Neuron Segmentation and Inner Structure Analysis of 3D Electron Microscopy Data

A thesis submitted
to Kent State University in
partial fulfillment of the requirements
for the degree of Master of Science

by

Xinyu Chang

August 2013
Thesis written by

Xinyu Chang

B.S. Dalian Foreign Languages University, 2009
M.S. Kent State University, 2013

Approved by

Dr. C.C. Lu, Advisor
Javed Khan, Chair, Department of Computer Science
Ray Craig, Dean, College of Arts and Sciences
# TABLE OF CONTENTS

LIST OF FIGURES ........................................................................................................ iv  
LIST OF TABLES ........................................................................................................ vi  
I. Introduction ................................................................................................................ 1  
II. Automatic Seed Point Identification ...................................................................... 3  
III. Axon Segmentation ................................................................................................... 5  
  3.1 Splitting Case ......................................................................................................... 6  
  3.2 Merging Case ......................................................................................................... 7  
  3.3 Self-adaptive threshold .......................................................................................... 7  
IV. Connect isolated region and Edge patching ............................................................ 8  
  4.1 Connect isolated region ......................................................................................... 9  
  4.2 Edge Patching ....................................................................................................... 10  
V. Inner Structure Analysis ............................................................................................ 13  
  5.1 Objective inner structure ...................................................................................... 13  
  5.2 None-objective inner structure ............................................................................. 14  
  5.3 Inner Structure Segmentation ............................................................................. 14  
  5.4 Histogram based separation .................................................................................. 15  
  5.5 Analysis ................................................................................................................ 17  
  5.5 Classification ........................................................................................................ 23  
  5.6 Merging ................................................................................................................ 23  
VI. Result and Discussion .............................................................................................. 26  
  6.1 User Interface ........................................................................................................ 26  
  6.2 Detailed Information ............................................................................................ 26  
  6.3 Colored inner structure ....................................................................................... 27  
  6.4 Axon outline shape and distribution ..................................................................... 28  
  6.5 Scatter Plot Chart .................................................................................................. 29  
  6.6 Axon List Table .................................................................................................... 29  
VI. Conclusion and future work .................................................................................... 30  
References ................................................................................................................... 31
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Fig.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Work flow chart</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Segmentation results of different preprocessing</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Seed point identification result</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Merge case and splitting case across slices</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>Self-adaptive threshold result</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>Boundary Holes</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>Edge Patching.</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Distance Ratio</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>Covered line.</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>More execution result</td>
<td>13</td>
</tr>
<tr>
<td>11</td>
<td>An example of the map data structure</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>Connected structures</td>
<td>16</td>
</tr>
<tr>
<td>13</td>
<td>Projected Histogram</td>
<td>17</td>
</tr>
<tr>
<td>14</td>
<td>The execution results of Histogram based separation</td>
<td>17</td>
</tr>
<tr>
<td>15</td>
<td>Density comparison between mitochondria and cell contour</td>
<td>18</td>
</tr>
<tr>
<td>16</td>
<td>Feature of mitochondria</td>
<td>19</td>
</tr>
<tr>
<td>17</td>
<td>Connected none-objective structues, and their segmentation results</td>
<td>21</td>
</tr>
<tr>
<td>18</td>
<td>Contour movement formula explanation</td>
<td>22</td>
</tr>
<tr>
<td>19</td>
<td>Rule for ER merging</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td>Rule for mitochondria merging</td>
<td>24</td>
</tr>
</tbody>
</table>
Fig. 21 User interface.................................................................................................................. 26

Fig. 22 Detailed information for all the inner structures. .......................................................... 27

Fig. 23 Result image of colored inner structure.......................................................................... 28

Fig. 24 Result of axon shape and distribution............................................................................. 28

Fig. 25 A scatter plot chart of 7 axons ....................................................................................... 29

Fig. 26 Axon list table.................................................................................................................. 30
LIST OF TABLES

Table 1 The execution result of Spotted Degree and Correlation ................................................. 20

Table 2 Execution result of Contour Movement Algorithm .......................................................... 22

