

# Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features

Aurélien Lucchi, Kevin Smith, Radhakrishna Achanta, Graham Knott, and Pascal Fua

**Abstract**—It is becoming increasingly clear that mitochondria play an important role in neural function. Recent studies show mitochondrial morphology to be crucial to cellular physiology and synaptic function and a link between mitochondrial defects and neuro-degenerative diseases is strongly suspected. EM microscopy, with its very high resolution in all three directions, is one of the key tools to look more closely into these issues but the huge amounts of data it produces make automated analysis necessary.

State-of-the-art computer vision algorithms designed to operate on natural 2D images tend to perform poorly when applied to EM data for a number of reasons. First, the sheer size of a typical EM volume renders most modern segmentation schemes intractable. Furthermore, most approaches ignore important shape cues, relying only on local statistics that easily become confused when confronted with noise and textures inherent in the data. Finally, the conventional assumption that strong image gradients always correspond to object boundaries is violated by the clutter of distracting membranes.

In this work, we propose an automated graph partitioning scheme that addresses these issues. It reduces the computational complexity by operating on supervoxels instead of voxels, incorporates shape features capable of describing the 3D shape of the target objects, and learns to recognize the distinctive appearance of true boundaries.

Our experiments demonstrate that our approach is able to segment mitochondria at a performance level close to that of a human annotator, and outperforms a state-of-the-art 3D segmentation technique.

**Index Terms**—Electron microscopy, segmentation, supervoxels, mitochondria, shape features.

## I. INTRODUCTION

IN addition to providing energy to the cell, mitochondria play an important role in many essential cellular functions including signaling, differentiation, growth and death. An increasing body of research suggests that regulation of mitochondrial shape is crucial for cellular physiology [10]. Furthermore, localization and morphology of mitochondria have been tightly linked to neural functionality. For example, pre- and post- synaptic presence of mitochondria is known to have an important role in synaptic function [34].

Mounting evidence also indicates that there is a close link between mitochondrial function and many neuro-degenerative

diseases. Mutations in genes that control fusion and division events have been found to cause neurodegenerative processes [26]. For example, mutations of the gene coding for a protein kinase called PINK1, which is known to regulate mitochondrial division, have been linked to a type of early-onset Parkinson's disease [46].

Unfortunately, because mitochondria range from less than 0.5 to 10  $\mu\text{m}$  in diameter [9], optical microscopy does not provide sufficient resolution to reveal fine structures that are critical to unlocking new insights into brain function. Recent Electron Microscopy (EM) advances, however, have made it possible to acquire much higher resolution images, and have already provided new insights into mitochondrial structure and function [39]. The data used in this work were acquired by a focused ion beam scanning electron microscope (FIB-SEM, Zeiss NVision40), which uses a focused beam of gallium ions to mill the surface of a sample and an electron beam to image the milled face [27]. The milling process removes approximately 5nm of the surface, while the scanning beam produces images with a pixel size of  $5 \times 5\text{nm}$ . Repeated milling and imaging yielded nearly isotropic image stacks containing billions of voxels, such as the ones appearing in Figure 1.

Analyzing such an image stack by hand could require months of tedious manual labor [40] and, without reliable automated image-segmentation tools, much of this high quality data would go unused. This situation arises in part from the fact that most state-of-the-art EM segmentation algorithms [25], [42] were designed for highly anisotropic EM modalities, such as *Transmission Electron Microscopy* (TEM). Such data tends to have a greatly reduced resolution in the  $z$ -direction, and associated segmentation algorithms often process slices individually to deal with the missing data. Our approach processes large 3D volumes in a single step, which is advantageous for isotropic FIB-SEM stacks. More generic Computer Vision algorithms that perform well on natural image benchmarking data sets such as the Pascal VOC (Visual Object Classes) data set [13] perform poorly on EM data, whether it is isotropic or not. There are several reasons for this. The amount of data in a typical EM stack is a major bottleneck, rendering these approaches intractable both in terms of memory and computation time. Furthermore, these approaches rarely account for important shape cues and often rely only on local statistics which can easily become confused when confronted with the noise and textures found in EM data. Finally, the conventional assumption that strong image gradients always correspond to significant boundaries does not hold, as illustrated in Figure 1.

To overcome these limitations, we advocate a graph parti-

A. Lucchi and K. Smith contributed equally to this work.

A. Lucchi, R. Achanta, K. Smith, and P. Fua are in the Computer, Communication, and Information Sciences Department; G. Knott is with the Interdisciplinary Center for Electron Microscopy, EPFL, Lausanne CH-1015 Switzerland. E-mail: firstname.lastname@epfl.ch.

Manuscript received December 23, 2010. Revised May 24, 2011.

Copyright (c) 2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

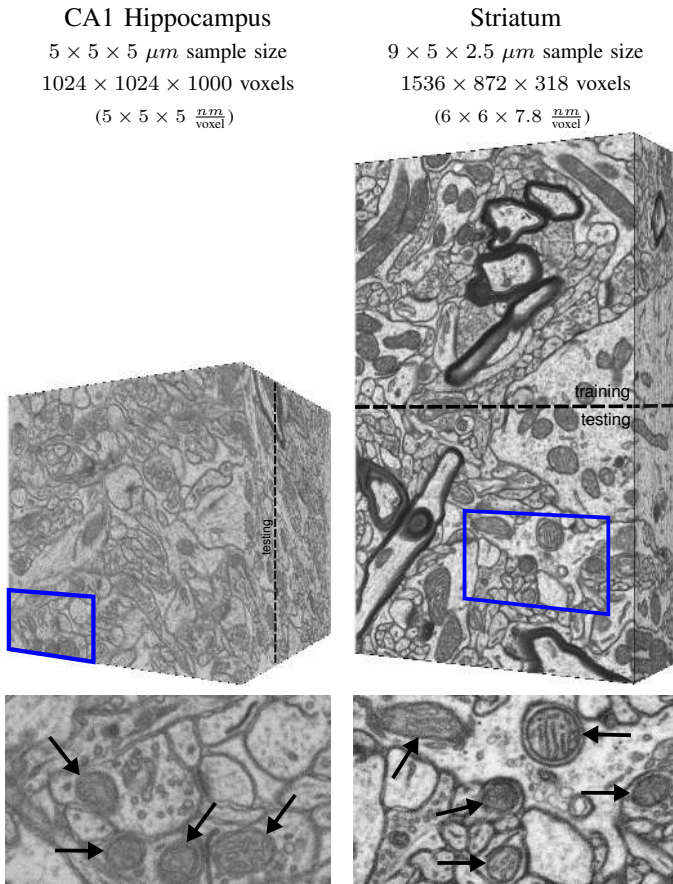


Fig. 1. *FIB-SEM data sets.* The top row contains 3D image stacks acquired using FIB-SEM microscopy. Details in the bottom row are taken from the blue boxes overlaid on the stacks. Mitochondria, which we wish to segment, are indicated by black arrows. The high resolution allows neuroscientists to see important details but poses unique challenges. FIB-SEM image stack dimensions are orders of magnitude larger than conventional images, which limits the usefulness of many state-of-the-art segmentation algorithms, as discussed in Sec. IV-D1. Further complicating the problem are the presence of numerous objects with distracting shapes and textures, including vesicles and various membranes. Finally, we can not rely on strong contrasts to indicate object boundaries. Note that the Striatum data is split into training and testing sections, denoted by a dashed line. A separate training stack is used for the CA1 Hippocampus (not shown).

tioning approach that combines the following components.

- **Operating on supervoxels instead of voxels.** We cluster groups of similar voxels into regularly spaced *supervoxels* of nearly uniform size, which are used to compute robust local statistics. This reduces the computational and memory costs by several orders of magnitude without sacrificing accuracy because supervoxels naturally respect image boundaries.
- **Including global shape cues.** The supervoxels are connected to their neighbors by edges and form a graph. Most graph segmentation techniques rely only on local statistics to partition the graph, ignoring important shape information. We introduce features that capture non-local shape properties and use them to evaluate how likely a supervoxel is to be part of the target structure.
- **Learning boundary appearance.** EM data is notoriously complex, violating the standard assumption that strong image gradients always correspond to significant

boundaries. Spatial and textural cues must be considered when determining where true object boundaries lay. We therefore train a classifier to recognize which pairs of supervoxels are most likely to straddle a relevant boundary. This prediction determines which edges of the supervoxel graph should most likely be cut during segmentation.

We demonstrate our approach for the purpose of segmenting mitochondria in two large FIB-SEM image stacks taken from the *CA1 hippocampus* and the *striatum* regions of the brain. We show that our approach performs close to the level of a human annotator and is much more accurate than a state-of-the-art 3D segmentation approach [52].

## II. RELATED WORK

In this section, we begin by examining previous attempts to segment mitochondria. We then broaden our discussion to include the use of machine learning techniques for other tasks in EM imagery. Finally, we discuss methods that rely on a graph partitioning approach to segmentation.

### A. Mitochondria Segmentation

As discussed in the introduction, understanding the processes that regulate mitochondrial shape and function is important. Perhaps due to the difficulty in acquiring the data, relatively few researchers have attempted to quantify important mitochondria properties in recent years. In [59], a Gentle-Boost classifier is trained to detect mitochondria based on textural features. In [43], texton-based mitochondria classification of melanoma cells is performed using a variety of classifiers including k-NN, SVM, and Adaboost. While these techniques achieve reasonable results, they consider only textural cues while ignoring shape information. A recent approach, described in [52], using state-of-the-art features and a Random Forest learning approach for segmentation has been successfully applied to 3D EM data in [32]. We compare our approach to [52] in Section IV.

In [44], shape-driven watersnakes that exploit prior knowledge about the shape of membranes are used to segment mitochondria from the liver. However, this approach is adapted to anisotropic TEM data. Recently, new features have been introduced to segment mitochondria in neural EM imagery. Ray features, first introduced in [51], were applied to 2D mitochondria segmentation in [36]. Inspired by Ray features, Radon-like features were proposed in [33], but have shown to perform significantly worse than Ray features in [55].

