogously, we introduce integers m_l and constraints (9) for merges caused by label l in y'. The final split and merge numbers s and m are just the sums of the label-wise splits and merges, ensured by (10) and (11).

Once the optimal solution of this ILP has been found, the variables a_{kl} can be used to determine which labels got split and merged, and thus to localize errors.

2.2. Discussion of the TED

2.2.1. Parameters

As formulated above, the TED and the boundary shift tolerance criterion introduce three parameters: α and β to score differently split and merge errors, and a threshold θ for the maximally permitted boundary shift.

α and β can be set straightforwardly as the effort or time needed to fix a split or merge error. This depends on the concrete application and the tools available to proofreaders. Since a study of the time needed to fix segmentations is beyond the scope of this paper, we will proceed as follows: We set $\alpha = \beta = 1$ for this discussion, so as to count the number of errors. For the experiments presented in Section 3, we will set $\alpha = 1$ and $\beta = 2$ to reflect that merges are usually more difficult to fix than splits. Note that, up to scale, the TED will be the same for equally scaled α and β. We allow them to be set independently anyway to obtain directly a time-to-fix estimate if α and β reflect time.

The distance threshold θ might not be as obvious to set. Setting this value requires us to find an answer to the unpleasant question until which point a deviation from the ground truth is just a tolerable dent in a segment or a real error that should be counted. A single threshold alone is unlikely to provide an answer to this question. But, following a popular philosophy, we think that explicit is better than implicit. By explicitly setting this value, we achieve two things: First, we know exactly how to interpret the values measured by the TED. Second, we confront ourselves with the aforementioned unpleasant question, which we hope will encourage us to come up with more elaborate tolerance criteria, tailored to the needs of specific applications.

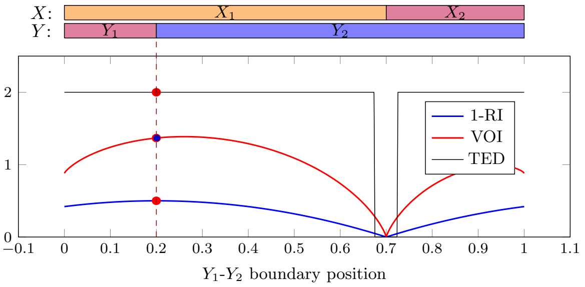


Figure 3: Comparison of Rand index (RI), variation of information (VOI), and tolerant edit distance (TED) as functions of object boundary displacements. Given a ground truth labeling X, the error measures are plotted as functions of the split position between two objects in a reconstruction Y. It can clearly be seen that TED assigns the same numbers (one split and one merge error) as soon as a given tolerance criterion is exceeded (0.025 in this example), regardless where the error happens. The TED is given as the sum of the possible errors, which is one split and one merge error unless the reconstruction object boundary is within the tolerated distance (0.025 in this example) to the true object boundary. VOI is in bits (lower is better) and 1-RI is 1 minus the ratio of agreeing pairs over all pairs (lower is better). The advantage of the TED is that it explicitly counts the topological errors made, regardless where in the segment they occur. Furthermore, small boundary shifts are not counted at all, whereas for RI and VOI their contribution can not be distinguished from real errors.

2.2.2. Shift of Object Boundary

To illustrate the behavior of different error measures in the case of object boundary displacements, we created a simple artificial 1D labeling consisting of two regions. In Figure 3, we show the errors of segmentations obtained by shifting the boundary between the objects. It can clearly be seen that TED assigns the same numbers (one split and one merge error) as soon as a given tolerance criterion is exceeded (0.025 in this example), regardless where the error happens. This is the desired outcome for applications like neuron segmentation, where it is important to count the number of topological errors regardless of how many voxels got affected.

2.2.3. Influence of Distance Threshold

In order to study the effect of the threshold distance for boundary shifts, we used an automatic segmentation

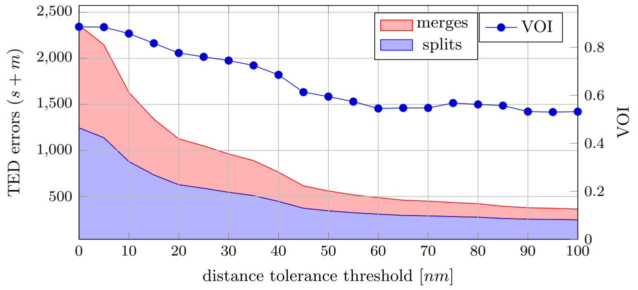


Figure 4: The tolerant edit distance (TED) on an automatically generated reconstruction as a function of the tolerated boundary shift.

