SEMI-AUTOMATIC SEGMENTATION OF MITOCHONDRIA ON ELECTRON MICROSCOPY IMAGES USING KALMAN FILTERING APPROACH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

AYNAZ MOHAMMADI ALAMDARI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MEDICAL INFORMATICS

MARCH 2016

SEMI-AUTOMATIC SEGMENTATION OF MITOCHONDRIA ON ELECTRON MICROSCOPY IMAGES USING KALMAN FILTERING APPROACH

Submitted by **Aynaz Mohammadi Alamdari** in partial fulfillment of the requirements for the degree of **Master of Science in The Department of Medical Informatics Middle East Technical University by**,

Prof. Dr. Nazife Baykal _____
Director, **Graduate School of Informatics**

Assoc. Prof. Dr. Yeşim Aydın Son _____
Head of Department, **Medical Informatics**

Prof. Dr. Ü. Erkan Mumcuoğlu _____
Supervisor, **Medical Informatics**

Examining Committee Members:

Assoc. Prof. Dr. **Yeşim Aydın Son** _____
Medical Informatics, METU

Prof. Dr. **Ü. Erkan Mumcuoğlu** _____
Medical Informatics, METU

Assist. Prof. Dr. **Aybar Can Acar** _____
Medical Informatics, METU

Assist. Prof. Dr. **Reza Hassanpour** _____
Computer Engineering, Cankaya University

Assoc. Prof. Dr. **Tolga Esat Özkurt** _____
Medical Informatics, METU

Date: 08.03.2016

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 work.

Name, Last name: Aynaz Mohammadi Alamdari

Signature : _____

ABSTRACT

SEMI-AUTOMATIC SEGMENTATION OF MITOCHONDRIA ON ELECTRON MICROSCOPY IMAGES USING KALMAN FILTERING APPROACH

Mohammadi Alamdari, Aynaz

MS., Department of Health Informatics

Supervisor: Prof. Dr. Ü. Erkan Mumcuoğlu

March 2016, 69 Pages

Mitochondria are membrane bound organelles found in most eukaryotic cells. Mitochondria provide cell's energy; hence they are called 'power houses of the cell'. The structure of mitochondria can be illustrated in an electron micrograph. This structure has two membranes: inner and outer. There is a gap between these two membranes, called inter-membrane space. Folds of inner membrane inside the mitochondria form the cristae. To study the relation between mitochondria's physical structure and its function, electron microscope tomography (EMT) is used to visualize mitochondria. EMT provides 3D structure of mitochondria in high resolution images. In the slices of tomographic images provided by EMT, mitochondria appear as elliptical structures. The cristae are also visualized in these images with various pathology and biological variations. One of the preferred method can be semi-automatic segmentation; since manual segmentation in medical images is time and energy consuming and tedious; moreover fully automatic methods also fail in medical images and cause incorrect results because of low quality of images and restrictions imposed by image acquisition. In this work, an endeavour is made to segment mitochondrial outer boundary using active contour, Kalman filter and optical flow. In the first slice of the images, a contour is provided by user. Then, for the other slices, position values and velocity values calculated using the active contour and optical flow (respectively) are combined with the Kalman filter to predict the points of the boundary in the next slice. In addition, a set of automatic and semi-automatic tools are developed to determine splitting and merging mitochondria, and to segment them.

Key words: Semi-automatic segmentation, mitochondria, active contour, optical flow, Kalman filter

ÖZ

ELEKTRON MIKROSKOBU GÖRÜNTÜLERI ÜZERİNDE KALMAN FİLTRELEME YAKLAŞIMINI KULLANILARAK YARI-OTOMATİK MITOKONDRI BÖLÜTLEME

Mohammadi Alamdari, Aynaz

Yüksek Lisans, Sağlık Bilişimi Bölümü

Tez Yöneticisi : Prof. Dr. Ü. Erkan Mumcuoğlu

Mart 2016, 69 Sayfa

Mitokondri çok ökariyotik hücrelerde bulunan zar bağlı organellerdir. Mitokondri hücrenin enerjisini sağlamaktadır; dolayısıyla 'hücrenin enerji evleri' olarak adlandırılır. Mitokondri yapısı, elektron mikrografıyla tasvir edilebilir. Bu yapı, iki membrana sahiptir: iç ve dış. Arası zar alanı olarak adlandırılan bu iki membran arasında bir boşluk vardır. Mitokondri içinde iç zarının kıvrımları kristaları oluşturur. Mitokondrinin fiziki yapısı ve işlevi arasındaki ilişkiyi incelemek için, elektron mikroskobu tomografi (EMT) mitokondriyi görselleştirmek için kullanılır. EMT yüksek çözünürlüklü görüntülerde mitokondri 3B yapısının görüntüsünü sağlar. EMT tarafından sağlanan tomografi dilimleri içinde, mitokondri eliptik yapı olarak görünür. Kristalarında çeşitli patoloji ve biyolojik varyasyonları bu görüntülerde görüntülenmiştir. Tercih yöntemlerinin biri yarı-otomatik bölütleme olabilir. Tıbbi görüntülerde elle bölümleme zaman / enerji tüketir, sıkıcıdır.Tam-otomatik yöntemlerse bazen başarısız olmaktadır. Çünkü görüntüler düşük kalitededir ve görüntülerin elde edilmesinde belli kısıtlamalar vardır. Bu çalışmada, aktif sınır, Kalman filtre ve optik akış yöntemleri kullanarak mitokondri dış sınırına yarı-otomatik bölümleme yöntemleri önerilmiştir. Görüntülerin birinci diliminde, bir sınır eğrisi kullanıcı tarafından çizilir. Daha sonra, tasarlanan algoritma aracılığıyla, sonraki dilimler için, pozisyon / hız değerleri hesaplanır. Aktif sınır ve optik akış yöntemleri ve Kalman filtresi ile bir araya getirilerek sınır noktalarının bir sonraki dilimdeki yerlerini tahmin edilir. Buna ek olarak, bir dizi otomatik ve yarı-otomatik araçlar, mitokondri bölme / birleştirme ve bölütleme için geliştirilmiştir.

Anahtar Sözcükler: yarı-otomatik bölümleme, mitokondri, aktif sınır, optic akış, Kalman filtre

To My Family

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ASSD	Average Symmetric Surface Distance
CRF	Conditional Random Field
EM	Electron Microscopy
EMT	Electron Microscopy Tomography
FCM	Fuzzy C-Means
GRIMS	Gaussian Rotation Invariant and Multi Scale
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
MSSD	Maximum Symmetric Surface Distance
NCMIR	National Center for Microscopy and Imaging Research
OOI	Object of Interest
RMSSSD	Root Mean Square Symmetric Surface Distance
ROI	Region of Interest
SBFSEM	Serial Block-Face Scanning Electron Microscopy
SEM	Scanning Electron Microscopy
SRG	Seeded Region Growing
TEM	Transmission Electron Microscopy
UsRG	Unseeded Region Growing

CHAPTER 1

INTRODUCTION

Mitochondria are membrane bound organelles found in most eukaryotic cells (1). The term mitochondrion (plural mitochondria) comes from Greek roots meaning "thread" and "gain", first introduced by Benda. Mitochondria are known to be dynamic organelles because their shape and size are highly variable. Mitochondrial shape affects its distribution and participation in apoptosis. This leads to the importance of mitochondrial dynamics in cells that have special dependence on mitochondrial function. Therefore studying structure of mitochondria is important for the functional state of mitochondria (2). Mitochondria have been primarily known as "power house of the cell". The energy producing function of mitochondria which is necessary for cell growth and biological activities is accomplished by a series of complex chemical reactions (2). It has been estimated that 90% of mammalian oxygen consumption is mitochondrial (3). However, it is also involved in other activities like cell signalling, cellular differentiation, cell division cycle, cell growth and programmed cell death known as apoptosis (4) . Mitochondria have two membranes: inner membrane and outer membrane, and there is a gap between these two membranes called inter-membrane space. Folds of inner membrane inside the mitochondria form the cristae and the space within the inner membrane is called the matrix . These mitochondrial structural features are all shown in Figure 1.

Visualizing mitochondria using electron microscope tomography can help us study the relation between the mitochondrial function and its physical structure. Some sample images of mitochondria using electron microscopy tomography technique is presented in Figure 2. A work by Palade and Sjostrand published in 1953 introduced electron micrographs showing mitochondria, since then electron microscopy (EM) has progressed a lot and has emerged as the leading technique for three dimensional (3D) structural analysis of unique complex biological specimens (5) and a powerful tool to study mitochondrial ultra-structure and function (6). EM provides greater resolution than light microscopy; it also enables 3D reconstruction of images using digital image analysis. Electron microscopy allows the study of mitochondrial morphology and its overall organization (7). Since the images of three dimensional objects generated by conventional EM are two dimensional, 3D imaging techniques

like Scanning Electron Microscopy (SEM) and Transmission Electron Microscopy (TEM) have been introduced.

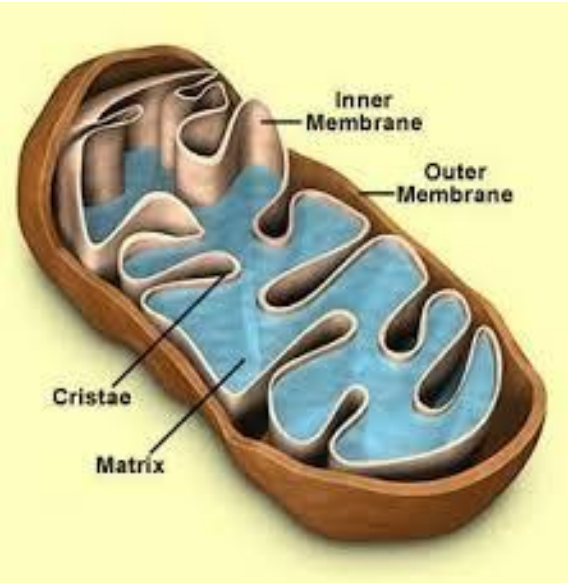


Figure 1 Mitochondrion Structural Features (Original image by **(8)**)

In TEM technique there is an electron gun that produces electron beam projected by electric field which is generated by condenser and objective lenses. The ultra-thin prepared specimen is placed and the electron beam penetrates through it; then its image is created on the fluorescent screen. There is a detector that measures the intensity of electron which is proportional to the intensity of produced image. The detector obtains the projection data by scanning image in different tilted angles of electron beams. (9), (10) A collection of large number of projections is created in this way then the reconstruction of 3D image of object will be possible. The reconstruction can be done by direct back projection or Fourier transforms as shown in Figures 3 and 4.

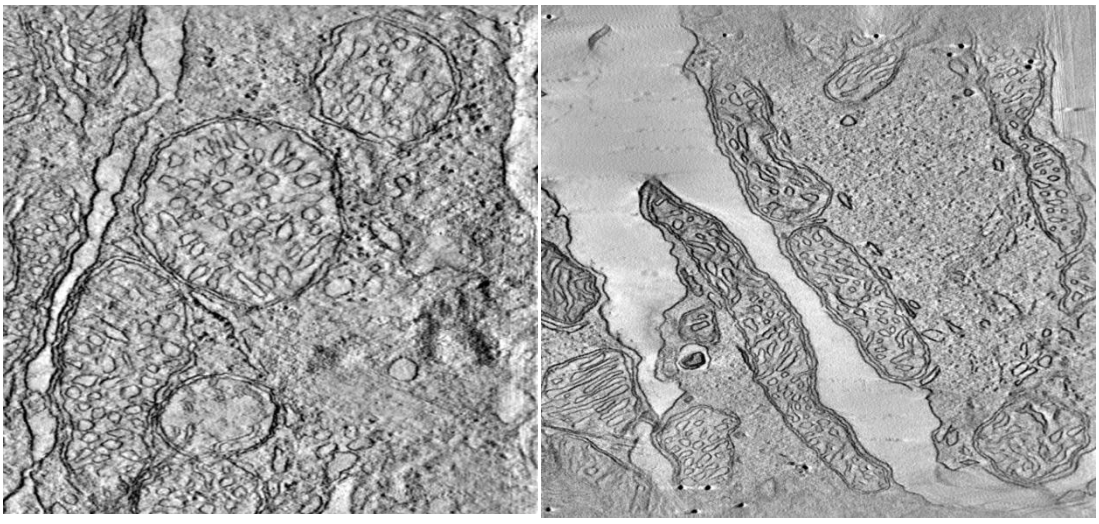


Figure 2 Samples of Electron Microscopy Tomography Images of Mitochondria

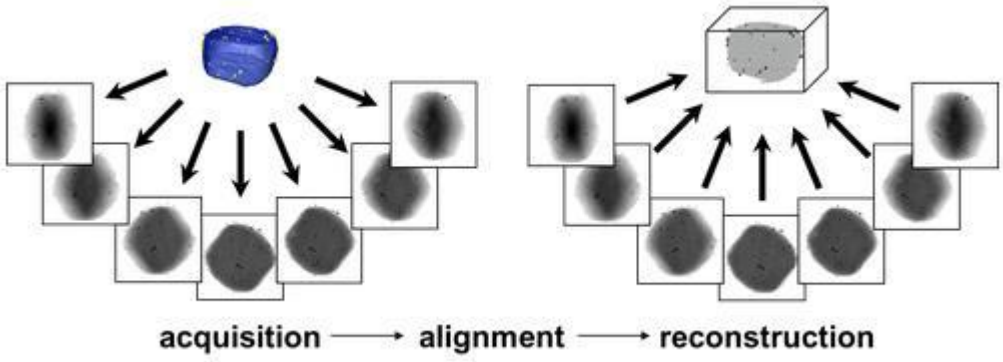


Figure 3. Tomographic reconstruction by back Projection in TEM (Original image by **(11)**)

In SEM, the scanning coils generate the electric field that deflects the focused electron beam on the specimen. The electrons that scattered from the surface of specimen have an intensity that is measured by a detector. The image generated by SEM is 2D. In order to have a 3D image of specimen a special type of SEM called Serial Block-Face Scanning Electron Microscopy (SBFSEM) is used. In SBFSEM very thin layers of specimen are cut with a diamond knife and then these layers are imaged with SEM technique so a stack of 2D images of every slice will be in hand to form 3D volumetric data. (12). Schematics of TEM and SEM techniques are shown in Figure 5.

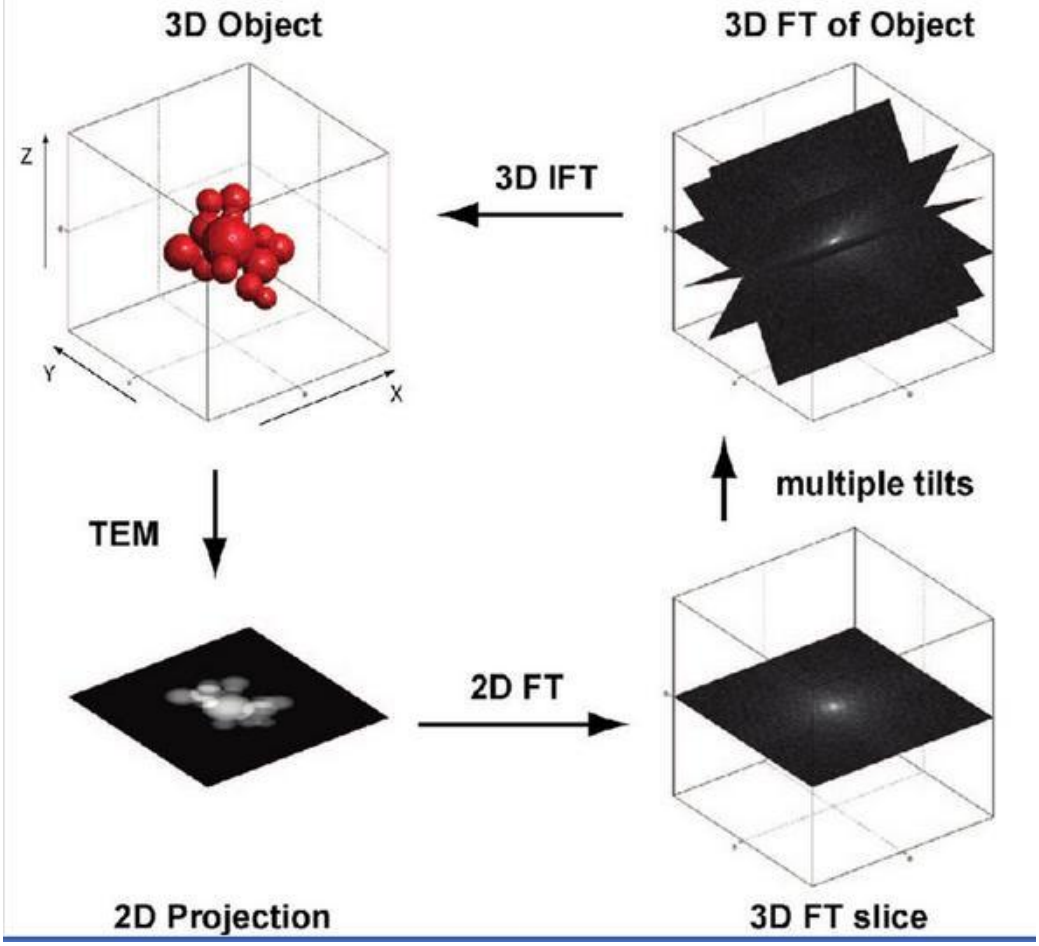


Figure 4. Tomographic reconstruction by Fourier Transform in TEM (Original image by **(11)**)

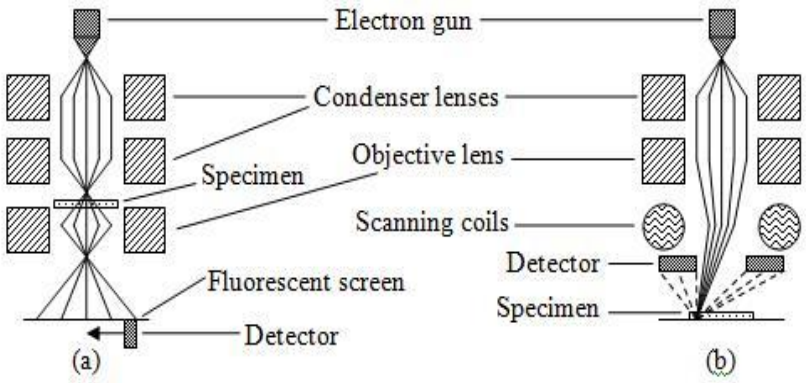


Figure 5. Schematics of a) Transmission Electron Microscopy (TEM) and b) Scanning Electron Microscopy (SEM) (Original image by **(13)**)

1.1 Motivation

Mitochondria are membrane bound organelles found in most eukaryotic cells (1). Mitochondria are known to be dynamic organelles because their shape and size are highly variable. Mitochondrial shape affects its distribution and participation in apoptosis that leads to the importance of mitochondrial dynamics in cells that have special dependence on mitochondrial function. Therefore studying structure of mitochondria is important for the functional state of mitochondria (2). The morphology of intracellular components is of great importance for neuroscientists. For instance, abnormalities in mitochondria morphology are seen in Parkinson disease or geometrical property of mitochondria can be used to distinguish cancer cells from normal ones. Manual segmentation in medical applications is very slow, tedious, time and energy consuming, needs training for the segmentation and has subjective results. In some cases fully automatic methods fail and cause incorrect results due to the difficulty of segmentation in medical images along of restrictions imposed by image acquisition, pathology; Moreover biological variation and fully automatic segmentation cannot deal with artefacts on its own. All of these issues lead the semi-automatic method to be the preferred method of all.

The main motivation for this study is the work by Tasel (14). In their work mitochondria are detected and segmented using their characteristics like elliptical shape and double membrane. In their work, they refine their results by using active contour and live wire algorithms. In this study, a method for segmentation of mitochondria is implemented then tested and compared with the results of Tasel (14). In another work by Çöçelli (15) semi-automatic methods were implemented to correct wrong segmentation results of Tasel (14) with visualization support. The tools that Çöçelli (15) used for correction of results are splitting, merging, deletion, selection of low scored mitochondria and initiation of auto-segmentation algorithm. The interaction of user is needed to correct the result of automatic segmentation whenever a mitochondrion is going to split and then the two mitochondria are considered one in the automatic segmentation and this is called splitting. To correct this Çöçelli (15) proposed a method to put a barrier between two mitochondria and construct a plane between them by arbitrarily defining three points. Merging is the case when two mitochondria are detected instead of one real mitochondrion. To correct this Çöçelli (15) provided an option in his application letting the user select the mitochondria which are wrongly detected as two separate mitochondria instead of one. Later the center of mass of two mitochondria are determined they are rotated until the line connecting centers of two masses become parallel to x-y axis then finding the peak points and finally deleting the points between these peak points on the contours. Deletion is used when a non-mitochondrial region is detected as mitochondria.