### B. Machine Learning in EM Imagery

Besides mitochondria segmentation, machine learning techniques have found their way into other tasks in EM imagery including membrane detection and dendrite reconstruction. We refer the reader to [23] for an excellent survey covering some of these applications. EM data poses unique challenges for machine learning algorithms. In addition to the large number of voxels involved, a variety of sub-cellular structures exist including mitochondria, vesicles, synapses, and membranes. As seen in Fig. 1, these structures can be easily confused when

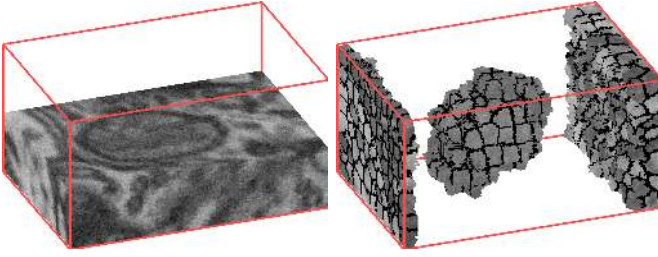


Fig. 2. Segmenting an image stack into supervoxels. (left) A cropped FIB-SEM image stack containing a mitochondrion. (right) The cropped stack is segmented using the SLIC algorithm into groups of similar voxels called *supervoxels*. For visualization, supervoxels in the center of the image stack have been removed, leaving supervoxels belonging to the mitochondrion interior and on the caps of volume. Boundaries between supervoxels are marked in black. Notice that voxels with similar intensities are grouped while respecting natural boundaries.

only local image statistics are considered, especially given the often low signal-to-noise ratio of the data. This is one of the reasons why algorithms that perform well on natural images are far less successful on EM data.

While a large body of research is dedicated to segmenting axons and dendrites from EM data, only a small fraction uses a machine learning approach. In [22], a Convolutional Network (CN) performs neuronal segmentation by binary image restoration. This work is extended in [21] by incorporating topological constraints. In [54], CNs are used to predict an affinity graph that expresses which pixels should be grouped together using the Rand index [49], a quantitative measure of segmentation performance. In another recent approach [25], a random forest classifier is used in a cost function that enforces gap-completion constraints to segment TEM slices.

Machine learning techniques have also been applied to detect membranes, a common preprocessing step in registration and axon/dendrite reconstruction. In [24], Neural Networks relying on feature vectors composed of intensities sampled over stencil neighborhoods are trained to recognize membranes in TEM image stacks. In [58], an Adaboost classifier is trained to detect cell membranes based on eigenvalues and Hessian features. A hierarchical random forest classification scheme is used to detect boundaries and segment EM stacks in [5].

### C. Segmentation by Graph-Partitioning

While active contours and level sets have been successfully applied to many medical imaging problems [12], they suffer from two important limitations: each object requires individual initialization and each contour requires a shape prior that may not generalize well to variations in the target objects. EM image stacks contain hundreds of mitochondria, which vary greatly in size and shape. Proper initialization and definition of a shape prior for so many objects is problematic.

In recent years, graph partitioning approaches to segmentation have become popular. They produce state-of-the-art segmentations for 2D natural images [50], [14], generalize well, and unlike level sets and active contours, their complexity is not affected by the number of target objects. In 2010, the top two competitors [11], [16] in the VOC segmentation challenge [13] relied on such techniques. Graph

---

### Algorithm 1 SLIC Supervoxels

---

```

/* Initialization */
Initialize cluster centers  $C_k = [I_k, u_k, v_k, z_k]^T$  by sam-
pling voxels at regular grid steps  $S$ .
Move cluster centers to the lowest gradient position in a
 $3 \times 3 \times 3$  neighborhood.
Set label  $l(i) = -1$  for each voxel  $i$ .
Set distance  $d(i) = \infty$  for each voxel  $i$ .

repeat
  /* Assignment */
  for each cluster center  $C_k$  do
    for each voxel  $i$  in a  $2S \times 2S \times 2S$  neighborhood
      surrounding  $C_k$  do
        Compute distance  $\delta_{ik}$  between  $C_k$  and voxel  $i$ .
        if  $\delta_{ik} < d(i)$  then
          set  $d(i) = \delta_{ik}$ 
          set  $l(i) = k$ 
        end if
      end for
    end for
  /* Update */
  Compute new cluster centers.
  Compute residual error  $E$ .
until  $E \leq \text{threshold}$ 

/* Post-processing */
Enforce connectivity.
```

---

partitioning approaches minimize a global objective function defined over an undirected graph whose nodes correspond to pixels, voxels, superpixels, or supervoxels; and whose edges connect these nodes [6], [8], [2]. The energy function is typically composed of two terms: the *unary term* which draws evidence from a given node, and the *pairwise term* which enforces smoothness between neighboring nodes. Some works introduce supplementary terms to the energy function, including a term favoring cuts that maximize the object's surface gradient flux [28]. This alleviates the tendency to pinch off long or convoluted shapes, which is important when tracking elongated processes [42]. However, as noted in [25], it cannot entirely compensate for weakly detected membranes and further terms may have to be added.

A shortcoming of standard graph partitioning methods, as we will discuss in Section III-C, is that most do not consider the shape of the segmented objects.

## III. METHOD

The first step of our approach is to over-segment the image stack into *supervoxels*, small clusters of voxels with similar intensities. All subsequent steps operate on supervoxels instead of individual voxels, speeding up the algorithm by several orders of magnitude. This step is described in Section III-A. Next, a feature vector containing shape and intensity information is extracted for each supervoxel, as described in Section III-B. The final segmentation is produced by feeding

the extracted feature vectors to classifiers that define the unary and pairwise potentials of a graph cut segmentation step described in Section III-C. The learning procedure and a list of parameters are provided in Section IV.

#### A. Supervoxel Over-segmentation

Many popular graph-based segmentation approaches such as graph cuts [6] become exponentially more complex as nodes are added to the graph. In practice, this limits the amount of data that can be processed. EM stacks can contain billions of voxels, making such methods intractable both in terms of memory and computation time. Even for moderately-sized stacks, standard minimization techniques [29], [60], [31] become intractable. By replacing the voxel-grid with a graph defined over supervoxels, we reduce the complexity by several orders of magnitude while sacrificing little in terms of segmentation accuracy.

To efficiently generate high-quality supervoxels, we extend our earlier superpixel algorithm, *simple linear iterative clustering* (SLIC) [48], to produce 3D supervoxels such as those depicted in Fig. 2. The approach used in SLIC is closely related to  $k$ -means clustering, with two important distinctions. First, the number of distance calculations in the optimization is dramatically reduced by limiting the search space to a region proportional to the supervoxel size. Second, a novel distance measure combines intensity and spatial proximity, while simultaneously providing control over the size and compactness of the supervoxels.

The supervoxel clustering procedure is summarized in the table marked Algorithm 1. Initial cluster centers are chosen by sampling the image stack at regular intervals of length  $S$  in all three dimensions. The number of supervoxels  $k$  and the number of voxels in the volume  $N$  determines the length,  $S = \sqrt{N/k}$ . Next, the centers are moved to the nearest gradient local minimum. The algorithm then assigns each voxel to the nearest cluster center, recomputes the centers, and iterates. After  $n$  iterations, the final cluster members define the supervoxels.

SLIC is many times faster than standard  $k$ -means clustering thanks to a distance function measuring the spatial and intensity similarities of voxels within a limited  $2S \times 2S \times 2S$  region

$$\delta_{ik} = \sqrt{\frac{(I_k - I_i)^2}{m^2} + \frac{(u_k - u_i)^2 + (v_k - v_i)^2 + (z_k - z_i)^2}{S^2}}, \quad (1)$$

where  $I$  is image intensity;  $u_i$ ,  $v_i$ , and  $z_i$  are the spatial coordinates of voxel  $i$ ;  $u_k$ ,  $v_k$ , and  $z_k$  are those of cluster center  $k$ . Normalizing the spatial proximity and intensity terms by  $S$  and  $m^1$  allows the distance measure to combine these quantities which have very different ranges. Simply applying a Euclidean distance without normalization would result in clustering biased towards spatial proximity. Supervoxel compactness is regulated by  $m$ . As seen in Figure 3, higher  $m$

values produce more compact supervoxels while lower  $m$  values produce less compact ones that more tightly fit the image boundaries.

To ensure that the total number of distance calculations remains constant in  $N$ , irrespective of  $k$ , the distance calculations are limited to a  $2S \times 2S \times 2S$  volume around the cluster centers. This makes the complexity  $O(N)$ , whereas a conventional  $k$ -means implementation would be of complexity of  $O(kN)$  where  $N$  is the number of voxels.

A post-processing step enforces connectivity because the clustering procedure does not guarantee that supervoxels will be fully connected. Orphan voxels are assigned to the most similar nearby supervoxels using a flood-fill algorithm. We refer the interested reader to [4] for further details.

We found SLIC to be particularly well adapted to EM segmentation as it delivers high quality supervoxels efficiently, provides size and compactness control, and can operate on large volumes. Besides SLIC, only a few algorithms are designed to generate supervoxels. In [57], supervoxels are obtained by stitching together overlapping patches followed by optimizing an energy function using a graph cuts approach. However, this approach performs worse than SLIC in terms of segmentation quality using standard measures [4], consumes too much memory, and it is 20 times slower with a worst case complexity is  $O(N^2)$ . A second alternative, used in [5], applies the watershed algorithm [57] to generate supervoxels. However, the size and quality of the watershed supervoxels are unreliable. Finally, other popular superpixel methods could potentially be extended to 3D, including Quickshift [35], Turbopixels [56], and the method of [14]. However, these methods all produce lower quality segmentations than SLIC in 2D [4], and are orders of magnitude slower: 13, 164 and 5 times slower, respectively. They also require much more memory. These comparisons are documented in [4].

#### B. Feature Vector Extraction

After extracting supervoxels, the next step of the algorithm is to extract feature vectors that capture local shape and texture information. For each supervoxel  $i$ , we extract a feature vector  $\mathbf{f}_i$  combining Ray descriptors and intensity histograms, written as

$$\mathbf{f}_i = [\mathbf{f}_i^{\text{Ray}^\top}, \mathbf{f}_i^{\text{Hist}^\top}]^\top, \quad (2)$$

where  $\mathbf{f}_i^{\text{Ray}}$  represents a Ray descriptor and  $\mathbf{f}_i^{\text{Hist}}$  represents an intensity histogram. For simplicity, we omit the  $i$  subscript in the remainder of the section.