Table 3 Inner Structure Classification based on Calculated Metrics ........................................... 23
Abstract- The 3D image generated by Serial block face scanning electron microscopy (SBFSEM) is in very large scale. Its data size could easily exceed 100 GB. The 3D image is constituted by over hundreds of slices in very high resolution. Due to the immenseness of the data scale, any manual operation or user intervention involved to the analysis is incredible time consuming. Current methods need the user to set the ROI or to correct the execution result, which are a huge amount of work. In this paper we introduce a completely automatic work flow to process and analyze the large image data. The whole work flow is implemented as a plugin based on an open-source framework, ImageJ, which is a public domain, Java-based image processing program. The plugin can automatically identify the seed point to start the whole process, segment single axons from the image, analyze their inner structures, obtain the information for each single axon like volume, number of mitochondria and number of smooth endoplasmic reticulum (SER), and finally visualize the analysis result.

I. Introduction

Electron Microscopy is playing its significant role since its inception in biological science field. Till now, to review nanometer scale cellular ultra-structure electron-beam is the only solution. Based on scanning EM instruments, recently developed "serial block face scanning EM" (SBFSEM) systems can generate 3D datasets containing thousands of TEM-like, serial images each day. They work by imaging the sample block face using a scanning electron beam and collecting backscattered electrons, and then an in-chamber diamond knife assembly shaves a thin section from the block face. Automated cycles of imaging and cutting generate image stacks of a hundred gigabytes in 24 hrs, with both large field dimensions (up to 500 um) and nanometer resolution. The EM data, which is in gray-scale TIFF format, different from other works in 3D EM data[5,6], our data are in horizontal direction. Numerous applications of this new technology exist including research, preclinical, and potentially clinical testing. By automating image generation, SBFSEM systems have provide a wealth of new data but also introduce a new
experimental bottleneck at the stage of image processing. A typical dataset acquired overnight takes a trained analyst one to two weeks to manually segment structures for 3D reconstruction and measure features of interest, which include axons and dendrites in the central nervous system. EM images are complex, information-rich grayscale images, in which features of interest are identified by following membrane boundaries and recognizing characteristic textures. Most structures cannot be extracted by simple density-based segmentation. As SBFSEM systems inevitably come into more widespread use, post-hoc image analysis will be a substantial problem.

The greatest bottleneck in analysis of 3D-EM datasets is the image analysis component. We have developed simple but effective tools (using ImageJ) that create novel pipelines (Fig. 1) for analysis. We have created a system to increase throughput by utilizing and developing fully automated approaches. The work flow covered the whole life circle of image processing from preprocessing to visualization of the execution result. We used Anisotropic Diffusion as preprocessing, and based on the preprocessed data a Mean Shift algorithm has been applied to identify the initial seed points automatically. With those initial seed points as starting point, a merging and splitting detectable region growing algorithm will segment each of the axon out through all the slices. With a self-adaptive threshold algorithm the region growing algorithm can obtain good result even some region in the 3D-EM image is darker. Due to the imperfect nature of all the threshold based segmentation, algorithms has been applied to connect those isolated regions and patch the incomplete edge of neuron cell. Based on the optimized segmentation result the inner structure regions has been extracted and analyzed one by one. They have been classified into categories, and according to their assigned category a merging algorithm through all the slices has been applied. So the inner structures belong to a single tissue will be merged
into one. And finally all the analysis result has been visualized into pellucid but informative table and charts, so any analyst can easily make conclusion.

II. Automatic Seed Point Identification

In order to achieve completed automatic function, it is necessary to make the plugin be able to identify the initial seed points automatically. So the whole process in many steps can be finished.
with a single click. To gain seed points for each axon as a starting point for subsequent segmentation, this process can be divided into three steps.

First, a nonlinear diffusion smoothing method is applied [1] as a preprocess to remove image noise and texture while keeping the significant parts of the image content, so the edges will be kept.

![Image of original, Gaussian blur, and anisotropic diffusion results](image)

Fig. 2 Segmentation results of different preprocessing

(a) is the original image; (b) is image after Gaussian Blur; (c) is the result after anisotropic diffusion.

