

OBJECT SEGMENTATION IN HYPERSPECTRAL IMAGES USING GRAPH CUTS BASED ON ACTIVE CONTOURS

by

Susi Huamán-De la Vega

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2010

Approved by:

Néstor J. Rodríguez, PhD
Member, Graduate Committee

Date

José A. Borges, PhD
Member, Graduate Committee

Date

Vidya Manian, PhD
President, Graduate Committee

Date

Darrell Hajek, PhD
Representative of Graduate Studies

Date

Hamed Parsiani, PhD
Chairperson of the Department

Date

Abstract of Thesis Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Master of Science

OBJECT SEGMENTATION IN HYPERSPECTRAL IMAGES USING GRAPH CUTS BASED ON ACTIVE CONTOURS

by

Susi Huaman-De la Vega

2010

Chair: Vidya Manian

Major Department: Electrical and Computer Engineering

The interest in object segmentation on hyperspectral images is increasing and many approaches have been proposed to deal with this area. In this project, we develop an algorithm that combines both the active contours and the graph cut approaches for object segmentation in hyperspectral images. The active contours approach has the advantage of producing sub-regions with continuous boundaries. The graph cuts approach has emerged as a powerful optimization technique for minimizing energy functions while avoiding the problems of local minima inherent in other approaches. The combination of the two models has robust object segmentation capability because it has the ability to avoid the local minima and provide a more global result. Additionally, graph cuts guarantee continuity and produce smooth contours, free of self-crossing and uneven spacing problems. Our approach uses both spatial information and spectral information from hyperspectral images and it can segment more than one object in an image. We tested our algorithm using real and synthetic hyperspectral images, and obtained good results. This algorithm can be applied in many fields and it should represent an important advance in the field of object segmentation.

Resumen de Tesis Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Maestría en Ciencias

SEGMENTACION DE OBJETOS EN IMÁGENES HIPERESPECTRALES USANDO EL CORTE DE GRAFOS BASADO EN CONTORNOS ACTIVOS

Por

Susi Huamán De la Vega

2010

Consejero: Vidya Manian

Departamento: Ingeniería Eléctrica y Computadoras

El interés en la segmentación de objetos sobre imágenes hiperespectrales está aumentando y muchos enfoques han sido propuestos para tratar con esta área. En este proyecto, nosotros desarrollamos un algoritmo que combina los enfoques de los contornos activos y el corte de grafos para la segmentación de objetos en imágenes hiperespectrales. El enfoque de los contornos activos tiene la ventaja de producir sub-regiones con fronteras continuas. El enfoque del corte de grafos ha emergido como una poderosa técnica de optimización para la minimización de funciones de energía y evitar los problemas de mínimos locales inherentes en otros enfoques. La combinación de los dos modelos tiene una capacidad robusta de segmentación de objetos, porque este tiene la habilidad de eliminar los mínimos locales y proveer un resultado mucho más global. Adicionalmente, el corte de grafos garantiza la continuidad y produce contornos suavizados, libres de los problemas de auto-cruces y espaciado asimétrico. Nuestro enfoque utiliza tanto la información espacial y la información espectral de las imágenes hiperespectrales y este puede segmentar más de un objeto en la imagen. Nosotros probamos nuestro algoritmo usando imágenes hiperespectrales

reales y sintéticas, y obtuvimos buenos resultados. Este algoritmo puede ser aplicado en muchos campos y esto podría representar un importante avance para el campo de la segmentación de objetos.

To GOD...
To my family...
And to my friends.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Vidya Manian for her support and guidance in the development of the present thesis. I also want to thank all the members of my graduate committee: Dr. José Borges, Dr. Néstor Rodríguez, and the representative of graduate studies: Dr. Darrell Hajek for his help in the final version of my thesis. Thanks to Dr. Pedro Rivera who substituted Dr. Néstor Rodríguez during my thesis defense. Additionally, I am very grateful to Dr. Edgar Acuña for his support and help when I was studying in the Computer Science Program.

This work was supported in part by the Department of Defense under grant HM1582-08-1-0047 and by Gordon-CenSSIS, the Bernard M. Gordon Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award Number EEC-9986821).