*1) Ray Descriptors:* Rays are a class of image features introduced in [51] that capture non-local shape information around a given point. We extend Ray features to 3D in this work, and propose a method for bundling a set of Ray features into a rotationally invariant descriptor. Ray features are attractive because they provide a description of the local shape relative to a given location. This formulation fits naturally into a graph partitioning framework because Rays can provide a description of the local shape for locations corresponding to every node in the graph. Descriptors commonly used for shape retrieval that rely on skeletonization or contours, including

<sup>1</sup> $S$  and  $m$  are the average expected spatial and intensity distances within a supervoxel, respectively.  $m$  can be adjusted to control compactness.

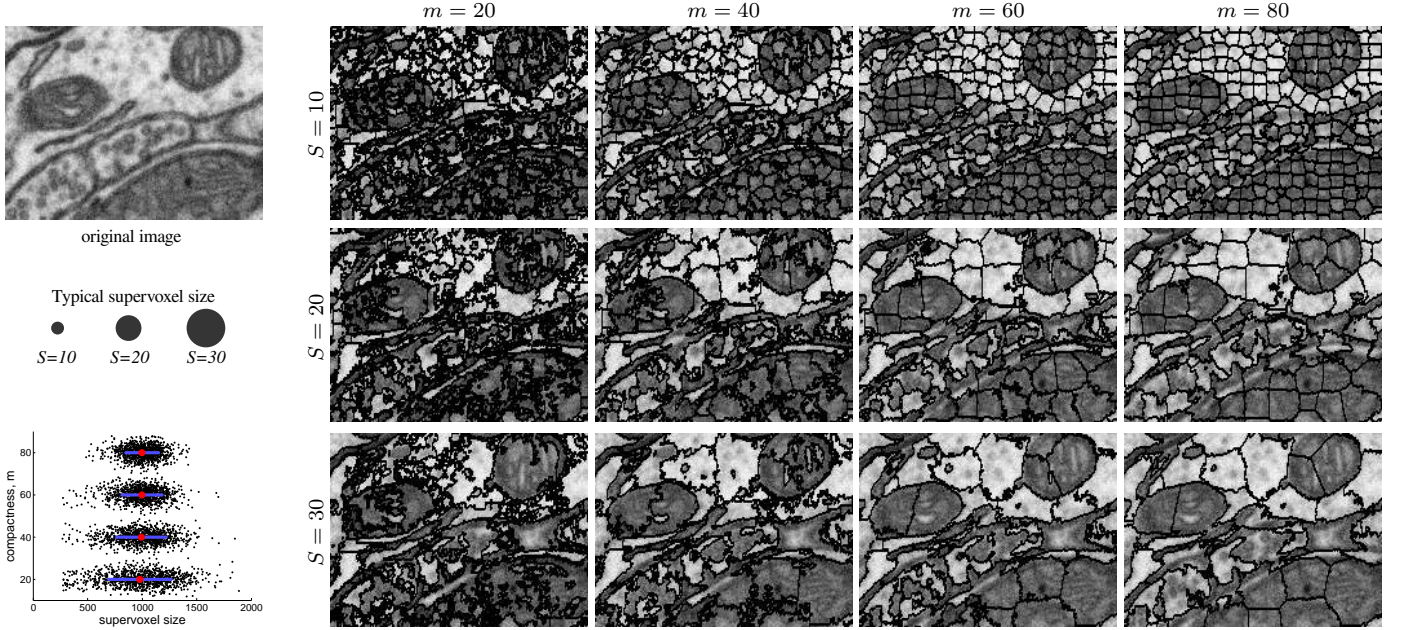


Fig. 3. Supervoxel size and compactness as a function of parameters  $m$  and  $S$  of Eq. 1. (top left) A cropped EM slice containing three mitochondria. (middle left) Typical supervoxels sizes for  $S = 10$ ,  $S = 20$ , and  $S = 30$ . (bottom left) Standard deviation of supervoxel size as a function of varying  $m$ . (right) A matrix of supervoxel segmentations showing the effect of varying  $m$  and  $S$ . Increasing  $m$  produces more compact, regular supervoxels. Increasing  $S$  increases supervoxel size. Note that supervoxels are three-dimensional, yet the images above show only a two-dimensional slice of each supervoxel.

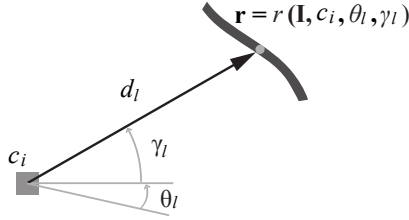


Fig. 4. Ray feature function  $r(\mathbf{I}, c_i, \theta_l, \gamma_l)$ . All components of the Ray descriptor depend on this basic function. For a given location  $c_i$ , it returns the location of the closest boundary point  $\mathbf{r}$  in direction  $l$  defined by angles  $(\theta_l, \gamma_l)$ .  $d_l$  is the corresponding distance from  $c_i$  to the boundary.

distance sets [18] and Lipschitz embeddings [19], do not have this property.

A Ray feature is computed by casting an imaginary ray in an arbitrary direction  $(\theta_l, \gamma_l)$  from a point  $c$ , and measuring an image property at a distant point

$$\mathbf{r} = r(\mathbf{I}, c_i, \theta_l, \gamma_l) \quad (3)$$

where the ray encounters an edge (depicted in Figure 4). In our implementation, edges are found by applying a 3D extension of the Canny edge detection algorithm [20].

For supervoxel  $i$ , we construct a *Ray descriptor* by concatenating a set of  $3L$  Ray features emanating from the supervoxel center  $c_i$ , where  $L$  is a fixed set of orientations. The  $L$  orientations are uniformly spaced over a geodesic sphere, as depicted in Figure 5, and defined by polar angles  $\Theta = \{\theta_1, \dots, \theta_L\}$  and  $\Gamma = \{\gamma_1, \dots, \gamma_L\}$ . The Ray descriptor for supervoxel  $i$  in an image stack  $\mathbf{I}$  at orientation  $(\theta_l, \gamma_l)$  is written

$$\mathbf{f}^{\text{Ray}}(\mathbf{I}, c_i, \theta_l, \gamma_l) = [f_{\text{ndist}}, f_{\text{norm}}, f_{\text{ori}}]^T, \quad (4)$$

where individual Ray features are given by

$$\begin{aligned} f_{\text{ndist}}(\mathbf{I}, c_i, \theta_l, \gamma_l) &= \frac{\|r(\mathbf{I}, c_i, \theta_l, \gamma_l) - c_i\|}{D}, \\ f_{\text{norm}}(\mathbf{I}, c_i, \theta_l, \gamma_l) &= \|\nabla \mathbf{I}(r(\mathbf{I}, c_i, \theta_l, \gamma_l))\|, \\ f_{\text{ori}}(\mathbf{I}, c_i, \theta_l, \gamma_l) &= \frac{\nabla \mathbf{I}(r(\mathbf{I}, c_i, \theta_l, \gamma_l))}{\|\nabla \mathbf{I}(r(\mathbf{I}, c_i, \theta_l, \gamma_l))\|} \cdot \frac{\mathbf{r} - c_i}{\|\mathbf{r} - c_i\|}, \end{aligned} \quad (5)$$

and  $\nabla \mathbf{I}$  is the gradient of the image stack.

In other words, each descriptor  $\mathbf{f}^{\text{Ray}}$  contains three Ray features that measure image characteristics at the nearest edge point  $\mathbf{r}$  given by Eq. 3. The features in Eq. 5 are

- $f_{\text{ndist}}$ , the most basic feature, simply encodes the distance from  $c_i$  to the closest edge  $d_l = \|r(\mathbf{I}, c_i, \theta_l, \gamma_l) - c_i\|$ . It is made scale-invariant by normalizing by  $D$ , the mean distance over all  $L$  directions,
- $f_{\text{norm}}$ , the gradient norm at  $\mathbf{r}$ ,
- $f_{\text{ori}}$ , the orientation of the gradient at  $\mathbf{r}$  computed as the dot product of the unit Ray vector and a unit vector in the direction of the local gradient at  $\mathbf{r}$ .

The final step is to align the descriptor to a canonical orientation, making it rotation invariant. It is important that the descriptor is the same no matter the orientation of the mitochondria, otherwise the learning step would have difficulty finding a good decision boundary. In Fig. 5(a), two perpendicular axes  $\mathbf{n}_1$  and  $\mathbf{n}_2$  define a canonical frame of reference for the descriptor. These axes are assigned specific locations in the feature vector shown in Fig. 5(b), and all other elements are ordered according to their angular offsets from  $\mathbf{n}_1$  and  $\mathbf{n}_2$ . To achieve rotational invariance, we re-order the descriptor such that  $\mathbf{n}_1$  and  $\mathbf{n}_2$  align with an orientation estimate.

To obtain an orientation estimate, Principle Component Analysis (PCA) is applied to the set of Ray terminal points, yielding two orthogonal vectors  $e_1$  and  $e_2$  in the directions of maximal variance of the local shape. Because  $e_1$  and  $e_2$  do

