SEMI-AUTOMATIC SEGMENTATION OF MITOCHONDRIA ON
TRANSMISSION ELECTRON MICROSCOPY IMAGES USING LIVE-WIRE
AND SURFACE DRAGGING METHODS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATICS

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

MAHDIEH FARZIN ASANJAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

MEDICAL INFORMATICS

September 2017

SEMI-AUTOMATIC MITOCHONDRIA SEGMENTATION ON ELECTRON MICROSCOPY IMAGES

Submitted by **MAHDIEH FARZIN ASANJAN** in partial fulfillment of the requirements for the degree of **Master of Science in Medical informatics, Middle East Technical University** by,

Prof. Dr. Deniz Zeyrek Bozşahin
Director, **Graduate School of Informatics**

_____

Asst. Prof. Dr. Yeşim Aydın Son
Head of Department, **Health informatics**

_____

Prof. Dr. Ünal Erkan Mumcuoğlu
Supervisor, **Health Informatics**

_____

 Asst. Prof. Dr. Reza Zare Hassanpour
Co-Supervisor, **Department of Computer Engineering, Çankaya University**

_____

Examining Committee Members:

Asst. Prof. Dr. Aybar Can Acar
Health Informatics, METU

_____

Prof. Dr. Ünal Erkan Mumcuoğlu
Health Informatics, METU

_____

Asst. Prof. Dr. Reza Zare Hassanpour
Computer Engineering, Çankaya University

_____

Assoc. Prof. Dr. Hadi Hakan MARAŞ
Computer Engineering, Çankaya University

_____

Asst. Prof. Dr. Nurcan Tunçbağ
Health Informatics, METU

_____

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this wok.

Name, Last name: MAHDIEH FARZIN ASANJAN

Signature: _____

ABSTRACT

SEMI-AUTOMATIC SEGMENTATION OF MITOCHONDRIA ON TRANSMISSION ELECTRON MICROSCOPY IMAGES USING LIVE-WIRE AND SURFACE DRAGGING METHODS

Farzin Asanjan, Mahdieh

M.Sc., Department of Medical Informatics

Supervisor: Prof. Dr. Ünal Erkan Mumcuoğlu

Co-Supervisor: Asst. Prof. Dr. Reza Zare Hassanpour

September 2017, 66 pages

Mitochondria are cytoplasmic organelles of the cell, which have significant role in the variety of cellular metabolic functions. Mitochondria act as the power plants of the cell, and are surrounded by two membranes. Significant morphological alterations are often due to changes in mitochondrial functions. A powerful technique in order to study the three-dimensional (3D) structure of mitochondria and its alterations in disease states is Electron microscope tomography. Due to the presence of various subcellular structures and imaging artifacts, the detection of mitochondria in electron microscopy images is a challenging problem. Another challenge is that each image typically contains more than one mitochondrion. Hand segmentation of mitochondria is tedious and time consuming and also special knowledge about the mitochondria is needed. Fully automatic segmentation methods lead to over segmentation and mitochondria are not segmented properly. Therefore, semi-automatic segmentation methods with minimum manual effort are required to edit the results of fully automatic segmentation methods. In this thesis two editing tools were implemented by applying spline surface dragging and interactive live-wire segmentation tools. These editing tools were applied separately on the results of fully automatic segmentation. 3D extension of these tools were also studied and tested. Dice coefficients of 2D and 3D for surface dragging using splines were 0.93 and 0.92. This metric for 2D and 3D for live-wire method were 0.94 and 0.91 respectively. The root mean square symmetric surface distance values of 2D and 3D for surface dragging was measured as 0.69, 0.93. The same metrics for live-wire tool were 0.60 and 2.11.

Keywords: Medical Image Segmentation, Semi-automatic methods, Transmission Electron Microscopy, Surface Dragging using Splines, Live-Wire

ÖZ

ELEKTRON MİKROSKOP TOMOGRAFISI GÖRÜNTÜLERINDE MITOKONDRILERIN YÜZEY SÜRÜKLEME VE CANLI-TEL YÖNTEMLERI KULLANARAK YARI-OTOMATİK BÖLÜTLENMESİ

Farzin Asanjan,Mahdieh

Yüksek Lisans, Sağlık Bilişimi Bölümü

Tez danışmanı: Prof. Dr. Ünal Erkan Mumcuoğlu

Yardımcı Tez danışmanı: Yrd. Doç. Dr. Reza Zare Hassanpour

Eylul 2017, 66 sayfa

Mitokondriler, hücresel metabolik işlevlerin çeşitliliğinde önemli rol oynayan hücrenin stoplazmik organelleridir. Mitokondri, hücrenin santralleri gibi davranır ve iki zar ile çevrilidir. Mitokondrinin fonksiyonlardaki değişiklikler genellikle önemli şekilsel değişikliklerle birlikte görülür. Elektron mikroskop tomografisi, mitokondrinin üç boyutlu (3B) yapısını ve hastalık durumlarındaki değişiklikleri incelemek için güçlü bir tekniktir. Elektron mikroskop görüntülerinde mitokondrinin saptanması,hücre içi yapılarının ve olguların görüntülenmesinin çeşitliliğinden dolayı zorlayıcıdır. Bir diğer zorluk, her görüntünün tipik olarak birden fazla mitokondri içermesidir. Mitokondrinin elle bölütlenmesi yorucudur, zaman alır ve mitokondriye ilişkin özel bilgiye ihtiyaç vardır. Tam otomatik yöntemler aşırı segmentasyona yol açar ve mitokondriler doğru bir şekilde bölünmez. Bu nedenle, tam otomatik bölümleme yöntemlerinin sonuçlarını düzenlemek için en az kullanıcı etkileşimiyle yarı otomatik bölümleme yöntemleri gereklidir. Bu tez, iki yüzey düzenleme aracı, spline yüzey sürükleme ve etkileşimli canli tel araçları uygulanarak oluşturulmuştur. Bu düzenleme araçları ayrı-ayrı tam otomatik segmentasyon yöntemin sonuçlarına uygulanmıştır. Bu araçların 3B uzantısı da incelenmiş ve test edilmiştir. 2B ve 3B Dice katsayısı yüzey surukleme için 0.93 ve 0.92 olarak ve canli tel yöntemi için 0.94 ve 0.91 olarak ölçüldü. 2B ve 3B ortalama yüzey mesafe değerleri yüzey surukleme yöntemi için 0.69, 0.93 ve canlı-tel yöntemi için sırasıyla 0.60 ve 2.11 dır.

 Anahtar Sözcükler: Tıbbi Görüntü Segmentasyonu, Yarı-Otomatik Yöntemler, İletimli Elektron Mikroskobu, Yüzey Sürükleme, Canlı-Tel

To My Family

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABELS

LIST OF FIGURES

LIST OF ABBREVIATIONS

EMT   Electron Microscope Tomography

TEM   Transmission Electron Microscope

SEM   Scanning Electron Microscopy

RLF   Random-like Features

ssTEM   serial section Transmission Electron Microscopy

CARTA   Clustering Aided Rapid Training Agent

CCDB   Cell Centered Database

NCMIR   National Center for Microscopy and Image Research

RMSSSD   Root-Mean-Square Symmetric Surface Distance

EM   Electron Microscopy

SBFSEM   Serial Block Face Scanning Electron Microscopy

RLF   Random Like Elements

ROI   Region Of Interest

VOI   Volume Of Interest

VNC   Ventral Nerve Cord

VTK   Visualization Toolkit

OMM   Outer Mitochondrial Membrane

CHAPTER 1

INTRODUCTION

In this study it was aimed to develop and implement some interactive segmentation methods in order to correct automatic segmentation results of mitochondrial regions of electron tomography images. The introduction section contains notable morphological properties of mitochondria and highlights imaging techniques which are used to visualize subcellular structures in detail. In the subsequent sections, the motivating factors, aims and scope of this study are described. A comprehensive review of the studies published in the literature is also presented.

1.1. Mitochondria

The term 'mitochondrion' is derived from a Greek word 'mitos' which means 'thread' and 'chondrion' which means 'granule' (1). Mitochondrion (plural mitochondria) is well-defined cytoplasmic organelle of the cell which takes part in a variety of cellular metabolic functions (2). The required energy for survival of the cell is provided by mitochondria. Since all the necessary biological energy of the cell are supplied by mitochondria, and this energy is obtained by oxidizing the substrates of the Krebs cycle, these organelles are important. Most of the energy of the cell in the form of adenosine triphosphate (ATP) is generated by these organelles and it is used a source of chemical energy (2). The mitochondria mentioned to as the 'power houses' of the cell. Almost all of the eukaryotic cells have mitochondria, however they are lost in the later stages of development of cell like in the red blood cells or in elements of phloem sieve tube. In other words, the mitochondria number per cell varies completely; for instance, in humans, there is not any mitochondria in erythrocytes (red blood cells) , whereas there may be hundreds or even thousands of mitochondria in liver cells and muscle cells (3). In diameter the

1

mitochondrion ranges from 0.5 to 1.0 micrometer. Mitochondrial disorder and cardiac dysfunction can effect human health and they also have significant role in the aging process.
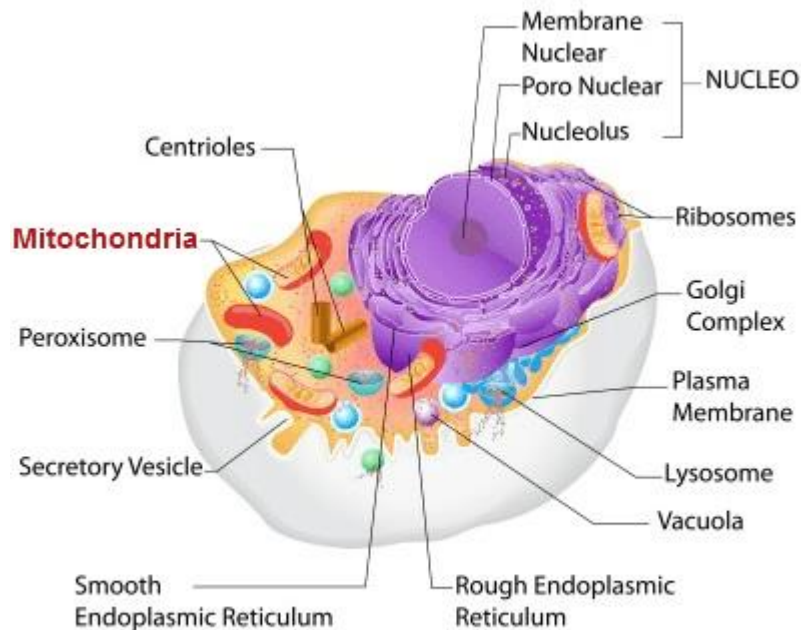
Figure 1: The generalized structure of a cell (2)
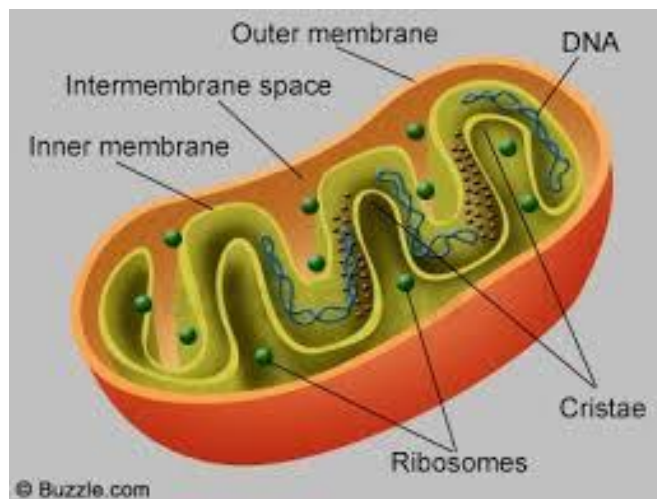
Figure 2: Mitochondrion structural features (2)

Electron microscope tomography techniques can be used to visualize the mitochondria in order to study the relation between function and physical structure of it.

In figure (1), the generalized structure of a cell is shown and structural features of mitochondria are shown in figure (2).

1.2.Imaging Techniques

Tomography is a method which is used to reconstruct the interior of an object from its projections. (See figure (3)). In literature, "tomography" means the visualizing the slices (3).

For any technique that makes use of the transmission electron microscope in order to collect projections of an object the term electron tomography is used. That is titles in various directions and uses these projections to reinstate the object in its totality. Three-dimensional reconstructions are provided by Electron microscope tomography with new insight into their structure and function(6). There are several electron microscopy techniques in order to obtain EMT images. Two common methods which are used to obtain electron microscopy are: Transmission electron microscopy (TEM) and Scanning Electron Microscopy (SEM). In the following subsections these techniques are described briefly.
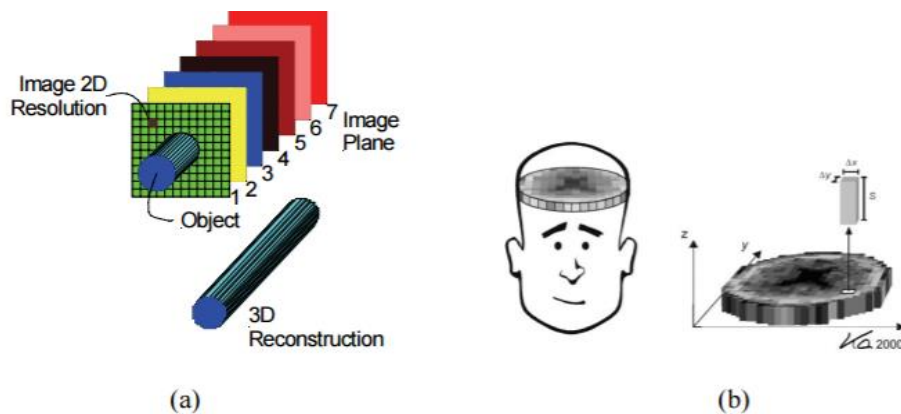
Figure 3: Basic principle of tomography (3)

1.2.1. Transmission Electron Microscopy (TEM)

The transmission electron microscope is a form of electron microscope using an electron gun to produce a voltage electron to illuminate the specimen and creates an image (7). Interacting the electrons which transmitted through the specimen, the image can be formed an image device, such as a fluorescent screen is used in order to magnify and focus on it. Since the electrons have small wavelength, TEMs provide images at a significantly higher resolution than light microscopes. At littler amplifications TEM images differentiate relies on upon the ingestion of electrons in the material, contingent upon the

thickness and synthesis of the material. At higher amplifications complex wave associations offset the force of the picture, contingent upon the master examination of observed images.

1.2.2. Scanning Electron Microscope

A scanning electron microscope is a type of electron microscope which uses a focused beam of electrons to produce images of an object by scanning it (8). The electrons connect with the atoms in the object, and create different signs that contain data about the specimen's surface geology and structure. Using a general raster scan pattern, the electron beam is scanned, and the detected signal is combined with the beam's position to produce an image. The resolution better than 1 nanometer can be achieved by SEM. Specimens can be seen in conditions such as high vacuum, low vacuum, wet conditions (in environmental SEM), and at an extensive variety of cryogenic or raised temperatures. To obtain 2D images, electron beams are reflected onto different locations in order to scan the whole surface. Utilizing a special type of SEM technique which is called Serial Block-Face Scanning Electron Microscopy (SBFSEM) acquisition of 3D volume can be achieved (9). In SBFSEM technique, cutting ultra-thin section from the surface using a diamond knife and imaging by SEM a stack of 2D images are obtained (See Figure 4). And a 3D volumetric data is formed by a series of these 2D images which associated to a single slice (See Figure 5), meanwhile, there is no limitation for the number of slices and specimen thickness. Typical image size of 3D datasets provided by this technique is 5 – 10 nm in x- and y-axes and 20 –80 nm in z-axis (10)(11). Comparing with TEM tomography, the slice thickness can be much higher than the lateral resolution.
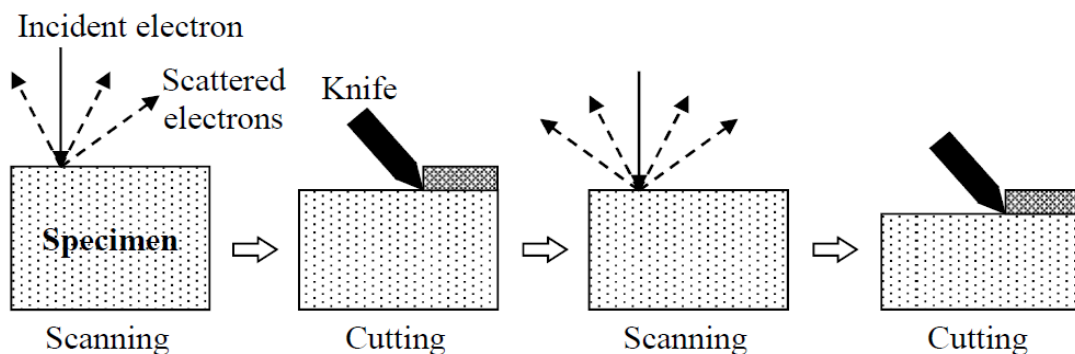
Figure 4: Scanning and cutting phases in Serial Block-Face Scanning Electron Microscopy (12)
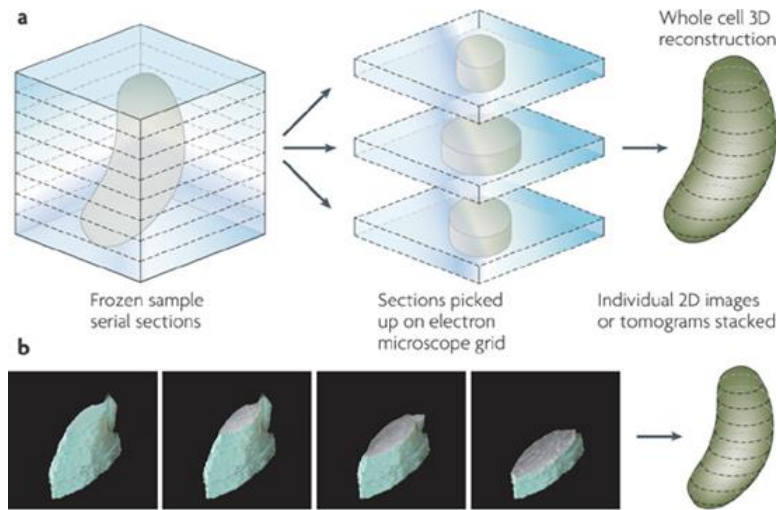
Figure 5: 3D volumetric images consist of several 2D images (slices)(16).

1.3. Motivation

It is progressively ending up noticeably obvious that mitochondria have critical part in neural capacity. Late reviews demonstrate that mitochondrial morphology is urgent to cell physiology and synaptic capacity and a connection between mitochondrial surrenders and neuro-degenerative illnesses is unequivocally suspected (13). The intracellular components' morphology is of great importance for neuroscientists. Characterizing the distribution and structure of mitochondria is useful in itself, to identify cellular states such as cancer due to the geometrical properties of mitochondria and also abnormalities in mitochondria morphology can be seen in Parkinson disease. High-resolution microscopy techniques has been used to generate large volumes of data with enough details for understanding the complex structure of mitochondria (14),(15). An accurate mitochondria detector would allow to clearly distinguish mitochondria membranes from other cell membranes and also will improve overall cell segmentation in the future.

Since manual segmentation in medical applications is very slow, tedious, time and energy consuming, and also needs training for the segmentation and having subjective results and, in some cases fully automatic methods fail and cause incorrect results because of restrictions imposed by image acquisition, pathology and biological variation in medical images and fully automatic segmentation cannot deal with artifacts on its own. All of these issues lead the semi- automatic method to be the preferred method of all. On the other hand, applying semi-automatic segmentation methods do not guarantee the result of the segmentation and in order to improve and correct the result of segmentation methods, it is needed to use editing tools to make better them. In this study two editing tools are developed to apply on the results of the automatic segmentation methods and refine them.

5

1.4. Problem Statement

Since there are often close to hundred slices per mitochondria, hand segmentation is tedious and time consuming and also needs expert users to do it hence, it can lead to many mistakes on the results of the segmentation. On the other hand, using fully automatic segmentation leads to segmentation errors.

