# Exploiting Enclosing Membranes and Contextual Cues for Mitochondria Segmentation

Aurélien Lucchi[2,⋆],Carlos Becker[1], Pablo Márquez Neila[1], and Pascal Fua[1]

[1] Computer Vision Laboratory, EPFL, Lausanne, Switzerland
[2] Department of Computer Science, ETHZ, Zürich, Switzerland

**Abstract.** In this paper, we improve upon earlier approaches to segmenting mitochondria in Electron Microscopy images by explicitly modeling the double membrane that encloses mitochondria, as well as using features that capture context over an extended neighborhood. We demonstrate that this results in both improved classification accuracy and reduced computational requirements for training.

## 1 Introduction

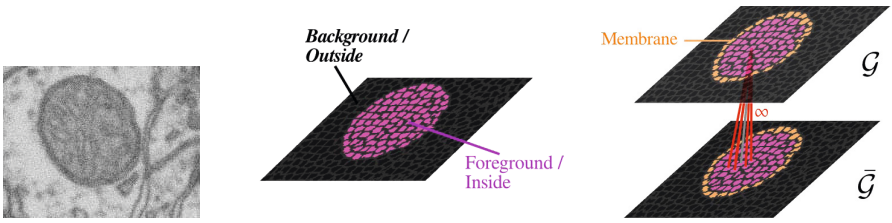
In addition to providing energy to the cell, mitochondria play an important role in many essential cellular functions including signaling, differentiation, growth and death. An increasing body of research suggests that regulation of mitochondrial shape is crucial for cellular physiology [1]. Furthermore, 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 functioning [2] and mounting evidence also indicates that there is a close link between mitochondrial function and many neuro-degenerative diseases [3, 4].

Since mitochondria range from less than 0.5 to 10 $\mu$m in diameter [5], block face scanning microscopes and their ability to image with isotropic resolution of up to $4nm$ are proved invaluable tools to study their exact structure. As a result, new approaches to analyzing the images they produce have begun to appear. For example, in [6] a Gentle-Boost classifier was trained to detect mitochondria based on textural features. In [7], texton-based mitochondria classification in melanoma cells was performed using a variety of classifiers including k-NN, SVM, and Adaboost. While these techniques achieve reasonable results, they incorporate only textural cues while ignoring shape information. More recently, more sophisticated features [8–10] have been successfully used in conjunction with either a Random Forest classifier [11] or a Structured SVM (SSVM) [12, 13]. The latter approach [12, 13] is state-of-the-art in terms of accuracy. In this paper, we show that it can be further improved by

- **Explicitly modeling membranes.** At the resolution we are working with, mitochondria have a clearly visible double membrane, as shown in Figure 1.

⋆ This work was accomplished while the author was in the Computer Vision Lab at EPFL and supported in part by the MicroNano ERC project.

P. Golland et al. (Eds.): MICCAI 2014, Part I, LNCS 8673, pp. 65–72, 2014.




(a) Input image (b) 2-class CRF (c) 3 class-CRF

**Fig. 1.** Input image (a slice through a 3D volume) shown in (a). Figure (b) shows the graph used for a standard 2-class CRF commonly used in segmentation [12, 13]. The pink and black colors correspond to the foreground and background classes while the gray color in (b) and (c) shows the boundary of the SLIC supervoxels. The 3-class CRF introduced in Section 2.1 is shown in (c). The outer layer of supervoxels originally labeled as foreground in (b) was converted to a third boundary class shown in orange.

Voxels can therefore be classified as being inside, between the two membranes, or outside. This three-class problem can be formulated so that the membrane class completely encloses the inside and can be solved exactly using the maxflow-mincut approach of [14], which makes it faster than having to rely on Belief Propagation as in [12, 13].

– **Introducing context-based features.** One of the difficulties with mitochondria segmentation is that purely local statistics are not informative enough. As a result, mitochondria voxels are easily confused with others such as those belonging to vesicles and context information has to be used for disambiguation purposes. In [12], this is done by using a linear SSVM with a non-linear transformation applied to the features. However, this approach has a very high worst case computational complexity. We will show that a better result can be obtained at a tenth of the computational cost by exploiting the ability of AdaBoost to process large amounts of training data to learn features that take into account extended neighborhoods around individual voxels [15] and using them to compute the data term of the above maxflow-mincut problem.

We will show on several datasets that this combination allows us not only improve upon the state-of-the-art in terms of accuracy but also to considerably speed-up the training and running times of our algorithms, which is significant when dealing with large amounts of data.

