

Refining Mitochondria Segmentation in Electron Microscopy Imagery with Active Surfaces

Anne Jorstad, Pascal Fua

Ecole Polytechnique Fédérale de Lausanne

Abstract. We present an active surface-based method for refining the boundary surfaces of mitochondria segmentation data. We exploit the fact that mitochondria have thick dark membranes, so referencing the image data at the inner membrane can help drive a more accurate delineation of the outer membrane surface. Given the initial boundary prediction from a machine learning-based segmentation algorithm as input, we compare several cost functions used to drive an explicit update scheme to locally refine 3D mesh surfaces, and results are presented on electron microscopy imagery. Our resulting surfaces are seen to fit very accurately to the mitochondria membranes, more accurately even than the available hand-annotations of the data.

1 Introduction

Meaningful statistics regarding the volumes, surface areas, and lengths of sub-cellular structures such as mitochondria and synapses in the brain are key to allowing neuroscientists to compare these objects in healthy people and those with degenerative brain diseases and to understand brain connectivity. These statistics require not just object detection, but also very accurate surface structure delineation.

State-of-the-art image segmentation algorithms such as [6, 1, 7, 10] produce reasonably good localization results, in that they are able to detect most instances of the object they are searching for, and provide a rough outline. However, they often fail to accurately define the detailed boundary surface of the object in question to the precision required for accurate geometric measurements, especially surface area. In this paper, we propose an explicit active surface scheme to improve the precision of segmentation results in the case of mitochondria in 3D electron microscopy (EM) imaging. To this end, we take into account their thick membranes, as seen in Figure 1, and compare several cost functions used to drive the local surface refinement.

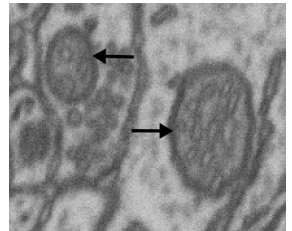


Fig. 1. Mitochondria have thick membranes.

Our method models the outer and inner membrane boundaries as active surfaces [5] in a 3D volume. Tied together by a spring-like proximity energy

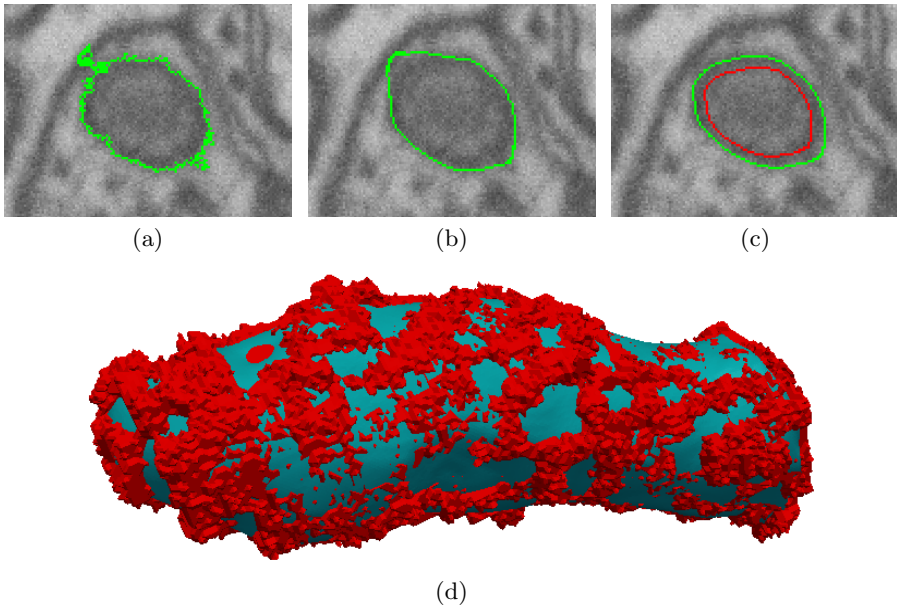


Fig. 2. Surface refinement process. (a) Cross section of 3D input segmentation before smoothing. (b) After Gaussian smoothing. (c) After running our algorithm. (d) Resulting 3D surface shown in blue with 3D input shown in red.

function, the surfaces are updated by minimizing energy terms defined by the image data at the outer and inner surfaces. The initial input to our algorithm is the output segmentation results from methods that use feature-based machine learning to predict whether or not a given voxel is contained within a structure of interest. As these objects are generally computed on supervoxels, the resulting output segmentations tend to be noisy and coarse, as shown in Figure 2. Often times, the final step of these algorithms involves simply smoothing the results so that they appear more visually pleasing, but this does not capture many of the fine surface details. Computing the correct 3D surfaces, whose surface areas will be meaningful for neuroscience applications, requires returning to the image data to perform further local surface updates, as we propose.