Table of Contents

ABSTRACT OF THESIS PRESENTED TO THE GRADUATE SCHOOL.....	II
RESUMEN DE TESIS PRESENTADO A ESCUELA GRADUADA	III
ACKNOWLEDGEMENTS.....	VI
TABLE OF CONTENTS.....	VII
FIGURE LIST	VIII
1 INTRODUCTION.....	1
1.1 OBJECTIVES.....	3
1.2 CONTRIBUTIONS OF THIS WORK	3
1.3 THESIS OVERVIEW	4
2 THEORETICAL BACKGROUND.....	5
2.1 HYPERSPECTRAL IMAGE.....	5
2.2 IMAGE SEGMENTATION.....	7
2.3 ACTIVE CONTOURS	8
2.4 GRAPH CUTS.....	9
2.5 RELATED WORK.....	13
3 METHODOLOGY	16
3.1 PROPOSED ALGORITHM DESCRIPTION	16
3.2 RUN TIME ANALYSIS	25
3.3 JUSTIFICATION OF THE CHANGE MADE	26
4 TESTING AND EVALUATION	27
4.1 DATA SET DESCRIPTION	27
4.2 EXPERIMENTAL RESULTS	28
4.2.1 Results of segmenting images with holes	31
4.2.2 Results of segmenting more than one object	31
4.2.3 Results of segmenting small objects	32
4.2.4 Results of segmenting large images.....	34
4.2.5 Results of segmenting images with noise	37
4.2.6 Results of segmentation after preprocessing the image.....	40
4.2.7 Comparison with original algorithm.....	43
4.2.8 Comparisons with segmentation band by band	45
4.2.9 Comparisons between segmentation and classification results.....	46
5 CONCLUSIONS AND FUTURE WORK	53

Figure List

Figures	Page
Figure 2-1: Spectral variation of three different components: Soil, Water and Vegetation (taken from [9]).....	6
Figure 3-1: Block diagram for object segmentation in hyperspectral Images.....	17
Figure 3-2: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color image from ENVI, (b) Gray scale of the hyperspectral image, and (c) Image formed by SAD.....	19
Figure 3-3: (a) Binary image obtained based on SAD with 15 thresholds value, and (b) Initial boundary for images 3.3 (a).....	20
Figure 3-4: Dilation process. (a) Size of CN=3. (b) Size of CN=10.....	22
Figure 3-5: Vertices in inner, current, and outer boundaries.....	22
Figure 3-6: New Boundary Computed.....	23
Figure 3-7: Node identification. (a) v_1, v_2, v_3 are merged into a new node v . (b) Self loops are deleted and parallel edges are replaced by a single edge. (Taken from [6]).....	24
Figure 4-1: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) 7 points of the object of interest (bands 54 35 15 RGB), (b) Initial contour with threshold=0, (c) Initial contour with threshold=66, (d) Initial contour with threshold=10. ...	29
Figure 4-2: Region selected from HYDICE Terrain sample scene (bands 168 93 85 RGB). Segmentation generated using 5 points, threshold=9 for both images, and (a) CN=3; (b) CN=7 and c) Merging segmentation result for (a) and (b).	29
Figure 4-3: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 67 70 16 RGB) using threshold=10 and CN=3. (a), (b) and (c) Entry points; (d), (e) and (f) Segmentation results for (a), (b) and (c) respectively.	30
Figure 4-4: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 5 points of the object of interest, threshold=10 and CN=5.....	31
Figure 4-5: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 40 35 15 RGB) using 10 points for the first object(blue) and 5 point for the second object(red), for both threshold=10 and CN=3.....	32
Figure 4-6: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color images from ENVI, and (b) Segmentation generated (bands 55 70 16 RGB) using Threshold=10; (a) 5 points and CN=5 for the first object (blue); and 4 point and CN=2 for the second object (red).	32
Figure 4-7: Region selected from HYDICE Forest. Segmentation generated (bands 49 37 18 RGB) using 6 points of the object of interest, threshold=10 and (a) CN=3. (b) CN=5.....	33
Figure 4-8: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 6 points of the object of interest, threshold=10 and CN=2.....	34

