Scalable Tracing of Electron Micrographs by Fusing Top Down and Bottom Up Cues Using Hypergraph Diffusion

Vignesh Jagadeesh, Min-Chi Shih, B.S. Manjunath, and Kenneth Rose*

Department of ECE and Center for Bioimage Informatics
University of California, Santa Barbara CA - 93106

Abstract. A novel framework for robust 3D tracing in Electron Micrographs is presented. The proposed framework is built using ideas from hypergraph diffusion, and achieves two main objectives. Firstly, the approach scales to trace hundreds of targets without noticeable increase in runtime complexity. Secondly, the framework yields flexibility to fuse top down (global cues as hyperedges) and bottom up (local superpixels as nodes) information. Subsequently, a procedure for auto-seeding to initialize the tracing procedure is proposed. The paper concludes with experimental validation on a challenging large scale tracing problem for simultaneously tracing 95 structures, illustrating applicability of the proposed algorithm.

Keywords: Tracking, Tracing, Electron Micrograph, Spectral Graphs.

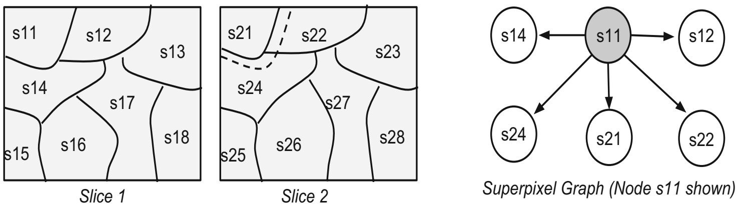
1 Introduction

Connectomics [1] is a sub-field of bio-informatics attempting to understand neuronal connectivity patterns from data acquired using microscopic imaging of neurons. This work focusses on the analysis of volumetric datasets from a connectome, acquired using electron microscopy at nanometer resolutions. The major image analysis challenge in tracing neuronal structures from Electron Micrographs (EM) are two-fold. Firstly, the datasets are extremely large with a requirement to scale algorithms to trace hundreds of targets (structures) simultaneously. Secondly, the structures present in the data undergo arbitrary deformations and topological changes that need to be accurately modeled. This work proposes a tracing model attempting to jointly satisfy the above requirements. The problem can be defined as one of extracting 3D reconstructions of hundreds of structures from a volumetric dataset in an accurate and computationally efficient manner. There are well established algorithms for single structure tracing in Electron Micrographs with deformations and topological changes. However, creating multiple binary segmentations by applying single structure tracing on

* This work was supported by NSF OIA 0941717 and NSF III 0808772. The authors thank Dr.Robert Marc, Dr.Bryan Jones and Dr.James Anderson from the Univ. of Utah for providing data used in experiments and for useful discussions.

every structure is problematic. Firstly, if a pixel is set to one in the binary masks of two structures, it is not clear which label takes ownership of the structure. Secondly, interactions between structures cannot be modeled in such a scenario. The obvious solutions of utilizing discrete Markov random fields(MRFs)/Level sets, though attractive may not be most suitable as verified by experiments in Section 3. In case of MRFs, scaling the number of labels has a direct impact on the runtime of algorithms like alpha expansions and behavior of such methods for segmenting hundreds of labels is not a well studied problem (though such problems have been looked into for stereo and optic flow). Level Set methods like the Chan-Vese model, have also not been shown to work on hundreds of labels.