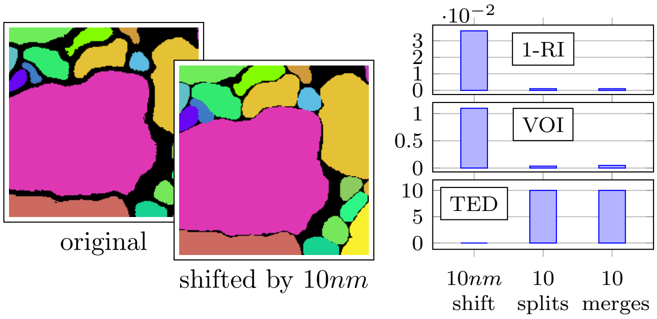


Figure 5: Comparison of error measures between the original ground truth (left) and three modifications. For the boundary shift experiment, the labels of the ground truth were dilated by $10nm$. For the *split* and *merge* experiments, ten random locations were chosen where the ground truth neurons were manually split or merged, respectively. Both RI and VOI assign better scores (*i.e.*, higher for RI, lower for VOI) to the *split* and *merge* experiments than to the *grow* experiment. The TED boundary shift tolerance was set to $20nm$ thus counts only the true morphological errors as false splits (FS) and false merges (FM).

result[3] and evaluated the TED for varying thresholds. Results are shown in Figure 4. The TED reveals that most of the errors occur within the range of about $50nm$, corresponding to about 12 pixels in the x-y-plane of this dataset. Depending on the biological question, those errors might be tolerable. In the same plot, we show the VOI of the closest tolerable relabeling to the ground truth under the given boundary shift threshold (*i.e.*, the equivalent of Figure 1 (d) on the proposal segmentation). From this example, we can see that the errors $< 50nm$ contribute quite significantly with 0.23 bits to the total VOI of 0.886, and thus can shadow true topological errors.

2.2.4. Comparison to RI and VOI

To demonstrate the main differences between TED and conventional error measures, we compare RI and VOI against TED for three manual modifications of the ground truth labeling of [25], shown in Figure 5. For the $10nm$ *shift* experiment, we dilated the boundaries of neurons in the ground truth by $10nm$. For the *splits* and *merges* experiment, we split and merged neurons at 10 randomly selected locations, respectively. It can be seen that the small shifts of object boundaries can have a significant contribution to the measures RI and VOI, which confirms our previous observation.

2.2.5. Localization of Errors

Due to the explicit tolerance criterion of the TED, errors can be localized in the volume. In Figure 6 we show example split and merge errors detected by the TED on an automatic segmentation result for the SNEMI dataset [26]. The boundary shift tolerance was set to $100nm$, which corresponds to $16.6 \times 16.6 \times 3.3$ voxels for this volume with a resolution of $6nm \times 6nm \times 30nm$. With this setup, the TED reveals true topological errors made by the automatic segmentation method. This allows analyzing the weaknesses of a method, which is both useful for model design as well as to communicate the limits of what can be done with a method to neuroscientists.

2.2.6. Runtime

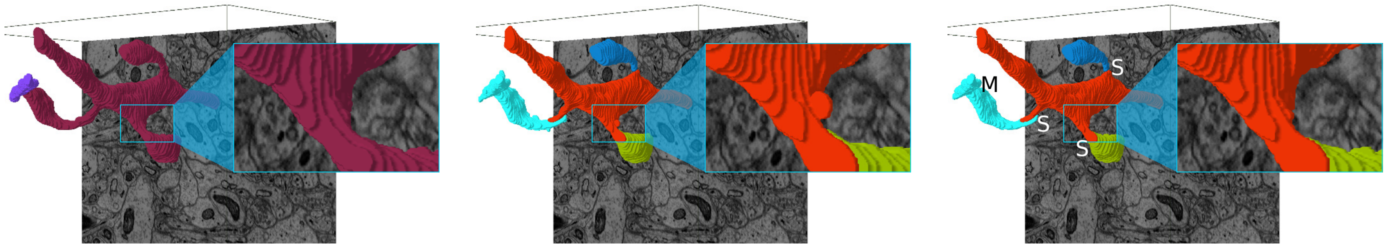
The runtime of the TED depends both on the size of the volumes and their discrepancy. The less similar two segmentations are, the more variables have to be introduced to represent the possible relabelings. This results in larger ILPs that are in general harder to solve.

We studied the impact of discrepancy on the runtime of the ILP by producing randomly generated segmentations. For that, we first created a reference segmentation by iteratively agglomerating supervoxels of a $1000 \times 1000 \times 100$ volume, using an affinity-based scoring function to propose the next merge[4]. We stopped the agglomeration at a manually set threshold to produce ~ 800 components. For the randomized segmentations, we added random noise of increasing intensity to the scoring function of the agglomeration to generate more and more discrepancies compared to the reference. Each segmentation obtained this way was compared against the reference segmentation. We measured the single-thread runtime on a Intel(R) Xeon(R) CPU with 2.2GHz, using Gurobi to solve the ILP. Results for 9 noise intensities (with 20 repetitions each) are shown in Figure 7. It can be seen that, although the number of errors goes up as high as 960, the vast majority of runs finished in less than $4s$. The number of variables in the ILPs ranged from 59397 (most similar segmentations) to 69105 (most dissimilar segmentations).