Figure 4-9: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 10 points of the object of interest, threshold=11 and CN=4.....	35
Figure 4-10: Regions selected from Washington D.C. Mall. Segmentation generated (bands 177 171 136 RGB) using 15 points of the object of interest, threshold=16 and CN=3.....	36
Figure 4-11: Regions selected from “Enrique Reef data” captured by AVIRIS. Segmentation generated (bands 50 100 16 RGB) using 5 points of the object of interest, threshold=10 and CN=2.....	37
Figure 4-12: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 5 points of the object of interest, threshold=10 and CN=5. (a) Result of the original images, (b) Result adding noise with n=10, (c) Result adding noise with n=100.....	38
Figure 4-13: (a) Combination of segmentation results from Figure 4-12 (a, and b), (b) Combination of segmentation result from Figure 4-12 (a and c).....	39
Figure 4-14: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 40 35 15 RGB) using 5 points, threshold=3 and CN=3 (a) Result of the original images, (b) Result adding Gaussian noise with n=100.....	39
Figure 4-15: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated using 8 points, threshold=8 and CN=3. (a) Result for original images with 128, (b) Result for images with 20 bands, and (c) Result for images with 10 bands.....	41
Figure 4-16: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color from ENVI, (b) Common pixels belonging to segmentation result for images from Figure 4-15, and (c) Combination of segmentation result from Figure 4-15.....	42
Figure 4-17: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated using 8 points, threshold=10 and CN=3 (a) Result for original images with 120, (b) Result for images with 10 bands, and (c) Result for images with 5 bands.....	43
Figure 4-18: Region selected from Fake Leaves hyperspectral image captured by SOC-700 (bands 40 35 15 RGB). (a) and (b) Initial contour, (c) Input point; (d) and (e) Segmentation result using original algorithm for Figure 4-18 ((a) and (b)) respectively; (f) Segmentation result using our approach with threshold=10 and CN=2.....	44
Figure 4-19: Regions selected from Washington D.C. Mall (bands 181 189 179 RGB). (a) and (b) Initial contour, (c) Input points; (d) and (e) Segmentation result using original algorithm for Figure 4-19 ((a) and (b)) respectively; (f) Segmentation result using our approach with threshold=16 and CN=5.....	45
Figure 4-20: Region selected from Fake Leaves hyperspectral image captured by SOC-700 (bands 54 36 16 RGB) Segmentation generated using 8 points, threshold=11 and CN=4 for all images. (a) Result for more than 25% fitting in the all segmentation results. (b) Result for more than 35% fitting in the all segmentation results. (c) Result for more than 50% fitting in the all segmentation results. (d) Segmentation result for our approach.....	46
Figure 4-21: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 10 points of the object of interest, threshold=10 and CN=11.....	47

Figure 4-22: Training classes for the classification.	48
Figure 4-23: Result of the image classification using supervised angle detection for Figure 4-21.	49
Figure 4-24: Result of the image classification using Mahalanobis distance for Figure 4-21.	50
Figure 4-25: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 57 36 13 RGB) using 8 points for the first object(round) and 7 point for the second object(rectangular), for both threshold=10 and CN=3.	51
Figure 4-26: Training classes for the classification.	51
Figure 4-27: Result of the image classification using supervised angle detection for Figure 4-25.	52
Figure 4-28: Result of the image classification using Mahalanobis distance for Figure 4-25.	52

1 INTRODUCTION

The interest in hyperspectral image analysis is increasing. Hyperspectral images are playing an important role in many fields. One of the reasons is because the hyperspectral images provide ample spectral information to identify and distinguish between spectrally similar but unique materials. Consequently, hyperspectral imagery has the potential for extraction of more accurate and detailed information [1] than the classic images.

Hyperspectral image data is increasingly available from a variety of sources, including commercial and government satellites; this is accompanied by an increase in spatial resolution and in the number of spectral channels [2]. Hyperspectral imaging technology has recently found applications in many fields, including agriculture, archaeology, biology, defense, forensics, medicine, pharmaceuticals and remote sensing, as well as surveillance for both military and civilian applications [3].

One important task in image analysis is segmentation, by which we mean the partitioning or dividing of an image into multiple sub-regions or parts according to their properties, e.g. intensity, color, and texture. Image segmentation consists of dividing an image into regions that presumably represent objects. These regions either have some measure of homogeneity within themselves, or have some measure of contrast with nearby objects or boundaries [4] [5].

