

Diverse Multiple Prediction on Neuron Image Reconstruction

Ze Ye¹, Cong Chen², Changhe Yuan², and Chao Chen¹

¹ Stony Brook University, Stony Brook, NY 11794
{ze.ye, chao.chen.1}@stonybrook.edu

² The City University of New York, New York, NY
{cong.chen, changhe.yuan}@qc.cuny.edu

Abstract. Neuron reconstruction from anisotropic 3D Electron Microscopy (EM) images is a challenging problem. One often considers an input image as a stack of 2D image slices, and consider both intra and inter slice segments information. In this paper, we present a new segmentation algorithm which builds a unified energy function and jointly optimize the per-slice segmentation and the inter-slice consistency. To find an optimal solution from the huge solution space, we propose a novel diverse multiple prediction method which also encourages diversity in partial solutions. We demonstrate the strength of our method in several public datasets.

1 Introduction

Thanks to recent technology breakthrough, the new serial-section Electron Microscope is able to collect high resolution neuron images from insects or animals' nerve tissue. Such *neuron images* contain detailed geometric information of the densely packed neuronal structures, e.g., dendrites and axons. Reconstructing these 3D neuronal structures helps neuroscientists characterizing different types of neurons and analyzing how they collaborate to achieve different functionalities.

A majority of publicly available neuron images are *anisotropic*, i.e., having fine resolution in two spatial dimensions, but a coarse resolution in the third dimension, e.g., $4 \times 4 \times 50 \text{ nm}^3$ [1] or $3 \times 3 \times 30 \text{ nm}^3$ resolution [2]. An automatic reconstruction algorithm often considers such an anisotropic image as a stack of high-resolution 2D slices, and takes a two-stage approach. First, it partitions, or say segments, each 2D slice into regions corresponding to different neuronal structures. Second, regions of consecutive slices are linked to each other based on their geometric consistency to form 3D structures. See Figure 1 for an illustration.

To segment each 2D slice into neuronal regions, one needs to identify the boundaries of the neurons, i.e., membranes. Different classifiers, such as random forests or Convolutional Neural Networks (CNNs), have been trained to classify each pixel as either belonging to the membranes or not [4, 5, 11]. While for 3D linking, one predicts the links between regions of consecutive slices based on their geometric information. Various methods have been proposed to solve this 3D linking problem effectively [8, 9, 13].

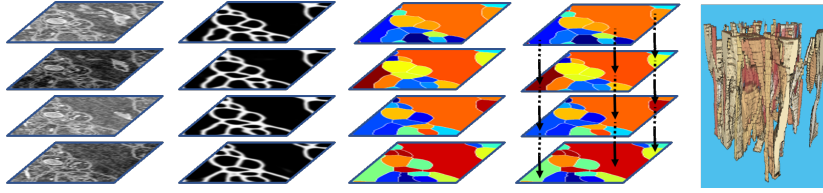


Fig. 1. From Left to right, raw images, likelihood map, segmentation of each slices, linked segmented images, reconstructed neurons.

A common issue of these existing approaches is the loose coupling of the two stages. Algorithms for the first stage only use the limited 2D observations, and thus fail to fully leverage the 3D information. In this paper, we propose a new method to jointly segment all 2D slices while considering their geometric consistency. We formulate the problem as a simple chain graphical model problem with huge state-space, i.e., all possible segmentations for each slice. To solve the problem, we progressively find segmentation candidates of each slice conditioned on those of previous slices.

At each slice, we maintain top B candidate segmentations to increase the chance of finding a better solution. The huge state-space makes it challenging to find high quality solution even with top B solutions per slice. We propose a novel *diverse multiple prediction* method to explore the top B *diverse* solutions as candidates. Underlying the algorithm is another graphical model so that the prediction of top B diverse solutions is tractable.

Instead of complicated geometric constraints [8, 9, 13], we use a computation friendly surrogate scoring function, namely, the Rand Index (RI), to model the geometric consistency between segmentations of consecutive slices. The RI effectively penalizes topological changes such as merge/split, while being robust to shifting. Thanks to the decomposability of RI, we turn each prediction step, i.e., finding the top B diverse segmentations of each slice, into another Markov random field problem and thus can solve it efficiently.

In summary, our contribution is twofold.