These results match our observations so far and can be summarized in the following way: If two segmentations are similar enough, the runtime of the TED seems to be moderate and an exact solution of the ILP is tractable in practice. Although we have so far not encountered intractable instances, we can not exclude their existence. We hypothesize that in such a case the segmentations in question would be very dissimilar and an approximate solution to the ILP would suffice.

[3] Obtained using SOPNET [9] on a publicly available EM dataset [25]

[4] We used the implementation http://github.com/funkey/waterz on a volume of neural tissue, for which we predicted voxel-wise affinities.



(a) ground truth x (b) proposal segmentation y (c) found relabeling of y closest to x

Figure 6: Errors found by the TED between a human generated ground truth x (a) and a proposal segmentation y (b), illustrated on two neurons (purple and red in ground truth). Small errors, as the one shown in the magnification, are tolerated and consequently removed in the found relabeling of y (c). Remaining errors are considered real splits (S) and merges (M).

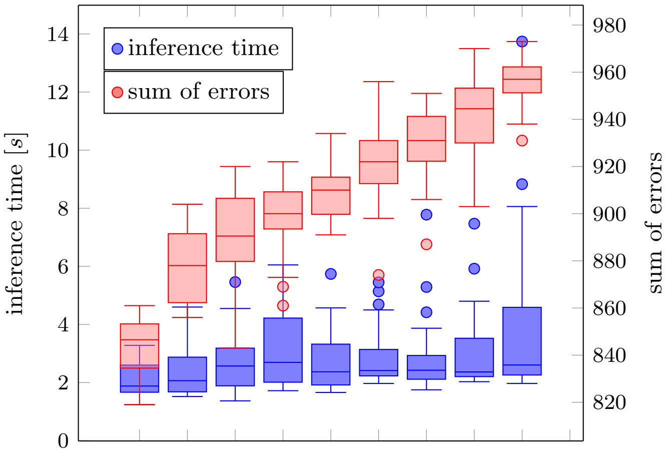


Figure 7: Runtime analysis of the TED computation on several randomized segmentations of a $1000 \times 1000 \times 100$ volume.

3. Learning of Assignment Models

In this section, we demonstrate that the TED can be used as a loss function to train neuron segmentation methods. Here, we focus on *assignment models* which gained popularity for the segmentation of anisotropic volumes of neural tissue. For that, we show first how the structured learning framework can be used to learn weights of a generic cost function. Second, we develop a tractable approximation of the TED that can be used as loss for structured learning. We report results on two publicly available datasets.

3.1. Assignment Models

Assignment models for anisotropic neuron segmentation introduce n binary indicator variables $\mathbf{z} \in \{0,1\}^n$ to represent possible assignments of 2D neuron candidates across consecutive pairs of sections of a volume (for an illustration see Figure 8, more details about assignment models can be found in [9, 10]). Linear constraints are formulated on the binary assignment indicators to ensure that a solution is consistent. In particular, the following set of constraints ensures that no pair of overlapping candidates are chosen (see also Figure 8 (g)):

$$\sum_{c \in C} \sum_{z_i \in \mathbf{z}_{\rightarrow c}} z_i \leq 1 \quad \forall C \in \mathcal{C}. \tag{12}$$

Here, $\mathcal{C}$ denotes the set of all conflict cliques, *i.e.*, sets of candidates that are mutually overlapping and $\mathbf{z}_{\rightarrow c}$ all assignment variables that link c to the previous section. For each conflict clique C, we require the number of assignment variables linking any candidate in it to the previous section to be at most 1. These constraints are accompanied by the following, which ensure a contiguous sequence of assignments (see also Figure 8 (h)):

$$\sum_{z_i \in \mathbf{z}_{\rightarrow c}} z_i - \sum_{z_i \in \mathbf{z}_{c \rightarrow}} z_i = 0 \quad \forall c. \tag{13}$$

Here, $\mathbf{z}_{c\rightarrow}$ denotes all assignments variables that link a candidate c to the next section. Noting that the above constraints are linear in $\mathbf{z}$, we can characterize the set of consistent solutions as

$$\mathcal{Z} = \{\mathbf{z} \in \{0,1\}^n | A\mathbf{z} \preceq \mathbf{b}\}, \tag{14}$$

where we write $\mathbf{a} \preceq \mathbf{b}$ to say that $\mathbf{a}$ is element-wise less than or equal to $\mathbf{b}$. Given a cost vector $\mathbf{c}$ for the assignment variables, the optimal assignment vector is the solution to the integer linear program

$$\min_{\mathbf{z} \in \mathcal{Z}} \langle \mathbf{c}, \mathbf{z} \rangle \,. \tag{15}$$