## 2 Method

As in [9], the first step of our approach is to over-segment the image stack into *supervoxels*, that is, small voxel clusters with similar intensities. The algorithm we use to compute them [16] lets us choose their approximate diameter, which we

take to be on the order of the known thickness of the outer mitochondrial membranes. As can be seen in Fig. 1(c), this means that membranes are typically one supervoxel thick. All subsequent computations are performed on supervoxels instead of individual voxels, which speeds them up by several orders of magnitude. Our task is now to classify these supervoxels as being inside the mitochondria, part of the membrane, or outside, as shown in Fig. 1(c). To this end, we introduce a three-class Conditional Random Field (CRF) described below.

### 2.1 Multi-class Conditional Random Fields

CRF [17] are graphical models used to encode relationships between a set of input and output variables. The one we use here is defined over a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ whose nodes $i \in \mathcal{V}$ correspond to supervoxels and whose edges $(i, j) \in \mathcal{E}$ connect nodes $i$ and $j$ if they are adjacent in the 3D volume. Each node is associated to a feature vector $x_i$ computed from the image data and a label $y_i$ denoting one of the three classes to which a supervoxel can belong. Let $Y$ be the vector of all $y_i$, which we will refer to as a *labeling*. The most likely labeling of a volume is then found by minimizing an objective function of the form

$$E^{\mathbf{w}}(Y) = \sum_{i \in \mathcal{V}} D_i^{\mathbf{w}}(y_i) + \sum_{(i,j) \in \mathcal{E}} V_{ij}^{\mathbf{w}}(y_i, y_j), \tag{1}$$

where $D_i$ is referred to as the unary data term and $V_{ij}$ as the pairwise term. The superscript denotes the dependency of these two terms to a parameter vector $\mathbf{w}$.

The unary data term $D_i$ is a weighted sum of image features described in Section. 2.2. The pairwise term is a linear combination of a spatial regularization term [12, 13] and a containment term. The spatial term is learned from data and reflects the transition cost between nodes $i$ and $j$ from label $y_i$ to label $y_j$. The containment term constrains the membrane class to completely enclose the inside class and to be at least one supervoxel thick, as originally proposed in [14]. As shown in Fig. 1(c), this is achieved by duplicating the graph $\mathcal{G}$ to $\bar{\mathcal{G}}$ and adding infinite cost edges emanating from voxels labeled as inside in $\mathcal{G}$ to the neighbors labeled as membrane or inside in $\bar{\mathcal{G}}$ (see red edges in Fig. 1(c)). This infinite cost effectively prohibits inside nodes to be next to outside nodes. The containment term is hand-defined and thus does not depend on any parameters. The set of parameters $\mathbf{w}$ to be learned are therefore the weights given to individual features in the unary term and the spatial regularization term. These parameters are learned with the Structured SVM (SSVM) framework of [13] that requires solving an inference problem on the supervoxel graph. The method of [14] greatly speed-ups this inference step by using graph-cuts instead of Belief-propagation.

### 2.2 Data term

In this section, we first briefly review the standard features used in the data term of competing approaches [8, 13] before introducing the contextual features we advocate using instead.

**Standard Features.** As the baseline, we used standard features found in the literature [8, 13] that capture local shape and texture information at each supervoxel. The features extracted are voxel intensity histograms and gradient magnitude, Laplacian of Gaussian, eigenvalues of the Hessian matrix, and eigenvalues of the structure tensor, computed at five different scales. The feature vectors consist of the concatenated features for the supervoxel of interest and those corresponding to its neighbors (adjacent supervoxels in the 3D volume).

**Contextual Features.** Even though the features described above can be computed efficiently and take surrounding supervoxels into account, this is done in a predetermined manner and over a limited spatial extent.

An early attempt at incorporating contextual information from further afield is found in [9], where Ray Features [18] were used to capture information about the mitochondria shape. Unfortunately, they rely on computing image gradients, which can be noisy. We have found experimentally that adding them into our CRF framework that already contains the standard features described above only had minimal impact.

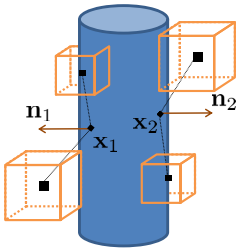



**Fig. 2.** Contextual Features. Given a mitochondria represented by the blue cylinder, the context surrounding voxels $\mathbf{x}_1$ and $\mathbf{x}_2$ is captured by summing responses of different channels inside cubes whose size and position with respect to the voxel is learned from training data.