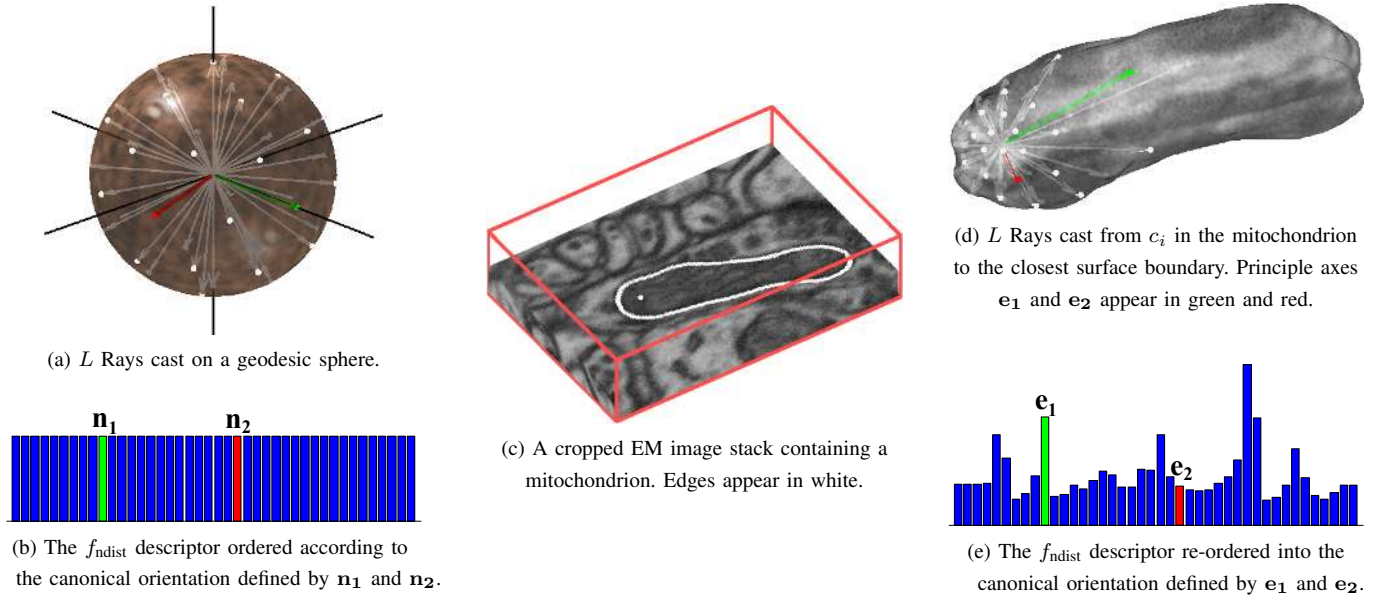


Fig. 5. *Rotation invariant 3D Ray descriptor.* (a)-(b) depict the Ray descriptor cast from the center of a unit sphere. The two axes defining the orientation of the descriptor  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are shown in green and red, respectively. (c) shows a cropped volume containing a mitochondria with boundaries highlighted in white. The white point corresponds to the location of the Ray descriptor in (d)-(e).  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are used to estimate the orientation of the descriptor and are aligned to the canonical orientation.

not necessarily correspond to any of the Ray vectors, we pick the two closest Ray vectors  $\mathbf{e}_1$  and  $\mathbf{e}_2$  to be the principle axes, as shown in Fig. 5(d). Finally, the extracted feature vector is re-ordered into the canonical orientation such that  $\mathbf{e}_1$  and  $\mathbf{e}_2$  correspond to  $\mathbf{n}_1$  and  $\mathbf{n}_2$ , as shown in Fig. 5(e). Note that the accuracy of the pose estimation depends on the number of Rays in the descriptor.

2) *Histogram Features:* Recall from Eq. 2 that the feature vector  $\mathbf{f}$  contains intensity histograms  $\mathbf{f}^{\text{Hist}}$  extracted for a given supervoxel  $i$  and its neighborhood. It complements the Ray features by providing low level intensity and texture cues. We tried several types of local texture and intensity features, including local binary patterns [38] and DAISY [53], but found that a simple histogram computed from a supervoxel  $i$  and its set of neighboring supervoxels  $\mathcal{N}$  yields the best results.  $\mathbf{f}^{\text{Hist}}$  is a concatenation of two  $b$ -dimensional histograms. The first one is extracted from the central supervoxel  $i$ , and the second from all supervoxels belonging to the neighborhood  $\mathcal{N}$  of  $i$ . We write

$$\mathbf{f}^{\text{Hist}}(\mathbf{I}, i) = \left[ h(\mathbf{I}, i, b), \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}_i} h(\mathbf{I}, j, b) \right]^\top, \quad (6)$$

where  $h(\mathbf{I}, j, b)$  is a histogram extracted from  $\mathbf{I}$  over the voxels contained in supervoxel  $j$ . Including the neighbors is necessary, because individual supervoxels are not very discriminative as their intensities are nearly uniform by design.

### C. Graph Cuts with Learned Potentials

The final step of our approach is to segment mitochondria using a graph cuts approach where the unary and pairwise potentials of the energy function incorporate shape cues and learned boundary appearance.

1) *Energy Function:* Graph partitioning approaches minimize a global objective function defined on an undirected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . In our work, nodes  $i$  correspond to supervoxels and edges connect neighboring supervoxels [6], [8], [2]. Our energy function takes the standard form,

$$E(y|x, \lambda) = \sum_i \underbrace{\psi(y_i|x_i)}_{\text{unary term}} + \lambda \sum_{(i,j) \in \mathcal{E}} \underbrace{\phi(y_i, y_j|x_i, x_j)}_{\text{pairwise term}}, \quad (7)$$

where  $\mathcal{E}$  is the set of edges and  $y_i \in \{0, 1\}$  is a class label assigned to  $i$  corresponding to the foreground and the background. The so-called *unary* term  $\psi$  encourages agreement between a node's label  $y_i$  and the local image evidence  $x_i$ .  $\phi$  is known as the *pairwise* term, which promotes consistency between labels of neighboring nodes  $i$  and  $j$ . The weight  $\lambda$  controls the relative importance of the two terms.

We segment the image stack by finding a graph cut that minimizes the energy function of Eq. 7. When the pairwise term is submodular<sup>2</sup>, which is the case in our formulation, a global minima of the energy function can be found using the mincut-maxflow algorithm [17]. This results in an optimal labeling

$$\hat{y} = \underset{y}{\operatorname{argmin}} E(y|x, \lambda). \quad (8)$$

However, following this standard approach does not mean that resulting segmentations are necessarily perfect, or even good. This is because, as is the case in most other works, the criterion being minimized fails to take shape information into account, even though it is crucial for effective segmentation. Another contributing factor is that the standard pairwise term

<sup>2</sup>The submodularity condition requires (1) that the unary term  $\psi(y_i|x_i)$  be positive. This is achieved by adding a constant to the energy without affecting the minimum. Submodularity also requires (2) that the pairwise term  $\phi(y_i, y_j|\cdot)$  satisfies the following condition:  $\phi(0, 0|\cdot) + \phi(1, 1|\cdot) \leq \phi(0, 1|\cdot) + \phi(1, 0|\cdot)$ . Note that the minimum energy of binary submodular functions can be found in polynomial time [30].

fails to properly encode the likelihood that edges correspond to mitochondrial membranes, due to the noisy nature of EM data and presence of distracting membranes. In the following subsections, we propose machine learning based solutions to these shortcomings.

2) *Learned Shape Cues in the Unary Term:* We train a Support Vector Machine (SVM) classifier to predict the unary term in Eq. 7 using the feature vector  $\mathbf{f}$  defined in Section III-B. Because  $\mathbf{f}$  includes rotationally invariant shape cues in the form of the Ray descriptor, the SVM injects important shape information into the unary term, which is taken to be

$$\psi(y_i|x_i) = \frac{1}{1 + P_\psi(y_i|x_i)}, \quad (9)$$

where  $y_i = 0$  indicates background,  $y_i = 1$  indicates foreground, and  $P_\psi$  represents the probability that  $i$  is within a mitochondria. Because the mitochondria have thick boundaries with specific gray-level statistics, the classifier is trained using manually annotated data with three labels  $\{BG, BD, MI\}$ , corresponding to background, boundary, and mitochondria instead of only background and mitochondria. Empirically, we found that introducing an explicit boundary class improved the classifiers' ability to recognize mitochondrial membranes from other membranes in the image stack. Thus, the SVM returns probabilities of being within a mitochondria  $P(MI|x_i)$ , within the boundary  $P(BD|x_i)$ , or outside  $P(BG|x_i)$ . Since the boundary label separates background regions from mitochondria regions, we write

$$P_\psi(y_i|x_i) = \begin{cases} P(BG|x_i) & , \text{ if } y_i = 0, \\ P(BD|x_i) + P(MI|x_i) & , \text{ otherwise.} \end{cases} \quad (10)$$

A three-way one-vs-rest SVM classifier was used to estimate  $P_\psi$ , using a Radial Basis Function (RBF) kernel whose parameters were optimized through cross validation to minimize the estimated generalization error.

Only a few previous graph-partitioning methods have attempted to incorporate shape information into the energy function, having done so only for 2D images. They can be categorized as either template or fragment-based. The first category fits shape templates to the image in an alignment or detection step. Templates represent target objects as either contours [15] or silhouettes [1], [42], which are learned or painstakingly constructed beforehand. Typically, a distance transform from the template is used to modulate the potential functions. The complexity of these types of approaches and the difficulty of simultaneously aligning multiple templates have restricted previous works to segment singular well-centered objects.

Fragment-based approaches match image patches extracted around a graph node to a predefined fragment code book in an attempt to encode shape information [3], [37]. However, for highly deformable objects such as mitochondria, an extremely large code book is necessary, making such an approach prohibitively expensive.

3) *Learned Boundary Appearance in the Pairwise Term:* Most graph-partitioning approaches define the pairwise term as a simple function which favors cutting edges at locations of abrupt color or intensity changes, such as the one proposed in [6]

$$\phi(y_i, y_j|x_i, x_j) = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) & , \text{ if } y_i \neq y_j \\ 0 & , \text{ otherwise,} \end{cases} \quad (11)$$

where the observation  $x_i$  is simply  $I_i$ , the intensity taken from node  $i$ , and  $\sigma$  is a constant. However, in EM imagery containing many distracting contours, this may backfire and result in erroneous cuts either along one of the many membranes found in the data or through a mitochondrial cristae.

We address this problem by learning from the data what types of image characteristics indicate a true object boundary and incorporating this information into the pairwise potential. The pairwise term  $\phi$  is defined as

$$\phi(y_i, y_j|x_i, x_j) = \begin{cases} \frac{1}{1 + P_\phi(y_i, y_j|x_i, x_j)} & , \text{ if } y_i \neq y_j, \\ 0 & , \text{ otherwise,} \end{cases} \quad (12)$$

where  $P_\phi$  is the SVM output probability that  $i$  is within the mitochondria and  $i$ 's neighbor  $j$  is outside. In our application, relevant boundaries are characterized by a very dark membrane separating bright cytoplasm on the exterior, and the dark textured interior of the mitochondria on the interior, as seen in Fig. 1. We therefore train the second three-way SVM using concatenated feature vectors from neighboring supervoxels  $i$  and  $j$

$$\mathbf{f}_{i,j} = [\mathbf{f}_i^\top, \mathbf{f}_j^\top]^\top, \quad (13)$$

where  $\mathbf{f}_i$  and  $\mathbf{f}_j$  are the feature vectors extracted from the individual supervoxels. The resulting classifier assigns probabilities to one of the three classes  $y_{ij} = \{0, 1, 2\}$  where class 0 corresponds to  $BD$ - $BG$  pairs, class 1 corresponds to  $BD$ - $BD$  pairs, and class 2 corresponds to any other combination of ground truth labels

$$P_\phi(y_i, y_j|x_i, x_j) = \begin{cases} P(y_{ij} = 0|x_i, x_j) & , \text{ if } y_i \neq y_j, \\ P(y_{ij} = 1|x_i, x_j) + P(y_{ij} = 2|x_i, x_j) & , \text{ otherwise.} \end{cases} \quad (14)$$

Very few other works use a more sophisticated pairwise potential than that of Eq. 11. While some incremental extensions based on Laplacian zero-crossings, gradient orientations, and local histograms exist [41], very few works go much further. A recent exception can be found in [2], where the authors define an interaction term that encodes geometric relations between multi-region objects. In [47], a set of boundary pixels extracted with an edge detector are pruned using a classifier such that only class-specific edges remain. These edges are attenuated in the pairwise term of the graph cuts segmentation.