Without loss of generality, we assume that the cost c_i for selecting an assignment z_i is a weighted sum of features $\boldsymbol{\phi}_i$ extracted for this assignment:

$$\mathbf{c} = \Phi \cdot \boldsymbol{w} = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n]^\mathsf{T} \cdot \boldsymbol{w}. \tag{16}$$

3.2. Learning of Model Parameters

Using the structured learning framework [27], we find the optimal $\boldsymbol{w}$ given annotated training data $(\boldsymbol{\phi}, \mathbf{z}')$. More specifically, we use the margin rescaling variant to find the weights $\boldsymbol{w}^*$ as the minimizer of

$$L(\boldsymbol{w}) = \lambda |\boldsymbol{w}|^2 + \max_{z \in \mathcal{Z}} \left[\langle \Phi \boldsymbol{w}, \mathbf{z}' \rangle - \langle \Phi \boldsymbol{w}, \mathbf{z} \rangle \right] + \Delta(\mathbf{z}', \mathbf{z}), \tag{17}$$

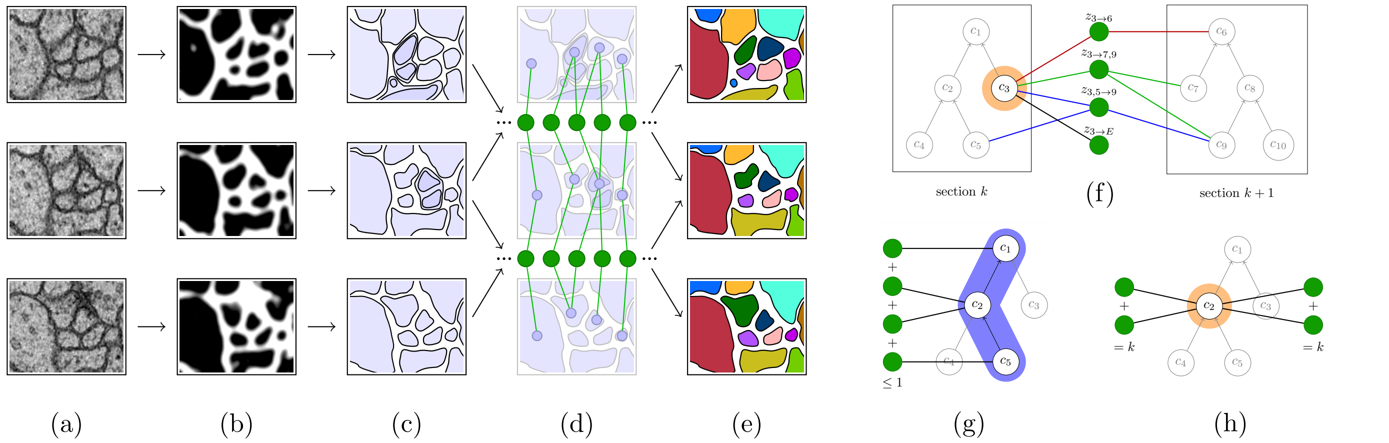


Figure 8: Assignment model for anisotropic neuron segmentation. From a stack of EM sections (a), a pixel classifier is used to predict membrane locations (b). Several, possibly overlapping, 2D neuron candidates are extracted for each section (c), and possible assignments are enumerated between candidates of adjacent sections (d). In the model, each assignment of candidates between two sections (f) is represented by a binary variable z_i and has an associated cost c_i for selecting it. By finding a cost-minimal $\mathbf{z}$ subject to constraints (g) and (h) yields a segmentation (e).

where λ is the regularizer weight and $\Delta(\mathbf{z}', \mathbf{z})$ is an application specific loss function. In order for this method to be successful, two problems need to be solved: (1) a representative training sample $\mathbf{z}'$ has to be found, and (2) a sensible loss function $\Delta(\mathbf{z}', \mathbf{z})$ has to be designed.

3.3. Training Sample $\mathbf{z}'$

Even apart from the difficulties in obtaining unambiguous human generated ground truth for the neuron segmentation problem in the first place, the provision of $\mathbf{z}'$ is not trivial: We have to find a member of $\mathcal{Z}$, *i.e.*, the set of all possible assignment vectors *using the found 2D neuron candidates*, that is as close as possible to the human annotated ground truth. We have to note that the extracted 2D neuron candidates can be imperfect and thus there might not be a $\mathbf{z} \in \mathcal{Z}$ that corresponds to the human annotated ground truth. Consequently, we have to accept that the training sample $\mathbf{z}'$ will only represent a best-effort solution and not the ground truth.