As we can see in Fig. 2 a, the binary images below are threshold segmentation results. The result of directly applying segmentation on original image is quit noisy. Comparing to traditional Gaussian filter (Fig. 2 b), nonlinear diffusion can preserve boundaries and does not cause significant edge blurring (Fig. 2 c). Moreover, since the texture has been removed, the segmentation result won’t be affected by the texture of inner structure. And we can also see the result in Fig. 2 c may have less number of white areas than Fig. 2 b, it is acceptable because some of the none-objective structure may be included into the foreground due to their lower density, the analysis result won’t be affected even those structures are ignored. More about objective and none-objective structures will be discussed in section V.
Second, a Mean Shift segmentation method [2] is performed to divide the whole image into multiple subregions. Mean Shift can divide the image into clusters without specifying neither the number of clusters nor the shape of each cluster. This method can cluster each pixel into one subregion according its gray value and spatial location. Then, small regions are removed, since the objective region usually have relative large area. Finally, we calculate the geometric center for each subregion based on cell’s boundary information.

After that we compute distance between each inner point and boundary, and then choose the point which has maximum distance as the geometric center of the cell. As showed in Fig. 3, these center points can be taken as seed points for region growth segmentation.

![Fig. 3 Seed point identification result](image)

III. Axon Segmentation

To segment each single axon across multiple slices, an 8-neighbor 2D region growing will be started from a seed point, all the pixels that are within the current threshold will be involved into
the flood. And the region growing will be end until no more pixels in the neighborhood of
currently obtained region is within the current threshold. The initial seed points could be multiple
and be obtained from the result automatically generated from the previous section, or manually
chosen by the user. Each tracing task from the initial seed point is independent.

![Diagram showing merging and splitting cases across slices](image)

**Fig. 4 Merge case and splitting case across slices**

the number marked on the figure is the possible region detection sequence.

By avoiding 3D region growing, execution speed is improved significantly. However cases of
merging and splitting happen across the slices (Fig. 4). These kinds of problems have been taken
into consideration. The tracing algorithm is capable of capturing the whole axon even it splits or
merges.

### 3.1 Splitting Case

When a region $A$ has been obtained by 2D region growing from an initial seed point in slice $S_i$,
the region will be projected on next slice $S_{i+1}$. By projecting $A$ to $S_{i+1}$ we get $A'$, a new seed point
that within the threshold will be chosen from $A'$. After a new region growing has been applied
starting from that seed point, we get $B$. The size of the new region $B$ in $S_{i+1}$ will be compared to
$A$, if there is a significant difference between the sizes, a new seed point inside of $A'-(A' \cap B)$
will be selected to start a new region growing. The new region we get each time will be included
into $B$. This process will be repeated until the size difference between two slices becomes insignificant.

3.2 Merging Case

Each time when the whole branch has been captured in $S_{i+1}$ by applying the algorithm in section 3.1, $S_i$ will be scanned for a merging case. By projecting $B$ to $S_i$ we get $B'$, a region Merging cases will be detected when $B' - (A \cap B')$ is significant, a new seed point will be set on $S_i$. If a new region is obtained, in the same way the upper slice $S_{i+1}$ will be scanned, and this process will be repeated until no region from upper slice could be obtained. Every time a new region is obtained only its previous slice is checked, which means that the new region’s splitting case in next slice will be ignored. However, to process this type of EM data, this algorithm is sufficient.

3.3 Self-adaptive threshold

If the threshold is fixed, some of the neuron cell is relatively darker (Fig. 5 a), and a fixed threshold will result in bad segment result (Fig. 5 b). To solve this problem a self-adaptive threshold algorithm has been applied. When the mean intensity of pixels collected by the flood continually stays close to the threshold, the threshold will be changed toward the background. When it is the other case that the mean density is far from the threshold, the threshold will be changed toward to original threshold. Both upper bound and lower bound has been set to prevent the threshold goes too distinct from the foreground. The threshold changing value $\Delta\text{Thre}$ is computed as:

$$\Delta\text{Thre} = \begin{cases} \log(P \cdot (\text{Mean}(S) - \text{Thre})) \cdot \lambda), & \text{Mean}(S) - \text{Thre} < P \\ \log(\text{Mean}(S) - \text{Thre}) \cdot \lambda), & \text{Mean}(S) - \text{Thre} > P \end{cases}$$

$\text{Thre}$ is current threshold, $S$ is the pixel set recently collected, $P$ is the scale that will trigger a threshold change, and it has been set to 5 in our case. If the initial threshold is 145, when
the mean density is between 145 and 150, the threshold change will be triggered to decrease the threshold. If the current threshold is 130, when the mean density is more than 135, the threshold change will be triggered to increase the threshold. $\lambda$ is a parameter to limit the changing degree.

