Efficient Correction for EM Connectomics with Skeletal Representation

Konstantin Dmitriev[12]
kdmitriev@cs.stonybrook.edu
Toufiq Parag[2]
paragt@seas.harvard.edu
Brian Matejek[2]
bmatejek@seas.harvard.edu
Arie E. Kaufman[12]
ari@cs.stonybrook.edu
Hanspeter Pfister[2]
pfister@seas.harvard.edu

[1] Stony Brook University
Stony Brook, NY, USA
[2] Harvard University
Cambridge, MA, USA

Abstract

Machine vision techniques for automatic neuron reconstruction from electron microscopy (EM) volumes have made tremendous advances in recent years. Nonetheless, large-scale reconstruction from teravoxels of EM volumes retains both under- and over-segmentation errors. In this paper, we present an efficient correction algorithm for EM neuron reconstruction. Each region in a 3D segmentation is represented by its skeleton. We employ deep convolutional networks to detect and correct false merge and split errors at the joints and endpoints of the skeletal representation. Our algorithm can achieve the same or close accuracy of the state-of-the-art error correction algorithm by querying only at a tiny fraction of the volume. A reduction of the search space by several orders of magnitude enables our approach to be scalable for terabyte or petabyte scale neuron reconstruction.

1 Introduction

Exhaustive knowledge of neuron wiring diagram offers valuable information to neuroscience [10, 29, 30]. Capitalizing on machine vision/learning techniques, EM connectomics seeks to uncover the biological neural network of animal brain from electron microscopy (EM) images. Consequently, the computational challenges in EM connectomics received steady attention of the computer vision community over the past ten years [1, 2, 4, 5, 6, 17, 17, 19, 22, 23, 26, 31, 32]. One critical computer vision task in EM connectomics is segmentation of 3D neuron shapes from large EM volumes.

Recent breakthroughs in the vision algorithms for 3D volumetric segmentations [13, 18] resulted in unprecedented accuracy in neural reconstruction. However, at a large scale, multiple iterations of an extremely costly flood filling algorithm [13] over terabyte size volume

still leaves behind many errors in the output segmentation [14]. Lee et al. [18] have also reported errors residues that people can readily identify, and, makes an important observation that it is more difficult for a human expert to find errors based on the inspection of EM images alone without the access to a 3D rendering. This insight alludes to a cause for the failure of existing segmentation methods that is more fundamental than algorithmic limitations.

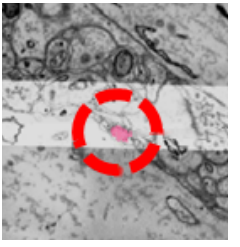


Figure 1: A 2D section of EM data, where two segments (rendered in magenta and green in Figure 2(a)) were disconnected due to an artifact.

The structure and organization of neurons in animal brain are extremely irregular and intricate. The shape of a neuron cell exhibits random and wide variation along its structure that often shrinks into an extremely narrow tubular shape. Such narrow parts of a cell are located in congested areas surrounded by other cells. When imaged by the EM apparatus, these areas cause significant ambiguity – even for human experts – to interpret.

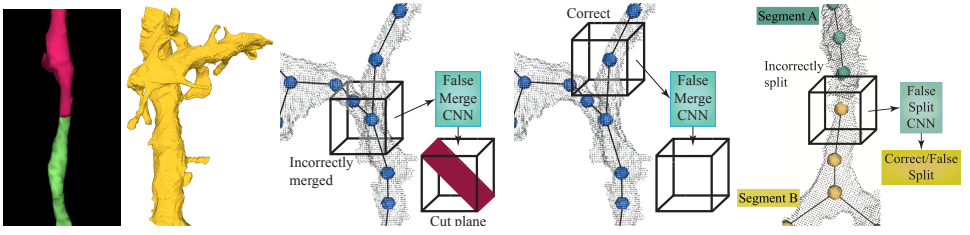
In addition, the generation of EM images of the animal brain tissue is a complex procedure comprising multiple consecutive stages of sample preparation techniques, section cutting mechanisms, and finally imaging the sections with an electron microscope. Despite many recent advances made in each of these stages, they are far from being perfect, and many of these tasks are still performed manually [8, 15]. As a result, the EM images of brain tissue contain a significant amount of artifacts of various kinds, often leading to severe distortion of the image.

The artifacts on EM images, in combination with the inherent ambiguity of neural structure and organization, lead to serious challenges to the voxelwise prediction, as well as subsequent agglomeration methods, relying only on EM image characteristics (intensity values and learned features of them). In Figure 1, we show one 2D section of a plane of EM volume corrupted by an artifact (a white horizontal strip in the middle); the segment (magenta) of a narrow neuron cell that was disconnected is circled in red (dashed line).