Çoçelli (17) and Alamdari (5) have applied semi-automatic segmentation method on the mitochondria since in these methods, the user only interacted in the first steps of segmentation which was done on the initialization part. After the initialization, the user could not interfere in the segmentation process and consequently, could not direct the process to the right segmentation while it created wrong and inaccurate results because of the noise in the image texture.

This study aimed to create 3D editing and correcting tools in order to make better the results of the mentioned methods. In the most semi-automatic segmentation methods such as above, the user can only interfere at the first steps of segmentation while some editing tools are needed to modify and refine the result of segmentation methods. Merge and split tools provided by Çoçelli were used to split the two mitochondria which were considered as a single mitochondrion and also merging the parts of a mitochondrion which were segmented as two different mitochondria but these tools could not be used to modify the part of the segmented boundary and again tools were needed to modify, edit, and make better these results and minimize the segmentation error. In this study, two 3D editing tools are provided to overcome this difficulty and make user able to edit the segmentation results by modifying a part of the boundary and change it until it satisfies the user. Alamdari has implemented three semi-automatic segmentation methods using active contour, Kalman filter with the aid of active contour, and Kalman filter with active contours and optical flow. In these methods, the first slice of mitochondrion was initialized by user to form the contour of snake and in the next slices, these initial points were used to calculate the membrane of mitochondria using the automatic algorithms specified. In some of the datasets, there are mitochondrion which split into two mitochondria or merge after some slices. In these methods, by specifying the merging or splitting slices, the opportunity was given to the user to correct the segmentation by specifying the splitting/merging points of mitochondria. In other words, by determining the splitting or merging slice, when the splitting slice number is executing, the program asks the user to specify the splitting/merging points.

1.5. Aims and Scope

In this thesis, it is aimed to develop a few user interactive editing tools in order to modify and refine the results of automatic and semi-automatic segmentation of mitochondria from TEM tomography images which were provided by NCMIR from their Cell-Centered

Database (CCDB) (18–20) in which each data set has hundreds of slices and each of slices consists of approximately ten mitochondria. Surface dragging and live-wire segmentation tools were implemented to provide these editing tools in two dimension and three dimensions. Applying these editing tools in 2D and expanding its effect on the all slices made it possible to apply these tools on 3D, too. Using surface dragging, helped us to modify the boundary without depending on the texture structure of the images and caused to eliminate the effect of the noise factor in the results while live-wire tool, focuses on the texture characteristics of the image and it was affected badly by noise and edge like details in the images.

1.6. Literature Review

Image segmentation is actually extracting the object of interest from the background image. The goal of segmentation is to simplify and also change the representation of an image into segments that are more meaningful and easier to analyze (21). In the following section a brief review for automatic and semi-automatic segmentation methods is provided.

1.6.1 Automatic Segmentation Methods

Since electron microscopy (EM) imaging techniques generates nano-scale images, the manual analyzing is infeasible and automated image analysis is required (22). Furthermore, completely programmed strategies are testing in light of the fact that various intracellular parts have sporadic shapes and have comparable neighborhood appearances. Texture and physical topologies of intracellular components varies a lot, and applying general image segmentation methods which have been proposed for natural image datasets on EM images leads to poor results (23). Therefore, segmentation methods for segmenting specific structures such as mitochondria must be optimized for EM images.

SeyedHosseini et al (14) proposed a method in order to segment EM images which is composed of four steps as follows: curve fitting, feature extraction, detection and pixel labeling. In the proposed method. Since the algebraic curves use all the information in a window and also are robust against noise and texture, main idea in this method was to use the power of them to extract both shape and textural features from input images. Using algebraic curves leads to use regional features which are more informative compared to pixel-wise features. The removed elements were utilized to prepare arbitrary woodland which recognizes mitochondria in information pictures. At long last a programmed pixel marking methodology is connected by finding associated parts of focus pixel in the patches that the classifier arranges them as positive examples. Testing the performance of current method on two different sets of EM images: mouse neutrophil and Drosophila ventral nerve cord (VNC) shows that this method is robust against the texture and noise in the EM images and also performs reasonably well even for low quality datasets like Drosophila VNC dataset.

Richard J Giuly et al (24) developed a multistep automatic segmentation method to segment the mitochondria from its background in serial block face SBFSEM data by using texture and shape information to identify mitochondria in a complex background which is consist of three steps. They demonstrated that texture based methods for mitochondria segmentation can be done with multiple steps that form an image processing pipeline. In their method (Cytoseg), firstly random forest classifier is applied directly to pixel values of 2D patches to recognize texture, and it had good performance as a patch classifier. Secondly, the detected contours in the probability map from step 1 are refined by using a contour pair classifier and finally a level set procedure is operated on the probability map from step 1 and seeded with salient contour regions from step 2 to smooth the final results. Applying this Cytoseg method in three various types of mouse brain tissues: cerebellum, dentate gyrus, and the CA3 division of hippocampus, shows that the patch classifier identifies the mitochondria texture but it creates many false positive pixels and by filtering them with a second classification step improves the overall accuracy. The level set operation which is consequently seeded with yield from past strides, smooth's the outcomes. Absolutely, the outcomes demonstrate that the utilization of form combine characterization and level set operations enhance division exactness contrasting and the aftereffects of applying patch arrangement alone.

Exactness in division of EM pictures is an imperative issue for some neuro imaging related undertakings because of the commotion and variety inalienable in tissue tests and in the EM science and image acquisition process. And on the other hand, using supervised methods requires large sets of training data which are often difficult to obtain in practice, Navlakha et al proposed an unsupervised algorithm to generate boundary-preserving super pixels by combining a salient watershed algorithm with robust region merging (25). In the proposed method, an unsupervised segmentation algorithm is designed specially to be robust to the types of variations and noise expected in EM images of brain tissue. At the first step, a novel watershed variant method using Canny (26) and probabilistic boundary (27) edges is developed to produce a coarse over-segmentation while strongly preserving edges in the image. And then a new region merging algorithm to merge the adjacent regions based on measure of similarity derived from intensity and texture features, is designed to reduce the number of super pixels. The proposed strategy on a benchmark datasets of loud EM images, beat two condition of-workmanship techniques utilizing two standard measures of over-and under-division blunder: Although this strategy has extra computational many-sided quality, the accuracy and time spent in EM images can be considered as an advantages of this method.

Seyedhosseini et al presented a novel strategy utilizing multi-scale relevant data with Random-like elements (RLF) to take in a progression of discriminative models keeping in mind the end goal to recognize the unbiased films in electron microscopy images (22). In this technique, the principle thought is to take preferences of setting images as various scales rather than a solitary scale which prompts furnishing a classifier with a wealthier arrangement of data. RLF is altered with a specific end goal to concentrate more data from various structures of the info images. Contrasting and a solitary scale display, the utilization of a multi-scale portrayal of the setting images gives the ensuing classifiers access to a

bigger relevant zone in a viable way. Applying this general technique to layer recognition in EM images, the outcomes shows that the proposed strategy outflanks condition of-workmanship calculations while keeping up about indistinguishable computational unpredictability.

A fully automatic modular hierarchical approach for neuron spatial structure segmentation and reconstruction using EM image is introduced by Ting Liu et al (28). They proposed a hierarchical 2D segmentation method with the merge tree structure and supervised classification that uses membrane probability maps as input then a supervised linking method is used to acquire inter-section reconstruction of neurons and also a semi-automatic 2D segmentation method, which takes advantages of the automatic intermediate results and improves 2D segmentation with minimized user interaction, is designed. Comparing the use of merge tree with the use of general graph structure, it is shown that the merge tree structure presents the most plausible segmentation hypotheses in more efficient way. According to the experimental results, the proposed automatic merge tree approach improves the 2D neuron segmentation accuracy substantially over thresholding the membrane probability maps at the best thresholds.

Cory Jones et.al exhibited an incomplete differential condition (29) with a novel development term to enhance the consequences of a managed learning calculation and furthermore acquainted another technique with speak to the subsequent pictures that takes into consideration a more powerful threshold t additionally enhance the outcome (30). Using these two process, the medium sized gap s in the cell membrane detection can be closed and also the error can be rand by as much as 9% over the initial supervised segmentation. These two processes are applied on two different data sets where the fist data set is a stack of 60 images from a serial section Transmission Electron Microscopy (ssTEM) data set of Drosophila first instar ventral nerve cord (VNC) and the second data set used for these experiments was a stack of 70 images of mouse neuropil. The test results demonstrate that when the holes are too vast, for example, in mouse neuropil dataset or there are not very many crevices to close, with the DNN likelihood maps on the Drosophila dataset, the change is not articulated.

Tasel (12) introduced a study that was based on curve fitting in order to extract mitochondria peripheral membrane from the back ground image. This algorithm started with the preprocessing step consist of applying resampling, smoothing and auto-contrast adjustment on the input tomography images. Then membrane strength was evaluated by a Hessian-based ridge detector. In order to generate energy maps, the summation of local energy within two windows with different size was calculated. The parabolic arc fitting operation in which the energy maps were used to determine the segments of mitochondrial peripheral membranes and cristae. These curves were utilized in a snake algorithm which extracts candidate segmentation regions. And at last, the results were fined by a validator function.

1.6.2. Semi-Automatic Segmentation Methods

Olabarriaga et al (30) have discussed existing interactive methods with the respect to the following aspects: the type of information provided by the user, how this information affects the computational part, and the purpose of interaction in the medical images segmentation process. The impact of each strategy on the accuracy, repeatability and interaction efficiency was analyzed. A sketch of a general interactive segmentation method is shown in figure (6). The main components are the user, the computational part, the interactive part, and the user interface. The capability of interactive methods was analyzed based on the type of user input, the consequence of user input, and the purpose of interaction and the role played by the user (31).
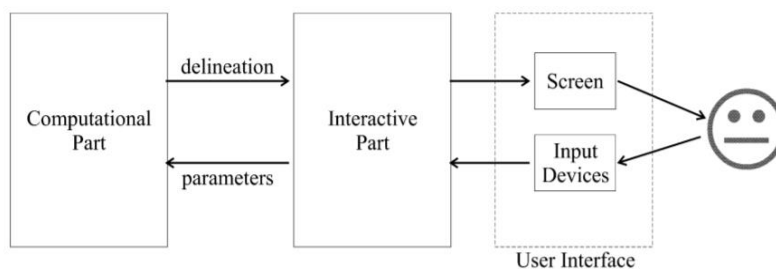
Figure 6: Main components of an interactive segmentation method (30).

The computational part corresponds to one or more part of program which is capable to generate a delineation of the object of interest given by some parameters. The interactive part is actually responsible for the intervention between the user and the computational part. This part translates the outcome produced by the computational part into visual feedback to the user and the data input by the user into parameters for the program. While the output and input devices controlled by the user interface do the actual communication between the computer and the user. The visual information displayed on the screen is analyzed by the user and the user reacts, providing feedback for the computation.

Reviewing several semi-automatic segmentation methods, three main types of user interaction during the interactive segmentation process were identified as: setting parameter values in a continuous or discrete interval; pictorial input directly on the image grid, and choosing from pre-defined options in a menu. Setting parameters may be simple to implement, but it needs an insight of the user in the function of the computational part of the method. This drawback can be minimized by training the human operator, or by immediately visualizing the effect of interaction on the screen. Although pictorial input is simple for the user, it is typically time consuming. Menu-driven input is the most efficient user interaction type. It limits the choices, and eliminates the need for hand-on manipulation of parameter values or the image on the screen. In menu-driven, the variation in the input data is reduced by constraining the degrees of freedom, therefore leads to more repeatable results. For all user interaction types, the program must provide visual feedback

to the user in tow moments: before the interaction to help the user plan the next action; and after the interaction to inform the user about the impact of the interactions.

Considering semi-automatic segmentation methods, five roles were identified for the user: accepting, or rejecting the delineation generated by computational part, correcting the outcome of the computational part, initializing or steering parameters for the computational part, selecting among the results generated automatically, and defining targets for the generation of a computational method.

The role of editing the result is considered to guarantee the accuracy even when the computational part cannot generate the correct delineation. The purpose of parameter initialization is to bootstrap the computational part in an efficient way. The purpose of steering is to keep the user in control of the entire segmentation process such that manual editing at the end is not necessary. Selecting among the results is considered to let the computer find a number of possible delineations and let the user decide on the most reasonable ones.

In their study, interactive segmentation methods were discussed based on accuracy, repeatability, and efficiency, too. Accuracy indicates the degree that to which the delineation of the object correspond to the truth. In order to get subjective evaluation of the accuracy, human experts rank the result generated by the computational method when operated by others. Computing objective evaluation, the results generated by the segmentation method is compared against the ground truth using different distance measures (30). Actually, a method is potentially accurate that provides full control to the user to generate any desired result. Another measurement is repeatability which evaluates to what extent applying the same method, we can obtain the same results. In order to obtain this metric, one human operator segments the same image and object several times and the results are compared. Variability in the results can be due to two factors: difference in the operation of the segmentation tool for example user clicks the mouse at different positions on the image, or difference in judgment which means the user considers the delineation to be located at different positions each time. Totally, a method generates which takes precautions to minimize the effect of the first type of variation.

The total elapsed time could be used as an indicator of the methods overall efficiency. For the computational part, the efficiency is the time needed by the computer to generate the result. For the interactive part, efficiency is inversely, proportional to the effort required from the user to accomplish the segmentation task. This effort is the amount and nature of user interventions. The amount of interaction is often estimated in terms of the numbers of mouse click. Due to the large number of components to manipulate, the selection and combination of primitives can be efficient. The purpose of the role of building the computational method is to let the user establish targets that guide the construction of an appropriate combination of operations to accomplish a given task.

In order to design an efficient interactive segmentation methods, the following strategies seem promising: Designing an integrated process for interaction and computation; using

pictorial input to the computational process; minimizing the amount of interaction by presenting options for user selection; involving the user in the initialization of the segmentation process in order to provide information can lead the method to the desired segmentation result more quickly; visualizing the working of the computational part to enable an effective user's response; keeping the user in the control during the whole process to generate accurate results; emphasizing computation after each interaction to generate repeatable results; adding intelligent behavior to evaluate the abstraction level of interaction; and adding intelligence to learn from interaction and reduce the need future interventions.

Paul A.Yushkevich et al (32) introduced an open source medical image processing application which is called ITK_SNAP and gives a mix of manual and semi-automatic tools for separating structures in 3D images information of various modalities and from various anatomical areas In this application, two well-known 3D active contour segmentation methods: Geodesic Active contours by Caselles et al (1993, 1997) and region competition by Zhu yuille (1996) were implemented. The user interface consists of three orthogonal views of a volumetric image and also a 3D view is provided to render the boundaries of segmented structures in visualization Toolkit (VTK) (Lorensen and cline, 1987; Schroeder et al., 1996). By providing live feedback and decreasing the number of unrelated features it is tried to maximize user efficiency and to provide a smooth learning curve. Using this application in order to validation in the context of caudate nucleus and lateral ventricle segmentation in child MRI demonstrates excellent reliability and high efficiency of 3D SNAP segmentation.

Malmberg et al (33) developed a tool to use Smart Paint in interactive segmentation methods. Since segmenting noisy images using brush tool alone leads to produce segmentation results with jagged boundaries, a smoothing operation which can be applied to the entire segmentation is provided as a preprocessing step. At the segmentation step the user adds or removes paint to segment the desired part by pushing the left mouse button or mouse right button respectively. Scrolling to the other slices is provided by using the mouse wheel. The paint layer can be turned on/off by pushing space bar. The brush size can be changed by ctrl and mouse wheel. If desired, the resulting boundary can optionally be smoothed by a 3*3 operator by pushing ctrl and A.

In order to evaluate the segmentation tool, it was used to segment the prostate image. As a quantitative measure of difference between the segmentation result and the ground truth available, Dice coefficient was used. Considering the required time using smart paint for segmentation, the minimum time was 2 min, 30 seconds, and the maximum time was 8 minutes and the accuracy of the results depends on the knowledge of the user about the anatomy of the prostate. Using this tool on completely different applications show that the smart paint is very effective tool for segmenting objects in 3D volume image even if only 2D display and interaction tools are available.

Takumi higaki et al developed a successful method for semi-automatic detection of plant organelles including mitochondria, amyloplasts, chloroplasts, etioplasts, and Golgi stacks

in transmission electron microscope images using their original image recognition framework clustering Aided Rapid Training Agent(CARTA) (34), combined with procedures to highlight and enlarge regions of interest on the image (35).

In this method by analyzing a collection of similar subdivided regions of TEM images the organelles could be detected semi-automatically. Firstly, square regions of interest, designated as 'windows' which are randomly located covering the entire TEM image. Collecting the similar windows are done as it is done in CARTA Framework. Then cluster analyses of the windows in the self-organizing map which refers to as the 'initial tiled map' and serves as the interface for users to manually search for target organelles, were performed based on the randomly selected features. User assigns the clusters and then the target window nodes are selected. Then the target in the TEM image is detected. This method was applied on different datasets of TEM images of Arabidopsis thaliana root tip tissues containing amyloplasts, A.thaliana embryo containing chloroplast, A.thaliana cotyledon tissue. The results showed that the proposed strategy was flexible and very precise technique to identify organelles in TEM images.

Cocelli(17), using curve fitting and active contour methods implemented a semi-automatic editing tools in order to edit the results of automatic segmentation methods on transmission electron microscopy images. The developed editing tools were splitting, merging, deletion, initiation of automatic segmentation algorithm at intended position, selection of low scored mitochondria and manual drawing. Splitting tools were used in order to split two mitochondria which were wrongly detected as a single mitochondrion by automatic segmentation method. The merging tool was used to merge the parts of mitochondrion which were detected as two different mitochondria. Deletion tool give the opportunity to delete the false detection which were done by automatic segmentation method. Applying these editing tools on the results of automatic segmentation method led to rise the Recall value from 45% to 70% which shows considerable improvements.

Alamdari(5), purposed semi-automatic segmentation methods by implementing active contour, Kalman filter with aid of active contour, and also, Kalman filter with active contours and optical flow. In these methods, user interacts by specifying the initial points on the membrane of mitochondria. These initial points are used to calculate the membrane of mitochondria in the proceeding slices using above automatic algorithms. Kalman filter uses the initial points' location on the current slice to estimate their location on the next slice. In the last method, the velocity information is provided by optical flow. In these methods the opportunity is given to the user to correct the segmentation by specifying the splitting ad merging points of mitochondria. When the splitting/merging slice number is executing, the program asks the user to specify the splitting/merging points. The results of these methods were very similar to each other. In the most cases the Dice coefficient showed accuracy more than 90% which means that these results were very close to the ground truth images. The RMSSSD metric was near to 1 approving that these results were almost the same as the ground truth images.

CHAPTER 2

Methods

In this thesis, two modification tools were developed in order to make correction or refinements on the boundary in to overcome the abnormalities caused by automatic segmentation methods: contour dragging using splines and live-wire interactive tool. Earlier a few other semi-automatic segmentation tools were developed for this purpose. Cocelli (17) developed semi-automatic merge and split tools to apply the results of automatic segmentation methods. Alamdari has developed a semi-automatic segmentation method using active contour, Kalman filter, and optical flow to segment the mitochondria. In her study, a contour was provided by the user. Then for other slices, the position value and velocity values were calculated using active contour and optical flow combined with Kalman filter to predict the boundary points on the next slice. Even though they were quite successful in order to segment the mitochondria from the background images but in some cases, the segmented boundaries were not correct. Additional interactive tools were necessary to edit the results of these methods. In this thesis, two editing tools using surface dragging using splines and live-wire method are provided to edit the segmentation results of these methods.

2.1. Surface Dragging Using Splines

In this method, the segmented surface inside the region of interest (ROI) is dragged using the multiple cubic B-splines (36).

In this method ROI is determined by a circle and using spline method the curve which passes through the center of ROI which is called control point, and the two intersection points of existed boundary, and ROI and also the midpoints of the lines between control point and intersection points is considered as desired boundary (Figure 7). The user can control the new boundary part by moving the mouse cursor as a control point which is considered as dragged control point.