Related work includes [12], which searches for membranes in medical images, but works only on 2D slices. The cost function in this model defines a minimum and maximum membrane thickness in 2D, between which there is no penalty, whereas we know a precise value of mitochondria membrane thickness in 3D, and perform our calculations accordingly in 3D. In [11], the best fit to a deformable membrane template of a certain topology is sought, but in our case a single or double surface update scheme that allows each surface to align with small detailed gradients in the image is sufficient. Shape priors are used successfully in [8], but in this work we focus on surface image information alone.

2 Method

Our goal is to precisely capture the true 3D boundaries of mitochondria surfaces, starting from the well-localized but rough binary segmentation data provided by [10]. The initial rough surfaces are Gaussian smoothed, but the resulting objects do not capture fine surface detail. We use an explicit active surface method that starts from an initialization of surface points as a 3D mesh and iterates in a gradient descent energy minimization scheme until either a local minimum is reached or a maximum number of iterations has been computed. We define energy functions that are based on the gradients of the image data near the membrane boundaries, and is also dependent on the proximity to a second surface tracing the inner boundary of the membrane.

The boundary of a mitochondrion is defined by a thick dark membrane. The most discriminatory image data exists at the outside boundary of this membrane, which is a local image gradient maximum, and can be found by minimizing $\frac{1}{1+\|\nabla I\|}$, as in [3]. The inner boundary of the membrane is also a local minimum of this function, but is a much weaker boundary than the outer membrane, as can be seen in Figure 1. Although it is weaker, considering image data from the inner surface can contribute to defining a more robust final outer surface. We will compare active surface schemes that jointly update both an inner and an outer surface, where each surface incorporates varying amounts of information from the current position of the other.

It is known from biology that organelles are generally smooth objects, and it is important that the final surfaces be smooth so that accurate surface areas of the objects can be computed and compared. We achieve this objective by a standard application of Laplacian smoothing as in [13], which minimizes the curvature of the surface, and where at each update, every point is redefined to be a weighted average of itself and its neighbors. At each iteration k over the mesh of 3D points X , we update

$$X^{k+1} = A^{-1} (\gamma X^k - w \nabla E^{\text{sum}}(X^k)), \quad (1)$$

where $E^{\text{sum}}(X) = \sum_{\mathbf{x}} E(\mathbf{x})$ is the cost function defined at each point to be minimized as described below, the matrix $A = \gamma I + \alpha L_n$, for relative weighting parameters γ and α , and L_n is the normalized graph Laplacian ($L(i, i) = 1$, $L(i, j) = \frac{-1}{\|\mathcal{N}_{v_i}\|_0}$, where $\|\mathcal{N}_{v_i}\|_0$ is the number of neighbors of vertex v_i). The multiplication by A^{-1} is performed in practice by LU decomposition. This smoothing operation is applied at the end of each iteration, updating the new point locations relative to those of their neighbors.

The directional force of each iteration is defined by an energy function depending on the image data, and also on the proximity between the inner and outer membrane surfaces. We will compare four energy functions E_I based on the image data at each surface point \mathbf{x} , one of which iterates towards regions of large image gradient (Eq. 2), while the others additionally take into consideration the

direction of the image gradient:

$$E_I^1(\mathbf{x}) = \frac{1}{1 + \|\nabla I(\mathbf{x})\|}, \quad (2)$$

$$E_I^2(\mathbf{x}) = \frac{1}{1 + \|\nabla I(\mathbf{x})\|} \left(\arccos \left(\hat{n}(\mathbf{x}) \cdot \widehat{\nabla I}(\mathbf{x}) \right) + \delta \right), \quad (3)$$

$$E_I^3(\mathbf{x}) = -\hat{n}(\mathbf{x}) \cdot \nabla I(\mathbf{x}), \quad (4)$$

$$E_I^4(\mathbf{x}) = \frac{1}{1 + \|\nabla I(\mathbf{x})\|} \left(1 - \left(\hat{n}(\mathbf{x}) \cdot \widehat{\nabla I}(\mathbf{x}) \right) * \left(\widehat{\nabla I}(\mathbf{x}) \cdot -\widehat{\nabla I}(\mathbf{x}_{\text{clspt}}) \right) \right). \quad (5)$$