Similar to human cognition, we believe it is difficult for an algorithm to infer the intricate structure of neurons exclusively from EM images that are potentially corrupted by the EM sample preparation and imaging. Some 3D representation of the segmented regions is essential to identify true neural morphology at challenging locations (as is the case for human experts). As an example, a 3D rendering in Figure 2(a) effectively resolves the ambiguity in joining the two incorrectly split segments on the 2D section of Figure 1 – the magenta region in Figure 2(a) corresponds to the segment circled in red in Figure 1. Similarly, a 3D view of an incorrectly merged segment in Figure 2(b) provides strong evidence that the vertical and horizontal branches of this segment should be separated.

We propose an efficient error correction of EM segmentation output that exploits the structural information of the 3D shapes. Each segment in an input segmentation is represented by a connected set of lines, henceforth called the skeleton, that emulates the segment centerline to capture as much morphological information as possible. Ideally, each split error should be identifiable at the terminal points of a skeleton. An erroneous merge of two cells should also be manifested in the skeleton as a new branch.

Our method exploits the skeleton joints (endpoints of skeleton lines) to detect and correct segmentation errors. We employ a two-stage deep CNN based error correction scheme first to fix the under-segmentation error and then repair the false splits in the merge corrected volume. Utilizing the skeleton joints reduces the number of search locations by five orders of magnitude and thus enables our method to be scalable to petabyte scale reconstruction. Experimentally, we have shown that the skeletal representation does not lose any volumetric



(a) (b) (c) Correcting false merge (d) No false merge (e) Correcting false split

Figure 2: 3D renderings can often resolve the ambiguity in neural reconstruction: (a) 3D view of two incorrectly split segments (magenta and green) due to an imaging artifact shown in Figure 1; (b) 3D view of two incorrectly merged segments. To correct such errors, we utilize skeleton joint of a segment: (c, d) at each skeleton joint, we apply a 3D CNN to predict the existence of a false merge and a cut plane given the 3D subvolumes of segmentation mask and grayscale images (Section 2.1); (e) later, a second 3D CNN is applied only at the terminal points of the skeleton to predict the connectivity of two segments A (cyan) and B (yellow) (Section 2.2).

information for the purpose of error correction. Our error correction method achieves an accuracy comparable to those of Zung et al. [34] by querying a tiny fraction of the volume.

2 Related Work

Although not presented in this line of thought, Zung et al. [34] have demonstrated the potential of a deep CNN based error correction approach for EM segmentation with high accuracy. However, their error detection network does not utilize the shape information already available in the 3D segments and needs to be applied on each voxel – a task that is immensely expensive. On a large reconstruction effort of a petabyte size volume, applying their algorithm will take 20 months on a 40 GPU cluster. Rolnick et al. [25] have proposed a strategy to detect only the false merge location using a deep net applied to a selected number of locations on a downsampled segmented volume. However, their work does not: (1) provide a concrete mechanism to downsample and sparse selection; (2) devise a method to *correct* the over- or under-segmentation errors; (3) quantify how the accuracy of correction deteriorates when a deep net is applied on sparse locations.

3 Method

Given an input segmentation result, our algorithm attempts to correct the two types of errors in it: false merge (under-segmentation) and false split (over-segmentation). Our algorithm consists of two successive stages: first, during the false merge correction stage (Section 2.1) we use a 3D CNN, which detects and corrects false merges in the initial segmentation; then, another 3D CNN is deployed to correct the false splits (Section 2.2) in the volume generated by the first stage or those false splits present in the segmentation originally. Both these stages rely on a skeletal representation of each segment within the input segmentation. We assume the availability of an effective skeletonization algorithm. For all experiments in this paper, we utilized the approach of Zhao and Plaza [33], which is an adaptation of the TEASAR algorithm [27] to generate the skeletons. For each segment $R \in \mathcal{R}$ in the input segmentation $\mathcal{R}$, we generate a 3D skeleton S, which is represented by the coordinates of its line endpoints or joints p_j.

(a) (b) (c) (d)

Figure 3: Detailed overview of false merge correction. (a) the CNN analyzes the subvolumes around the joints (blue) of a skeleton of a selected segment (gray) of input segmentation; (b) prediction from CNN (red); (c) cut plane approximation (claret); (d) separated segments (green and yellow).

In Figures 2(c)-(e), we overlay for demonstration a skeleton composed of a set of lines, and their line endpoints or joints, on an automatically segmented region. For the identification and rectification of the under-segmentation errors, we employ a CNN at each skeleton joint to detect whether or not a false merge is present and to predict the set of voxels that are later used to construct a cut plane to separate the two or more incorrectly merged segments. This process is illustrated in Figures 2(c),(d) and described in details Section 3.1. Afterward, our method also checks each extreme or terminal points of all the skeletons (these are the line endpoints at the extremities of the skeletons) with another CNN to determine whether or not the segment should be connected to any of its adjacent segments, as shown in Figure 2(e) (Section 3.2).