Proposed Solution: The image stack is assumed to be made up of superpixels linked to each other in three dimensions, forming a graph. As an example, Figure 1a shows superpixel segmentation of two consecutive slices, say s13 represents third superpixel in slice 1. The superpixel graph is constructed by introducing edges between the superpixel of interest (s11) and its spatial (s12, s14) and temporal neighbors (s21, s22, s24). Hyperedge construction from top down information is illustrated in Figure 1b. Maroon dotted circle is the output of top down detector grouping red, green, blue and yellow blobs leading to a hyperedge. Hyperedges based on k-nearest neighbors are similarly constructed. The key idea is to model the label propagation across image sequences as the solution of a hypergraph diffusion equation in a 3D superpixel hypergraph. In doing so, one is presented with a model having some very desirable properties. Firstly, a flexible framework that can utilize top down (coarse object location) and bottom up (local image structure) information results. Secondly, the diffusion has a closed form solution with complexity weakly dependent on the number of labels. In



(a) Three Dimensional Superpixel Graph Construction

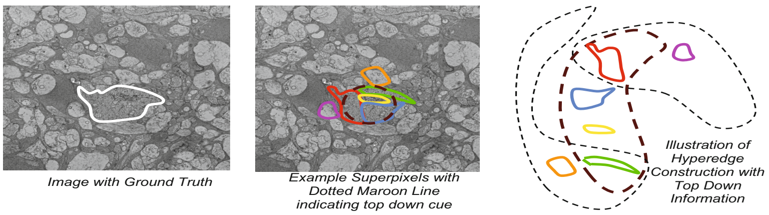


Image with Ground Truth
Example Superpixels with Dotted Maroon Line indicating top down cue
Illustration of Hyperedge Construction with Top Down Information

(b) Hyperedge Construction from Superpixels

Fig. 1. Construction of the 3D superpixel hypergraph

other words, the complexity remains unchanged in spite of an increase in the number of labels. The primary contributions of this work include: ***An efficient tracing framework based on hypergraph diffusion that fuses top down and bottom up cues*** and a method for ***Automatic Target Seeding***.

Related Works: The works by [7,8,6,12] are good sources of reference for EM image analysis. Further, [5] utilized hypergraphs for unsupervised video segmentation, in contrast to the supervised case the proposed approach deals with. Salient aspects of the proposed 3D tracing framework are scalability to hundreds of labels, modeling higher order interaction between segments, introduction of global contour cues using hyperedges, generic autoseeding for fully automatic tracing, and semi-supervised nature, amenable to user interaction if needed. Our claim of originality is in the framework comprising the salient aspects listed above. In related work, [12] propose an interesting technique utilizing pairwise segment interactions on EM data from a mouse (gradient based), but do not lay emphasis on user interaction or scalability. The data used in this work is very different and is from a rabbit retina (noisy regional texture based). Techniques similar to [12] did not perform well on our datasets, leading us to compare with the state of the art on rabbit retina and relevant tracing techniques. Our work is intended as a scalable replacement to the graph cut solvers used in [6], as will be established by experimental results.

2 Proposed Model

The model is initially presented in terms of a bigraph. Subsequently, it is extended to the hypergraph case. The intuition behind the graph diffusion energy is motivated by semi-supervised learning [13], where structure of the data manifold is utilized along with a sparse initial labeling of data points (y) to arrive at a final labeling(f). Consider a bigraph $G = (V, E, w)$, comprising a vertex set $v \in V$ with weights between nodes $\{u, v\} \in G$ denoted by $w(u, v)$. Further, let $d(u)$ denote the degree of node u. Considering a two label model, $f \in \{-1, 1\}^{|V|}$ is the classification function to be estimated, and $y \in \{-1, 1\}^{|V|}$ is the initial labeling vector. The following equation can be interpreted as follows, estimate a labeling function f over graph G whose smoothness is measured by a smoothness cost term, and which does not deviate too much from initial labeling: $\underset{f\in R^{|V|}}{\operatorname{argmin}} \frac{1}{2} \underbrace{\sum_{\{u,v\}\in V} w(u,v)\left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}}\right)^2}_{Smoothness\ Cost} + \underbrace{\mu||f-y||^2}_{Deviation\ from\ Seeds}$.