Many approaches have been proposed and developed for hyperspectral image segmentation [6] [7]. This thesis presents an algorithm that uses both active contours and graph cut approaches for object segmentation. The object segmentation algorithm proposed

in [8] is improved and extended to hyperspectral images. The approach presented here works with different types of hyperspectral images. Moreover, most algorithms for object segmentation segment only one specific object from its background [8] [9], whereas this method can segment more than one object based on the measure of similarity between two pixels.

The primary aim of this work is to improve the efficiency of object segmentation results in hyperspectral images. We do this by using both spatial and spectral information. Our approach produces results with enough visual accuracy and versatility that it can be applied in many fields and it should represent an important advantage for the segmentation field in hyperspectral images.

Our method has the following advantages:

- It has the ability to avoid the local minima and provide a more global result.
- The incorporation of graph cuts in the technique guarantees continuity and leads to smooth contours, free of self-crossing and uneven spacing problems.
- It has the ability to segment more than one object.
- It has the ability to segment objects with holes.
- It has the ability to segment small objects.
- Our application can be used with different types of hyperspectral images taken from a variety of sensors.
- The results from this application can be used in a variety of applications in different fields.

1.1 Objectives

The main objective of this thesis was to develop an application to extend and improve the object segmentation using Graph Cuts Based Active Contours (GCBAC) developed in [8] so that it could be applied with hyperspectral images. We developed an application for object segmentation. It was our intention to:

- Create an application which could segment more than one object.
- Develop an algorithm that can be applied without modification to many types of hyperspectral images.
- Test and validate our algorithm, experimenting with different sets of hyperspectral images acquired by different sensors.
- Report on the result of our tests and validation.

1.2 Contributions of this work

In this thesis, the object segmentation algorithm for 2-d and 3-d images proposed in [8] is improved and extended to hyperspectral images. In some cases an object segmentation algorithm works only with a specific hyperspectral image, and when someone needs to apply same algorithm with another image it is necessary to adopt it to the new image. In some cases it is not feasible to do this. Our application works with many hyperspectral images data sets taken from different sensors.

Moreover, algorithms for object segmentation often segment only one specific object from its background [8] [9] whereas our application can segment multiple objects. Another benefit of this work is that our application has a simple, intuitive interface that helps users.

Sometimes it is necessary to segment the entire hyperspectral image in order to find a particular object. This requires a lot of time and work. With this application, the user can segment only the objects of interest, thereby saving time and effort.

1.3 Thesis Overview

Chapter two contains a literature review in which we describe the background theory of hyperspectral images, the active contour approach, the graph cut approach, and image segmentation. All of these topics are important for understanding and developing the proposed application. Other related works are also described.

Chapter three presents the methodology used to develop the application for object segmentation in hyperspectral image. We explain step by step what the application does.

Chapter four shows the segmentation results using different input parameters and data sets of several hyperspectral images.

Finally, chapter five contains our conclusions and suggestions for future work.

2 THEORETICAL BACKGROUND

This chapter explains the theory related to object segmentation in hyperspectral images. We give a brief description of the important concepts in order to help to understand the proposed application. We conclude this chapter discussing related work.

2.1 Hyperspectral Image

A hyperspectral image is, in essence, a picture that contains both spatially and spectrally continuous data. It is typically collected and represented as a data cube with spatial information collected in the x-y plane, and spectral information represented in the z-direction.

The spatial resolution determines the detail of information obtained and the total area imaged by the sensor. The detail discernible in an image is dependent on the spatial resolution of the sensor and refers to the size of the smallest possible feature that can be detected [10]. If the pixels are too large, then multiple objects are captured in the same pixel and become difficult to identify. If the pixels are too small, then the energy captured by each sensor-cell is low, and the decreased signal-to-noise ratio reduces the reliability of measured features.

The spectral resolution taken from sensors depends on their precision, which is the width of each band of the spectrum that is captured. Spectral resolution describes the ability of a sensor to distinguish fine wavelength intervals. If the scanner identifies a large number of fairly narrow frequency bands, it is possible to identify objects even if said objects are only captured in a handful of pixels [11]. The finer the spectral resolution, the narrower the

wavelength's range for a particular channel or band. Thus, it can represent features of various colors based on their reflectance in each of these distinct wavelength ranges.