![Fig. 5 Self-adaptive threshold result](image)

(a) (b) (c)

(d) (e) (f)

**Fig. 5 Self-adaptive threshold result** (a) is the darker area in axon cell. (b) is the segmentation result of (a) without adaptive threshold. (c) is the segmentation result of (a) with adaptive threshold. (d) is a none-darker area in axon cell. (e) is segmentation result of (d). (f) is the segmentation result of (d) with adaptive threshold.

The segmentation result with adaptive threshold has been shown in **Fig. 5 c**, we can see that the whole cell body has been captured even the average density there is quite different from normal cells. And the adaptive threshold doesn’t affect the segmentation result in none-darker areas (**Fig. 5 b**), because the segmentation results with this method (**Fig. 5 f**) and without (**Fig. 5 e**) are almost the same.

**IV. Connect isolated region and Edge patching**

After the segmentation in section III, due to some of the inner structures are beside the contour (**Fig. 7 a**), holes may be visible inside the boundary of the main axon. Many of the holes are too close to the contour of the cell and appear to be connected to the background (**Fig. 7 b**). To
tackle this problem we developed an algorithm to patch the incomplete boundary of the axon, as a result the inner structure of the axon within the segmented boundary can be analyzed.

4.1 Connect isolated region

Some of the foreground parts are isolated small region apart from the main region of neuron cell, (Fig. 6a), in this case with a chain code boundary detection algorithm, we get information of the edge from every individual segmented region when its perimeter is below a certain value, which indicates that it is a small region, after checking its neighbor area a line will be draw between two points that has the minimum distance, one point is from the edge of the small region, the other is from its neighboring region. As a result, the small region could be considered as a part of the main cell (Fig. 6b).

![Fig. 6 Boundary Holes](image)

(a) red area shows isolated small region. (b) the orange lines is the execution result of connecting small regions, the green line is the execution result of Edge Patch, which is discussed in later of this section.

However the formula of getting the distance is weighted, for in these neuron EM images it is always preferable to draw a line to connecting the regions in a horizontal direction, so the distance in the horizontal direction has more weight in this case. The weighted distance $\text{dist}$ is obtained from the formula below.
\[ \text{dist} = \begin{cases} 
    d + \frac{d}{1 + |\Delta x/\Delta y|} & (\Delta y \neq 0) 
    
    d & (\Delta y = 0)
\end{cases} \]

Where \( d \) is the distance between two points. \( \Delta x \) is the difference in horizontal direction and \( \Delta y \) is difference in vertical direction.

### 4.2 Edge Patching

**Fig. 7 Edge Patching.** (a) shows the inner structures being adjacent to the contour on the preprocessed image. (b) is the segmented result of (a). (c) is the execution result of Edge Patching algorithm. (d) is a normal curve on the edge.

Based on result of connecting isolated small region, better Edge Patching result could be obtained (**Fig. 6 b**). To draw lines to patch the incomplete edge and enclose an inner structure, the first thing to do is to tell the difference between a normal curve (**Fig. 7 d**) on the edge and an inner structure stick to the contour, one of the most significant difference is the ratio of direct distance between two points divided by their distance along the edge. In **Fig. 8** we can see the obvious ratio difference.
**Fig. 8 Distance Ratio** the green line shows the distance along the edge and the red line shows the direct distance. We can see that the inner structure beside the edge has significant larger ratio.

To locate the two points that draw the line there are 6 steps;

1) Pick a range that goes along the edge of the cell until it reaches the end.

2) When the ratio of the start point and end point of the range reaches the threshold, keep going, recording the position that has the minimum ratio until the ratio goes beyond the threshold again.

3) Get the two points in the range that have the minimum ratio and fulfill a sequence of rules in the position recorded in last step. The rules are:
(i) No black pixels between the elected two points.

(ii) The distance along the edge should larger than half of current range.

(iii) None of them should be on other line connected.

4) Draw line between the two points obtained from step 3).