Instead, we advocate here using the context-aware features first introduced in [15] for synapse segmentation[1] and demonstrate that they can be adapted for a different purpose and are therefore much more generic that initially claimed. These *context cues* capture context information in an extended neighborhood around voxels of interest by summing responses of different channels inside arbitrary-sized cubes, as depicted in Fig. 2. The extent of the neighborhood is learned by boosting up to a maximum size of 80 voxels. The location of each cube is relative to the voxel of interest and to the orientation estimate $\mathbf{n}$ at that point, computed from the Hessian matrix eigenvectors [15]. These locations and corresponding channels are learned automatically by running AdaBoost on the training data, which requires almost no parameter tuning. Since the number of possible context cue features can be in the order of hundreds of thousands, using AdaBoost is key to selecting a small subset of them based on training data.

To integrate these features into our CRF model, we treat the output of each one of the 1200 weak learners that compose the final AdaBoost classifier as a feature vector component for the unary data term. We then re-learn weights for the weak learners that are optimal when used in conjunction with the pairwise term of Eq. (1).

[1] Code publicly available at `http://cvlab.epfl.ch/software/synapse`

## 3 Experimental Results

To validate our approach we used the two large labeled Electron Microscopy image stacks depicted in Fig. 3. The first one is publicly available[2] and represents a $1024 \times 1024 \times 165$-voxel volume of 5nm voxel size from the *CA1 hippocampus*. The second stack comes from the *striatum*, a subcortical brain region. It is of size $711 \times 872 \times 318$ and of voxel size $6 \times 6 \times 7.8$ nm. Each stack is divided into two equally-sized sub-volumes, one for training and the other one for testing.

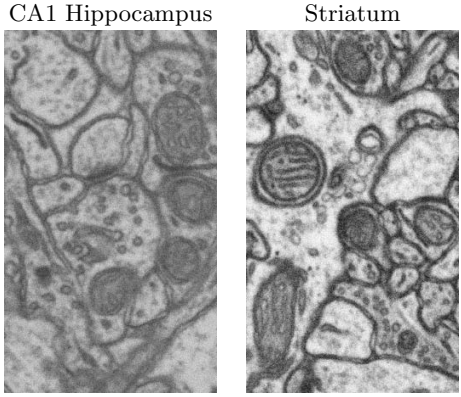



**Fig. 3.** *EM data sets.* Slices cut from two EM stacks used for evaluation. Mitochondria are indicated with black arrows.

Performance is measured in terms of the *Jaccard index*, commonly used for image segmentation [9, 13, 15, 19]. We report the voxel-based *Jaccard index* for the foreground class, which is representative for this task since the mitochondria are the object of interest being segmented. The multi-class CRF returns predictions for the membrane class which can be of particular interest for biologists. We treat it as part of the foreground class for quantitative evaluation purposes so as to facilitate comparison with the other methods, which produce only binary foreground/background labels.

The performance for the different baselines on the test set is summarized in Table 1. We report results when using standard features from Sec. 2.2 (*Std.*), their kernelized version (*Kernel.*) introduced in [12], or the context cues of Section 2.2 (*Ccues*). As described in [12], kernelizing means transforming the features non-linearly using a 2-step approach. First, we train a non-structured kernel SVM using the standard features extracted from $N = 40000$ randomly sampled supervoxels. This yields a set of support vectors that are then used to compute new feature vectors whose components are the kernel distances of the original feature vectors to the support vectors. The three types of features are fed to different classifiers, either two or three-class SSVM or AdaBoost. The *2-class* model minimizes the energy as Eq. 1 but without the containment term.

From Table 1 it can be observed that our approach with *3-class* and context cues outperforms the others, especially those that use the 2-class model and ignore the membrane prior. The next best result is obtained using the *3-class* model with kernelized features, followed by the *2-class* one with context cue features. We attribute the good performance of the 3-class model to two reasons. First, at the 5 nm resolution we are working with, membranes have a

[2] `http://cvlab.epfl.ch/data/em`

**Table 1.** Segmentation performance measured with the Jaccard index of the foreground class for two EM datasets. We report results for different set of features (*Std.*, *Kernel.*, *Ccues*) and different classifiers (*2-class* CRF, *3-class* CRF and Adaboost).