- Propose a simple chain graphical model for the reconstruction problem. The unified principled framework is easier to tune and to generalize.
- Propose a *diverse multiple prediction* algorithm to solve the inference problem.

It efficiently searches the solution space and finds a high-quality solution.

Experiments on several public datasets prove the advantage of our method. Further ablation studies show the benefits of different technical contributions.

2 Method

We create a unified chain graphical model to take both 2D slice information and 3D geometric consistency into consideration. Given an anisotropic neuron image with M slices, we treat each slice as a node and each pair of adjacent slices as an edge. Denote by S the space of all possible segmentations and $s^j \in S$ a segmentation

of slice j . We search for a joint segmentation of the whole image stack $\mathbf{s} = (s^1, s^2, \dots, s^{M-1}, s^M) \in S^M$ given the observation $\mathbf{x} = (x^1, x^2, \dots, x^{M-1}, x^M)$. Using the chain conditional random field (CRF) formulation, we can find the best segmentation sequence by computing the maximum a posteriori (MAP):

$$\operatorname{argmax}_{\mathbf{s} \in S^M} P(\mathbf{s}|\mathbf{x}) = \operatorname{argmin}_{\mathbf{s} \in S^M} \exp(-E(\mathbf{s}; \mathbf{x}))/Z(\mathbf{x}), \quad (2.1)$$

$$E(\mathbf{s}; \mathbf{x}) = \sum_{j=1}^M E_j(s^j; x^j) + \sum_{j=1}^{M-1} E_{j,j+1}(s^j, s^{j+1}), \quad (2.2)$$

where $Z(\mathbf{x}) = \sum_{\mathbf{s} \in S^M} \exp(-E(\mathbf{s}; \mathbf{x}))$ is the partition function. The energy $E(\mathbf{s}; \mathbf{x})$ is a sum of unary and binary energy terms: $E_j(s^j; x^j)$ measures how likely slice j has segmentation s^j given x^j . $E_{j,j+1}(s^j, s^{j+1})$ measures the geometric similarity between segmentations of slices j and $j+1$. Note that there is no observation term in binary term since the geometric similarity is independent to observation $\mathbf{x}$. For convenience, we drop the observation term $\mathbf{x}$ or x^j when referring to energy term $E_j(s^j; x^j)$ for the rest of the paper.

Energy terms. We adapt the tree-structured CRF [12] to express the unary term in Equation (2.2): $E_j(s^j) = E_j(y^j) = \sum_{(u,v) \in \mathcal{E}} \theta_{u,v}(y_u^j, y_v^j) + \sum_{u \in \mathcal{V}} \theta_u(y_u^j)$. Here $\mathcal{V}$ and $\mathcal{E}$ are the node set and edge set of a *hierarchical merging tree*, i.e., a tree constructed by running a watershed region merging algorithm over the likelihood map trained on CNNs [5]. Nodes of the tree correspond to regions in the image. Edges decide how different regions are merged into larger ones during the watershed merging process. We use θ to express the energy term of each node u or edge (u, v) while the value of y indicates whether the corresponding node is a correct segment. See Figure 2 for an illustration. Note each tree-structured CRF is constructed based on a given likelihood of slice j , while the parameters in the model is trained over all training slices. The tree-structured CRF provides an energy for each segmentation s^j and is treated as the unary energy term of our overall chain CRF (Equation (2.2)).

We use RI to represent binary term: $E_{j,j+1}(s^j, s^{j+1}) = -\eta RI(s^j, s^{j+1})$. The RI can explicitly enforce geometric consistency between two consecutive segmentations and it can be calculated very efficiently. Here the η is the weight of the binary term (inter-slice weight) that indicates how strong the model seeks to enforce inter-slice geometry similarity.

Inference. Equation (2.1) has a huge solution space, S^M ; it is exponential to the number of slices, M , and the space of segmentations S is exponential to the size of each slice. Finding the global optimal solution from such huge solution space is infeasible. Instead, we propose an efficient prediction algorithm to find a high quality solution. We first propose to use a multiple prediction method to solve the problem (Section 2.1). However, during the multiple prediction, only a limited number of partial solutions are maintained. To further improve the quality of these partial solutions, and thus the final prediction, we propose a novel *diverse multiple prediction* method (Section 2.2). In each prediction step, the diverse multiple prediction finds the top solutions that are not only high in scores, but also *diverse*.