Here $\hat{n}(\mathbf{x})$ is the surface normal vector at \mathbf{x} with unit length, $\widehat{\nabla I}(\mathbf{x})$ is the normalized image gradient direction vector, and $\mathbf{x}_{\text{clspt}}$ is the closest point on the second surface to point \mathbf{x} on the first surface. As described above, minimizing Equation (2) updates the surface towards local image gradient maxima. Equations (3), (4), and (5) additionally encourage surfaces whose normal vectors point in the direction of dark to light in the image. The intuition behind Equation 3 is that when the surface normal is in line with the image gradient, $\hat{n}(\mathbf{x}) \cdot \widehat{\nabla I}(\mathbf{x}) \approx 1$, when they are perpendicular $\hat{n}(\mathbf{x}) \cdot \widehat{\nabla I}(\mathbf{x}) \approx 0$, and when they are pointing in opposite directions, $\hat{n}(\mathbf{x}) \cdot \widehat{\nabla I}(\mathbf{x}) \approx -1$, so \cos^{-1} of this dot product is 0 when the vectors are aligned, with values up to π when they are pointing in opposite directions. Using the \cos^{-1} means that the true angle between the two vectors is measured and penalized linearly. This value is then offset by δ so that it is never 0 and the other term encouraging large gradient magnitudes can still direct the surface updates even when the surface is aligned with the image gradient. In Eq. 4, the image gradient is not normalized, resulting in the simplest expression that penalizes a surface normal that is not aligned with the image gradient, while also progressing towards a large image gradient. From the first dot product of Equation 5, the surface is encouraged to align with the image gradient, while the second dot product encourages the inner and outer surfaces to be locally parallel; this quantity is again scaled by the image gradient magnitude term.

We want to take advantage of the fact that the membrane has both an outer and an inner boundary, so we will consider a cost function that optimizes the two surfaces together. Mitochondria membranes are known to have a very consistent thickness, so we add to the energy function a term that penalizes points on the surfaces that are not the known distance r apart. For each point x_i on surface S_1 , the closest point $\mathbf{x}_{\text{clspt}}$ is found on surface S_2 (and vice versa), and for convenience of notation we define the closest point to \mathbf{x}_i as

$$\mathbf{x}_{\text{clspt}} = \arg \min_{\mathbf{x}_j \in S_2} \|\mathbf{x}_i - \mathbf{x}_j\|. \quad (6)$$

The distance $\|\mathbf{x}_i - \mathbf{x}_{\text{clspt}}\|$ is penalized via a Gaussian centered at the standard membrane thickness, resulting in the proximity cost function:

$$E_P(\mathbf{x}_i \in S_1) = \left(\frac{\|\mathbf{x}_i - \mathbf{x}_{\text{clspt}}\| - r}{\sigma} \right)^2. \quad (7)$$

In order to understand the influence of tracing the inner membrane surface in addition to the outer one, we will also consider a weighting between the inner and outer energies

$$E_I^{\text{total}}(\mathbf{x}_i \in S_1) = (1 - w_{\text{other}})E_I(\mathbf{x}_i) + w_{\text{other}}E_I(\mathbf{x}_{\text{clspt}}) \quad (8)$$

where the weight w_{other} is a value between 0 and 0.5. We note that even if $w_{\text{other}} = 0$ the locations of the two membranes influence each other via the proximity term E_P .

The resulting function to be optimized is therefore

$$E(\mathbf{x}_i \in S_1) = w_I E_I^{\text{total}}(\mathbf{x}_i) + w_P E_P(\mathbf{x}_i), \quad (9)$$

for relative weighting parameters w_I and w_P (and similarly for points on the second surface S_2). In order to calculate the minimizing surface, the gradient descent scheme from Equation 1 is applied, where the gradient term is defined to be

$$\nabla E(\mathbf{x}_i^k) = w_I \nabla \widehat{E}_I^{\text{total}}(\mathbf{x}_i^k) + w_P \nabla \widehat{E}_P(\mathbf{x}_i^k). \quad (10)$$

We note that all energy gradients in our algorithm are normalized so that their relative influence can be adjusted explicitly using the weights w_I , w_P and w_{other} .