Using Bernstein polynomial of degree 3 which is called Bezier cubic, the curve can be obtained:

$$z(t) = (1 - t)^3 z_0 + 3 (1 - t)^2 tz_1 + 3(1 - t)t^2 z_2 + t^3 z_3, \quad (1)$$

Where t varies between 0 and 1 and Z_0, Z_3 are the intersection points of existed boundary and ROI, and Z_1, Z_2 are the midpoints of the lines Z_0- Z_C and Z_3- Z_C considering ZC as a control point. After determining ROI, when the user moves the mouse scroll, the position of the mouse is considered as dragged position of the center of ROI and the curve is calculated and updated according to the new values of the Z_0, Z_1, Z_2, Z_C, Z_2, and Z_3. It is illustrated in Figure 6.

Z_c: is the dragged position of the center of ROI.

Z_0, Z_3: the intersection points of ROI and existed boundary

Z_1: the midpoint of the vector between Z_c and Z_0

The user can control and change the desired boundary by moving the mouse scroll and whenever he is satisfied by the new boundary, he can fix it by double clicking on the image.
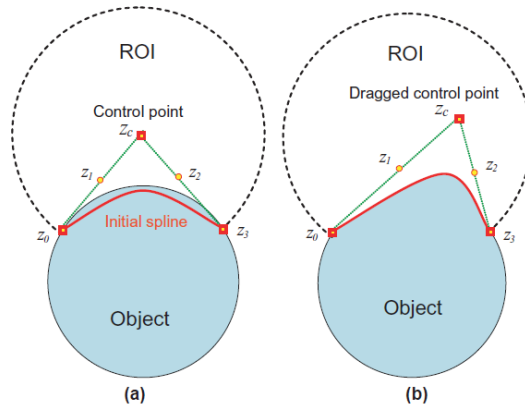
Figure 7: Region of interest and dragged center of it (36).

16

2.1.1. 3D Surface Dragging Using Splines

In order to apply this method to 3D space, instead of ROI, volume of interest (VOI) is considered. In the implementation, VOI is considered so that the center of sphere is a point on the center slice and it is obvious that the intersection of this sphere with each slice is a circle (figure 8) and the radius of this circle, 'a' is calculated using equation 3 and the details were shown in figure 9.

Considering:

$$R^2 = a^2 + (R - h)^2 \tag{2}$$

$$a = \sqrt{2Rh - h^2} \tag{3}$$
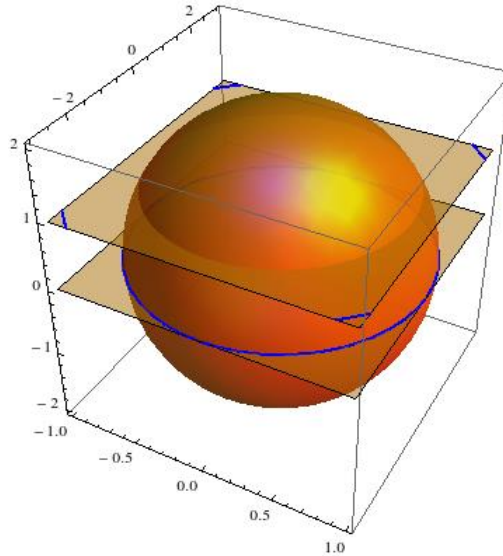
Figure 8: The intersection of each slice with the VOI is a circle in 2D with the center in the VOI center coordinates and radius a.
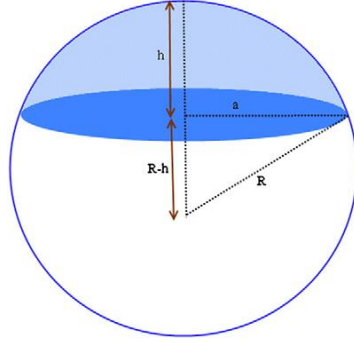
Figure 9: Calculating the radius of volume of interest in the intersection part with each slice (35).

In 2D, on each slice we have the control point which is fixed(center of ROI) and "a" the radiusof ROI which is calculated by Equation(3). Therefore the the bazier curve can be calculated.

Actually, the user has to do the refinment on the center slice in 2D and then by pressing "apply_all buttun" , "a" for each slice is calculated and the Z_0, Z_1, Z_2, Z_C, Z_2, and $Z3$ can be obtained and the new boundary part is updated on each slice which leads to apply refinement on mitochondrion in 3D.

In 3D, instead of spherical VOI, we can use cylindrical VOI. In this way the ROI has the same radius in all slices. Using this VOI, because of the changes in the size and the shape of the mitochondria through the slices, in some cases there is not intersection between the ROI and the mitochondria either. So using cylindrical VOI may or may not improve the edits.

2.2. Interactive Live-Wire Boundary Extraction Tool

As the second method for editing segmentation method the live-wire method (37) is used. In this method boundary segmentation is figured as a graph searching problem. The nodes are pixels and the edges are made between a pixel and its eight neighbors. The target is finding the optimal path between a start node and a set of objective nodes. Boundary is defined as optimal path with minimum cumulative cost path from a start node to a global node where the cumulative cost of path is sum of the local costs of nodes on the path and is defined as below where $f_G(q)$ is the gradien magnitute , f_z is zero crossing, and f_D is the gradient magnitude between pixels p and q:

$$l(p,q) = \omega_G \cdot f_G(q) + \omega_z \cdot f_z(q) + \omega_D \cdot f_D(p,q) \qquad (4)$$

18

In equation 4, l(p,q) is the local cost for directed link from pixel p to a neighbor pixel q and ω_G , ω_Z , ω_D are their weight and are set to 0.43,0.43,0.14 respectively as suggested in (37).

In order to calculate the zero crossing (38), firstly the image is convolved with a laplacian edge operator(39). Laplacian is 2nd spatial derivative of an image which lays emphasis on the regions with sharp changes. It can be used to detect the edges.

For an image with pixel intensity values I(x,y), the Laplacian is calculated as:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \qquad (5)$$

We can calculate this equation using a convolution filter. The input for this filter is single gray level image. The output is another gray level image. Three commonly used discrete approximations to the laplacian filter are illustrated in figure (10).

0	1	0
1	-4	1
0	1	0

1	1	1
1	-8	1
1	1	1

-1	2	-1
2	-4	2
-1	2	-1

Figure 10: Three commonly used Laplacian filters.

Positive Laplacian Operator is a standard mask with negtive element in the center and corner elements of zero and negative Laplacian operator is a standard mask, with center element of positive and zero elements in all coeners and all other elements in the mask are -1. Applying Laplacian kernel on the image leads to reduce the amount of data (pixels) to process and maintains the "structural" aspect of the image. The output of convolution of the image with a laplacian operator is considered as I_L. Therefore zero crossing is applied on output to make it binary by letting $f_Z(I)=0$ where $I_L(q)=0$ or has a neighbor with a different sign and $f_Z(q)=1$ otherwise.

In other words in order to calculate the zero crossing which is an important step in the Laplacian edge detector for each pixel I(x, y) the following steps are done:

1. Considering four neighbors, left, right, up and down.

2. If they all have the same sign as pixel I(x, y), then this pixel is not a zero crossing.

3. Else, if pixel I(x, y) has the smallest absolute value compared to its neighbors with opposite sign, then it is a zero crossing.

The gradient of an image providing two pieces of information measures how it is changing. The magnitude of the gradient determines whether the image is changing quickly or not, while the direction of it demonstrates the direction in which the image is changing most rapidly (40). In this study, using gradient direction, associating a high cost for sharp changes in the boundary direction, it is tried to add smoothness constraint to the boundary. Supposing G_x and G_y as image gradients in x and y directions respectively (equation 6), D (p) which is a unit vector and normal to the gradient direction at point P, is

$$D\,(p) \; = \; (Gy\,(p), -Gx\,(p)) \tag{6}$$

where G_x, G_y are the image gradient in the x,y directions respectively. The gradient direction is calculated with the following equation:

$$f_D(p, q) = \frac{2}{3\pi}\left\{\cos[d_p(p, q)]^{-1} + \cos[d_q(p, q)]^{-1}\right\} \tag{7}$$

and vector dot products are :

$$
\begin{aligned}
d_p(p, q) &= D(p){\cdot}L(p, q)\\
d_q(p, q) &= L(p, q){\cdot}D(q)
\end{aligned} \tag{8}
$$

And L(p,q) is the direction of the link between p and q so that the direction of the link is made normal.

$$L(p, q) = \begin{cases} q - p & \text{if} \quad D(p){\cdot}(q - p) \geq 0 \\ p - q & \text{if} \quad D(p){\cdot}(q - p) < 0 \end{cases} \tag{9}$$

In the cases that, the edge or link between two pixels are perpendicuar , or nearly perpendicular to the link between them but have similar gradient direction, have a high cost. Therefore, the direction feature has low cost in the cases that the gradient direction of the two pixels are like each other and the link between them.

Using first order derivative the gradient magnitude for pixel p with coordinates i,j is calculated using equations 10 and 11. The steps to calculate the first order derivative are shown in figures (10) and (11).
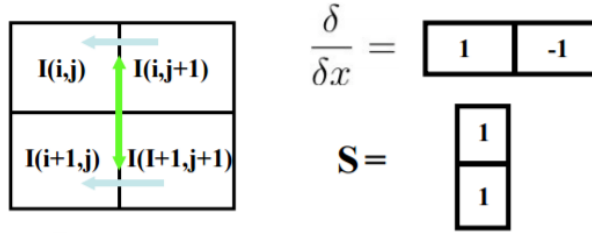
Figure 11: Computing first order derivative- Gx.

$$\frac{\partial I}{\partial x}(i,j) = \frac{1}{2}\Big(\big(I(i,j+1) - I(i,j)\big) + \big(I(i+1,j+1) - I(i+1,j)\big)\Big) \qquad (10)$$
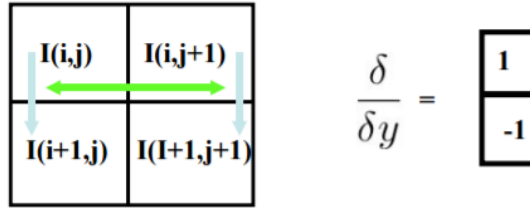
Figure 12: Computing first order derivative- Gy.

$$\frac{\partial I}{\partial y}(i,j) = \frac{1}{2}\Big(\big(I(i+1,j) - I(i,j)\big) + \big(I(i+1,j+1) - I(i,j+1)\big)\Big) \quad (11)$$

The gradient magnitude is scaled and inverted using linear ramp function. The gradient feature f_G is :

$$f_G = 1 - \frac{G}{\max(G)} \qquad (12)$$

In this implememntation the second derevative of the mage was used in rder to calculate the gardient magnitude. But Sobel filter could be used in order to get more precise edges.

2.2.1. 3D Interactive Live -Wire Boundary Extraction Tool

Using live-wire, after selecting the seed point, when the user moves mouse, the optimal path between seed point and current mouse position, which is considered as target pixel, is calculated and determined, and by double clicking this path is frozen and considered as a part of boundary.

In order to apply this method on 3D, the user applies this method on 2D and then by pressing "apply_all_slices" button, the seed point which was determined by user on the first slice, is moved to one of its neighbors with high gradient magnitude on the next slice. Since the seed point and the target pixel are determined successfully (mostly), the optimal path between them on the current slice is again optomally calculated.

21

In order to find neighbor pixel with the highest gradient magnitude, eight neighbors of seed pixel is are checked and then the neighbor with the highest gradient magnitude is considered as the seed pixel for that slice.

2.3. Proposed Methods

In this study, it is aimed to create tools in order to modify parts of segmentation boundaries using dragged spline method and also interactive live-wire method to modify and refine the result of automatic segmentation method. Applying each of mentioned methods separately, could help us to modify segmentation results. In the following sections, tool provided by surface dragging mode was described in details and the required user interactions in order to apply this tool on the mentioned tomography images are discussed. And interactive live-wire editing tool was completely explained in the subsequent section. In figure (14), the user interface for applying surface dragging is shown.

Figure 13: User interface applying surface dragging.

2.3.1 Surface Dragging Using Splines

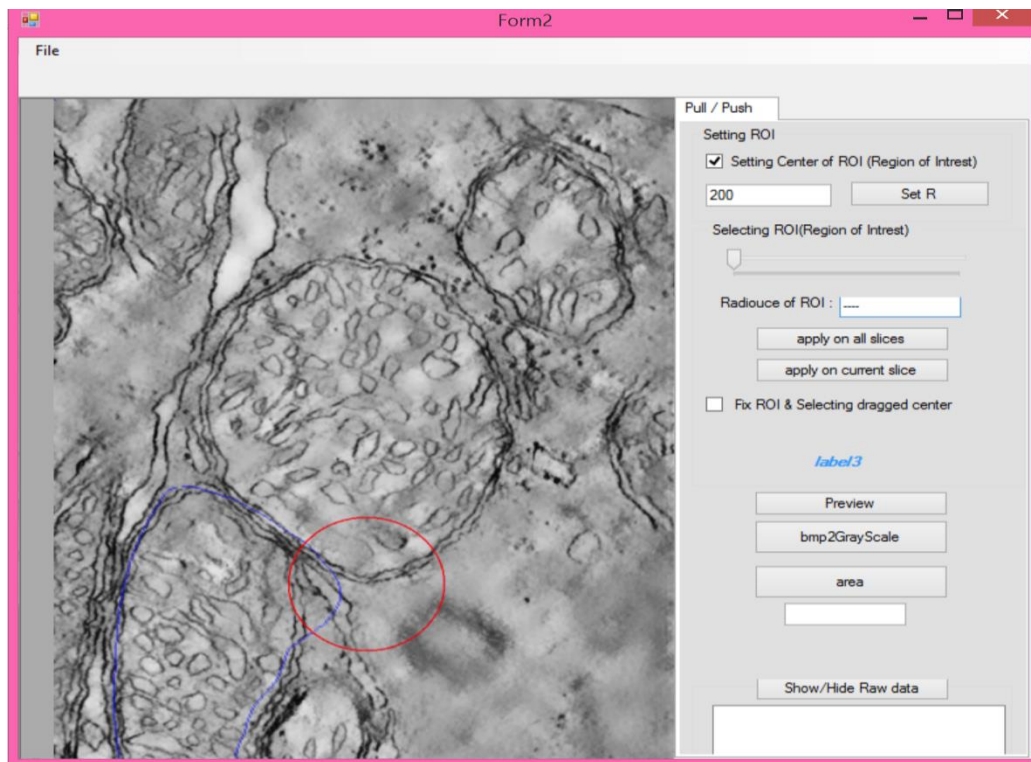In order to refine the boundary extracted by automatic methods, spline dragging method is used. In order to apply this method, the following steps are needed to apply. (See figure 13).

```
┌──────────────┐    ┌──────────────┐    ┌──────────────┐    ┌──────────────┐
│ Input image  │───▶│  Set Radius  │───▶│   Set ROI    │───▶│ Drag the     │
│              │    │   of ROI     │    │              │    │ center of ROI│
└──────────────┘    └──────────────┘    └──────────────┘    └──────────────┘

┌──────────────┐    ┌──────────────────────┐    ┌──────────────────────┐
│ Drawing the  │◀───│ Check whether the new│    │ Update the boundary  │
│ bazier curve │───▶│ boundary part is     │───▶│ if it is accepted    │
│              │    │ acceptable or not    │    │                      │
└──────────────┘    └──────────────────────┘    └──────────────────────┘
```
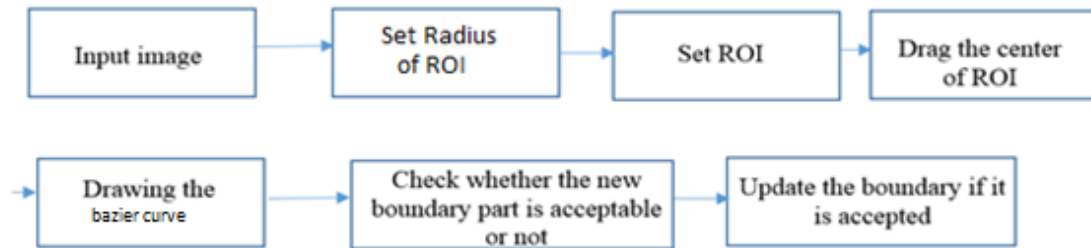
Figure 14: Flowchart of boundary editing using surface dragging.

As it is discussed in the next chapter, setting the proper value for the radius of the ROI not only affects the result of segmentation but also it affects the number of required steps in order to refine the boundary as it is expected.

In order to apply surface dragging method, user at first step has to set the value of radius of ROI which the default value is set to 100. Then using mouse cursor, the ROI is set by clicking on the desired part of boundary which should be edited. By double clicking on the image the ROI is fixed. After setting ROI, user can drag the center of ROI by clicking on the image. The Bezier Cubic (41) which passes through the intersection points of ROI and boundary and also the dragged center of ROI. Whenever the user moves the mouse cursor the dragged center is changed and consequently the new curve is calculated and updated on the image. This part is shown by red color on the figure 15. In the case that the user accepts the curve, by double clicking the curve is fixed. In the 2D step by clicking on the current slice is part of the boundary is updated and saved.

To apply this method on 3D, the dragging method should be done on the middle slice. After fixing the calculated curve user can press apply on-all-slices button. Then according to the following calculations the selected part of existed boundary is updated on the all other slices. The ROI which is considered as a circle in 2D, is handled on the other slices in the way that this area (ROI) is considered as a sphere (VOI).

In order to apply sphere (volumetric) ROI in 3D, the center of ROI in all slices is set to the same coordinate of center of ROI in the middle slice. The value of radius of ROI on the other slices is calculated as equation 3. The total view of all ROIs on the all slices form a sphere whose center is the center of ROI on the middle slice and the radius is the radius of ROI on the middle slice. It is illustrated in details in figure 9.
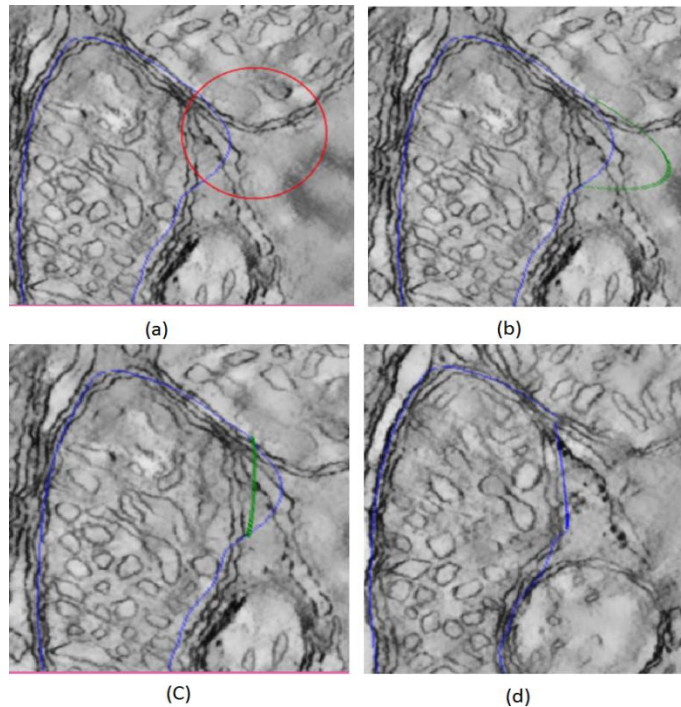
Figure 15: Steps have done for applying surface dragging on cone dataset a) setting ROI b) moving mouse cursor until it is satisfying c) satisfying d) updated boundary (result).

2.3.2. Interactive Live -Wire Boundary Extraction Tool

As it mentioned in the method part, live-wire segmentation tool is figured as a graph searching issue, and the objective is to find the optimal path between a start node and a set of target nodes. The pixels are the nodes of graph. The edges are the path between each pixel and its eight neighbors. Boundary finding is considered as finding the comprehensively optimal path from start node to objective node so that the optimal path is considered as minimum cumulative cost path between them. In this thesis, the optimal path between start node and end node is calculated by using Dijkstra algorithm(42). Dijkstra algorithm is generally used in graph searching problems in order to find the shortest path between two specified nodes. The Dijkstra algorithm is given as below:

The node at which we are starting is called the initial node. The cost of node Y is the distance from the initial node to Y. Dijkstra's algorithm will assign some initial distance values and will try to improve them step by step.