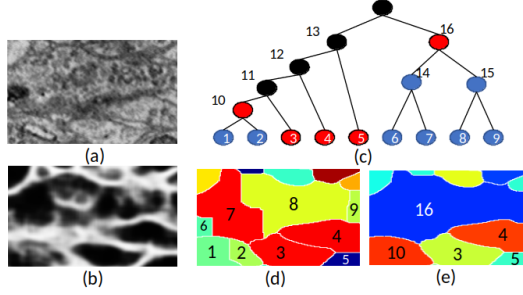


Fig. 2. (a) Raw image patch. (b) Membrane probability map. (c) Merging tree which black nodes are undersegment and blue nodes are oversegment. (d) Nine segments corresponded to leaf nodes. (e) Final segmentation.

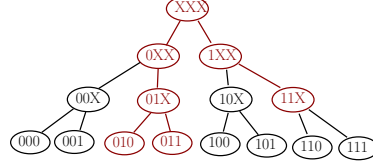


Fig. 3. A search tree for a 3-node chain graph, with each node having two possible values (0 and 1). An example beam search with $B = 2$ is highlighted with red.

2.1 Prediction Algorithms

We iteratively determine the values $s^1, s^2, \dots, s^M$. At each step, we find preferred s^j given solutions of previous nodes $s^1, \dots, s^{j-1}$. To introduce our final algorithm in the next section, we first propose a few baseline multiple prediction methods.

Rewrite Equation (2.2) as $E(\mathbf{s}) = E_1(s^1) + \sum_{j=2}^M \Psi_{j-1,j}(s^{j-1}, s^j; x^j)$, where $\Psi_{j-1,j}(s^{j-1}, s^j; x^j) = E_{j-1,j}(s^{j-1}, s^j) + E_j(s^j; x^j)$. To compute the optimal segmentation $\hat{\mathbf{s}} = (\hat{s}^1, \dots, \hat{s}^M)$, a straightforward algorithm, called the *greedy search (GRE)*, starts by finding the s^1 optimizing E_1 , i.e., $\hat{s}^1 = \operatorname{argmin}_{s^1} E_1(s^1; x^1)$. Next, it iteratively finds the optimal s^j conditioned on $\hat{s}^{j-1}$ that has already been determined, i.e., $\hat{s}^j = \operatorname{argmin}_{s^j} \Psi_{j-1,j}(\hat{s}^{j-1}, s^j; x^j)$. Figure 3 shows a *search tree* in which each internal *search node* corresponds to a partial solution and each leaf search node corresponds to a complete solution. The greedy algorithm finds a leaf search node using a greedy method: starts from the root. At each internal search node, it finds the best child and continues, until a leaf is reached.

Multiple prediction. The idea of multiple prediction is to keep track of $B > 1$ many top partial solutions until the last level. Note this is the same as the well-known beam search method. We start by finding the top B solutions of $E_1(s^1; x^1)$. At the j -th iteration, using partial solutions from the $(j-1)$ -th iteration, we find the top B partial solutions with length j . For each partial solution of length $j-1$, $(\hat{s}^1, \dots, \hat{s}^{j-1})$, we extend it to B different partial solutions with length j . In particular, we compute the top B candidate segmentations for slice j conditioned on the previous ones, namely,

$$\hat{s}_1^j, \dots, \hat{s}_B^j = \operatorname{argmin}_{s_1^j, \dots, s_B^j: s_b^j \neq s_{b'}^j, \forall b \neq b'} \sum_{b=1}^B \Psi_{j-1,j}(\hat{s}^{j-1}, s_b^j; x^j). \quad (2.3)$$

We then get B partial solutions by appending each of them to the end of the existing partial solution, namely, $\{(\hat{s}^1, \dots, \hat{s}^{j-1}, \hat{s}_b^j) \mid b = 1, \dots, B\}$. We compare all of them and keep the B with the minimal energy evaluated on

$E_1, \Psi_{1,2}, \dots, \Psi_{j-1,j}$. At the last iteration, when $j = M$, we select the best solution. Figure 3 illustrates the beam search with $B = 2$ on the search tree.