3 Discussion

Electron Microscopy data is notoriously difficult to work with, because the texture and intensity variations across all parts of the images are so similar. If the initial segmentation that our algorithm is given is not within a reasonably small radius of the true surface boundary, there is nothing an active surface algorithm making only local updates can do to progress to the true boundary, such as in Figure 3. There are far too many local gradient maxima between the found object segmentation and the true boundary, even after significant smoothing, and these cases are beyond the scope of the local update routines using only image data that are being compared in this study. This type of error must be fixed by a stronger initial segmentation algorithm, and for the purposes of this study, we remove the found mitochondria whose volumes are less than half of the true objects, so that we can understand the strengths of the methods being presented.

The update schemes presented in this work rely on having reasonably strong image gradients to iterate towards. While the outer mitochondria membrane boundaries do have acceptably strong image gradients, so do many other nearby structures that can easily distract from the organelle in question. This is why referencing the inner membrane can help. However, the gradients at the inner membrane boundaries are weak. Instead of using gradients, the proposed method can further benefit from an inner boundary delineation defined by a statistical model based on region texture, such as in [4]. This will be a focus of future work.

The bottleneck of the presented methods is the computation required to find the closest points between surfaces. We use an axis-aligned bounding box tree to

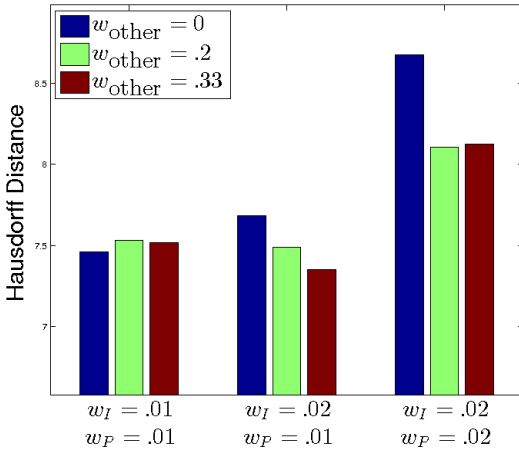


Fig. 5. For parameter verification, Hausdorff Distance results are presented for varying values of w_I , w_P , and w_{other} . All tests are performed using E_I^1 .

testing we used a publicly available dataset¹ that consists of a $1024 \times 768 \times 165$ image stack. We obtained from the authors the output of their algorithm [10], which correctly finds most mitochondria in the stack albeit with fairly rough boundaries. These boundaries were Gaussian smoothed before processing, and our method is then meant to update these smoothed surfaces to capture fine surface details. We first used the standard Marching Cubes algorithm of [9] to generate a 3D mesh corresponding to the outer boundary of each mitochondrion. We then initialized the inner membrane surface by eroding the initial segmentation by the expected membrane thickness. The mesh surfaces are defined by 50,000 vertices on average, but this quantity varies considerably, ranging from 10,000 to 114,000 vertices depending on the size of each mitochondrion and how much of it is contained within the image stack.

Parameter ranges were tuned on an artificial 3D image of a smoothed black ball with a thick boundary on a white background. The fixed parameters used in our experiments are as follows: segmentation smoothing $\sigma = 5$, smoothing for image gradients $\sigma = 5$, membrane thickness $r = 4$, $\sigma = 1$, Laplacian parameters $\gamma = 1$ and $\alpha = .5$, and the number of iterations is 100. In Figure 5 we present a range of results for a single cost function (E_I^1) as the weighting parameters are varied over a small range of values found to perform well. Observing that w_P should be no greater than 0.01, we present the rest of our experiments using the parameters $w_I = 0.01$ and $w_P = 0.01$, but as all nearby parameter values produce similar final results, the methods are seen to be stable. In future work better optimization schemes can be derived that modify the weights throughout the optimization.