The main contribution of this thesis is the implementation and adaption of the algorithm by Jurrus (16). Jurrus (16) used their technique for the detection and segmentation of axons. In this thesis some parts of the algorithm are modified and extended to segment mitochondria in 3D images. In addition, a special technique for splitting and merging mitochondria cases are also proposed and tested.

1.2 Background and previous work

1.2.1 Segmentation Background

Medical image segmentation has been important because of its usefulness in visualizing and comparing human body's inner structure. Cell and sub cellular segmentation in biomedical images is of great importance because it helps in diagnosis of diseases. It is also helpful in cell biology researches. The segmentation methods can be categorized in three categories: manual, semi-automatic and automatic segmentation of these structures. Manual segmentation is very slow, tedious, time and energy consuming in medical applications; at the same time, it requires training and being very accurate and detailed for the one doing the segmentation and the results are non-reproducible. (17) Manual segmentation has a bad reputation because of subjective results (18). In fully automatic segmentation there is no need for training and the data are reproducible but it limits the user's authority (17). Sometimes fully automatic methods fail and cause incorrect results; this happens more in medical images because segmentation is difficult in this type of data for the sake of restrictions imposed by image acquisition, pathology and biological variation. Fully automatic segmentation cannot deal with artefacts on its own. All of these issues cause the semi-automatic method to be the preferred method of all (17). In order to select an appropriate segmentation technique, type of image and the application area should be considered.

Automatic image segmentation methods can be categorized (19) as follows:

a. Intensity based methods

b. Discontinuity based methods

c. Region based methods

d. Clustering methods

e. Graph based methods

f. Pixon based methods

g. Hybrid methods

a. Intensity based methods

One of the approaches to segment an image according to its intensity level is called threshold based approach which is of the most frequently used ones. The threshold operation is a grey level remapping operation in such a manner that it maps a grey-valued image to a binary image. After thresholding, the result is an image of two segments identified by the pixel values of 0 and 1 respectively as shown in Equation1.

$$G(x,y) = \begin{cases} 1 & for\ I(x,y) \geq t \\ 0 & for\ I(x,y) < t \end{cases} \qquad (Eq.1)$$

Where G(x, y) is the output image and I(x, y) is the input image and *t* is threshold value. (20)

b. Discontinuity based methods

If there are objects in an image, there will be the boundaries. The existence of boundaries leads to formation of edges. The edges are defined as the changes in intensity that causes the discontinuity in the pixels. In other words, a boundary that partitions two homogenous regions is called the edge and identifying and locating these discontinuities is called edge detection. Edge detecting operators can be in one of these two categories: first order derivative operators then the second one. They can be named Perwitt, Robert, Sobel and the Feri Chen mask as first order derivative operators. These operators convolve a small square kernel with the image to compute the gradient. Second order derivative operators are as Laplacian of Gaussian operator and Canny edge operator. Second order operators give reliable results. Edge detection techniques clarify whether the pixels are of an edge or not. (20)

c. Region based methods

These methods are based on this thought that a pixel and the pixels in its neighbourhood share similar characteristic and are considered to be in one region different from other regions whose pixels have different characteristics from these ones. A simple way of classifying pixels in regions is comparing a pixel with its neighbour pixel and checking it's similarity to that pixel, if it is similar it is considered to be in that region, if not, the similarity test will stop. The region based methods are of two types: 1. Region growing methods 2. Region splitting and merging methods

Region growing based methods:

This method segments the image into regions having some predefined criteria in hand. There are two types of region growing segmentation methods: 1. Seeded region growing (SRG) which is semi-automatic; 2. Unseeded region growing (UsRG) method which is automatic segmentation method. In SRG method an initial seed point is specified by the user (it causes the method to be a semi-automatic method). The seed point is a test pixel with the characteristics of the located region. There may be one or more sets of seeds according to the users' will. In each step a pixel is added to one of the seed sets. The choice of seed is important due to its dependant algorithms. In (UsRG) there is no user intervention and no seed point; it starts with assumption that the first pixel is a member of assumed A_i region, then for the next pixel the difference measure of test pixel, from the mean value of the statistics of the region, decides whether the pixel is in the region or not. In this way, if the difference is less than a threshold, the test pixel is considered to be in the region, if not, a new region assumed A_j is created and the test pixel gets in that region. This continues for every pixel. (20)

Region split and merging methods:

The basis of this method is quadtree. In the first step it assumes the image as one whole region; then by having predefined criteria in hand it divides the image into

four quadrants. Every quadrant is checked with that predefined criteria and, if necessary, dividing those quadrants too till there is no quadrant to be divided or the criteria are satisfied. The Figure 6 illustrates the method better. (20)

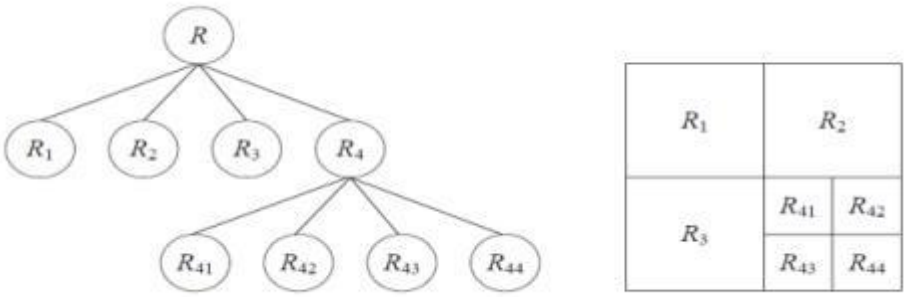


Figure 6. Quadtree (Original image by **(19)**)

d. Clustering based methods:

Clustering is gathering pixels with similar characteristics into one cluster whose characteristic is different from others. The usual categorization of clustering methods can be partitional clustering methods and hierarchical methods.

Hierarchical methods: hierarchical clustering can be classified into two methods: agglomerative and divisive. Agglomerative is a bottom-up approach in which each pixel is considered to have its own cluster, the pixels close to each other merge successively till all clusters merge together and create one cluster or until a condition makes it stop. Actually it stops when the desired structure is obtained. The divisive approach is a top-down approach in which all pixels form one cluster and then they are divided into sub clusters until the desired structure is obtained. All the merging and dividing decisions are made according to a similarity measure (20) , (21).

Partitional clustering: In this method number of clusters should be known beforehand and it starts from an initially partitioning. It gets updated by changing the clusters which the pixels belong to. There are various algorithms of partitional clustering but the most famous ones are k means and Fuzzy C-Means (FCM) algorithms. In k means, the algorithm starts with initial clusters; every cluster is represented by its center or mean. In each step, pixels go to the clusters with the smallest distance from cluster center according to Euclidean distance between them, and then the cluster centers are recalculated. The center of each cluster is calculated as the mean of all pixels that belong to that cluster. Various stopping criteria might be used. Fuzzy C-Means allow a pixel to be a member of different clusters with different membership coefficient. The clustering algorithms use various features of image like its intensity, texture, etc. (20).

e. Graph based methods: If it is assumed that the graph $G = (V, E)$ where $V = \{V1, V2, V3, \ldots\ldots, Vn\}$ a set of vertices of graph, in image is the vertices are pixels and E is a set of edges connecting these vertices$E = (Vi, Vj)$. Each edge has its corresponding weight $W(Vi, Vj)$. In the case of image segmentation the weight can be any measure of dissimilarity between two connected pixels like

difference in intensity, motion, location, etc. An image can be divided to nonempty components which are graphs themselves. Among various methods of graph based methods the most used one is graph cut method in which the dissimilarity function can be designed as graph cut. The graph cut method, as its name implies, partitions the graph by graph cut which is a subset of edges. These edges partition the graph into two disjoint sets. Figure 7 best illustrates the graph cut method.

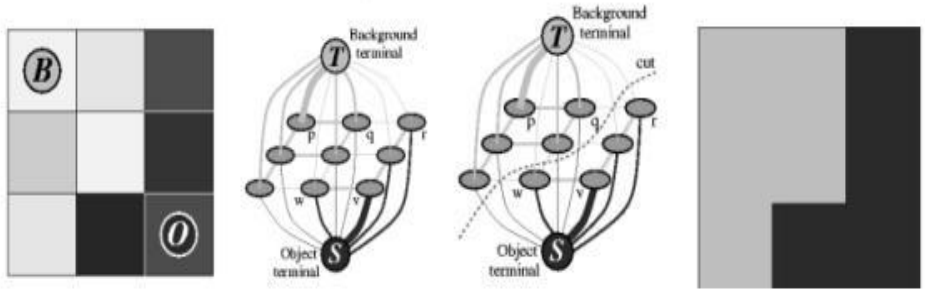


Figure 7. Graph Cut (Original image by **(19)**)

f. Pixon based methods: In the traditional pixon based method, first the pixons are formed then the segmentation is done. Forming the pixons is done in three steps: 1. providing a pseudo image with the same resolution of original one 2. Forming pixons with the use of anisotropic diffusion filter 3.extracting pixons with the use of hierarchical methods. After pixons' extraction, the image can be considered to have a graph structure; if every pixon has a label, the segmentation is done. Combining the pixons will continue till satisfaction of stop criteria happens and the final segmented image will be achieved.

g. Hybrid methods: Combination of various methods can be used to achieve a better performance.

1.2.2 Semi-Automatic Segmentation

For many years automatic segmentation was one of the most important tasks in image processing field. Since the medical images are most of the time of low quality, bad defined and restricted by image acquisition, pathology and biological variations, automatic segmentation methods often fail in these types of images. After some years semiautomatic and interactive segmentation became the solution for this problem. In this type of segmentation, the user intervenes in the process of segmentation by drawing the borders correcting and editing the work done by automatic segmentation and actually validating and correcting errors. In semiautomatic segmentation the interaction between the user and the image is very low and limited in comparison with interactive (manual) segmentation.

In (22) , two methods for semi-automatic segmentation were introduced, one based on fuzzy connectedness and the other one based on watersheds. They improve fuzzy connectedness by introducing the paradigm of competitive learning and omitting thresholding. The main idea in fuzzy connectedness is to compute a map which indicates connectedness of every pixel in the image with relation to a specified pixel in the OOI (Object of Interest). In the present form of fuzzy connectedness method just one object can be detected at any time; So there is a need to threshold the connectedness map but in the improved fuzzy connectedness method introduced by (22) more than one object are labelled by the user and labelling of every seed is done by computing affinity of every seed and finally choosing the maximum affinity for every seed. In the proposed watershed method in Bonnet (22) despite the original watershed that algorithm detects the local minima of the spatial gradient, as the seeds for algorithm to find watersheds, the user provides the seeds for objects that are needed to be segmented and also background connected parts by clicking on them, then starting the watershed algorithm. At the end of the procedure, every pixel of image is labelled the same as the seed it belongs to. In Bonnet (22) two tests are done and in these tests the fuzzy connectedness based algorithm with competition shows better results than the fuzzy connectedness based algorithm with no competition and in the specified test the competitive fuzzy connectedness based approach performs better than the competitive watersheds method. In most of the cases two methods of performance are very similar and it is recommended two algorithms to be applied and one of them to be chosen by the user.

In Piali Das (23) the focus is on industrial applications; the image is of integrated chips that requires transistor gates to be segmented from the image. In this paper they use graph cut method for semi-automatic segmentation. They assume that the object to be segmented is compact in shape. One of the issues they consider is parameter selection that if it is done by the user, it will need more interaction; that causes not to be in the semi-automatic group of segmentations. Values of parameters have great influence on the result of algorithm and segmentation. If the set of images are not so variable, parameter values can be selected beforehand but in the case of variable images, it is not possible to select parameter values beforehand. For every set of images a set of parameter values work well, so they think of a method that runs the segmentation algorithm for a range of parameters and then select the parameter values with higher quality. The quality is measured by ‘quality check’ which decides whether the selected parameter is appropriate or not, if the segment does not pass this ‘quality check’, the graph cut step should be repeated with new parameter values. This process is done iteratively until the resulting segment passes the ‘quality check’. With the use of this quality check the best parameter values, which can be used for the algorithm, may be detected. Their algorithm produces highly accurate segmentation in real time.

In Xie (24) they use active contour and multi-scale curve editing to interactively segment medical images. The algorithm they propose works in three steps. In the first step the user selects the ROI (region of interest) manually and then revises automatically generated control points by mouse clicks. Second automatic active contour segmentation is done and then in last step, the user edits the contour manually by dragging the control points with multi-scale spacing. In the first step that user selects ROI manually and then automatically control points are generated

by curve fitting method, named Hermite Cubic Curve, which smoothly interpolate between control points in this way, the user drags a point and no other point's location changes until the user confirms it, then all other control points change location according to Hermite Cubic Curve. In the last step where manual multi-scale spacing is done by the user, there may be some areas in the border that does not need any intervention; meanwhile, there may be areas that need correcting. It may be done in some levels until the user decides if it converge the correct segmentation. In different levels, the number of control points increases and the space between them decreases, the number of control points change hierarchically according to the length of region boundary. They test their algorithm on CT images, MRI images and ultrasound images but they do not have ground truth to compare the results, so they do not have a formal quantitative evaluation. The time necessary for whole process is real time, no matter the region of boundary is small or large.

In Yu-Chi J. Hu (17) they propose a method for semiautomatic segmentation using brush strokes to provide input for the algorithm based on CRF (Conditional Random Field) segmentation. Their focus is on separating target tissue from background. The user inputs are used to extract energy function terms and the form of energy function is based on CRF and the final solution is determined by a graph min s-t cut algorithm. In brief, it can be said that the user firstly chooses seed points on one image by paint brushes from both target and non-target regions, and on the following slices the boundary term is estimated from samples in the training slice, but the regional term is estimated from the new brush strokes on the current slice being segmented. In any slice, the user can retrain the model if statistics are not applicable any more. Their test done on liver shows that there is no boundary leakage in this algorithm unlike the region growing and level set methods because of smoothness weight in CRF and global optimal solution in graph cut, and the time needed for interaction in this method is very low in comparison with original graph cut method.

1.2.3 Previous Study

In the study of Aurélien Lucchi (25) they used super-voxel based segmentation of mitochondria on EM (Electron Microscopy) images. They clustered similar voxels into regularly spaced groups of uniform size and named them super-voxels, then they extracted feature vectors to get local shape and texture information for each super-voxel, in the last step segmentation of mitochondria was done by using a graph cut approach which uses unary and pairwise potentials of the energy function to incorporate shape cues and learned boundary appearance. The data set they used comes from two different locations in the brain: hippocampus and striatum .For hippocampus they use 165 slices of image for testing purpose and 200 slices for training from the whole 1024×1024×1000 image stack. For the striatum, the training and test sets are both of size 768×872×318. The study exposes that the approach studied is able to segment mitochondria at a performance level close to that of a human annotator. There are also failures in this approach such that it can include part of nearby membrane with the mitochondria or neighboring mitochondria are merged or clusters of vesicles can be mistaken for mitochondria as a result of their similar texture.

In a work by Dilip. K. Prasad (26) a segmentation algorithm that can model elliptic shape of cells or sub-cellular structures is proposed. The data set they use is of images of mixed cell types (mitochondria and lysosomes) and the images are fluorescence confocal microscope images. The algorithm is of three steps: pre-processing, ellipse fitting and ellipse filtering. The most important part of the algorithm is ellipse fitting step. Among other methods for ellipse fitting, they choose least squares based methods. The method chosen for ellipse fitting has a major role in the performance of algorithm. Their results show that the geometry based least squares fitting method has better performance than other least squares methods. After ellipse fitting step, ellipse filtering can be done with the aid of information about datasets in hand. Information about dataset images and priori information about cells can help having better filtering. In filtering step, unreasonable ellipses are removed. It is shown that both low and high contrast images are very well segmented even in images with occluded structures.

In the study of Kendrick Cetina (27) they compare nine image feature descriptors with two classifiers on electron microscopy images of mitochondria and synapses. The feature descriptors they use are: **A**. Simple Window and Histogram which is a very good descriptor for texture segmentation. **B**. Local Binary Patterns which generate a binary code for every pixel and the feature vector is created using histogram of LBP binary codes. **C**. GRIMS (Gaussian Rotation Invariant and Multi Scale) apply linear Gaussian filters to each image with different scales. The feature vector is of size 4n because there are 4 features for every scale. **D**. Ray descriptors: In objects like mitochondria that have strong edges, the ray features show good performance in extracting shape features. **E**. Difference of Gaussian's: In this feature a blurred image is subtracted from another blurred image, but in the paper (27) four subtractions at different scales are done. **F**. Laplacian of Smoothed Image: different scales of Gaussian kernel are used because there is a relationship between the size of blob structures and the size of Gaussian kernel used for smoothing. **G**. Eigenvalues of the structure tensor: The gradients of a function are put into a matrix named structure tensor which represents a point's directions of the gradients. **H**. Histogram of Oriented Gradients: dividing image into small regions called cells and having the gradient directions of all pixels in the cell, the feature vector of HOG is created. **I**. Radon like features use both texture and geometric information about image to segment the structure to be more effective. In this paper they use two classifiers: a Gaussian and Random Forest classifiers. The results show that Random Forest classifier performance is better than Gaussian classifier and among nine feature descriptors GRIMS and simple window descriptors show the better performance than the others.

In the study of Tasel (14) they tried to detect and segment the mitochondria automatically. After automatic detection of mitochondria which is based on elliptical shape and double membrane boundary characteristic of mitochondria, they use active contour to refine the results; in the next step there are seed points which are automatically selected along the contour to be used in live wire graph search algorithm which will finalize the segmentation. They test their algorithm on four images with multiple mitochondria and they automatically detect 52 mitochondria which 42 of them are true but 10 are false. Comparing with a trained reader's segmentation, 91% Dice similarity coefficient is achieved.

In the study by Çöçelli (15) he tries to correct and optimize the results gained by Tasel (14). There are some undetected or wrongly detected mitochondria in the work by Tasel (14) that Çöçelli wants to find a solution for them. He uses semi-automatic and user guided methods which use the information extracted from auto segmentation in his work, also his algorithm is tightly bounded to visualization of the sources. Mainly his work is visualization which is in two phases; first visualizing from electron microscope images and secondly is visualization from snakes detected by automatic segmentation algorithm.

In the study by Jurrus (16) axon tracking is done with the use of active contour, Kalman filter and optical flow. Their algorithm uses feature tracking methods in the images which are in fact 3D volume treated as sequence of 2D images. The set of images they use are acquired by SBFSEM. Firstly the user denoises the images then in the first slice defines axon locations and in the proceeding slices with the use of Kalman filter combined with positional and velocity measurements, then tracks the contour points. With comparison to full-manual tracking this method shows significant time saving, in this method user can intervene and correct axon locations when necessary. Failures to track some axons might also occur in this system.

1.3 Aims and Scope

In this study, it is aimed to segment mitochondria of 3D Electron Microscopy images with a semi-automatic method. Active contour is selected for segmenting mitochondria in every slice of image stack. Because mitochondria is a dynamic and shape changing organelle, optical flow is used to estimate its velocity and Kalman filter is used to estimate its location in the next slice using information from active contour and optical flow . Semi-automatic method is applied such that the user initiates the contour and the algorithm starts. The initial contour is placed near the edge, then image forces draw contour to edges, Kalman filter estimates the location of points on the contour in next slice using velocity from optical flow and placement from active contour.

Three different methods are applied to images here. The methods are as follows: active contour, active contour with Kalman filter and the third method using active contour, Kalman filter and optical flow.

Active contour is preferred here because by changing energy terms, the best result can be achieved, and Kalman filter is used because it can smooth the result of active contour and estimate where the points on contour will go in the next slice of images. optical flow can help us find the velocity of points changing place. It is discussed which method have better segmentation results.

1.4 Problem Statement

As manual segmentation in medical applications is very slow, tedious, time and energy consuming and, at the same time, it requires trained and very accurate and detailed one for the segmentation, and its results being non reproducible makes it not to be a preferred segmentation method. (17) Manual segmentation also has a bad reputation because of subjective results (18). This causes fully automatic segmentation methods to come into existence. Fully automatic segmentation methods

have the benefit of saving time and energy and increasing productivity. In fully automatic segmentation there is no need for training and the data are reproducible but it limits the user's authority (17). Sometimes fully automatic methods fail and cause incorrect results, this happens more in medical images because segmentation is difficult in this type of data because of restrictions imposed by image acquisition, pathology and biological variation and this is because parameter determination in fully automatic segmentation methods is very difficult due to differences in every sample and also fully automatic segmentation cannot deal with artefacts on its own. All of these issues cause the semi-automatic method to be the preferred method of all (17). That is why it is also preferred to use semi-automatic segmentation method here.