1. Assign to every node a tentative distance value: set it to zero for our initial node and to infinity for all other nodes

24

2. Mark all nodes unvisited. Set the initial node as current node. Create a set of the unvisited nodes called the unvisited set consisting of all the nodes.

3. For the current node, consider all of its unvisited neighbors and calculate their tentative distances.

4. When we are done considering all of the neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.

5. If the destination node has been marked visited (when planning a route between two specific nodes) or if the smallest tentative distance among the nodes in the unvisited set is infinity (when planning a complete traversal; occurs when there is no connection between the initial node and remaining unvisited nodes), then stop. The algorithm has finished.

6. Select the unvisited node that is marked with the smallest tentative distance, and set it as the new "current node" then go back to step 3.

In this thesis, the nodes are the image pixels. The distance between nodes p and q is the cost associated to the directed link from pixel p to a neighbor pixel q. This local cost is calculated by equation 4. Using Dijkstra algorithm, the optimal path (shortest path) between start node and goal node is obtained.

Since we are trying to edit only an especial part of image, we try to find the optimal path in a small part of the image. It is not necessary to calculate the path between start node and all other nodes (start pixel and all other pixels) on the image. The image size is very huge for this calculation and also calculating the path for all pixels is time consuming. In order to apply interactive live-wire segmentation tool, it is needed to set ROI in to avoid unnecessary computational time. After setting the ROI, by clicking on the ROI the start nod is set. According to the following formula for calculating the local cost function the costs are calculated for graph which is formed by the pixels of ROI and the edges between each pixel and its eight neighbors. Then by clicking for the second time the goal node is set. Then the minimal cumulative path between the start nodes and goal node is calculated. If the user is satisfied with the optimal path, this path can be saved on the current slice.

In 3D, in order to apply this method, again it is needed to apply this method on the middle slice and then by applying the method on the all slices the ROI for each slices is set so that the over view of the slices on all slices forms a sphere. In this method in order to initialize the algorithm on the other slices the first node which is set by user's mouse click, is carried on the pixel of the next slice and then by considering the 3*3 neighborhood of this pixel, the pixel with the highest gradient is considered as start node and also the goal node is considered as a pixel with the high gradient magnitude in 3*3 neighborhood.

2.4. Data Set

In this experiment datasets from CCDB provided by the National Center for Microscopy and Image Research (NCMIR) are used (18–20). Eight data sets which demonstrate variety in image contrast and mitochondrial membrane and crista patterns were chosen in which each dataset approximately consists hundreds of slices and each slice contains several mitochondria. In this data set collection, mitochondria appearance and shapes and sizes differ significantly among the datasets as it is illustrated in Figure 16. Some attributes of the datasets such as voxel size, dataset thickness and image size are listed in Table (1). Totally, there are 622 slices and 96 mitochondria in the dataset collection. Sample TEM images of mitochondria from this dataset are shown in figure (16).

Table 1: Properties of dataset utilized as a part of the analyses

Dataset base name	Image Size (width * height)	Number of slices	Number of Mitochondria per slice
6_22.sub	1960 * 2560	91	14
bcpl-d.sub	720 * 878	61	6
Cone.sub	736 * 1010	97	9
gap18_sub	350 * 600	54	6
Mac_serial_sub	907 * 1172	111	20
Od.sub	1960 * 2560	91	34
pedicle	950 * 1280	31	6
Spherul24mos1	1996 * 1996	86	1
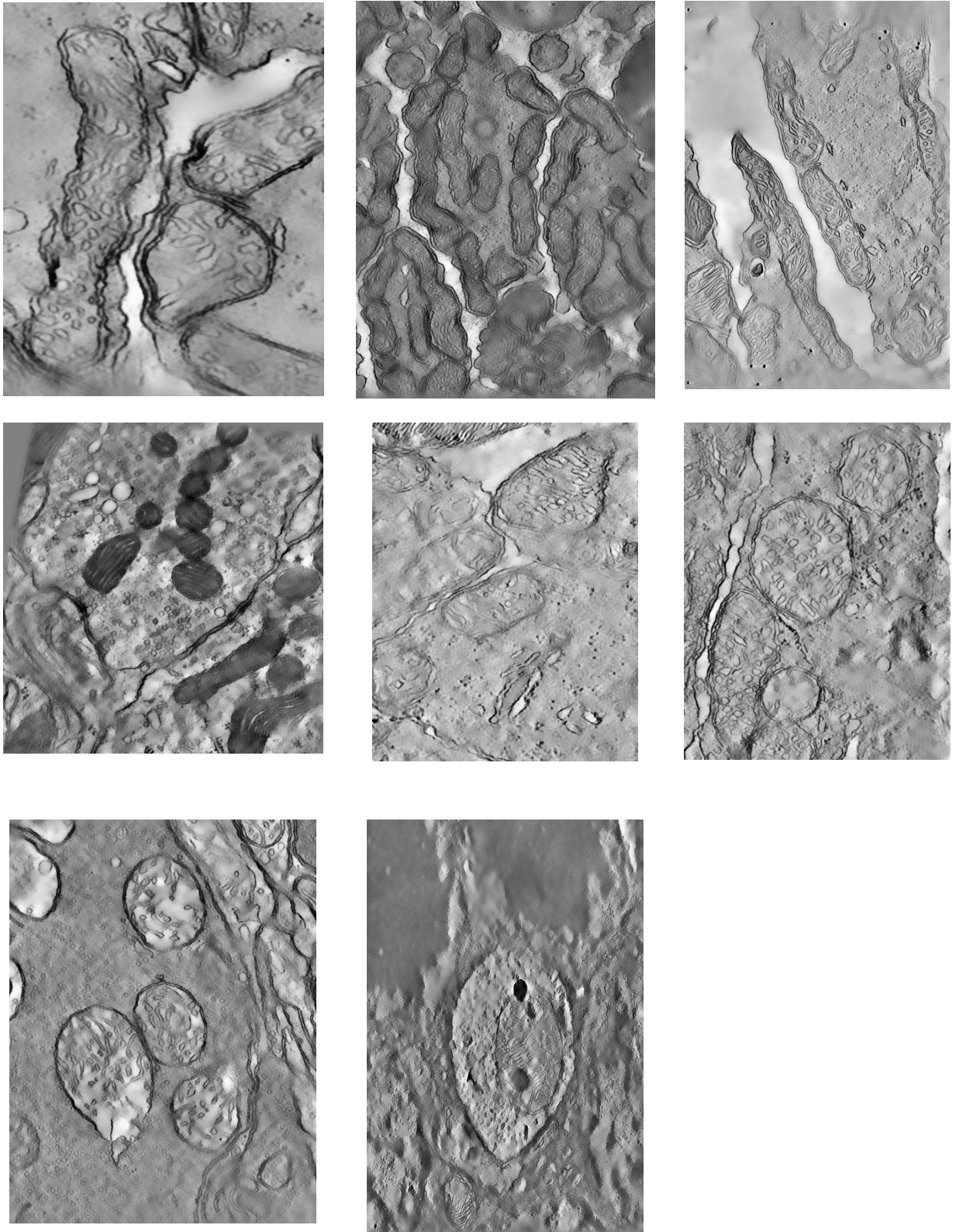
Figure 16: Sample TEM images of mitochondria from the used dataset (17–19).

In figure (17) some outputs of automatic segmentation method are shown.
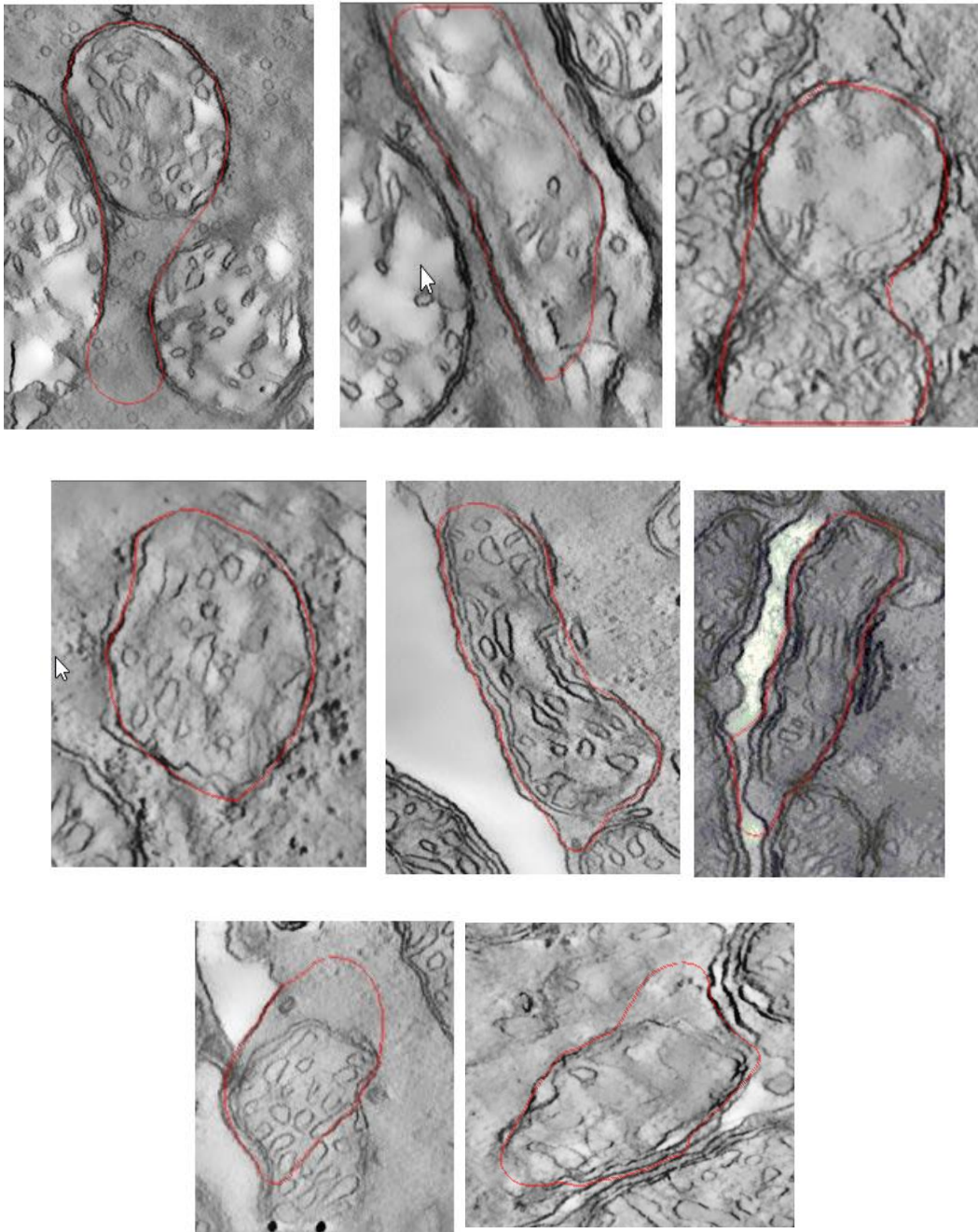
Figure 17: Samples of automatics segmentation method's output.

Sample test cases from 'pedicle' dataset are shown in figure (18). The ground truth images corresponding to this sample test cases are provided in figure (19).

a) mitochondion #1   b) mitochondrion #2

mitochondorion #3
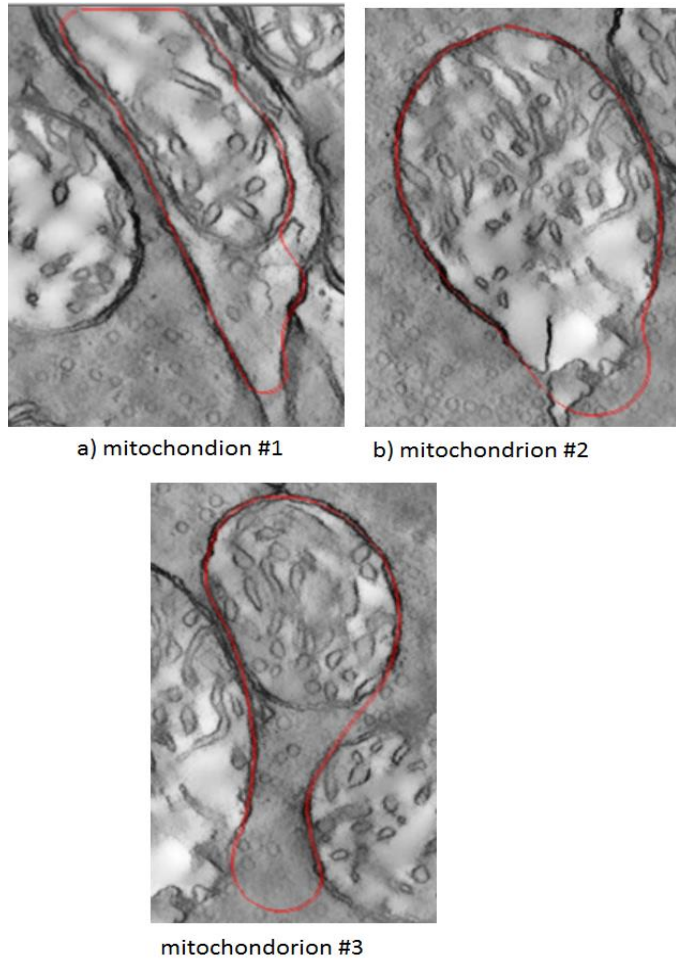
Figure 18: Sample test cases from pedicle data set. The red contours show the result of automatic segmentation by Tasel (12).

a) mitocondrion #1   b) mitochondrion #2

c) mitochondrion #3
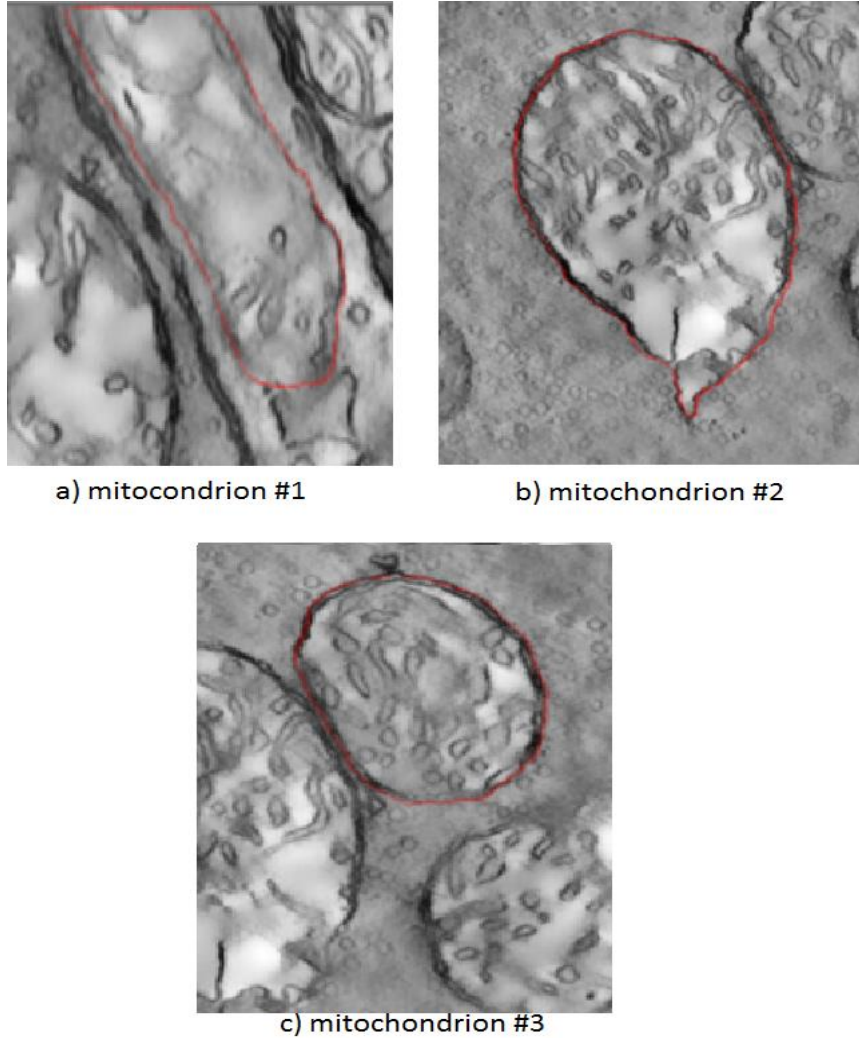
Figure 19: Ground truth images corresponding to sample test cases above.

2.5.Ground Truth

Ground truth is a term which is utilized as a part of different fields to allude to data gave by direct perception instead of data given by induction (43). Simply it means a set of measurements that is known to be much more accurate than measurements from the system that is testing.

In this field, ground truth images are images in which the desired object is segmented from the background by experts and these boundaries are considered to be accurate and reliable, and also are used to check or calibrate a recognition or segmentation system (43). These ground truth images were obtained by specifying the outer membranes of mitochondria via a manual segmentation process using IMOD software (12) . An expert in University of California, San Diego has done the markings figure 19.
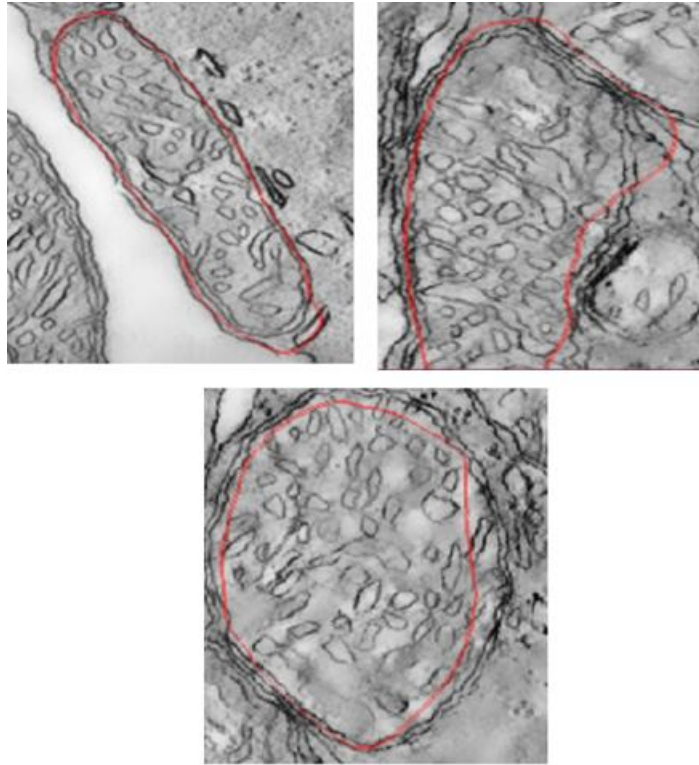
Figure 20: Examples of the automatic segmentation results which should be edited.

In order to be able to see the effect of editing on 3D construction of the mitochondria, anatomical planes are used. In order to show when the boundary of mitochondria is edited in 2D by the methods, how these changes affect the 3D construction of it.

A plane is a 2D slice through 3D space, and it helps us to have a precise view of the inside of 3D constructions. These planes can be considered as different hypothetical geometric planes used to divide the human body and also can be used to describe the locations of structures in relation to the other structures or locations in the body (44). Three commonly used planes are sagittal, coronal and transverse. Considering these planes as a cross section of a plane with object body, axial plane divides the body of the object into two parts, upper and lower parts and sagittal plane does it as right and left sides, coronel does it as front and back parts. These planes are shown in figure 20.
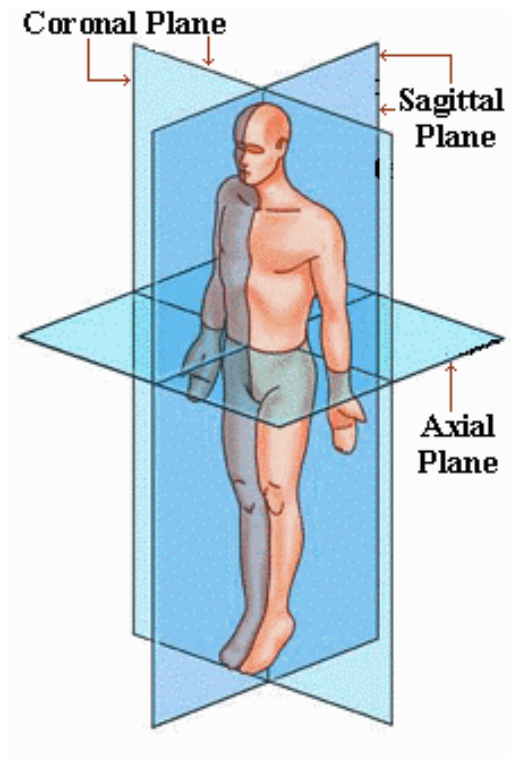
Figure 21: Anatomical planes: axial, sagittal and coronal (43).

