To: Dr. Saundra F. DeLauder, Dean, School of Graduate Studies and Research

The members of the Committee approved the Thesis of Shana Kai Brown
Candidate's Name

as presented on 4/7/16.
Date

We recommend that it be accepted in partial fulfillment of the requirements for the degree

Master of Science in Applied Mathematics
Degree Name Major/Program Name

SMakrogianni
Advisor
Department Math Date 4/19/2016

Member
Department Math Date 4/19/2016

Member
Department Math Date 4/21/2016

External Member
Affiliation Computer Science Date 4/19/2016

Approved

N Edwards
Department Chairperson or Designee
Department Math Date 4/21/2016

Academic Dean or Designee
College CMWSI Date 4/21/16

Saundra f DeLauder
Dean, School of Graduate Studies and Research
Date 4/29/16

GRAPH THEORETIC IMAGE ANALYSIS METHODS WITH APPLICATIONS TO COMPUTER VISION AND BIOMEDICINE

by

SHANA KAI BROWN

A THESIS

Submitted in partial fulfillment of the requirements for
the degree of Master of Science in the Applied
Mathematics Graduate Program
of Delaware State University

DOVER, DELAWARE
May 2016

COPYRIGHT

Copyright © 2016 by Shana Kai Brown. All rights reserved.

DEDICATION

To My Beautiful Mother Erleen Joy Brown

ACKNOWLEDGEMENTS

I would like to thank LSAMP for the funding and giving me the opportunity to complete my Masters degree.

I would like to thank Dr. Sokratis Makrogiannis for giving me the opportunity of being apart of his research group MIVIC (Mathematical Imaging and Visual Computing) and supporting me through my thesis.

I would like to thank my research group (MIVIC) for encouraging me, working with me and assisting me with my research.

I would like to thank my mom (Erleen Joy Brown), my dad (William Brown), family and friends for always encouraging me, supporting me and constantly pushing me to be better.

I would like to thank my friend Leon Hunter for helping me improve my programming skills and assisting me with my research.

ABSTRACT

In this thesis we use graph cut methods to solve the image segmentation problem. Image segmentation is used to partition pixels of an image that have a strong correlation (color, intensity, texture) into different regions such as object and background. Graph cut algorithms are efficient ways to solve computer vision and computer graphics problems. They use a graph model to represent the visual content and identify the optimal cuts that divide the image into regions by energy function optimization. The energy function includes a data and a smoothness term, and is embedded in a weighted and directed graph. The optimal solution is obtained by finding a minimum cut.

First, we will study mathematical methods and algorithms for image segmentation using graph optimization. Next, we will study and compare multiple graph cut techniques including methods that approximate geodesic distances for segmentation. We will study implementations of the main graph cut technique and apply it to color and gray scale images using MATLAB software. Then, we will test these graph cut techniques at different levels of noise to see how the noise influences the segmentation accuracy. After that, we will validate and compare these techniques on standard and medical data sets. Finally, we will measure segmentation accuracy using a function that uses the color error, region size, and the number of final region.

The goal of this thesis is to show that graph-based techniques can be used to delineate the objects in a visual scene for applications in computer vision and biomedicine.

TABLE OF CONTENTS

TITLE PAGE i

COPYRIGHT

DEDICATION ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

TABLE OF CONTENTS v

LIST OF FIGURES OR ILLUSTRATIONS vii

LIST OF TABLES xi

CHAPTER 1 INTRODUCTION 1

- 1.1 Image Segmentation 1
  - 1.1.1 Thresholding 2
  - 1.1.2 Edge-Based 3
  - 1.1.3 Region-Based 4
  - 1.1.4 Model-Based 5
  - 1.1.5 Graph-Based 8
  - 1.1.6 Why Do We Use Image Processing? Applications 9

CHAPTER 2 FUNDAMENTALS OF GRAPH THEORY 10

- 2.1 Graph Theory 10
  - 2.1.1 Definitions 10
- 2.2 Image Processing Methods That Use Graph Theory 13
  - 2.2.1 Normalized Cuts (N-Cuts) 13
  - 2.2.2 Graph Cuts 14
  - 2.2.3 Minimal Spanning Trees 16
  - 2.2.4 Registration 16
  - 2.2.5 Entropic Graphs 16

CHAPTER 3 METHODS OF GRAPH CUTS 18

- 3.1 Minimum Cut/ Maximum Flow 18
  - 3.1.1 Augmented Path 21
  - 3.1.2 Push Relabel 21
  - 3.1.3 Boykov and Kolmogorov New Algorithm 22
  - 3.1.4 Chan Vese 25
  - 3.1.5 Chan Vese Two Stage 26
  - 3.1.6 Mumford Shah 27
  - 3.1.7 Rousson Deriche 27

CHAPTER 4 EXPERIMENTS, COMPARISON, RESULTS 30
4.1 Experiment 1 30
4.2 Experiment 2 33
4.3 Experiment 3 34
CHAPTER 5 CONCLUSION 45
REFERENCE LIST 46
CURRICULUM VITAE 49

LIST OF FIGURES OR ILLUSTRATION

1.1 Partitioned image [10]. 6

1.2 Corresponding quad tree. R represents the entire region [10]. 7

1.3 A curve is given by the zero level set of function ϕ, is the boundary between the regions $\{(x, y) : \phi(x, y) > 0\}$ and $\{(x, y) : \phi(x, y) < 0\}$ [27]. 8

1.4 Grid graph [2]. 9

2.1 Example of a point. 11

2.2 Example of a line. 11

2.3 Example of a vertex. 11

2.4 Example of an edge. 12

2.5 Example of a graph. 12

2.6 Point 'a' and 'b' are adjacent vertices. Line 'ab' and line 'be' are adjacent edges [18]. 13

2.7 Flow chart for the N-cut grouping algorithm. 15

3.1 (a) Image with seeds (b) Graph (c) Graph cut (d) Segmentation result [24]. 20

3.2 Example of the search trees S (red nodes) at the end of the growth stage when path (yellow line) from the source s to the sink t is found. Active and passive nodes are labeled by letters A and P, correspondingly. Free nodes appear in black [3]. 23

3.3 Flow chart that represents Chan Vese algorithm. 26

3.4 Flow chart that represents Mumford Shah algorithm. 28

3.5 Flow chart that represents Rousson Deriche algorithm. 29

4.1 Example of Experiment 1: This figure shows the original and segmented images of a ping wing on a dark background from the BSD 200 data base and four different segmentations of the image. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmentation at 3 percent noise. The set of images in the bottom right was the segmented at 5 percent noise. 31

4.2 Example of Experiment 1: This figure shows the original and segmented images from the BSD 200 database of a couple of bears and four different segmentations of the image. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was segmented at 3 percent noise. The set of images in the bottom right was segmented at 5 percent noise. . . 32

4.3 Example of Experiment 1: This figure shows different stages of the Original Graph Cut Methods of a building from the BSD 200 database. The first set of images in the top left was segmented at 0 percent noise. The second set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmentation at 3 percent noise. The set of images in the bottom right was segmented at 5 percent noise. 35

4.4 Example of Experiment 1: This figure shows different stages of the Original Graph Cut Method of three owls from the BSD 200 database. The first set of images in the top left was segmented at 0 percent noise. The second set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was segmented at 3 percent noise. The last set of images in the bottom right was segmented at 5 percent noise. 36

4.5 Results of Experiment 1: Graph corresponding to the results of four methods (Chan Vese, Chan Vese Two Stage, Rousson Deriche, Mumford Shah) on the BSD 200 database shown in Table 4.1 at different noise levels. The different bars represent the mean at that specific noise level and the black error bars represents the standard deviation. 38

4.6 Results of Experiment 1: Graph corresponds to the results of the Original Graph Cut Method on the BSD 200 Database shown in Table 4.1 at different levels of noise. The different bars represent the mean at that specific noise level and the black error bars represents the standard deviation. 39

4.7 Example of Experiment 2: This figure shows the original and segmented image of a banana from the GT database and four different segmentations of the image. GT labels represents what a "perfect" segmentation should look like. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmented at 3 percent noise. The last set of images in the bottom right was segmented at 5 percent noise. . . 40

4.8 Example of Experiment 2: This figure shows the original and segmented image of a book on a Table from the GT database and four different segmentations of the image. GT labels represents what a "perfect" segmentation should look like. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmented at 3 percent noise. The set of images in the bottom right was the segmented at 5 percent noise. . 41

4.9 Results of Experiment 2: Graph corresponds to the four methods in Table 4.2. The different bars represent the mean at a specific noise level and the black error bars represents the standard deviation. . . 43

4.10 Example of Experiment 3: Shows the original and segmented image of the lower leg. 43

4.11 Example of Experiment 3: Shows the original and segmented image of the lower leg. 44

4.12 Example of Experiment 3: Shows the original and segmented image of the lower leg. 44

LIST OF TABLES

4.1 Results of Experiment 1: Mean YLGC and standard deviation YLGC of five different graph cut techniques on BSD 200 database. 37

4.2 Results of Experiment 2: Mean YLGC and standard deviation YLGC of four different graph cut techniques of 30 images from GT database. 42

4.3 Results of Experiment 3: Mean YLGC and standard deviation YLGC of Original Graph Cut Technique on lower leg. 42

Chapter 1

INTRODUCTION

1.1 Image Segmentation

Image segmentation is the process of partitioning an image into multiple segments. It began in 1970, but as of now there is no robust technique for solving the segmentation problem. Image segmentation problems can be understood as partitioning the image elements (pixels/voxels) into different categories. The main goal of image segmentation is to distinguish objects from the background in an image. Image segmentation can be viewed as a pattern recognition problem because it requires segmentation of pixels. By constructing a graph from an image, segmentation problems can be solved by using techniques for graph cuts in graph theory. The image segmentation problem can be approached in many different ways, such as region-based segmentation, boundary based segmentation, N-cuts, graph cuts and thresholding. Each technique has its pros and cons. But, some techniques give a better segmentation than others. Some segmentation techniques don't work for all images.

The main purpose of image segmentation is to divide an image into parts that have a strong correlation with the objects or the background. The more information you obtain prior to the segmentation process the better the outcome of the segmentation results. There are three main segmentation catagories: thresholding, edge based segmentation and region based segmentation. Image segmentation algorithms are developed based on two basic properties of intensity values discontinuity based and similarity based. In the discontinuity-based approach the partition is done based on some rapid changes in gray level intensity of the image such as detection of isolated points, detection of lines and edge detection. In the similarity based approach segmentation is done based on grouping of pixels which are based on some features such as thresholding, region splitting, region growing, clustering, region merging, k means clustering and fuzzy c means clustering.

1.1.1 Thresholding

Thresholding is a fast and simple image segmentation process. A threshold is a brightness constant used to segment objects and backgrounds. Thresholding can be done by first finding edge segments and then attempting to connect the segments into the boundaries or; the pixels of an image are partitioned based on their intensity value and/or properties of these values. Let $f(x,y)$ represent an image composed of light objects on a dark background. If a threshold T is selected that differentiates the object and background, then any point (x,y) in the image where $f(x,y) > T$ is called a object point. If $f(x,y) \leq T$ it is called an background point. Hence g(x,y)is the segmented image defined by:

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T \\ 0, & \text{if } f(x,y) \leq T \end{cases}$$

Global threshold, local threshold, semi threshold and variable threshold are some of the different types of thresholds. The process given in the formula above is referred to as global thresholding when T is a constant appropriate over an entire image. The expression variable thresholding is used when the value of T changes over an image. If the value of T at any point (x,y) in an image depends on properties of a neighborhood of (x, y), then the terms local thresholding or regional thresholding can be used to represent variable thresholding.

Threshold selection methods are used to determine threshold automatically. If information after segmentation is known the threshold is selected to ensure that this property is satisfied. The threshold can be determined as a minimum between the two highest local maxima. The threshold of the optimal thresholding is determined as the closest gray-level corresponding to the minimum probability between the maxima of two or more normal distributions. Optimal threshold tend to result in minimum error segmentation.

1.1.2 Edge-Based

Edge based segmentation detects edge pixels and then link them together to form contours. Edge detecting operators are used to locate the edges of an image. The edges found are the key component of the edge-based segmentation. Edges can be identified using Laplacian, LoG, Gradient and Canny filtering. Edges are linked if the magnitude of the gradient criteria is satisfied and if the direction of the gradient vector criteria is satisfied.

There are many problems associated with edge-based segmentation the two most common problems are, when the edges appear where there is no border and no edges are present where borders exist. In edge relaxation, edge properties are considered as neighboring edges. If there is an existence of a border then the local edge strength increases. If there is no evidence of a border then the local edge strength decreases. Using the global relaxation process continuous borders are constructed. There are three types of region borders an inner, outer and extended. The inner border is always considered in the region but the outer borders are not. Extended border is a single common border between adjacent regions.