| | Std. + 2-class | Std. + 3-class | Kernel. + 2-class | Kernel. + 3-class | Ccues + AdaBoost | Ccues + 2-class | Ccues + 3-class |
|---|---|---|---|---|---|---|---|
| Hippocampus | 67.6% | 68.9% | 71.7% | 72.3% | 69.5% | 72.8% | **74.1%** |
| Striatum | 79.3% | 82.5% | 80.8% | 83.4% | 79.3% | 83.2% | **84.6%** |

visible extent and the voxels within them form texture patterns that are different from those inside. Treating the inside and membrane voxels as one single class is therefore a more complex learning task. Furthermore, this specific 3-class problem allows for exact inference and therefore does not incur the penalty of having approximate inference as would have to be done in generic 3-class problems.

As observed in [15], hand-drawn ground truth near mitochondria borders is not always very accurate. As a result, even correctly labeled voxels near the boundary may impact the Jaccard index negatively due to annotation errors. To eliminate this undesirable effect, as in [15], we add an exclusion zone around the mitochondria border for evaluation purposes and report results as a function of its width. The resulting plots for the two top-performing approaches and the 2-class baseline are shown in Fig. 4. Note that our method outperforms the others independently of the exclusion zone width, achieving a difference of up to 10% in Jaccard index with respect to the 2-class approach.

Example segmentation outputs are shown in Fig. 5, where it can be seen that the results of the Ccues + 3-class approach are more accurate than other methods that fail to detect some mitochondria or erroneously insert extra ones.

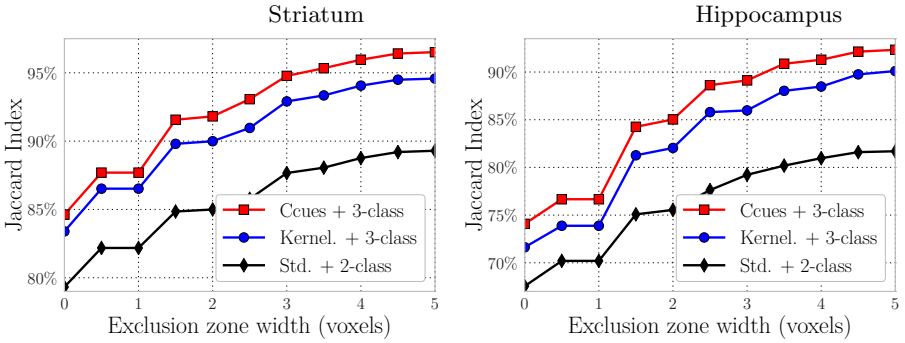



**Fig. 4.** Jaccard index as a function of the exclusion zone width, used to mitigate annotation errors closer to the mitochondria membrane in the ground truth. Our approach enforcing consistently outperforms the others.

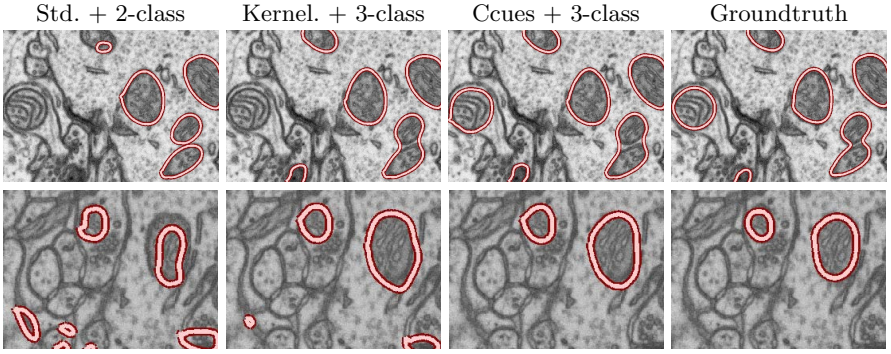



**Fig. 5.** Segmentation results on the Striatum (top row) and Hippocampus (bottom row) datasets. The Ccues + 3-class method correctly segments all mitochondria in this example, while other methods fail to detect some mitochondria or erroneously insert extra regions. The images above correspond to slices from the test volume and are best viewed in color. The 3D results are shown in the supplementary material.