3D construction of mitochondria was achieved by merging all 2D slices of mitochondria. It is considered as body of object. The intersection point of all three planes is considered as midpoint of all slices and was called as center point. The axial plane of mitochondria was achieved by the horizontal cross section of the object of body with a plane. The sagittal view was achieved by cross section of all slices with a vertical plane. Finally, the coronal plane was provided by intersection of all slices with plane which is perpendicular to axial plane and vertical plane. The provided 3D view for data set "6_22.sub" is illustrated in figure (22).

2.6. User Interface
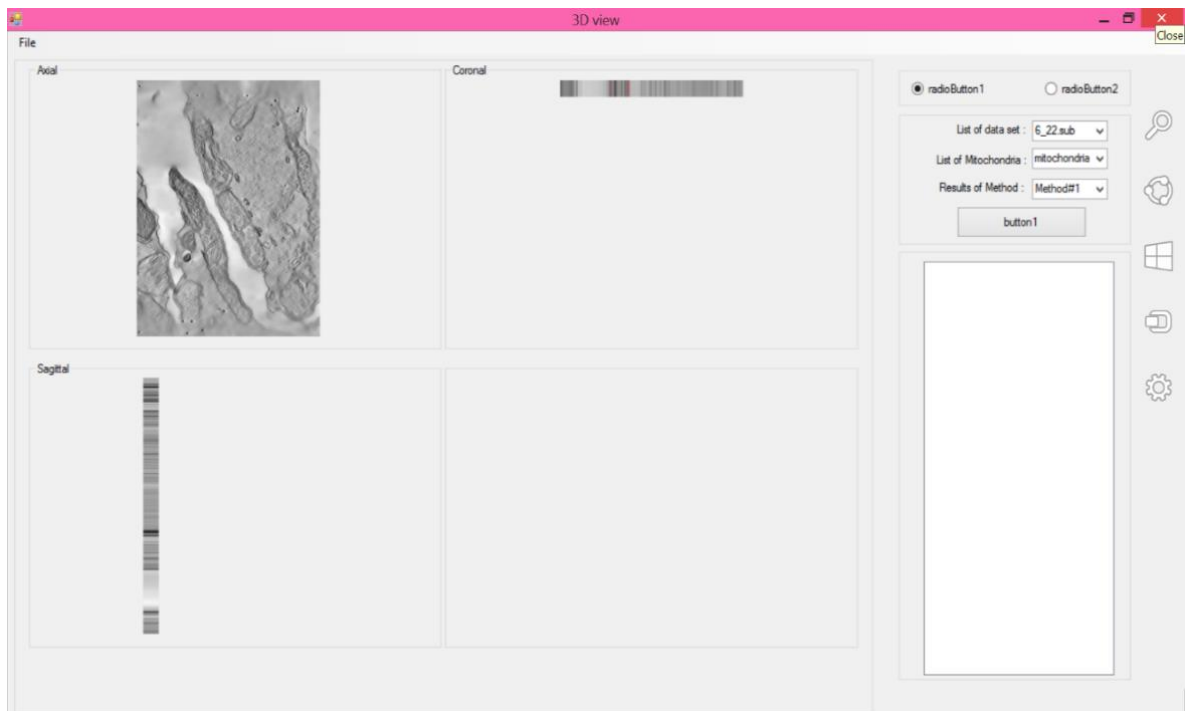
Figure 22: 3D view of data set "6_22.sub".

As it was shown in the figure 22, using these planes we tried to track the effect of editing on the 3D view of the mitochondria and to visualize the changes in three planes. Since the number of slices comparing with the size of each slice was less, the information provided by 3D view is not more useful for non-specialist user to gain information.

In order to track through all slices and get information about the editing results, a 2D view of the result of automatic segmentation (input for my methods), result of editing, and also ground truth images were provided so by mouse wheel movements the data in three of the views are changed in a way that all three view shows the same slice but with three mentioned methods results. This 2D view is shown in the figure (23).
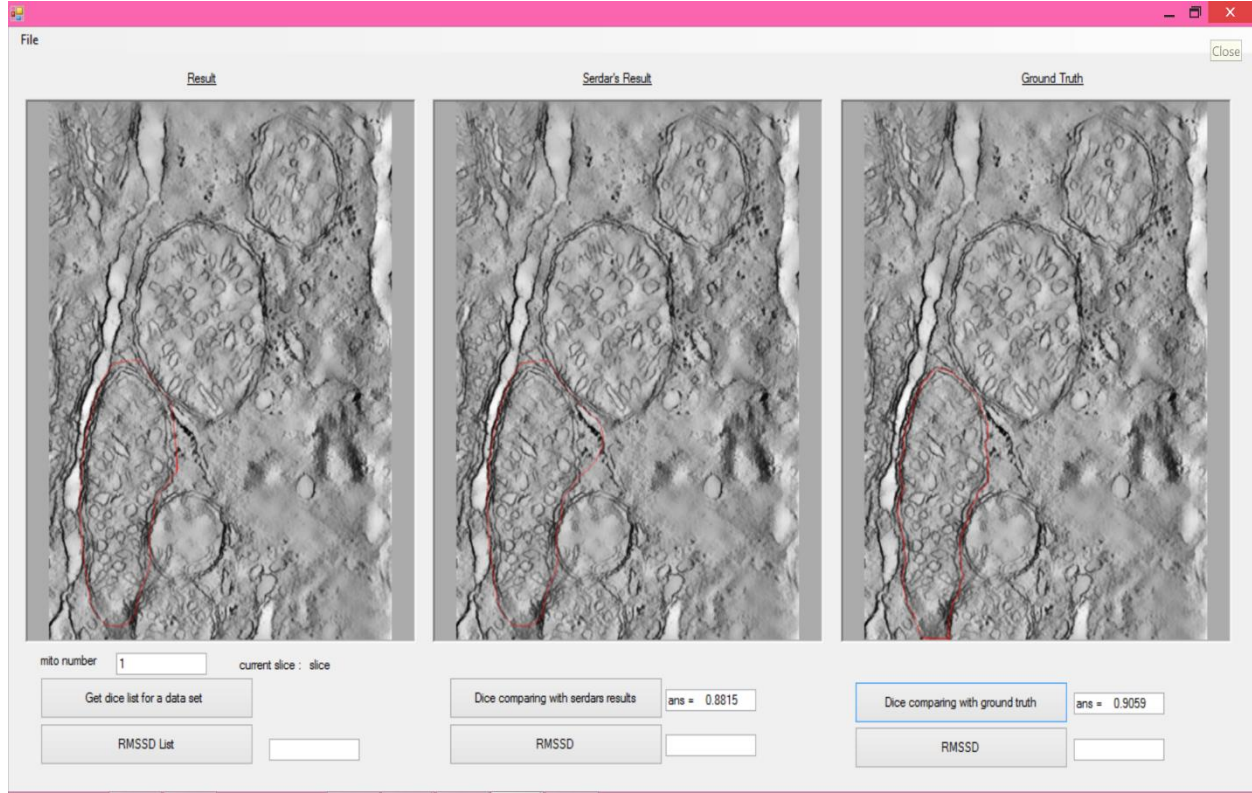
Figure 23: 2D view to compare the results. The right panel contains the ground truth images. In the middle panel, the result of Tasel's method is provided and in the left panel, the result of editing with dragging splines is shown.

2.7. Evaluation Metrics

In order to evaluate the results of the methods, three common metrics for evaluating medical images segmentation results are calculated: Dice coefficient (similarity metric), Root Mean Square Symmetric Surface Distance (RMSSSD), and t-test. These metrics are explained in the subsequent sections. The reason why these metrics were chosen is also explained.

2.7.1. Dice Coefficient

Dice coefficient (45), which is additionally called the overlap index, is the most utilized metric in approving medical volume divisions. Dice is used as a measure of the reproducibility as a statistical validation of manual annotation then the combine insightful cover of the rehashed division is estimated using the Dice (figure 24). The formula is defined by
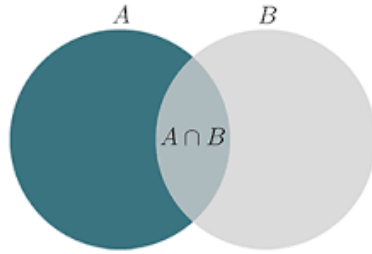
Figure 24: Illustration for calculating Dice coefficient (44).

$$\text{Dice} = \frac{2 \mid A \cap B \mid}{|A| + |B|} \tag{9}$$

In this study I tried to calculate the volumetric Dice coefficient for each mitochondrion in all data sets. Considering the Dice coefficient of each mitochondrion in all slices the volumetric Dice coefficient of a mitochondrion in data set with N slices can be achieved by:

Average Dice coefficient for each mitochondria is:

$$\text{Average \_Dice} = \frac{1}{N} \sum_{i=0}^{N} D_i \tag{11}$$

where i=0, 1 …., N and N is max slice number and D_i is 2D Dice coefficient for ith mitochondrion.

2.7.2. Root-Mean-Square Symmetric Surface Distance

The root-mean-square symmetric surface distance (RMSSSD) is utilized as often as possible to gauge the contrasts between two surfaces (made up of points). It is defined as average squared error distance between a set of real values and paired observed value(45).

In order to calculate the RMSSSD for these data sets, the results of both spline method and live-wire method were compared with ground truth.

The corresponding points on ground truth images and obtained border points are not determined. Therefore, for all pixels of the first border, the closest pixel from other border and vice versa should be obtained using Euclidean distance and at the end, the distances are squared and then averaged.

$$\text{RMSSSD}(\widehat{G}, \widehat{S}) = \sqrt{\frac{1}{n} \left(\sum (\{d|d = \min_j \left\| \widehat{S}(i) - \widehat{G}(j) \right\|, i = 1,2, \ldots, N\} \right.}$$

$$\left. \cup \{d|d = min_i \|\widehat{S}(i) - \widehat{G}(j)\|, j = 1,2, \ldots, M\})^2 \right)$$

(12)

Where $\widehat{S}$ and $\widehat{G}$ are segmented contour and ground truth contour's boundary points, and n is total number of points on segmented contour and ground truth image, and N , M are number of points on the contours, respectively.

In some cases, there is only a small disparity between the result and ground truth image. Since the ratio of overlap part to non-overlap part is more, measuring the difference between them with Dice coefficient does not reveal this disparity. In figure (25), disparity between results of automatic method and results of editing tools are shown.
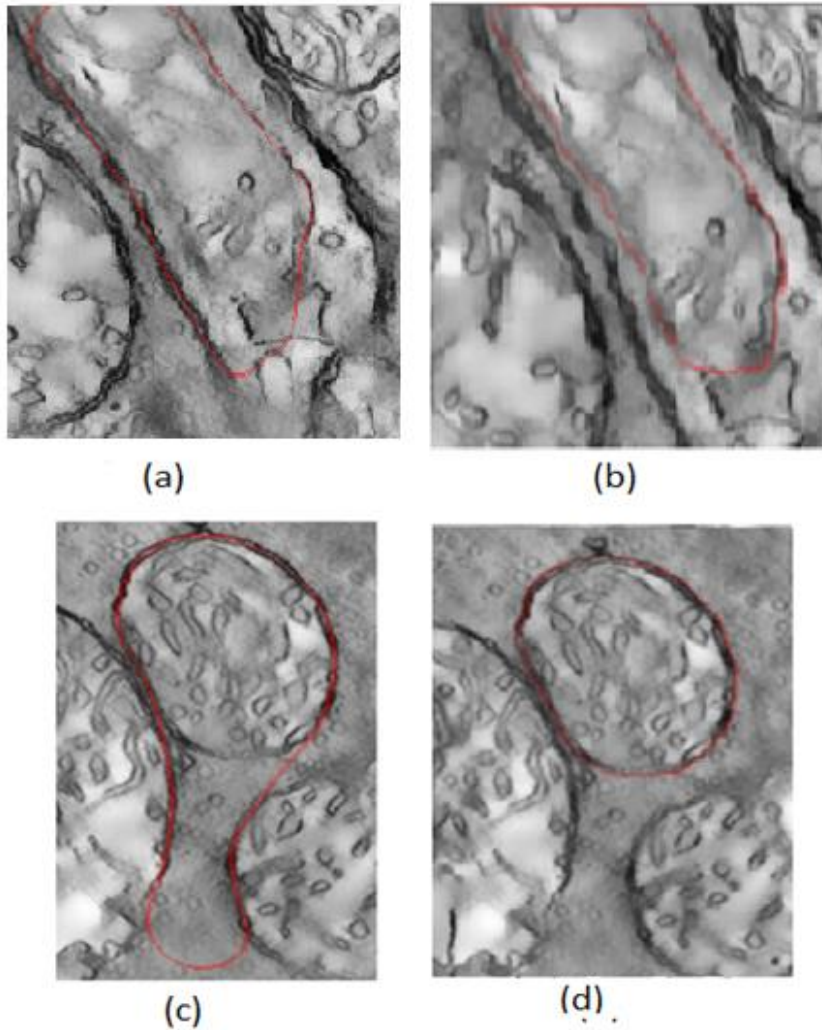
Figure 25: Disparity between results of automatic method and results of editing tools: a), c) results of automatic method, b), and d) results of 2D live-wire.

2.7.3. T_test

Actually, t-test assesses whether the means of two groups are statistically different from each other or not. While performing a t-test, it is typically attempted to discover confirmation of a critical contrast between populace implies (2-test t) or between the populace mean and a hypothesized value (1-test t)information is the ascertained contrast in units of standard error (46). The more prominent the size of T (it can be either positive or negative), the more prominent the confirmation against the invalid speculation that there is no significant contrast among them. If the magnitude of T is closer to 0, it implies it is more probable there isn't a huge distinction between two example gatherings. (See figures (26) and (27)).
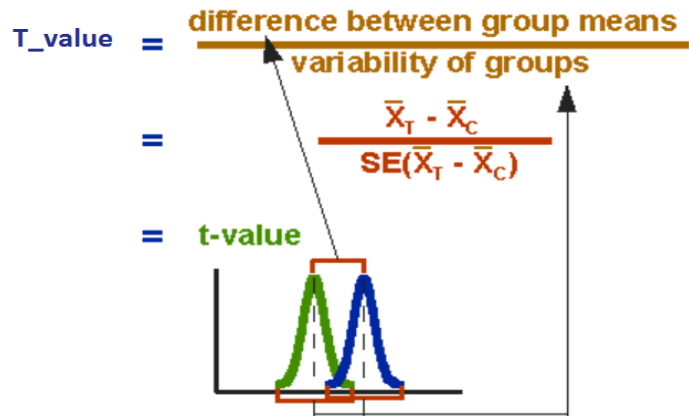
Figure 26: Definition of T-test (46).
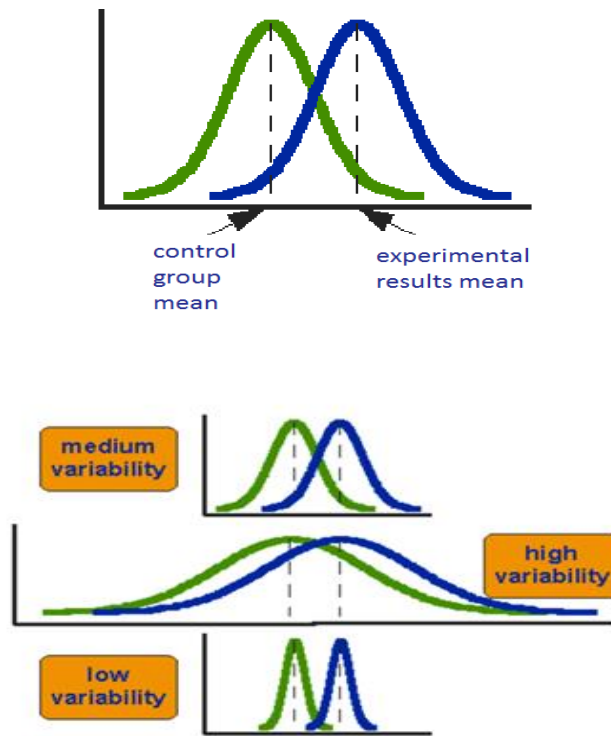
Figure 27: Three scenarios for differences between means (46).

CHAPTER 3

RESULTS

The results of these editing tools were analyzed and compared with the automatic segmentation (12) and also Alamdari's semi-automatic segmentation (5) results. The quantitative results of analyzing the results by the mentioned metrics are given in the following section.

3.1. Quantitative Results

Two mentioned metrics, Dice coefficient and RMSSSD are applied on all available mitochondria. Table (2) demonstrates an examination between the results of 2D and 3D methods in which the surface dragging method was applied.