The mitochondria images used here are provided by The National Center for Microscopy and Imaging Research (NCMIR) from a Cell-Centered Database Project sustained by Martone (28) , (29) and (30) . Eight datasets are provided and each of them approximately contains hundreds of slices and in each slice a number of mitochondria are recognizable. The information about data sets is provided in Table 1.

The mitochondrion in these data sets need to be segmented but, as it is discussed earlier, fully automatic methods or manual methods are not good solutions for segmentation of mitochondria; so a semi-automatic method is needed.

Table 1 Data Set Information

Data set #	Data set name	Number of slices	Number of mitochondria	Cell/Mito type	Mito condition
1	cone.sub	97	9	cone inner segment	normal
2	6_22.sub	91	14	rod inner segment	aged
3	bclpb-d.sub	61	6	rod inner segment	lead exposed
4	gap18_sub	54	6	rod inner segment	aged
5	mac_serial_sub	111	20	MAC	normal
6	od.sub	91	34	rod inner segment	normal
7	pedicle	31	6	pedicle	normal
8	spherule24mos1_	86	1	spherule	aged

CHAPTER 2

METHODS

2.1 Active contour

Snake is an energy minimizing and controlled continuity spline which with the control of external constraint forces and image forces can make its way toward features like lines and edges. They actually try to extract features of interest in images. They lock onto nearby edges, localizing them accurately (31). To have a model with the ability to interact with higher level processes (31) design energy functions whose local minima provide a set of solutions for higher level processes. Adding proper energy terms to the minimization guides snakes toward local minimum in order to find the desired solution. Actually the solution is found using techniques of variational calculus.

In Michael Kass (31) finding edge is different from traditional one in that it considers connectivity of contour, corners and hence detailed structure of the locally optimal contour. The energy functional used in Michael Kass (31) is as in Equation 2 where $V(s)$ stands for the position of snake $V(s) = (x(s), y(s))$

$$E_{snake} = \int_0^1 E_{snake}(V(s))ds = \int_0^1 E_{int}(V(s)) + E_{image}(V(s)) + E_{con}(V(s))\, ds \quad \text{(Eq.2)}$$

Where E_{int} represents the internal energy of the contour, E_{image} is the image forces, and E_{con} is the external constraints.

The internal energy is calculated by Equation 3.

$$E_{int} = (\alpha(s)|vs(s)|2 + \beta(s)|vss(s)|2) \quad \text{(Eq.3)}$$

The first derivative shows an extreme at the edge (maximal positive or negative steepness) so the first order term makes the snake act like a membrane and second order term makes it act like a thin plate.

In another work by Shah (32) it was mentioned that if the points are not in even space from each other, the first order derivative term will be incorrect by a factor of d_i^2 , where d_i is the distance between points i and i-1. (32) The first order derivative is calculated in Equation 4.

$$\frac{d_{vi}}{d_s} \approx |(v_i - v_{i-1})^2| = (x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 \quad \text{(Eq.4)}$$

An active contour is a continuous curve with deforming ability being controlled by internal and external forces. (33)The external forces can be achieved from image data or imposed as constraints whereas internal forces define the physical properties of the mod (34). The internal forces used in Sahiner B (33) are continuity and curvature of the contour which impose a smoothness constraint on the contour and the external

forces are balloon force and negative of the gradient magnitude which push the contour toward salient image features like edges. To do the segmentation there should be an initial boundary which changes shape iteratively in a manner that internal and external forces can be minimized along the contour.

The contour is represented by a set of points that form the vertices of a polygon $v(c) = (x(c), y(c))$. The energy to be minimized is defined as in Equation 5.

$$E = \sum_{c=1}^{N} [w_{curv} E_{curv} + w_{con} E_{con} + w_{grad} E_{grad} + w_{bal} E_{bal}] \quad \text{(Eq.5)}$$

Where curv, con, grad and bal stand for curvature, continuity, gradient, balloon force respectively. The w in all terms is weight term which should be considered properly to have suitable segmentation in every case.

Curvature Energy is defined as the estimation to second derivative of the contour (33) as in Equation 6.

$$E_{curv} = |v_{c-1} - 2v_c + v_{c+1}| \quad \textit{(Eq.6)}$$

This term should be minimized like other terms because when it is large it is the result of small angle between two sides of the polygon that meet at vertex v(c)

When the angle between two sides of the polygon at vertex v(c) is small, this term is going to be large, as it is wanted the contour to be smoothed and it is aimed to eliminate curvatures from the contour then this term should be minimized so this term can smooth the contour.

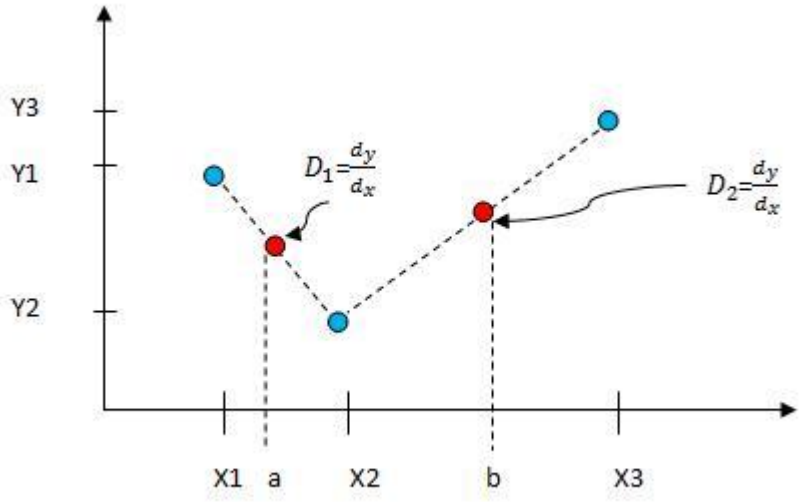


Figure 8. Second Derivative Representation

$$\frac{d^2y}{dx^2} = \frac{D_2 - D_1}{a - b} \qquad (Eq.7)$$

As apparent in Figure 8 as the angle between two sides of the polygon gets smaller, the second order derivative will get larger and when the angle gets larger so the distance between a and b gets larger so the second derivative result will get smaller. Second derivative is defined as in Equation 7.

Continuity energy, E_{con} is defined as the deviation of the length of the line segment S_c between two vertices v_c and v_{c+1} from average line segment $\overline{S} = \sum \frac{Sc}{N}$, $Sc = v_{c+1} - v_c$ (33). This term tries to put the vertices along the contour in even space from each other so the continuity of the contour and it prevents information loss because there are no vertices far from each other in comparison with other points or vertices.

Length of line segment $Sc = \sqrt{(X_{c+1} - X_c)^2 + (Y_{c+1} - Y_c)^2}$ $(Eq.8)$

Gradient energy, E_{grad} is represented by the negative of the gradient magnitude. (33) To find the image gradient magnitude first partial derivatives in horizontal and vertical directions should be found and then calculating the magnitude of the partial derivative vector. G_c is the gradient magnitude of the image, I is the image and I(x , y) is the intensity of the pixel at location (x , y) in the image.

x-1 , y-1	x , y-1	x+1 , y-1
x-1 , y	x , y	x+1 , y
x-1 , y+1	x , y+1	x+1 , y+1

Figure 9 partial derivatives

If Figure 9 shows the intensities of the pixels with specified locations, the horizontal and vertical gradients G_x , G_y are defined as in Equations 9 and 10 respectively.

$G_x = I(x + 1, y) - I(x - 1, y)$ (Eq.9) $G_y = I(x, y+1) - I(x, y-1)$ (Eq.10)

Magnitude of the gradient is: $\sqrt{Gx^2 + Gy^2}$

Gradient energy is: $-G_c$

As the gradient magnitude is larger in the edges, to minimize the function the gradient energy term should be negative of the gradient magnitude so minimizing this term attracts the contour to the object edges.

Balloon energy, makes the curve act like a balloon. This energy causes the curve pass through weak edges with respect to the inflation force. The normal direction of

the contour vector at vertex v(c) is the average of normal ones to two sides of vertex v(c). The balloon energy, E_{bal} is the cosine of the angle between the normal vector at vertex v(c) and the vector v′(c) − v(c). v′(c) is the coordinate of point in the neighbourhood of vertex v(c). If the w_{bal} is negative, the energy component encourages the contour to act like a balloon inflating and if it is positive, the energy guides the contour to go inside it. (33)

To minimize all these energy terms a greedy algorithm is used which was proposed first by Shah (32) . This greedy algorithm optimizes the contour iteratively; it starts with manually provided initial points and finding its way with updates to the location of points in a manner that minimizes all energy terms of all initial points on the contour (33).

2.2 Kalman Filter

Kalman filter is a recursive solution so as new measurements come to account they can be processed, it is a set of mathematical equations that uses vector algebra to minimize the mean of squared errors and estimates the state of a process. (35) Kalman filter is named after Rudolf E. Kálmán. Kalman filter is important because of its small computational requirement, recursive property and its being optimal estimator. Applications that use Kalman filter are as global positioning systems and smoothing the noisy input. Kalman filter predicts future state using past state information.

$\widehat{w}_k$ is the state vector consist of system information at the time t.

A is the state transition matrix of size n×n which applies the effects of state k-1 to state k.

Time Update (prediction)	Measurement Update (correction)
$\widehat{w}_k = A\, w_{k-1}$ $\hat{P}_k = A\, P_{k-1}\, A^T \mathrm{Q}$	$K_k = \hat{P}_k H^T (H \hat{P}_k H^T + R)^{-1}$ $w_k = \widehat{w}_k + K_k(z_k - H\widehat{w}_k)$ $P_k = (I - K_k H)\hat{P}_k$

z_k is the measurement vector.

H is the transition matrix contain measurement values that come from state vector parameters.

K is n×m matrix called the gain or blending factor.

R is the measurement error covariance.

Q is the measurement noise covariance.

$\hat{P}_k$ is a posterior error covariance of the current state estimate.

Kalman filter tries to predict state $\hat{w}_k$ of a process with a measurement z_k. As R approaches zero the Kalman gain matrix weights H^{-1} more $K_k = \frac{\hat{P}_k H^T}{H\hat{P}_k H^T + R}$ (36)

2.3 Optical Flow

The apparent motion of brightness patterns in an image sequence is called optical flow. The rate of change of spatial arrangement of objects in the image can be shown by optical flow. (37) . To visualize movement of objects in the sequence of images the approximation of the optical flow field is needed which is the movement of corresponding pixels in the image and is shown in Figure 10. The optical flow at a point can be computed using the points in its neighbourhood; because the velocity field (change in x and y direction) at a point in the image has two components but the intensity change due to motion yields one constraint. If for example in an image there is a pattern that the intensity along one of the image coordinates is constant and it changes along the other one then movement of the pattern in one direction causes change in the brightness at that point but the movement in the other direction yields no change so the movement in the direction with constant intensity cannot be measured locally. That is why the method (37) proposed is working global, calculation of every flow vector is based on the all points in the image. (37)

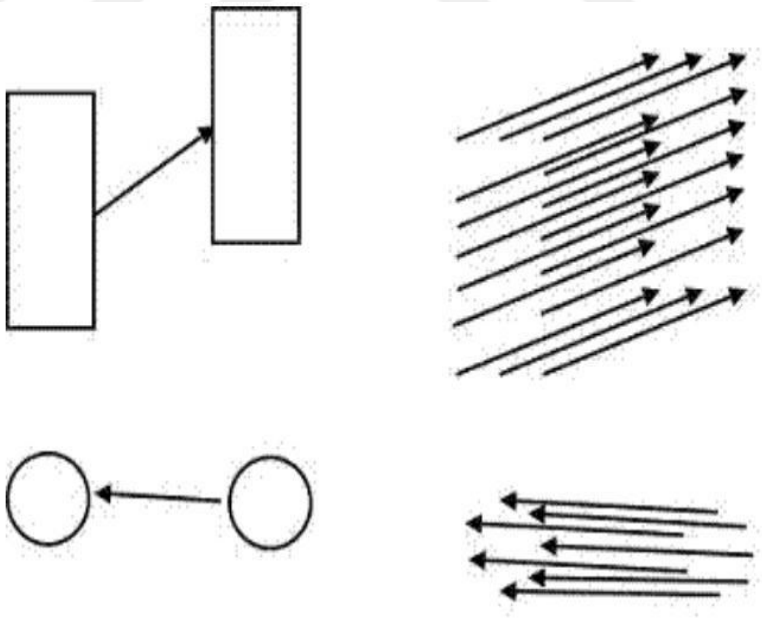


Figure 10. Representation of movement of objects and corresponding pixel movement (Original image in **(37)**)

G.Schunck (37) proposes a method to calculate optical flow. There are two assumptions in their work. First they assume that the surface of object being imaged should be flat. To guarantee preventing variation in brightness of image due to shading this assumption is made. (37) They assume that the brightness is constant; it

means the intensity value of a small region in an image remains constant over time despite of its change in position. (38)

Brightness constraint:

$E(x, y, t)$ denotes for the image brightness at point (x, y) at the time t . According to the assumption, the brightness of a point in the sequence of images being tested is constant; so

$$\frac{\partial E}{\partial t} = 0$$

The equation that shows the relation between the change of brightness at a point in the image and the motion of brightness pattern is like the equation 11.

$$\frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 \quad \text{(Eq.11)}$$

$$u = \frac{dx}{dt} \qquad v = \frac{dy}{dt}$$

By introducing E_x, E_y , E_t as the partial derivatives of image brightness in x, y, t directions respectively the above equation can be written as in Equation 12.

$$E_x u + E_y v + E_t = 0 \quad \text{(Eq.12)}$$

To estimate partial derivatives of a point in the center of a cube like the one in Figure 11. Where the column index j corresponds to x direction in the image and row index i corresponds to y direction in the image and k lies in the time direction. The following equations are used in (37).

$$E_x \approx \frac{1}{4}\{E_{i,j+1,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i+1,j,k+1}\}$$

$$E_y \approx \frac{1}{4}\{E_{i+1,j,k} - E_{i,j,k} + E_{i+1,j+1,k} - E_{i,j+1,k} + E_{i+1,j,k+1} - E_{i,j,k+1} + E_{i+1,j+1,k+1} - E_{i,j+1,k+1}\}$$

$$E_t \approx \frac{1}{4}\{E_{i,j,k+1} - E_{i,j,k} + E_{i+1,j,k+1} - E_{i+1,j,k} + E_{i,j+1,k+1} - E_{i,j+1,k} + E_{i+1,j+1,k+1} - E_{i+1,j+1,k}\}$$

(Eq.13)

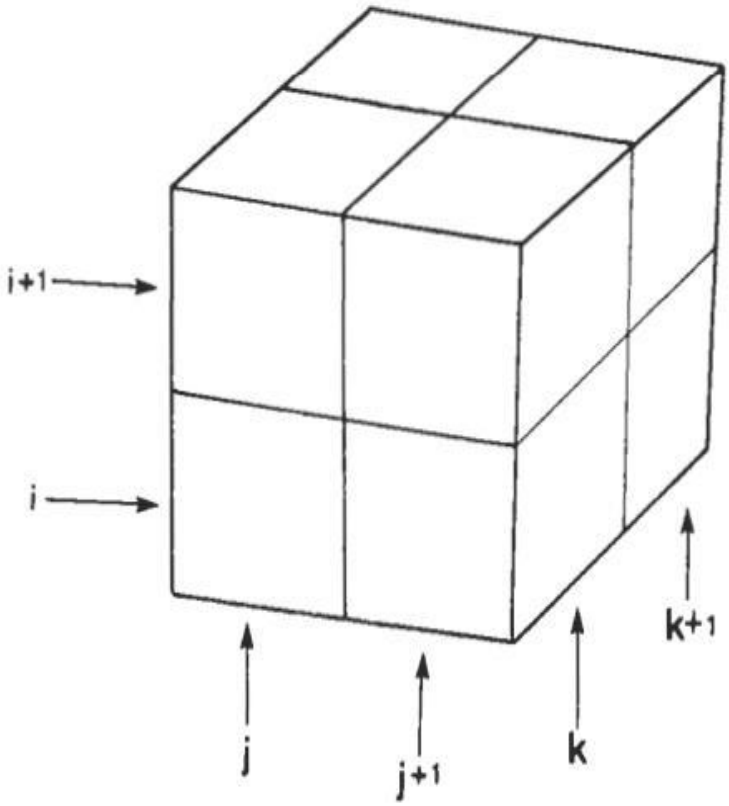


Figure 11. The cube for describing the partial derivatives (Original image in **(37)**)

The smoothness constraint:

"The spatial smoothness constraint (spatial term) comes from the observation that neighboring pixels generally belong to the same surface and so have nearly the same image motion." (38) . Actually since the neighbouring points on the object have similar velocities, the neighbouring pixels in the image also have similar velocities so their velocities vary smoothly in the image. (37) They use the sum of squares of the Laplacians of the x and y coordinates for the smoothness measurement as in Equation 14.

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \qquad \nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \qquad \text{(Eq.14)}$$

To estimate the Laplacians of u and v the following form is used in G.Schunck (37)

$$\nabla^2 u \approx k\left(\bar{u}_{i,j,k} - u_{i,j,k}\right) \quad \nabla^2 v \approx k\left(\bar{v}_{i,j,k} - v_{i,j,k}\right)$$

$$\bar{u}_{i,j,k} = \frac{1}{6}\left\{u_{i-1,j,k} + u_{i,j+1,k} + u_{i+1,j,k} + u_{i,j-1,k}\right\}$$
$$+ \frac{1}{12}\left\{u_{i-1,j-1,k} + u_{i-1,j+1,k} + u_{i+1,j+1,k} + u_{i+1,j-1,k}\right\}$$

$$\bar{v}_{i,j,k} = \frac{1}{6}\left\{ v_{i-1,j,k} + v_{i,j+1,k} + v_{i+1,j,k} + v_{i,j-1,k}\right\}$$
$$+ \frac{1}{12}\left\{v_{i-1,j-1,k} + v_{i-1,j+1,k} + v_{i+1,j+1,k} + v_{i+1,j-1,k}\right\}$$

Table 2 Laplacian

$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
$\frac{1}{6}$		$\frac{1}{6}$
$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$

The Laplacian is calculated by subtracting the value at a pixel from its neighbour points with the weights shown in Table 2.

Minimization:

To minimize sum of errors in both the brightness constraint $E_x u + E_y v + E_t$ and smoothness constraint $\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2$ two factors err_b and err_c are used as shown in Equations 15 and 16.

$$err_b = E_x u + E_y v + E_t \qquad \text{(Eq.15)}$$

$$err_c = \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 \quad \text{(Eq.16)}$$

The total error factor is $err_b + err_c$ and a weighting factor α^2 which can be chosen properly by knowing the fact that the error is proportional to the noise in the measurement.

$$(\alpha^2 + E_x^2)u + E_x E_y v = (\alpha^2 \bar{u} - E_x E_t) \qquad \text{(Eq.17)}$$

$$E_x E_y \mathrm{u} + (\alpha^2 + E_y^2)v = (\alpha^2 \bar{v} - E_y E_t) \qquad \text{(Eq.18)}$$

Then by multiplying both sides by $E_y^2 u$ in Equation 17 and $E_x^2 v$ in Equation 18 there will be

$$(\alpha^2 + E_x^2 + E_y^2)u = +\left(\alpha^2 + E_y^2\right)\bar{u} - E_x E_y \bar{v} - E_x E_t \qquad \text{(Eq.19)}$$

$$(\alpha^2 + E_x^2 + E_y^2)v = -E_x E_y \bar{u} + (\alpha^2 + E_x^2)\bar{v} - E_y E_t \qquad \text{(Eq.20)}$$

An alternate form for Equations 19 and 20 are as follows in Equations 21 and 22 which are resulted by adding and subtracting $E_x^2 \bar{u}$ from right hand side.

$$(\alpha^2 + E_x^2 + E_y^2)(u - \bar{u}) = -E_x[E_x \bar{u} + E_y \bar{v} + E_t] \qquad \text{(Eq.21)}$$

$$(\alpha^2 + E_x^2 + E_y^2)(v - \bar{v}) = -E_y[E_x\bar{u} + E_y\bar{v} + E_t] \qquad \text{(Eq.22)}$$

Now for each point in the image there are these two Equations 21, 22 but solving these equations simultaneously with standard methods will be costly so an iterative method is used as in Equations 23 and 24.

$$u^{n+1} = \bar{u}^n - E_x[E_x\bar{u}^n + E_y\bar{v}^n + E_t]/(\alpha^2 + E_x^2 + E_y^2) \qquad \text{(Eq.23)}$$

$$v^{n+1} = \bar{v}^n - E_x[E_x\bar{u}^n + E_y\bar{v}^n + E_t]/(\alpha^2 + E_x^2 + E_y^2) \qquad \text{(Eq.24)}$$

One can iterate until it stabilizes before next image frame or have a fix number for iterations in every image frame.