1.1.3 Region-Based

Region based segmentation locates the middle of an object then it expands outward until it reaches the boundary of the object. If a region based image segmentation satisfies the following condition its segmentation is complete.

1)$\cup_{i=1}^{s} R_i = 0$

2)$R_i \cap R_j = \varnothing$, if $i \neq j$

The maximum region homogeneity conditions $H(R_i)$ = TRUE when $i = 1, 2, S$ (eq.2) and $H(R_j \cup R_i)$ where $i \neq j$ and R_i adjacent to R_j. Region based segmentation considers gray-levels from neighboring pixels using one of the three basic approaches, region merging, region splitting, and split-and-merge region growing.

In the region merging method, all pixels in the image are given a unique label and is assigned to a region. Next, assess the neighboring regions of all regions and decide if the similarity measure evaluates to true for all pairs of neighboring regions. If the similarity measure is true for any two neighboring pixels then these pixels are assigned the same label. Similarity measure in this case can be region minimum, region maximum or region mean values etc. This process continues until no more merging is possible. The order in which the regions are chosen will have an effect on the results.

All pixels are seeds. Region growing process begins with a set of seeds either predefined seeds or randomly chosen seeds. Then a similarity measure must be selected based on gray level difference, texture or color. For example regions with homogeneous gray level; the gray level difference similarity measure may be more appropriate. For textured regions the texture feature may be suitable. There are two methods that can be used to finish this segmentation process. The first one is, seeds can be selected only from objects of interest then continue to grow regions as long as the similarity criterion is satisfied. The second one is, seeds can be selected from the entire range of gray levels in the image. Then grow regions until all pixels in the image belong to a region. Some disadvantages to this process is finding a good starting point and suitable criteria for similarity measure.

Split and merge region growing process separates the image into regions depending on a similarity measure. Then the regions are merged based on the same or different similarity measure. First, similarity measure is chosen based on texture, mean variance etc. Then the Original image is divided into four sub images. If any sub image is not uniform based on the similarity measure, then that sub image is divided into four more sub images shown in Figures 1.1 and 1.2. After each iteration, compare neighboring regions and merge regions if it is uniform. Then, stop when no further merging is possible. This method has the advantage of using both approaches.

1.1.4 Model-Based

Model based segmentation gives labels to pixels by matching the priori known object model to the image data. This segmentation method assigns deterministic labels to pixels by using low-level features (discontinuity and homogeneity).

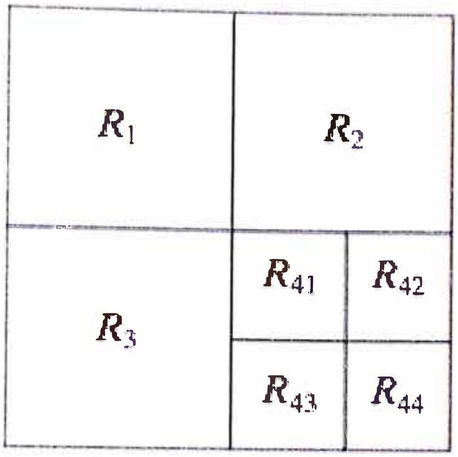


Figure 1.1: Partitioned image [10].

Active Contours (Snakes)

Active contours provide an efficient way of segmentation of curves within an image domain that can move under the influence of external and internal forces. These forces are defined such that the snakes will form a border around an objects boundary. Active contours are used to find the contour of an object by forming a snake around its boundary. Active contours add priori information of properties to the image before preforming segmentation. This makes the process of locating boundaries easier.

The energy function for calculating snakes that we want to minimize is defined by $E_{snakes} = E_{internal} + E_{external} + E_{constraint}$ where $E_{internal}$ is the internal energy of the snakes which depends on the key properties of the curve and is the sum of elastic energy and bending energy, defined by $E_{internal} = E_{elastic} + E_{bending} = \int_s \frac{1}{2}(\alpha \mid V_s \mid^2$

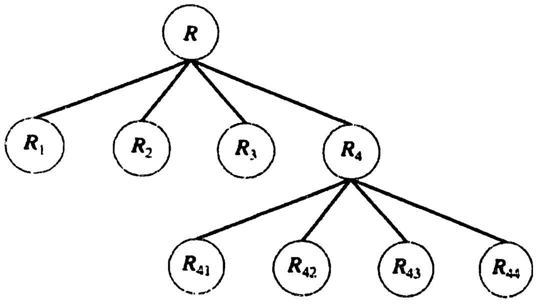


Figure 1.2: Corresponding quad tree. R represents the entire region [10].

$+\beta \mid V_{ss} \mid^2)ds$. $E_{external}$ is the external energy of the snakes which is derived from an image so that E_{image} takes on its smaller values at the feature of interest, defined by $E_{ext} = \int_s E_{image}(v(s))ds$. The last term $E_{constraint}$ gives rise to external constraint forces.

Level sets

In 1988, Osher and Sethian proposed an effective implicit representation for evolving curves and surfaces known as level sets. Level sets allow for automatic change of topology such as merging and breaking. These calculations are made on a fixed rectangular grid. The segmentation of the image plane Ω is computed by locally minimizing an appropriate energy functional defined by $E(C) = -\int \mid \nabla I(C) \mid^2 ds + v_1 \int \mid C_s \mid^2 ds + v_2 \int |C_{ss}|^2 ds$ where C_s and C_{ss} represents the first and second derivative with respect to the curve parameter s. $-\int \mid \nabla I(C) \mid^2 ds$ is the external energy which represents the image information. The last two terms represent the internal energy of the contour which measures the stiffness and the length of the contour. By implementing the gradient descent equation $\frac{\partial C}{\partial t} = -\frac{\partial E(C)}{\partial C} = F \cdot n$ The boundary C is evolved from some initialization in direction of the negative energy gradient.

A given curve C is represented implicitly, as the zero level set of a scalar Lip Schitz continuous function $\phi : \Omega \rightarrow \Re$ which is also known as the level set function. where $\phi(x,y) > 0$ in $\omega, \phi(x,y) < 0$ in $\Omega \amalg \omega, \phi(x,y) = 0$ on $\partial\omega$. Figure 1.3.

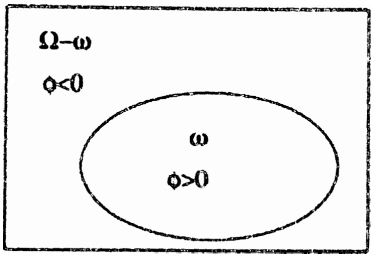


Figure 1.3: A curve is given by the zero level set of function ϕ, is the boundary between the regions $\{(x,y) : \phi(x,y) > 0\}$ and $\{(x,y) : \phi(x,y) < 0\}$ [27].

1.1.5 Graph-Based

Graph based segmentation models the impact of pixel neighborhoods on a given cluster of pixels or pixel, given that the image is homogeneous. In graph based partitioning methods the image is represented as a weighted, undirected graph $G(V, E)$ where the pixels of a graph are represented as nodes (V) and edges (E) that connect nodes belonging to neighboring pixels. As a result a grid graph is created. Figure 1.4.

The graph is represented by a $W \times H$ image that has $n = WH$ nodes and $m = W(H-1) + H(W-1)$ edges. The weight function $w : E \rightarrow \Re$ measures the dissimilarity between the adjacent pixels. The graph is then partitioned into sub

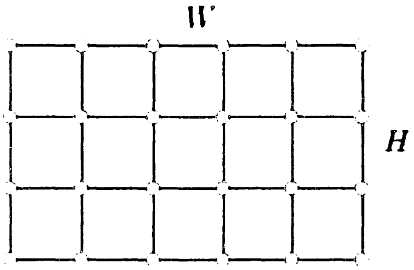


Figure 1.4: Grid graph [2].

graphs depending on the similarity criterion.

1.1.6 Why Do We Use Image Processing? Applications

Image segmentation is useful when the content of an image needs to be analyzed. Image segmentation is an essential tool in medical image analysis and is used in many different applications. In medical imaging, image segmentation is used to locate tumors and other pathologies. Analyze anatomical structures, brain MRI segmentation, detect multiple sclerosis, tissue structures, cells, muscle blood vessels, quantification, surgical planning and much more. Graph cut techniques are also useful in every day applications such as image cropping, n dimensional image segmentation, colorization, image reconstruction, image editing and more. Other applications of image segmentation are facial recognition, machine vision, finger print recognition, stereo, traffic control system and locate objects in a satellite etc.

Chapter 2

FUNDAMENTALS OF GRAPH THEORY

2.1 Graph Theory

An undirected graph $G = (V, E)$ is defined as a set of nodes V and a set of undirected edges E that connect the nodes V. In an undirected graph the set E contains edges that connect unordered pairs of vertices. In an undirected graph $e_{ij} = e_{ji}$ where $e_{ij} \in E$ every edge link has exactly two distinct vertices. By assigning each edge a weight, the graph becomes an undirected weighted graph. A directed graph is defined as a set of nodes (Vertices V) and a set of ordered pair of vertices V or directed edges E that connect the nodes.

Every image can be represented as a graph. In graph theory a cut is a set of edges $c \in E$ such that the two terminals become separated on the induced graph $G = (V, E/C)$. Denoting a source terminal as S and sink terminal as T, a cut (S,T) of $G = (V, E)$ is a partition of the vertices into S and $T = V/S$ into two disjoint subsets such that $t \in T$ and $s \in S$. Every cut has a cost that is defined as the sum of the edges. Two basic cuts in graph theory are minimum cut and maximum cut. A cut is minimum when the size of the cut is not larger than the size of any other cut. A cut is maximum when the size of the cut is not smaller than the size of any other cut.

2.1.1 Definitions

(`http://www.tutorialspoint.com/graph_theory/graph_theory_fundamentals.htm`)

Point

A point is a precise position in a one dimensional, two dimensional, or three-dimensional space. A point can be represented as a dot. Points are denoted by a letter in the alphabet.

Figure 2.1: Example of a point.

Line

A line connects two points. It can be represented by a solid line.

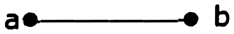


Figure 2.2: Example of a line.

Vertex

A vertex or node is a point where multiple lines intersect. A vertex also can be represented by a letter of the alphabet.

Figure 2.3: Example of a vertex.

Edge

An edge (arc) is a line that connects two vertices. Edges can be formed from a single vertex, without a vertex an edge cannot be formed.

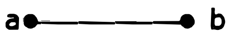


Figure 2.4: Example of an edge.

Graph

A graph G is defined as $G = (V, E)$ where V is a set of all vertices in the graph and E is the set of all edges in the graph.

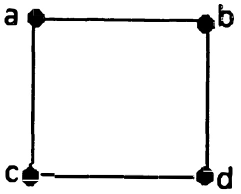


Figure 2.5: Example of a graph.

Adjacency

In a graph, two vertices are said to be adjacent, if there is an edge between the two vertices. The adjacency of vertices is maintained by the single edge that is connecting those two vertices.

In a graph, two edges are said to be adjacent, if there is a common vertex between the two edges. the adjacency of edges is maintained by the single vertex that is connecting two edges.

Cycle

A cycle is a path that starts and ends with the same vertex.

Tree

A tree is a connected, acyclic graph.

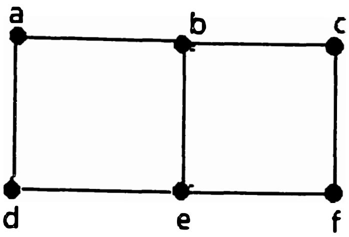


Figure 2.6: Point 'a' and 'b' are adjacent vertices. Line 'ab' and line 'be' are adjacent edges [18].

2.2 Image Processing Methods That Use Graph Theory

2.2.1 Normalized Cuts (N-Cuts)

In [10] the set of points in an arbitrary feature space are represented as a weighted undirected graph $G = (V, E)$. The nodes of the graph denoted V are the points in the feature space and the edges denoted E are formed between every pair of nodes. The weight on each edge denoted $w(i, j)$ is a function of the similarity between nodes i and j. By removing edge connection between two nodes a graph can be partitioned into two disjoint sets $A, B, A \cup B = V, A \cap B = \varnothing$. The cut is the degree of dissimilarity between the two disjoint sets. It can be computed as total weight of the edges that have been removed. The cut is represented by the formula below:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$$

The ideal bi partitioning of a graph is the one that minimizes the cut value above. The Normalized cut (N-cut) is a disassociation measure that computes the cut cost as a fraction of the total edges connections to all nodes in the graph. The normalized cut is represented by the formula below:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

Where

$$assoc(A,V) = \sum_{u \in A, t \in V} w(u,t)$$

is the total connection between nodes in A to all nodes in the graph. $assoc(B,V)$ is defined similarly. The normalized cut is used as the partitioning criterion. Researchers showed that by embedding the normalized cut problem in the real value domain, an approximate discrete solution can be found efficiently. By computing the optimal partition of the normalized cut equation we obtain the following results:

$$y_1 = arg.min_{z^T z_0 = 0} \frac{z^T D^{-1/2}(D-W)D^{-1/2} z}{z^T z}$$

Where D is an $N \times N$ matrix with d on its diagonal and W is a $N \times N$ symmetrical matrix with $W(i,j) - w_{ij}$. Researchers concluded that the real valued solution to the normalized cut problem is found with the second smallest eigenvector of the generalized eigensystem. These results show that the graph partitioning criterion using graph cuts can be computed efficiently by solving a generalized eigenvalue problem. The grouping algorithm consists of the following steps in Figure 2.7.