The presented approach is fundamentally different from that proposed in Zung et al. [34]. They use one deep net for detection and another for correction of both false split and false merge errors. In contrast, we employ a deep CNN for detecting and correcting false merges and another CNN for fixing false split errors. Dividing the over- and under-segmentation into two separate network simplifies the tasks considerably. As a result, our network architecture can achieve almost the same accuracy as in Zung et al. [34] with a simpler design. Due to our use of skeletal representation, an exhaustive search on the whole volume is not required in our method; especially for false splits, our network needs to be applied only at skeleton endpoints which are extremely sparse with respect to the full volume.

3.1 False Merge Correction

Our method for false merge correction consists of three main steps that are illustrated in Figure 3: (1) for each joint p_j, a 3D CNN generates a voxelwise probability map of a potential cut plane within a 3D subvolume P_{EM} centered at p_j (Figure 3(a)); (2) the probability maps of the overlapping volumes are combined using the maximum operation and binarized using a threshold of 0.5. The binary voxels of the predicted cut planes are clustered together based on the predefined distance threshold (Figure 3(b)); (3) The final split is generated using the predicted cut plane clusters and its 3D planar approximations (Figure 3(c),(d)).

3.1.1 False Merge Corrector CNN

Our CNN for dealing with under-segmentation has two goals: (1) detect whether or not there is a false merge at any particular location, and, (2) to predict the set of voxels that constitutes the cut plane for a correction. Both of these tasks are carried out by two designated branches of our CNN, as illustrated in Figure 4. One of the branches possesses fully connected layers

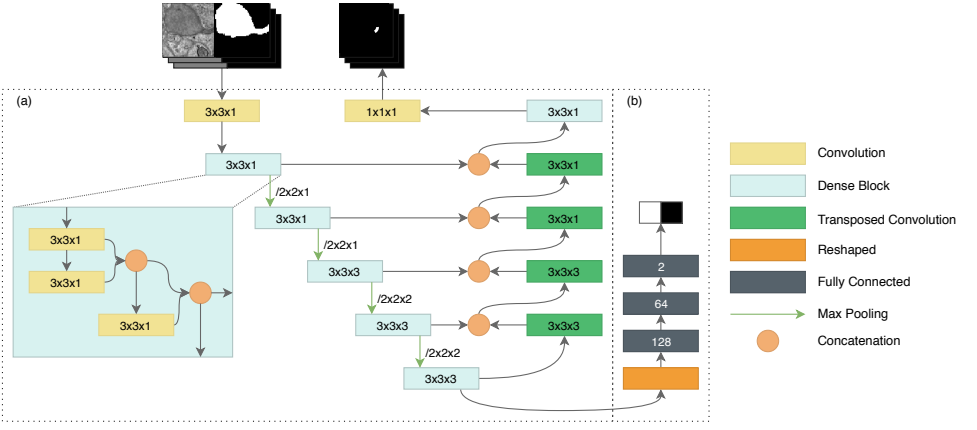


Figure 4: Architecture of our cut plane predicting CNN. Given the raw EM image and a binary mask of a label as input, the fully convolutional branch (a) of the CNN predicts a binary mask of the cut plane to split the incorrectly merged segments, and (b) the fully-connected branch detects whether a false merge is present.

with a final sigmoid activation function for the false merge detection. The other branch is fully convolutional that predicts the set of voxels for the cut plane. The input to our network is a 3D patch with two channels: the raw EM image and a corresponding binary mask of a segmented region, and the outputs are a binary label and a binary mask of the cut plane if an error is present or blank if there is none.

CNN Architecture: The fully-convolutional branch consists of one convolutional layer, followed by five consecutive densely connected convolutional blocks [10], each includes three convolutional layers followed by one pooling layer. The output activations of each dense convolutional block are upsampled with transposed convolutions to match the input size, concatenated, and followed by a dense block and a softmax classifier. All convolutional layers, except the last layer, are followed by the rectified linear unit (ReLU) activation function. To overcome the anisotropic nature (z or depth dimension has a lower resolution than those of x, y plane) of the EM images, the first convolutional layer, the first two densely connected convolutional blocks, pooling layers and the corresponding upsampling layers utilize 2D operations, while the following layers operate in 3D.

The upsampling layers allow us to propagate the high-level activations to the higher layers. However, the impetuous upsampling can quickly become computationally complex due to the cumulative increase of the number of high-resolution activation maps. The symmetrical densely connected upsampling layers [3] can mitigate this inadequacy. However, such an approach adds extra complexity of the convolution operations over a large number of intermediate activations maps in the dense blocks in the upsampling path and might show a tendency to overfit [20]. We propose to use layers of transposed convolutions with strides to upsample the output of each dense block. To avoid the buildup of high-resolution activation maps, we limit the number of upsampled activations proportional to their depth in the model.