The above formulation can be extended to a hypergraph [14], generalizing the notion of an edge linking a pair of nodes to a hyperedge linking multiple nodes. The intuition behind the hypergraph model is similar to the bigraph case, except for the fact that smoothness is over multiple nodes constituting a hyperedge. Consider a hypergraph $G = (V, E, w)$, comprising a vertex set $v \in V$, an edge set $e \in E$ and a set of weights w. A hyperedge e comprises of a set of nodes $\mathbf{v_e} \subset V$ that form a clique inside the hyperedge. The degree of a hyperedge is $\delta(e) = |\mathbf{v_e}|$, while the degree of a vertex is defined by $d(v) = \sum_{e \in E, v \cap \mathbf{v_e} \neq \varnothing} w(e)$. The incidence matrix $H \in \Re^{|V| \times |E|}$ contains binary elements h(v,e) taking the value 1 if $v \in v_e$, and 0 otherwise. $D_e \in \Re^{|E| \times |E|}$ and $D_v \in \Re^{|V| \times |V|}$ refer to the diagonal matrices of hyperedge and vertex degrees. The ultimate goal is to perform estimation of a smooth function f on the graph, given an initial labeling $y \in \{-1, 1\}^{|V|}$. The formulation to accomplish the same is given by: $\underset{f \in R^{|V|}}{\operatorname{argmin}} \frac{1}{2} \underbrace{\sum_{e \in E} \sum_{\{u,v\} \subset e} \frac{w(e)}{\delta(e)} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2}_{Smoothness\ Cost} + \underbrace{\mu ||f - y||^2}_{Deviation\ from\ Seeds}$.

Defining the matrix $\Theta = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$, and $\triangle = I - \Theta$, it is straightforward to obtain a closed form solution on label certainties by, $f = (1-\zeta)(I - \zeta\Theta)^{-1} y, \zeta = \frac{1}{1+\mu}$. In the context of image labeling (see Figure 1), $v \in V$ corresponds to a set of superpixels in an image sequence, $e \in E$ refer to the hyperedges constructed by including higher order neighbors on the 3D superpixel graph, thus forming the matrix H. Inferring using the above equation would result in a class marginal on each superpixel that would lead to a tracing result. The use of transductive hypergraph learning for supervised tracing in EM stacks is our first contribution.

Extension to Multiple Labels and Uncertainty Characterization: The above formulation can be extended to the multiple label case in a straightforward manner. The vectors f, y used for the two class problems are now transformed to matrices $F, Y \in \Re^{|V| \times |\mathcal{L}|}$, where column j of $\{F, Y\}$ correspond to the probability of label j to be associated with every node in the graph. The entry $Y(i,j)$ is set to 1 if node i has a label j associated with it, and $F(i,j)$ yields the probability of node i to be associated with label j after diffusion. Alternately, each row i of the matrices F, Y can be interpreted as the probabilities of node i to be associated with each label. The associated inference is given by: $F = (1 - \zeta)(I - \zeta\Theta)^{-1} Y, \zeta = \frac{1}{1+\mu}$. A side benefit of the above formulation is the fact that uncertainty of solutions can be characterized from the entropy computed using rows of F. Computing uncertainty estimates would point towards confidence of the algorithm in its solutions, and it can readily probe the user for assistance using an active learner in interactive settings.

Low Level Features and Graph Weights: The feature representation of superpixels plays an important role in the end results. We utilize gray scale and Local Binary Pattern (LBP) based texture histograms [9] for characterizing appearance of superpixels. The distance between histograms is modeled using the symmetric Kullback Leibler divergence, assuming independence between gray scale and texture channels. Assuming $h_{gray}(i)$ and $h_{lbp}(i)$ respectively to denote the gray scale and texture histograms of superpixel i, the dissimilarity between superpixel i and j is constructed as: $KL[h(i), h(j)] = \sum_k h(i,k) \ln \frac{h(i,k)}{h(j,k)} + h(j,k) \ln \frac{h(j,k)}{h(i,k)}$ $w(i,j) = \exp\left(-KL[h_{gray}(i), h_{gray}(j)] + KL[h_{lbp}(i), h_{lbp}(j)]\right)$.