2.2 Diverse Multiple Prediction

Although efficient, aforementioned prediction algorithms are not necessarily optimal. As we could only keep B partial solutions at each step, the multiple prediction algorithm may end up with a suboptimal solution. To address this issue, we propose to enforce that the partial solutions are not only low in energy, but also high in *diversity*. By enforcing diversity, the partial solutions have a better chance to include a high-quality one.

We propose a new method: *diverse multiple prediction*. At each iteration, instead of the top B solution, we compute the top B solution that are sufficiently diverse. We adapt the *m-diverse best* formulation by Batra *et al.* [3]. First, we find the best solution, then the second best which is at least δ away from the best. The b -th solution is the best with at least δ dissimilarity from the previously $b - 1$ chosen solutions. Using the Hamming distance measure between two segmentations $\text{dist}(\cdot, \cdot)$, the b -th solution is:

$$\hat{s}_b^j = \operatorname{argmin}_{s^j \in S} \text{cost}(s^j) \quad \text{s.t. } \text{dist}(s^j, \hat{s}_{b'}^j) \geq \delta, \forall b' \in \{1, \dots, b-1\} \quad (2.4)$$

Here the cost function $\text{cost}(s^j)$ is $E_1(s^1)$ when $j = 1$ and $\Psi_{j-1,j}(\hat{s}^{j-1}, s^j)$ otherwise. The problem can be reduced into a sequence of B many MAP inference tasks [3]. A hyperparameter λ tunes the weights of diversity.

It is essential that the cost function can be written as a tractable graphical model, so that its MAP inference can be computed efficiently. Using Rand Index as binary term, we can rewrite the cost function $\Psi_{j-1,j}$ as a tree-structured graphical model energy. Therefore, the m -diverse-best solutions can be solved efficiently. For convenience, we denote by the top B diverse solution as $\text{mdiv}(E_1; x^j)$ or $\text{mdiv}(\Psi_{j-1,j}; x^j, \hat{s}^{j-1})$ depending on the relevant energy term. The pseudocode of our diverse multiple prediction algorithm is given in Algorithm 1.

3 Experiments

In this section, we validate our proposed method using public datasets. We also use ablation study to closely inspect the behavior of our prediction method.

Datasets. We use two public datasets: the *Drosophila* first instar larva ventral nerve cord (VNC) [1] and mouse cortex (MOU) [2]. Both datasets were used as public challenges. So their test sets do not have ground truth annotation. We use their training sets for evaluation. VNC contains 30 consecutive slices for training, each of size 512x512. The resolution is 4x4x50 nm³. We use a three-fold cross validation and report the mean performance over the validation sets. Due to the nature of the problem, the validation set needs to be consecutive slices rather than random slices. For fold 1, we use slices 1-10 for validation and 11-30 for training. Similarly, for folds 2 and 3, we use 11-20 and 21-30 to validate,

Algorithm 1 Diverse Multiple Prediction

Require: Input image $\mathbf{x}$ **Ensure:** Segmentation $\mathbf{s}$

- 1: $(\hat{s}_1^1, \dots, \hat{s}_B^1) \leftarrow \text{mdiv}(E_1; x^j)$ $\triangleright$ The top B diverse solutions for slice # 1.
 - 2: $\mathcal{S} \leftarrow \{(\hat{s}_1^1), \dots, (\hat{s}_B^1)\}$ $\triangleright$ The top B partial solutions with length 1.
 - 3: **for** $j = 2$ **to** M **do**
 - 4: $\mathcal{S}' \leftarrow \emptyset$
 - 5: **for** $(\hat{s}_1^{j-1}, \dots, \hat{s}_B^{j-1}) \in \mathcal{S}$ **do**
 - 6: $(\hat{s}_1^j, \dots, \hat{s}_B^j) \leftarrow \text{mdiv}(\Psi_{j-1,j}; x^j, \hat{s}^{j-1})$ $\triangleright$ For each length $j-1$ partial solution in $\mathcal{S}$, compute the top B diverse solution conditioned on $\hat{s}^{j-1}$ and x^j .
 - 7: **for** $b = 1$ **to** B **do**
 - 8: $\mathcal{S}' \leftarrow \mathcal{S}' \cup \{(\hat{s}_1^j, \dots, \hat{s}_B^j)\}$ $\triangleright$ Add B length j partial solutions into $\mathcal{S}'$.
 - 9: **end for**
 - 10: **end for**
 - 11: Resize $\mathcal{S}'$. Only keep top B partial solutions in $\mathcal{S}'$ with the minimal partial energy $E_1(\hat{s}^1; x^1) + \sum_{j'=2}^j \Psi(\hat{s}^{j'-1}, \hat{s}^{j'}; x^{j'})$.
 - 12: $\mathcal{S} \leftarrow \mathcal{S}'$
 - 13: **end for**
 - 14: **return** The solution $\hat{\mathbf{s}} \in \mathcal{S}$ with the minimal energy $E(\mathbf{s}; \mathbf{x})$.
-