5) If the two points covered another pairs of elected points, erase the line drawn between the points which have been covered (Fig. 9).

![Fig. 9 Covered line.](image)

6) Update the boundary information with the new line connected.

For the size of inner structures varies significantly, we need to pick different length q between a certain range of to patch different size of gaps. Typically, with larger q, a larger inner structure can be patched, with a lower q, a smaller inner structure can be patched. Even sometimes a big hole may have very big length along the edge (even larger than the maximum q), by apply this algorithm multiple times, the line drawn previously will be covered by later ones, and in this way the large hole will be patched. Since the algorithm is very efficient, we pick q between range from 600 pixels to 30 pixels, and the algorithm will execute multiple times with different q so inner structures in every sizes are covered. We can see more execution result in Fig. 10
V. Inner Structure Analysis

5.1 Objective inner structure

1) Mitochondria

2) ER
5.2 None-objective inner structure

5.3 Inner Structure Segmentation

With the method in section IV, now we have the inner structures enwrapped inside the axon cell boundary. Which means it is very easy to tell a white area is inner structure or just the background in this binary image. To segment mitochondria and ER from the extracted axonal elements in the processed binary image, a data structure has been constructed. It is a 2D matrix list, and the size of the list is the slice number that the axon crossed. For the matrix in each slice, they have their own width $w_i$ and height $h_i$, $w_i = \text{MAX}(x) - \text{MIN}(x)$ and $h_i = \text{MAX}(y) - \text{MIN}(y)$, which means it is the rectangle area an axon covers. In the matrix, the region information is stored in five values.

1. Unchecked, all the matrix will be initialized in this value.

2. Undecided, a region that has been covered by flood but is not certain whether it is background or inner structure for they are both in color white.

3. Foreground, the axon body with black color will be set to this value.

4. Background, region around the body.

5. Structure ID, once an inner structure is detected in the matrix the region will be assigned to its corresponding ID, the IDs are a sequence of ordered number.
In the beginning all the entropies are assigned Unchecked value. A flood is started from a pixel \(i\) with Unchecked value in the matrix, when the color of the pixel is black, its matrix value will be directly assigned to Foreground. If that pixel is white in color, it will involve all its white neighbors into the flood and assign their value in the matrix to Undecided. When the flood has gone outside the boundary, all the pixels that have been recruited will be given the Background value in the matrix. Otherwise, if the flood stopped without going outside the boundary, the matrix values will be assigned to a new Structure ID. In Fig. 11, the green area shows the covered region by current flood, since the flood is neither finished nor has reached the boundary of the matrix the value on the map remained Undecided. An inner structure list has also been built to store relevant information like size, pixels, type, Structure Id, Merged ID for later analysis.

5.4 Histogram based separation

Some line shaped structure and Mitochondria also connected together, in this case some special segmentation result (Fig. 12) may be obtained.
In order to analyze them separately a histogram based algorithm has been applied. This algorithm is based on an assumption that ER structures extend in the horizontal direction. The histogram is obtained by projecting the number of pixels in vertical direction (Fig. 13 upper).

When a significant change is detected, the value before and after the change remains stable and one side of the histogram is within the ER structure criteria, a histogram pattern like this will be detected (Fig. 13 lower).
Fig. 13 Projected Histogram. The upper picture is an ER connected to the contour of the cell, the lower picture is the corresponding histogram.

The structure is next separated in the position where the significant change occurs. Some examples has been shown in Fig. 14

Fig. 14 The execution results of Histogram based separation

5.5 Analysis

After having obtained all the inner structure areas, in this section 5 categories of information will be extracted from those areas.

1) Circularity. The circularity $c$ can be achieved from the following formula;
\[ c = \frac{4\pi S}{l^2} \]

S is the area of the inner structure and \( l \) is its perimeter. \( c \) is ranged from 0 to 1. SERs are ribbon-shaped, so they have much lower circularity than mitochondria, which is in circular shape.

2) **Mean**, the mean value of density. Mean is an important standard to classify mitochondria and other circular inner structure. For the cell contour that stretched into the cell body always have higher density, and mitochondria always have lower density (Fig. 15).