¹ <http://cvlab.epfl.ch/data/em>

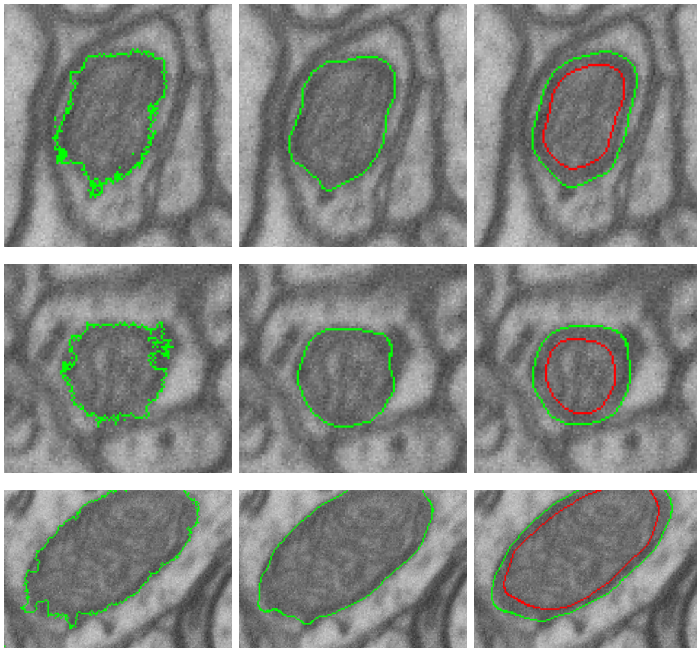


Fig. 6. From noisy initial segmentations (left), we observe that incorporating an outer and inner membrane proximity term (right) into the surface refinement energy function results in a more precise surface extraction than when it is not used (center). See also Figure 2. This result is the primary contribution of this work.

4.1 Qualitative Evaluation

A cross-section of the full segmentation surface refinement results is presented in Figure 10 along with the final surfaces of the whole image stack. The smooth surfaces our method produces are seen to be very consistent with the image data. At this scale, the differences between the four energy function compared is visibly negligible.

We observe that performing surface updates that track both the inner and outer membranes together, and penalizes discrepancies between them, results in more accurate final surface reconstructions. For several examples see Figures 2 and 6. This is the fundamental contribution of this work.

Of the four energy functions we propose, the most basic function E_I^1 performs remarkably well considering that it only depends on the magnitude and not the direction of the image gradients. This function cannot in principle distinguish between the inner and outer membranes, and so should be more prone to iterating towards false membranes. However, it is interesting to note that the other cost functions, which were designed to improve upon E_I^1 , do not appear to perform substantially better. It is observed that E_I^2 and E_I^3 provide very similar results, in spite of the fact that E_I^2 depends on more meaningful quantities. This is likely due to the fact that EM imagery is so noisy that precise numerical rela-

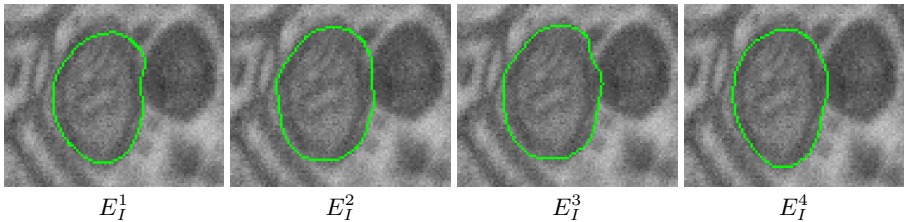


Fig. 7. Example surface refinements from each of the energy functions considered. We find that the function that does not take any directional term into account (E_I^1) and the function that encourages the two surfaces to point in opposite directions (E_I^4) are typically the most accurate.

tions between pixels is not as meaningful as in other contexts. The final energy function E_I^4 that also enforces that the two surfaces should be locally parallel, in addition to being aligned with the image data, performs somewhat better than E_I^2 and E_I^3 . For an example visual comparison of the four energy functions, see Figure 7.