2.2.2 Graph Cuts

In this paper we discuss different methods of graph cuts. In graph cut technique first we create a weighted, directed graph based on the original image. Object and

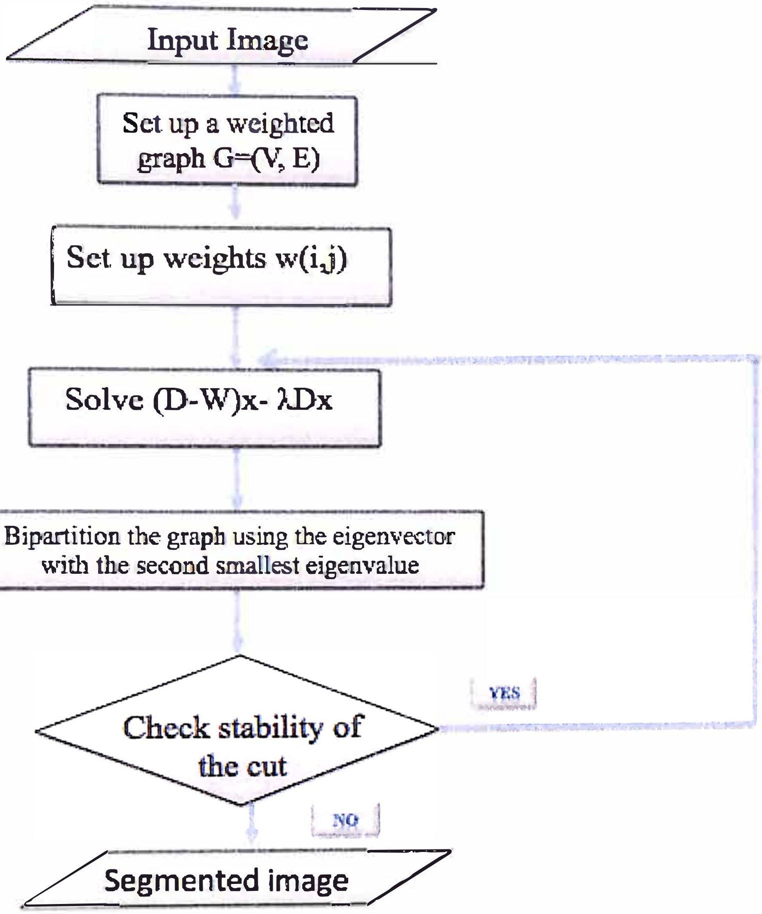

Input Image
Set up a weighted graph G=(V, E)
Set up weights w(i,j)
Solve (D-W)x- λDx
Bipartition the graph using the eigenvector with the second smallest eigenvalue
Check stability of the cut
YES
NO
Segmented image

Figure 2.7: Flow chart for the N-cut grouping algorithm.

background seeds can be detected frequently by the use of prior information. Then edge cost for s and t link are compared over the whole image. The max flow is then found using max flow graph optimization. Finally the final $s - t$ cut solution defines a partition or segmentation of the original image.

2.2.3 Minimal Spanning Trees

A tree is a connected graph without cycles. When all the vertices of a tree are connected it is called a spanning tree. A minimal spanning tree is a spanning tree of connected, undirected graph where all the vertices are connected with the minimal total weight for its edges. A minimal spanning tree is a spanning tree whose weights is not larger than the weight of any other spanning tree. The weight of a tree is the sum of the edges. A single graph can have many spanning trees. For the graph $G = (V, E)$, the spanning tree is $E' \subseteq E$ such that the subset of edges spans all vertices. So that there are no cycles the number of edges is one less than the number of vertices.

2.2.4 Registration

Image registration is the process of aligning or overlapping two or more images of the same view taken at different angles or at different points in time. This process involves labeling one image as the reference image. Geometric transformations are applied to the other images so that they align with the reference image. A geometric transformation maps locations in one image to new locations in another image and it allows you to capture common features in different images. The main step of the image registration process is to determine the correct geometric transformation parameters.

2.2.5 Entropic Graphs

Entropic graph methods are used to approximate similarity measures for image registration. Entropic methods use a matching criterion. This criterion can be defined as relative entropy of the feature distribution. Any graph whose normalized total weights is a consistent estimator of α-entropy is an entropic graph. Minimum spanning trees is an example of entropic graphs. Entropic graphs does not need to compute histogram density in order to approximate information divergence in an attempt to improve bias between good and bad image matches. Entropic methods can take non-linear relations between features.

Chapter 3

METHODS OF GRAPH CUTS

3.1 Minimum Cut/ Maximum Flow

Graph cut algorithms are used to solve computer vision and computer graphics problems. Graph cut algorithms preform optimal and accurate segmentation with minimum cut and maximum flow. Graph cut segmentation use min cut/max flow algorithms to partition an image into regions. A flow network is defined as a directed graph where an edge has a nonnegative capacity. If f is a flow, then the net flow across the cut (s, t) is defined to be $f(s, t)$. Where $f(s, t)$ is defined as the sum of all edge capacities from S to T subtracted from the sum of all edge capacities from T to S. The capacity of the cut (s, t) is $c(S, T)$ which is the sum of the capacities of all edges from S to T. A minimum cut is a cut whose capacity is the minimum over all cuts of the graph G. A maximum cut is a cut whose capacity is the maximum over all cuts of the graph. The optimal solution is obtained by finding the minimum cut.

Graph cuts solve a region based segmentation problem by using minimum cut and maximum flow algorithms. The direct use of minimum cut maximum flow algorithms was image reconstruction. Graph optimization algorithm were used for image reconstruction by Boykov, Kolmogorou, Funka-Lee and Jolly. The method identifies one or more points as the object or background. Background points are called seeds and seeds serve as segmentation hard constraints. Soft constraints reflect boundary and/or region information. The general version of the cost function C calculated on an image segmentation f follows the Gibbs model. The Gibbs model is defined by

$C(f) = C_{data(f)} + C_{smooth(f)}$.

The goal is to preform optimal graph cuts to minimize the cost function $C(f)$. We define an arc-weighted graph $G_{st} = (V \cup \{s, t\}, E)$ with source and sink nodes s and t representing the object and background. s and t are also called terminal nodes. V is the set of nodes corresponding to the image pixels. Segmentation labels are represented by the terminal points s and t. The arcs E in G_{st} is the set of weighted edges and can be represented as a an n link or an t link. The n link connects neighboring pixels whose cost is derived from the smooth term $(C_{smooth(f)})$. The t link connect pixels and terminals with cost derived from the data term $(C_{data(f)})$.

Minimum cut maximum flow algorithm find the max flow from source s to sink t, where the capacity of each is given by cost. The minimum cut is equal to the maximum flow. The s-t cut in G_{st} partitions the set into two disjoint subsets s and t by the saturation of arcs. The cost of the s-t cut is total cost of arcs in the cut. The s-t cut with the minimum cost is the minimum s-t cut. Figure 3.1

Let I denote the set of image pixels. Let N denote the set of all directed pairs of pixels (c, p) where $p, q \in I$. Let O be sets of image pixels of objects and background seeds. Where O is a subset o fAV, B is a subset of V and $O \cap B = \varnothing$. When each image pixel i_k take a binary label L_k element of obj, bgd where obj are objects and bgd is the background labels a segmentation is determined by the Label vectors $L = \{L_1, L_2 .... L|I|\}$. The segmentation cost or energy function is given by $C(L) = R(L) + B(L)$ where λ is the weight,

$$R(L) = \sum_{p \in I} R_p(L_p)$$

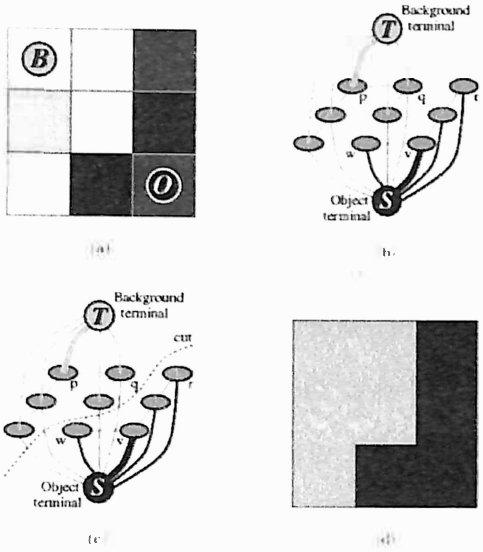


Figure 3.1: (a) Image with seeds (b) Graph (c) Graph cut (d) Segmentation result [24].

is the region term and

$$B(L) = \sum_{p,q \in N} B_{p,q}\delta(L_pL_q)$$

is the boundary term where

$$\delta(L_pL_q) = \begin{cases} 1, & \text{if } L_p \neq L_q \\ 0, & \text{otherwise} \end{cases}$$

In a graph cut segmentation seeds can be added to improve results. If seeds for object and background form small patches, one can compete intensity priors $P(I \mid O)$ and $P(I \mid B)$. Costs are defined by $R_p(obj) = -lnP(I_p \mid O)$, $R_p(bgd) = -lnP(I_p \mid B)$, $B(p,q) = exp(-\frac{(I_pI_q)}{2\varphi^2}\frac{1}{\|p,q\|})$ where $\| p,q \|$ denotes the distance between pixels p,q. As a result $B(p,q)$ is high for small differences between image values $\mid I_p - I_q \mid < \varphi$.

Cost $B(p,q)$ is low for boundary locations where $\mid I_p - I_q \mid < \varphi$

Most existing minimum cut maximum flow algorithms use the augmented path method or the push-relabel method.

3.1.1 Augmented Path

Augmented paths algorithm work by pushing flow along non-saturated paths from the source to the sink until the maximum flow in the graph is reached. Augmented path algorithms continuously documents information about the distribution of the current s/t flow denoted f among the edges G using a residual graph G_f. At the initialization, there is no flow from the source to the sink. With each new iteration, the algorithm finds the shortest s/t path along the non-saturated edges of the residual graph. If a path is found, then the flow is implemented by pushing the maximum possible flow that saturates at least one of the edges in the path. Each augment increases the total flow from the source to the sink $f = f + df$. Then a new $s - t$ path is found and then you keep repeating the previous steps until no new path is found. The maximum flow is reached when any s/t path crosses at least one saturated edge in the residual graph. Finally the s and t graph nodes are separated resulting in the partition. The use of the shortest path is an important factor that improves theoretical running time complexities for algorithms based on augmenting paths.

3.1.2 Push Relabel

Push-relabel algorithm for maximum flow optimization maintains a labeling of nodes with a lower bound estimate of its distinct sink node along the shortest non-saturated path. Graph cut segmentation algorithm create an arc-weighted directed graph corresponding to the image to be segmented. Then you determine the difference between objects and background seed. By using source and sink nodes to connect all seeds to either the source or sink. The algorithm pushes excess flow towards nodes with smaller estimated distance to the sink. Now, determine the appropriate arc cost with each link of the image and use maximum flow graph optimization to determine the graph cut. The distance progressively increase as edges are saturated by push operations. Flow that is not delivered to the sink is eventually drained back to the source. The minimum s-t cut solution identifies the graph nodes that correspond to the image boundaries, separating the objects and the background.

3.1.3 Boykov and Kolmogorov New Algorithm

Boykov, Kolmogorov developed a new algorithm in [3], in an attempt to improve empirical performance of standard augmented path techniques on graphs in vision. The new algorithm developed is based on augmented paths by building search trees for detecting augmented paths. Two non-overlapping search trees are built one from the source and one from the sink. In tree S, all edges from each parent node to its children are non-saturated. In tree T, edges from children to their parents are non-saturated. Nodes that do not belong to S or T are called free nodes. $S \subset V, s \in S, T \subset V, t \in T, S \bigcap T = \varnothing$. Nodes in a search tree can either be active (outer border in the tree or passive (internal). Active nodes allow trees to grow by acquiring new children (along non-saturated edges) from a set of free nodes. Passive nodes cannot grow. An augmented path is found as soon as an active node from one tree detects a neighboring node from the other tree SHOWN in Figure 3.2.