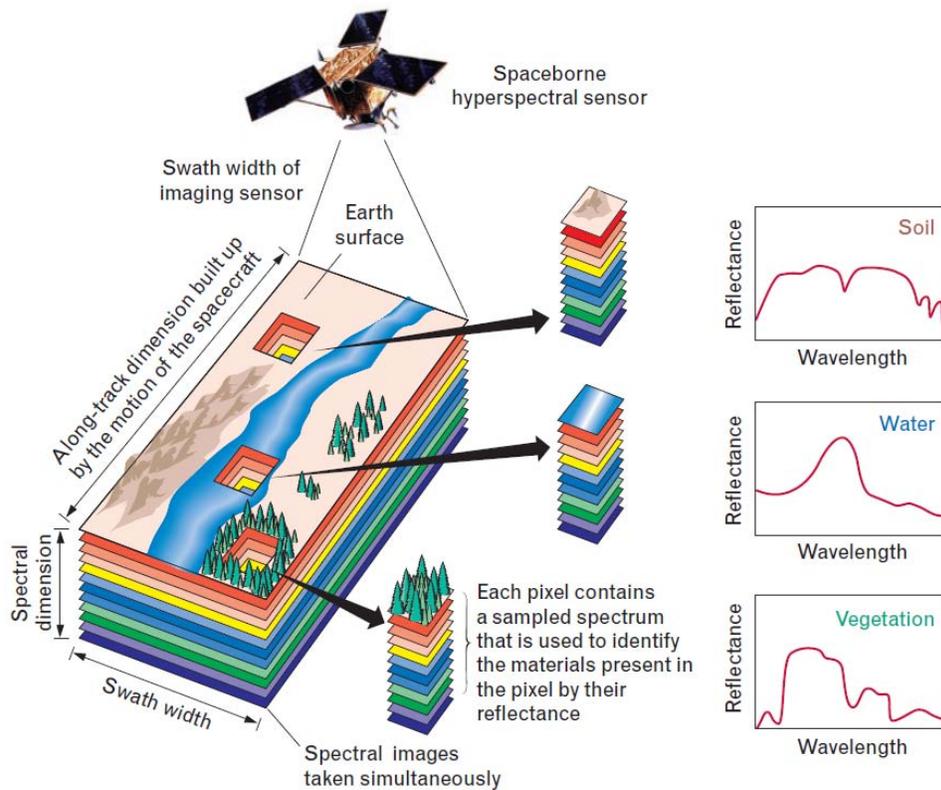


Figure 2-1: Spectral variation of three different components: Soil, Water and Vegetation (taken from [12]).

Hyperspectral images provide ample spectral information to identify and distinguish between spectrally similar but unique materials. Consequently, hyperspectral imagery provides the potential for more accurate and detailed information extraction than is possible with other types of remotely sensed data.

The study of hyperspectral images is becoming important in a number of areas. There are certain issues that must be considered when working with hyperspectral images. These

include the problems that arise from the acquisition of data by heterogeneous sensory systems and the massive data sets to be processed. The amounts of data imply a need for efficient encoding in image transmission and archiving, and the need for efficient algorithms for data processing and visualization. There are products that work with the problem of efficient data processing and visualization. Among these are ENVI [13], which is a commercial product; or MultiSpec [14], which is a freeware product. These programs are product of long-term research and development projects.

2.2 Image Segmentation

Image segmentation aims to extract regions by dividing an image into disjoint sets of pixel segments. Segmentation is a process for separating an image into its constituent regions based on a few properties, e.g. intensity, color, or texture. Segmenting an image consists of partitioning the image into homogeneous regions. These regions have similar characteristics or properties [15].

Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels located in the same region or contour is similar with respect to some characteristic or property [16].

There are many different techniques of image segmentation [6] [7], and the one chosen would depend on the type of images and the desired objectives. The level to which segmentation is taken depends on the problem being solved. That is, segmentation should stop when the region of interest in the application has been isolated. Due to this property of

problem dependence, autonomous segmentation is one of the most difficult tasks in image analysis [17].