![Fig. 15 Density comparison between mitochondria and cell contour](image)

3) **Area**, the number of pixels. SERs are inclined to be appearing in small size, and mitochondria occupies larger region. Therefore, when an inner structure have very small area will not be considered as mitochondria, on the other hand, if an inner structure has very huge size it won’t be considered as SER.

4) **Spotted Degree**, a texture descriptor. With the criteria set on density mean, most of the none-objective structure with large size can be separated from mitochondria. However, to separate rest of them the texture information should be analyzed. Due to the noisy nature of EM data (Fig. 16 a), General texture analysis method like Second-Order Statistical Texture Analysis [4], which generates grey-tone spatial dependence matrices (GTSDM) in four directions, is not sufficient to tell the distinction between structures. By applying
Gaussian Blur the noise can be removed to some degree (Fig. 16 b). However the tiny structure size and irregular pattern of texture effected on the analysis result.

![Fig. 16 Feature of mitochondria](image)

(a) are original images in 1200% scaling degree, (b) are images after gaussian blur in 800% scaling degree. (c) are binary images has been set a threshold according to their means, Upper image of (c) is mitochondria and the lower image is a none-objective structure.

To tackle this problem. There is a essential deffernece between mitochondria and other structures. That is the distribution of pixels with lower density (Fig. 16 c). We developed simple but effective way to describe this distinction. There are 4 steps.

1. Obtain the density mean of the target structure.

2. Set the threshold to mean+25, remove all the pixels on the periphery that is beyond the threshold.

3. Scan each line, Count the number of ranges that have more than 5 and less than 20 consecutive pixels beyond the threshold.

4. Sum the number counted in each line and divide it by structure size.
Finally we set the threshold for Spotted Degree to 8. The execution result has been shown in Table 1.

<table>
<thead>
<tr>
<th>Inner Structure</th>
<th>Type</th>
<th>Spotted Degree</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>None-Objective Structure</td>
<td></td>
<td>6.266</td>
<td>0.843</td>
</tr>
<tr>
<td>None-Objective Structure</td>
<td></td>
<td>4.412</td>
<td>0.785</td>
</tr>
<tr>
<td>None-Objective Structure</td>
<td></td>
<td>0.000</td>
<td>0.801</td>
</tr>
<tr>
<td>Mitochondria</td>
<td></td>
<td>10.884</td>
<td>0.806</td>
</tr>
<tr>
<td>Mitochondria</td>
<td></td>
<td>17.061</td>
<td>0.812</td>
</tr>
</tbody>
</table>

Table 1 The execution result of Spotted Degree and Correlation.

5) Contour Movement. Some of the none-target inner structures, due to their low intensity some part of them have been segmented into the foreground (Fig. 17a), and some of the cell contour adjacent to those structures (Fig. 17b). Both of them resulted in irregular shapes. For sometimes their circularity and mean density are close to mitochondria. And because of their uneven texture, texture analysis also hard to tell their difference. To separate them from mitochondria a contour movement algorithm has been applied.
The equation of calculate contour movement $C$ is like:

$$C = \left( \sum_{i=0}^{p} \frac{\text{abs} (\Delta d_i) / d_i}{p} \right) \left( \sum_{i=0}^{p} \frac{\text{abs} (\Delta d_i - \Delta d_{i+1})}{p} \right) \times 100$$

In this approach, the center point and size of each object are not unified. Instead of that, as it is shown in Fig. 18 the difference rate $\Delta d_i$ between distances from center point to the contour pixels $d_i$ and $d_{i+1}$ has been compared. The comparison is between two pixels in on the contour in a certain interval $L$. $L$ is number of continual pixels on the contour covers a fixed percentage of $p$. $p$ is the number of pixels on the contour. And in our case the percentage has been set to 2%. The left part of the equation is the changing moving rate along the edge, and the right part of the equation is actually the contour smoothness.
All the inner structures that have been recognized as Mitochondria will go through the process described in this section, type of inner structure with $C > 1.68$ will be recognized as Mitochondria candidates.

The execution result is like Table 2, we can see that regular shape and irregular shape can result in significant difference in Contour Movement Value. By designate a fixed $L$, the small contour movement for big area will become less significant, only the rough outline will affect its contour movement value.