The fully-connected branch is trained to perform a binary classification of the input 3D patch. More formally, a block of three consecutive fully-connected layers utilizes the features extracted by the last densely connected convolutional block to infer if the input patch contains an example of the incorrectly merged labels. The first two fully-connected layers are followed by the ReLU activation function, while the last layer ends with a sigmoid function.

Model Training and Optimization: The examples for the training dataset were generated by extracting 3D subvolumes centered at random locations within the training volume. To avoid oversampling locations from large segments, we first randomly choose a segment R from $\mathcal{R}$ and then uniformly sample a random point location within R. We collect a balanced set of examples for false and correct merges in the segmentation. However, the initial segmentations typically possess too few false merge errors to construct a reasonably large training set for our purpose. This is because most of the existing segmentation algorithms are biased to minimize under-segmentation [7, 23]. Therefore, we additionally augment the training data by artificially merging correctly segmented labels from $\mathcal{R}$.

Our network was trained to minimize a joint loss of a binary voxelwise cross-entropy loss computed against the target mask of the cut plane and a binary cross-entropy loss for the fully-connected branch. However, due to the small size of the target mask, which generally occupied less than 1% of the 3D patch, the naive and straightforward training approach leads to the network getting trapped in the local minima without predicting any cut planes. To avoid this, we initially train the network on the highly dilated binary masks of the cut planes and then fine-tune the weights while gradually decreasing the dilation rate. Consequently, such a stepwise process can also be viewed as training a merge error detector on the poorly localized dilated cut planes, which is being fine-tuned to a corrector with the decrease of the dilation rate.

3.1.2 Planar Approximation

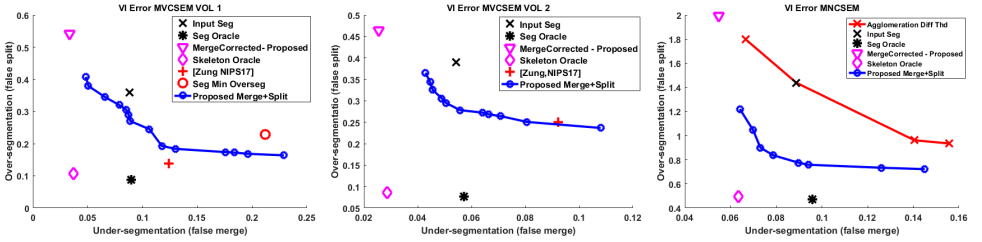
The voxelwise predictions from the fully convolutional CNN do not form in practice an exact cut plane. We approximate the cut plane by fitting a surface to the cluster of discretized CNN detections. The voxelwise probability map, generated by our CNN, is binarized using a threshold of 0.5. Each voxel of the resulting volume belongs to one or more potential cut planes. We first cluster the voxels together based on a distance threshold, which is empirically determined and then additionally filter the clusters based on the size to remove the erroneous isolated points.

Finally, we perform a surface fitting to the voxels of each cluster. Empirically, we found that a simple 3D plane fitting produced the best results. In particular, given a set of the coordinates $\{(x_i, y_i, z_i)\}_{i=1}^{k}$ of the voxels of a particular cluster, we fit a plane $l_1 x + l_2 y + l_3 z + l_4 = 0$. The combination of the cut plane cluster and the corresponding planar approximation represents the final cut plane between the incorrectly merged labels.

3.2 False Split Correction

After rectifying the false merges in the segmentation volume, we apply another 3D CNN at the skeleton terminal points to determine which neighboring segments belong to the same neuron cell and therefore should be connected. Our approach finds all the extreme or terminal points of the skeletal representations by searching joints that have only one neighbor. For a terminal point τ_R of a segment R, our method explores a small volume V_{τ_R} centered at τ_R to locate the set of neighbors $\mathrm{Nbr}(R|\tau_R) = \{N | N \in \mathcal{R} \wedge \mathrm{overlap}(N, V_{\tau_R}) \neq \emptyset\}$ of R that has an overlap with V_{τ_R}. For every possible candidate for continuation, $\{(R,N) | N \in \mathrm{Nbr}(R|\tau_R)\}$, a false split rectifier CNN predicts whether or not R and N belong to the same cell and therefore should be joined, as illustrated by Figure 2(e).

Given the grayscale images and the binary masks of N and R within the small volume V_{τ_R}, the over-segmentation corrector 3D CNN predicts a confidence value on whether N and



(a) VI curve MVCTEM Vol 1 (b) VI curve MVCTEM Vol 2 (c) VI curve MNCSEM

Figure 5: (a, b) VI curves for the experiments on MVCTEM, and (c) on MNCSEM data. The x and y axes correspond to under- and over-segmentation VIs, respectively. Lower VI implies better segmentation. *The VI errors are computed in bits* ($\log_2$).

R should be connected. For this purpose, we have employed a convolutional net with five convolution layers, each with dimension $3 \times 3 \times 3$, parametric ReLU activation and max-pooling, and two fully connected layers (with dropout) at the end. The false merge corrected volume contains more false-splits than the input volume segmentation and distribution of true and false split cases is highly imbalanced. We use class weights to normalize the imbalance within each batch of samples.