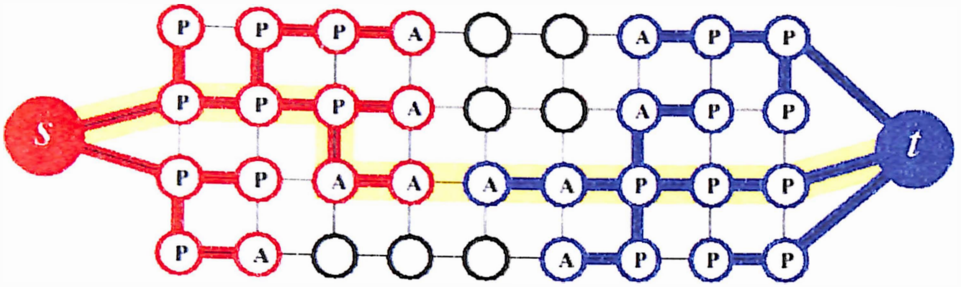


Figure 3.2: Example of the search trees S (red nodes) at the end of the growth stage when path (yellow line) from the source s to the sink t is found. Active and passive nodes are labeled by letters A and P, correspondingly. Free nodes appear in black [3].

Algorithm [10]

1. Growth stage

Search trees S and T grow until they touch giving an s/t path

i The search tree expands.

ii Active nodes explore adjacent non-saturated edges and acquire new children from a set of free nodes.

iii New nodes become active members of the corresponding search tree.

iv When all neighbors of a given active node are explored, the active node becomes passive.

v The growth stage terminates if an active node encounters a neighboring node that belongs to the opposite tree.

2. Augmentation

The found path is augmented and search tree breaks into forest.

- i Augments the path found at the growth stage
- ii Since we push through the largest flow possible, some edge in the path become saturated.
- iii Some of the nodes in the tree S and T may become orphans. Basically the edges linking them to their parents are no longer valid.
- iv The augmented phase may split the search tree S and T into forests.
- v The source s and the sink t are still roots of two of the trees, while orphans from roots of all other trees.

3. Adaption

Trees S and T are restored.

- i The goal of the adoption stage is to restore the single-tree.
- ii Find a new valid parent for each orphan. Parent should be connected through a non saturated edge.
- iii If there is no qualifying parent, the orphan is removed from S or T and is made into a free node. All the of its former children are orphans.
- iv This stage terminates when no orphans are left. The search structures of S and T are restored.

v Since some orphan nodes in S and T may become free, the adoption stage results in contraction of these sets.

After the 3rd stage is complete the algorithm returns to the growth stage. The algorithm terminates when the search tree S and T cannot grow and the trees are separated by saturated edges. Hence a maximum flow is achieved.

3.1.4 Chan Vese

Using graph cut optimization Chan Vese is a region based segmentation method that implements graph cuts based on the Mumford Shah model. The Chan Vese model minimizes the intra-region intensity variance and the Euclidean Length of the segmentation boundary which is a geodesic in an image based N-D Riemann space (`http://cbia.fi.muni.cz/user_dirs/gc_doc/namespaceGc_1_1Algo_1_1Segmentation_1_1ChanVese.html`). Approximation using graph cuts is an alternative method for Riemann metric. The energy functional which the chan vese model minimizes is defined by

$$E_{CV}(C, c_1, c_2) = \lambda_1 \int_{\Omega_1} (f(x) - c_1)^2 dx + \lambda_2 \int_{\Omega_2} (f(x) - c_2)^2 dx + \mu \mid C \mid$$

where

- $f(x)$ is the image
- C is the contour separating the regions (foreground and background)
- c_1 is the mean intensity value of the background
- c_2 is the mean intensity value of the foreground
- $\lambda_1, \lambda_2, \mu$ are fixed parameters

Chan Vese Algorithm

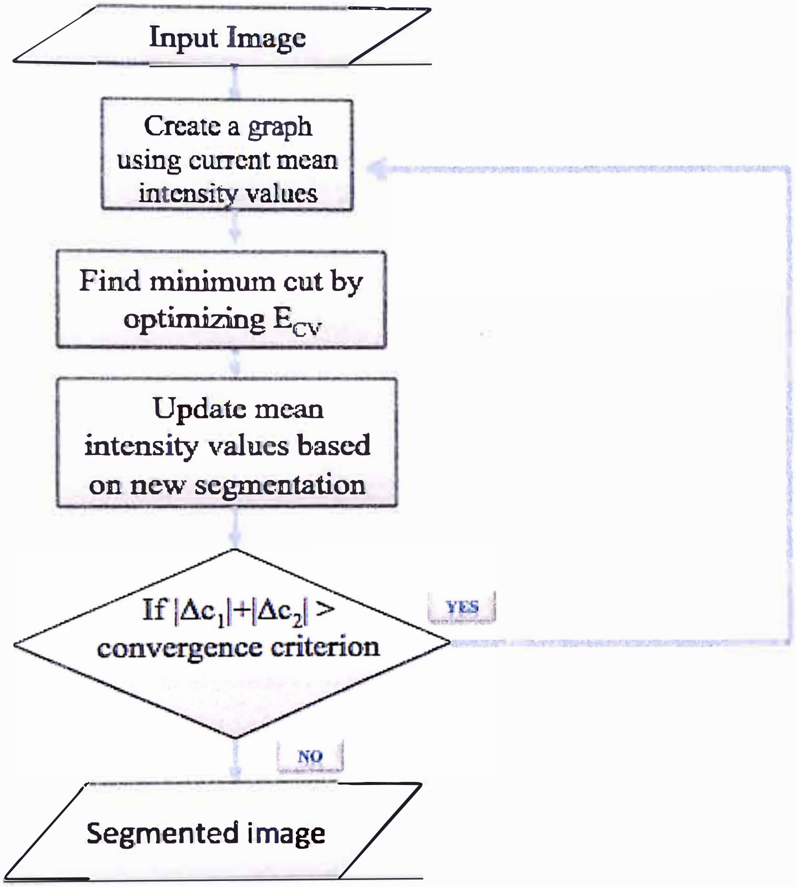

Input Image
Create a graph using current mean intensity values
Find minimum cut by optimizing E_{CV}
Update mean intensity values based on new segmentation
If $|\Delta c_1|+|\Delta c_2| >$ convergence criterion
YES
NO
Segmented image

Figure 3.3: Flow chart that represents Chan Vese algorithm.

3.1.5 Chan Vese Two Stage

Chan Vese two stage for graph cut optimization is computed by calculating the original Chan Vese segmentation with a larger neighborhood only around a small band around the boundary obtained in the first stage. The benefits of this approach is a faster computation and less memory consumption.

3.1.6 Mumford Shah

Mumford Shah model for image segmentation using graph cuts optimization is based on α-expansion algorithm. The input image U consists of k piecewise constant regions. Every region is characterized by the mean intensity of that region c_i. The energy functional being minimized is defined by

$$E_{MS}(C, c_1, ..., c_k) = \mu \mid C \mid + \sum_{i=1}^{k} \lambda_i \int_{\Omega_i} (f(x) - c_i)^2 dx$$

When the Mumford Shah model has two regions it is equivalent to the Chan Vese model. Mumford Shah model is more generic then the Chan-Vese. Chan-Vese is a special case of Mumford Shah technique for a piecewise constant model for foreground and background. (`http://cbia.fi.muni.cz/user_dirs/gc_doc/namespaceGc_1_1Algo_1_1Segmentation_1_1MumfordShah.html`)

Mumford Shah Algorithm

3.1.7 Rousson Deriche

The Rousson Deriche Bayesian Model for graph cut optimization is a regional based foreground/background image segmentation method. The energy functional which this method tries to minimize is defined by

$$E_{RD}(C, c_1, \sigma_1^2, c_2, \sigma_2^2) = \lambda \int_{\Omega_1} e_1(x) dx + \lambda \int_{\Omega_2} e_2(x) dx + \mid C \mid$$

where

- $f(x)$ is the input image
- C is the contour separating the regions (foreground and background)

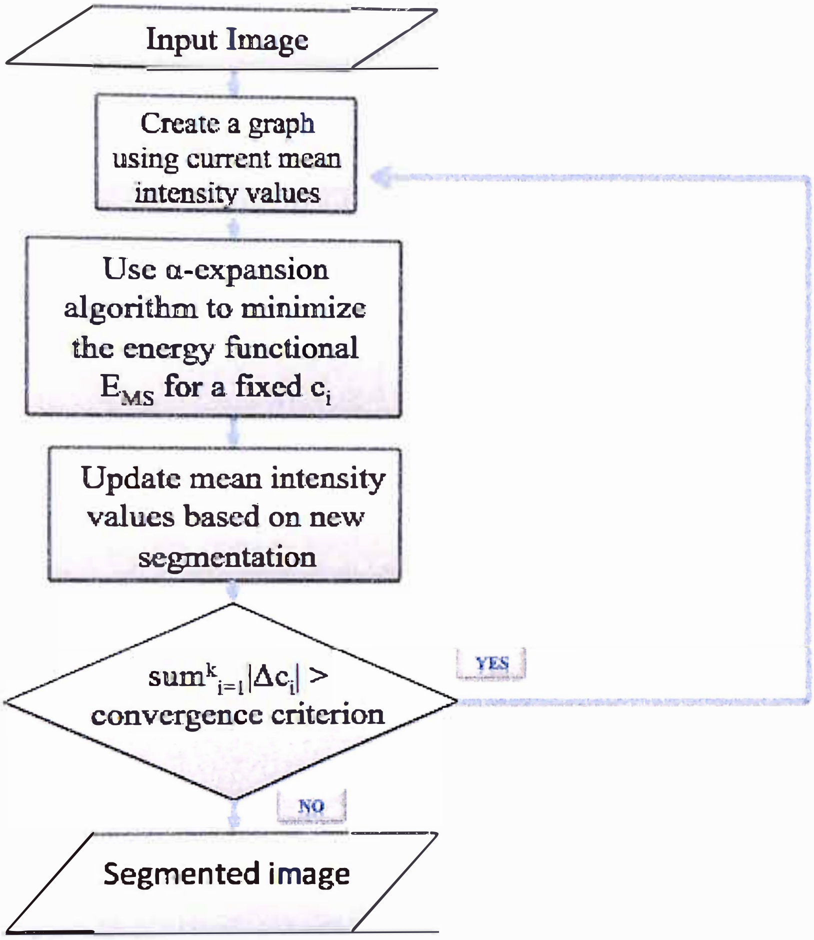

Input Image
Create a graph using current mean intensity values
Use α-expansion algorithm to minimize the energy functional E_{MS} for a fixed c_i
Update mean intensity values based on new segmentation
$\mathrm{sum}^k_{i=1}|\Delta c_i|$ > convergence criterion
YES
NO
Segmented image

Figure 3.4: Flow chart that represents Mumford Shah algorithm.

- c_1 is the mean intensity value of the background
- c_2 is the mean intensity value of the foreground
- λ is a fixed parameters σ_1^2 and σ_2^2 is the intensity variance of the foreground and background e_k is defined by $e_k = \frac{(f(x)-c_k)^2}{\sigma_k^2} + log\sigma_k^2$ (`http://cbia.fi.muni.cz/user_dirs/gc_doc/namespaceGc_1_1Algo_1_1Segmentation_1_1RoussonDeriche.html`)

Rousson Deriche Algorithm

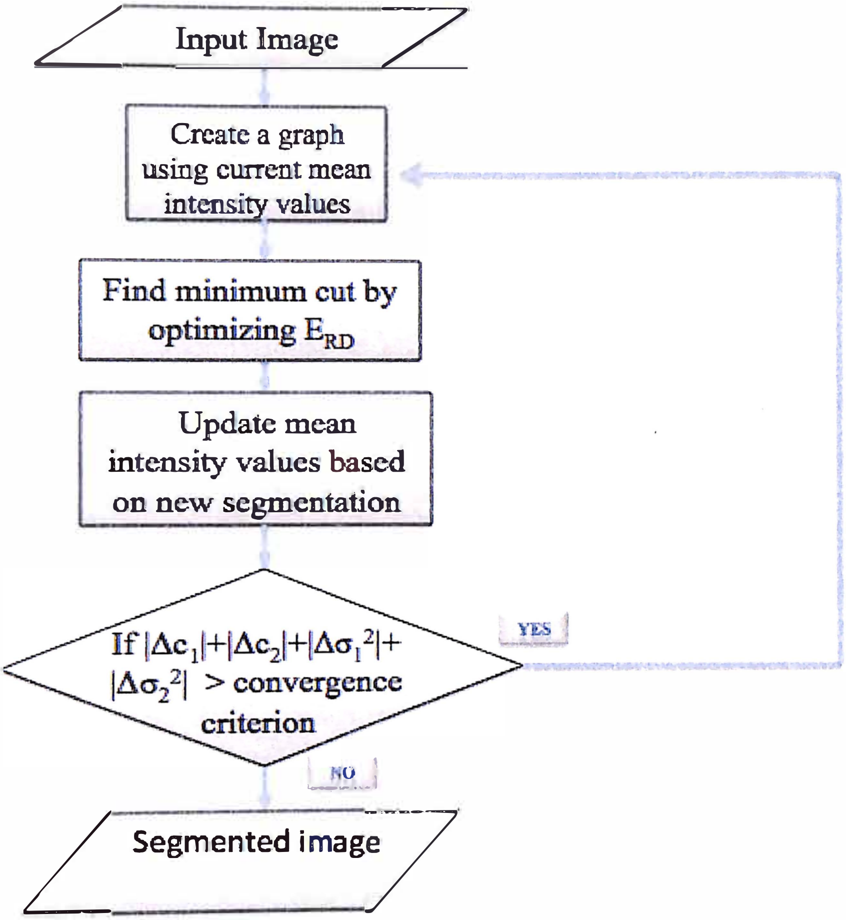

Input Image
Create a graph using current mean intensity values
Find minimum cut by optimizing E_{RD}
Update mean intensity values based on new segmentation
If $|\Delta c_1|+|\Delta c_2|+|\Delta \sigma_1^2|+|\Delta \sigma_2^2|$ > convergence criterion
YES
NO
Segmented image

Figure 3.5: Flow chart that represents Rousson Deriche algorithm.