2.4 Proposed Methods

In Elizabeth Jurrus (16) it is proposed an algorithm that uses active contour, Kalman filter and optical flow to track axons in SBFSEM. It is used the algorithm to track axons in slices through a volume by updating position and velocity information in the model.

The algorithm of Elizabeth Jurrus (16) is used to segment mitochondria in Electron Microscopy images which are a set of 2D images in slices, Figure 12 can better show how the slices of images can be used. The aim is to segment mitochondria in a set of images. If it is assumed 3D volume as a sequence of 2D images in time then segmentation methods can be applied to the volume.

Three different methods are applied to images, the methods are as follows:

1. Active contour: In this method the first slice is initialized by the user and in the proceeding slices the contour will be the previous slice‘s contour which will be updated after some iterations of running active contour in the current method and this will be initial contour for the next slice and this will continue until the end of slices.
2. Active contour with Kalman filter: In this method the first slice is initialized by user and the proceeding slices start with the previous slice ‘s contour which will be the initial contour for the current slice, then by the aid of Kalman filter in estimating the position of each point of contour in the next slice, algorithm will continue.
3. Active contour with Kalman filter and Optical flow: In this method like the previous methods user initializes the contour for the first slice and in the next slices active contour provides position information for Kalman filter while optical flow provides velocity information for Kalman filter to predict where every point on the contour will go in the next slice.

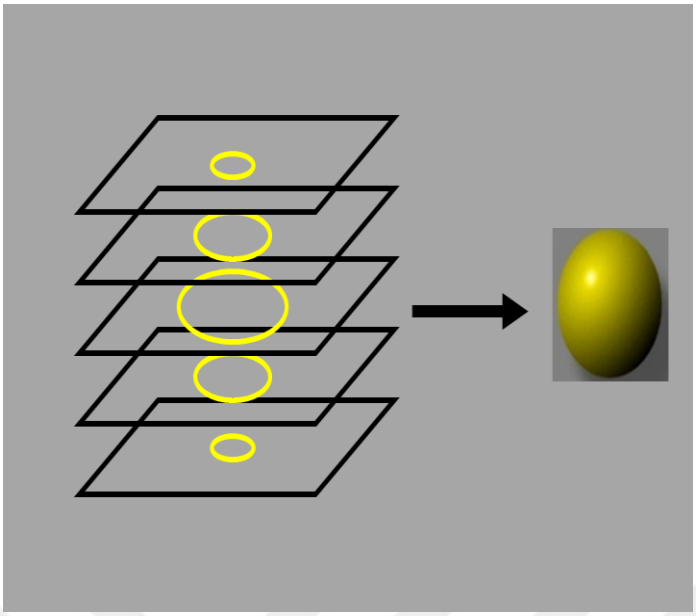


Figure 12 Slices of images which contain mitochondrion in our data sets

The algorithm is as follows, it starts with pre-processing step which is done in a work by Tasel (14), they firstly apply an auto contrast algorithm in order to have very similar contrast in different datasets and also for easy parameter settings they interpolate the images to have sizes of 2.0 nm pixel size in all data sets, this size is appropriate for both detection and segmentation of mitochondria. The last step in pre-processing is bilateral filtering which is useful in reducing the noise in the non-mitochondrial region. After pre-processing images are ready to be segmented so user can initialize contour on the first slice of images.

Pre-processing is done in 3 steps in the Tasel paper (14). These three steps are as follows: 1) Auto-contrast adjustment 2) Resampling 3) Smoothing.

(1) Auto-contrast adjustment: Brightness of an image is the overall lightness or darkness of the image and the difference of brightness in objects and regions of the image is called contrast (39). Images with good quality and contrast are considered as strong requirement in some areas like medical image analysis and for this purpose some applied methods are named image enhancement techniques which process the image and help it be visually more suitable than the original one. Automatic contrast enhancement is needed in medical images to have better visual inspection. Contrast enhancement is essential in medical imaging for better interpretation of images by making object features easier to distinguish (40), (41). In the work by Tasel et al they use auto-contrast algorithm which is needed for normalizing the gray values of the image into a certain range. They did this by setting 0.5% of the lowest gray value of the histogram to the minimum gray value and setting 0.5% of the highest gray value of the histogram to the maximum gray value. The other pixel values are normalized between minimum and maximum values.

(2) Resampling: In the second step of pre-processing they interpolate the images to have the fixed pixel size for all the images of all 8 data sets. This is done to have

easier parameter tuning. The pixel size is set to 2.0 nm. As the mitochondrial membranes are 4-6 nm apart, 1-3 nm pixel size might be considered a good range. Choosing less than 1nm/pixel will cause smaller imaged area than optimal for typical mitochondrial sizes. Choosing larger than 3nm/pixel will cause membrane resolution loss.

(3) Smoothing: Generally all medical images contain some visual noise. The noise lowers quality of images. To reduce noise level in images they have tested three different methods, Gaussian filtering, Anisotropic diffusion and Bilateral filtering. a) *Gaussian Filtering*: is the convolution of Gaussian kernel with the image. b) *Anisotropic diffusion*: is a type of smoothing that preserves edges. The amount of smoothing in different regions of image is determined by the magnitude of gradients of the image. In the edges that magnitude of gradient is high; diffusion is weak so smoothing is less. In the areas with low magnitude of gradient there is a strong diffusion which causes high amount of smoothing. c) Bilateral filtering: By means of Bilateral filtering images can be smoothed with edges preserved, with the aid of nonlinear combination of nearby image values. Bilateral filter uses two weighting functions, domain weights and range weights. It actually operates in both the domain and the range of the image. The main idea in Bilateral filtering is that it smoothes the image in the domain of the image and it does not smooth the image when the pixels are far from each other in the range of the image, in other words it smoothes the image when pixels are similar to each other but does not smooth when the pixels are not similar to each other. The simplest similarity factor that can be considered in the range of the image can be intensity of the image. The definition of Bilateral filter is given in the Equation 26.

$$K_s = \sum_{p\in\Omega} f(p-s)g\left(I_p - I_s\right) \qquad \text{(Eq.25)}$$

$$J_s = \frac{1}{K_s}\sum_{p\in\Omega} f(p-s)g(I_p - I_s)I_p \qquad \text{(Eq.26)}$$

In the Equation 26, Js is the filtered pixel intensity, p is coordinate of current pixel and s is coordinate of center pixel of kernel and Ω is set of all pixel coordinates in the local neighbourhood in the kernel. $f(p-s)$ is used to measure geometric distance between two points of p and s . $g\left(I_p - I_s\right)$ is used to measure photometric similarity between $I_p$and I_s.

Domain Weighting:

Usually a standard Gaussian filter is used for this purpose so $f(p-s)$ can be defined as:

$$f(p-s) = e^{-\frac{d(p-s)^2}{2\sigma_d^2}} \qquad \text{(Eq. 27)}$$

$$d(p-s) = \|p-s\| = \sqrt{p^2+s^2} \qquad \text{(Eq. 28)}$$

Where $d(p-s)$ is the Euclidean distance between p and s.

Range Weighting:

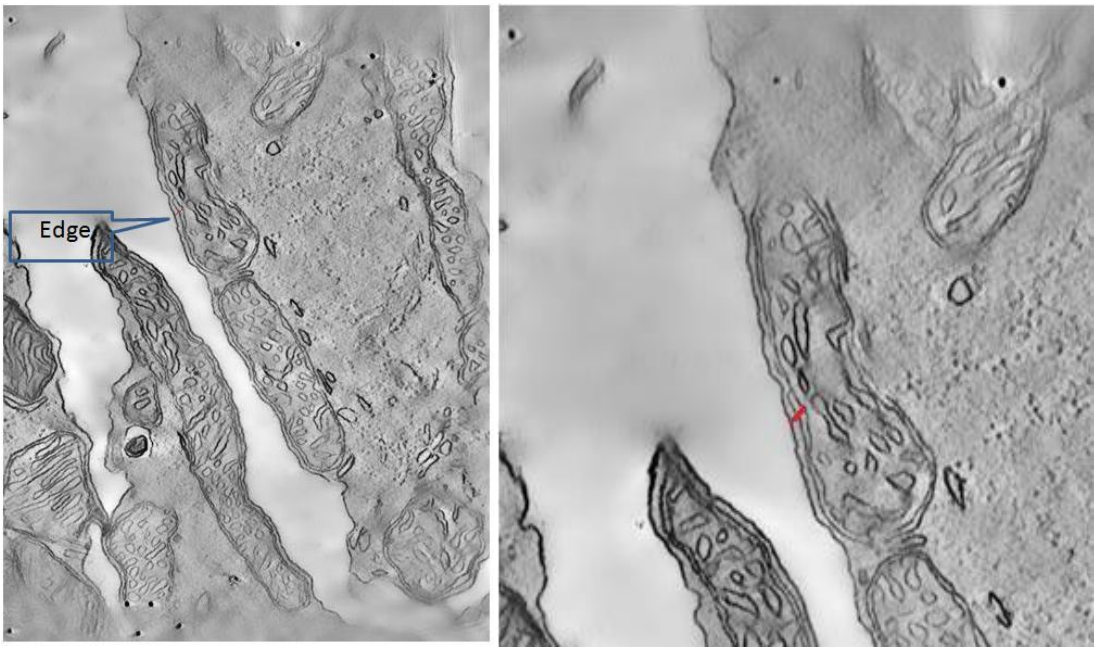
Usually the difference of intensity values is calculated by a Guassian function. As given in Equation 29.

$$g(I_p - I_s) = e^{-\frac{\delta(I_p - I_s)}{2\delta_r^2}} \qquad \text{(Eq.29)}$$

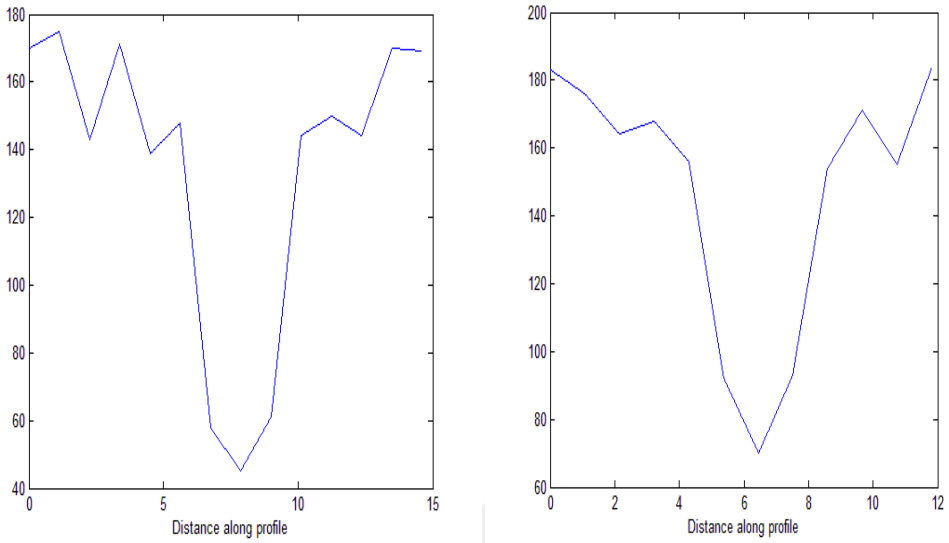
Where $(I_p - I_s) = \|I_p - I_s\| \ . \|I_p - I_s\|$, the difference between two pixels p and s, always this difference is the intensity value differences of two pixels p and s $\|I_p - I_s\| = I_p - I_s$. (42)

In the work by Tasel (14) there is a comparison between three methods of smoothing the Gaussian, Anisotropic diffusion and Bilateral filtering which shows that Bilateral filtering causes best results than the other methods tested so it has been tested on the images which are smoothed by bilateral filtering.

The data set images have huge sizes; so in this study, it is done down sampling on the images and the result is studied and it is shown that the edges are not influenced so much and the edges are not vanished. Before down sampling, it is applied a Gaussian filter with standard deviation value of 0.5 and the down sample is done with the factor of 2 which is a suitable factor for down sampling here because the edges are not so influenced and this is shown in Figure 13. Down sampling is a process in which the number of pixels of an image is reduced by removing the pixels. A low pass filter like Gaussian blurring is usually needed when down sampling the image and reducing the size of image. The Gaussian filter prevents aliasing which can be the result of down sampling. Aliasing is loss of some high frequency information in the image.



(a) (b)



(c) (d)

Figure 13 - Location of the edge tested and profile of it before and after down sample, (a) is the location of the edge ,(b) is the location of the edge zoom out (c) profile of the selected edge before down sample (d)Profile of the edge after down sample with factor 2

2.4.1 Method 1: active contour

In this work it is better to place initial contour on the target contour then iteratively the snake will be attracted toward target contour by forces that control the shape and location of the snake in the image. The mentioned forces are internal and external forces. Internal forces control the smoothness of contour where external forces guide contour toward edges.

The contour is represented by vertices of a polygon. The vertices are represented by v(c) and defined as $v(c) = (x_c, y_c), c = 1, \dots, N$ and the polygon is shown in Figure 14.

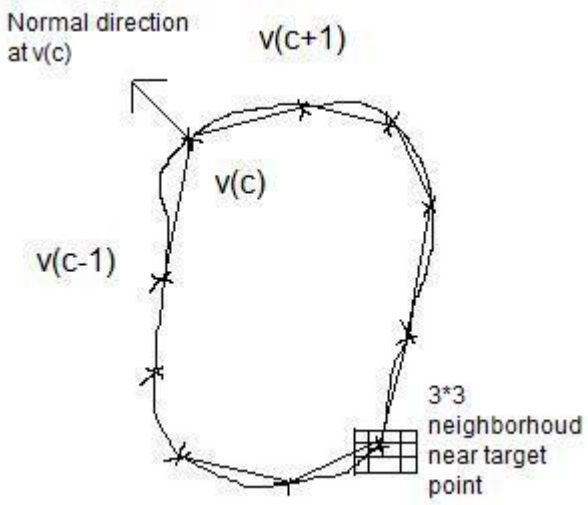


Figure 14 The contour and vertices of polygon on it

In this work, it is assumed a 3×3 neighbourhood near every point on the contour and in calculations of energies in Active contour method the minimum energies of all neighbourhood points are calculated and the point with minimum energy will be the final position of the point in every iteration, Here it is tested and concluded that 3 iterations is enough to segment mitochondria properly.

The energy which is to be minimized is defined in Equation 30.

$$\mathrm{E} = \sum_{c=1}^{N}[\, \mathrm{w_{curv}}\, \mathrm{E_{curv}} + \mathrm{w_{con}}\, \mathrm{E_{con}} + \mathrm{w_{grad}}\, \mathrm{E_{grad}} + \mathrm{w_{bal}}\, \mathrm{E_{bal}}] \quad \text{(Eq.30)}$$

E_{curv} is the curvature energy and is defined as $E_{curv} = ||v(c-1) - 2v(c) + v(c+1)||$ and it is calculated for all points in the neighbourhood. This energy term helps the contour to have a smooth shape and prevents sharp corners.

E_{con} is the continuity energy and defined as E_{cont}=$|\ S_c - \bar{S}\ |$ where $\mathrm{Sc} = \sqrt{(\mathrm{X_{c+1}} - \mathrm{X_c})^2 + (\mathrm{Y_{c+1}} - \mathrm{Y_c})^2}$ and $\bar{S} = \sum \frac{S_c}{N}$, N is number of points on contour. This term maintains regular spaces between points on the contour. Here S_c will be length of line segment from every point in the 3×3 mask to the next point on the contour.

E_{grad} is the gradient energy term and is defined as $E_{grad} = -|\ G_c\ |$ where G_c is the gradient magnitude that is defined as $G_c = \sqrt{{G_x}^2 + {G_y}^2}$ where

$G_x = I(x+1, y) - I(x-1, y)$, $G_y = I(x, y+1) - I(x, y-1)$ and $I(x, y)$ is the image intensity in position (x, y) of the image. G_x *and* G_y stand for horizontal and vertical partial derivatives. For all points in the mask the gradient magnitude is calculated and for the pixels in the border of the image, padding will be done there. Table 3 can help in understanding it.

Table 3 The mask used to calculate gradient magnitude

(x-2 , y-2)	(x-1 , y-2)	(x, y-2)	(x+1 , y-2)	(x+2 , y-2)
(x-2 , y-1)	(x-1 , y-1)	(x , y-1)	(x+1 , y-1)	(x+2 , y-1)
(x-2 , y)	(x-1 , y)	(x , y)	(x+1 , y)	(x+2 , y)
(x-2 , y+1)	(x-1 , y+1)	(x , y+1)	(x+1 , y+1)	(x+2 , y+1)
(x-2 , y+2)	(x-1 , y+2)	(x, y+2)	(x+1 , y+2)	(x+2 , y+2)

Minimizing this term guides the contour toward edges.

E_{bal} is balloon energy term and is defined as cosine of the angle between normal vector at vertex $v(c)$ and the vector $v'(c) - v(c)$. The normal direction to the contour at vertex $v(c)$ is defined as the average of the normals to the two sides of the vertex $v(c)$.

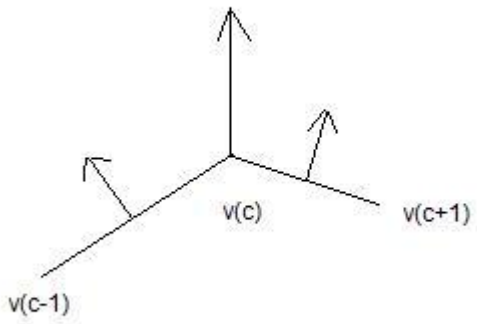


Figure 15 Normal vector at vertex v(c)

Here to find the normal to the two sides of polygon first the vector connecting two vertices $v(c), v(c-1)$ and the vector connecting $v(c), v(c+1)$ will be found as in Figure 15. Then for each of these vectors, finding a point in the middle of these and the two perpendicular vectors to this point is needed. The two perpendicular vectors to this point are $\overrightarrow{k_n} = \begin{bmatrix} -k_x \\ k_y \end{bmatrix}$ and $\begin{bmatrix} k_y \\ -k_x \end{bmatrix}$. Finding the vectors from these middle points to the center point which coordinates have been provided by the user clicks is also needed. Finally to know which perpendicular vector is the vector, pointing out of the contour, the cosine of the angle between perpendicular vector and the vector toward center will help, because the vector with cosine of positive value will be the vector pointing out of contour. It is needed to determine the normal pointing outside of contour; because the contour is going to get greater. Actually other energy terms will control contour also not to get much greater or greater than needed. Figure 16 well describes this process.

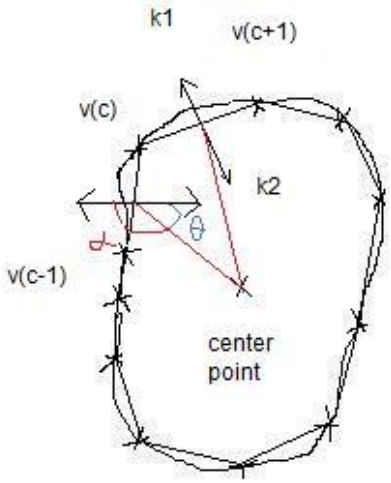


Figure 16 Normal vectors

In the first method in this thesis just Active contour is used to segment mitochondria in dataset images. In the first slice of images the user initializes contour but in the next slices result of previous slice will be the initial contour then by running Active contour 3 iterations iteratively in our case the final contour for the current slice will be resulted and will pass to the next slice as the initial contour.

2.4.2 Method2: Kalman filter with active contour

In this method like the previous one first the slice is initialized by user but for the following slices Kalman filter with the aid of active contour estimates where the contour will go.

Kalman filter estimates the location of mitochondria in the next slice by sampling the image and then correcting the estimation. The Mitochondria are represented by a set of points. These points have their own Kalman filter updating the state $w_k = [\, x_k\, , y_k, u_k, v_k\,]^T$ where $[\, x_k\, , y_k]$ is the position of the point on the contour and $[\, u_k\, , v_k]$ is the velocity at the point. Kalman filter consists of three computations: $\widehat{w}_k$ prediction, z_k the measurement, and w_k the correction.