In order to find this best-effort solution in a principled way, we assign a local ground truth matching score g_i to each assignment and then select a consistent solution that minimizes this score. Let $\Omega = [1, W] \times [1, H] \times [1, D]$ be the set of all discrete pixel locations in a stack of size $W \times H \times D$. We assume a ground truth labeling $x : \Omega \mapsto K$ that assigns a unique label $k \in K$ to each ground truth segment in the volume. Let $u(i)$ and $v(i)$ denote the section indices that are linked by assignment z_i. We denote by $A_i \subset \Omega$ the set of pixels of section $u(i)$ and $v(i)$ that are merged by the assignment z_i. Similarly, let $G_i^k \subset \Omega$ denote the set of pixels that are labeled to belong to the same region k in the ground truth, limited to the sections $u(i)$ and $v(i)$. For each pair of assignment i and ground truth label k, we compute a similarity g_i^k that rewards overlap between the sets A_i and G_i^k and punishes set differences:

$$g_i^k = \underbrace{|G_i^k \cap A_i|}_{\text{overlap}} - \underbrace{\left(|G_i^k \setminus A_i| + |A_i \setminus G_i^k|\right)}_{\text{set difference}} . \qquad (18)$$

The final matching score g_i of an assignment z_i is the maximal similarity with any ground truth label:

$$g_i = \max_{k \in K} g_i^k . \qquad (19)$$

The scores g_i reflect, for each assignment z_i, how well it locally fits to the ground-truth. We use these scores to find the overall best assignment $\mathbf{z}$ by solving the following ILP:

$$\mathbf{z}' = \operatorname*{arg\,max}_{\mathbf{z} \in \mathcal{Z}} \quad \langle \mathbf{g}, \mathbf{z} \rangle . \qquad (20)$$

Note that this ILP is maximizing the sum of similarities for all assignments. This way that we find a consistent solution (in terms of the constraints introduced in Section 3.1) that maximizes similarity with the provided ground truth.

3.4. Loss $\Delta(\mathbf{z}', \mathbf{z})$

Ideally, we would use the error measure that we use to evaluate the results of our automatic segmentation as $\Delta(\mathbf{z}', \mathbf{z})$. However, we have to make sure that the maximization in (17) is still tractable.

To this end, we suggest a first order approximation of the TED to be used as $\Delta(\mathbf{z}', \mathbf{z})$: For each assignment variable z_i, we estimate its contribution l_i to the TED score. If $z_i = z_i'$, no error was introduced by z_i and hence its contribution is 0. If, however, $z_i \neq z_i'$, the resulting segmentation will deviate from the best-effort solution. In order to estimate the contribution of an erroneous z_i to the TED score, we compute the TED score between two segmentations $y_{\mathbf{z}'}$ and $y_{\bar{\mathbf{z}}(i)}$: $y_{\mathbf{z}'}$ denotes the segmentation obtained from the best-effort solution $\mathbf{z}'$ and $y_{\bar{\mathbf{z}}(i)}$ denotes the segmentation obtained by $\mathbf{z}'$, but with z'_i inverted[5]. More formally, we set

$$l_i = (1 - 2z'_i)\,\mathrm{TED}(y_{\mathbf{z}'}, y_{\bar{\mathbf{z}}(i)}) \quad \text{and} \quad c = \sum_{i:z'_i=1} -l_i, \tag{21}$$

where some of the contributions l_i turn into rewards (negative values) for *using* an assignment, *i.e.*, when the corresponding $z'_i = 1$. This linearization allows us to model the loss as a linear function of $\mathbf{z}$:

$$\Delta(\mathbf{z}', \mathbf{z}) = \langle \mathbf{l}, \mathbf{z} \rangle + c \approx \mathrm{TED}(y_{\mathbf{z}'}, y_{\mathbf{z}}), \tag{22}$$

which favorably plugs into (17). In fact, the loss augmented inference problem for a given $\boldsymbol{w}$ has the same structure as the inference problem (15) itself, for which we already know that it is tractable in practice:

$$\max_{z \in \mathcal{Z}} \langle \mathbf{l} - \Phi\boldsymbol{w}, \mathbf{z} \rangle + \underbrace{\langle \Phi\boldsymbol{w}, \mathbf{z}' \rangle + c}_{\text{constant}}. \tag{23}$$

3.5. Results

We use two publicly available datasets for our experiments, which we refer to as Drosophila [25], which consists of two stacks of 20 EM sections with $4 \times 4 \times 40nm$ resolution ($1024 \times 1024 \times 20$ pixels), and Mouse Cortex [26], which consists of two stacks of 100 EM sections with $6 \times 6 \times 30nm$ resolution ($1024 \times 1024 \times 100$ pixels).

We split the parts for which ground truth was available into two stacks of equal size (2×10 sections for Drosophila and 2×50 sections for Mouse Cortex). For each dataset, we trained all methods on a sample $\mathbf{z}'$ (see Section 3.3) extracted from the first stack and report the results on the second stack.

We trained and evaluated the assignment model implemented in Sopnet [9], using membrane predictions from [28], and 2D neuron candidates extracted from component trees [9]. We used the default features implemented in Sopnet for Φ.