Chapter 4

EXPERIMENTS, COMPARISON, RESULTS

4.1 Experiment 1

In Experiment 1 first, we segmented 200 images from the BSD database (`https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/`) using five different graph cut techniques (Original Graph Cut Method, Chan Vese, Chan Vese two stage, Rousson Deriche, Mumford Shah). The Original Graph Cut technique was segmented in color shown in Figure 4.3 and figure 4.4. Whereas the other four methods were segmented in gray scale shown in Figure 4.1 and Figure 4.2. Then, we compared the original image from the database to the segmented images. To compute segmentation accuracy we use YLGC error function. This helped to determine how efficient the segmentation was. This experiment was repeated three times at different noise levels. We applied the Gaussian noise filter where the mean was set to zero and the variance was set to 1 percent, 3 percent and 5 percent. In Table 4.1 we reported the mean and standard deviation of YLGC over the BSD 200 data set for five different graph cut techniques. Figure 4.5 and Figure 4.6 are graphs that represent the data in Table 4.1.

YLGC is an equation created by J. Liu and Y.-H. Yang that measures the segmentation accuracy [21].

$$YLGC = \sqrt{\frac{N_R}{h \cdot w \cdot c}} \cdot \sum_{i=1}^{N_R} \sigma i^2 / \sqrt{card_i}$$

where

- h- height

- w- width
- c- number of image channels
- N_R- number of final regions
- σi^2- color error over region i
- $card_i$- number of pixels inside region i

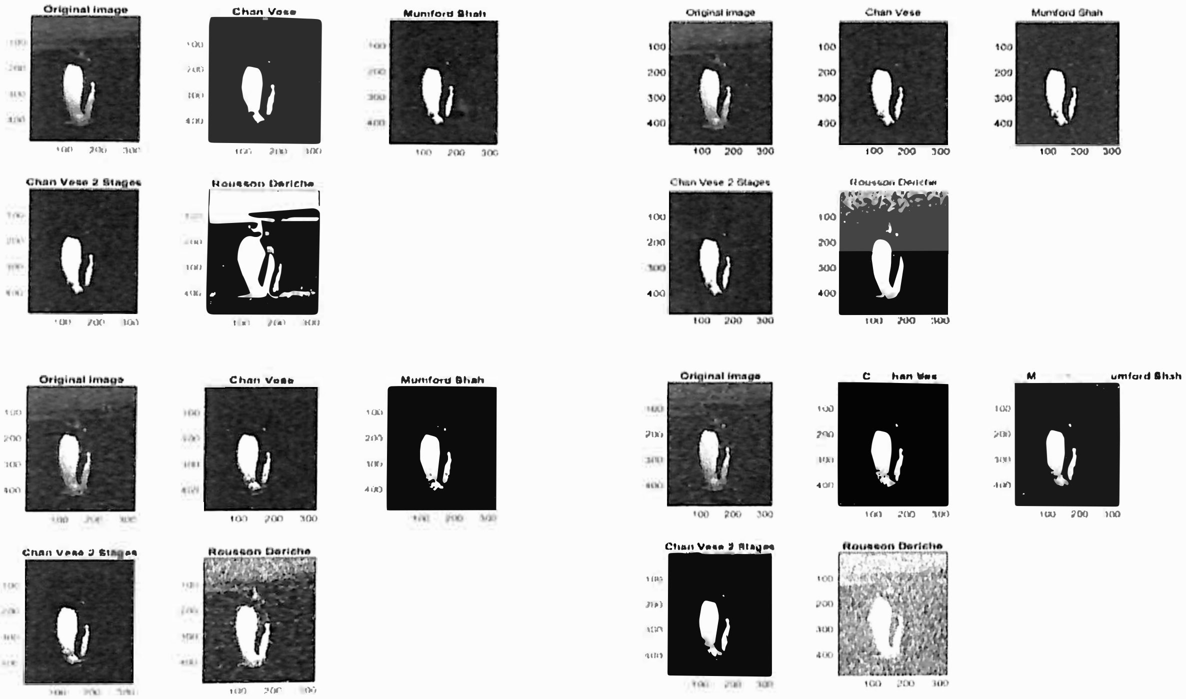


Figure 4.1: Example of Experiment 1: This figure shows the original and segmented images of a ping wing on a dark background from the BSD 200 data base and four different segmentations of the image. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmentation at 3 percent noise. The set of images in the bottom right was the segmented at 5 percent noise.

In Experiment 1, The Original Graph Cuts method cannot be compared to the

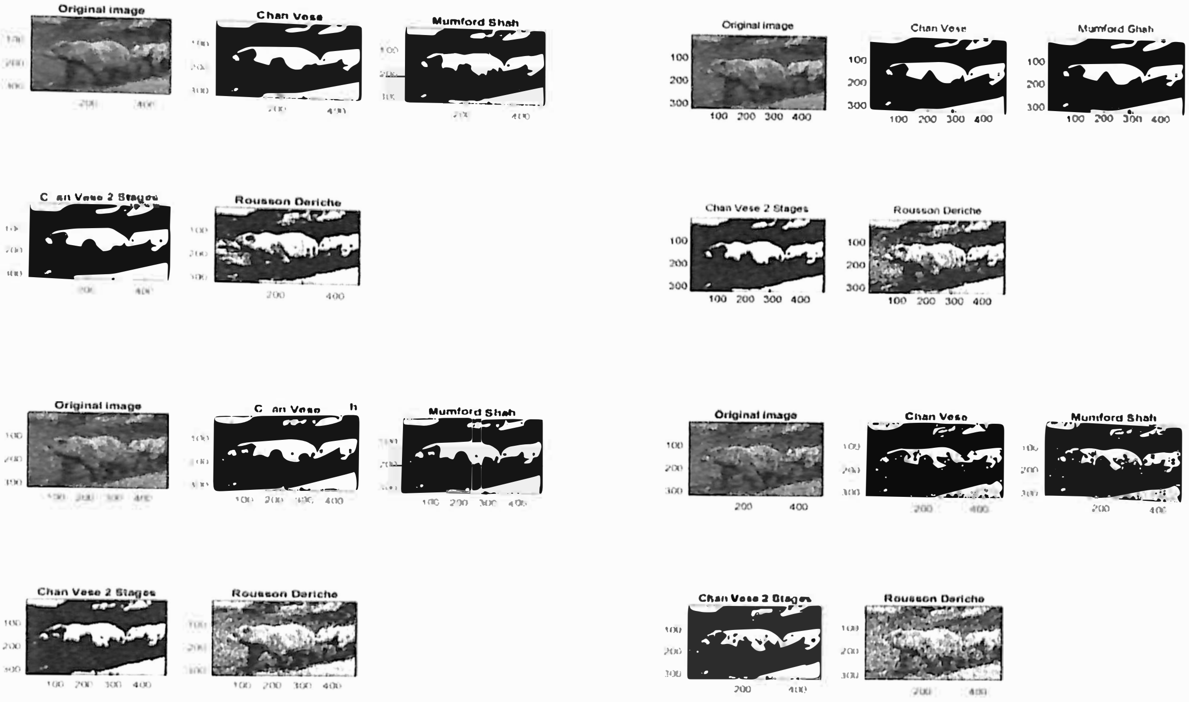


Figure 4.2: Example of Experiment 1: This figure shows the original and segmented images from the BSD 200 database of a couple of bears and four different segmentations of the image. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was segmented at 3 percent noise. The set of images in the bottom right was segmented at 5 percent noise.

other four methods: Chan Vese, Chan Vese Two Stage, Rousson Deriche and Mumford Shah. These four methods create two regions (foreground and background) whereas the Original Graph Cut Method creates a variable number of regions. Based on the results obtained in Table 4.1 and Figure 4.5 the Mean YLGC and Standard Deviation YLGC was identical for Chan Vese and Mumford Shah. This is because the Mumford Shah algorithm is equivalent to the Chan Vese algorithm when Mumford Shah has two regions. Chan Vese Two Stage segmentation cost is similar to the Chan Vese.

Rousson Deriche had the lowest segmentation cost at each noise level. Therefore Rousson Deriche method for graph cuts produced more accurate segmentation than the other methods according to YLGC. Overall as the noise level is increased in the Chan Vese, Chan Vese Two Stage, Rousson Deriche and Mumford Shah methods the accuracy of segmentation decreases. This is shown in Figures 4.1 and 4.2. Therefore these methods are more efficient when there is no noise present.

4.2 Experiment 2

Similar to Experiment 1, in Experiment 2 first we segmented 30 images from the GT database (`http://research.microsoft.com/en-us/um/cambridge/projects/visionimagevideoediting/segmentation/grabcut.htm`) using four different graph cut techniques (Chan Vese, Chan Vese two stage, Rousson Deriche and Mumford Shah). Then, we applied the Gaussian noise filter to all images in the database. The Gaussian filter was controlled at 0 mean and the variance was changed to 1 percent, 3 percent and 5 percent. All of the four methods were segmented in gray scale shown in Figures 4.7 and 4.8. Then, we compared the original image from the database to the segmented images. To compute segmentation accuracy we use YLGC error function. This helped to determine the efficiency of the segmentation. All of these methods segment the images into two regions foreground and background. For every image in the GT database there is a "perfect" segmented image that corresponds to the original image. In Table 4.2 and Figure 4.9 we reported the mean and standard deviation of YLGC over the GT data set for four different graph cut techniques.

The results obtained in Experiment 2 shown in Table 4.2 and Figure 4.9 indicate similar trends in segmentation accuracy as Experiment 1 with respect to noise level and selected method. Overall, the Rousson Deriche method produced lower segmentation costs than the other methods except for the case of no noise where it produces approximately equal YLGC to the other methods. In addition, as the noise level is increased the segmentation accuracy decreases for all methods. This is shown in Figures 4.7 and 4.8.

4.3 Experiment 3

In Experiment 3, first we segmented 20 pQCT (peripheral Quantitative Computed Tomography) images of the lower leg from the Shape modeling database using the Original Graph Cut method. We attempted to segment the images into five sections (air, muscle, fat, cortical bone and endostial bone) shown in Figures 4.10, 4.11 and 4.12. Then, we compared the original image from the database to the segmented images. To compute segmentation accuracy we use YLGC error function. In Table 4.3 we reported the mean and standard deviation of YLGC over the lower leg data for the original graph cut technique.

Based on the examples and results of Experiment 3, the Original Graph Cut technique was efficient in distinguishing the different components of the lower leg. The air, muscle, fat, cortical bone and endostial bone was successively segmented into different regions represented by different colors on the color map.

