1 Introduction

New high-throughput microscopy techniques are now capable of generating large multi-terabyte datasets, creating a need for an automatic technique to segment subcellular structures and ultimately perform structural analysis on a large scale. Characterizing the distribution and structure of mitochondria is useful in itself, to identify cellular states such as cancer [1]. Additionally, having an accurate mitochondria detector could potentially increase the accuracy of full cell segmentation in electron microscopy (EM); mitochondria are virtually ubiquitous and tend to introduce ambiguity during automatic detection of cell processes [2]. An accurate mitochondria detector would allow an automatic segmentation routine to clearly distinguish mitochondria membranes from other cell membranes and potentially improve overall cell segmentation in the future.

Automatic segmentation of tomographic reconstructions of neuropil is challenging because the images are densely packed with various components, and algorithms tuned to segment one particular type of component often give incorrect results for images with other types of components present. Further complicating the problem, limited-angle tomographic reconstruction introduces a missing wedge artifact, which makes accurate segmentation difficult, even for a human expert. The goal of this work is to produce an automatic segmentation method that tolerates the artifact inherent to EM tomography and isolates the mitochondria based on both local and global properties of the mitochondria structure.

2 Automatic Segmentation Method

The segmentation process uses three stages, (1) local feature extraction, (2) contour processing, and (3) 3D object segmentation. In stage 1, image enhancement is performed on the original volume to detect voxels that belong to mitochondria. The voxel classifier uses supervised learning with a random forest. In stage 2, contours in X-Y planes of the volume are detected, and a naive Bayes classifier is used to classify the contours as mitochondrion or not, based on a set of contour properties. In stage 3, scaled contours of mitochondria are used to

initialize a geodesic active contour operation (from the ITK[1] library) which produces 3D blobs in an array of voxels. For viewing and editing, the blobs, which represent mitochondria, are then converted to Jinx[2] XML using a Jinx import function and automatically labeled as members of the ontological class Mitochondrion. (Jinx is an ontology-based segmentation and analysis tools for electron tomographic data.)

2.1 Stage 1: Voxel Classification

In stage 1, a voxel classifier is used to assign each voxel a probability that it is part of a mitochondrion. The features in Table 1 were used with a random forest classifier as described in [3]. In Figure 2, views (A) and (B) show the resulting probability map.

2.2 Stage 2: Contour Classification

To detect contours, the image is thresholded at 20%, 40%, 60%, and 80% of the total intensity, yielding a set of four binary volumes. A standard contour detection function (available in OpenCV[3]) was used to identify contours in all the volumes. After detection, contour classification serves to eliminate the extraneous contours that do not belong to mitochondria. A naive Bayes classifier is used to classify the contours as mitochondrion or not.

$$p_{i,j} = \exp\left(\frac{-(c_i - v_{i,j})^2}{2\sigma_i^2}\right)$$

The value $p_{i,j}$ represents a probability that the contour j belongs to a mitochondrion give that it has a particular value $v_{i,j}$ for feature i. The constant c_i represents the ideal value for feature i, given that the contour is part of a mitochondrion. Four features were used: (1) ellipse overlap, (2) perimeter, (3) average gray value of the original image inside the contour, and (4) area of the contour. Ellipse overlap is a measure of how elliptical the contour is. To compute it, an ellipse is fit to the contour using a standard routine in OpenCV and the

