Entropy policy for supervoxel agglomeration of neurite segmentation

Tristan Hascoet
Graduate School of System Informatics
Kobe University
Kobe, Japan
tristan.hascoet@gmail.com

Baptiste Metge
baptiste.metge@gmail.com

Tetsuya Takiguchi
Center for Urban Safety
Kobe University
Kobe, Japan
takigu@kobe-u.ac.jp

Yasuo Ariki
Center for Urban Safety
Kobe University
Kobe, Japan
ariki@kobe-u.ac.jp

***Abstract*—The field of connectomics aims to map the interconnections between biological neurons within nervous systems at the scale of single synapses to gain insights into the structure and functional organization of biological neural networks. A critical task for the success of the connectomics enterprise is the segmentation of neurites from high precision electron microscopy (EM) images. State-of-the art approaches addressing this challenging problem typically proceed in three steps: First cell boundaries are predicted by a Convolutional Neural Network (CNN) from the raw EM images. Second, boundary maps are oversegmented into supervoxels. Finally, supervoxels are agglomerated to produce the segmentation output. In this work, we focus on the task of supervoxel agglomeration within such a neurite segmentation pipeline. Existing greedy agglomeration algorithms proceed by iteratively merging supervoxel in order of decreasing probability. In this work, we propose an alternative to such greedy agglomeration policy by modeling the uncertainty yielded by merge operations. Instead of greedily merging boundaries with highest merge probability, we propose to merge boundaries so as to reduce the entropy of the boundary probability distribution. We validate our idea on a standard benchmark and show important boosts in performance compared to the standard greedy merge policy. Our algorithms comes with negligible additional computational cost and can be directly integrated within existing greedy agglomeration frameworks.**

***Index Terms*—Segmentation, supervoxel agglomeration, Entropy minimization**

I. Introduction

Precise reconstruction of neural connectivity is of great importance to understand the functional organization of biological nervous systems. 3D electron microscopy (EM) can capture large volume of neuronal tissues within nano-scale precision, which allows for the identification of even the smallest neuronal objects like vesicles. With advances in imaging technologies, EM systems can now produce terabytes of images within hours. Manually annotating each of the neurons within such large EM volumes is simply impractical as it would requires lifetimes of manual labeling from highly skilled experts to segment. Hence, high-precision segmentation models are needed to automate the reconstruction of neuronal circuits.

Current state-of-the art models for 3D neurite segmentation proceeds in three steps, as illustrated in Figure 1. First, a CNN is trained to detect boundaries between neurons in the raw EM images. In a second step, over-segmentation maps are computed from the boundary map. This is typically done by non-parametric algorithms like Watershed [9]. These procedures typically lead to severe over-segmentation. We refer to these over-segmented regions as *supervoxels*.

Supervoxels are then merged in the last step of this pipeline, using either greedy local objectives [3]–[5] or globally optimal objectives [6]–[8]. While global objectives tend to perform favorably [8], these solutions are unlikely to scale from small-scale research benchmarks to the large-scale settings of practical interest. Hence, we focus our attention on greedy agglomeration approaches. Greedy agglomeration algorithms assign an aggregation score to each pair of adjacent supervoxels, and supervoxel pairs of highest score are greedily merged at each step of the agglomeration process. Figure 2 illustrates one step of greedy supervoxel agglomeration. In its most basic form, the mean boundary probability between two adjacent supervoxels, as computed by the CNN used in the first step of the pipeline, is used as aggregation score. Other algorithms compute aggregation scores from more complex hand-crafted features [4], [5].

In this work, we propose a simple twist to existing greedy agglomeration algorithms. The main insight behind our method is as follows: As a side-effect, merging two supervoxels impacts the aggregation score of neighboring supervoxel pairs. Certain merge operations increase the entropy of the aggregation score distribution, which increases the difficulty of subsequent merge operations. Other merge operations decrease the entropy of the aggregation score distribution, which eases the decision of subsequent merge operations. Hence, in addition to the aggregation score between supervoxels, we also consider the entropy shift of the global aggregation score distribution induced by the merging operation. We prioritize merging of supervoxels that reduce uncertainty in the distribution of the aggregation scores, in order to ease subsequent merging decisions.

In the following section, we relate our work to the existing literature on neurite segmentation and supervoxel agglomeration. Section 3 briefly defines the mathematical notions and notations used in the presentation of our proposed algorithm (in Section 4). Section 5 presents our experiments and results.