#### IV. RESULTS

In this section, we first provide details related to the experimental setup and the FIB-SEM data. We then list the parameters we used and describe the learning procedure. We then present our mitochondria segmentation results, investigate some of the trade-offs of our approach, and finally compare our approach to a state-of-the-art method.

TABLE I  
PARAMETERS AND SETTINGS

Parameter	Value(s)	Notes
$S$	10	Normalized spatial distance. Controls the number of voxels per supervoxel.
$m$	40	Normalized intensity distance. Controls supervoxel compactness.
$n$	5	Number of iterations required for supervoxel clustering to converge.
$L$	42	Number of Ray directions. Corresponds to vertices on a geodesic sphere.
$\rho$	$\approx 50$	Number of Ray features computed per supervoxel.
$\sigma_G$	9	Variance of Gaussian derivative filter used to compute gradient in $f_{\text{ori}}$ and $f_{\text{norm}}$ .
$\sigma_C$	(8,10)	Variance used in 3D Canny edge detection for (CA-1 Hippocampus, Striatum).
$t_l$	(8,14)	Lower threshold used in 3D Canny edge detection for (CA-1 Hippocampus, Striatum).
$t_u$	(16,27)	Upper threshold used in 3D Canny edge detection for (CA-1 Hippocampus, Striatum).
$b$	10	Number of histogram bins. $\mathbf{f}^{\text{Hist}}$ concatenates two $b$ -bin histograms from $i$ and $i$ 's neighborhood.

#### A. Experimental Setup

The data used in our experiments, shown in Figure 1, come from two different locations in the brain. The first image stack represents a  $5 \times 5 \times 5 \mu\text{m}$  section taken from the *CA1 hippocampus*, corresponding to a  $1024 \times 1024 \times 1000$  volume which contains  $N \approx 10^9$  total voxels. The resolution of each voxel is approximately  $5 \times 5 \times 5 \text{ nm}$ . The second section measures approximately  $9 \times 5 \times 2.5 \mu\text{m}$ , and was taken from the *striatum*, a subcortical brain region. This image stack contains  $1536 \times 872 \times 318$  voxels, with a  $6 \times 6 \times 7.8 \text{ nm}$  resolution.

Because of the forbiddingly large amount of labor involved in generating an accurate ground truth for such large volumes, we annotated sub-volumes for training and testing purposes. The testing sub-volume for the CA1 hippocampus consists of the first 165 slices of the  $1024 \times 1024 \times 1000$  image stack, as indicated by the dotted line in Figure 1. A separate image stack from another hippocampus sample containing 200 similarly sized slices was annotated for training our algorithm.

For the striatum, the  $1536 \times 872 \times 318$  volume was fully annotated and split into a training and test set, as indicated in Figure 1.

Each of these sub-volumes had a size of  $768 \times 872 \times 318$ . The results provided in Figure 7 and Table II are computed on the test sub-volumes after training the classifiers on the training sub-volumes. The segmentations shown in the top row of Figure 6 are over the entire  $1024 \times 1024 \times 1000$  image stack for the hippocampus data including the test volume and unannotated data, while the striatum segmentations are shown only for the test sub-volume.

#### B. Parameters and Implementation Details

A summary of parameters used in our experiments is provided in Table I. The sampling interval  $S$  for supervoxel

centers introduced in Section III-A was chosen empirically. The resulting supervoxels contain approximately 1000 voxels on average. Supervoxels of this size typically fit within the membranes which helps to ensure that superpixels do not straddle boundaries. As discussed in Section IV-D1, using supervoxels decreases the computational complexity by several orders of magnitude as compared to what would have been required to operate directly on voxels. A strength of the SLIC supervoxel generation scheme is that  $S$  value can be adapted if the image resolution were to be changed. The compactness factor  $m$  was chosen empirically and provides a good compromise between compactness and boundary adherence. The typical neighborhood size of a supervoxel is  $|\mathcal{N}| \approx 8$  for the  $m$  and  $S$  values given in Table I.

The ray descriptors  $\mathbf{f}^{\text{Ray}}$  of Eq. 4 are  $3L = 126$  dimensional vectors, consisting of 3 Ray feature types and  $L$  orientations. We have found  $L = 42$  to be a good trade-off between computational complexity and angular resolution for the rotational alignment discussed at the end of Section III-B. Rays terminate when they encounter edges found in a 3D Canny edge map [20], whose parameters  $\sigma_G$ ,  $\sigma_C$ ,  $t_l$ , and  $t_u$  must be tuned to the data. Because the Canny edge detector can easily miss edges or add spurious ones, we increase robustness by shooting rays from 5% of the voxels within each supervoxel—50 in our case—for each direction and average the results. It is those averages that we use for classification.

All parameters of our algorithm were fixed for both data sets, except for parameters related to the 3D canny edge detector which was adjusted due to differences in contrast between the two data sets.

#### C. Experiments and Evaluation

We evaluate our segmentation in terms of the so-called *Jaccard index*, or *VOC score* [13] to measure segmentation quality when ground-truth data is available. It is computed as

$$\text{VOC} = \frac{\text{True Pos}}{\text{True Pos} + \text{False Pos} + \text{False Neg}}, \quad (15)$$

which is the ratio of the areas of the intersection between what has been segmented and the ground truth, and of their union. As an alternative to the Jaccard index, we also considered using the Rand index [21] which attempts to penalize topological segmentation errors. However, since the Rand index does not account for all types of topological errors and the Jaccard index is the *de facto* standard in the Computer Vision community, we report our results using the latter.

Table II summarizes the segmentation results of our approach and several baseline methods for the hippocampus

TABLE II  
SEGMENTATION RESULTS MEASURED BY THE VOC SCORE [13]

	Method				
	Ilstik	Standard $\mathbf{f}^{\text{Hist}}$	Learned Cube $\mathbf{f}$	Standard $\mathbf{f}$	Learned $\mathbf{f}$
Hippocampus	61%	63%	68%	81%	84%
Striatum	58%	60%	60%	70%	74%

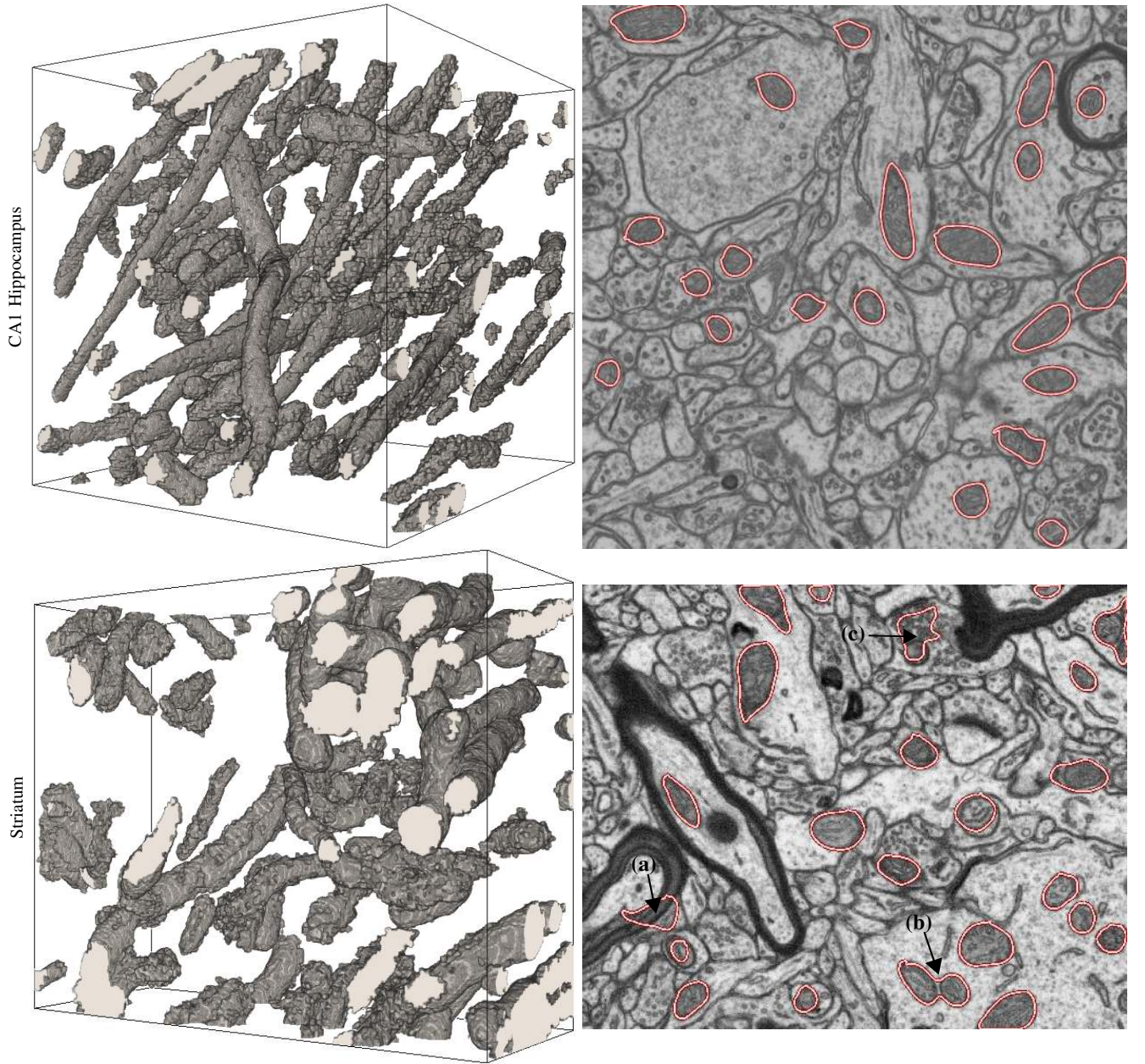


Fig. 6. *Segmentation of mitochondria from FIB-SEM image stacks and 3D reconstructions.* We applied our approach to two FIB-SEM test stacks acquired from different brain regions. The left column shows the 3D reconstructions of extracted mitochondria. Renderings were produced using V3D [45]. The right column shows segmentation results on individual image slices taken from the image stack. Automatically segmented mitochondria are marked by red contours. Most mitochondria are correctly segmented, but some mistakes remain. Failure modes are indicated by black arrows. (a) Dendritic or axonal membranes in close proximity to a mitochondrion can confuse our algorithm, causing it to include part of the nearby membrane with the mitochondrion. (b) Occasionally, neighboring mitochondria are erroneously merged by the smoothness constraint in graph cuts when the space between the membranes is very small. (c) A cluster of vesicles is mistaken for a mitochondrion. The texture of vesicles can appear deceptively similar to that of mitochondria.