<table>
<thead>
<tr>
<th>Inner Structure</th>
<th>Contour Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Image" /></td>
<td>0.07</td>
</tr>
<tr>
<td><img src="image2" alt="Image" /></td>
<td>0.043</td>
</tr>
</tbody>
</table>
5.5 Classification

Using the data obtained from the texture analysis algorithm, we can divide the inner structures into 4 classification categories (Table 3).

<table>
<thead>
<tr>
<th>Type</th>
<th>Circularity</th>
<th>Mean</th>
<th>Area</th>
<th>Spotted Degree</th>
<th>Contour Movement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncertain</td>
<td>&gt;0.451</td>
<td>-</td>
<td>&lt;500</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>&lt;0.451</td>
<td>&gt;120</td>
<td>&lt;2000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ER</td>
<td>&lt;0.451</td>
<td>&lt;120</td>
<td>&lt;2000</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mitochondria</td>
<td>&gt;0.451</td>
<td>-</td>
<td>&gt;500</td>
<td>-</td>
<td>&gt;0.168</td>
</tr>
<tr>
<td>Candidate</td>
<td>&gt;0.451</td>
<td>-</td>
<td>&gt;500</td>
<td>&lt;8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>&gt;0.451</td>
<td>&lt;100</td>
<td>&gt;500</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Mitochondria</td>
<td>&gt;0.451</td>
<td>&gt;100</td>
<td>&gt;500</td>
<td>&gt;8</td>
<td>&lt;0.168</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>&gt;100</td>
<td>&gt;2000</td>
<td>&gt;8</td>
<td>&lt;0.168</td>
</tr>
</tbody>
</table>

Table 2 Execution result of Contour Movement Algorithm

Table 3 Inner Structure Classification based on Calculated Metrics
5.6 Merging

In this section, according to the information we collected in previous step, the inner structures that are actually belonging to a single tissue will be merged into one. Structures across slices which are classified as ER or Mitochondria with high certainty are designated as the starting point of merging. Since ER structures may shift significantly between slices a wider area is searched to locate inner structures that are part of a single ER. Both ER Structure and ER candidates are merged (Fig. 19).

![Rule for ER merging](image1)

Fig. 19 Rule for ER merging.

For Mitochondria, a smaller area than itself in its neighbor slice is searched, since it is relatively stable across slices. Uncertain mitochondrial candidates and Mitochondria will be merged (Fig. 20).

![Rule for mitochondria merging](image2)

Fig. 20 Rule for mitochondria merging
Merging across slices will be started from those structures Ni having more certainty as a type of ER or Mitochondria and are designated as the starting point of merging. The matrix list obtained from the previous steps (Fig. 11) will be used. By applying the algorithm, inner structures that are connected by high certainty type will be assigned to the same Merged ID. For the ER, which structure shifts a lot between slices, a wider area in both upper slice and lower slice are searched to locate the inner structures that are actually a single ER. Uncertain Structure, ER Structure and ER candidates will be merged. And for mitochondria a smaller area than itself in its neighbor slice will be searched, for it is relatively stable across the slices. Uncertain Structure, Mitochondria Candidate and Mitochondria will be merged.

The algorithm is outlined below:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IS: INNER STRUCTURE</td>
<td></td>
</tr>
<tr>
<td>FOR ALL IS.type = Mitochondria or IS.type = ER</td>
<td></td>
</tr>
<tr>
<td>N = IS;</td>
<td></td>
</tr>
<tr>
<td>IF (N. Merged_ID = NONE)</td>
<td>Assign N a new Merged_ID;</td>
</tr>
<tr>
<td>FOR ALL IS in upper and lower slice</td>
<td></td>
</tr>
<tr>
<td>X = IS;</td>
<td></td>
</tr>
<tr>
<td>IF (X adjacent to N AND (N. Type == Mitochondria AND (X.type = Mitochondria OR Mitochondria Candidate OR Uncerntain)) ) OR (N. Type == ER AND (X.type = ER OR Uncerntain)) OR (N. Type == Mitochondria Candidate) AND the spatial inner structure N belongs to has more than 2 Mitochondria) THEN</td>
<td></td>
</tr>
<tr>
<td>IF (X. Merged_ID = NONE) THEN</td>
<td></td>
</tr>
<tr>
<td>X. Merged_ID = N. Merged_ID;</td>
<td></td>
</tr>
<tr>
<td>ELSE</td>
<td></td>
</tr>
<tr>
<td>FOR ALL IS.Merged_ID = X. Merged_ID</td>
<td></td>
</tr>
<tr>
<td>IS.Merged_ID = N. Merged_ID;</td>
<td></td>
</tr>
</tbody>
</table>
VI. Result and Discussion