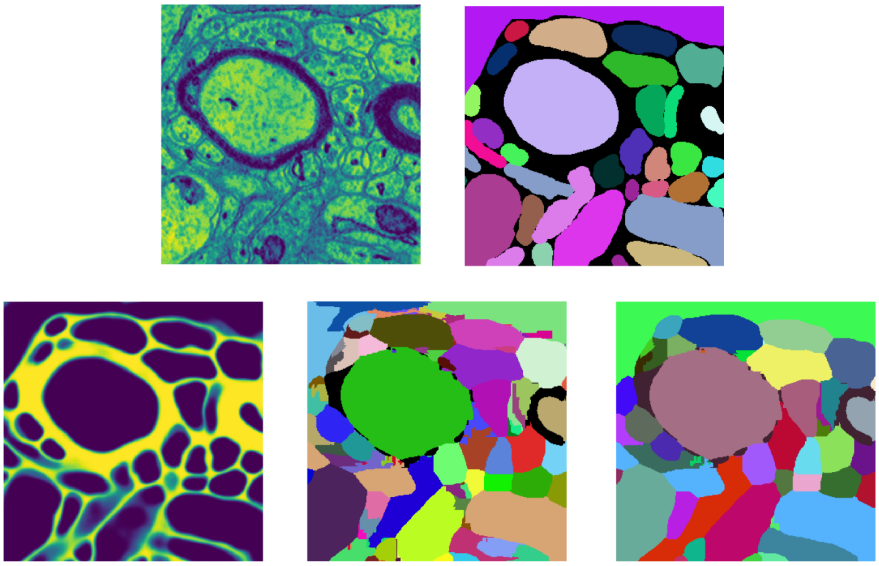

Fig. 1. Illustration of the segmentation pipeline. Top: Input EM image and ground-truth segmentation. Bottom: Successive steps of the segmentation pipelines (from left to right): Boundary map prediction, over-segmentation into supervoxels and supervoxel aggregation. The output of the supervoxel aggregation is the final output of the segmentation pipeline.

II. Related Work

Oversegmentation into supervoxels followed by agglomeration has been proposed as a strategy for segmenting EM images in [1], [2], [4], [5]. The vast majority of previous works have used the Watershed algorithm [9] to over-segment the boundary maps into supervoxels.

To aggregate super-voxels, the MULTICUT [6], and lifted MULTICUT algorithms [7], [8] train a classifier to predict the connectivity of fragments that are then clustered by solving a computationally expensive combinatorial optimization problem.

More related to our work, both CELIS [3] and GALA [4], [5] train a classifier to predict aggregation scores between adjacent supervoxels. Aggregation scores are used by a greedy hierarchical agglomeration strategy to produce the segmentation output. These works have emphasized learning of the scoring function, often using hand-designed features as input, which tends to increase the computational complexity of agglomeration during inference.

Lee *et al*. [10] have found that scoring pairs of supervoxels with a single hand-designed feature, the mean boundary probability of voxels adjacent to supervoxels pairs, often produces good agglomeration accuracy. They conjecture that the quality of convolutional network outputs have improved so much in recent years that the benefits of hand-designed features as used by the GALA algorithm have drastically decreased.

Our work can be integrated within any existing greedy agglomeration framework [3], [5], [10]. Following, Lee *et al*., we will use the mean boundary probability as aggregation score instead of more complex GALA-like features. Instead of using the aggregation scores of super-voxel pairs to define merging priority, we propose to use the shift in entropy resulting from merge operations as a merge priority function.

A. Framework

B. Mathematical Notation

In this section, we introduce the mathematical framework used to present our proposed method. A supervoxel over-segmentation map can be represented by an undirected graph $G = \{V, E\}$, called the supervoxel adjacency graph, in which each node $v \in V$ represents a supervoxel and edges $e \in E$ connect adjacent supervoxels $(u, v) \in V \times V$. Greedy agglomeration algorithms iteratively merge pairs of supervoxels: Each iteration of the algorithm produces a new graph $G_t = \{V_t, E_t\}$, in which two nodes of the previous iteration (u_{t-1}, v_{t-1}) have been merged into one so that $|V_t| = |V_{t-1}| - 1$ Starting from an input graph $G_0 = \{V_0, E_0\}$, as computed by the Watershed algorithm, the algorithm outputs a graph $G_T = \{V_T, E_T\}$ after T iterations. The output nodes V_T define the final segmentation output of the agglomeration step. The goal of greedy agglomeration algorithms is to minimize a segmentation metric d between V_T and a ground-truth segmentation S. The next section details the choice of metric d used in our experiments. We denote by $S^*(v)$ the ground-truth label of a supervoxel v.