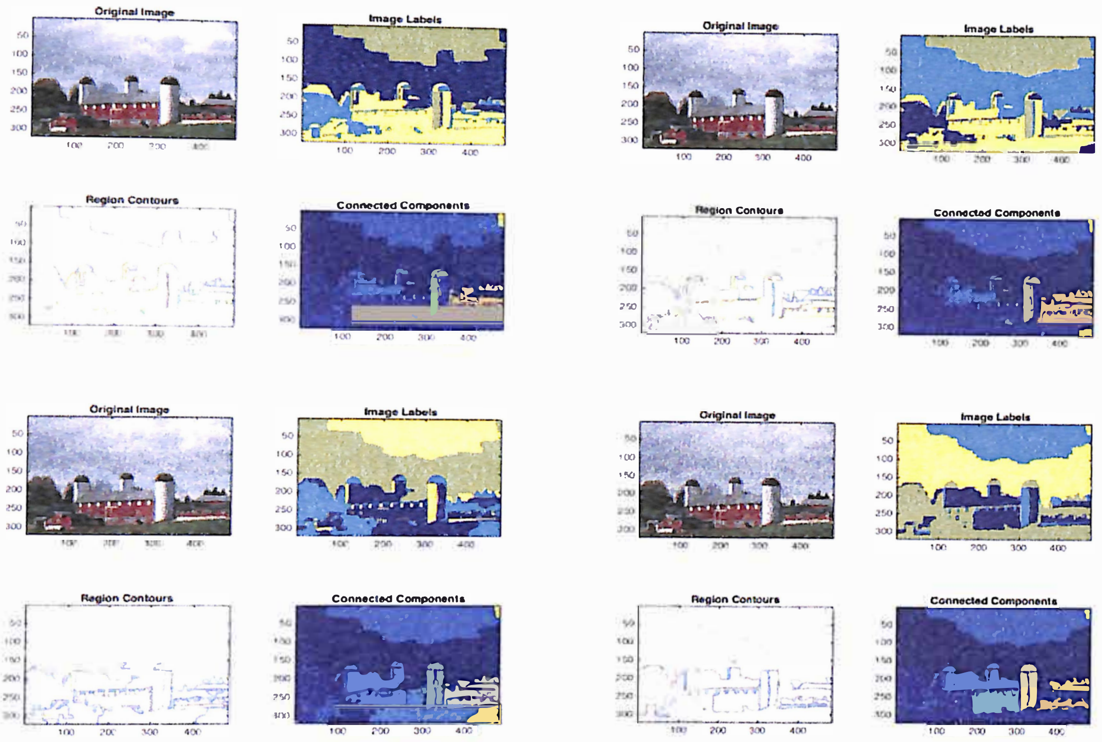


Figure 4.3: Example of Experiment 1: This figure shows different stages of the Original Graph Cut Methods of a building from the BSD 200 database. The first set of images in the top left was segmented at 0 percent noise. The second set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmentation at 3 percent noise. The set of images in the bottom right was segmented at 5 percent noise.

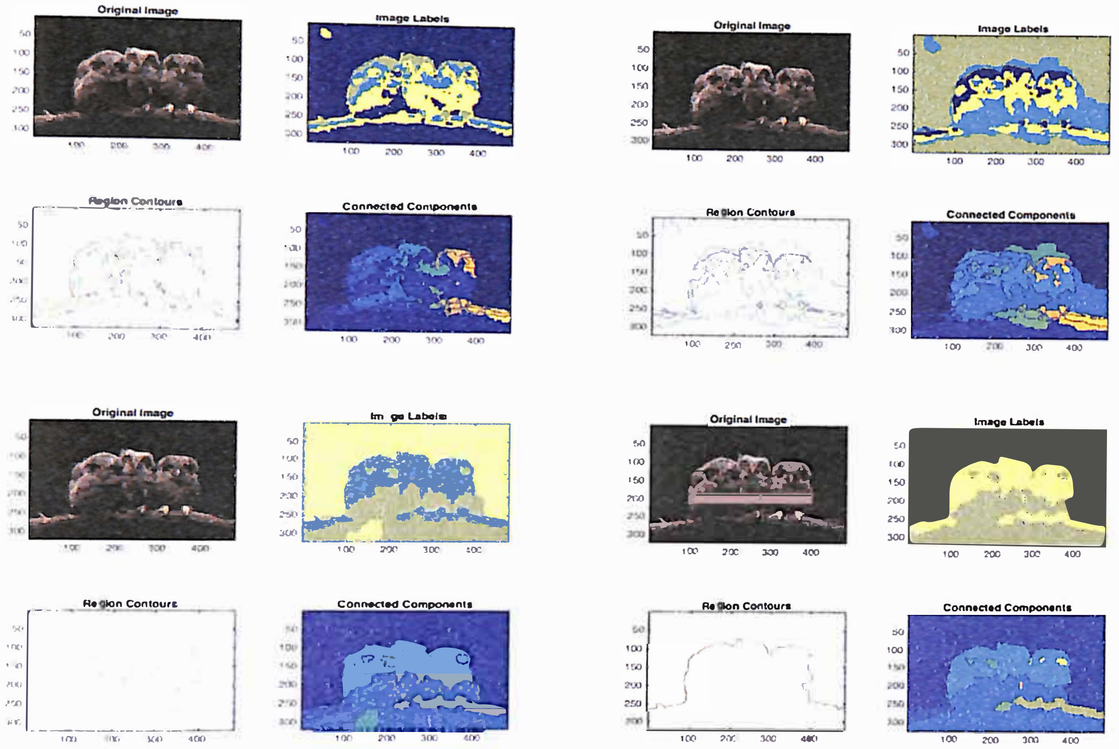
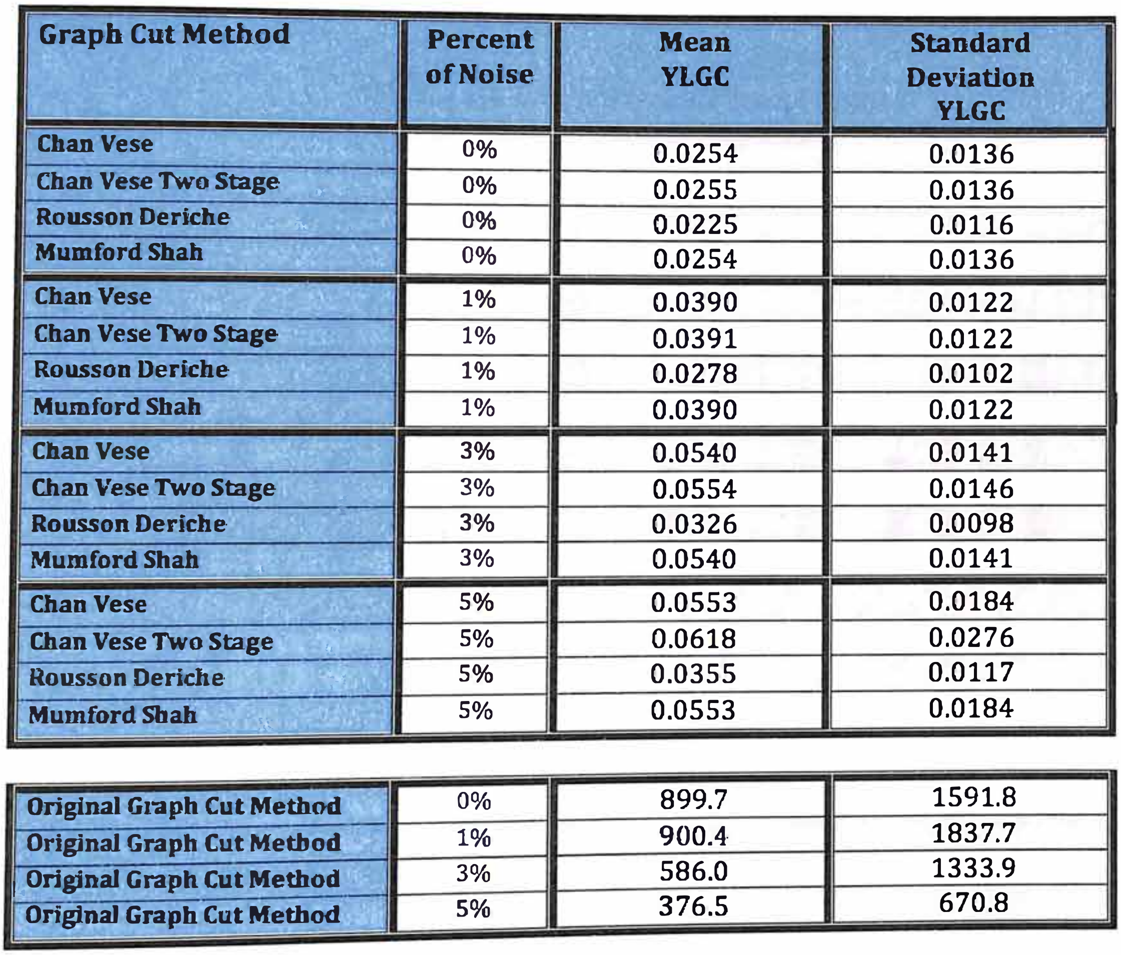


Figure 4.4: Example of Experiment 1: This figure shows different stages of the Original Graph Cut Method of three owls from the BSD 200 database. The first set of images in the top left was segmented at 0 percent noise. The second set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was segmented at 3 percent noise. The last set of images in the bottom right was segmented at 5 percent noise.



Graph Cut Method	Percent of Noise	Mean YLGC	Standard Deviation YLGC
Chan Vese	0%	0.0254	0.0136
Chan Vese Two Stage	0%	0.0255	0.0136
Rousson Deriche	0%	0.0225	0.0116
Mumford Shah	0%	0.0254	0.0136
Chan Vese	1%	0.0390	0.0122
Chan Vese Two Stage	1%	0.0391	0.0122
Rousson Deriche	1%	0.0278	0.0102
Mumford Shah	1%	0.0390	0.0122
Chan Vese	3%	0.0540	0.0141
Chan Vese Two Stage	3%	0.0554	0.0146
Rousson Deriche	3%	0.0326	0.0098
Mumford Shah	3%	0.0540	0.0141
Chan Vese	5%	0.0553	0.0184
Chan Vese Two Stage	5%	0.0618	0.0276
Rousson Deriche	5%	0.0355	0.0117
Mumford Shah	5%	0.0553	0.0184

Original Graph Cut Method	0%	899.7	1591.8
Original Graph Cut Method	1%	900.4	1837.7
Original Graph Cut Method	3%	586.0	1333.9
Original Graph Cut Method	5%	376.5	670.8

Table 4.1: Results of Experiment 1: Mean YLGC and standard deviation YLGC of five different graph cut techniques on BSD 200 database.

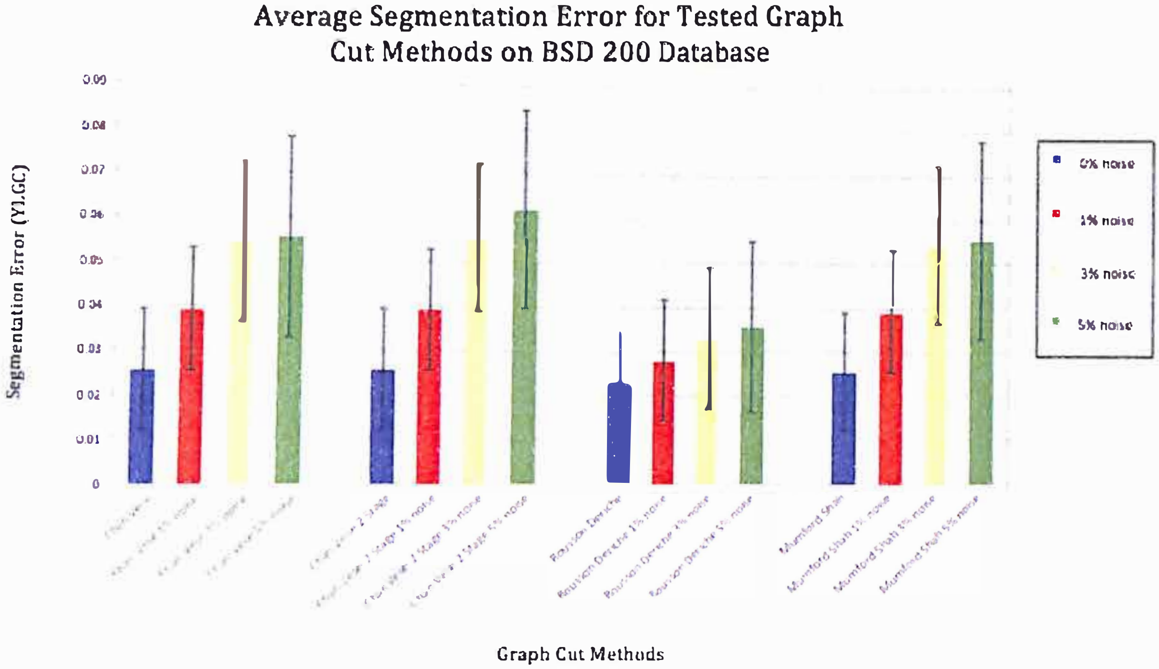


Figure 4.5: Results of Experiment 1: Graph corresponding to the results of four methods (Chan Vese, Chan Vese Two Stage, Rousson Deriche, Mumford Shah) on the BSD 200 database shown in Table 4.1 at different noise levels. The different bars represent the mean at that specific noise level and the black error bars represents the standard deviation.