4.2 Quantitative Evaluation

The dataset we use includes hand-annotations provided by an expert that roughly denote the voxels that correspond to mitochondria in the images. As is fairly common in such annotations created frame-by-frame, noticeable jumps can often be observed between consecutive frames because the human annotator was not consistent in deciding from one frame to the next where a mitochondrion boundary ended, for example see Figure 8. This implies that, while sufficient for large-scale coarse mitochondria segmentation, the hand annotations are not perfectly accurate. We attempted to improve the hand annotations by going through each frame of a mitochondrion and carefully marking each boundary pixel, a process that takes about two minutes per frame, or about five hours per mitochondrion. This process is not tenable for a data set of any reasonable size. We also attempted to obtain more precise segmentations using the 3D carving tool provided in *ilastik* 1.0², the Interactive Learning and Segmentation Toolkit, which implements the seeded watershed algorithm to perform interactive image segmentation. Extracting the object boundaries in 3D instead of going through individual 2D frames alleviates the problem of frame-by-frame discrepancies that were demonstrated in Figure 8. Unfortunately, this tool, similar to any automatic segmentation tool, relies on object boundaries having strong image gradients, which is simply not always the case in EM imagery, as described above. The resulting “carved out” objects include sections of smooth boundaries as desired, but also large regions of grossly incorrect surfaces where the segmentation has ballooned into the surrounding regions, or been caught significantly inside the mitochondrial structure where a stronger gradient has been found, as

² www.ilastik.org

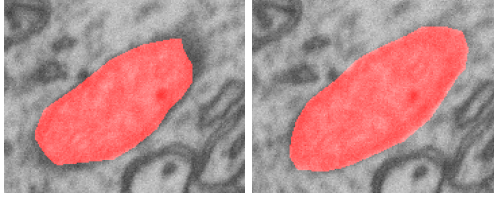


Fig. 8. Two adjacent image slices from the reference hand-annotations. The inconsistency between images slices that comes from labling each slice independently leads to locally inaccurate surfaces. The surfaces constructed by the methods presented in this paper are seen to be more accurate than the provided hand annotations.

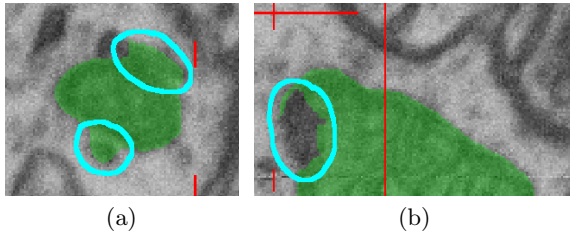


Fig. 9. Examples of the ilastik carving tool (a) oversegmenting and (b) undersegmenting a mitochondrion object. Note that these images are from the same segmentation, and it tends to be impossible to correct the error in one direction without further exacerbating the error in the other. (The red lines are part of the ilastik interface and can be ignored.)

in Figure 9. These segmentations must then be hand-corrected frame-by-frame as before, and the overall processing time at over two hours per mitochondrion is still unmanageable on a large scale.

It is therefore understandable that the precise surface refinements we compute in this work are often visually better than the noisy hand annotations available for reference, yet can be penalized numerically when compared against them. To attempt to alleviate the discrepancy between image frames, the hand annotations are smoothed with the same Gaussian smoothing σ as the images. This means, however, that the initial smoothed segmentations and the hand annotations are both highly smoothed blobs, and are often closer to each other than they are to the fine surface details that they have smoothed across. We find that while our method is observed to consistently arrive at accurate image boundaries, as seen above, comparing our results to the hand annotations does not result in solutions that are numerically as much stronger than the smoothed initial segmentations as we would expect. Nevertheless, for general verification purposes we compare our results with the provided hand annotations.

Image segmentation accuracy is often measured by the Jaccard index, but as we are interested in surface precision and not the volume overlap that the Jaccard index quantifies, we will instead perform our numeric evaluations using a measure of surface similarity. The Hausdorff Distance measures how close two surfaces are

<i>Method</i>	<i>Hausdorff Distance</i>		
Input segmentation	12.518		
Smoothed input segmentation	8.144		
	$w_{\text{other}} = 0$	$w_{\text{other}} = .2$	$w_{\text{other}} = .33$
E_I^1	7.460	7.530	7.518
E_I^2	8.054	8.012	8.017
E_I^3	8.006	8.028	8.071
E_I^4	7.713	7.991	7.987

Table 1. Surface refinement results for each of the energy functions E_I compared to the provided hand annotations, where the values presented represent the average Hausdorff Distance over all mitochondria in the image stack. It is again observed that E_I^1 and E_I^4 perform more strongly than the other two energy functions described.