The aggregation score f defines the probability of two adjacent supervoxels sharing the same ground-truth label. In GALA, f is learned from manually crafted features. Following [10], we use the average boundary probability of voxels located on the boundary of adjacent supervoxels.

A merge priority function defines the order in which edges (*i.e.*; pairs of supervoxels) should be merged by the algorithm. Existing greedy agglomeration algorithms use the aggregation

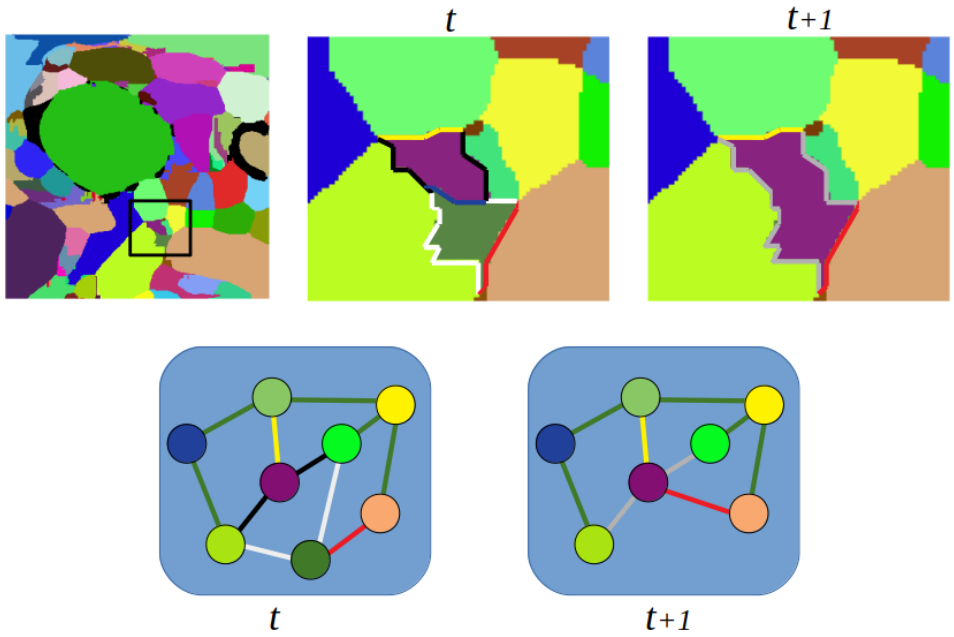

Fig. 2. Illustration of one step of a greedy agglomeration algorithm. On the top, from left to right: one full slice of 3D EM volume, Zoom in on the supervoxel segmentation at time t and Zoom in on the supervoxel segmentation at time $t+1$ after merging two adjacent supervoxels. The red and yellow boundaries remain unchanged after the merge operations. However, the black and white boundaries have been merged into one same boundary (gray) at time $t+1$. The blue boundary does not exist anymore after the merge operation. On the bottom, from left to right: Illustration of the supervoxel adjacency graph at time t and $t+1$

score f as a merge priority function: At each step t, the algorithm finds the edge e of highest aggregation score and merges the supervoxels u,v adjacent to e:

$$e = (u, v) \in E \tag{1a}$$
$$f : E \rightarrow [0, 1] \tag{1b}$$
$$f(e) = p(S^*(u) = S^*(v)) \tag{1c}$$
$$e_t = argmax_{e \in E_t} f(e) \tag{1d}$$

Figure 2 illustrates one step of this greedy agglomeration process: In this example, the blue edge $e_t = (u, v)$ has highest aggregation score $f(e_t)$ at time t. Hence, supervoxels u and v (In green and purple) are merged at time $t+1$. Algorithm 1 formalizes this greedy agglomeration procedure:

Input:
Supervoxel adjacency graph: $G_0 = \{V_0, E_0\}$
Output:
Aggregated adjacency graph: $G_T = \{V_T, E_T\}$:
Init:
Initialize $t = 0$
while $P > 0.5$ **do**
  $e^* = argmax_{e \in E'}(f(e))$
  $G_{t+1} \leftarrow Merge(G_t, e^*)$
  $t \leftarrow t + 1$
end

Algorithm 1: DE Policy algorithm.