3.5.1. Comparison of Learning Methods

We compare the structured learning method proposed in Section 3 to random forests (RF) as proposed in [9, 19], support vector machines (SVM), and *overlap*. RF and SVM learn to score each assignment based on positive and negative examples provided by $\mathbf{z}'$ (see Section 3.3). As a baseline, *overlap* uses the number of overlapping pixels of an assignment across sections as score. Since these methods need a prior for the selection of assignments, we trained RF and SVM on a subset of the training data (5 sections for Drosophila, 40 sections for Mouse Cortex) and used the rest to validate a prior for RF, SVM, and *overlap* with a grid-search minimizing the Hamming distance to $\mathbf{z}'$.

To study the performance of the structured learning method, we compare our loss SL-TED (see Section 3.4) against three baselines: SL-Ham, SL-VOI, and SL-RI. SL-Ham uses the Hamming distance of $\mathbf{z}$ to $\mathbf{z}'$ for $\Delta(\mathbf{z}', \mathbf{z})$. SL-VOI and SL-RI use the same linear approximation scheme we developed for the TED (see (21)), but with VOI and RI as error measures instead of TED. For the computation of SL-TED, we evaluated the TED allowing boundary shifts up to $\theta = 100nm$, with weights $\alpha = 1$ and $\beta = 2$ to account for the fact that merges lose geometric information and thus usually take more time to repair than splits.

Results are shown in Table 1. We report errors for several commonly used measures for neuron segmentation: Rand Index (RI), Variation of Information (VOI), Anisotropic Edit Distance [9] (AED, note that we refer to the inter FP/FN as FS/FM), and TED. The TED counts topological errors that are not considered boundary shifts as false splits (FS) and false merges (FM). Splits of the ground truth background label are false positives (FP) and merges involving the reconstruction background label false negatives (FN). For the time-to-fix (TTF) estimate, we again set the time needed for fixing a split to $\alpha = 1$ and for fixing a merge to $\beta = 2$. The structured learning methods are in general superior to *overlap*, RF, and SVM, with the best results being obtained by training on SL-TED. Training on the TED-approximation SL-TED does indeed minimize the TTF. Furthermore, RI, VOI, and AED are minimized. Our results also reveal interesting differences between error measures: Although the best solutions in terms of TED have also best RI, VOI, and AED, we see a discrepancy in the mid-field: on Drosophila, SVM scores much better than RF in terms of VOI and slightly better in terms of RI. However, TED on a clearly defined criterion shows that the numbers are misleading and in fact RF has less errors in total and shorter TTF.

3.5.2. Learning from Skeletons

We show on Mouse Cortex that our method to find a training sample $\mathbf{z}'$ allows us to train on skeleton annotations as well. Skeleton annotations are not volumetric, *i.e.*, instead of labeling every pixel, only the centerline of the neuron is provided as training data. In practice, this saves a lot of manual labeling effort such that larger volumes can be annotated. To simulate skeleton annotations and compare them to the learning outcome of complete ground truth, we skeletonized each ground truth label of the training stack. For that, we shrunk each 2D connected component of one label in each EM section to a single pixel at its center of mass. Consequently, we adjusted the search for the training sample $\mathbf{z}'$ to not consider the set difference term in (18). The results of training with SL-TED on the $\mathbf{z}'$ obtained this way are shown in Table 2. Although significant, the loss in accuracy might be compensated by the time saved to annotate only skeletons for training.

[5] Since the constraints (14) might not allow inverting single variables in isolation, we identify a minimal group of variables that have to be inverted as well to obtain a consistent solution: for each assignment i, we find an assignment vector $\bar{\mathbf{z}}(i) \in \mathcal{Z}$ that has $z_i \neq z'_i$ and minimizes the Hamming distance to $\mathbf{z}'$.

Mouse Cortex Dataset

			VOI				AED					TED		
method	Rand	split	merge	total	FP	FN	FS	FM	total	FP	FN	FS	FM	TTF
overlap	0.9939	0.668	0.192	0.860	1,553	2,404	3,114	1,666	8,737	155	179	678	57	1,305
RF	0.9936	0.375	0.291	0.666	1,048	2,546	3,014	1,451	8,059	23	151	273	68	734
SVM	0.9572	0.507	1.434	1.940	2,998	3,761	5,155	4,587	16,501	4	147	129	167	761
SL-Ham	0.9933	0.348	0.309	0.657	895	2,258	2,735	1,333	7,221	23	138	243	82	706
SL-VOI	0.9870	0.525	0.899	1.424	799	2,466	2,884	1,325	7,474	14	127	161	141	711
SL-RI	0.9797	0.514	1.047	1.561	780	2,604	3,004	1,291	7,679	14	129	163	143	721
SL-TED	**0.9948**	0.331	0.275	**0.606**	838	2,297	2,752	1,268	**7,155**	18	135	229	82	**681**