Complexity of Algorithm. The complexity of inversion is cubic $\mathcal{O}(|V|^3)$ in the number of nodes, as is evident from the equation for inferring multiple labels. Since the matrix considered is sparse, efficient sparse solvers can be employed leading to considerable reduction in running time. The matrix inversion of the graph Laplacian has the greatest computational load, while the matrix multiplication with the label vector Y is of lower complexity than the inversion. As a result, an increase in the columns of Y (additional targets) does not affect the overall time complexity of the algorithm. An intuitive way of looking at the solution is that the graph Laplacian models the entire 3D stack (primary target and contextual information), and a diffusion utilizing this graph Laplacian yields a simple and efficient method for label propagation. Finally, if solutions need to be corrected during interactive segmentation, the computed inverse matrix can be cached, resulting in extremely fast responses to user corrections. Now two important questions arise, Can the hyperedges be utilized to induce a top down global contour cue? (Global Cue Detectors) and is it possible to automatically initialize the number of targets present in the field of view? (Automatic Seeding). The following discussions answer the above questions followed by experimental validation of the proposed ideas.

Global Cue Detectors. As has been described, global detectors are outputs of any algorithm that gives a rough grouping of the nodes in a graph. In the current problem, any algorithm that gives a probable association between superpixels over the 3D volume, thus modeling higher order correlation over the stack is called a global cue detector. The idea is to learn edge profiles using Boosted Edge Learning (BEL) [4], followed by a pass of watershed transform for obtaining 2D segments. The 2D segments are associated in an unsupervised manner across the third dimension using the Floyd-Warshall all source shortest path algorithm [3] to generate probable global cues, see Figure 2a. These cues define association rules between superpixels, thus modeling longer range correlations. Subsequently, the k-nearest neighbors of every superpixel are also used in constructing hyperedges constituting additional global cues.

An important observation to be made is that the k-nearest neighbor hyperedges enforce a Potts style prior encouraging spatial smoothness of labels among superpixels using pairwise interactions. However, in scenarios where information on association between superpixels over larger spatial neighborhoods (across space and the third dimension) are available, they can be readily encoded using the hyperedges for promoting label smoothness. The top down cue detectors serve the purpose of defining these larger interaction neighborhoods over which label smoothness is encouraged.

Automatic Seeding. Another important problem that arises is the automatic initialization (seeding) of targets for efficient tracing. While the most widely used strategy for target initialization is based on user marked seeds in the first

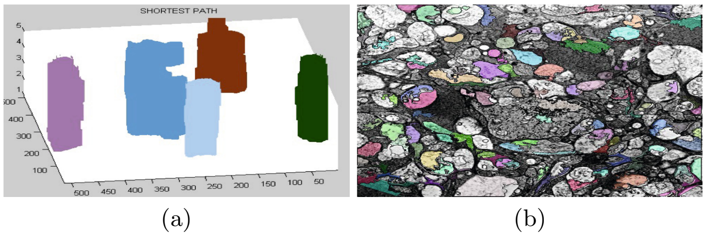


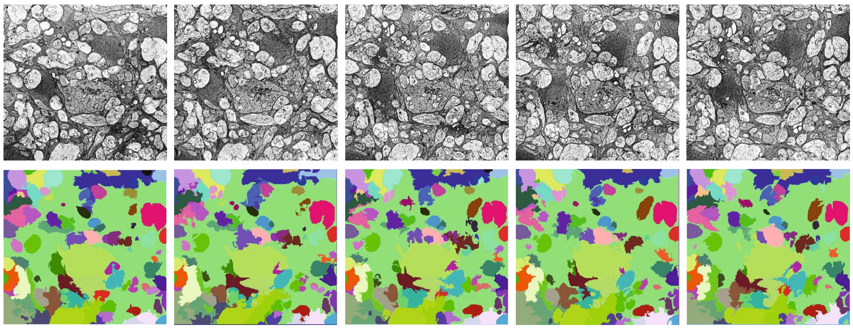
Fig. 2. Figure of the left illustrates outputs from the top down cue detector that serves to construct hyperedges. Figure on right illustrates results of the automatic seeding.