1 Available at http://www.itk.org
2 http://openccdb.org
3 http://opencv.willowgarage.com

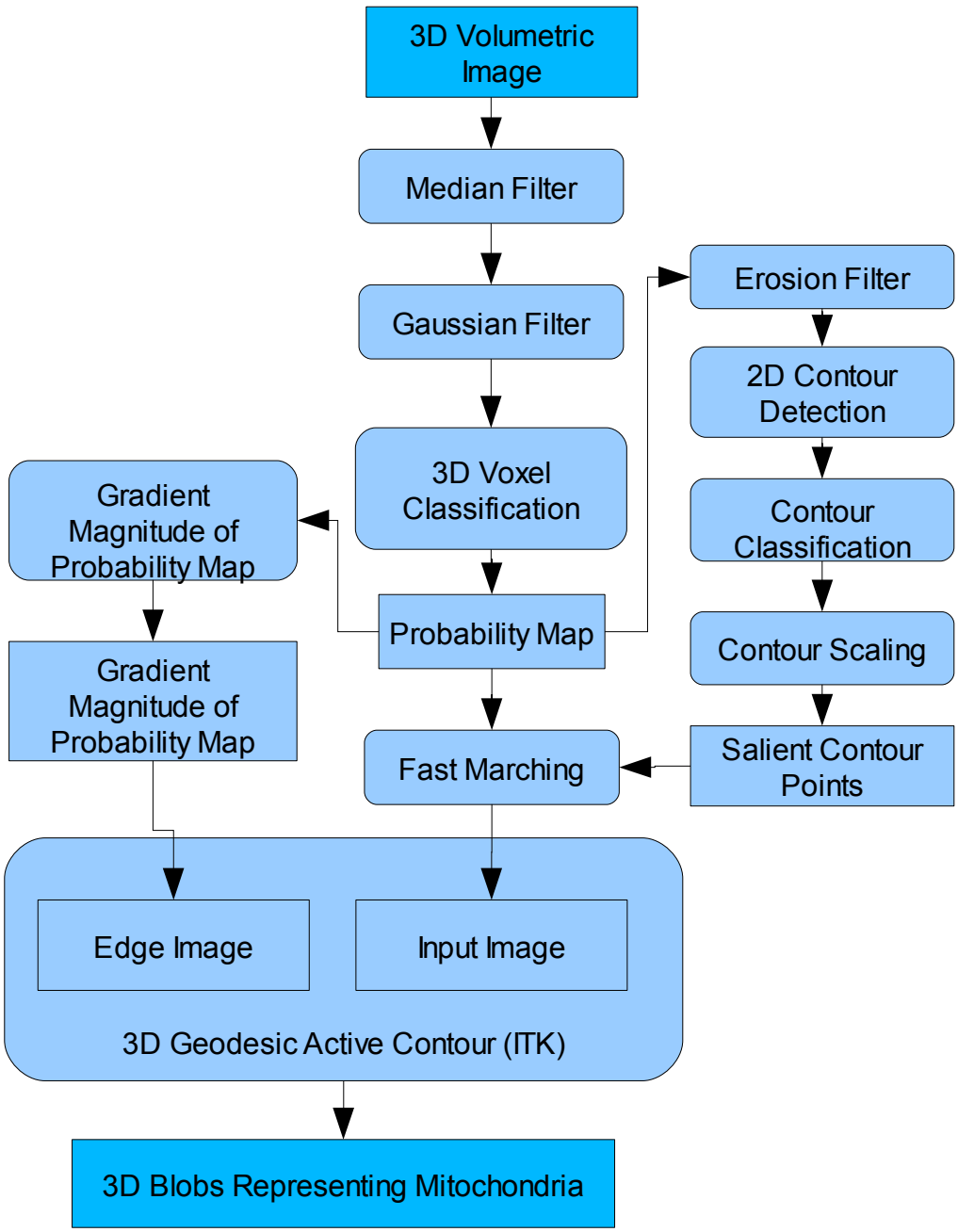

Figure 1: Diagram of the mitochondrion segmentation system. Rounded boxes represent operations, and rectangles indicate data.

fraction that the area inside the ellipse overlaps the actual contour area gives the ellipse overlap value.

P_j represents the probability that contour j belongs to a mitochondrion given all of its features:

$$P_j = p_{1,j}\, p_{2,j}\, p_{3,j}\, p_{4,j}^2$$

Rotation Invariant Features at a Point	Statistics of a 3x3x3 neighborhood
● Gray Value ● Gradient Magnitude ● Sorted eigen values of structure tensor	● Mean ● Standard Deviation ● Third Moment ● Fourth Moment ● Minimum ● 0.25 Quantile ● Median ● 0.75 Quantile ● Maximum

Table 2: Local features used for voxel classification

The term $p_{4,j}$ was squared because experimentally, it was found that emphasizing that feature gives better results. The values for c_i, σ_i, and the threshold for classification were chosen manually with some experimentation. Results are show in Figure 2, views (C) and (D).

2.3 Stage 3: 3D Object Extraction

In stage 3, a "geodesic active contour" level set filter (GeodesicActiveContourLevelSetImageFilter from the ITK library) is used to produces 3D blobs that represent mitochondria. The geodesic active contour filter takes two input images (1) the result of an input fast march filter and (2) the edge potential map. The filter performs a level set operation to generate a final result. (Details regarding the filter are documented in the ITK user's manual.) Proper seeding of the input fast mach is critical to produce an accurate segmentation. The contours detected in stage 2 provide the initial seed points. They are scaled to 1/2 original size so that the march tends to start from inside the original contours rather than at their edges.

The gradient magnitude of the 3D probability map image is used as the edge potential image (see Figure 1). This produces a force so that the geodesic active contour filter operation tends to fill until it reaches the boundary of mitochondria and non-mitochondria voxels. Results are show in Figure 2, views (E) and (F). View (F) shows a slice of raw output. View (E) is the automatic segmentation output with some false positive blobs removed manually.