Drosophila Dataset

			VOI				AED					TED		
method	Rand	split	merge	total	FP	FN	FS	FM	total	FP	FN	FS	FM	TTF
overlap	0.9906	0.309	0.340	0.648	179	517	648	254	1,598	13	58	201	99	528
RF	0.9864	0.934	0.518	1.452	181	585	556	252	1,574	1	175	108	35	529
SVM	0.9890	0.804	0.230	1.034	366	357	593	537	1,853	10	86	224	84	574
SL-Ham	0.9959	0.309	0.080	0.389	241	234	375	250	1,100	14	63	227	47	461
SL-VOI	0.9959	0.301	0.101	0.402	214	268	400	202	**1,084**	14	60	243	54	485
SL-RI	0.9958	0.301	0.109	0.410	202	288	419	192	1,101	16	59	243	60	497
SL-TED	**0.9960**	0.299	0.087	**0.386**	224	249	382	239	1,094	15	63	215	50	**456**

Table 1: Comparison of segmentation results of different learning methods on two anisotropic EM datasets.

Mouse Cortex Dataset

			TED		
method	FP	FN	FS	FM	TTF
volumetric ground truth	18	135	229	82	**681**
skeleton ground truth	17	114	188	152	737

Table 2: Reconstruction results on Mouse Cortex after training on different ground truth types: *volumetric* uses the original ground truth, *skeleton* a skeletonized version. We show false splits and false merges (FS and FM), false positives and false negatives (FP and FN), and an estimated time-to-fix (TTF), as reported by the TED measure.

3.5.3. Runtimes

The bottleneck of our method is the computation of the coefficients l_i needed for the TED approximations SL-TED, since for every binary variable in the $\mathbf{z}'$ the TED has to be evaluated. For Mouse Cortex and Drosophila, $\mathbf{z}'$ contained 277,874 and 20,890 variables, respectively. Computing the coefficients took 64.3h for Mouse Cortex and 4.8h for Drosophila on a 12 core Intel Xeon CPU with 3.47 GHz. By noting that the influence of a single variable flip is usually local, the computation of the TED could be limited to constant size subvolumes around the variable of interest, such that the effort of computing the coefficients scales linearly with the best-effort size. Structured learning with SL-TED took 30m for Drosophila and 1h45m for Mouse Cortex on 10 cores of a Intel Xeon CPU with 2.6 GHz. We used an iterative cutting plane method[6] to minimize the convex learning objective (17) to optimality. The maximization in (17) has been solved with an ILP to optimality (using the Gurobi solver) in each iteration as well.

4. Conclusions

We presented the TED, a novel measure for segmentation comparison, which tolerates small errors based on an explicit tolerance criterion and therefore focusses on counting true topological errors. As such, it is suited to report an effort or time to fix estimate.

A current limitation of the TED is the restriction to use local tolerance functions, *e.g.*, a boundary shift up to a certain threshold. More complex tolerance criteria that do not factorize over regions are currently not expressible. Although they could in theory be incorporated into the ILP by adding auxiliary variables, it remains questionable whether the resulting problem is still tractable. Even though we did not observe that empirically, it is already conceivable in the current formulation that an optimal solution to the ILP can not be found in reasonable time. This could in particular be the case if ground truth and proposal segmentation differ a lot and a very lax tolerance criterion is used. In these cases, approximate solutions to the proposed ILP might be worth considering.

[6] Source code available at http://github.com/funkey/sbmrm

Besides being a tool to assess the quality of a segmentation, we also showed that the TED can be used to train a neuron segmentation algorithm.

We believe that the key for the superior performance of training using the TED compared to other losses is the consideration of topological errors. Previous attempts tried to correctly classify each assignment decision and did not take into account the severity of a wrong decision in terms of split and merge errors in the result. Training on a TED approximation overcomes this problem.

It is worth noting that the boundary shift we used as a tolerance criterion is just one example of how to use the TED for training and evaluation. Depending on the biological question, more or less deviations from the ground truth can be permitted. For example, boundary shifts could be tolerated to an extent that locally depends on the diameter of the ground truth neuron. In future work, it will be interesting to investigate the use of the TED for more general biomedical image processing problems with more specific tolerance criteria.

5. Acknowledgements

We thank the reviewers for their valuable suggestions. This work was funded by SNF grant 149861.

6. References

[1] D. R. Martin, C. C. Fowlkes, D. Tal, J. Malik, A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics, in: ICCV, Vol. 2, 2001, pp. 416–423.

[2] D. R. Martin, C. C. Fowlkes, J. Malik, Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues, PAMI.

[3] P. Arbeláez, M. Maire, C. C. Fowlkes, J. Malik, From Contours to Regions: An Empirical Evaluation, in: CVPR, 2009.

[4] W. M. Rand, Objective Criteria for the Evaluation of Clustering Methods, Journal of the Americal Statistical Association 66 (1971) 846–850.