4 Experimental Results

Our method was trained and evaluated on two datasets of EM volumes. The training procedures, as well as the parameters for the CNNs used to fix the under- and over-segmentation errors remained the same on both datasets. The inputs to the false merge and false split corrector were grayscale image volumes and binary segmentation masks of sizes $36 \times 192 \times 192$ and $20 \times 192 \times 192$, respectively, as described in Sections 3.1 and 3.2. Our framework was implemented in Python, using Keras library with TensorFlow backend. We use Adam optimizer initialized with the learning rate of $0.0001, \beta_1 = 0.9, \beta_2 = 0.999$ [16]. We augmented the training data by rotation, flip and small translation for training both networks. All segments smaller than 2000 voxels along with all the branches less than 20 voxels long were removed from the skeleton computation [33] to avoid spurious branches and tiny fragments, which are less likely to be useful for error correction. The quantitative evaluation was reported using the variation of information (VI) score, which is a standard evaluation metric in connectomics literature [7, 18, 21, 23].

Given the trained networks and computed skeletons, we apply the CNN correctors on the test volumes sequentially, first to fix the under-segmentation error and then to correct the over-segmentation error. As the improving under-segmentation is prioritized over false splits, we allow the false merge correction to increase the over-segmentation VI slightly in the first stage.

4.1 Correction on Mouse Visual Cortex TEM Data (MVCTEM)

For the first experiment, six non-overlapping EM volumes $2048 \times 2048 \times 256$ voxels, (resolution $3.6 \times 3.6 \times 40\ nm^3$, referred to as MVCTEM) imaged from mouse visual cortex using TEM apparatus were collected from the authors of [34]. The ground truth volumes were generated by correcting an initial segmentation from the method of Lee et al. [18]. Out of

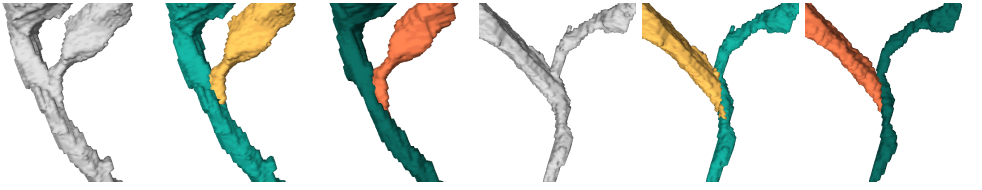


Figure 6: Examples of the corrected erroneously merged segments: (a), (d) input region; (b), (e) segments split by our false merge corrector; (c), (f) ground truth labels.

these six volumes, four volumes were used for training and the remaining two, which were used as validation and test sets by Zung et al. [34], were utilized as test volumes.

The over- and under-segmentation VI errors of the input, intermediate and final outputs of our method, along with those from [34], are plotted in y and x axes of Figure 5, respectively, following the convention in EM connectomics community [7, 23]. Ideally, we wish to achieve a 0 VI for both false merge and split, that is, lower VI is better. Following most studies in EM connectomics, the *VI errors are computed using* $\log_2$*, that is, in bits, as opposed to nats in Zung et al. [34].* As a result, the values reported here will be different from those in Zung et al. [34].

The VI error of the input segmentation is marked as $\times$ in Figure 5. First, we checked what percentage of false split errors is amenable to correction in the ideal scenario. For this purpose, we used an oracle, which utilizes the ground truth segmentation as the reference, to repair all the over-segmentation errors in the input volume. The resulting VI is plotted as $*$ (Seg oracle). The VI of Seg oracle sets a lower bound of errors that can be attained by fixing the false splits in the input segmentation.

After correcting merge errors with our method the under-segmentation VI is reduced substantially (marked as $\triangledown$, dubbed Merge corrected) – from 0.088 to 0.033 in Vol 1 (Figure 5(a)) and from 0.054 to 0.025 in Vol 2 (Figure 5(b)) – with a slight increase in over-segmentation VI. Although the difference in under-segmentation VI appears to be small, they correspond to significant improvement because the error is computed over a large volume of data (1.07 Gigavoxels). In order to further exhibit the results of our technique, we show the 3D views of two false merge corrections in Figure 6. These results demonstrate that our false merge detection and correction technique based on skeleton joints is very effective to detect and correct false merges.

On the merge corrected volume, we applied another oracle that operates only on skeleton terminal points to check whether or not two segments in the volume should be connected. The VI of this oracle corrected result, shown as $\diamond$ (Skeleton oracle) in Figure 5 sets the lower bound of error attainable from the merge corrected volume. A comparison between the over-segmentation VI errors of the two oracles (Seg oracle as $*$ and Skeleton oracle as $\diamond$) suggests that our skeleton based approach should ideally be able to identify almost all the false split mistake locations that one can locate using a segmentation ground truth. In other words, the skeleton extreme points are sufficient to determine the potential over-segmentation errors with a perfect classifier.