3 Results

The volume automatically segmented had dimensions 268x325x40. The training set for voxel classification had dimensions 64x62x40 and was a subvolume of the full volume. The required training set is relatively small, so it can be created quickly. Very large training sets, although they may perform better, can be prohibitively time consuming to produce.

The error was calculated as:

$$E = \frac{f_p + f_n}{N},$$

where f_p is the number of false positives, f_n is the number of false negatives, and N is the number of voxels in the volume.

Using all three stages, the final error was 0.049. For comparison, the amount of error was also computed for stage 1 alone. Stage 1 outputs a probability map volume with values ranging continuously between 0 and 1, so output was thresholded. The threshold was chosen so that the same number of true positives were detected as with all 3 stages. The error rate for stage 1 alone was then computed to be 0.155, about 3.2 times as much as the error with all stages, due to false positives. Essentially, the importance of the second two stages is to eliminate false positives that are difficult to classify using local features alone.

4 Integration with Manual Segmentation

Automatic segmentation produces a set of 3D blobs in raster format, which represent mitochondria. These can be converted to Jinx XML using a Jinx import function and automatically labeled as members of the ontological class Mitochondrion. Manual correction of the results can also be performed through Jinx. These routines are available through the Cytoseg[4] tool of Open CCDB[5]. The project is written in Python and uses the pythonxy[6] platform (which includes scipy[7] and ITK image processing tools).

4 http://cytoseg.googlecode.com
5 http://openccdb.org
6 http://www.pythonxy.com
7 http://www.scipy.org

5 Conclusion

A new method for automatic segmentation of mitochondria was presented. This work shows that use of additional stages (contour classification and 3D level sets) improves accuracy of the detection system significantly. The system detects and segments mitochondria in data which is cluttered with other structures and