[5] R. Unnikrishnan, M. Hebert, Measures of Similarity, in: Seventh IEEE Workshop on Applications of Computer Vision, 2005.

[6] R. Unnikrishnan, C. Pantofaru, M. Hebert, Toward objective evaluation of image segmentation algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 929–944.

[7] G. E. Sosinsky, J. Crum, Y. Z. Jones, J. Lanman, B. Smarr, M. Terada, M. E. Martone, T. J. Deerinck, J. E. Johnson, M. H. Ellisman, The combination of chemical fixation procedures with high pressure freezing and freeze substitution preserves highly labile tissue ultrastructure for electron tomography applications, Journal of Structural Biology 161 (3) (2008) 359–371.

[8] A. Cardona, Towards semi-automatic reconstruction of neural circuits, Neuroinformatics 11 (1) (2013) 31–33.

[9] J. Funke, B. Andres, F. A. Hamprecht, A. Cardona, M. Cook, Efficient Automatic 3D-Reconstruction of Branching Neurons from EM Data, in: CVPR, 2012, pp. 1004–1011.

[10] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller, J. Lichtman, H. Pfister, Large-scale automatic reconstruction of neuronal processes from electron microscopy images, Medical Image Analysis 22 (1) (2015) 77–88.

[11] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, D. B. Chklovskii, Machine Learning of Hierarchical Clustering to Segment 2D and 3D Images, Plos One 8 (8) (2013) e71715.

[12] T. Parag, S. M. Plaza, L. K. Scheffer, Small Sample Learning of Superpixel Classifiers for EM Segmentation- Extended Version, in: CoRR, Vol. abs/1406.1, 2014.

[13] G. B. Huang, V. Jain, Deep and Wide Multiscale Recursive Networks for Robust Image Labeling, in: arXiv preprint arXiv:1310.0354, 2014.

[14] J. Funke, J. Klein, F. Moreno-Noguer, A. Cardona, M. Cook, Structured Learning of Assignment Models for Neuron Reconstruction to Minimize Topological Errors, in: ISBI, 2016.

[15] H. Peng, P. Chung, F. Long, L. Qu, A. Jenett, A. M. Seeds, E. W. Myers, J. H. Simpson, BrainAligner: 3D Registration Atlases of Drosophila Brains, Nature Methods 8 (6) (2011) 493–498.

[16] W. Denk, K. L. Briggman, M. Helmstaedter, Structural neurobiology: missing link to a mechanistic understanding of neural computation, Nature reviews Neuroscience.

[17] M. London, M. Häusser, Dendritic Computation, Annual Review of Neuroscience 28 (1) (2005) 503–532.

[18] T. Kröger, S. Mikula, W. Denk, U. Köthe, F. A. Hamprecht, Learning to segment neurons with non-local quality measures, in: MICCAI, Vol. 16, 2013, pp. 419–427.

[19] J. Funke, J. Martel, S. Gerhard, B. Andres, D. C. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, H. Pfister, A. Cardona, M. Cook, Candidate Sampling for Neuron Reconstruction from Anisotropic Electron Microscopy Volumes, in: MICCAI, 2014, pp. 17–24.

[20] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. Mendenhall, W. C. Abraham, K. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. Tapia, J. Lichtman, S. H. Seung, Boundary Learning by Optimization with Topological Constraints, in: CVPR, 2010.

[21] H. J. H. Jiang, S. Fels, J. Little, A Linear Programming Approach for Multiple Object Tracking, in: CVPR, 2007.

[22] F. Jug, T. Pietzsch, D. Kainmüller, J. Funke, M. Kaiser, E. van Nimwegen, C. Rother, E. W. Myers, Optimal Joint Segmentation and Tracking of Escherichia Coli in the Mother Machine, in: BAMBI, 2014.

[23] A. Vazquez-Reina, D. Huang, M. Gelbart, J. Lichtman, E. Miller, H. Pfister, Segmentation Fusion for Connectomics, in: ICCV, 2011.

[24] B. X. Kausler, M. Schiegg, B. Andres, M. Lindner, U. Köthe, H. Leitte, J. Wittbrodt, L. Hufnagel, F. A. Hamprecht, A discrete chain graph model for 3d+t cell tracking with high misdetection robustness, in: ECCV, 2012.

[25] S. Gerhard, J. Funke, J. Martel, A. Cardona, R. D. Fetter, Segmented anisotropic ssTEM dataset of neural tissue (2013).

[26] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cirean, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, H. S. Seung, Crowdsourcing the creation of image segmentation algorithms for connectomics, Frontiers in Neuroanatomy 9 (2015) 142.

[27] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, Y. Singer, Large margin methods for structured and interdependent output variables, Journal of Machine Learning Research 6 (2005) 1453–1484.

[28] D. C. Ciresan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images, in: NIPS, Vol. 25, 2012, pp. 2843–2851.