respectively. MOU has 100 consecutive training slices, each with size 1024x1024 (resolution 3x3x30 nm³). We use a three-fold cross validation and report the mean performance, using slices 1-33, 34-66, and 67-100 as validation sets, respectively.

Table 1. ARE of all methods on the two datasets.

	DMP	MP	GRE	CRF	LIU
VNC	 0.0172 	0.0191	0.0197	0.0235	0.0244
MOU	 0.0271 	0.0275	0.0279	0.037	0.0375
MOU-3D	 0.0863 	0.0894	0.090	0.1147	0.1151

Baselines. We compare our method, diverse multiple prediction (DMP), with two state-of-the-arts. LIU [10] prunes the hierarchical merging tree in a greedy manner. CRF [12] constructs and trains tree-structured CRF on hierarchical merging trees, but ignores inter-slice geometric constraints. We also compare with two baselines that have been introduced in this paper: greedy search (GRE) and multiple prediction (MP). Note we focus on the segmentation task, so we do not compare with various methods on computing better boundary likelihood maps [4–7]. Instead, we assume a same boundary likelihood map for all methods. We use the boundary likelihood map from [4] for VNC, and the one from [5] for MOU. The hyperparameters of all methods are fine-tuned to ensure fairness.

For our own method, we choose the inter-slice weight $\eta = 600$ and set the upper bound of solution candidate $B = 6$. For diverse multiple prediction method, we set the diversity weight $\lambda = 0.1$.

Experiment setting. All compared methods output joint segmentations of all slices. We directly evaluate the performance by comparing the 2D segmentation with ground truth annotation over all slices and aggregate. The goal is to show that using inter-slice geometric consistency and advanced search strategy, we can improve the per-slice segmentation quality. We report the results on both VNC and MOU. Furthermore, since the segmentations can be linked together to reconstruct the 3D neurons (Figure 1), we may evaluate all the methods by applying a common 3D linking tool [9] and compare the 3D reconstruction results. We report the results only on MOU (called MOU-3D), as such 3D reconstruction ground truth annotation is not available for VNC.

We measure the performance using the standard Adaptive Rand Error (ARE) score, which ranges between 0 and 1 (the smaller the better).³ The results of all methods over VNC, MOU, and MOU 3D linking are reported in Table 1. A qualitative example is also shown in Figure 4. Highlighted areas are where the difference occurs. We observe that MP improves upon GRE, while DMP achieves even better results.

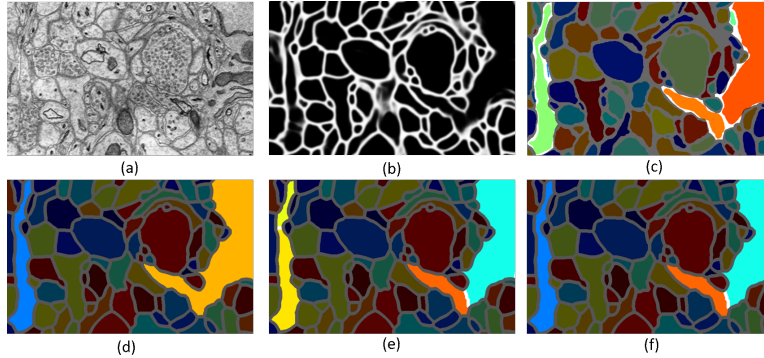


Fig. 4. (a) Raw image. (b) Membrane probability map. (c) Ground truth. (d) GRE. (e) MP. (f) DMP.