Evaluating the Dice similarity between obtained results and also the ground truth images, it was tried to find out that to what extend theses boundaries are similar. If the two boundaries are similar, the common portion of their inside region is approximately near to the sum of their region and leads to have Dice similarity near to one while two different boundaries lead to Dice similarity near to zero. A sample result of surface dragging tool, the result of automatic method, and corresponding ground truth image is shown in Figure 27. In every data set there are several mitochondria in each slice. In order to evaluate the methods, according to the shape complexity of the mitochondria, and also because of the position of the mitochondria on the image, three mitochondria from each data set were chosen to show the results. 3D RMSSSD values refer to the RMSSSD metric for the result of applying each method in 3D. In all tables '#mitochondria' refers to the mitochondria number (id) (figure 28). In tables, 3D Dice coefficient and 3D RMSSSD refer to the result of applying the methods in 3D. It does not mean to calculate the metrics in 3D because there are no significant changes in Z direction.
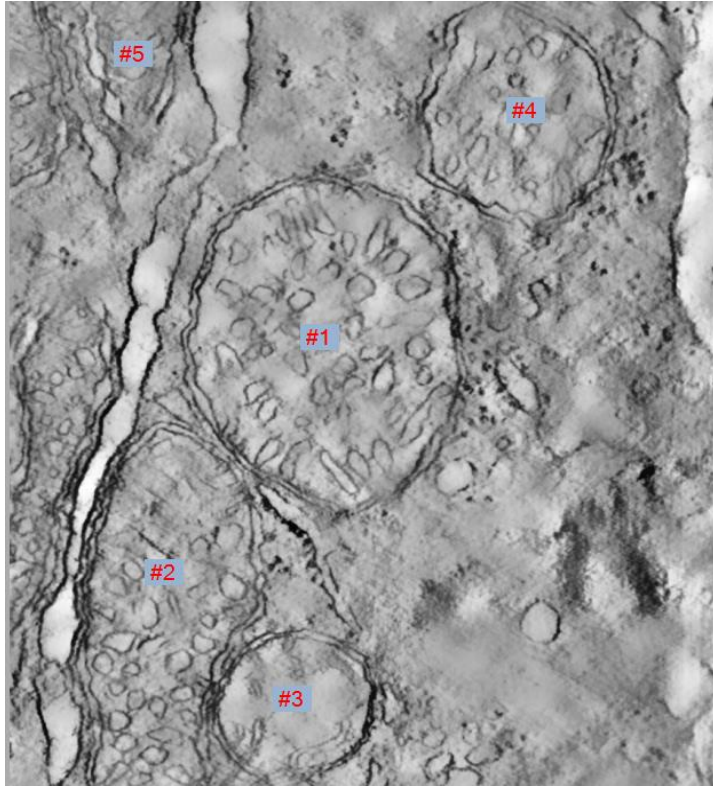
Figure 28: Sample slice from cone.sub dataset with the #mitochondria number

Table 2: The results of surface dragging in both 2D and 3D for each dataset.

| Dataset name | mitochondrion number | Before editing (Auto segmentation by Tasel) | | Dice coefficient | | RMSSSD (in units of pixels) | |
		Dice coefficient	RMSSSD	2D	3D	2D	3D
6_22.sub	#1	0.84	10	0.941	0.9186	0.65	0.72
	#13	0.89	7	0.9467	0.9327	0.58	0.85
	#6	0.91	15	0.9488	0.9326	0.48	0.53
bcpl-d.sub	#1	0.87	19	0.9285	0.9153	0.89	1.07
	#4	0.82	8	0.9121	0.9242	0.78	1.1
	#5	0.87	14	0.9217	0.9205	0.68	1.13
Cone.sub	#1	0.83	21	0.9404	0.9352	0.65	1.02
	#2	0.9	61	0.9397	0.9344	0.43	0.84
	#3	0.81	13	0.9043	0.9075	0.72	1.05
Gap18_sub	#1	0.86	22	0.9488	0.9379	0.43	0.81
	#2	0.83	19	0.9425	0.9205	1.01	0.74
	#3	0.79	17	0.9352	0.9177	0.98	0.67
mac_serial_sub	#7	0.85	9	0.9321	0.9201	0.87	1.03
	#12	0.81	23	0.9402	0.9106	0.34	1.01
	#11	0.79	7	0.9352	0.9177	1.01	0.85
Od.sub	#3	0.79	11	0.9324	0.9069	0.72	1.02
	#5	0.81	15	0.9215	0.9131	0.93	1.16
	#6	0.78	44	0.9149	0.9248	0.75	1.14
pedicle	#2	0.73	25	0.9321	0.9225	0.57	1.04
	#3	0.83	12	0.9306	0.9123	0.48	1.1
	#5	0.87	101	0.9298	0.9068	0.45	1.06
Average	-	-	-	0.93	0.92	0.69	0.93

In the experiments, the default value for the radius of ROI is set to 100. In figure (29) ROI with different values are shown. In some cases, applying method with small value of radius requires to apply the method more than once to achieve the desired result while setting it to a large value lead to unsatisfied results.
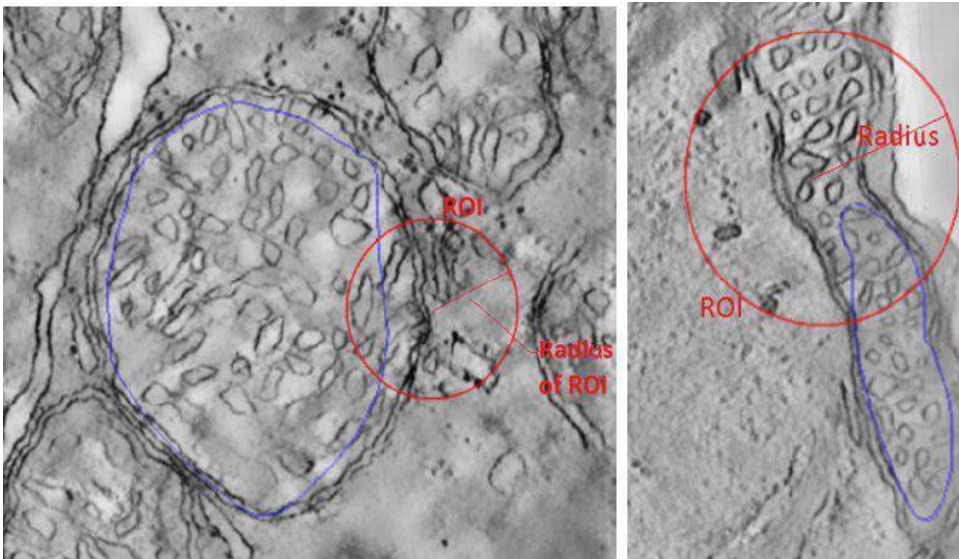
Figure 29: R=100, R=200

User can easily change the radius of ROI to the proper value according to the part of the boundary which should be refined and corrected.

Looking more closely, Table (2), it is obvious that applying surface dragging method in both 2D and 3D leads to acceptable results. Dice coefficient metric which is approximately 1.0. The edited boundaries are almost the same as the ground truth images since this method is unrelated to the texture of the image and by selecting the proper part to correct, the desired result is achievable.

The RMSSSD metric for the mentioned method resulted to values are nearly 0.0 which reveals that the slipping value for the resulted boundaries are nearly zero and so the resulted boundaries are the same as the ground truth images. In Table (3) the results of Dice coefficient and RMSSSD metrics are shown for the interactive live-wire segmentation tool.

Table 3: The results of live-wire method in both 2D and 3D for each dataset.

Dataset name	mitochondrion number	Before editing (Auto segmentation by Tasel)		Dice coefficient		RMSSSD (in units of pixels)	
		Dice coefficient	RMSSSD	2D	3D	2D	3D
6_22.sub	#1	0.84	10	0.9596	0.924	0.32	1.99
	#13	0.89	7	0.9607	0.9129	0.68	2.13
	#6	0.91	15	0.9624	0.9273	0.51	1.96
bcpl-d.sub	#1	0.87	19	0.9231	0.9023	0.73	2.31
	#4	0.82	8	0.924	0.9133	0.68	2.1
	#5	0.87	14	0.9236	0.9032	0.74	2.27
Cone.sub	#1	0.83	21	0.9375	0.9369	0.48	1.91
	#2	0.9	61	0.9452	0.9421	0.23	1.8
	#3	0.81	13	0.9276	0.9001	0.67	2.25
Gap18_sub	#1	0.86	22	0.9493	0.9274	0.58	1.94
	#2	0.83	19	0.9559	0.9261	0.9	1.95
	#3	0.79	17	0.9581	0.9025	0.63	2.3
mac_serial_sub	#7	0.85	9	0.9402	0.9015	0.85	2.32
	#12	0.81	23	0.9468	0.9203	0.23	2.01
	#11	0.79	7	0.958	0.9025	0.89	2.29
Od.sub	#3	0.79	11	0.9423	0.9148	0.68	2.07
	#5	0.81	15	0.9316	0.9042	0.82	2.26
	#6	0.78	44	0.9345	0.9129	0.64	2.15
pedicle	#2	0.73	25	0.9412	0.9082	0.51	2.24
	#3	0.83	12	0.9395	0.9103	0.43	2.2
	#5	0.87	101	0.9436	0.9051	0.37	2.05
Average	-	-	-	0.94	0.91	0.6	2.11

Comparing the results of 3D live-wire on the all data set with the results of 3D surface dragging showed that the former has inferior performance since this method depends on the texture characteristics of the image. On the other hand, these images contain a lot of noise and edge like details. The Dice coefficient for these results demonstrated that the distinction between the results and ground truth images are far from those of dragged spline

methods. The RMSSSD value for the results of 3D live-wire show that there was disparity between these results and the ground truth images (figure 30).
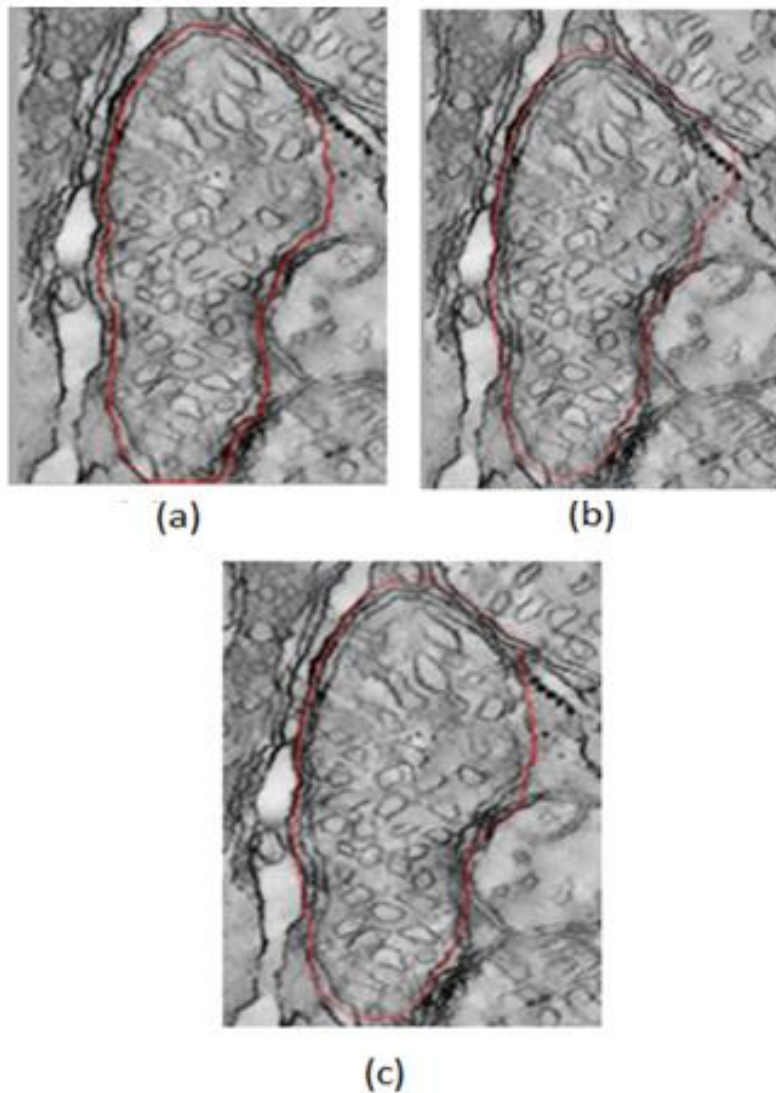
Figure 30: a) Ground truth image b) Result of automatic segmentation c) Result of editing the output of automatic segmentation using 2D live-wire.

The RMSSSD value for 2D implementation of live –wire and surface dragging method reveal the superiority of the 2D implementation compared to 3D. In 3D implementation ROI is considered as intersection of slice with VOI. In some cases, that the mitochondria get smaller by moving through the slices, the VOI have no intersection with the boundary or only small part of boundary was edited. It is illustrated in figure (31).

44
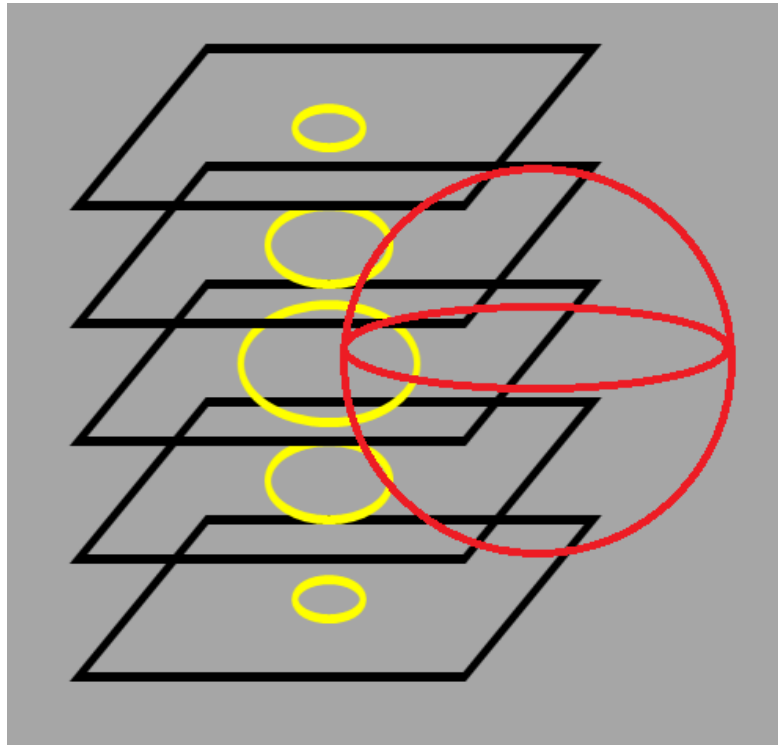
Figure 31 : Intersection of mitochondria by VOI (5)

As it is shown in figure (31), while applying methods in 3D, when the mitochondria gets smaller in some slices there is no intersection between VOI and the boundary. Therefore, maybe only a small part of boundary has intersection. In this situation the results of 2D is better than 3D. Because in 2D the user selects the ROI and consequently, this leads to better result. A sample mitochondrion which gets smaller in the different slices is shown in figure (32).
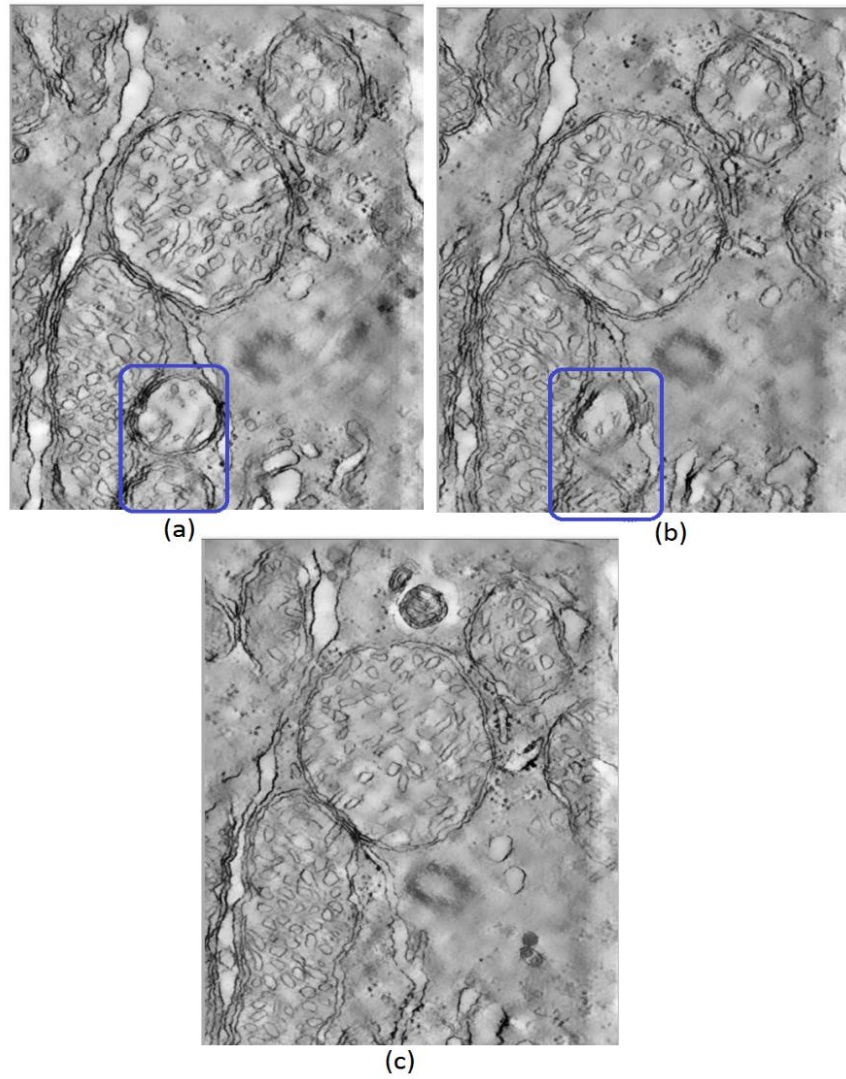
Figure 32: The marked mitochondria in (a) gets smaller moving through slices (b), this mitochondrion vanishes slowly in the remaining slices.
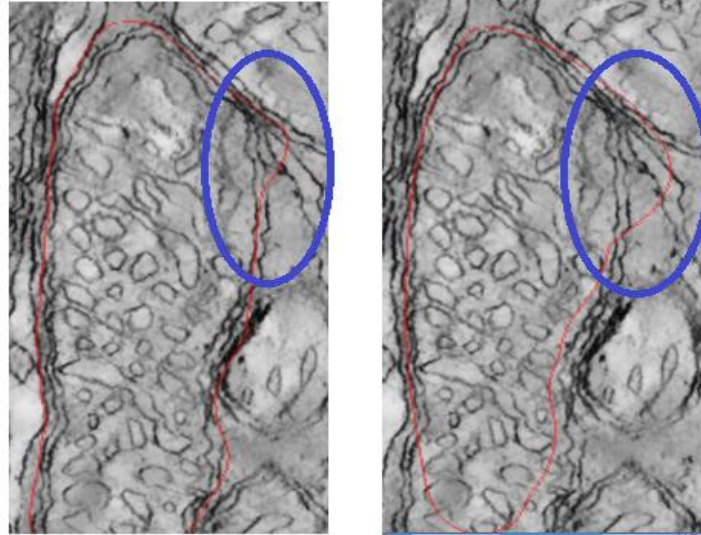
Figure 33: A sample mitochondria from cone dataset.