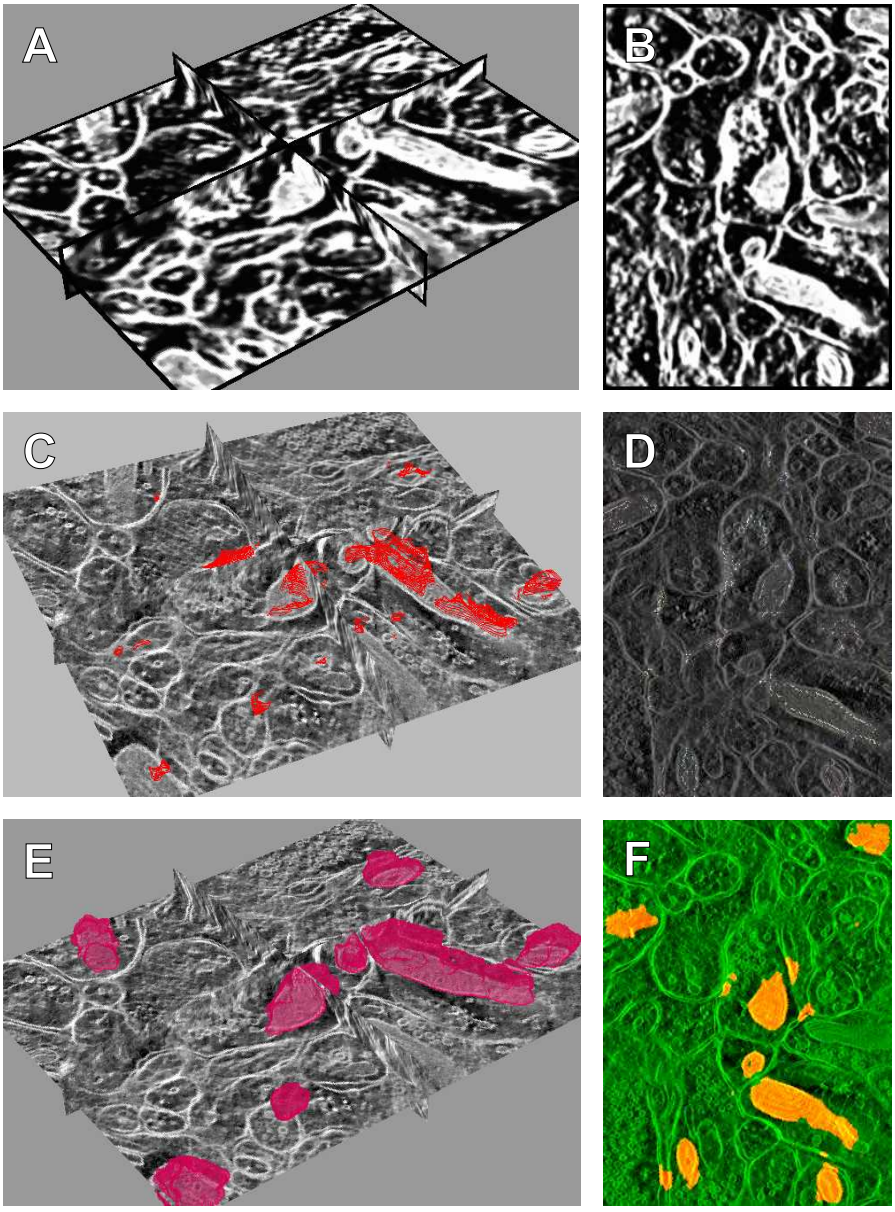

Figure 2: (A) 3D view of voxel classification. (B) 2D view of voxel classification. (C) 3D view of contours detected and classified as salient. (D) 2D view of contours detected and classified as salient. (E) 3D view of automatically segmented blobs representing mitochondria. Some blobs were removed manually. (F) 2D view of automatically segmented blobs representing mitochondria with no manual post processing.

has the missing wedge artifact typical of electron microscopy. To maximize the usefulness to the community, the code is freely available and was developed as part of an open source project, Cytoseg. As future work, we plan to improve the accuracy of automatic detection further and extend the system to detect other subcellular structures.

6 Acknowledgments

We would like to thank James Obayashi for his meticulous manual segmentation work. We would also like to thank Steve Lamont for creating an import function in Jinx compatible with this automatic segmentation system.

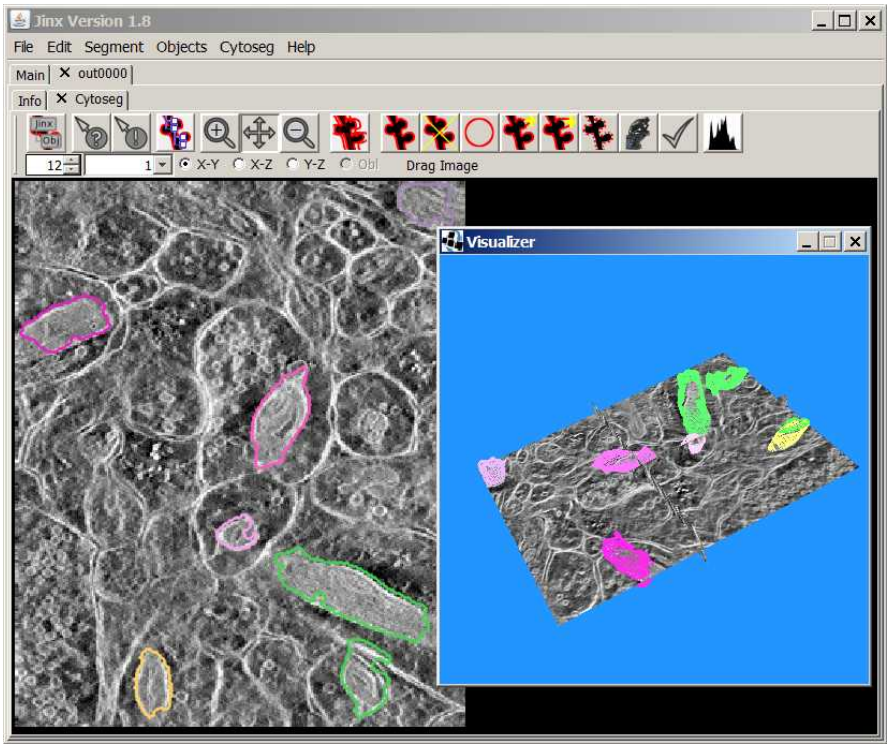

Figure 3: Automated segmentation of mitochondria in a 3D electron tomographic volume of cerebellar neuropil and output of extracted objects as instances of SAO.

7 References

[1] R. Narasimha, H. Ouyang, A. Gray, S.W. McLaughlin, and S. Subramaniam, “Automatic joint classification and segmentation of whole cell

3D images," *Pattern Recognition*, vol. 42, 2009, pp. 1067-1079.

[2] Y. Mishchenko, "Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs," *Journal of Neuroscience Methods*, vol. 176, 2009, pp. 276-289.

[3] B. Andres, U. Köthe, M. Helmstaedter, W. Denk, and F.A. Hamprecht, "Segmentation of SBFSEM volume data of neural tissue by hierarchical classification," *DAGM*, Springer, 2008, pp. 142–152.