frame, it may not always be possible to seed hundreds of targets manually. In order to solve the above problem, a technique for automatic seeding is proposed. The result of the seeding algorithm can be utilized for initializing the matrix Y. The question asked for auto-seeding is: *Is it possible to pick a set (cardinality $|\mathcal{L}|$) from the K superpixels from the first slice that are as different from one another in appearance ($A_{i1} \in \Re^m$) and spatial positions ($S_{i1} \in \Re^2$)?* The above problem can be solved by estimating an indicator vector $z \in \{0,1\}^K$ that minimizes a cost comprising distances between selected points in spatial and feature spaces. We formalize the above intuition as a relaxed quadratic program [2] (QP) $\underset{z}{\text{argmin}}\sum_{i=1}^{|V|}\sum_{j=1}^{|V|} w_{ij}z_iz_j, s.t \quad \sum_{i=1}^{N} z_i = |\mathcal{L}|$. Rounding the solution of QP yields the desired set of superpixels to be used as seeds for tracing, see Figure 2b. The weights w in the above equation can be constructed in a manner discussed previously.

Contour Refinement. The result of hypergraph diffusion achieves regional homogeneity but is not always edge aware. We utilize an edge based active contour based on hidden Markov models (HMM) [10][11]. Contours resulting from hypergraph diffusion initialize the edge based active contour. For each contour, a trellis with states sampled as points along normals to the contour is instantiated. These points represent the states of the HMM, and any path through the trellis is a potential contour candidate. The Viterbi decoding algorithm yields the final contour passing through strong image gradients.

3 Experiments

Experiments are reported on a tracing task over two separate stacks of electron micrographs. For the purpose of studying the behavior of diffusion in isolation from seeding, contours are manually initialized in the first frame. The metrics used for validating the tracing are the F-measure and Rand index, two commonly used metrics in the segmentation literature. Justification for scalability is given by the running time of algorithms on an Intel Core i7 860 @ 2.8GHz machine. Further, two variants of the proposed idea are utilized in experiments. The HGraph3D method attempts to perform diffusion through the entire 3D graph



(a) Result of hypergraph diffusion on Dataset-I

Method	Median	Mean	Standard Deviation	Time(sec.)
Level Set Tracking	0.55	0.53	0.25	1080
BiGraph3D	0.16	0.23	0.26	78
HGraph3D (Proposed)	0.68	0.66	**0.21**	108
HGraph Propagate (Proposed)	**0.71**	**0.78**	0.22	**34**
Graph Cuts, P^n Model	0.67	0.77	0.28	1320

(b) F-Measure and Running Time on Tougher Dataset-I (95 targets, 5 slices)

Method	Level Set Tracking	HGraph3D	HGraph Propagate	P^n Graph Cuts
Median	0.87	**0.91**	0.89	**0.91**
RunTime(sec.)	378	210	**61**	950

(c) F-Measure and Running Time on Easier Dataset-II (30 targets, 10 slices)

Method	Frame1	Frame2	Frame3	Frame4	Frame5
Level Set Tracking	0.82	0.78	0.75	0.72	0.69
BiGraph3D	0.81	0.70	0.64	0.60	0.58
HGraph3D (Proposed)	**0.88**	**0.86**	0.83	0.79	0.77
HGraph Propagate (Proposed)	**0.88**	**0.86**	**0.85**	0.83	**0.80**
Graph Cuts, P^n Model	-	0.81	**0.85**	**0.84**	**0.80**

(d) Rand Indices on Tougher Dataset-I (95 targets, 5 slices)