The false split correction CNN was trained on segmentations of training volumes, corrected by our under-segmentation correction CNN. The o in Figure 5 shows the VI values of false split correction on the merge corrected test volumes using the CNN and joining technique, described in Section 3.2. Each point corresponds to a threshold on the false split corrector CNN confidence. The VI curves for our merge and split correction are either the

Table 1: Search space reduction by skeletal representation

Size	MVCTEM Vol 1	MVCTEM Vol 2	MNCSEM
Volume size (voxels)	1.07×10^9	1.07×10^9	0.816×10^9
Query points (skeleton joints)	40621	41513	62815
Reduction in search (vol ratio)	3.866×10^{-5}	3.78×10^{-5}	7.69×10^{-5}

same or very close to those produced by the state-of-the-art method [34] (plotted in +) [1]. The fact that a simple vanilla CNN for fixing over-segmentation can achieve such a high success rate suggests that employing a more involved CNN such as [9, 28] can further reduce, if not eliminate, all the split errors at a low under-segmentation VI value.

For reference, we also plot the error of an initial segmentation generated by Lee et al. [18] with a mean agglomeration parameter that generates low over-segmentation in o (Seg min overseg) for one of the volumes (Vol 1, the same result from Vol 2 was unavailable to us). The significantly lower over- and under-segmentation VI values of our algorithm than those of Seg min overseg suggests that our method can fix errors that cannot be addressed by parameter tuning of Lee et al. [18].

The crucial benefit our skeleton based approach offers is the several orders of magnitude reduction in the number of search locations to detect an error. In Table 1, we list the size of the volume to be corrected and the number of locations our method needs to query to detect and correct errors for different datasets used in our experiments. As the third row of Table 1 indicates that our method reduces the search space by five orders of magnitude while achieving the same or very close accuracy of the state-of-the-art method.

4.2 Correction on Mouse Neo Cortex SEM Data (MNCSEM)

The second dataset consists of two non-overlapping volumes of $1335 \times 1809 \times 338$ voxels (resolution of $6 \times 6 \times 30$ nm) and was collected from the authors of [15]. Separate volumes were used in training and testing. The ground truth of these volumes was generated by manual tracing on images (not by correcting an existing segmentation). In an attempt to simulate large-scale neuron reconstruction, we divided the large volumes (both train and test) into smaller overlapping blocks, generated segmentation independently in each block and then merged them together to produce the input segmentation to be used in our correction algorithm. Each small block was segmented using 3D affinity prediction and agglomeration technique of Parag et al. [24].

The experimental setup and parameters of this experiment were the same as those in the MVCTEM experiment. The under- and over-segmentation errors for input segmentation, segmentation oracle output, our merge correction, skeleton oracle on merge corrected volume, and the result of our merge+split correction are plotted as ×, ∗, ▽, ◇, o, respectively in Figure 5(c). In addition, we also show the VI values of input segmentation computed at different thresholds of the agglomeration technique in ×. Our method was able to yield considerably lower overall VI (o) compared to that of the segmentation output (×) at a considerably low under-segmentation error. The skeleton oracle output also implies that in the ideal scenario, the skeletal representation can capture the locations of false split error. The higher VI values of the oracle outputs are due to the boundary mismatch between the automatic segmentation result and the hand traced ground truth. We have also noticed five orders of magnitude reduction in search space in this dataset (see Table 1).

[1] It is worth mentioning here that the correction of [34] used different input segmentations and parameters for Vol 1 and Vol 2

5 Discussion

In this paper, we describe an efficient and effective deep 3D CNN based correction approach for EM neural reconstruction. Utilizing the 3D shape information provided by a skeletal representation, our method has been shown to fix under- and over-segmentation errors created by a 3D segmentation algorithm. We show quantitatively that the skeletal representation contains virtually all volumetric information necessary for error correction purposes. Therefore, with a perfect classifier, one can detect and repair all the mistakes in the volume. By applying the 3D CNNs only on skeleton joints, our method has been able to match the performance of the state-of-the-art method.

Operating 3D CNNs only on skeleton joints reduces the search space for error detection by five orders of magnitude. Such reduction in runtime for error correction modules is critical for the methods to be scalable on very large, for example, petavoxel size, neural reconstruction. This is a crucial benefit our method offers that existing methods lack. It is worth noting that skeletons from segmented volume do not come free of cost – one must take the time for computing skeletons into account to calculate the overall cost of neural reconstruction. For the experiments reported in this paper, we did not invest any effort to develop a new skeleton generation algorithm or optimize the efficiency of existing implementation. There exist faster algorithms for generating skeletons from 3D bodies that one can use to improve the speed.