The idea we present in this paper is to use a merge function different from the aggregation score f. Instead of merging the most likely pairs of super-voxels, we propose a merge priority function that takes into account the uncertainty resulted by a merge operation in the local neighborhood of the merge operation.

C. Evaluation

The goal of supervoxel agglomeration algorithm is to minimize an error function $d(V_T, S)$ between a ground-truth segmentation S of the neurons and the output segmentation V_T of the algorithm. Several methods have been proposed to evaluate the quality of segmentation in the literature. The Rand index (RI) [12], which evaluates pairs of points in a segmentation, has long been the preferred evaluation metrics.

However, Nunez-Iglesias *et al.* [5] have shown several disadvantages of the RI metrics, such as being sensitive to rescaling and having a limited useful range. Instead, they advocate the use of the variation of information (VI) metric [11]. The VI metric is defined as a sum of the conditional entropies between two segmentations:

$$VI(V_T, S) = H(V_T|S) + H(S|V_T) \tag{2}$$

where V_T is our candidate segmentation and S is our ground truth, and H is the conditional entropy. VI can be intuitively understood as the answer to the question: given the ground truth (S) label of a random voxel, how much more information do we need to determine its label in the candidate segmentation (V_T)? Errors in VI scale linearly in the size of the error, whereas the RI scales quadratically. This makes the VI metric more suitable to study the accuracy of segmentation models in

large volumetric data. Following, Nunez-Iglesias *et al*. [5], we will report our results in terms of VI in the following section.

III. Entropy policy

Existing greedy agglomeration algorithms use the aggregation score as merge priority function: they minimize the risk of merge errors by aggregating the pair of supervoxels that have the highest probability of sharing the same ground-truth label. Instead, we propose to use a merge priority function different from the aggregation score. To motivate our idea, we introduce the notion of adjacency graph entropy $h(G)$. We define $h(G)$ as the entropy of the aggregation scores over the edges V of a supervoxel adjacency graph G:

$$h(G) = -\sum_{e \in E} f(e) \times log(f(e)) \tag{3}$$

The entropy of a graph G measures the uncertainty of the candidate merge operations in G. A graph G with low entropy corresponds to a graph with relatively easy to assess merge decisions, while a graph G with high entropy contains many uncertain candidate merge operations. Hence, greedily aggregating supervoxels of a high entropy graph G is likely to result in many merge errors and poor segmentation results. Low entropy graph G are likely to yield fewer merge errors, hence better segmentation results.

One side-effect of a merge operation is to change the connectivity patterns between supervoxels of the adjacency graph G. Figure 2 (bottom) illustrates the changes in G_{t+1} after merging edge e_t. As illustrated in this figure, the black and white edges of G_t are merged into gray edges in graph G_{t+1}. Hence merging operations impact the entropy of the adjacency graph (Equation 3.) as they modify the graph connectivity E. We write $\delta h(e_t)$ the variation of the adjacency graph entropy resulting by merging the supervoxels adjacent through $e_t \in E_t$:

$$\delta h(e_t) = h(G_{t+1}) - h(G_t) \tag{4}$$

Ideally, we would like merge operations at time t to reduce the entropy of the adjacency graph at time $t+1$, i.e., we would like to merge edges with minimum entropy delta:

$$e_t = argmin_{e \in E_t} \delta h(e) \tag{5}$$

However, an edge e_t may have low entropy delta but low aggregation score. Although merging such edges would ease subsequent merging operations by reducing the graph entropy, it is very likely to produce merge errors. Hence, a good merge policy should merge edges with both high aggregation scores and low entropy delta. This implies a trade-off between entropy delta and aggregation scores in the aggregation procedure. In the following subsections, we propose two different implementations of such a trade-off.

A. Lambda-Entropy policy

The lambda-entropy (LE) policy defines a merge priority function g as follows:

$$g : E \to \mathbb{R} \tag{6a}$$
$$\lambda \in [0, 1] \tag{6b}$$
$$g(e) = (1 - \lambda) \times f(e) - \lambda \times \delta h(e) \tag{6c}$$
$$e_t = argmax_{e \in E_t} g(e) \tag{6d}$$

The LE policy trades off entropy delta and aggregation score by introducing a weighting factor λ, which balances the importance given to entropy delta relatively to the aggregation score. The aggregation procedure can be performed following Algorithm 1, using the merge priority function g instead of f.