In cases like figure (33), part of the boundary which we try to edit varies while moving among the slices. This figure is another example to show that why the result of 2D spline method is better than 3D one because of the dragged control point. The barcharts for Dice coefient and RMSSSD and also Error bars are shown in the following figures (34),(35),(36),(37),(38) and (39).
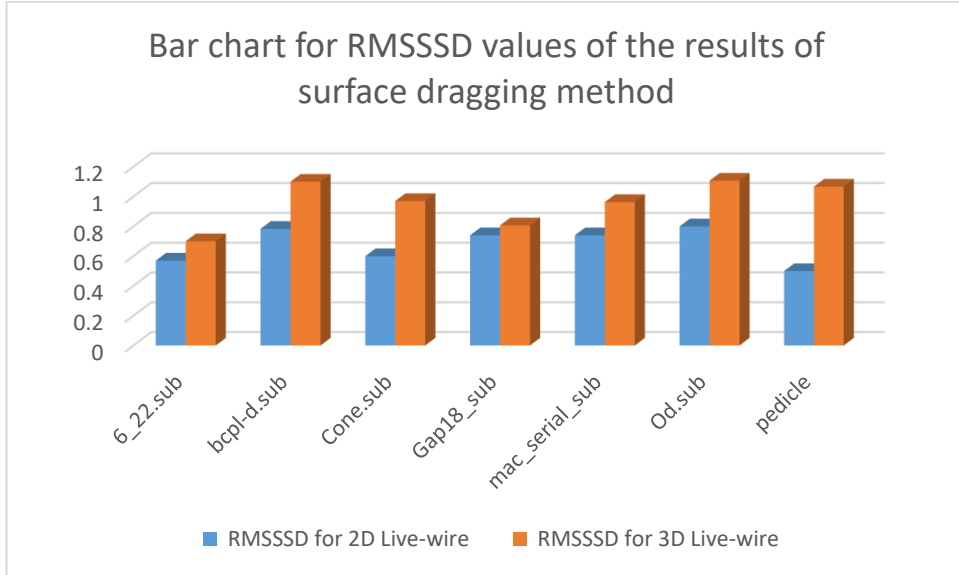
Figure 34: Bar chart for RMSSSD values of the results of surface dragging method, in units of pixels.
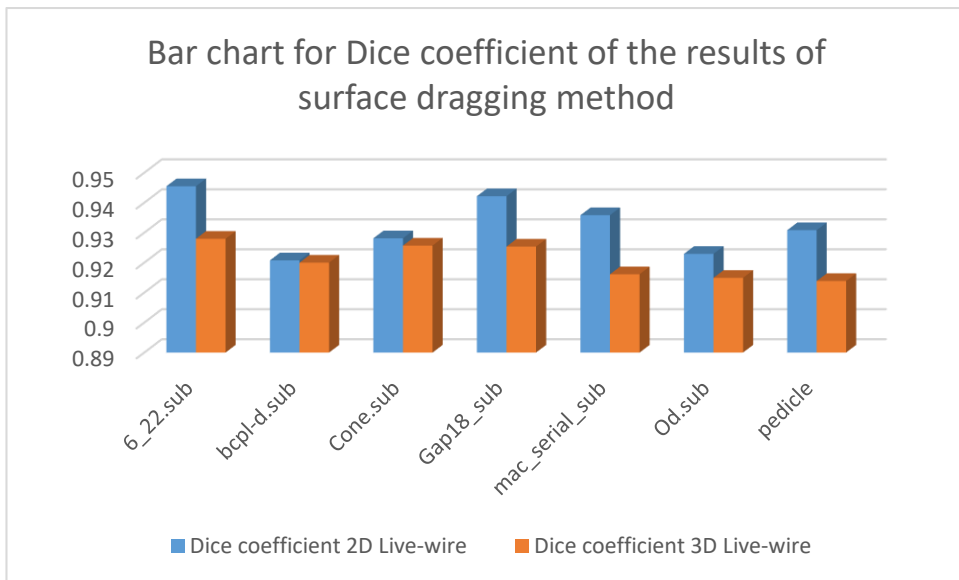
Figure 35: Bar chart for Dice coefficient of the results of surface dragging method.
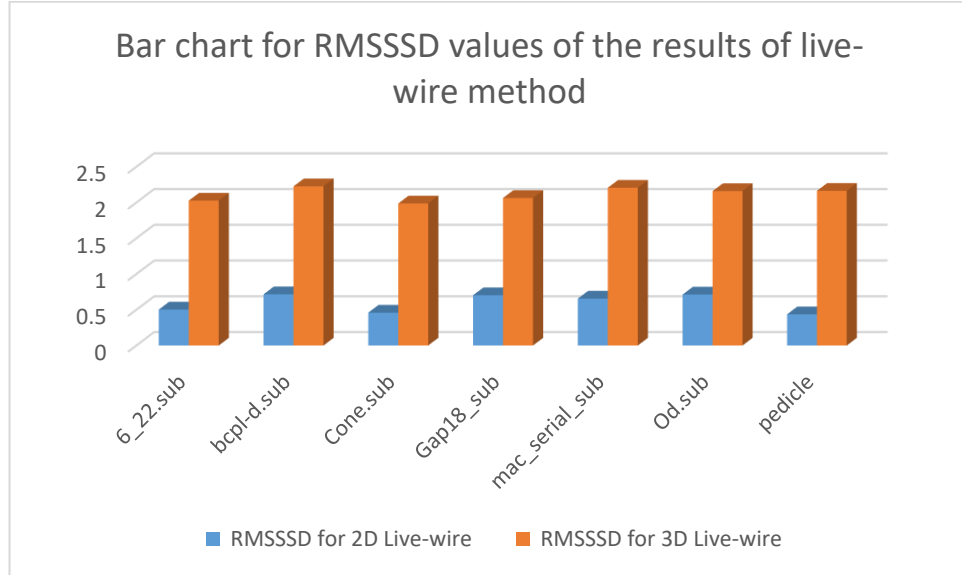
Figure 36: Bar chart for RMSSSD values of the results of live-wire method, in units of pixels.
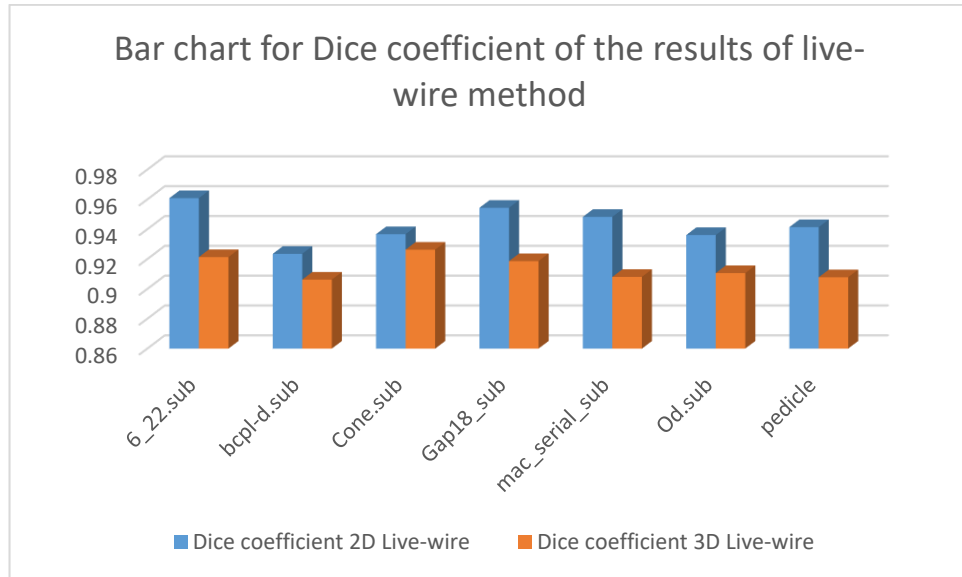
Figure 37: Bar chart for Dice coefficient of the results of live-wire method.
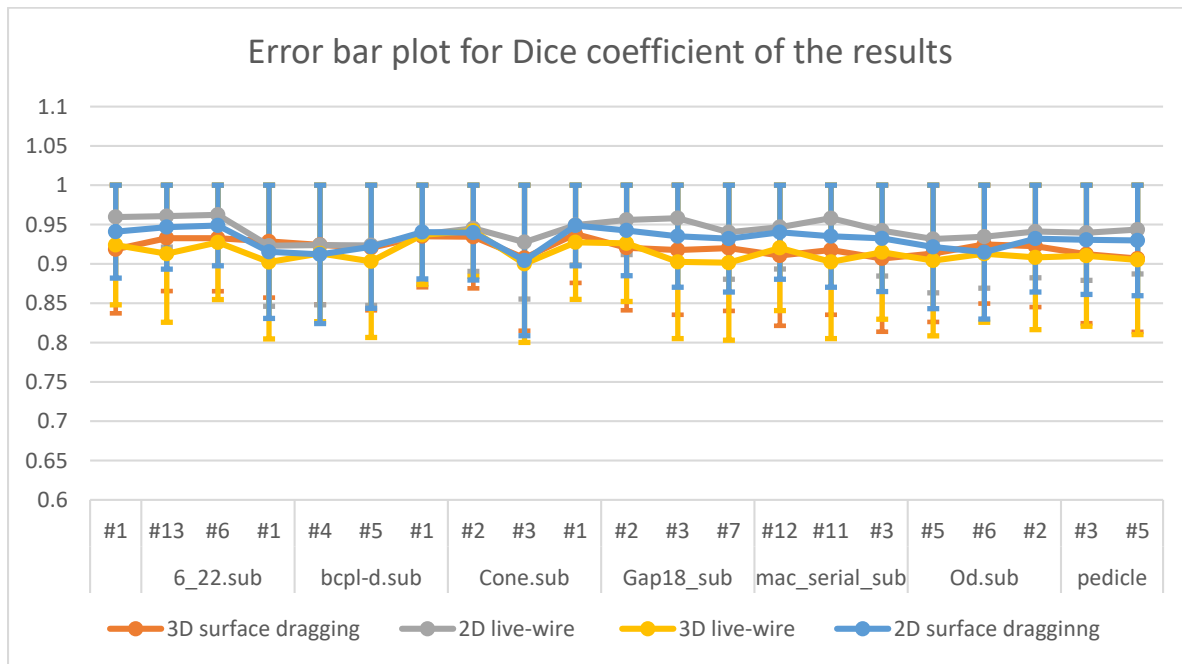
Figure 38: Error bar plot of Dice coefficient of the results.
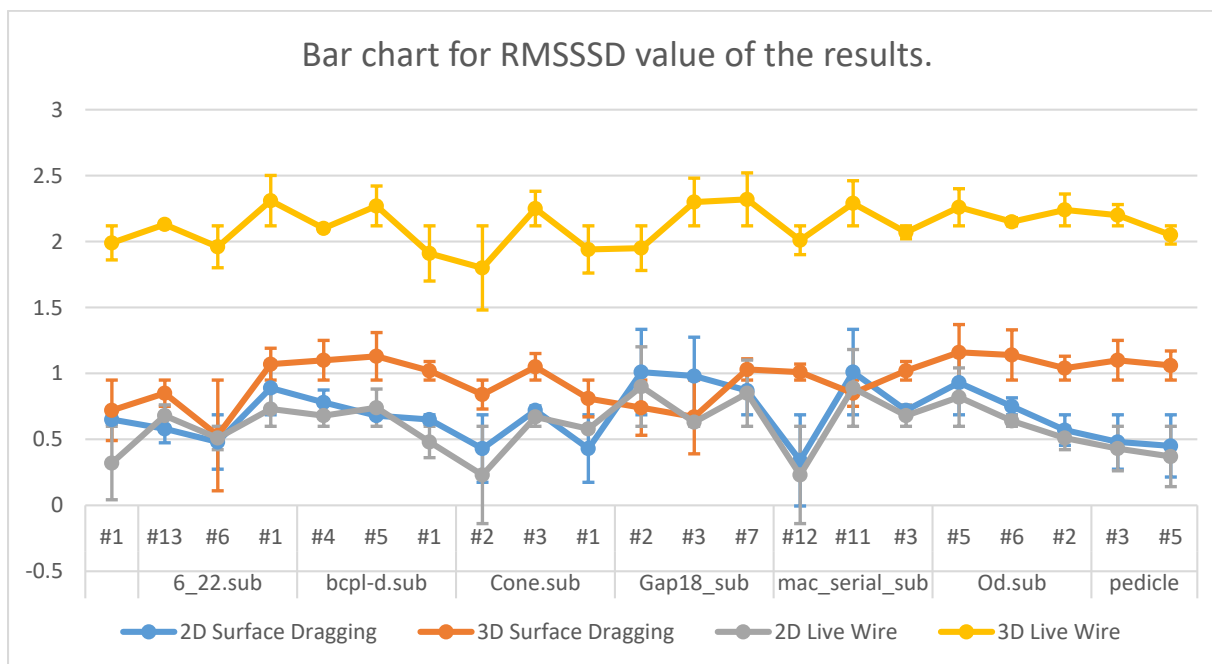
Figure 39: Bar chart for RMSSSD value of the results.

From the results it is clear that the live-wire 2D method has the least error and therefor the most similarity to the ground truth images. In contrast, in the most of the cases 3D live-wire method has the worst results. However, it is not general and we have some exceptions that the 3D surface dragging method shows worse results as well. The errors are the difference between the results of the used methods and the ground truth images. In some cases, all four methods converge together meaning that all used methods have similar results. However, in some datasets, the used methods show scatter results.

In each dataset, the mitochondria which do not have considerable changes through the slices both 2D and 3D methods led to good results and the evaluation metrics showed better results. In the contrary, the mitochondria which has considerable changes through the slices the 3D dragged spline and also 3D live-wire methods led to worse results and consequently the Dice coefficient has worse results than 2D methods. (Figures (40) and (41)).
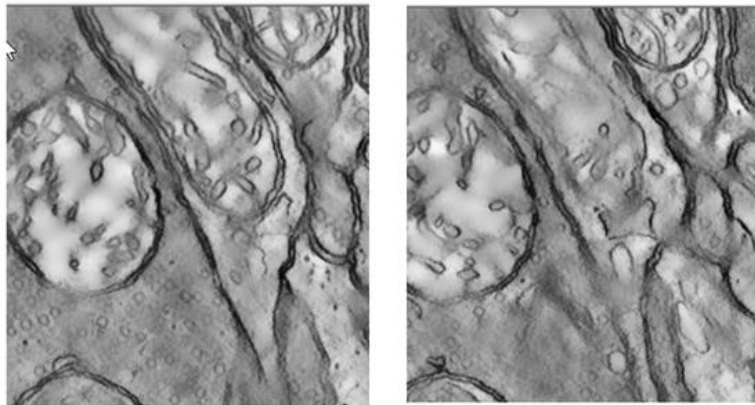
Figure 40: sample mitochondria in pedicle dataset in different slices.
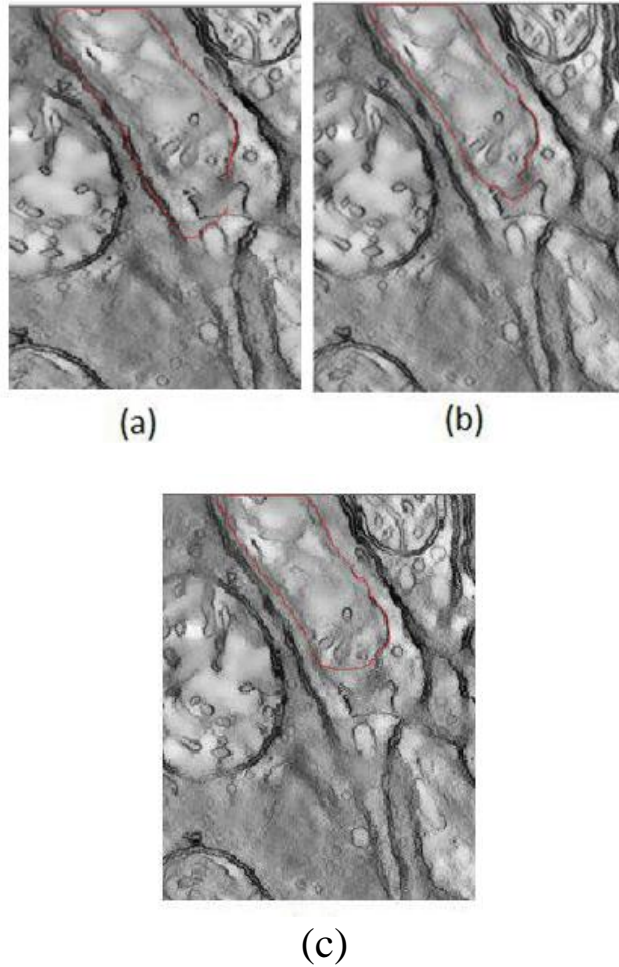
Figure 41: a) result of 2D live-wire b) result of 3D live-wire c) 2D surface dragging

As it is shown in figure (40), some mitochondria get faded in some slices. Applying live-wire method in 3D, live-wire tool snaps to parts with high edge characteristics and lead to worse result than 2D live-wire. The result of 2D and 3D live-wire tool are shown in figure (41). As t mentioned the live-wire tool, snap to strong edges between seed points.

T-test is applied on the results considering hypothesis null as there is high differences between the segmentation results of current correction method and the ground truth images, and also hypothesis 1 as there is not a significant difference between two mentioned boundaries which means the current method perform well on the given data sets.

In order to apply t-test on the obtained results comparing with ground truth results, the area inside each mitochondrion in each slice for the result of this study and ground truth images were considered and statistics t-score is calculated and the results are illustrated in table (4).

High value of t shows that there is huge distinction between current outcomes and ground truth images while the nearer T is to zero, the more probable there isn't a huge contrast difference between two groups of data.

The T-test results applied to the results of 2D and3D Surface Dragging and live-wire methods.

Table 4: The t-test results for 2D and 3D surface dragging and live-wire

	2D Surface Dragging	3D Surface Dragging	2D Live-Wire	3D Live-Wire
t-value	0.019	0.026	0.013	0.033

T-value for both 2D and 3D methods of live-wire and surface dragging was calculated. The values show that the results of 2D live-wire method were more likely to the ground truth images and the results of 3D live-wire had the worst similarity with ground truth images. In surface dragging method, 2D method had better results than 3D because in most cases the mitochondria had considerable changes moving among the slices.

3.1.1 Results Comparison

These editing tools' results are compared with the results of Tasel's and Alamdari's work. This information is provided in Table (5).

Table 5: The comparison of the Dice coefficinet results with those of Tasel's and Alamdari's.

| Dataset | Tasel's Dice Coefficient (Merged) | Alamdari's | | | 2D Surface Dragging | 2D Live-Wire | 3D Surface Dragging | 3D Live-Wire |
		Active Contour Method	Active Contour and Kalman Filter	Kalman Filter with Active Contour and Optical Flow				
bclpb-d	0.89	0.88	0.88	0.88	0.93	0.92	0.92	0.90
Gap18_sub	0.87	0.87	0.87	0.86	0.94	0.92	0.93	0.91
Pedicle	0.84	0.94	0.94	0.94	0.95	0.93	0.94	0.91
Spherule24 mos1*	0.97	0.96	0.96	0.96	-	-	-	-
Average	**0.89**	**0.91**	**0.91**	**0.91**	**0.94**	**0.92**	**0.93**	**0.91**

Comparing the results of these three mentioned works shows that the Alamdari and Tasel's methods led to nearly the same results but only in 'Pedicle' data set Alamdari's method had better results. Applying the editing tools provided in this study, on these results, led to better results with the Dice coefficient. Comparing 3D surface dragging method and 3D live-wire method, surface dragging had better results because in live-wire method the noise and edge like details caused the live-wire to snap to pixels with high edge characteristics and misdirected it. It should be mentioned that Alamdari's results are taken from her thesis and they may not refer to the same mitochondria on which the current editing tools were done.

In Spherule24mos1 dataset, the Tasel's method could detect almost the exact mitochondrion and so this dataset did not edit. The result of automatic segmentation method is shown in figure 42.

Figure 42: Mitochondria detected by Tasel's automatic method in spherule24mos1 (11).

The average Dice coefficient for the work done by Alamdari, for three of the used methods, active contour, Kalman filter, and optical flow were calculated as 0.90. Editing these results by surface dragging tool and live-wire interactive editing tool improved them and led to 0.94 and 0.92 values for Dice coefficient, respectively.

3.1.2 Method Run Times and User Interaction Times

Two female users and one male user from computer engineering department and one female user from statistic department applied these tools on the given dataset. The ages of users were 27,29,26,31 respectively. The technical characteristics of the computers used by each user are provided in the Table (6).

Table 6: The technical characteristics of used computers by users

User	RAM	CPU	Brand
User#1	4GB	Intel core i7	Lenovo
User#2	4GB	Intel core i5	Del
User#3	4GB	Intel core i3	Asus
User#4	4GB	Intel core i5	HP

After that all users got familiar with the tools, using chronometer, at each step the required time was recorded by the users. The mean time required for each mitochondrion in per dataset is provided in Table (7) according to each method.

Table 7: Mean run time per mitochondria for each dataset to edit using spline method and live-wire method in 2D and 3D.

Data set	#mitoch ondrion	Mean run time ± std Dragged Spline (units of minutes)		Mean run time ± std Live-wire (units of minutes)	
		2D	3D	2D	3D
6_22.sub	#1	42.47 ± 2.32	1.25 ± 0.08	57.63 ± 2.52	1.90 ± 0.10
	#13	63.70 ± 3.10	1.58 ± 0.12	78.87 ± 3.85	2.60 ± 0.13
	#6	84.93 ± 4.07	1.92 ± 0.17	100.10 ± 4.97	3.30 ± 0.17
Bcpl-d.sub	#1	35.58 ± 1.65	1.42 ± 0.10	45.75 ± 2.12	2.25 ± 0.10
	#4	21.35 ± 1.18	1.08 ± 0.08	31.52 ± 1.63	1.55 ± 0.07
	#5	49.82 ± 2.17	1.75 ± 0.10	59.98 ± 2.85	2.95 ± 0.17
Cone.sub	#1	33.95 ± 1.73	1.08 ± 0.07	50.12 ± 2.70	1.55 ± 0.08
	#2	79.22 ± 3.90	1.75 ± 0.13	95.38 ± 4.48	2.95 ± 0.15
	#3	113.17 ± 5.40	2.25 ± 0.25	129.33 ± 5.73	4.00 ± 0.22
Gap18_sub	#1	50.40 ± 2.35	1.92 ± 0.23	59.40 ± 3.07	3.30 ± 0.17
	#2	18.90 ± 1.10	1.08 ± 0.07	27.90 ± 1.48	1.55 ± 0.07
	#3	25.20 ± 1.25	1.25 ± 0.08	34.20 ± 1.95	1.90 ± 0.10
Mac_serial_sub	#7	38.85 ± 1.85	1.08 ± 0.10	57.35 ± 2.70	1.55 ± 0.08
	#12	77.70 ± 3.55	1.58 ± 0.12	96.20 ± 4.48	2.60 ± 0.00
	#11	64.75 ± 2.97	1.42 ± 0.10	83.25 ± 4.25	2.25 ± 0.13
Od.sub	#3	31.85 ± 1.63	1.08 ± 0.07	47.02 ± 2.43	1.55 ± 0.07
	#5	53.08 ± 2.93	1.42 ± 0.13	68.25 ± 3.25	2.25 ± 0.12
	#11	84.93 ± 4.37	1.92 ± 0.32	100.10 ± 4.80	3.30 ± 0.18
Pedicle	#2	10.85 ± 0.62	1.08 ± 0.15	16.02 ± 0.77	1.55 ± 0.08
	#3	18.08 ± 0.95	1.42 ± 0.08	23.25 ± 1.25	2.25 ± 0.13
	#5	3.62 ± 0.32	0.75 ± 0.05	8.78 ± 0.55	0.85 ± 0.05
Average		47.73 ± 2.35	1.43 ± 0.12	60.50 ± 2.94	2.28 ± 0.11

Time required to apply these methods in order to obtain the desired boundary, varies according to the method and also according to what extend the user was familiar to the application. Different users have tried these two methods in order to apply on the given data set. Applying methods on 2D, user has to apply the method on each data set and some

data sets involves more than 100 slices and also some mitochondria need more than one step of editing. So applying for example 3 steps of editing on Mac_serial_sub with 111 slices takes at least one and half hour. Totally spline method is faster than live-wire method because in live-wire method the editing in done on ROI with radius 100. This lead to apply one edit in two or three steps instead of one step which led to increase the required run time.