Ablation study. To further investigate the behavior of our method, we carry out the ablation study on three hyperparameters: number of partial solutions to memorize B , inter-slice energy weight η , and diversity weight λ on VNC dataset [1]. Figure 5(a) shows the ARE scores of three proposed methods with regard to the B values. Note that all three methods are equivalent when $B = 1$. DMP converges to the best score even with a small B . MP needs a larger $B = 6$.

³ ARE is closely related to Rand index but is the corrected-for-chance version, and thus is not decomposable like RI.

The second parameter we study is the inter-slice weight η . Figure 5(b) shows those three ARE scores with η ranged from 0 to 1200. Note that all three methods are equivalent to CRF when $\eta = 0$ since inter-slice consistency is no longer in effect and three methods eventually generate local best solution in each slice.

The last parameter is λ which measures the diversity in our diverse multiple prediction method. Figure 5(c) shows the performance of our method (DMP) with regard to λ .

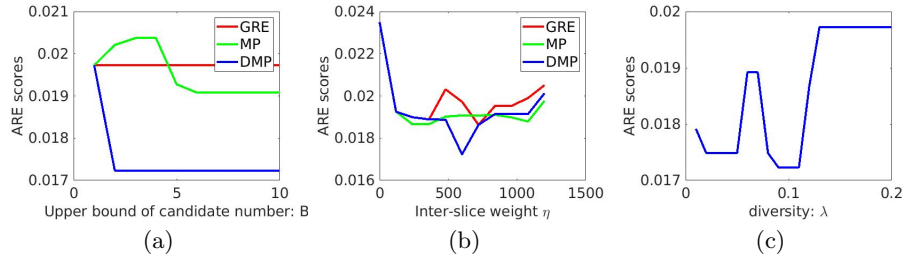


Fig. 5. Ablation study on three parameters: B , η and λ .

Acknowledgements. This work was partially supported by NSF IIS-1855759, CCF-1855760 and IIS-1829560.

References

1. Arganda-Carreras, I., Seung, H., Cardona, A., Schindelin, J.: Segmentation of neuronal structures in em stacks challenge-isbi 2012 (2012)
2. Arganda-Carreras, I., Seung, H., Vishwanathan, A., Berger, D.: 3d segmentation of neurites in em images challenge-isbi 2013 (2013)
3. Batra, D., Yadollahpour, P., Guzman-Rivera, A., Shakhnarovich, G.: Diverse m-best solutions in markov random fields. In: European Conference on Computer Vision. pp. 1–16. Springer (2012)
4. Ciresan, D., Giusti, A., Gambardella, L.M., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: Advances in neural information processing systems. pp. 2843–2851 (2012)
5. Fakhry, A., Peng, H., Ji, S.: Deep models for brain em image segmentation: novel insights and improved performance. *Bioinformatics* **32**(15), 2352–2358 (2016)
6. Fakhry, A., Zeng, T., Ji, S.: Residual deconvolutional networks for brain electron microscopy image segmentation. *IEEE transactions on medical imaging* **36**(2), 447–456 (2017)
7. Funke, J., Tschopp, F.D., Grisaitis, W., Sheridan, A., Singh, C., Saalfeld, S., Turaga, S.C.: Large scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence* (2018)
8. Kaynig, V., Fuchs, T.J., Buhmann, J.M.: Geometrical consistent 3d tracing of neuronal processes in sstem data. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 209–216. Springer (2010)

9. Liu, T., Jones, C., Seyedhosseini, M., Tasdizen, T.: A modular hierarchical approach to 3d electron microscopy image segmentation. *Journal of neuroscience methods* **226**, 88–102 (2014)
10. Liu, T., Seyedhosseini, M., Ellisman, M., Tasdizen, T.: Watershed merge forest classification for electron microscopy image stack segmentation. In: *Proceedings/IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision*. vol. 2013, p. 4069. NIH Public Access (2013)
11. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical image computing and computer-assisted intervention*. pp. 234–241. Springer (2015)
12. Uzunbas, M.G., Chen, C., Metaxas, D.: An efficient conditional random field approach for automatic and interactive neuron segmentation. *Medical image analysis* **27**, 31–44 (2016)
13. Vitaladevuni, S.N., Basri, R.: Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. pp. 2203–2210. IEEE (2010)