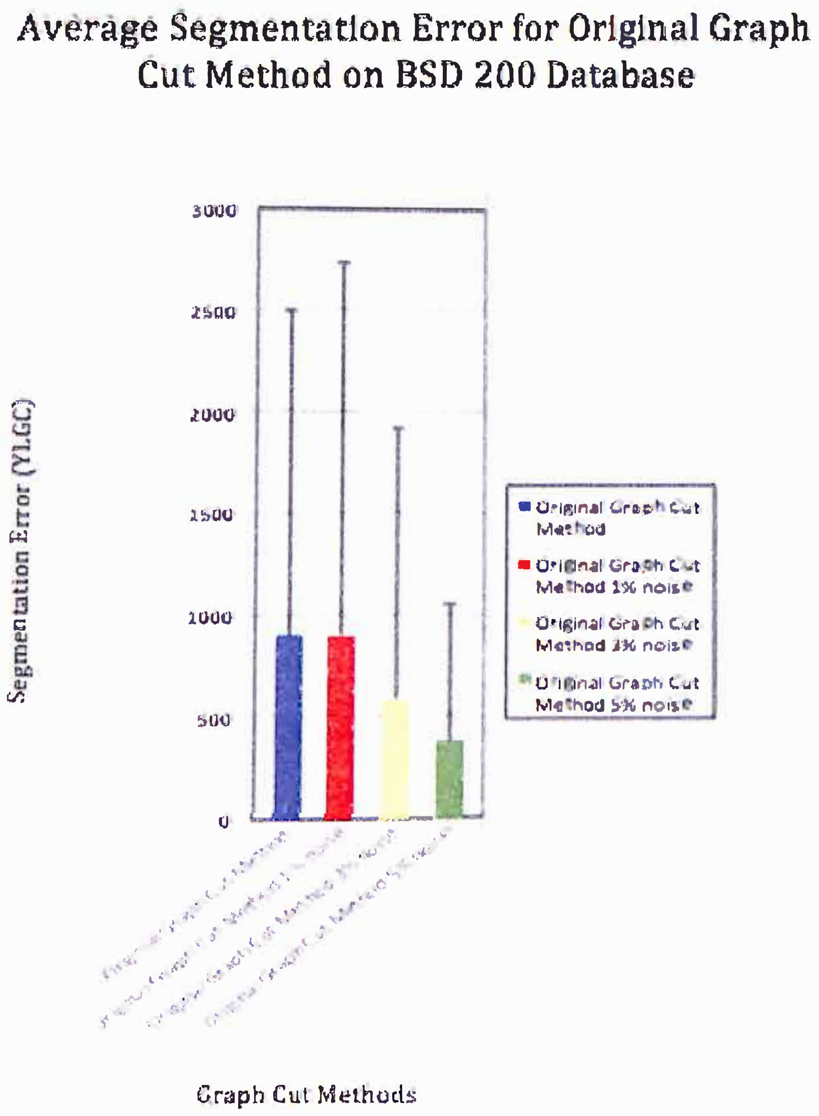


Figure 4.6: Results of Experiment 1: Graph corresponds to the results of the Original Graph Cut Method on the BSD 200 Database shown in Table 4.1 at different levels of noise. The different bars represent the mean at that specific noise level and the black error bars represents the standard deviation.

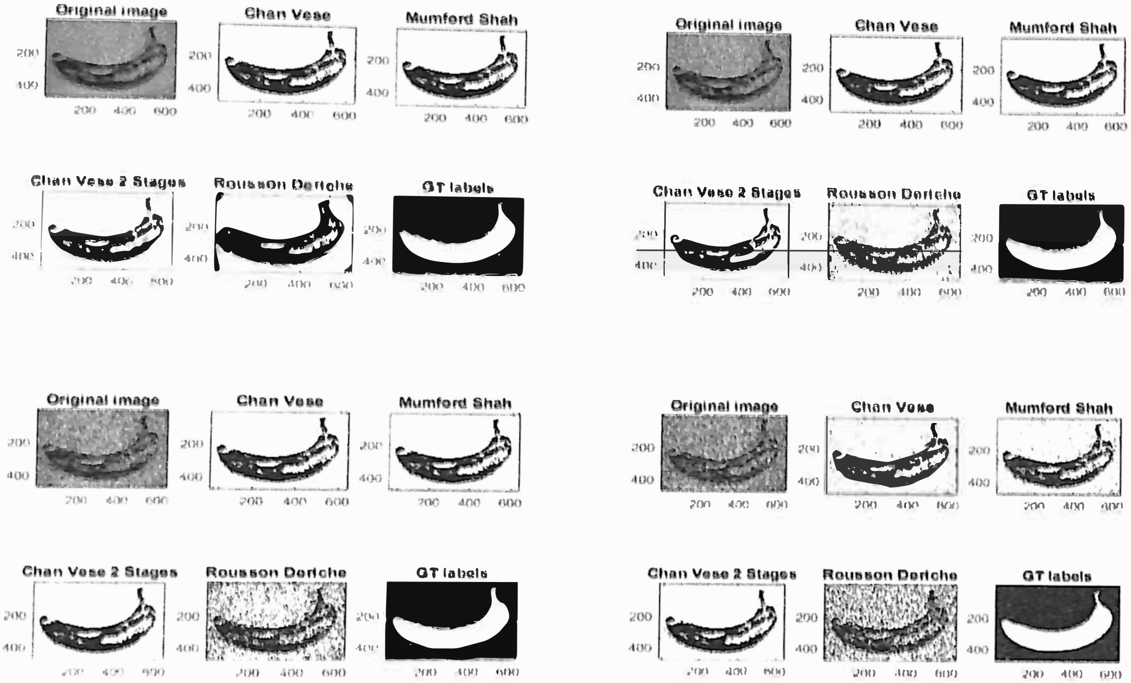


Figure 4.7: Example of Experiment 2: This figure shows the original and segmented image of a banana from the GT database and four different segmentations of the image. GT labels represents what a "perfect" segmentation should look like. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmented at 3 percent noise. The last set of images in the bottom right was segmented at 5 percent noise.

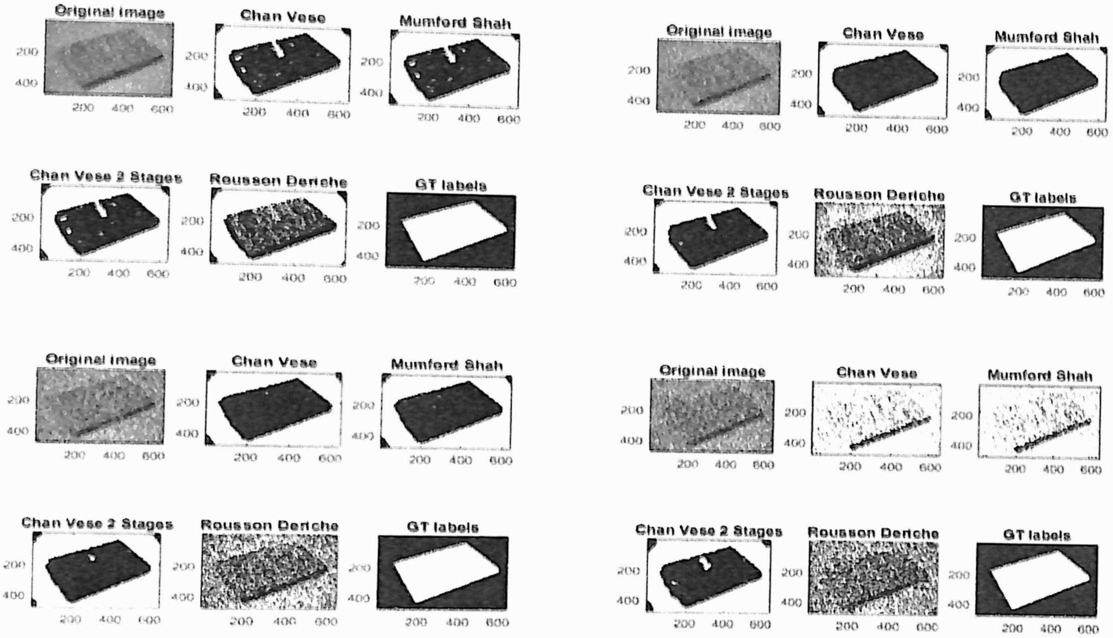
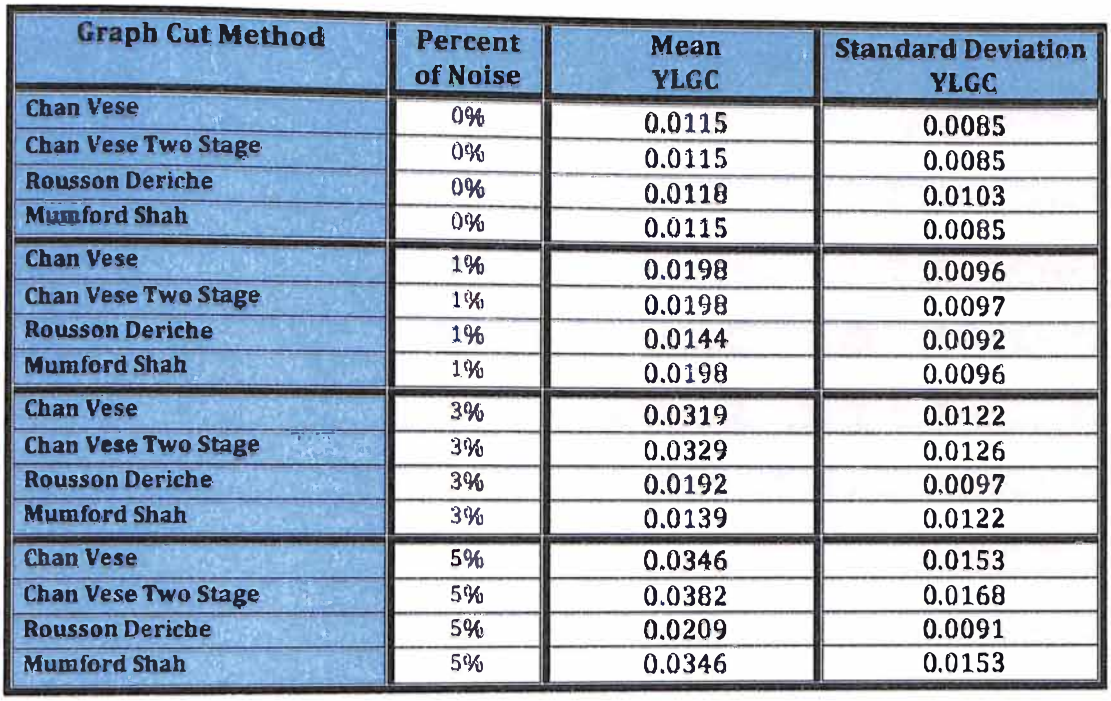
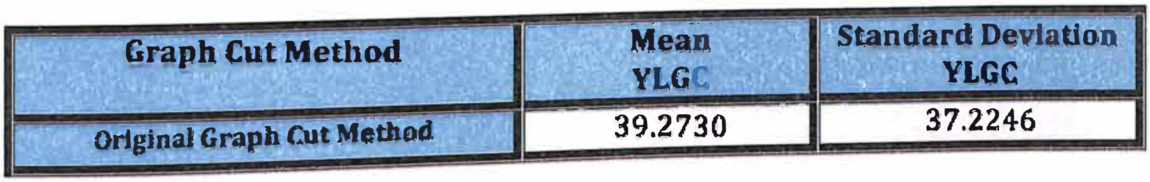


Figure 4.8: Example of Experiment 2: This figure shows the original and segmented image of a book on a Table from the GT database and four different segmentations of the image. GT labels represents what a "perfect" segmentation should look like. The first set of images in the top left was segmented at 0 percent noise. The set of images in the top right was the segmented at 1 percent noise. The set of images in the bottom left was the segmented at 3 percent noise. The set of images in the bottom right was the segmented at 5 percent noise.



Graph Cut Method	Percent of Noise	Mean YLGC	Standard Deviation YLGC
Chan Vese	0%	0.0115	0.0085
Chan Vese Two Stage	0%	0.0115	0.0085
Rousson Deriche	0%	0.0118	0.0103
Mumford Shah	0%	0.0115	0.0085
Chan Vese	1%	0.0198	0.0096
Chan Vese Two Stage	1%	0.0198	0.0097
Rousson Deriche	1%	0.0144	0.0092
Mumford Shah	1%	0.0198	0.0096
Chan Vese	3%	0.0319	0.0122
Chan Vese Two Stage	3%	0.0329	0.0126
Rousson Deriche	3%	0.0192	0.0097
Mumford Shah	3%	0.0139	0.0122
Chan Vese	5%	0.0346	0.0153
Chan Vese Two Stage	5%	0.0382	0.0168
Rousson Deriche	5%	0.0209	0.0091
Mumford Shah	5%	0.0346	0.0153

Table 4.2: Results of Experiment 2: Mean YLGC and standard deviation YLGC of four different graph cut techniques of 30 images from GT database.