$\widehat{\boldsymbol{w}}_{\boldsymbol{k}}$ **:** To calculate current slice's prediction state previous state's correction state is used as shown $\widehat{w}_k = A\, w_{k-1}$ where $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$, velocity value is assumed to be constant and matrix A helps us assume that but position of point in the next slice will be computed using current position. Third and fourth rows of matrix A help us have constant velocity values.

$\boldsymbol{z}_{\boldsymbol{k}}$ **:** In the measurement state z_k the position information comes from active contour and velocities come from the difference of positions in previous slice and current slice.

$\boldsymbol{w_k}$ **:** Kalman filter combines the predicted state $\hat{w}_k$ and measured state z_k to have corrected state w_k, the final aim is to find this corrected state.

$w_k = \hat{w}_k + K_k(z_k - H\hat{w}_k)$ Where K_k is the Kalman gain matrix and is defined as $K_k = \hat{P}_k H^T (H\hat{P}_k H^T + R)^{-1}$ where $\hat{P}_k$ is a posterior error covariance of the current state estimate, R is measurement noise covariance and it is a 4×4 diagonal matrix. H is n×n matrix that relates the state to measurement z_k. H might be of changeable size in practice but here in this model it is assumed constant and is identity matrix of size 4.

To define $\hat{P}_k$ it is needed to define the a priori and posterior estimate errors e_k and $\hat{e}_k$ as $\hat{e}_k \equiv w_k - \hat{w}_k$ and $e_k \equiv w_k - \hat{w}_k$ where $\hat{w}_k$ is the a priori state estimate at step k giving knowledge about process prior to step k and w_k is the a posterior state estimate at step k given z_k the measurement state. The a posteriori estimate error covariance is $\hat{P}_k = E[\hat{e}_k \hat{e}_k{}^T]$ and the a priori estimate error covariance is $P_k = E[e_k e_k^T]$. (35) $\hat{P}_k = AP_{k-1}A^T Q$ where Q is a 4×4 diagonal matrix which represents process noise covariance and here it is assumed it is constant. K_k is chosen so that minimizes P_k. P_k is updated after each estimation by $P_k = (1 - K_k H)\hat{P}_k$. H defines the relationship between the measurement and the model. As mentioned earlier Q is assumed to be constant but the measurement noise can modelled like in the following form (16). R can be modelled at a contour point using membrane strength metric. The strength of a membrane is defined as follows and it is defined as the second derivative in perpendicular direction to membrane $\mu = \frac{d^2}{d\vec{n}^2} I'$ where I' is the intensity along vector n which is perpendicular to membrane point. A box filter is also used to smooth the noise and is oriented in the direction d perpendicular to vector n. It is best illustrated in the Figure 17.

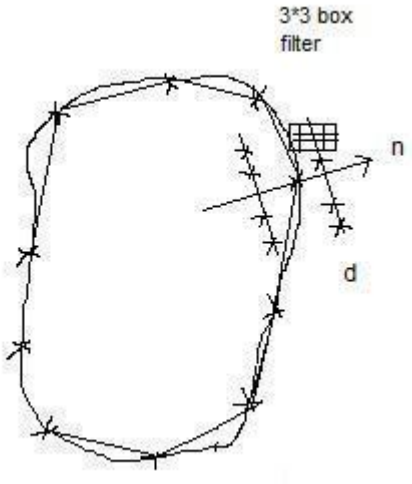


Figure 17 shows the directions to membrane

To have intensity values of image in the positions which 3×3 mask is applied, a linear interpolation is done to get the pixel intensities. Linear interpolation is

calculated as follows but for the sake of better running time it should be forgotten. Figure 18 shows how it can be calculated.

$$Q = p(1-a) + ra$$

$$Z = s(1-a) + ta$$

$$V = Q(1-b) + Za$$

$$0 \leq a < 1\,, 0 \leq b < 1$$

(Eq.31)

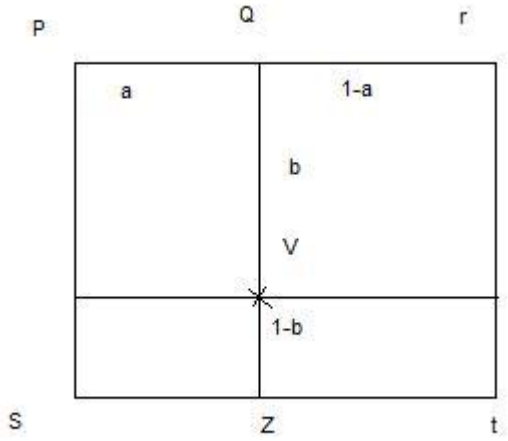


Figure 18 shows how linear interpolation is calculated

To scale membrane strength to a value between 0 and 1 a calculation is done to have $\mu' = 1 - e^{(-\frac{\mu^2}{c})}$ where c is constant value defining a strong membrane.

2.4.3 Method 3: Kalman filter with active contour and optical flow

In this method like the other two methods first slice is initialized by user clicks and the initial contour is provided by the user but Kalman filter with the aid of active contour and Optical flow will estimate position of points on the contour in the next slices.

Kalman filter works as in the method2 with a difference in that the measurement state z_k, the position information comes from active contour and velocity information comes from optical flow.

Motion is one of intrinsic properties of the world and it is a part of our visual experience. Each point on a 3D surface moves along a 3D path in camera-centered coordinates but when projected in the image plane each point produces 2D path $\vec{x}(t) \equiv (\mathrm{x(t)}, \mathrm{y(t)})^{\mathrm{T}}$ with the direction of $\frac{d\vec{x}(t)}{d(t)}$ which is also velocity. The 2D

velocity for points of a surface is called the 2D motion field. An approximation to the motion field is the goal of optical flow. Optical flow is the result of relative motion of the object and the viewer. Here in this case as mitochondria are dynamic and change the shape in every slice of data set images; so there is an optical flow to measure. Figure 19 can best illustrate what is optical flow.

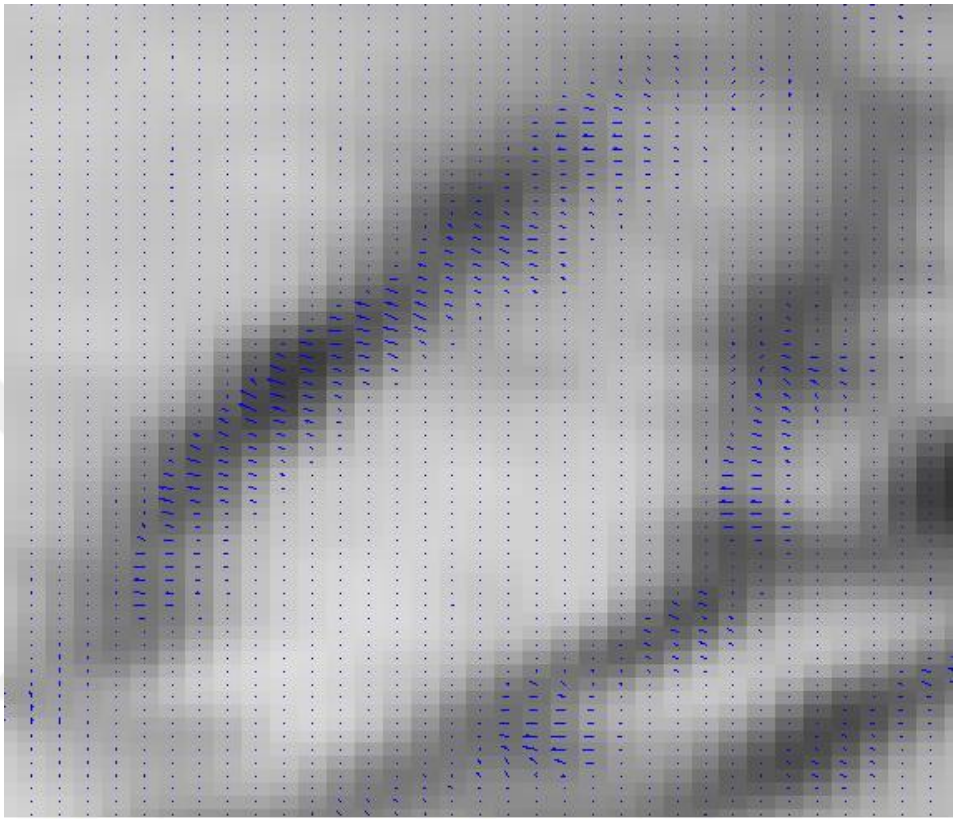
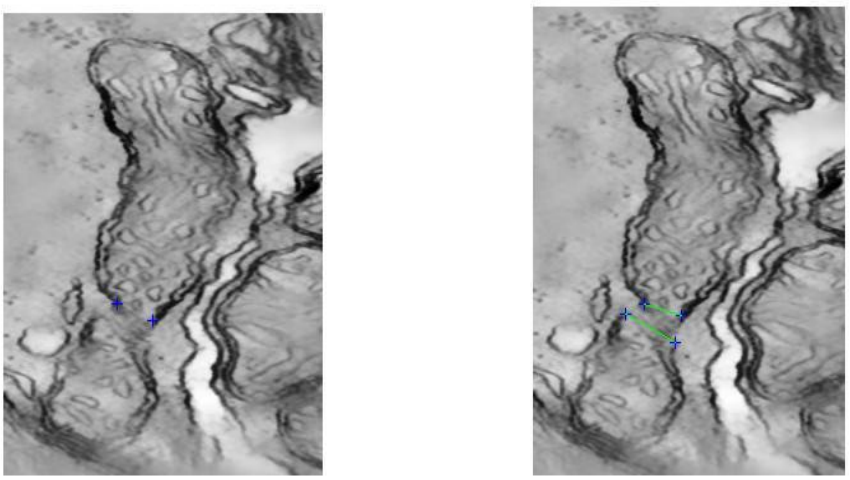


Figure 19 Optical Flow shown in a part of image with motion

2.4.4 Merging, Splitting, Vanishing

There are some mitochondria that start as one mitochondrion but after some slices split and become two mitochondria, this is named splitting. In the automatic method, we try to detect the slice number in which splitting occurs if there is any with two rules, distance check rules and edge strength rules. But in some cases, when there is no splitting, the algorithm detects wrong splitting. In the semi-automatic method, there is an option in the application that lets the user state the slice number in which the mitochondrion is going to split. Figure 20 can best illustrate splitting mitochondrion.



(a) (b)

Figure 20 splitting mitochondrion with its splitting points (a) the two splitting points of one mitochondrion (b) all splitting points

After determining splitting points, active contour is executed for 10 iterations, because in less iterations the contour cannot find its way and the result will not be satisfying. This 10-iterations execution of active contour is just in the slice where splitting occurs.

2.4.4.1 Automatic Splitting

In this method, the distance of every point at the contour from all other points is calculated in every slice, then from one slice to another slice, the difference of these distances is calculated. Until 10^{th} slice, the difference is the difference between two consequent slices, but after 10^{th} slice, the difference of one slice and the 10 slices before is calculated. Actually, we want to find if the distances between points are changing so much in comparison with 10 slices before the current slice. In the algorithm, we try to find two points which are far from each other in one slice and become near to each other after 10 slices. If there is any two points with this criterion, the edge strength check is needed to find out if splitting points are correctly detected. The edge strength check is somehow similar to distance check in this way, that it compares the gradient values of the points on the splitting line with respect to 10 slices before current slice. The splitting region has lower gradient magnitudes and this can help us define the edge strength check. From distance check, we have a line between splitting points so if we check the gradient magnitudes of the points and a 5×5 neighborhood of the points on the line, and compare these gradients with 10 slices before, we can find out if this region's gradient magnitude difference has become higher, in comparison to 10 slices before or not; if the difference is higher than some threshold the detected points are considered as splitting points. But as the detected splitting points might not be the real ones, or not very near to real ones, then

the gradient of the splitting line also may not help us much. After determination of two splitting points by the algorithm, a line is drawn between these two points. The angle between the splitting line and vertical line and the angle between the splitting line and horizontal line are calculated. The smaller angle is determined. With this way it can be inferred that the mitochondrion is in x direction or in y direction. The horizontal and vertical lines are displayed in the Figure 21. If the mitochondrion is vertical, then the points on the contour with y values less than y value of the points on the line, are gathered in one mitochondrion. Another mitochondrion will have the points with y values greater than line's y values. If the mitochondrion is horizontal, then x values are compared. This is how one mitochondrion splits and become two separate mitochondria.

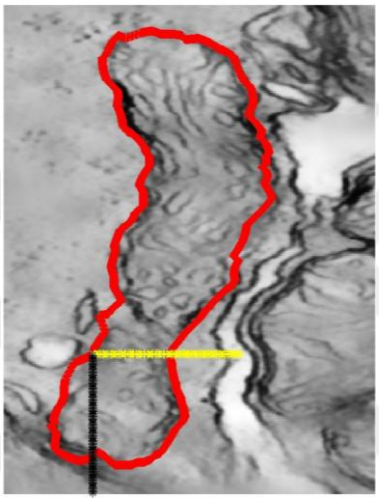
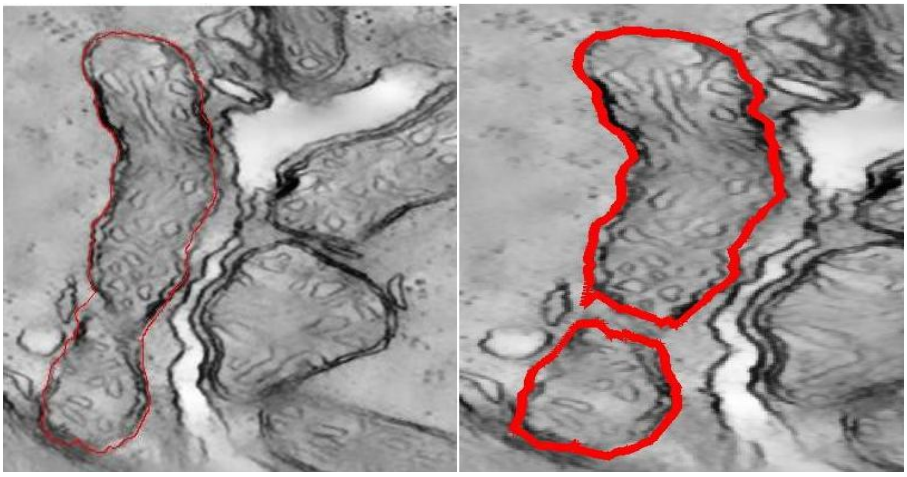


Figure 21 Horizontal and vertical lines in the splitting region

After splitting is completed, the active contour is executed 10 iterations just for the splitting slice, after that, the program continues normally. This method for the mitochondria with splitting, works not bad, but for the mitochondria which do not split, it may find some false splitting points. Semi-automatic method can help us segment mitochondria more exactly in the case of splitting mitochondria.

2.4.4.2 Semi-automatic splitting

As stated before in the automatic splitting section, there might be wrong segmentations, but in the semi-automatic method, that is not the case. In this method, the user states the slice number in which the splitting occurs, after execution, the program pauses in the splitting slice and waits for the user to specify the splitting points. In this method, these splitting points are considered to be four points, two for every mitochondrion that will be the result of splitting as shown in Figure 20. After determination of four splitting points, the points between these points are removed. Then every two points on every splitting mitochondrion are connected to each other and the algorithm continue with two separate mitochondrion instead of one mitochondrion. This is an option in the application that more than one mitochondrion can be segmented. In the last step to complete the splitting, the algorithm continues with 10 iterations of active contour. A sample of splitting mitochondrion before and after splitting is presented in Figure 22.



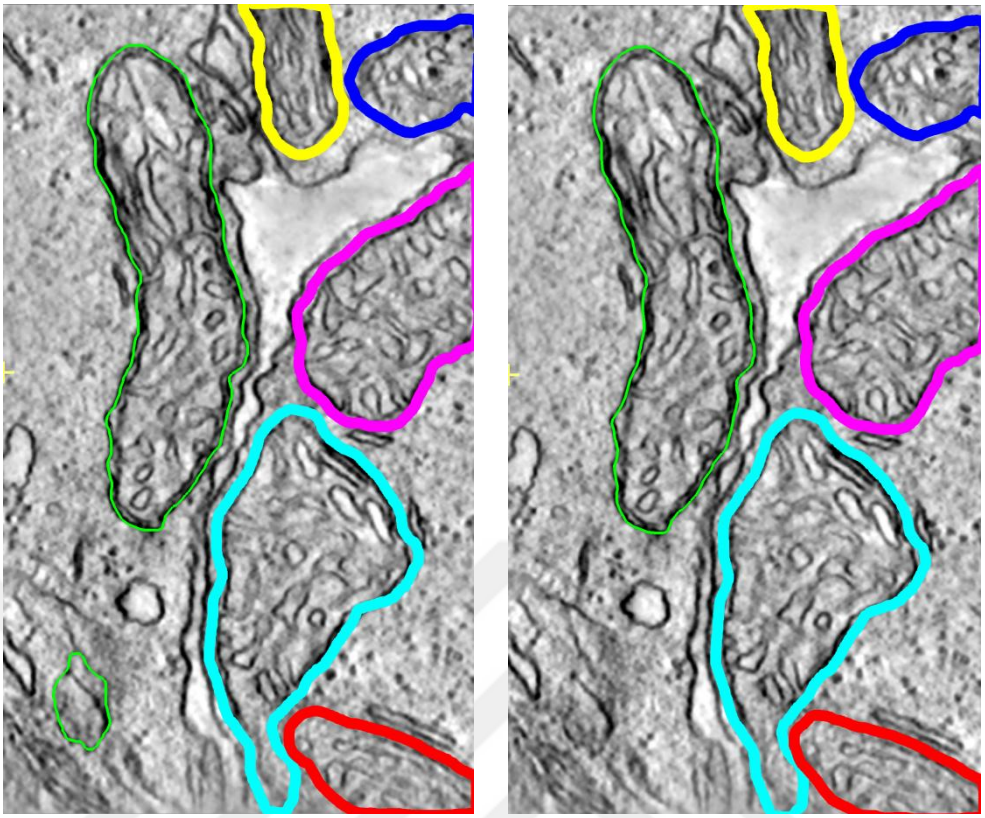
(a) (b)

Figure 22 Splitting mitochondria (a) before splitting (b) after splitting

2.4.4.3 Vanishing

In some of data sets there are mitochondria which vanish after some slices. Figure 23 presents a mitochondrion before and after it disappears. In the images that are manually segmented, the disappearing mitochondrion loses its edges little by little after some slices, and finally it disappears; but in our algorithm, in all three methods, the disappearing mitochondria does not get smaller so much because it sticks to edges and higher gradients, so, we should find some way to recognize disappearing mitochondria.

If we want to automatically determine that a mitochondrion is going to disappear after some slices , in the case of disappearing mitochondria, it will work properly, but there are some data sets in which the membranes of mitochondria are not so much distinguishable. In other words the gradient magnitudes in the edges of mitochondria are also low in comparison to other data sets, so, in these types of data sets, this algorithm will cause an existing mitochondrion to disappear wrongly. In the automatic method, we will check gradient magnitude of all points on the contour, and if more than fifty percent of points on the contour have a gradient magnitude less than a threshold that mitochondrion is detected as disappearing mitochondrion. User interaction can help us fix it, the application provides the user an option to enter the slice number in which the mitochondrion is going to disappear. If there are more than one mitochondrion in the program to be segmented and one of them disappears, then having the slice number, algorithm can decide which one is going to disappear in this way. It calculates the gradient magnitude of all points on the contours of all mitochondria in use and the mitochondrion with less value for mean gradient magnitudes of points on the contour will disappear in the next slice.