Another direction is to embed the skeleton generation in the region growing and agglomeration algorithms (flood filling [13], zwatershed [18], waterz [7]). The incremental approach of fitting the largest possible sphere appears to be possible to combine with region growing technique connecting voxels based on affinity values. This approach will minimize the overhead of computing the skeletons from segmentations. We hope the results of our method will motivate research in both computing the skeletons faster and utilizing them for error correction of volume segmentation.

Acknowledegment: The authors gratefully acknowledge Jonathan Zung, Sebastian Seung of Princeton Neuroscience Institute, NJ and Clay Reid, Allen Institute for Brain Sciences, WA for sharing their results with us. This research has been partially supported by the National Science Foundation grants IIS1447344, IIS1527200, IIS1607800, NRT1633299, CNS1650499, the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/Interior Business Center (DoI/IBC) contract number D16PC00002, the Marcus Foundation, and the National Heart, Lung, and Blood Institute of the NIH under Award Number U01HL127522. The content is solely the responsibility of the authors and does not necessarily represent the official view of the NIH. Additional support was provided by the Center for Biotechnology, a New York State Center for Advanced Technology; Cold Spring Harbor Laboratory; Brookhaven National Laboratory; the Feinstein Institute for Medical Research; and the New York State Department of Economic Development under contract C14051.

References

[1] Bjoern Andres, Thorben Kroeger, Kevin L. Briggman, Winfried Denk, Natalya Korogod, Graham Knott, Ullrich Koethe, and Fred A. Hamprecht. Globally optimal closed-surface segmentation for connectomics. *Proc. of European Conference on Computer Vision (ECCV)*, pages 778–791, 2012.

[2] Thorsten Beier, Björn Andres, Ullrich Köthe, and Fred A. Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. *Proc. of European Conference on Computer Vision (ECCV)*, pages 715–730, 2016.

[3] Kevin Briggman, Winfried Denk, Sebastian H. Seung, Moritz N. Helmstaedter, and Srinivas C. Turaga. Maximin affinity learning of image segmentation. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 1865–1873, 2009.

[4] Kevin Briggman, Winfried Denk, Sebastian H. Seung, Moritz N. Helmstaedter, and Srinivas C. Turaga. Maximin affinity learning of image segmentation. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 1865–1873, 2009.

[5] Dan Ciresan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 2843–2851, 2012.

[6] Jan Funke, Bjoern Andres, Fred A. Hamprecht, Albert Cardona, and Matthew Cook. Efficient automatic 3D-reconstruction of branching neurons from em data. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1004–1011, 2012.

[7] Jan Funke, Fabian D. Tschopp, William Grisaitis, Chandan Singh, Stephan Saalfeld, and Srinivas C. Turaga. A deep structured learning approach towards automating connectome reconstruction from 3D electron micrographs. *arXiv preprint arXiv:1709.02974*, 2017.

[8] Kenneth J. Hayworth, Shan C. Xu, Zhiyuan Lu, Graham W. Knott, Richard D. Fetter, Juan C. Tapia, Jeff W. Lichtman, and Harald F. Hess. Ultrastructurally smooth thick partitioning and volume stitching for large-scale connectomics. *Nature Methods*, 12(4):319–322, 2015.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, 2015.

[10] Moritz Helmstaedter, Kevin L. Briggman, Srinivas C. Turaga, Viren Jain, Sebastian H. Seung, and Winfried Denk. Connectomic reconstruction of the inner plexiform layer in the mouse retina. *Nature*, 500(7461):168, 2013.

[11] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[12] Viren Jain, Benjamin Bollmann, Mark Richardson, Daniel R. Berger, Moritz N. Helmstaedter, Kevin L. Briggman, Winfried Denk, Jared B. Bowden, John M. Mendenhall, Wickliffe C. Abraham, Kristen M. Harris, Narayanan Kasthuri, Ken J. Hayworth, Richard Schalek, Juan C. Tapia, Jeff W. Lichtman, and Sebastian H. Seung. Boundary learning by optimization with topological constraints. *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2488–2495, 2010.

[13] Michał Januszewski, Jeremy Maitin-Shepard, Peter Li, Jörgen Kornfeld, Winfried Denk, and Viren Jain. Flood-filling networks. *arXiv preprint arXiv:1611.00421*, 2016.

[14] Michał Januszewski, Jörgen Kornfeld, Peter H. Li, Art Pope, Tim Blakely, Larry Lindsey, Jeremy B. Maitin-Shepard, Mike Tyka, Winfried Denk, and Viren Jain. High-precision automated reconstruction of neurons with flood-filling networks. *bioRxiv*, page 200675, 2017.