and striatum test sets. Adding the Rays to the feature vector  $\mathbf{f} = [\mathbf{f}_i^{\text{Ray}^\top} \mathbf{f}_i^{\text{Hist}^\top}]^\top$  (*Standard f*) is compared to histogram features alone  $\mathbf{f} = \mathbf{f}^{\text{Hist}}$  (*Standard f<sup>Hist</sup>*). We also report results for learning the pairwise term of Eq. 12 with the full feature vector (*Learned f*). Finally, Table II also contains the results obtained using Ilastik [52], and results obtained by replacing the supervoxels with regularly space cubes (*Learned Cube f*). The discussion in the next section provides further details for each method.

The VOC scores reported in Table II were computed by fixing the value of  $\lambda$  to a value determined through a cross-

validation process on the training data. Typically,  $\lambda$  ranged from 0.07 to 0.13.

In the left of Figure 6, 3D reconstructions of mitochondria extracted from the test volumes using our approach are provided. In the right column of the same figure, segmentation results on individual image slices are shown where segmented mitochondria are marked by red contours. The total training and processing time was 23 hours for the hippocampus data set and 7 hours for the striatum data set on a 8-core Intel Xeon CPU 2.4 GHz machine with 48 GB RAM.

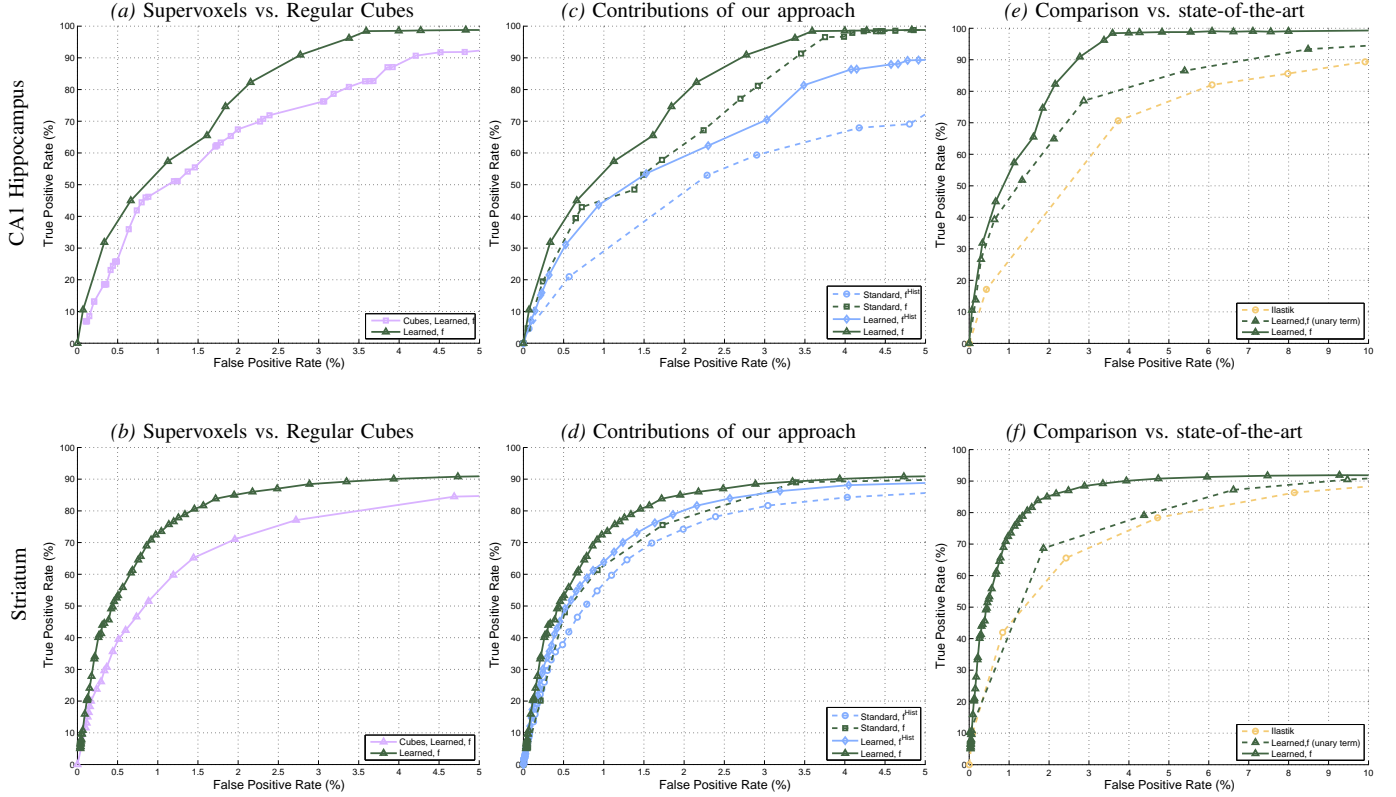


Fig. 7. *Segmentation results.* The top row contains results obtained for the CA1 Hippocampus data, and the bottom row contains results obtained for the Striatum data. (a)-(b) *Supervoxels vs. regular cubes.* We compare segmentation results obtained using SLIC supervoxels of size  $S = 10$  to simple  $10 \times 10 \times 10$  cubes. Supervoxels, which respect boundaries in the image stack, significantly outperform the cubes while similarly reducing computational complexity. (c)-(d) *Contributions of our approach.* The dashed blue line labeled “Standard,  $f^{\text{Hist}}$ ” represents a baseline result obtained by using a unary term that only depends on the histogram features of Eq. 6 and a contrast-based pairwise term given in Eq. 11. Replacing this pairwise term by the learned one of Eq. 12 results in the improved solid blue curve labeled “Learned,  $f^{\text{Hist}}$ .” An even larger improvement is obtained by introducing the Ray features of Eq. 2, producing the green dashed curve labeled “Standard,  $f$ .” Finally, combining the learned pairwise term and the Ray features yield the high quality result denoted by the solid green curve labeled “Learned,  $f$ .” (e)-(f) *Comparing our approach to Ilastik* [52]. We trained the publicly available Ilastik software on the same data we used to train our SVMs and evaluated the segmentations. The solid green curve was generated using our approach. Results obtained using Ilastik appear in as yellow dotted lines. Because Ilastik includes neither smoothing nor regularization, we plot results obtained by thresholding the unary term of Eq. 9 in our approach for a more fair comparison. The dotted curves essentially compare Ilastik’s local texture features to our shape and texture features. Note that thresholding the unary term does not perform as well as our full approach but still better than Ilastik, indicating that the features we use are better adapted to the task at hand. As noted in Section IV-C, the ROC-like plots in (a)-(f) were generated by varying the weight  $\lambda$ , thus changing the influence of the unary and pairwise terms in the energy function of Eq. 7 (with the exception of the dotted curves in (e) and (f), which are conventional ROCs).

#### D. Discussion

We now investigate several aspects of our approach in further detail. We will show the computational advantages of SLIC supervoxels, the benefits of using Ray descriptors, and the performance gained from learning the pairwise term. We also compare our approach against the state of the art, and discuss failure modes of our approach.

These discussions refer to results appearing in the ROC-like curves appearing in Figure 7. The ROC-like curves provided in Figure 7 explore points within the operating regimes of the various method we discuss. To generate these curves, we vary the value of  $\lambda$ , thus changing the influence of the unary and pairwise terms in the energy function of our approach. This results in variations in the true positive rate (TPR) and false positive rate (FPR) of the segmentation, albeit in a non-linear fashion. True ROC curves, like the dotted lines in Figures 7(e) and 7(f) are obtained by varying a classification or decision threshold for independent elements (supervoxels in our case). Most of the curves in Figure 7 were generated by jointly labeling supervoxels using information from their

neighbors through graph cuts, thus, strictly speaking, they are not ROCs. However, they still provide valuable insight into how consistently our algorithm performs over a range of false positive rates.

1) *Computational Advantage of SLIC Supervoxels:* The major bottleneck in our approach is in applying graph-cuts, which has a worst case complexity of  $O(|\mathcal{E}| |\mathcal{V}|^2)$ , where  $|\mathcal{E}|$  is the number of edges and  $|\mathcal{V}|$  is the number of vertices [7]. Using supervoxels instead of voxels reduces  $|\mathcal{V}|$  by several orders of magnitude (a factor of 1000 given the parameters described in IV-B), and therefore significantly speeds up the processing. It is also important to note that memory limitations make it impossible to process a graph of the size required by EM data sets such as ours on a conventional computer. The graph-cuts implementation of [7] requires  $40\mathcal{V} + 32\mathcal{E}$  bytes to store the graph on a 64-bit machine, which translates to a 227GB memory footprint (for 6-connectivity) or a 852GB memory footprint (for 26-connectivity) for the graph required by the *CA1 hippocampus* volume. Using supervoxels with

our parameters reduces the memory consumption to a more manageable size of 296MB.

As an alternative to supervoxels, one might consider downsampling the data to reduce processing time and memory consumption. However, doing so reduces the quality of the segmentation. This is because supervoxels adhere to local image boundaries, whereas downsampling does not. To demonstrate this effect, we compare segmentations obtained using our method with SLIC supervoxels to segmentations obtained by replacing the supervoxels with regularly spaced  $10 \times 10 \times 10$  cubes, which have roughly the same size but ignore boundaries. The results appear in Figure 7(a) and Figure 7(b). Results using our method with SLIC supervoxels are denoted *Learned, f* while the down-sampled results are labeled *Cubes, Learned f*.