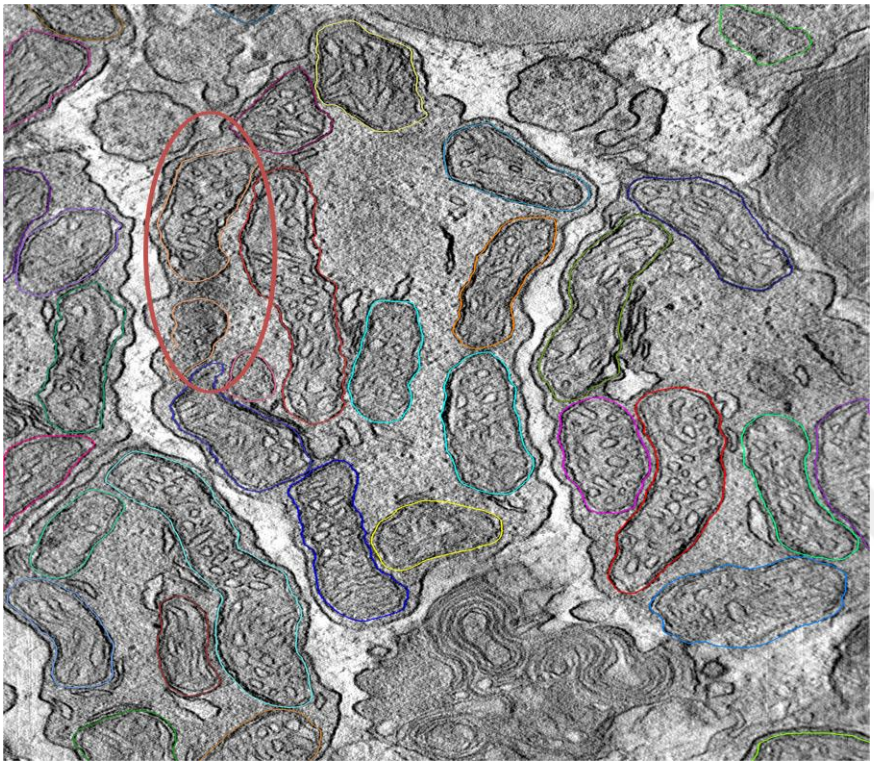
(a) (b)

Figure 23 Disappearing mitochondrion in the lower bottom corner (a) before disappearing and (b) after disappearing

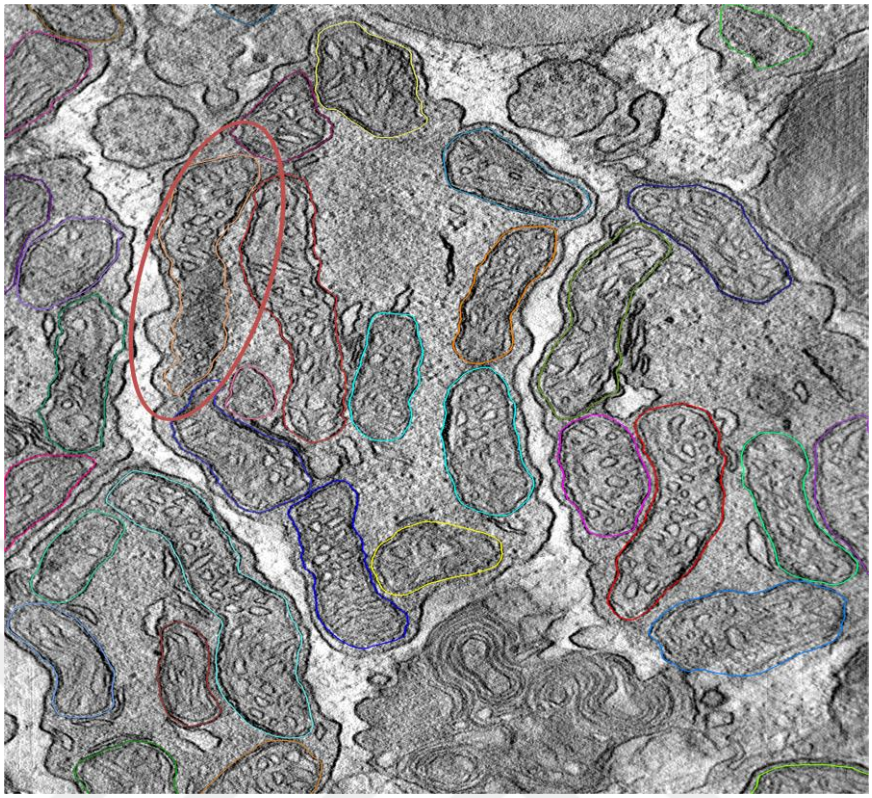
2.4.4.4 Merging

There are cases when two mitochondria merge with each other after some slices and become one mitochondrion instead of two separate mitochondria. This can be best illustrated in Figure 24. In the application implemented in this thesis there is an option for the user to enter the slice number in which the two mitochondria are going to merge. Of course it should be considered that the two mitochondria should be segmented simultaneously in the application. This is also an option in the application for this purpose. If the user wants, he/she can segment more than one mitochondrion at the same time. Merging just can occur if more than one mitochondrion is segmenting in the application. In the slice number of merging the points on the contour of two merging mitochondria that are very close to each other than other points on the contour of both mitochondria are going to disappear and the mitochondria would merge. As it can be seen in the two merging mitochondria in Figure 24, the merging points from two mitochondria are not very close to each other. In the merging algorithm in this thesis, as the nearest points from two merging mitochondria, which might be in every probable distance, (here it is assumed that they can be near to each other 10 pixels or less), become near to each other, the nearest points merge with each other and become one mitochondrion. The process of merging starts with the execution of merging slice number (specified by the user). It is assumed that the points from two mitochondria which have distance less than 10 pixels are in the merging region. The points in the merging region are removed from two merging mitochondria. Remained points on the contour of two mitochondria are

investigated and four points on the contour of two mitochondria are selected. Two of the points are from one mitochondrion which are the nearest two points to the other two points from the other mitochondrion. After that the four points are specified, in the distance of every connecting two points from two mitochondria, some other points connecting these points are selected with a distance of 0.1 pixel from each other. Connecting these points will merge the two mitochondria. In the last step after merging active contour is executed 10 iterations.



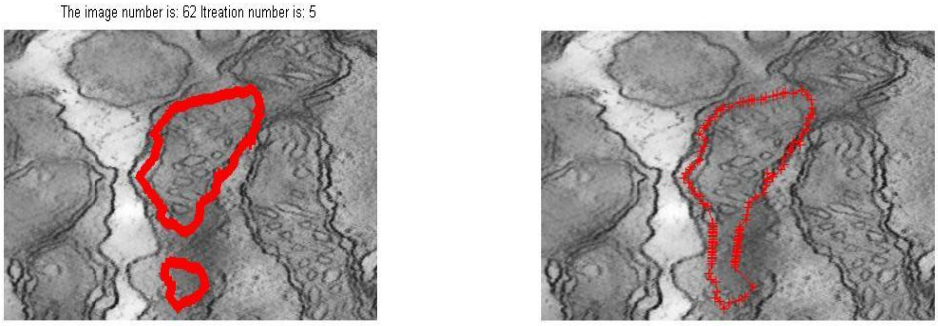
(a)



(b)

Figure 24 Merging mitochondrion (a) before merging (b) after merging

The result of this merging mitochondria, is shown in Figure 25.



(a) (b)

Figure 25 Merging mitochondria (a) before merging (b) after merging

2.5 Metrics to Evaluate Segmentation Performance

To measure the performance of the algorithm, visually comparing the output of the algorithm with the ground truth is not enough; there should be some metrics to measure the performance accurately. Ground truth is a data set image in which specialists segment mitochondrion manually using IMOD application which is developed to make easier viewing of 3D image data of biological structures. (43) A sample of ground truth image in IMOD is presented in Figure 26.

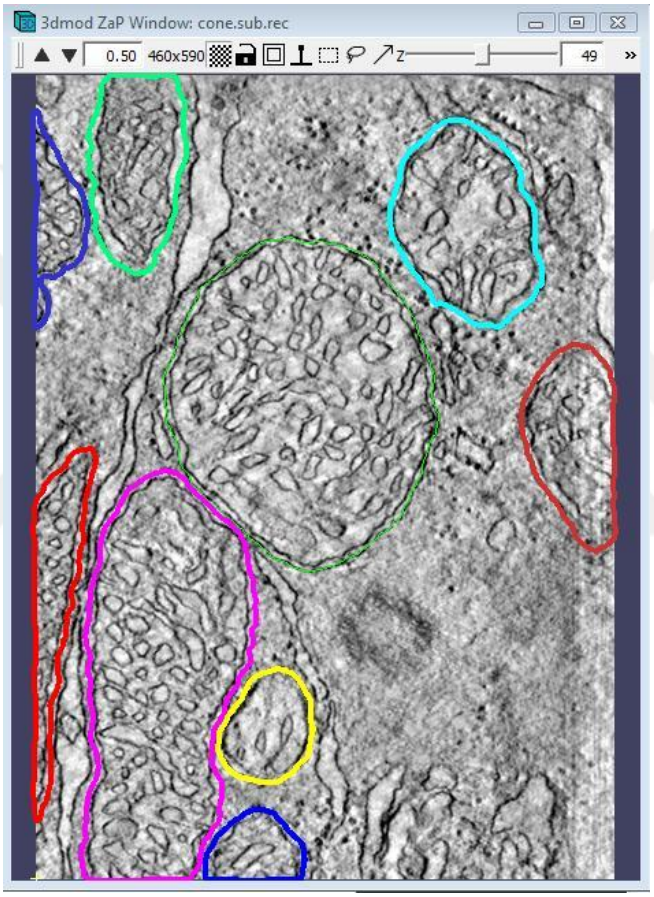


Figure 26 Sample image of Ground Truth in IMOD

There are some metrics to evaluate the contour accuracy and some of them will be discussed here. (44)

Dice coefficient is a statistic to compare two samples to know how similar they are. It is also known as similarity measure. The two samples are ground truth contour and segmented contour by our algorithm. Dice coefficient measures the ratio of the intersection and union of two samples and is defined as in Equation 32. Where S stands for the segmented contour and G stands for ground truth contour.

$$DSC(G,S) = \frac{2|S \cap G|}{|G| + |S|} \qquad (Eq.32)$$

Average Symmetric Surface Distance (ASSD) is another metric to evaluate contour accuracy. In this metric first of all border pixels of the two sets should be obtained. 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 averaged. ASSD is defined as in Equation 33.

$$ASSD(\hat{G},\hat{S}) = \frac{1}{n}\sum(\{d|d = min_j\|\hat{S}(i) - \hat{G}(j)\|, i = 1,2,\dots,N\} \cup \{d|d = min_i\|\hat{S}(i) - \hat{G}(j)\|, j = 1,2,\dots,N\}) \qquad (Eq.33)$$

Where $\hat{S}(i)$ and $\hat{G}(j)$ are segmented contour and ground truth contour's boundary points, N and M are number of points on the boundary of ground truth and segmented contour.

The Root Mean Square Symmetric Surface Distance (RMSSSD) is similar metric to ASSD with the difference that in RMSSSD the distances are averaged and then are squared.

Maximum Symmetric Surface Distance (MSSD) is another metric which returns the maximum of distance calculated in ASSD.

Here the two metrics Dice coefficient and RMSSSD are calculated to evaluate the segmentation accuracy. Dice coefficient is used because it is comparing the area of the two segmented and ground truth contours. RMSSSD is selected among other metrics because ASSD will have similar results as Dice coefficient. But RMSSSD will give some information about the boundary error of the segmented contour.

2.6 Setting Parameters

There are parameters in this work that need to be set to proper values. First of all the mask size needed for active contour energy terms calculation is set to be 3×3 and not larger because if it is chosen the larger value for it, unrelated pixels come into account and the segmentation will be very unrelated.

Weights of energy terms in active contour are set to appropriate values to segment mitochondria properly. These weight terms are used to balance the relative influence of energy terms. The weights relative sizes rather than absolute sizes are significant. Here it is not chosen the greater values for the weight of Balloon energy term; because in this work initial points of contour should be near the target contour and the contour does not need to get greater in every iteration. The other weights of energy terms should be in balance with each other to have better segmentation. For example if the weight of gradient energy term is notably larger than other weights, the point will hold on to a point with strong gradient and segmentation will be worse. The weights are set to be as follows: $w_{curv} = 1.6$,$w_{con} = 4$, $w_{grad} = 1.8$, $w_{bal} = 0.014$. A 3×3 neighbourhood near target point on the contour is considered. Three number of iterations for the active contour is considered enough for a good segmentation.

For Kalman filter there are some parameters that have to be set properly so the algorithm performs better. Q matrix is a diagonal 4×4 matrix with 0.6 in diagonal values. H matrix is identity matrix of size 4. Value of c=4.0 is better because this value causes higher value for μ in strong edges. P_{k-1} is identity matrix of size 4×4 in most of the examples of Kalman filter. Therefore, this is set to be identity. The distance of points to calculate the membrane strength for R matrix is set to one pixel; because more than that will cause loss of information near the point of interest. The box filter used for calculation of R is set to be of size 3×3. W_{k-1} is set to be the contour points which are calculated by active contour.

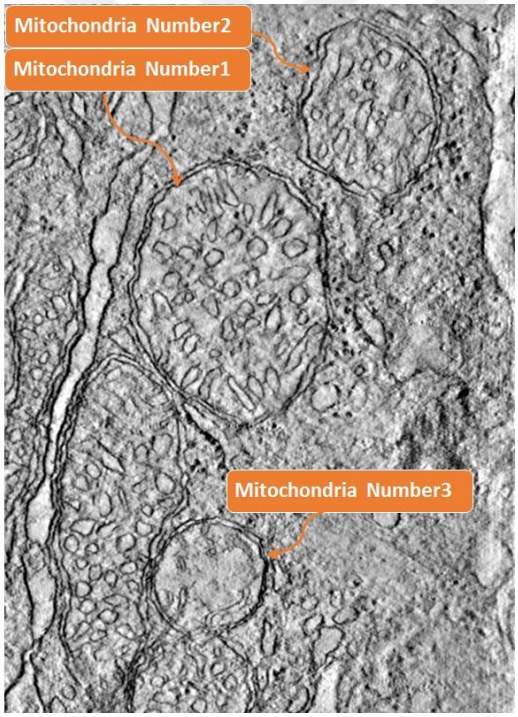
The parameter for optical flow is $\propto^2$ since it is iterative approach it is better to start with larger $\propto^2$ and then decreasing its value to achieve greater robustness. By testing different values for $\propto^2$ it is decided the value of it when segmentation is better than the other segmentations with other values of $\propto^2$. The value of $\propto$ is set to be 15. The other parameter that should be taken into account is iteration number in optical flow which is set to 20 iterations because after this number of iterations there are no more changes in the result of segmentation.

CHAPTER 3

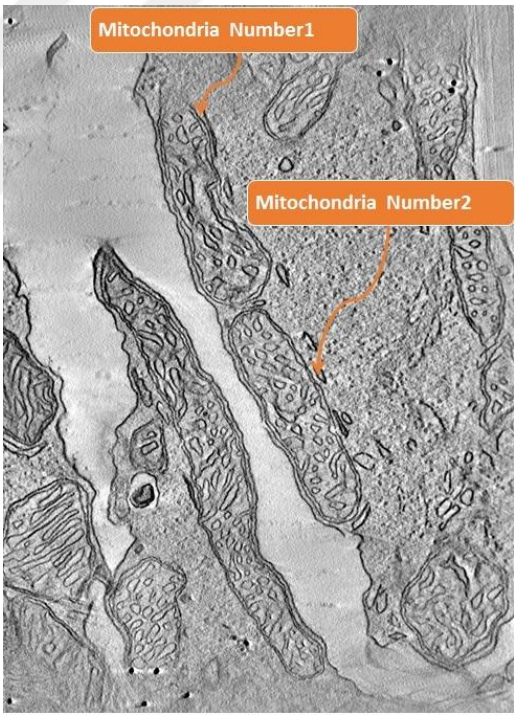
RESULTS

3.1 Test Cases

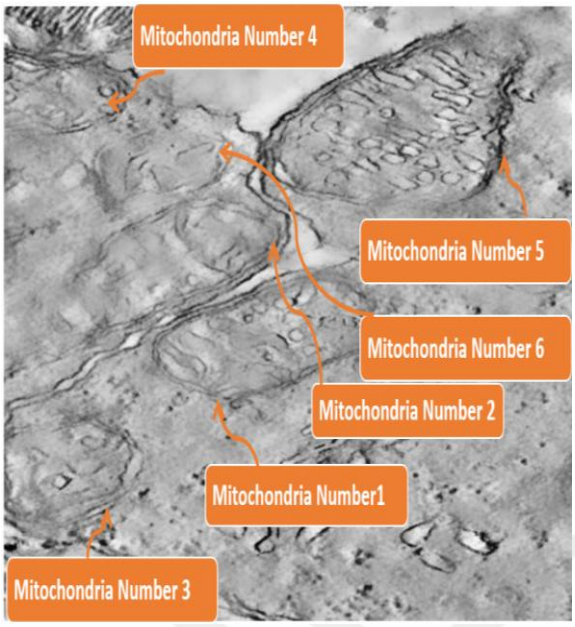
Three methods (active contour, Kalman filter with active contour and Kalman filter with active contour and optical flow) were applied to 28 mitochondria from eight different data sets as shown in Figure 27. Two different evaluation metrics were used for evaluation: Dice coefficient and RMSSSD. Dice and RMSSSD are presented in Tables 4 and 5.



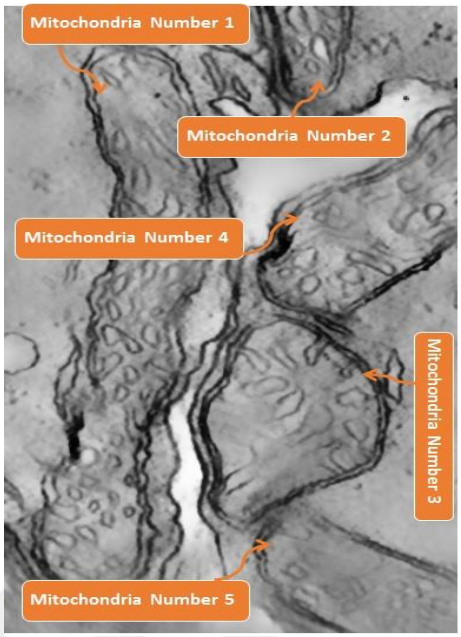
a) DataSet1 ‘cone. Sub ‘



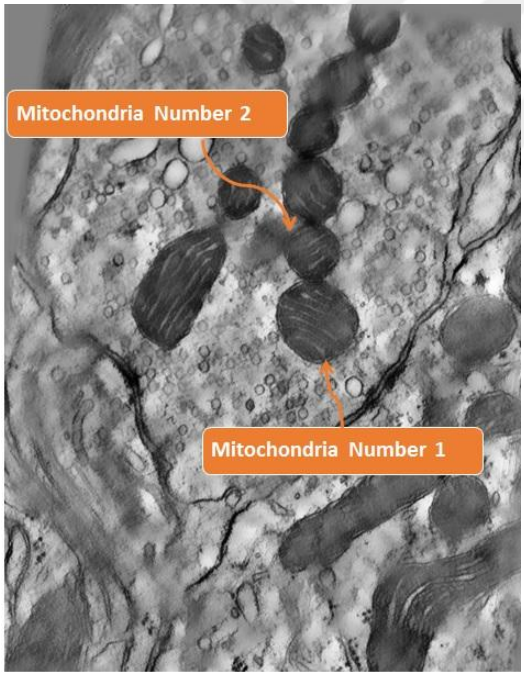
b) DataSet2 ‘6_22.sub’



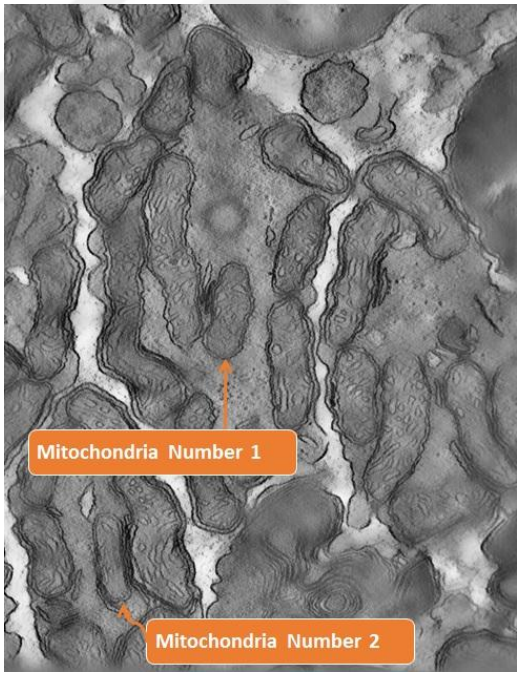
c) DataSet3 ‘bclpb-d.sub‘



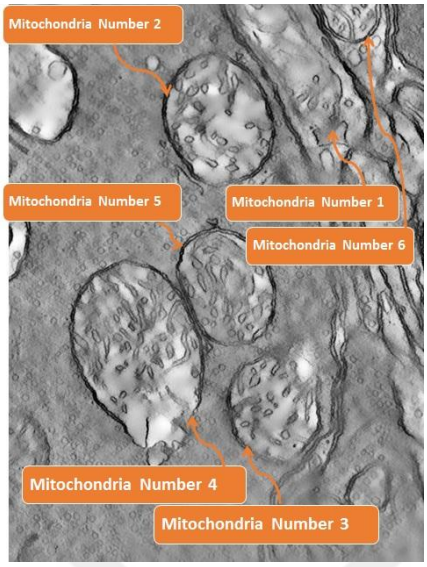
d) DataSet4 ‘gap18_sub’

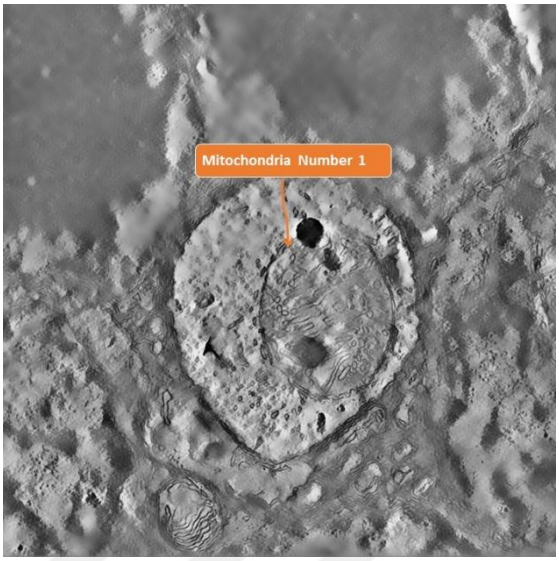


e) DataSet5 ‘mac_serial_sub ‘



f) DataSet6 ‘od.sub’

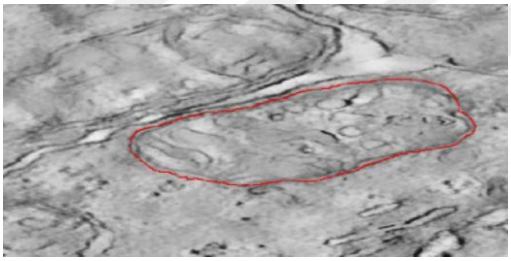




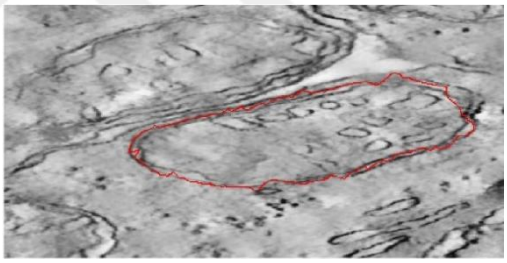
g) DataSet7 ‘pedicle ‘ h) DataSet8 ‘spherule24mos1_’

Figure 27 DataSet images and their mitochondrion

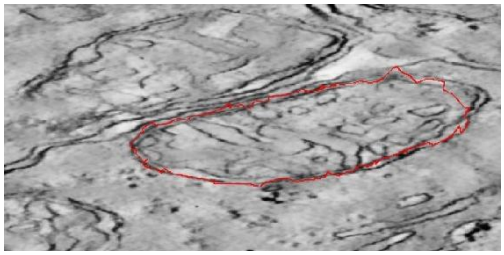
Example resulting contours are displayed in Figure 28 (since the number of slices is too big for this example) only the slice numbers which are multiples of ten are displayed.