B. Delta-Entropy policy

Different from the LE policy, the delta-entropy (DE) policy directly uses the entropy delta as merge priority function. To prevent from merging edges of low aggregation score, the DE policy only merges edges e_t with aggregation scores above a given threshold P_t. Starting with a high threshold P_0, the DE policy merges all edges e with aggregation scores higher than P_0. Once all such edges have been merged, a lower threshold $P_1 = P_0 - \Delta$ is set to allow additional edges of lower aggregation score to be merged. The DE policy proceeds by iteratively decrementing the aggregation threshold until a stop value. The DE policy is fully parameterized by Δ, and is formalized in Algorithm 2.

Input:
Supervoxel adjacency graph: $G_0 = \{V_0, E_0\}$
Parameter Δ
Output:
Aggregated adjacency graph: $G_T = \{V_T, E_T\}$:
Init:
Initialize $P_0 = 1 - \Delta$
Initialize $t = 0$
while $P_t > 0.5$ **do**
  $E' = \{e_t | e_t \in E_t, f(e_t) > P_t\}$
  if $|E'| > 0$ **then**
    $e^* = argmin_{e \in E'}(\delta h(e))$
    $G_{t+1} \leftarrow Merge(G_t, e^*)$
    $P_{t+1} \leftarrow P_t$
  else
    $P_{t+1} \leftarrow P_t - \Delta$
  end
  $t \leftarrow t + 1$
end

Algorithm 2: DE Policy algorithm.

IV. Experiment and results

We evaluate our proposed method on the SNEMI3D dataset [13]. The SNEMI3D dataset consists of a $1024 \times 1024 \times 100$ voxels EM volume, fully annotated with individual neuron labels. We use a sub-volume of $320 \times 320 \times 100$ voxels as

validation set to evaluate our different agglomeration policy and the remainder of the dataset as training data. We train a 3D-Unet as proposed in [10] on the training set to predict cell boundaries, and use a variant of the Watershed algorithm, as proposed in [14], to over-segment the boundary map into supervoxels. As suggested in [10], we use the mean boundary probability as aggregation score, and evaluate the VI score obtained by greedy agglomeration using different merge priority functions.

As a baseline, we use the greedy a score policy used by previous works [4], [5], [10]. This policy iteratively merges pairs of super-voxel in decreasing order of aggregation scores. We compare the results obtained by our proposed policies to this baseline.

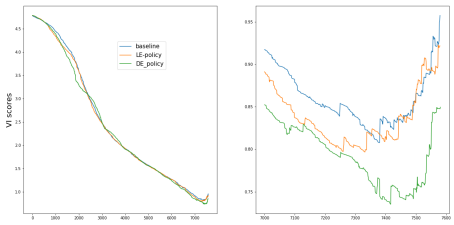

Fig. 3. Evolution of VI scores along the aggregation process using different merge policy (left) throughout the full agglomeration process, and (right) zoomed in on the last few steps of the agglomeration process.

Figure 3 shows the evolution of the VI score along the iterative agglomeration process following different policies. A lower VI score means a more accurate segmentation. The baseline aggregation policy scores a minimum VI of 0.81. In comparison, the best VI scores obtained by the *DE* and *LE* policy are respectively 0.73 and 0.79.

Figure 4 illustrates the dynamics of each policy along each step of the agglomeration process. This figure shows the evolution of the aggregation score f and the entropy delta δh at each iteration of the merge process. The baseline policy iteratively merges supervoxels in order of descending aggregation scores. Hence, the aggregation score smoothly decreases with each iteration step. The LE-policy trades off entropy delta and aggregation score with a λ weighting factor. Hence the aggregation score continually decreases with variations due to the different entropy delta δh of supervoxel pairs. The DE-policy proceeds by first merging all supervoxel pairs with aggregation score above a threshold $P_0 = 95\%$ in increasing order of entropy delta. It then iteratively lower the threshold P_t as all supervoxel pairs above the current threshold have been merged. Hence, the aggregation score decreases sharply between each threshold values, while the entropy delta smoothly increases at each step.