6.1 User Interface

In this section the output of the whole process and its visualization will be discussed. The user interface, which is implemented based on the ImageJ framework, is an instance of GenericDialog class (Fig. 21). The button above named Anisotropic Diffusion 2D can execute the preprocessing step for the user. The Initial Density textbox is for the initial threshold for the segmentation method in section II. And the five textboxes under it is the thresholds for inner structure classification. The Show detailed info checkbox is to turn on or off some detailed information result (Fig. 12) for inner structure analysis. It is also allowed to set the calibration for each of the pixel in order to calculate the volume of a neuron cell. And for the last textbox, which is the Original Image Path, is for use to specify the path of original images. Because for most of time the user may import a preprocessed sequence of image and start the whole process.

6.2 Detailed Information

In Fig. 22 shows the detailed information of all the inner structures, the red text above each inner structure is a unique identifier referenced to the result table of computed data. The number before # is Axon Id, number after # is inner structure Id, and the letter after structure is structure type
(U: uncertain M: Mitochondria MC : Mitochondria Candidate ER: Endoplasmic Reticulum), and the number at the end is the Id of merged tissue it belongs to. By move to upper slice and lower slice in order to checking their Id of merged tissue the user can see their merging and splitting status conveniently. For example we can see that the inner structure with Id 1#153ER27 and 1#152ER27, both of them has been assigned same tissue Id, which means that they are actually belonging to same ER.

Fig. 22 Detailed information for all the inner structures.

6.3 Colored inner structure

The result image in Fig. 23 mitochondria has been colored in red, and ER has been colored in green. By doing this, the segment and analysis result can be examined intuitively. We can see that the connected ER and mitochondria are separated by Histogram Based Separation, and on
the left side of the cell some none-objective inner structure has been excluded from the colored result just like we expected. And since the preprocessing part of Anisotropic Diffusion is our bottle neck of efficiency, this step can be implemented on GPU in the future.

![Fig. 23 Result image of colored inner structure](image)

6.4 Axon outline shape and distribution

The result in Fig. 24 shows the outline of axons and their distribution in a single slice. By dragging the scroll bar we can see their movement across the slices. Each of the axon has been assigned a unique color, so a very clear view can be obtained by observing this result image.

![Fig. 24 Result of axon shape and distribution](image)
6.5 Scatter Plot Chart

In Fig. 25 a scatter plot chart shows the distribution of axons according to their number of SER and number of mitochondria. The size of the circle represents the volume of the axon. The number marked on each circle is axon id, the same id showed in Fig. 22. And the color of each circle is the same with the axons in Fig. 24, so they can be easily identified. From the scatter plot chart, we can conclude that the number of mitochondria, number of ER and the volume of axon are correlated.

Fig. 25 A scatter plot chart of 7 axons

6.6 Axon List Table

In Fig. 26 a list of Axons has been shown in a result table. They can be easily located by their axon Id. The result table allows the user check specific output data.
VI. Conclusion and future work

With our segment and analysis methodologies we are able to extract axons from the massive 3D electron microscopy images, and identify those objective inner structures of mitochondria and SER. Finally output the visualized segmentation and analysis results. Our most significant advantage is that the whole process doesn’t involve any user intervention, and most of the algorithms applied are lightweight, the execution result can be obtained efficiently even the data set is in large scale. Our test data is sized as 3978 x 3054 x 20. Test machine CPU: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz; Memory: 8.00 GB; Win 7 64-bit operating system. It took 383630ms (6.39 min) to process all the 22 initial points. For the processing for each single axon is highly independent, we can include parallel computing in our future work. In a SIMD environment, all the tasks could be executed at the same time. As a result the efficiency would improve significantly. Also, it would be better if the segmented axon with inner structure could be displayed in 3D, with the contour assigned opacity the user could clearly see the distribution of inner structures.
References