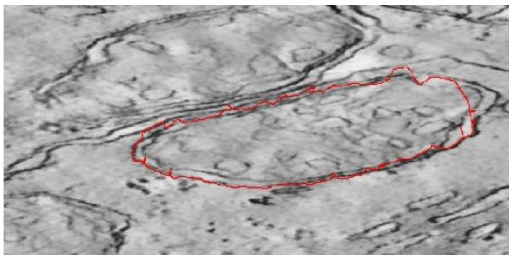
a) Slice number 1 (drawn by the user)



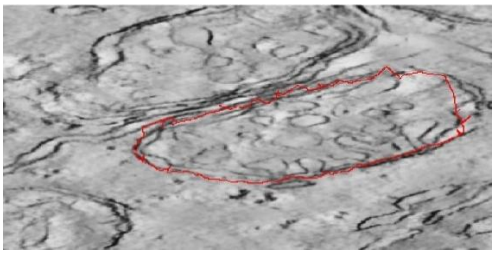
b) Slice number 10

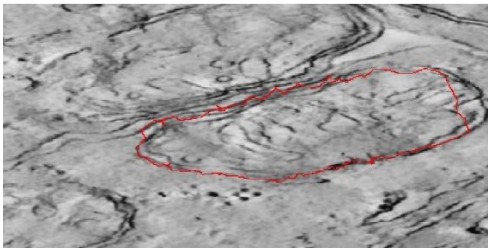


c) Slice number 20

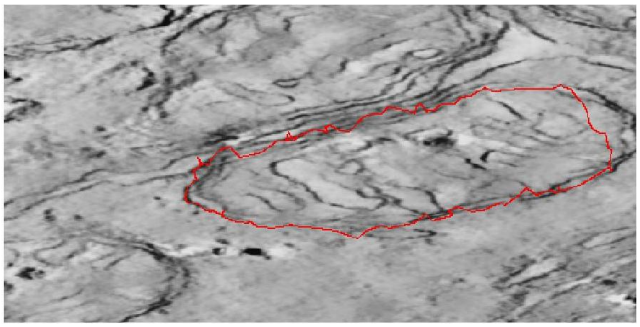


d) Slice number 30





e) Slice number 40 f) Slice number 50



g) Slice number 60

Figure 28. Mitochondrion number 2 of data set" bclpb-d.sub" in different slices from 1 to 60

Table 4(cont.) Segmentation accuracy of mitochondria (Dice coefficient)

Data Set	Mitochondria Number	Active Contour	Kalman filter & active contour	Kalman filter & active contour & Optical flow
		Dice Coefficient		
1	1	0.97	0.97	0.95
	2	0.86	0.87	0.88
	3	0.91	0.92	0.89
2	1	0.92	0.92	0.91
	2	0.95	0.95	0.94
3	1	0.94	0.93	0.94
	2	0.88	0.88	0.85
	3	0.92	0.92	0.89
	4	0.91	0.91	0.90
	5	0.88	0.88	0.87
	6	0.84	0.84	0.85
4	1	0.92	0.89	0.91
	2	0.85	0.85	0.84
	3	0.91	0.91	0.90
	4	0.92	0.92	0.92
	5	0.73	0.77	0.76
	6	0.88	0.88	0.83
5	1	0.91	0.93	0.90
	2	0.88	0.86	0.89
6	1	0.94	0.94	0.90
	2	0.86	0.86	0.85
7	1	0.90	0.90	0.89
	2	0.97	0.97	0.97

	3	0.90	0.90	0.90
	4	0.97	0.97	0.97
	5	0.97	0.97	0.97
	6	0.90	0.91	0.92
8	1	0.96	0.96	0.96

Table 5 Segmentation accuracy of mitochondria (RMSSSD) in units of pixels

Data Set	Mitochondria Number	Active Contour	Kalman filter & active contour	Kalman filter & active contour & Optical flow
		RMSSSD		
1	1	0.9	0.9	1.3
	2	1.8	1.8	1.6
	3	0.7	0.7	0.8
2	1	0.9	0.9	1.0
	3	0.4	0.4	0.4
3	1	0.9	0.9	0.8
	2	1.3	1.3	0.8
	3	0.9	1.0	1.1
	4	0.8	0.9	0.9
	5	1.4	1.0	1.1
	6	1.4	1.3	1.3
4	1	0.7	0.9	0.8
	2	0.9	1.0	0.9
	3	0.7	0.7	0.8
	4	0.3	0.3	0.3
	5	1.1	1.1	1.0
	6	0.7	0.7	0.7
5	1	1.1	1.1	1.2
	2	0.8	0.9	0.8
6	1	0.6	0.6	0.7
	2	0.9	0.9	0.9
7	1	0.4	0.4	0.5
	2	0.6	0.6	0.6
	3	1.1	1.1	1.0
	4	0.7	0.6	0.7
	5	0.4	0.4	0.4
	6	0.2	0.2	0.2
8	1	1.1	1.1	1.0

The Dice and RMSSSD values of every mitochondrion in every slice is calculated and then the average value of every mitochondrion in all slices is calculated and displayed in Tables 4 and 5.

As dice coefficient reaches one it represents better segmentation which is more similar to ground truth segmentation. The value of RMSSSD shows how much the boundary of segmented contour is different from the boundary of ground truth contour. RMSSSD measures the boundary error.

In the best segmentation we should have Dice coefficient equal to one and RMSSSD equal to zero.

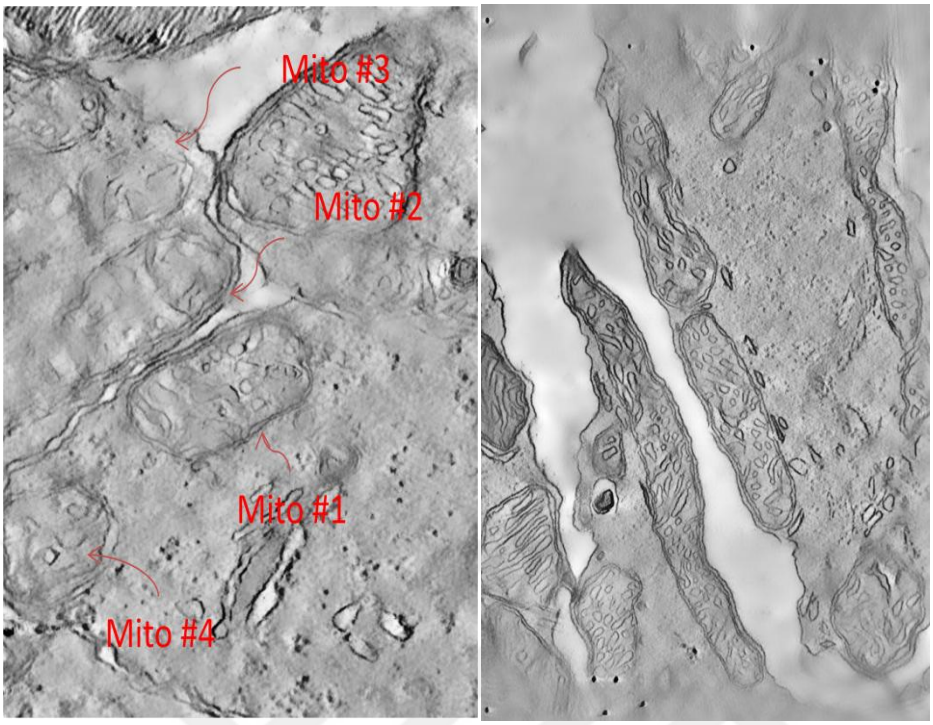
In most of the cases when Dice coefficient gets higher (better), RMSSSD metric gets smaller (better) accordingly. There are also cases where Dice coefficient is better (nearer to one) than RMSSSD (near to zero). These are the cases where the two regions, segmented region and ground truth region, are very similar to each other except in some parts where there is large boundary errors. Figure 29 can help understand the case. The two regions are similar to each other except in the areas which are specified with red circle in the Figure 29.

Figure 29 Ground truth region (in blue) and segmented region (in red)

The results displayed in Tables 4 and 5 are the results of semi-automatic method using splitting/merging/vanishing detecting techniques. There is one case for each of these techniques in Tables 4 and 5. In other words there is one mitochondrion needing splitting, one other mitochondrion needing merging and one mitochondrion needing applying vanishing technique in all mitochondria tested in this thesis.

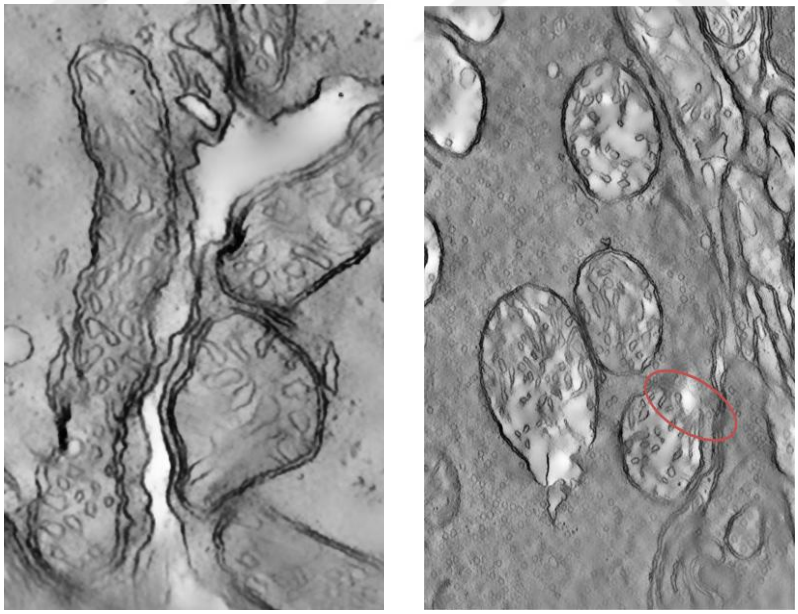
In some of data sets like 'bclpb-d' (data set number 3) the edges are not so strong and the quality is low in comparison to other data sets, so, the results are not so satisfying as seen in Figure 30(a). There are also some cases in the data sets that mitochondria change shape very slowly in each slice and the edges are strong edges, so the results are better than the other cases like data sets '6_22' and 'gap18_sub' where the results are better. These data sets are shown in the Figure 30(b) and (c) .There are some cases of data sets in which some part of mitochondrion is not very clear and the edges of that part are not strong. In these cases, after some slices, the snake will lose the correct boundary and may go very far from the mitochondrion boundary. This will cause less accurate results like the data set ' Pedicle ' as shown in Figure 30(d) . There are also cases in which mitochondrion is very clear and has strong edges in the first slice where the user initializes it, but after some slices, some part of mitochondrion become unclear (have blurry and weak edges), like the sample in data

set 'cone.sub', mitochondrion number one which is specified in the Figure 30(e) in the slice with unclear edges and Figure 30(f) in the slice with more clear edges.
In the algorithm used in this thesis, there is no options for the user to correct the snake or points on the contour, but if there was such an option, the results can be corrected when it starts to lose boundary, and this could help better segmentation in the next slices. When a point starts deviating away from the boundary, then in the proceeding slices, it will get farther away from the boundary if there is no user interaction, but if in the first slice of deviation the user corrects the problematic point, then, in the next slices, it will stick to the boundary and we will have better and more accurate segmentation. It should be stated that in most of the cases when the user initialize the contour in the first slice, it is a little bit different from ground truth, in our test cases it also occurs. For example, when comparing with ground truth in the first slice, there is not 99% dice coefficient, in all cases, it is less than 99% and it continues like that until the last slice, and in most of the cases it decreases gradually; it should be considered that it starts from 98,97 (or in some cases even lower than that), and decreases gradually. In some cases, ground truth starts from some slices other than first slice to segment a mitochondrion, but in the slice that it starts segmentation the mitochondrion is not seen very well, in other words, it's edges are not very clear but after some slices, it can be more clear. If our algorithm starts from the slice which the ground truth have started segmentation, the result would not be good because the mitochondrion is not seen, but if we start segmentation after some slices when the mitochondrion is more clear, the result of segmentation will be better. But, because in the two metrics that we compare the results with ground truth we should consider all slices that contain the mitochondrion in the ground truth, then the result of comparison will not be good. This case can be seen in data set ' gap18_sub ' mitochondrion number 5, the slice that ground truth has started segmentation of this mitochondrion is presented in Figure 30(g). The mentioned mitochondrion is in the right upper corner of the image. This mitochondrion causes the data set to have lower dice coefficient.



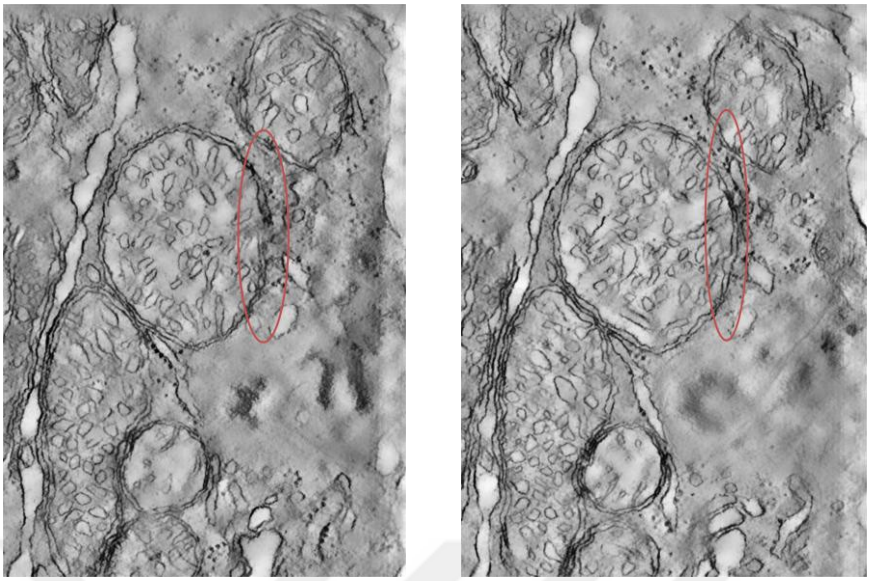
(a) (b)

(a)Data set 'bclpb-d' (data set number 3) where the edges are weak and image quality is low in comparison to other data sets (b) Data set '6_22' where edges are strong and mitochondria in this data set change shape very slowly



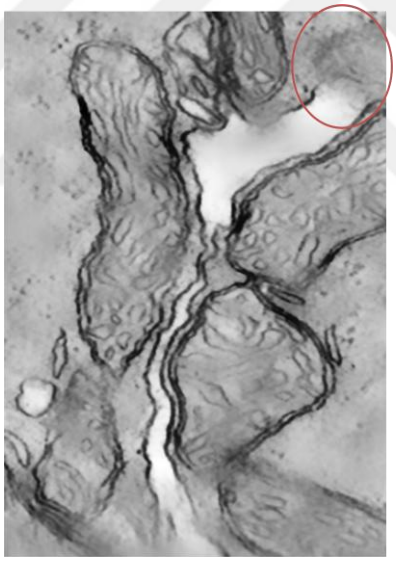
(c) (d)

(c) Data set 'gap18_sub' with strong edges and slowly changing shape of mitochondria (d) Data set ' Pedicle ' with weak and unclear edges in some part of a mitochondrion which is specified with red ellipse



(e) (f)

(e) A mitochondrion in data set 'cone.sub' with weak and unclear edges in some slice in some part of it specified with red ellipse (f) Mitochondrion in data set 'cone.sub' with strong edges in the same part of mitochondrion as specified in (f) in proceeding slices.



(g)

(g)Starting slice of a mitochondrion in data set 'gap18_sub ' which is specified in red and starting slice is not the first slice of data set images.

Figure 30 Some of data sets and their mitochondria

As apparent from Table 4, the values of Dice coefficient and RMSSSD for all three methods are very similar to each other. To distinguish how statistically significant difference there is between the methods a TTest is done. It is comparing each two methods with each other.

Table 6 TTest

	Method 1 & 2	Method 1 & 3	Method 2 & 3
T value	0	0.017	0.016

As apparent from Table 6 there is not statistically significant difference between three methods. Method 1 (active contour) is sufficiently good. The reasons might be a) By limiting iteration number (3 iterations), we limit the possible excursions (from one slice to next) that can occur in active contour. Hence, Kalamn filter function may not needed. b) Mitochondria in the data sets change shape (from one slice to the next) very minimally.