It is clear that downsampling produces significantly worse segmentations than using similarly sized SLIC supervoxels. Consequently, downsampling reduces the VOC score by 14 to 16%, as shown in Table II.

2) *Benefits of Ray Descriptors*: The Ray descriptor  $\mathbf{f}^{\text{Ray}}$  in the feature vector of Eq. 2 captures important information about the shape of mitochondria. Without it, the feature vector contains only local information provided by the intensity histograms  $\mathbf{f}^{\text{Hist}}$ . To demonstrate the importance of including shape information, we compare our method using the full feature vector  $\mathbf{f} = [\mathbf{f}_i^{\text{Ray}^\top} \mathbf{f}_i^{\text{Hist}^\top}]^\top$  to our method using only histogram features  $\mathbf{f} = \mathbf{f}^{\text{Hist}}$ .

The results appear in Figure 7(c) and Figure 7(d). Blue lines denote the results obtained using  $\mathbf{f} = \mathbf{f}^{\text{Hist}}$ , while green lines incorporate the Ray features  $\mathbf{f} = [\mathbf{f}_i^{\text{Ray}^\top} \mathbf{f}_i^{\text{Hist}^\top}]^\top$ . Dashed lines and solid lines correspond to a standard or learned pairwise term, which are discussed in the next section. Rays significantly improve the segmentation performance. Without them, the VOC score drops by 18% (see Table II).

Looking at Figure 8, we can see the discriminative power of the combined feature vector. In Figure 8(b) the mitochondria probabilities output by the SVM of Eq. 9 are shown. Directly thresholding these probabilities already results in reasonably good segmentations (Fig. 8(c)).

3) *Learning the Pairwise Term*: Further improvement to segmentation performance is gained by learning the pairwise term of Eq. 12. Results obtained using the standard pairwise potential of [6], which uses a gradient based approach of the form given in Eq. 11, is shown in Figure 7(c) and Figure 7(d) as dashed lines. Replacing this pairwise potential with one that learns which types of image characteristics indicate a true object boundary (Eq. 12) results in a significant increase in performance, as indicated by the solid lines.

This corresponds to an increase in the VOC score by approximately 4%. In Figure 8(d), segmentation results using the learned pairwise term with graph cuts significantly improves the segmentation produced by the unary term in Figure 8(c). For the purpose of this experiment, we set  $\sigma = \frac{1}{2E[\hat{I}_i - \hat{I}_j]^2}$  in Eq. 11, where  $\hat{I}_i$  is the average intensity within supervoxel  $i$  and  $E[\cdot]$  denotes the expectation over supervoxels.

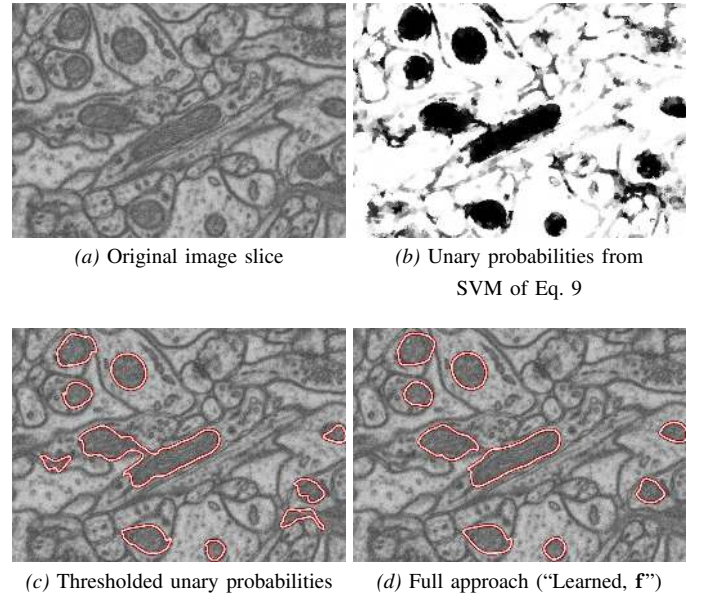


Fig. 8. *Thresholding unary SVM predictions vs. our learned pairwise approach.* (a) Original image slice. (b) Unary mitochondria probability from SVM of Eq. 9 (dark pixels indicate probable mitochondria). (c) Segmentation results obtained by directly thresholding (b). (d) Results obtained with our full approach using graph cuts with a learned pairwise term ("Learned,  $\mathbf{f}$ "). The TPR was set to 85% in (c) and (d).

4) *Comparing against a state-of-the-art method*: The Interactive Learning and Segmentation Tool Kit (Ilastik) is a software package for image classification and segmentation [52]. It allows for interactive labeling of an arbitrary number of classes in data sets of various dimensionality. Similar to the work of [43] which also segments mitochondria, Ilastik uses texture cues as well as color and edge orientation in a machine learning framework to perform segmentation. Ilastik's Random Forest classifier can provide real-time feedback of the current classifier predictions, allowing it to perform interactive or fully automatic classification and segmentation.

We provided Ilastik with the same training data used to train our approach, and compare its output to ours in Figure 7(e) and Figure 7(f). In addition to comparing Ilastik to our full approach, we also plot results obtained by simply thresholding probabilities of Eq. 9 that define the unary term in the energy function. We do this to provide a more fair comparison of our features against those of Ilastik, which does not include a smoothing or regularization step.

While Ilastik achieves a reasonable segmentation, our approach consistently outperforms it, even when using only the unary term. As shown in Table II, our full approach outperforms Ilastik by a margin of 23% on the hippocampus data and 16% on the striatum, as measured by the VOC score. Example segmentations comparing our method to Ilastik are provided in Figure 9. Ilastik mistakenly labels vesicles as mitochondria and has trouble with other various membranes and synapses. Without the global shape information provided by the Ray features such mistakes are difficult to avoid.

5) *Failure modes*: Qualitatively our segmentation results are very promising. Note that the 84% VOC score achieved by our algorithm is outstanding in terms of results reported

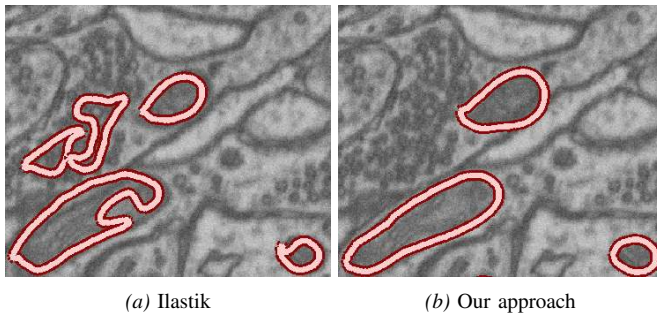


Fig. 9. Visual comparison of our results vs. Ilastik. (a) The voxels of a particular slice that are labeled as being within mitochondria by Ilastik are marked by a red contour. These include a number of voxels that belong to vesicles instead of mitochondria. (b) These mistakes disappear when using our approach.

in the VOC challenge [13]. However, this number should be taken with a grain of salt, as the VOC Challenge contains 21 categories of objects, while we only deal with 2 – the mitochondria and the background. Despite the promising results of our approach, there is still room for improvement. Examples of three failure modes are indicated by arrows in Figure 6. Dendritic or axonal membranes in close proximity to mitochondria can confuse our algorithm, causing it to include part of the nearby membrane with the mitochondria. Occasionally, neighboring mitochondria are erroneously merged as a result of smoothness enforced by graph cuts when the space between the membranes is very small. Finally, clusters of vesicles are mistaken for mitochondria because texture of vesicles can appear deceptively similar to that of mitochondria.

The shallow depth of the training data in the  $z$ -direction could account for some of these failure modes, as very few mitochondria were fully contained within the training volumes. Increasing the amount of training data or enhancing the learning procedure using a bootstrapping approach could potentially reduce these errors. Furthermore, it would be relatively simple to exploit the fact that graph-cut minimization allows for efficient user interaction [6]. This means that, given an adequate interface, remaining errors could be quickly corrected by the user.

## V. CONCLUSION

While the EM image stacks used in this work contain over a billion voxels, they span volumes smaller than  $10 \times 10 \times 10 \mu\text{m}$ , which represents less than a billionth of the volume of the entire mouse brain. If it is ever to be mapped in its entirety, efficient automatic segmentation methods, such as the one we propose in the work, will be required.

Our fully automatic approach to segment mitochondria from FIB-SEM image stacks overcomes the limitations of standard graph-partitioning approaches by: operating on supervoxels instead of voxels for computational efficiency, by using 3D Ray descriptors to model shape in the unary term, and by using a learning approach to model the appearance of the boundary in the pairwise term. We have demonstrated the computational efficiency of using supervoxels, and experimentally shown the increases in segmentation quality attributed with using Ray

descriptors and learning to model boundaries in the pairwise term. Our experiments have also demonstrated that our approach outperforms a state-of-the-art 3D segmentation method, and that our segmentation closely matches the performance of human annotators.

While the focus of our work is on the segmentation of mitochondria in FIB-SEM image stacks, the proposed techniques should be applicable to other cellular structures in EM as well as in other forms of microscopy. Future work will investigate this. We will also focus on learning boundaries using higher-order cliques, exploring the use of other features, and applying our technique to additional types of data.

## ACKNOWLEDGMENTS

This work was supported in part by the EU ERC project MicroNano. The authors would like to thank Christopher Sommer for his help with Ilastik, Yunpeng Li for his ideas and help with the manuscript, Maco Bohumil for labelling the Striatum dataset and the developers of V3D which was used to render the 3D reconstructions.