Finally, Table 1 and 2 show the final VI scores yielded by the DE-policy and LE-policy, respectively, for different hyper-parameters. The results presented in Figure 3 and 4 were computed with the optimal hyper-parameter choice: $\lambda = 0.3$ and $\Delta = 0.05$. The DE policy, with sufficiently low Δ seems

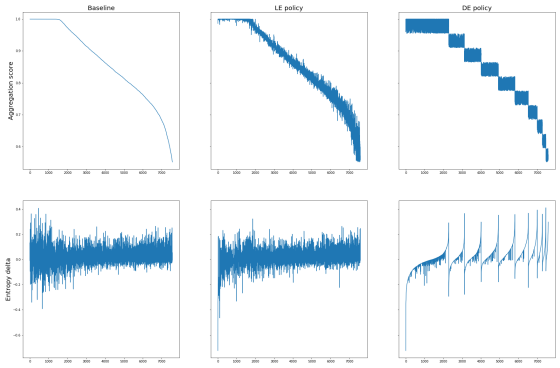

Fig. 4. Illustration of the different merge policy dynamics along the aggregation process. Top: Evolution of the aggregation score f. Bottom: Evolution of the entropy delta δh From left to right, the different plots show results for the baseline aggregation policy, DE-policy and LE-policy.

to perform the best. We recommend using the DE-policy with small Δ in the range of 0.05 to 0.1.

TABLE I
VI SCORES OF DE POLICY AGGREGATION FOR DIFFERENT Δ.

Δ	VI
10^{-6}	0.81
0.05	0.73
0.075	0.79
0.1	0.76
0.16	0.81
0.25	0.97
0.5	2.51

TABLE II
VI SCORES OF LE POLICY AGGREGATION FOR DIFFERENT λ.

λ	VI
0	0.81
0.1	0.81
0.2	0.80
0.3	0.79
0.4	0.81
0.5	0.84
0.6	0.92
0.8	1.16
1	2.51

V. CONCLUSION

Precise neurite segmentation is a major task in connectomics. The ability to quickly and accurately segment individual neurons from large-scale EM images is primordial to shed some light into the mysteries of the brain. A critical step in state-of-the-art segmentation pipelines is the aggregation of oversegmented super-pixels. In this paper, we proposed a simple entropy policy for the greedy agglomeration of supervoxels. Our policy boosts the accuracy of greedy agglomeration algorithms, is easy to implement, and comes with negligible

computational cost, so that it can be easily integrated within any existing greedy agglomeration framework.

REFERENCES

[1] Viren Jain, Srinivas C. Turaga, K Briggman, Moritz N. Helmstaedter, Winfried Denk, and H. S. Seung.Learning to Agglomerate Superpixel Hierarchies. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira,and K. Q. Weinberger, editors,Advances in Neural Information Processing Systems 24, pages 648656.Curran Associates, Inc., 2011.

[2] John A Bogovic, Gary B Huang, and Viren Jain. Learned versus handdesigned feature representations for3d agglomeration.arXiv preprint arXiv:1312.6159, 2013.

[3] Maitin-Shepard, Jeremy B., et al. "Combinatorial energy learning for image segmentation." Advances in Neural Information Processing Systems. 2016.

[4] Nunez-Iglesias, Juan, et al. "Graph-based active learning of agglomeration (GALA): a Python library to segment 2D and 3D neuroimages." Frontiers in neuroinformatics 8 (2014): 34.

[5] Nunez-Iglesias J, Kennedy R, Parag T, Shi J, Chklovskii DB (2013) Machine Learning of Hierarchical Clustering to Segment 2D and 3D Images.

[6] Andres, Bjoern, et al. "Globally optimal closed-surface segmentation for connectomics." European Conference on Computer Vision. Springer, Berlin, Heidelberg, 2012.

[7] Keuper, Margret, et al. "Efficient decomposition of image and mesh graphs by lifted multicuts." Proceedings of the IEEE International Conference on Computer Vision. 2015.

[8] Beier, Thorsten, et al. "Multicut brings automated neurite segmentation closer to human performance." Nature Methods 14.2 (2017): 101.

[9] Vincent, Luc, and Pierre Soille. "Watersheds in digital spaces: an efficient algorithm based on immersion simulations." IEEE Transactions on Pattern Analysis & Machine Intelligence 6 (1991): 583-598.

[10] Lee, Kisuk, et al. "Superhuman accuracy on the SNEMI3D connectomics challenge." arXiv preprint arXiv:1706.00120 (2017).

[11] Meil, Marina. "Comparing clusterings by the variation of information." Learning theory and kernel machines. Springer, Berlin, Heidelberg, 2003. 173-187..

[12] Rand WM (1971) Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66 846850: 6.

[13] Kasthuri, Narayanan, et al. "Saturated reconstruction of a volume of neocortex." Cell 162.3 (2015): 648-661.

[14] Aleksandar Zlateski and H. Sebastian Seung. Image Segmentation by Size-Dependent Single LinkageClustering of a Watershed Basin Graph.CoRR, abs/1505.00249, 2015.