in space by finding the minimum distance from each point on one surface to the other surface, and vice versa, and then returns the maximum of these minimum distances. The Hausdorff Distance is computed for each mitochondrion in the image stack separately, and the average result is presented in Table 1. We also provide these measurement results on the input data and the Gaussian smoothed version of the input data, which is what we use to initialize our algorithm. These results show that our proposed method does produce an improvement, albeit not as large as visual inspection would indicate. As discussed above, this is essentially an artifact of the inaccuracy of the hand annotations, which are highly smoothed, and therefore very similar to the smoothed input segmentation, while often penalizing our results when they are *more* accurate than those provided. This phenomenon has been previously observed in [2], where in order to avoid the inaccuracies at the boundaries of hand-annotated EM data the authors ignore all voxels near the structure boundaries when comparing segmentations. What we can say from the numerical results is that our method does reduce outliers. We refer back to the qualitative results above for further verification of the accuracy of the presented method.

We observe both qualitatively and quantitatively that the cost functions E_I^1 and E_I^4 perform better than the others. While E_I^1 compares the most positively with the hand annotations, qualitatively it appears that E_I^4 performs at least as well. Numerically it also appears that making use of $w_{\text{other}} > 0$ actually hurts the process. This means that while using the proximity term from Equation 7 is helpful, using the current gradient values of both surfaces to define the direction in which each individual surface should step is actually confusing the process. This is likely due to the fact that the point proximity is not symmetric, and so the corresponding surface points are not always jointly meaningful, as discussed in Section 3.

The original goal of this work was to construct more accurate surfaces so that analyses involving their surface areas can be as precise as possible for biological applications. Unfortunately, the “ground truth” surface area of a mitochondrion simply cannot be obtained, and so we must instead focus on refining surfaces to be as accurate as possible given the image data.

5 Conclusion

We have compared several energy functions used to drive an explicit surface refinement algorithm, with the goal of making local adjustments to the output of mitochondria segmentation algorithms in challenging EM imagery. The well-localized but rough initial segmentation is smoothed, and the methods then update the surfaces towards local surface details. Qualitative and quantitative results are provided that demonstrate the effectiveness of the method. We observe that our method is able to provide mitochondria surface segmentation results that are more accurate than the available hand annotations. The resulting smooth surfaces can be used in biological applications where it is important to be able to accurately calculate and compare the surface areas of mitochondria.

References

1. Becker, C., Ali, K., Knott, G., Fua, P.: Learning Context Cues for Synapse Segmentation in EM Volumes. In: MICCAI (September 2012)
2. Becker, C., Ali, K., Knott, G., Fua, P.: Learning Context Cues for Synapse Segmentation. TMI 32(10), 1864–1877 (October 2013)
3. Caselles, V., Kimme, R., Sapiro, G.: Geodesic Active Contours. In: Computer Vision, 1995. Proceedings., Fifth International Conference on. pp. 694–699 (1995)
4. Chan, T.F., Vese, L.A.: Active Contours Without Edges. In: TIP. pp. 266–277 (2001)
5. He, L., Peng, Z., E., B., Wang, X., Han, C.Y., Weiss, K.L., Wee, W.G.: A Comparative Study of Deformable Contour Methods on Medical Image Segmentation. Image and Vision Computing 26(2), 141–163 (2008)
6. Kreshuk, A., Straehle, C.N., Sommer, C., Koethe, U., Knott, G., Hamprecht, F.: Automated Segmentation of Synapses in 3D EM Data. In: ISBI (2011)
7. Laptev, D., Vezhnevets, A., Dwivedi, S., Buhmann, J.M.: Anisotropic ssTEM Image Segmentation Using Dense Correspondence Across Sections. In: MICCAI. pp. 323–330 (2012)
8. Leventon, M.E., Grimson, W.E., Faugeras, O.: Statistical Shape Influence in Geodesic Active Contours. In: CVPR. pp. 316–323 (2000)
9. Lorensen, W., Cline, H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: SIGGRAPH. pp. 163–169 (1987)
10. Lucchi, A., Li, Y., Fua, P.: Learning for Structured Prediction Using Approximate Subgradient Descent with Working Sets. In: CVPR (June 2013)
11. Prevos, R., Cuingnet, R., Mory, B., D, L.C., Ardon, R.: Incorporating Shape Variability in Image Segmentation via Implicit Template Deformation. MICCAI pp. 82–89 (2013)
12. Vazquez-Reina, A., Miller, E., Pfister, H.: Multiphase Geometric Couplings for the Segmentation of Neuronal Processes. In: CVPR. pp. 2020–2027 (2009)
13. Vollmer, J., Mencl, R., Mueller, H.: Improved Laplacian Smoothing of Noisy Surface Meshes. In: Computer Graphics Forum. vol. 18, pp. 131–138. Wiley Online Library (1999)