## REFERENCES

- [1] A. Ali, A. Farag, and A. El-Baz. Graph Cuts Framework for Kidney Segmentation With Prior Shape Constraints. In *Conference on Medical Image Computing and Computer Assisted Intervention*, pages 384–92, 2007.
- [2] A. Delong and Y. Boykov. Globally Optimal Segmentation of Multi-Region Objects. In *International Conference on Computer Vision*, pages 285–292, 2009.
- [3] A. Levin and Y. Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. In *European Conference on Computer Vision*, pages 581–94, 2006.
- [4] R. Achanta. *Finding Objects of Interest in Images Using Saliency and Superpixels*. PhD thesis, EPFL, 2010.
- [5] B. Andres, U. Koethe, M. Helmstaedter, W. Denk, and F. Hamprecht. Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. In *Annual Symposium of the German Association for Pattern Recognition*, pages 142–52, 2008.
- [6] Y. Boykov and M. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *International Conference on Computer Vision*, pages 105–12, 2001.
- [7] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004.
- [8] C. Rother, V. Kolmogorov, and A. Blake. “GrabCut” - Interactive Foreground Extraction Using Iterated Graph Cuts. In *ACM SIGGRAPH*, pages 309–314, 2004.
- [9] N. Campbell, B. Williamson, and R. Heyden. *Biology: Exploring Life*. Pearson Prentice Hall, first edition, 2006.
- [10] S. Campello and L. Scorrano. Mitochondrial shape changes: Orchestrating cell pathophysiology. *EMBO Reports*, 11(9):678–84, 2010.
- [11] J. Carreira and C. Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 3241–48, 2010.
- [12] D. Padfield, J. Rittscher, N. Thomas, and B. Roysam. Spatio-temporal cell cycle phase analysis using level sets and fast marching methods. *Medical Image Analysis*, 13(1):143 – 155, 2009.
- [13] M. Everingham, L. V. Gool, C. W. , J. Winn, and A. Zisserman. The Pascal Visual Object Classes Challenge 2010 (VOC2010) Results.
- [14] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [15] D. Freedman and T. Zhang. Interactive Graph-Cut Based Segmentation With Shape Priors. In *Conference on Computer Vision and Pattern Recognition*, pages 755–62, 2005.
- [16] J. Gonfaus, X. Boix, J. Weijer, A. Bagdanov, J. Serrat, and J. Gonzalez. Harmony Potentials for Joint Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 3280–87, 2010.

- [17] D. Greig, B. Porteous, and A. Seheult. Exact Maximum a Posteriori Estimation for Binary Images. *Journal of the Royal Statistical Society*, 51:271–279, 1989.
- [18] C. Grigorescu and N. Petkov. Distance Sets for Shape Filters and Shape Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):1274–1286, 2003.
- [19] G. Hjaltason and H. Samet. Properties of Embedding Methods for Similarity Searching in Metric Spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):530–49, 2003.
- [20] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The Itk Software Guide*.
- [21] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, K. Briggman, W. Denk, J. Mendenhall, W. Abraham, K. Harris, N. Kasthuri, K. Hayworth, R. Schalek, J. Tapia, J. Lichtman, and H. Seung. Boundary Learning by Optimization with Topological Constraints. In *Conference on Computer Vision and Pattern Recognition*, pages 2488–95, 2010.
- [22] V. Jain, J. Murray, F. Roth, S. Turaga, V. Zhigulin, K. Briggman, M. Helmstaedter, W. Denk, and H. Seung. Supervised Learning of Image Restoration with Convolutional Networks. In *International Conference on Computer Vision*, pages 1–8, 2007.
- [23] V. Jain, H. S. Seung, and S. Turaga. Machines that Learn to Segment Images: A Crucial Technology for Connectomics. *Current Opinion in Neurobiology*, 20:1–14, 2010.
- [24] E. Jurrus, A. Paiva, S. Watanabe, J. Anderson, R. Whitaker, B. Jones, R. Marc, and T. Tasdizen. Detection of Neuron Membranes in Electron Microscopy Images Using a Serial Neural Network Architecture. *Medical Image Analysis*, 14(6):770–783, 2010.
- [25] V. Kaynig, T. Fuchs, and J. Buhmann. Neuron Geometry Extraction by Perceptual Grouping in ssTEM Images. In *Conference on Computer Vision and Pattern Recognition*, pages 2902–09, 2010.
- [26] A. Knott, G. Perkins, R. Schwarzenbacher, and E. Bossy-Wetzel. Mitochondrial Fragmentation in Neurodegeneration. *Nature Reviews. Neuroscience*, 9(7):505–18, 2008.
- [27] G. Knott, H. Marchman, D. Wall, and B. Lich. Serial Section Scanning Electron Microscopy of Adult Brain Tissue Using Focused Ion Beam Milling. *Journal of Neuroscience*, 28(12):2959–64, 2008.
- [28] V. Kolmogorov and Y. Boykov. What Metrics Can Be Approximated by Geo-Cuts, or Global Optimization of Length/area and Flux. In *International Conference on Computer Vision*, pages 564–71, 2005.
- [29] V. Kolmogorov and C. Rother. Minimizing Nonsubmodular Functions With Graph Cuts-A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1274–1279, 2007.
- [30] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized Via Graph Cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.
- [31] N. Komodakis, G. Tziritas, and N. Paragios. Fast, Approximately Optimal Solutions for Single and Dynamic MRFs. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [32] A. Kreshuk, C. N. Straehle, C. Sommer, U. Koethe, G. Knott, and F. Hamprecht. Automated Segmentation of Synapses in 3D EM Data. In *ISBI*, pages 220–223. IEEE, 2011.
- [33] R. Kumar, A. Vazquez-Reina, and H. Pfister. Radon-like Features and their Application to Connectomics. In *Workshop on Mathematical Methods in Biomedical Image Analysis*, 2010.
- [34] D. Lee, K. Lee, W. Ho, and S. Lee. Target Cell-Specific Involvement of Presynaptic Mitochondria in Post-Tetanic Potentiation at Hippocampal Mossy Fiber Synapses. *The Journal of Neuroscience*, 27(50):13603–13, 2007.
- [35] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi. Turbopixels: Fast Superpixels Using Geometric Flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2290–97, 2009.
- [36] A. Lucchi, K. Smith, A. Radhakrishna, V. Lepetit, and P. Fua. A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images. In *Conference on Medical Image Computing and Computer Assisted Intervention*, pages 463–71, September 2010.
- [37] M. Kumar, P. Torr, and A. Zisserman. OBJ Cut. In *Conference on Computer Vision and Pattern Recognition*, pages 18–25, 2005.
- [38] T. Mäenpää. *The Local Binary Pattern Approach to Texture Analysis-Extensions and Applications*. University of Oulu, Oulu Finland, 2003.
- [39] C. Mannella, M. Marko, and K. Buttle. Reconsidering Mitochondrial Structure: New Views of an Old Organelle. *Trends Biochem. Sci.*, 15:37–38, 1997.
- [40] B. Marsh, D. Mastronarde, K. Buttle, K. Howell, and J. McIntosh. Organellar Relationships in the Golgi Region of the Pancreatic Beta Cell Line, HIT-T15, Visualized by High Resolution Electron Tomography. *Proc. Nat. Acad. Sci.*, 98:2399–2406, 2001.
- [41] E. Mortensen and W. Barrett. Interactive Segmentation With Intelligent Scissors. *Graphical Models and Image Processing*, 60:349–384, 1998.
- [42] N. Vu and B. Manjunath. Graph Cut Segmentation of Neuronal Structures from Transmission Electron Micrographs. In *International Conference on Image Processing*, pages 725–728, 2008.
- [43] R. Narasimha, H. Ouyang, A. Gray, S. McLaughlin, and S. Subramaniam. Automatic Joint Classification and Segmentation of Whole Cell 3D Images. *Pattern Recognition*, 42:1067–1079, 2009.
- [44] H. Nguyen and Q. Ji. Shape-Driven Three-Dimensional Watersnake Segmentation of Biological Membranes in Electron Tomography. *IEEE Transactions on Medical Imaging*, 27(5):616–628, 2008.
- [45] H. Peng, Z. Ruan, F. Long, J. Simpson, and E. Myers. V3D Enables Real-Time 3D Visualization and Quantitative Analysis of Large-Scale Biological Image Data Sets. *Nature Biotechnology*, 28(4):348–353, 2010.
- [46] A. Poole, R. Thomas, L. Andrews, H. McBride, A. Whitworth, and L. Pallanck. The PINK1/Parkin Pathway Regulates Mitochondrial Morphology. *Proceedings of the National Academy of Sciences of the United States of America*, 105(5):1638–43, 2008.
- [47] M. Prasad, A. Zisserman, A. Fitzgibbon, M. Kumar, and P. Torr. Learning Class-Specific Edges for Object Detection and Segmentation. In *Indian Conference on Computer Vision, Graphics and Image Processing*, pages 94–105, 2006.
- [48] A. Radhakrishna, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic Superpixels. Technical report, EPFL, June 2010.
- [49] W. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [50] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- [51] K. Smith, A. Carleton, and V. Lepetit. Fast Ray Features for Learning Irregular Shapes. In *International Conference on Computer Vision*, pages 397–404, 2009.
- [52] C. Sommer, C. Straehle, U. Koethe, and F. Hamprecht. Interactive Learning and Segmentation Tool Kit. In *Systems Biology of Human Disease*, pages 230–33, 2010.
- [53] E. Tola, V. Lepetit, and P. Fua. Daisy: an Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5):815–830, 2010.
- [54] S. Turaga, J. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. Seung. Convolutional Networks can Learn to Generate Affinity Graphs for Image Segmentation. *Neural Computation*, 22:511–538, 2010.
- [55] A. Vazquez-Reina, M. Gelbart, D. Huang, J. Lichtman, E. Miller, and H. Pfister. Segmentation Fusion for Connectomics. In *International Conference on Computer Vision*, 2011.
- [56] A. Vedaldi and S. Soatto. Quick Shift and Kernel Methods for Mode Seeking. In *European Conference on Computer Vision*, pages 705–18, 2008.
- [57] O. Veksler, Y. Boykov, and P. Mehrani. Superpixels and Supervoxels in an Energy Optimization Framework. In *European Conference on Computer Vision*, pages 211–224, 2010.
- [58] K. Venkataraju, A. Paiva, E. Jurrus, and T. Tasdizen. Automatic Markup of Neural Cell Membranes using Boosted Decision Stumps. In *IEEE Symposium on Biomedical Imaging: From Nano to Macro*, pages 1039–42, 2009.
- [59] S. Vitaladevuni, Y. Mishchenko, A. Genkin, D. Chklovskii, and K. Harris. Mitochondria Detection in Electron Microscopy Images. In *Workshop on Microscopic Image Analysis with Applications in Biology*, 2008.
- [60] J. Yedidia, W. Freeman, and Y. Weiss. *Understanding Belief Propagation and Its Generalizations*, pages 239–269. Morgan Kaufmann Publishers Inc., 2003.