**Table 2.** Training time in minutes for $T = 1000$ iterations. Note that using kernelized features increases training time by almost an order of magnitude.

| | Std. + 2-class | Std. + 3-class | Kernel. + 3-class | Ccues + AdaBoost | Ccues + 3-class |
|---|---|---|---|---|---|
| Hippocampus | 3809 | 275 | 2365 | 96 | 265 |
| Striatum | 4530 | 213 | 2311 | 102 | 282 |

We conducted a time analysis of the different methods evaluated in this paper. We ran each method on a 8-core Intel Xeon CPU 2.4 GHz machine with 200 GB RAM. As shown in Table 2, the 3-class models are much faster to train than the 2-class models. The total number of training points is of the order of $860K$ and $820K$ for the Hippocampus and Striatum datasets. Yet, we could only use a maximum of $40K$ to train this approach in a reasonable time.

## 4 Conclusion

We presented a segmentation framework that exploits discriminative contextual features and a CRF model with geometric constraints to model organelles with enclosing membranes. We demonstrated that it produces superior performance in the specific case of mitochondria, though this approach is generic and could be applied to many other biological structures such as the wide array of cells present in all living creatures. The code and datasets used in this paper is available at `www.cvlab.epfl.ch`.

## References


1. Campello, S., Scorrano, L.: Mitochondrial Shape Changes: Orchestrating Cell Pathophysiology. EMBO Reports 11(9), 678–684 (2010)
2. Lee, D., Lee, K., Ho, W., Lee, S.: Target Cell-Specific Involvement of Presynaptic Mitochondria in Post-Tetanic Potentiation at Hippocampal Mossy Fiber Synapses. The Journal of NeuroScience 27(50), 13603–13613 (2007)
3. Knott, A., Perkins, G., Schwarzenbacher, R., Bossy-Wetzel, E.: Mitochondrial Fragmentation in Neurodegeneration. Nature Reviews. Neuroscience 9(7), 505–518 (2008)
4. Poole, A., Thomas, R., Andrews, L., Mcbride, H., Whitworth, A., Pallanck, L.: The Pink1/parkin Pathway Regulates Mitochondrial Morphology. Proceedings of the National Academy of Sciences of the United States of America 105(5), 1638–1643 (2008)
5. Campbell, N., Williamson, B., Heyden, R.: Biology: Exploring Life. Pearson Prentice Hall (2006)
6. Vitaladevuni, S., Mishchenko, Y., Genkin, A., Chklovskii, D., Harris, K.: Mitochondria Detection in Electron Microscopy Images. In: Workshop on Microscopic Image Analysis with Applications in Biology (2008)
7. Narasimha, R., Ouyang, H., Gray, A., McLaughlin, S., Subramaniam, S.: Automatic Joint Classification and Segmentation of Whole Cell 3D Images. PR 42, 1067–1079 (2009)
8. Sommer, C., Straehle, C., Koethe, U., Hamprecht, F.: Interactive Learning and Segmentation Tool Kit. In: Systems Biology of Human Disease, pp. 230–233 (2010)
9. Lucchi, A., Smith, K., Achanta, R., Knott, G., Fua, P.: Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features. TMI 31(2), 474–486 (2011)
10. Kumar, R., Vazquez-Reina, A., Pfister, H.: Radon-Like Features and Their Application to Connectomics. In: Workshop on MMBIA (2010)
11. Kreshuk, A., Straehle, C.N., Sommer, C., Koethe, U., Knott, G., Hamprecht, F.: Automated Segmentation of Synapses in 3D EM Data. In: ISBI (2011)
12. Lucchi, A., Li, Y., Smith, K., Fua, P.: Structured Image Segmentation Using Kernelized Features. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 400–413. Springer, Heidelberg (2012)
13. Lucchi, A., Li, Y., Fua, P.: Learning for Structured Prediction Using Approximate Subgradient Descent with Working Sets. In: CVPR (June 2013)
14. Delong, A., Boykov, Y.: Globally Optimal Segmentation of Multi-Region Objects. In: ICCV, pp. 285–292 (2009)
15. Becker, C., Ali, K., Knott, G., Fua, P.: Learning Context Cues for Synapse Segmentation. TMI 32(10), 1864–1877 (2013)
16. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Suesstrunk, S.: SLIC Superpixels Compared to State-Of-The-Art Superpixel Methods. PAMI 34(11), 2274–2281 (2012)
17. Lafferty, J., Mccallum, A., Pereira, F.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: ICML (2001)
18. Smith, K., Carleton, A., Lepetit, V.: Fast Ray Features for Learning Irregular Shapes. In: ICCV, pp. 397–404 (2009)
19. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge (VOC 2010) Results (2010)