3.2 Multiple initializations

Table 7 Dice Coefficient for three methods with multiple initializations

Data set name	Mitochondrion number	Initialization number	Method 1	Method 2	Method 3
cone.sub	1	1	0.97	0.97	0.95
		2	0.97	0.97	0.96
		3	0.95	0.95	0.94
		Mean	**0.96**	**0.96**	**0.95**
		Standard Deviation	**0.0115**	**0.0115**	**0.01**
	2	1	0.86	0.87	0.88
		2	0.86	0.87	0.82
		3	0.90	0.90	0.86
		Mean	**0.87**	**0.88**	**0.85**
		Standard Deviation	**0.023**	**0.017**	**0.03**
6_22.sub	1	1	0.92	0.92	0.91
		2	0.94	0.94	0.94
		3	0.90	0.90	0.90
		Mean	**0.92**	**0.92**	**0.91**
		Standard Deviation	**0.02**	**0.02**	**0.02**
bclpb-d.sub	1	1	0.94	0.94	0.93
		2	0.94	0.94	0.93
		3	0.95	0.95	0.95
		Mean	**0.94**	**0.94**	**0.94**
		Standard Deviation	**0.005**	**0.005**	**0.011**
gap18_sub	1	1	0.91	0.91	0.90
		2	0.91	0.91	0.90
		3	0.91	0.91	0.92
		Mean	**0.91**	**0.91**	**0.91**
		Standard Deviation	**0**	**0**	**0.011**
pedicle	1	1	0.97	0.97	0.97
		2	0.98	0.98	0.98
		3	0.97	0.97	0.97
		Mean	**0.97**	**0.97**	**0.97**
		Standard Deviation	**0.006**	**0.006**	**0.006**

For some of data sets, the initialization step of the algorithm has been done for multiple times and the results are gathered in Table 7. As it is apparent from Table 7 the results of different initializations are similar to each other but in some cases when the initial points are different it causes the results to change accordingly. It is important that in the first slice when the user initializes the contour, he/she selects proper points on the membrane to be the initial points. When the initial contour is more similar to the ground truth contour then in the proceeding slices the results will be expected to be better. For example in the second mitochondrion of the data set '6_22.sub ' the dice coefficient of the first slice is 0.96. But for the third mitochondrion of the same data set, Dice coefficient in the first slice is 0.94.
Another option that can change the results with multiple initializations might be the number of points on the contour. For example the second mitochondrion in the data set '6_22.sub ' is initialized with 99 points and the third mitochondrion is initialized with 47 points on the contour. The results with larger number of points on the contour is in general better than the lower number of points on the contour. In the cases when the number of points are similar but the initialization points are not located similar to each other then, the volume segmentation results of different initializations can be different. The number of points on the contour is as important as selecting points at correct location on the membrane. In the case of data set 'cone.sub', mitochondrion number 2 is initialized differently in this manner that the cut window of mitochondrion is different from other initialization numbers. In the other two cases of data set ' cone.sub ' the cut window is not properly selected by the user. The user should pay attention to the size of cut window because if window size is such that the mitochondria pass through this window in the proceeding slices the segmentation will go very wrong.

3.3 Results Comparison

Table 8 Results comparison, this thesis results and Tasel's work results

Data Set Name and Number	Tasel 's Dice coefficient (Original)	Tasel 's Dice coefficient (Merged)	Method1 's Dice	Method2 's Dice	Method3 's Dice
bclpb-d (dataset #3)	0.91	0.89	0.88	0.88	0.88
gap18_sub(dataset#4)	0.85	0.87	0.87	0.87	0.86
Pedicle (dataset#7)	0.74	0.84	0.94	0.94	0.94
spherule24mos1_(dataset#8)	0.98	0.97	0.96	0.96	0.96

In the study of Tasel (14), he does detection and segmentation of mitochondria both. If after detection, the ratio of overlapping region to overall area is greater than 30% then the snakes are merged. In Table 8 we compare our three method results with Tasel's original and merged results both. As apparent from Table 8 the results of Tasel's method and ours are similar to each other but in the case of data set ' Pedicle ' the results gained in this thesis is much higher than the results of Tasel's work. In other data sets the results are very similar to each other, a small difference in the results might be because Tasel's method was based on 3D information. Instead, my methods are based on 2D information (slice by slice). In 3D method he also considers the gap between slices of images, so, that might be a reason

for his results to be higher and better than the results gained in this thesis

3.4 Duration of use of methods

The major part that user spends more time on, is the initialization section of the application in which the user specifies the contour points, the points on the membrane of mitochondrion, in the first slice of each data set for any mitochondrion to get that mitochondrion segmented. This time is up to mitochondrion size, number of points that user wants to specify, the speed of user in initialization, and also the distance of points from each other which is varied depending on the user decision. If the user wants to have more accurate segmentation, he/she should specify more points with very little distance from others that will take more time to initialize, and also more execution time in comparison to less point on the contour. The more points on the contour match to mitochondrion membrane, the more accurate segmentation results we will have. So user must be careful with the initialization.

In our tests, two users (both females, of ages 26 and 30) used the application, and their interaction time was measured. Both of the users were familiar with computer applications and university graduates. They tried to initialize snakes according to the ground truth provided by CCDB to decrease wrong segmentations. Each of data sets was evaluated at least once, because of time limitations. In some of the data sets users tested some of mitochondria, but in some others, all mitochondria were tested.

The major execution time of the algorithm which is implemented in this thesis is related to three main steps used in this thesis. They are active contour, Kalman filter and optical flow. Table 9 presents average execution time of these steps. The execution time for each mitochondrion that is tested in this thesis is measured in every slice. Then the mean value of execution time in all slices for a mitochondrion is calculated and presented in Table 9. It shows that much time is spent on optical flow, because of calculating partial derivatives for the whole image. When the image size is greater, the execution time of optical flow is higher. The time recorded for execution of active contour is the time for five iterations. There is no other target value to compare ours with, but we can consider time as a valuable statistic to measure the quality of an application. The user interaction times as well as algorithm run times are tabulated in Table 9. As there are one case for each of merging/splitting/vanishing detection techniques, the time that user spends when specifying splitting points for splitting mitochondrion tested in the semi-automatic method is measured and is 12.06 seconds. The time that user1 spends for detecting the splitting and merging slice number is approximately 1 minute whereas user2 spends about 2 minutes to find splitting slice number and about 1 minute to find merging slice number.

Table 9 Algorithm's duration time and user interaction time

Data Set number	Mitochondrion number	Initialization time (user1)	Initialization time (user2)	Active contour run time in seconds-per slice	Kalman filter run time in seconds-per slice	Optical flow run time in seconds -per slice
1	1	92.2	111.4	5.7	0.07	14.8
	2	51.5	88.0	3.1	0.05	8.8
	3	76.3	72.3	5.2	0.08	14.9
	4	44.7	66.7	2.2	0.03	4.3
2	1	145.7	140.4	28.3	0.2	25.5
	2	63.2	73.2	3.3	0.03	12.9
	3	83.3	114.8	4.7	0.06	19.8
3	1	123.1	94.1	4.7	0.1	6.7
	2	134.4	82.4	3.2	0.05	8.1
	3	162.4	77.1	2.6	0.04	3.6
	4	94.2	84.9	2.3	0.04	2.0
	5	153.0	128.1	3.2	0.05	9.2
	6	61.6	82.4	5.2	0.10	7.4
4	1	96.2	119.7	4.3	0.07	5.3
	2	61.5	76.5	2.2	0.04	0.92
	3	82.9	100.6	3.6	0.06	3.2
	4	97.6	100.1	5.5	0.1	3.3
	5	38.7	51.8	1.2	0.01	1.8
	6	69.8	109.7	2.8	0.05	1.7
5	1	70.8	95.6	3.2	0.05	3.8
	2	41.0	62.2	2.0	0.03	1.7
	3	55.0	72.1	2.4	0.04	2.2
6	1	81.3	80.8	3.6	0.07	3.6
	2	74.4	78.9	3.2	0.06	4.3
	3	57.2	82.4	2.6	0.05	2.9
7	1	97.9	147.9	5.0	0.1	14.0
	2	137.7	135.3	7.1	0.1	14.1
	3	85.2	94.6	3.7	0.06	8.1
	4	112.4	135.9	6.6	0.07	17.3
	5	100.9	105.9	4.4	0.07	9.7
	6	87.1	86.4	3.3	0.06	2.7
8	1	104.7	133.4	9.0	0.08	25.4

CHAPTER 4

CONCLUSION AND FUTURE WORK

The main purpose of this study was to segment mitochondria by using semi-automatic methods on transmission electron microscopy images. The methods implemented in this thesis were active contour, Kalman filter with the aid of active contour, and lastly, Kalman filter with active contour and optical flow. In each of these methods, the first slice of a mitochondrion was initialized by the user, (user specifies a few points on the membrane of mitochondria to form the contour of snake), and in the proceeding slices, these initial points are used to calculate the membrane of mitochondria using the automatic algorithms specified. Kalman filter helps the contour to estimate the point's location in the next slice by calculating it's velocity using point's position in the slice before and the current slice in the second method, but in the last method it gets velocity information from optical flow. The results from the three methods are very similar to each other. TTest is done and shows that there is no significant difference between three methods. The reason might be because by limiting iteration numbers in active contour, possible excursions are limited, so, Kalman filter may not be needed. The second reason might be because of very minimal changes in the shape of mitochondria in data sets.

In some of data sets, there are some mitochondria which after some slices split and become two mitochondria instead of one and, in some cases, two mitochondria become one mitochondrion after some slices. The three methods in this thesis cannot determine these exceptions, so there should be some techniques that determine these exceptions. For this purpose, we proposed splitting and merging techniques. Using these techniques, the results improved significantly. These techniques give an opportunity to the user to correct segmentation by specifying the splitting points of mitochondria. In other words the user can determine the splitting slice before program starts execution, when the splitting slice number reaches, the program waits for the user to specify the splitting points. There is also an option for the user to determine in which slice the mitochondrion is going to disappear when there is vanishing mitochondrion in the data set.

The time spent in the algorithm, used in this thesis, for each of active contour, Kalman filter and optical flow, are measured and is dependent on data sets complexity (image size and number of slices) and size of mitochondrion to be tested. Also the initialization of mitochondria tested in this thesis is done by two users and the time needed for the users to initialize the mitochondria is measured.

The results obtained from the algorithm implemented in this thesis were compared with the results from the work done by Tasel (14). In most of data sets the results are very similar to each other but, in the cases where result is different, the difference is because of differences in methods. Tasel 's algorithm is based on 3D information but in this thesis the algorithms are 2D based.

Some of mitochondria were initialized multiple times and the segmentation results were obtained. Multiple initialization was done with various number of points and different point distances and different point locations for some mitochondria to test these influence the results much or not. The results show that with larger number of points on the contour, better (higher) Dice coefficient and (smaller) RMSSSD values are obtained. In the cases where the initial contour is more similar to ground truth, the segmentation results will be better.

There were some mitochondria in some data sets which start to exist after some slices. In one case we test this kind of mitochondria in the exact slice number, which in the CCDB ground truth have started to segment, and in another case, we start segmentation after some slices of CCDB starting slice number and the segmentation results are better because in that slice the mitochondrion is clearer and the edges are stronger.

In most of the cases the dice results show a segmentation accuracy of 90 % and more which means the segmentation results of the algorithms used in the thesis are very close to the segmentation result of ground truth provided by CCDB. The results also in most of the cases show RMSSSD value of about one pixel which is quite satisfactory. If we compare the results without applying splitting/merging /vanishing detecting techniques then we can see a considerable improvement in segmentation results in the case these techniques were applied. However, such types of mitochondria that require splitting/merging/vanishing detection are very rare among our data sets.

Future work

The segmentation algorithm proposed in this thesis may also be applied on cristae structures. Mitochondrial cristae are structures which increase surface area to provide greater space for processes that occur across this membrane, so their shape is also important and worth studying. There are studies which analyse and segment cellular components such as cistae and study how their shape change through time (45) , (46) and (47). It should be investigated if the segmentation algorithm proposed in this thesis could be adapted for cristae structures in the mitochondria.

In the first step of the algorithm, proposed in this thesis, the user has to initialize the contour points on the mitochondrion membrane. There is no chance for the user to correct wrongly specified points, and existence of such an option could help for a more user friendly application.

In some cases the snake goes far from the mitochondrion membrane, if there was an option in the application by which the user could correct the snake points, this could help, because if the snake continue with wrong points in the proceeding slices, it will go farther than mitochondria membrane.

In the cases where snake points pass through membrane or stay inside the mitochondria there could be an option or tool in the application which allows the user to drag boundary to the mitochondria membrane.

As data sets are large with many slices, parallel processing could speed up the segmentation process and increase user satisfaction.

REFERENCES

1. *Evolutionary biology: essence of mitochondria.* Henze K, Martin W. 2003, Nature, pp. 426(6963): 127–8.

2. *Functions and dysfunctions of mitochondrial dynamics.* Scott A. Detmer, David C. Chan. 2007, Nature Reviews Molecular Cell Biology, pp. 870-879.

3. *Ultrastructure of the Mitochondrion and Its Bearing on Function and Bioenergetics.* Rossignol, Giovanni Benard and Rodrigue. s.l. : Antioxidants & Redox Signaling, 2008, Antioxidants & Redox Signaling, pp. 1313-1342.

4. *Mitochondria: more than just a powerhouse.* McBride HM, Neuspiel M, Wasiak S. 2006, Curr Biol, pp. 551-60.

5. Emerson H. Duke, Stephen R. Aguirre. *3D Imaging: Theory, Technology and Applications (Computer Science, Technology and Applications).* s.l. : Nova Science Pub Inc, 2010.

6. *Mitochondria and Their Role in Cardiovascular Disease.* Marín-García, J. 2013, Springer Science, p. 500.

7. *Electron microscopy morphology of the mitochondrial network in gliomas and their vascular microenvironment.* Arismendi-Morillo, Gabriel. 2011, Biochimica et Biophysica Acta, pp. 602-608.

8. conservapedia. [Online] [Cited: 15 05 2015.] http://www.conservapedia.com/File:Mitochondriafigure1.jpg.

9. *Transmission electron microscopy:physics of image formation and analysis.* Reimer, L. 1997, Springer Series in Optical Sciences.

10. *A fast reconstruction algorithm for electron microscope tomography.* Sandberg, K., Mastronarde, D.N. & Beylkin G. 2003, Journal of Structural Biology, pp. 61-72.

11. https://www.int.kit.edu/1731.php. *www.int.kit.edu.* [Online] [Cited: 17 05 2015.] https://www.int.kit.edu/1731.php.

12. *Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure.* Denk, W. & Horstmann, H. 2004, PLos Biol.

13. Taşel, Faris Serdar. *PHD Proposal.* Ankara : Middle East Technical University, 2012.

14. *Automatic detection of mitochondria from electron microscope tomography images: a curve fitting approach.* Tasel, Serdar F., et al. 2014, SPIE.

15. Çöçelli, Mustafa. *Semi automatic/user guided segmentation of mitochondria on transmission electron microscopy images.* Ankara : s.n., July 2015.

16. *Axon tracking in serial block-face scanning electron microscopy.* Elizabeth Jurrus, Melissa Hardy, Tolga Tasdizen, P. Thomas Fletcher, Pavel Koshevoy, Chi-Bin Chien, Winfried Denk, Ross Whitaker. 2008, Medical Image Analysis.

17. *Semi-Automatic Medical Image Segmentation with Adaptive Local Statistics in Conditional Random Fields Framework.* Yu-Chi J. Hu, Michael D. Grossberg, and Gikas S. Mageras. 2008, Conf Proc IEEE Eng Med Biol Soc, pp. 3099-3102.

18. *Interaction in the segmentation of medical images: a survey.* Silvia D Olabarriaga, Arnold W. M. Smeulders. 2001, Medical Image Analysis, pp. 127-142.

19. *Image Segmentation Methods: A Comparative Study.* A. M. Khan, Ravi. S. 2013, International Journal of Soft Computing and Engineering , pp. 84-92.

20. *Image segmentation based on Edge detection using boundary code.* Uemura, Takumi, Koutaki, Gou and Uchimura, Keiichi. 2011, International Journal of Innovative Computing, Information and Control, pp. 6073-6083.

21. Maimon, Oded, Rokach, Lior. Data Mining and Knowledge Discovery Handbook. Springer New York Dordrecht Heidelberg London : Springer Science Business Media, 2010, pp. 981-998.

22. *Two methods for semi-automatic image segmentation.* Bonnet, J. Cutrona and N. Marbella, Spain : s.n., 2001. Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing. pp. 467-471.

23. *Semiautomatic segmentation with compact shape prior q.* Piali Das, Olga Veksler, Vyacheslav Zavadsky, Yuri Boykov. 2009, Image and Vision Computing, pp. 206-219.

24. *Interactive Medical Image Segmentation Using Snake and Multiscale Curve Editing.* Xie, Wu Zhou and Yaoqin. s.l. : Hindawi Publishing Corporation, 2013, Jourlib journal, p. 13 pages.

25. *Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features.* Aurelien Lucchi, evin Smith, adhakrishna Achanta, Graham Knott, and Pascal Fua. s.l. : IEEE Transactions on Medical Imaging, 2011.

26. *Fast Segmentation of Sub-Cellular Organelles.* Dilip. K. Prasad, Chai Quek and Maylor K.H. Leung. 2012, International Journal of Image Processing.

27. *A Comparative Study of Feature Descriptors for Mitochondria and Synapse Segmentation.* Kendrick Cetina, Pablo Marquez-Neila and Luis Baumela. 2014. International Conference on Pettern Recognition. pp. 3215-3220.

28. *A cell-centered database for electron tomographic data.* Martone, M.E., A. Gupta, M. Wong, X. Qian, G. Sosinsky, B. Ludäscher, and M.H. Ellisman. 2002, Journal of structural biology, pp. 145-155.

29. *The cell centered database project:an update on building community resources for managing and sharing 3D imaging data.* Martone, M.E., J. Tran, W.W. Wong, J. Sargis, L. Fong, S. Larson, S.P. Lamont, A. Gupta, and M.H. Ellisman. 2008, J Struct Biol, pp. 220-231.

30. *The cell-centered database.* Martone, M.E., S. Zhang, A. Gupta, X. Qian, H. He, D.L. Price, M. Wong, S.Santini, and M.H. Ellisman. 2003, Neuroinformatics, pp. 379-395.

31. *Snakes: Active contour models.* Michael Kass, Andrew Witkin , Demetri Terzopoulos. 1988, International Journal of Computer Vision, pp. 321-331.

32. *A fast algorithm for active contours and curvature estimation.* Shah, D. J. Williams and M. 1991, CVGIP: Image. Understanding, pp. 14-26.

33. *Computer-aided characterization of mammographic masses: accuracy of mass segmentation and its effects on characterization.* Sahiner B, et al. 2001, IEEE Trans Med Imaging, pp. 1275-84.

34. *On active contour models and balloons.* COHEN, Laurent D. 1991, CVGIP:IU, pp. 211-218.

35. Bishop, Greg Welch and Gary. *An Introduction to the Kalman Filter*. Chapel Hill : University Of North Carolina at Chapel Hill, 2006.

36. Oyeleke, Julius. *Kalman tracking for image processing applications.* 2010.

37. *Determining Optical Flow.* G.Schunck, Berthold K.P. Horn and Brian. 1981, Artificial Intelligence 17, pp. 182-203.

38. *Learning Optical Flow.* Deqing Sun, Stefan Roth, J.P. Lewis, and Michael J. Black. 2008, Springer-Verlag Berlin Heidelberg, pp. 83–97.

39. Smith, Steven W. *The Scientist and Engineer's Guide to Digital Signal Processing.* s.l. : California Technical Publishing, 1997.

40. *Automatic contrast enhancement by histogram warping.* Dodgson, Mark Grundland and Neil A. Warsaw, Poland : Springer Netherlands, 2004. Computational Imaging and Vision. pp. 293-300.

41. *Towards automatic image enhancement using genetic algorithms* . Munteanu, C. and Rosa, A. La Jolla, CA : IEEE , 2000. Evolutionary Computation. pp. 1535-1542.

42. Eitz, Mathias. *Bilateral Filtering.* Berlin, Germany : s.n., 21 November 2006.

43. *Computer visualization of three-dimensional image data using IMOD.* Kremer, J.R., D.N. Mastronarde, and J.R. McIntosh. 1996, J Struct Biol, pp. 71-76.

44. *Editorial: 3D Segmentation in the Clinic: A Grand Challenge II - Liver Tumor Segmentation.* Du, Xiang Deng and Guangwei. 2008, MICCAI Workshop on 3D Segmentation in the Clinic.

45. *Insight into mitochondrial structure and function from electron tomography.* T.G. Frey, C.W. Renken, G.A. Perkins. 2002, ELSEVIER , Biochimica et Biophysica Act, pp. 196-203.

46. *Electron Tomography of Mitochondria from Brown Adipocytes Reveals Crista Junctions.* G. A.Perkins, J. Y. Song, L. tarsa, T. J. Deerinck, M. H. Ellisman, T. G. Frey. s.l. : Journal of Bioenergetics and biomembranes, 1998, Vols. 30, No 5.

47. *Shape-Driven Three-Dimensional Watersnake Segmentation of Biological Membranes in Electron Tomography.* Hieu Nguyen and Qiang Ji. 2008, IEEE Transactions on Medical Imaging, pp. 616-628.

48. Çöçelli, Mustafa. *Semi automatic/user guided segmentation of mitochondria on transmission electron microscopy images.* Ankara : s.n., July 2015.

49. *Segmentation of mitochondria in electron microscopy images using algebraic curves.* Mojtaba Seyedhosseini, Mark H. Ellisman, and Tolga Tasdizen.

50. Don Wayne, Fawcett. M.D. *The Cell.* Philadelphia London Toronto Mexico City Rio de Janeiro Sydney Tokyo : W. B. Saunders Company , 1917.