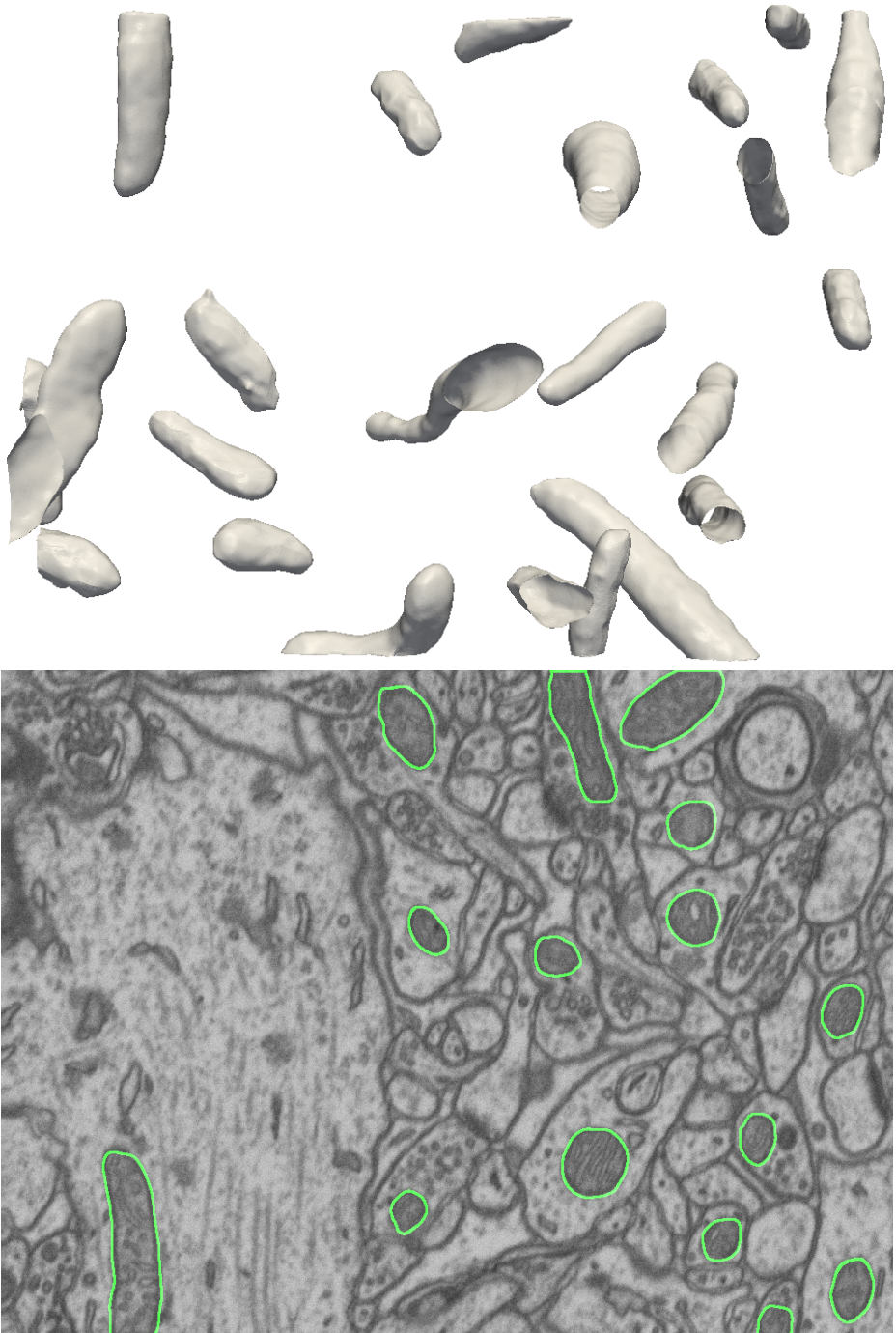


Fig. 10. Final surface fitting results on the full 3D image stack and a cross section.