Fig. 3. Validation of the Proposed Tracing Framework

and thus performs a one shot optimization. On the other hand, HGraph Propagate attempts to propagate contours in a slice by slice manner with segmentation of one slice being the prior for subsequent slice. Figure 3(a) illustrates the result of tracing all structures over the first few frames of the dataset. In order to place the proposed algorithm in context with existing state of the art techniques, we compare performance with Graph Cuts based P^n model [6], Level Sets using the Chan-Vese model, and bi-graph diffusion, see Figure 3. ***The striking aspect of experiments is the running time of algorithms. For instance, HGraph Propagate runs in 34 seconds on Dataset-I without compromising accuracy, in comparison to graph cuts which takes 1320 seconds. This is a speed up of almost 35***$\times$**.** Table 3(b),3(c) reports statistics on F-Measures against the ground truth over 475 (Dataset-I) and 300 (Dataset-II) contours respectively on two different datasets with corresponding run times. Dataset-I is much more challenging due to appearance variability and larger number of labels. Similarly, Table 3(d) also reports the average Rand index of Dataset-I over all structures in every frame. In conclusion, this paper presented a simple tracing technique that easily scales to hundreds of labels. Experimental results on electron micrographs and comparisons to state of the art illustrate the method's applicability. Future work includes stable auto seeding using Minimum Description Length, deployment on larger distributed computing infrastructures, and active learning based interactive tracing.

References

1. Anderson, J., Mohammed, S., Grimm, B., Jones, B., Koshevoy, P., Tasdizen, T., Whitaker, R., Marc, R.: The viking viewer for connectomics: scalable multi-user annotation and summarization of large volume data sets. Journal of Microscopy 241(1), 13–28 (2011)
2. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge Univ. Press (2004)
3. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to Algorithms. The MIT Press (2001)
4. Dollar, P., Tu, Z., Belongie, S.: Supervised learning of edges and object boundaries. In: CVPR, pp. 1964–1971. IEEE Computer Society (2006)
5. Huang, Y., Liu, Q., Metaxas, D.: Video object segmentation by hypergraph cut. In: CVPR, pp. 1738–1745. IEEE (2009)
6. Jagadeesh, V., Vu, N., Manjunath, B.S.: Multiple Structure Tracing in 3D Electron Micrographs. In: Fichtinger, G., Martel, A., Peters, T. (eds.) MICCAI 2011, Part I. LNCS, vol. 6891, pp. 613–620. Springer, Heidelberg (2011), `http://dx.doi.org/10.1007/978-3-642-23623-5_77`
7. Jurrus, E., Hardy, M., Tasdizen, T., Fletcher, P., Koshevoy, P., Chien, C., Denk, W., Whitaker, R.: Axon tracking in serial block-face scanning electron microscopy. Medical Image Analysis 13(1), 180–188 (2009)
8. Kaynig, V., Fuchs, T.J., Buhmann, J.M.: Geometrical Consistent 3D Tracing of Neuronal Processes in ssTEM Data. In: Jiang, T., Navab, N., Pluim, J.P.W., Viergever, M.A. (eds.) MICCAI 2010, Part II. LNCS, vol. 6362, pp. 209–216. Springer, Heidelberg (2010), `http://dx.doi.org/10.1007/978-3-642-15745-5_26`
9. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI 24(7), 971–987 (2002)
10. Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
11. Shih, M.C., Rose, K.: Hidden markov models for tracking neuronal structure contours in electron micrograph stacks. In: ISBI, pp. 1377–1380. IEEE (2012)
12. Vazquez-Reina, A., Gelbart, M., Huang, D., Lichtman, J., Miller, E., Pfister, H.: Segmentation fusion for connectomics. In: ICCV, pp. 177–184. IEEE (2011)
13. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: NIPS, vol. 16, pp. 321–328 (2004)
14. Zhou, D., Huang, J., Scholkopf, B.: Learning with hypergraphs: Clustering, classification, and embedding. In: NIPS, vol. 19, p. 1601 (2007)