Graph Cut Method	Mean YLGC	Standard Deviation YLGC
Original Graph Cut Method	39.2730	37.2246

Table 4.3: Results of Experiment 3: Mean YLGC and standard deviation YLGC of Original Graph Cut Technique on lower leg.

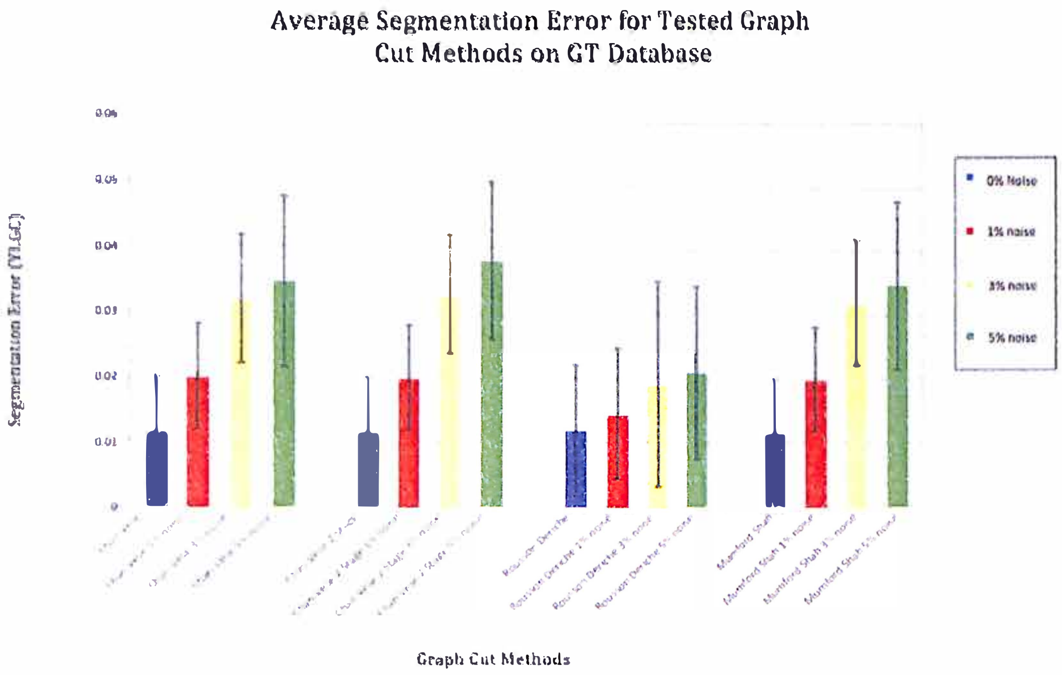


Figure 4.9: Results of Experiment 2: Graph corresponds to the four methods in Table 4.2. The different bars represent the mean at a specific noise level and the black error bars represents the standard deviation.

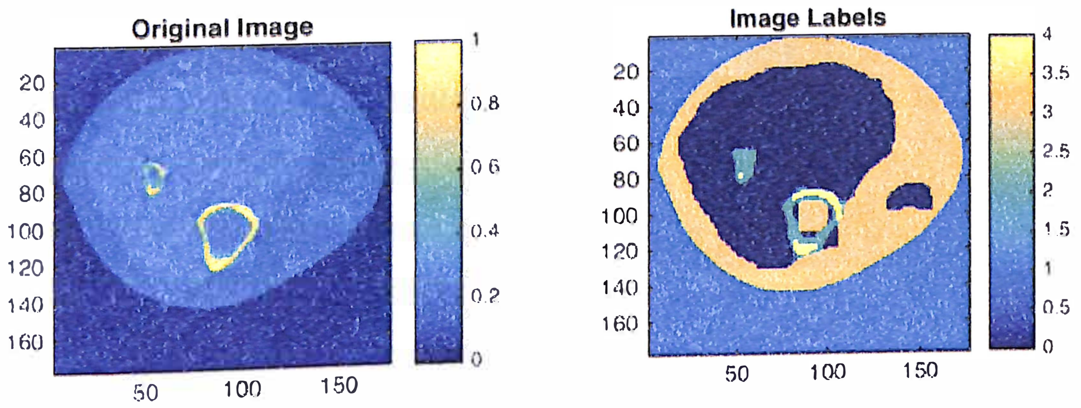


Figure 4.10: Example of Experiment 3: Shows the original and segmented image of the lower leg.

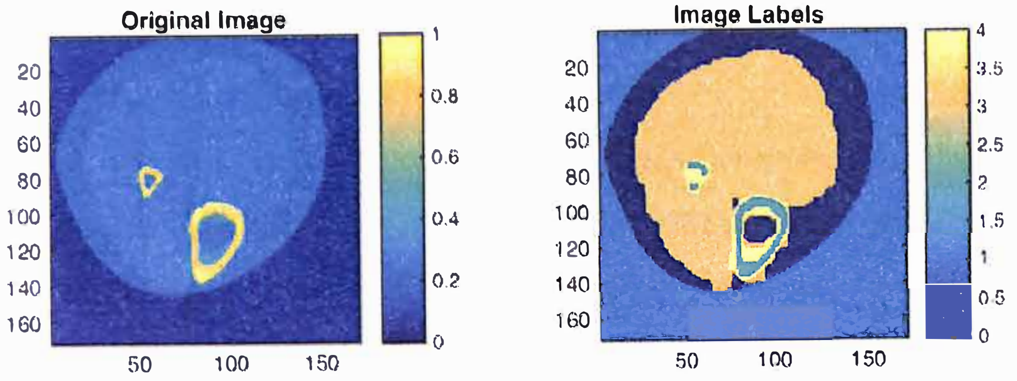


Figure 4.11: Example of Experiment 3: Shows the original and segmented image of the lower leg.

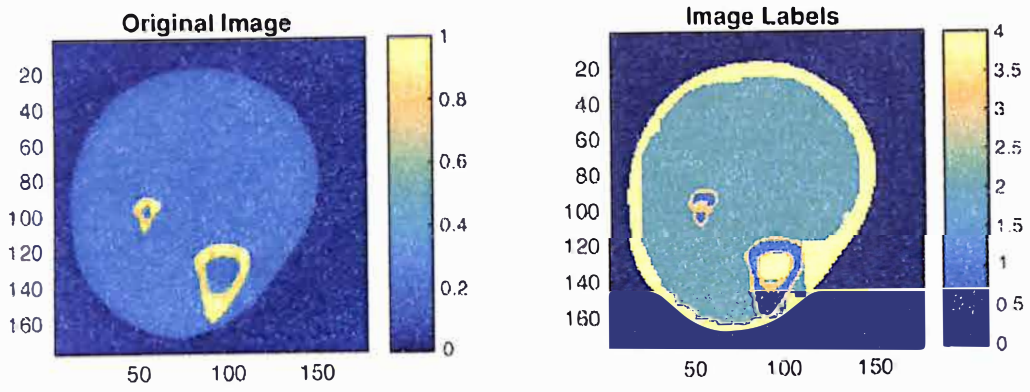


Figure 4.12: Example of Experiment 3: Shows the original and segmented image of the lower leg.

Chapter 5

CONCLUSION

In this thesis we presented Graph Cuts methods for image segmentation. Image segmentation is the process of extracting the main object of interest from the image. We evaluated segmentation accuracy over BSD 200, GT masks and lower leg medical images. The method used to evaluate segmentation accuracy was the YLGC error function. Graph Cuts segmentation techniques produce useful segmentation results for standard and medical images. Different methods work better for different images. Some methods work better on noisy images and some methods work better when there is no noise on an image.

This work can be extended in the following ways: 1. develop different objective functions for optimization, 2. apply graph cuts to regions entities instead of pixels to improve segmentation accuracy, 3. include region based texture and color features in object and background prototypes, and 4. segment different organs and anatomical structures in medical imaging data.

REFERENCE LIST

[1] B. R. Dandu A. Chopra. Image segmentation using active contour model. *International Journal of Computational Engineering Research / ISSN:2250-3005*, 2:819–822, 2012.

[2] I. Laszio I. Fekete B. Dezso, R. Giachetta. Experimental study on graph-based image segmentation methods in the classification of satellite images. *EARSel eProceedings*, 11:12–24, 2012.

[3] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on pattern analysis and machine intelligence*, 26:359–374, 2001.

[4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, November 2001.

[5] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Img. Proc.*, 10(2):266–277, February 2001.

[6] Ondej Dank and Pavel Matula. Graph cuts and approximation of the euclidean metric on anisotropic grids. In *VISAPP International Conference on Computer Vision Theory and Applications*, pages 68–73, Portugal, 2010. Institute for Systems and Technologies of Information, Control and Communication. Udleno ocenn Best Student Paper Award.

[7] P.F. Felzenszwalb and D.P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[8] V. Ganesan. Minimum spanning trees.

[9] Andrew V. Goldberg and Robert E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, October 1988.

[10] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition).* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[11] A. Rutten M. Prokop M. A. Viergever I. Isgum, M. Staring and B. van Ginneken. Multi-atlas-based segmentation with local decision fusion-application to cardiac and aortic segmentation in ct scans. *IEEE Trans Med Imaging*, 28(7):1000–1010, July 2009.

[12] Pushmeet Kohli and Philip H. S. Torr. Dynamic graph cuts for efficient inference in markov random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12):2079–2088, December 2007.

[13] Matthew J. McAuliffe, Francois M. Lalonde, Delia McGarry, William Gandler, Karl Csaky, and Benes L. Trus. Medical image processing, analysis & visualization in clinical research. In *Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems*, CBMS '01, pages 381–, Washington, DC, USA, 2001. IEEE Computer Society.

[14] Shah J. Mumford, D. Optimal approximation by piecewise smooth functions 359 360 and associated variational problems. *Communications on Pure and Applied Mathematics*, 42:577–684, 1989.

[15] N. Ahuja N. Xu and R. Bansal. Object segmentation using graph cuts based active contours. *Comuter Vision and Image Understanding*, 107(3):210–224, 2007.

[16] H. F. Neemuchwala. Entropic graphs for image registration. Master's thesis, 2005.

[17] O.Demetz. Graph cuts in computer vision, year=2009,. Master's thesis.

[18] M. Rousson and R. Deriche. A variational framework for active and adaptative segmentation of vector valued images. *Proceedings of IEEE workshop on Motion and Video computing, 2002, pp.56-61*, pages 56–61, 2002.

[19] G. Economou S. Makrogiannis and S. Fotopoulos. A region dissimilarity relation that combines feature-space and spatialinformation for color image segmentation. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,*, 35(1):44–53, 2005.

[20] K. W. Fishbein C. Schreiber L Ferrucci S. Makrogiannis, S. Serai and R.G Spencer. Automated quantification of muscle and fat in the thigh from water-,fat-, and nonsuppressed mr images. *Journal of Magnetic Resonance Imaging*, 35(5):1152–1161, 2012.

[21] S. Fotopoulos S. Makrogiannis, G. Economou and N.G. Bourbakis. Segmentation of color images using multiscale clustering and graph theoretic region synthesis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* ., 35(2):224–238, 2005.

[22] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000.

[23] A. M. Sirghie. Abdominal fat segmentation using graph cut methods, 2012. DTU supervisors: Rasmus Larsen, Professor, rl@imm.dtu.dk, DTU Informatics, Knut Conradsen, Professor, kc@imm.dtu.dk, DTU Informatics.

[24] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision.* Thomson-Engineering, 2007.

[25] George Stockman and Linda G. Shapiro. *Computer Vision.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[26] P.V.S.S.R.C Mouli T. N. Janakiraman. Image segmentation based on minimal spanning tree and cycle. *IEEE.*, pages 215–219, 2007.

[27] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *Int. J. Comput. Vision*, 50(3):271–293, December 2002.

[28] Xiang Zeng, Wei Chen, and Qunsheng Peng. Efficiently solving the piecewise constant mumford-shah model using graph cuts. Technical report, 2006.

CURRICULUM VITAE

Shana Kai Brown

Education

Masters of Science: Fall 2014- Spring 2016, Department of Mathematical Sciences, Delaware State University, Dover, Delaware. Major in Applied Mathematics.

Bachelor of Science: Spring 2012- Spring 2014, Department of Mathematical Sciences, Delaware State University, Dover, Delaware. Major in Applied Mathematics.

Deans List Each Semester

Awards and Presentations

The Mathematical Association of America. "Graph Cut Energy Minimization Techniques for Image Segmentation", November 14, 2015.

LSAMP Research Symposium. "Soliton Solutions and Differentiable Equations", April 2014

Other Experience

Miss Mathematics, August 2013- May 2014

Math Club Vice President, August 2013- May 2014

Math Club Treasure, August 2012- May 2013

LSAMP Program, March 2013- May 2016

SMILE Program, March 2013- May 2016

Computer Skills

C++

Matlab

Latex

MS-Office

Maple

Python