[15] Narayanan Kasthuri, Kenneth J. Hayworth, Daniel R. Berger, Richard L. Schalek, José A. Conchello, Seymour Knowles-Barley, Dongil Lee, Amelio Vázquez-Reina, Verena Kaynig, Thouis R. Jones, Mike Roberts, Josh L. Morgan, Juan C. Tapia, Sebastian H. Seung, William G. Roncal, Joshua T. Vogelstein, Randal Burns, Daniel L. Sussman, Carey E. Priebe, Hanspeter Pfister, and Jeff W. Lichtman. Saturated reconstruction of a volume of neocortex. *Cell*, 162(3):648–661, 2015.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Kisuk Lee, Aleksandar Zlateski, Vishwanathan Ashwin, and Sebastian H. Seung. Recursive training of 2D-3D convolutional networks for neuronal boundary prediction. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 3573–3581, 2015.

[18] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and Sebastian H. Seung. Superhuman accuracy on the SNEMI3D connectomics challenge. *arXiv preprint arXiv:1706.00120*, 2017.

[19] Ting Liu, Miaomiao Zhang, Mehran Javanmardi, Nisha Ramesh, and Tolga Tasdizen. SSHMT: Semi-supervised hierarchical merge tree for electron microscopy image segmentation. *Proc. of European Conference on Computer Vision (ECCV)*, pages 144–159, 2016.

[20] Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang W. Koh, Quoc V. Le, and Andrew Y. Ng. Tiled convolutional neural networks. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 1279–1287, 2010.

[21] Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B. Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PloS ONE*, 8(8):e71715, 2013.

[22] Toufiq Parag, Stephen Plaza, and Louis Scheffer. Small sample learning of superpixel classifiers for EM segmentation. *Proc. of International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 389–397, 2014.

[23] Toufiq Parag, Dan C. Ciresan, and Alessandro Giusti. Efficient classifier training to minimize false merges in electron microscopy segmentation. *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 657–665, 2015.

[24] Toufiq Parag, Fabian Tschopp, William Grisaitis, Srinivas C. Turaga, Xuewen Zhang, Brian Matejek, Lee Kamentsky, Jeff W. Lichtman, and Hanspeter Pfister. Anisotropic EM segmentation by 3D affinity learning and agglomeration. *arXiv preprint arXiv:1707.08935*, 2017.

[25] David Rolnick, Yaron Meirovitch, Toufiq Parag, Hanspeter Pfister, Viren Jain, Jeff W. Lichtman, Edward S. Boyden, and Nir Shavit. Morphological error detection in 3D segmentations. *arXiv preprint arXiv:1705.10882*, 2017.

[26] William G. Roncal, Michael Pekala, Verena Kaynig-Fittkau, Dean M. Kleissas, Joshua T. Vogelstein, Hanspeter Pfister, Randal Burns, R. Jacob Vogelstein, Mark A. Chevillet, and Gregory D. Hager. VESICLE: volumetric evaluation of synaptic interfaces using computer vision at large scale. *arXiv preprint arXiv:1403.3724*, 2014.

[27] Mie Sato, Ingmar Bitter, Michael A. Bender, Arie E. Kaufman, and Masayuki Nakajima. TEASAR: Tree-structure extraction algorithm for accurate and robust skeletons. *Proc. of IEEE Pacific Conference on Computer Graphics and Applications*, pages 281–449, 2000.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[29] Shin-ya Takemura, Yoshinori Aso, Toshihide Hige, Allan Wong, Zhiyuan Lu, Shan C. Xu, Patricia K. Rivlin, Harald Hess, Ting Zhao, Toufiq Parag, Stuart Berg, Gary Huang, William Katz, Donald J. Olbris, Stephen Plaza, Lowell Umayam, Roxanne Aniceto, Lei-Ann Chang, Shirley Lauchie, Omotara Ogundeyi, Christopher Ordish, Christopher Shinomiya, Aya andSigmung, Satoko Takemura, Julie Tran, Glenn C. Turner, Gerald M. Rubin, and Louis K. Scheffer. A connectome of a learning and memory center in the adult drosophila brain. *eLife*, 6:e26975, 2017.

[30] Shin-ya Takemura, Aljoscha Nern, Dmitri B. Chklovskii, Louis K. Scheffer, Gerald M. Rubin, and Ian A. Meinertzhagen. The comprehensive connectome of a neural substrate for 'ON'motion detection in drosophila. *eLife*, 6, 2017.

[31] Amelio Vazquez-Reina, Michael Gelbart, Daniel Huang, Jeff W. Lichtman, Eric Miller, and Hanspeter Pfister. Segmentation fusion for connectomics. *Proc. of IEEE International Conference on Computer Vision (ICCV)*, pages 177–184, 2011.

[32] Shiv N. Vitaladevuni and Ronen Basri. Co-clustering of image segments using convex optimization applied to EM neuronal reconstruction. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2203–2210, 2010.

[33] Ting Zhao and Stephen M. Plaza. Automatic neuron type identification by neurite localization in the drosophila medulla. *arXiv preprint arXiv:1409.1892*, 2014.

[34] Jonathan Zung, Ignacio Tartavull, Kisuk Lee, and Sebastian H. Seung. An error detection and correction framework for connectomics. *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 6818–6829, 2017.