Actually, in spline tool, run time includes opening the first slice, selecting ROI, moving the dragged center until the result is satisfying, fixing and finally saving the result. Then the user can move to the next slice by scrolling the mouse. The number of required edits for each mitochondria varies and also each data set has different number of slices. Some mitochondria only need one editing step while some others need more than 3 editing steps which significantly affects the run time. In live-wire method, run time involves setting ROI, setting seed, moving mouse on the image until the result is satisfying, and finally saving the result. In live-wire method since the code is in primary stage of development, editing is done in 100* 100 patches which takes more time compared with the spline method. Because the user has to work on multiple ROI in order to correct part of a mitochondria.

Applying editing tool on 2D takes a lot of time to complete because the user has to apply edits in all slices. In 3D mode, the user does the edits only on the middle slice and then the editing was done automatically on the all other slices. Comparing 2D and 3D methods' run time, it is clear that 3D method in both dragged spline and live-wire take less time than 2D which needs to apply the editing tool on each slice manually.

It is necessary to compare the editing time required (of the tools applied here in this thesis) with the San Diego group's hand segmentation times. According to the San Diego group, when only hand segmentation is used (as in ground truth segmentation), the time to segment the outer mitochondrial membrane (OMM) depends in part on the size of the mitochondrion and how curvy the membrane is. For all the datasets except for the Mac_serial_sub, the amount of time to trace one contour of the OMM in one slice is approximately 65 seconds on average. For the Mac_serial_sub, it takes approximately 55 seconds.

In 2D surface dragging method, in order to edit '6_22', 'bcpl-d.sub', 'Cone.sub', 'Gap18_sub', 'Od.sub' and 'pedicle' datasets, it takes 42, 10, 46,35,33,38 seconds to edit one mitochondrion in one slice respectively. While in hand segmentation this time is about 65 seconds for a mitochondrion in each slice in each data set.
In order to edit one mitochondrion in each slice in '6_22', 'bcpl-d.sub', 'Cone.sub', 'Gap18_sub', 'Od.sub' and 'pedicle' datasets using 2D live-wire method it takes approximately 52,45,57,45,42,48 seconds , respectively. While in hand-segmentation it takes 65 seconds to trace the mitochondrion in each slice. In general, 2D methods takes less time than hand segmentation method.

In 3D methods, the user has to do the editing in one slice and the editing on the other slices is done automatically. In 3D methods, the required time in order to edit one mitochondrion

in all slices varies between 60 and 240 seconds. While in hand segmentation this time is at least 2015 seconds for pedicle dataset with 31 slices (considering 65 seconds for a mitochondrion in one slice). Therefore, comparing the required time for 3D editing methods with that of hand-segmentation method it shows that 3D methods are faster than hand-segmentation method.

CHAPTER 4

CONCLUSION AND FUTURE WORK

The main purpose of thesis was to develop and implement editing tools to correct automatic segmentation results. For this purpose, surface dragging and live-wire segmentation methods were implemented and applied on the Tasel's results. These methods were applied on 2D and also 3D. The 3D construction of a mitochondria was gained by considering all slices together. In order to apply these methods on 3D, firstly, these methods were applied on the middle slice of the method. The region of interest considered as a sphere over whole slices with the center in the coordinates of ROI on the middle slice. The radius of ROI was set to 100 pixels. The ROI for each slice could be obtained by intersection of sphere with the current slice. While applying surface dragging, the ROI is obtained by intersection of sphere with the current slice, the intersection points of ROI with existed mitochondria boundary and also the dragged center of ROI which was the same as coordinates of dragged center in middle slice, were considered and using Bezier cubic formula, the curve which passes through these points are drawn and then desired art of the boundary was updated. In order to apply live-wire method on 3D, after setting ROI as it was done in previous method, the start node and end node which were selected on the middle slice, were carried to one of their 8 neighbor pixels which have higher gradient magnitude than other 7 neighbors. Then the path between start node and end node with minimum cumulative path cost was considered as new part of the boundary and the existed boundary was updated.

After applying these methods on the given dataset, in order to evaluate the results of methods, three metrics, Dice coefficient, RMSSSD, and t-test were calculated in both 2D and 3D. Volumetric Dice coefficient and average Dice coefficient were also calculated in order to compare the obtained results and ground truth images.

Comparing these results, it was seen that applying 2D live-wire led to better results in which the Dice coefficient were approximately one which means that the obtained results were approximately the same as ground truth images, and the RMSSSD were near to zero which means that dislocation of the points of the boundaries where the least and near to zero. But

59

editing by 2D methods on all slices, is a very time consuming process (approximately 30-times more editing time) which may outweigh the benefits of better segmentation (1-2% improvement in Dice coefficient, but better improvements in RMSSSD) as compared with the 3D methods.

If the editing time is not considered, the 2D methods led to better results than 3D of both live-wire and dragging methods (by 1-2% improvement in Dice coefficient, but better improvements in RMSSSD). 3D live-wire method showed worse segmentation results than 3D surface-dragging method since in the latter method, the method does not relate to the texture of the image, noise or edge like details.

In some cases, the edges are faded and also unclear, in some other cases such as od.sub, the mitochondria are so close to each other that while using live-wire method, since it is tried to find the optimal path with the high edge characteristics, the boundary of the coherent mitochondria can be detected as the optimal path. In order to reduce the impact of the coherent mitochondria, this method is implemented so that only the ROI part of the image instead of the whole image is considered to find the optimal path.

Considering the execution time, this time for live-wire was more than that of surface dragging method because in live-wire method each editing step was done in ROI with 100*100 pixels. So, one (rather big) editing step of surface dragging can be achieved by more than three steps of live-wire, which results in considerably more editing times as seen in Table 7.

Comparing the results of these editing tools with the results of the work done by Tasel, it was shown that these editing tools, led to better results and these results were more similar to ground truth image but the required time was higher than hand-segmentation time. Dice similarity of both results was calculated to get the similarity between the results and the ground truth images. For Tasel's results, the average Dice coefficient was 0.78 while this amount for the results of live-wire was obtained 0.94. It was clear that editing the segmentation results of Tasel's work by surface dragging editing tool, made the results better. While applying the surface dragging tool on the results of Tasel's work, increased the Dice coefficient from 0.79 to 0.93. It showed that this editing tools have improved the results of automatic segmentation method however, the required time should not be ignored.

4.1. Limitations:

In some data sets like bcpl-d.sub as it is shown in figure 43, the images have a lot of noise and the boundaries of the mitochondria not only are not sharp but also are faded. Detecting these boundaries using surface dragging is better and feasible comparing with the

interactive live-wire method in which the texture of the image affects the methods results adversely.
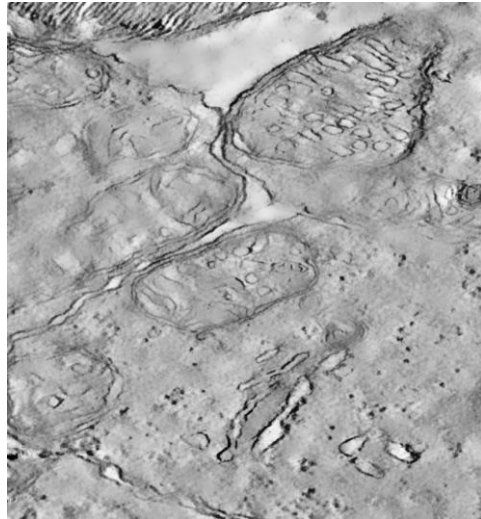

Figure 43: In bcpl-d.sub dataset, the boarders are faded and unclear.

In some other cases such as od.sub mitochondria are located near each other so that in applying interactive live-wire method, there are other mitochondria's boundaries and this affects while trying to find the optimal path between two points on the image. The part of the boundary of the nearest mitochondrion can be detected as a part of the current mitochondrion which affects the results. In order to decrease the effect of this problem, it is tried to apply the method only on the ROI so that other strong edges which are not related to the current mitochondria are ignored and are not taking into account while finding the optimal path (Figure 44).
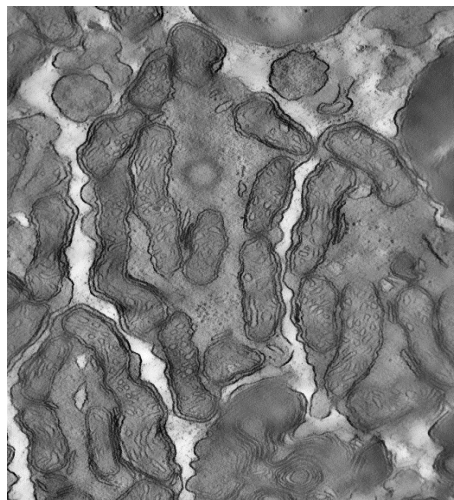

Figure 44: Od.sub dataset. There are many mitochondria in each slice (19).

4.2. Future Work

Live-wire implementation was not as efficient as it should be. Therefore, editing time does not reflect its full potential. Our tool has to be faster so that live-wire can be used in a much larger ROI so that boundary editing (of a region that covers a large area) can be done by the user in one-shot, rather than in multiple ROIs of size 100x100.

The editing tools proposed in this study may also be applied on cristae structures. Cristae (inner membrane of mitochondria) have a large amount of surface area for chemical reactions to occur. Cristae shape determines the assembly and stability of respiratory chain super complexes (RCS) and therefore the efficiency of mitochondria-dependent cell growth depends on cristae shape (47). Therefore cristae shape is studied in the literature very often(48). In many studies, the cellular components such as cristae were analyzed and segmented It should be investigated whether the editing tools proposed in this thesis could be adapted for cristae structure in the mitochondria or not.

The proposed editing tools are not the only effective semi-automatic methods in order to edit the segmentation of mitochondria. The merging and splitting tools proposed by Cocelli (17) can be combined with these tools for more accurate and fast segmentation of inner membrane and outer membrane.

In GUI interface, Visualization Tool Kit (VTK) can be used to give more information about the effect of 2D edits on 3D structure of mitochondria by using the rendering tool. Furthermore, An ITK pipeline combined with a VTK pipeline to represent the result of segmentation of liver from CTA data sets directly in 3-D using VTK was proposed by Selver et al (49). The same sequence of pipelines can be used to segment the mitochondria from the background image and visualize the results of segmentation of mitochondria in 3D.

REFERENCES

[1]   Karnkowska, Anna, et al. "A eukaryote without a mitochondrial organelle." *Current Biology* 26.10 (2016): 1274-1284.

[2]   Wegrzyn J, Potla R, Chwae Y-J, Sepuri NB V, Zhang Q, Koeck T, et al. Function of Mitochondrial Stat3 in Cellular Respiration. Science. 2009;323(February):793–8.

[3]   Herman, Gabor T. Fundamentals of computerized tomography: image reconstruction from projections. Springer Science & Business Media, 2009.4.

[4]   Mohammadi Alamdari A. Semi Automatic segmentation of mitochondria from electron microscopy images using Kalman filtering approach. Middle East Technical university.

[5]   Terrence G. Frey, 1 Guy A. Perkins 2, Ellisman and MH. Electron Tomography of Membrane-Bound Cellular Organelles.

[6]   Williams, David B., and C. Barry Carter. "The transmission electron microscope." Transmission electron microscopy. Springer Us, 1996. 3-17.

[7]   Clarke, Ian C. "Articular cartilage: A review and scanning electron microscope study." Bone & Joint Journal 53.4 (1971): 732-750.

[8]   Denk W, Horstmann H. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. PLoS Biol. 2004;2(11).

[9]   Chavan V, Willis J, Walker SK, Clark HR, Liu X, Fox MA, et al. Central presynaptic terminals are enriched in ATP but the majority lack mitochondria. PLoS One. 2015;10(4):1–19.

[10]   Wanner AA, Krrschmann MA, Genoud C, Kirschmann MA, Genoud C. Challenges of microtome-based serial block-face scanning electron microscopy in neuroscience. J Microsc. 2015;259(2):137–42.

[11]   Taşel fs. Detection and Segmentation of Mitochondria From Electron Microscope Tomography Images. 2016.

[12]   Wallace, Douglas C. "Mitochondrial diseases in man and mouse." Science 283.5407 (1999): 1482-1488.

[13]   Seyedhosseini, Mojtaba, Mark H. Ellisman, and Tolga Tasdizen. "Segmentation of mitochondria in electron microscopy images using algebraic curves." Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on. IEEE, 2013.

[14]   Narasimha R, Ouyang H, Gray A, McLaughlin SW, Subramaniam S. Automatic

joint classification and segmentation of whole cell 3D images. Pattern Recognit. 2009;42(6):1067–79.

[15]   Subramaniam JLSM& S. Cryo-electron tomography of bacteria: progress, challenges and future prospects. Nat Rev Microbiol. 2009;7, 666– 675:doi:10.1038/nrmicro2183.

[16]   Mustafa Ç. Semi-automatic/User-guided Segmentation of Mitochondria on Transmission Electron Microscopy Images.

[17]   Martone ME, Zhang S, Gupta A, Qian X, He H, Price DL, et al. The cell-centered database: a database for multiscale structural and protein localization data from light and electron microscopy. Neuroinformatics. 2003;1:379–95.

[18]   Martone ME, Tran J, Wong WW, Sargis J, Fong L, Larson S, et al. The Cell Centered Database project: An update on building community resources for managing and sharing 3D imaging data. J Struct Biol. 2008;161(3):220–31.

[19]   Martone ME, Gupta A, Wong M, Qian X, Sosinsky G, Ludäscher B, et al. A cell-centered database for electron tomographic data. J Struct Biol. 2002;138(1–2):145–55.

[20].   Huang, Xiaolei, and Gavriil Tsechpenakis. "Medical image segmentation." Information Discovery on Electronic Health Records 10 (2009): 251-289.

[21]   Seyedhosseini M, Kumar R, Jurrus E, Giuly R, Ellisman M, Pfister H, et al. Detection of Neuron Membranes in Electron Microscopy Images Using Multi-scale Context and Radon-Like Features BT - Medical Image Computing and Computer-Assisted Intervention --MICCAI 2011. Med Image Comput Comput Interv --MICCAI 2011. 2011;6891:670–7.

[22]   Lucchi A, Smith K, Achanta R, Lepetit V, Fua P, Jiang T, et al. A Fully Automated Approach to Segmentation of Irregularly Shaped Cellular Structures in EM Images\rMedical Image Computing and Computer-Assisted Intervention – MICCAI 2010. 2010;6362:463.

[23]   Giuly R, Martone M, Ellisman M. Automatic Detection and Segmentation of Mitochondria in 3D Electron Tomographic Images. CytosegGooglecodeCom. (August 2015).

[24]   Navlakha S, Ahammad P, Myers EW. Unsupervised Segmentation of Noisy Electron Microscopy Images Using Salient Watersheds and Region Merging.

[25]   Ding L, Goshtasby A. On the canny edge detector. Pattern Recognit. 2001;34(3):721–5.

[26] B OO, Gomez A, Keraudren K, Schuh A, Bai W, Shi W, et al. Probabilistic Edge Map (PEM) for 3D Ultrasound Image Registration and Multi-atlas Left Ventricle Segmentation. 2015;1(Lv):223–30.

[27] Liu T, Jones C, Seyedhosseini M, Tasdizen T. A modular hierarchical approach to 3D electron microscopy image segmentation. J Neurosci Methods. 2014;226:88–102.

[28] Leitmann G. Partial Differential Equations and Finite-Difference Methods in Image Processing, Part 1: Image Representation. Optimization. 1977;23(September):65–91.

[29] Jones, Cory, et al. "Neuron segmentation in electron microscopy images using partial differential equations." Biomedical Imaging (ISBI), 2013 IEEE 10th International Symposium on. IEEE, 2013.

[30] Olabarriaga SD, Smeulders AWM. Interaction in the segmentation of medical images. Med Image Anal. 2001;5:127–42.

[31] Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, et al. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. Neuroimage. 2006;31(3):1116–28.

[32] Malmberg, Filip, et al. "Smart paint a new interactive segmentation method applied to MR prostate segmentation." MICCAI Grand Challenge: Prostate MR Image Segmentation 2012 (2012).

[33] Higaki T, Kutsuna N, Hasezawa S. CARTA-based semi-automatic detection of stomatal regions on an Arabidopsis cotyledon surface. 2014;26:9–12.

[34] Higaki T, Kutsuna N, Akita K, Sato M, Sawaki F, Kobayashi M, et al. Semi-automatic organelle detection on transmission electron microscopic images. Sci Rep. 2015;5:7794.

[35] Kang Y, Engelke K, Kalender WA. Interactive 3D editing tools for image segmentation. Med Image Anal. 2004;8(1):35–46.

[36] Barrett WA, Mortensen EN. Interactive live-wire boundary extraction. Med Image Anal. 1997;1(4):331–41.

[37] Haralick RM. Digital Step Edges from Zero Crossing of Second Directional Derivatives. IEEE Trans Pattern Anal Mach Intell. 1984;PAMI-6(1):58–68.

[38] Huertas A, Medioni G. Detection of intensity changes with subpixel accuracy using Laplacian-Gaussian masks. IEEE Trans Pattern Anal Mach Intell. 1986;PAMI-8(5):651–64.

[39] Canny J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell. 1986;PAMI-8(6):679–98.

[40] Riškus A. Approximation of a cubic Bézier curve by circular arcs and vice versa. Inf Technol Control. 2006;35(4):371–8.

[41] Graph D, Graph E, Graph H, Matrix A, Matrix I. Dijkstra s algorithm. 2010;2009–10.

[42] Krig S. Ground Truth Data, Content, Metrics, and Analysis. Comput Vis Metrics. 2014;283–311.

[43] Godfrey, A. C. R. M. D. O. G., et al. "Direct measurement of human movement by accelerometry." Medical engineering & physics 30.10 (2008): 1364-1386.

[44] Taha AA, Hanbury A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. BMC Med Imaging. 2015;15:29.

[45] Massey, Adam, and Steven J. Miller. "Tests of hypotheses using statistics." Mathematics Department, Brown University, Providence, RI 2912 (2006).

[46] Cogliati S, Frezza C, Soriano ME, Varanita T, Quintana-cabrera R, Corrado M, et al. Mitochondrial Cristae Shape Determines Respiratory Chain Supercomplexes Assembly and Respiratory Efficiency. Cell. 2013;155(1):160–71.

[47] Nguyen H, Ji Q. Shape-driven three-dimensional watersnake segmentation of biological membranes in electron tomography. IEEE Trans Med Imaging. 2008;27(5):616–28.

[48] Selver, M. Alper, et al. "Semi-automatic segmentation methods for 3-D visualization and analysis of the liver." MIE. 2014.

[49] Mehnert A, Jackway P. An improved seeded region growing algorithm. Pattern Recognit Lett. 1997;18(10):1065–71.

[50]   Revol C, Jourlin M. A new minimum variance region growing algorithm for image segmentation. Pattern Recognit Lett. 1997;18:249–58.