Object segmentation is a simple method that partitions an image into two segments: “object” and “background” [5]. There are a variety of these methods [9] [18], and each of them comes with its own set of features. Object segmentation can also be called object/background segmentation. Finding objects amid heterogeneous and cluttered backgrounds is a primary goal for many image analysts. In object segmentation, the size of the object is generally considerably less than that of the background. Object segmentation can also be used to detect anomalous objects (i.e., parts of the image that are different from the background) [3]. Object segmentation facilitates analysis of hyperspectral data.

2.3 Active Contours

There are a large number of segmentation methods in the literature. Among these, the technique of active contours has been widely analyzed and studied [19] [20]. This technique has become quite popular for a variety of applications, particularly in image segmentation [16]. This methodology is based upon the utilization of deformable contours which conform to various object shapes and motions.

We can think of an active contour as a snake that bends its body along the edge of the object. An important property of an active contour is the ability to make continuous edges where the edges are poorly defined. Even if the edge is weak or broken it will make a continuous contour [16]. The active contour technique is based on the idea of deforming on

initial curve to the boundary of objects under some constraints from the image using techniques of curve evolution.

The basic idea of the active contour is to start with initial boundary shapes represented in a form of closed curves, i.e. contours.

$$C(s) = \{(x(s), y(s)) : 0 \leq s \leq 1\} \quad \mathbf{2.1}$$

Then they are iteratively modified by applying shrink/expansion operations according to the constraints of the image. Those shrink/expansion operations, called contour evolution, can be performed by the minimization of an energy function [19] [8]. An energy function often used by traditional active contours is:

$$E = E_{\text{internal}} + E_{\text{external}}$$

The internal energy is designed to hold the curve together and to keep it from bending excessively [19] [8]. This allows control of the object shape, but requires careful adjustment of the weights for different kinds of object boundaries, or even at different stages of deformation [8]. The internal energy makes the curve become smooth, while the external energy leads the curve towards the edge of objects in the image [20]. The external energy term determines the criteria of contour evolution depending on the image [19].

2.4 Graph Cuts

New fast techniques have emerged for energy minimization techniques based on graph cuts [5]. These techniques can be applied to a restricted class of energy functions of discrete variables. An advantage of these methods is that in certain cases they can produce a global minimum of the energy or in other cases a local minimum, with some strong

properties. The basic terminologies that help us to better understand our approach pertaining to graphs are (concepts taken from [21]):

- **Graph, Network:** An abstraction of relationships among objects. Graphs consist exclusively of nodes and edges.
- **Weighted graphs and networks:** A weighted graph associates a label (weight) with every edge in the graph. Weights are usually real numbers.
- **Node, Vertex:** Objects ("things") represented in a graph. These are almost always rendered as round dots.
- **Edge, Link:** Relationships represented in a graph. These are always rendered as straight or curved lines.
- **Adjacent:** Two edges are adjacent if they have a node in common; two nodes are adjacent if they have an edge in common.
- **Route:** A sequence of edges and nodes from one node to another. Any given edge or node might be used more than once.
- **Path:** A route that does not pass any edge more than once. If the path does not pass any node more than once, it is a simple path.
- **Connected:** If some route exists from every node to every other, the graph is connected.
- **Loop, cycle:** A path which ends at the node where it began.

Let $G(V, E)$ be an edge weighted graph where V is defined as set of vertices v of the graph G , and E is defined as a set of edges (u, v) that connect nodes u and v . Each edge $(u, v) \in E$ in the graph is assigned a nonnegative weight (cost) represented by the term W_e

A graph G is called an undirected graph, when its edges have no orientation, they are not ordered pairs. A **directed** graph G is a graph where the set of edges are **ordered** pairs of vertices called directed edges [5]. In these types of graphs, an edge (u,v) is considered a directed from u to v , where u is called the head and v is called the tail of the edge [21].

A directed graph will often contain two additional special nodes (terminals), usually called the source $s \in V$ and the sink $t \in V$. The number of edges entering the source must be zero, and the number of edges leaving the sink must be zero.

A cut is a subset of edges $C \subset E$ such that the terminals s and t become separated. It is normal in combinatorial optimization to define the cost of a cut as the sum of the costs of the edges that it separates [5].

$$|C| = \sum_{e \in C} w_e$$

The minimum s - t cut problem is to find a cut C that separates s and t with the smallest cost [5] [8].

A flow network is a directed graph $G(V,E)$, with a source s and sink t , and a function $f: V \times V \rightarrow R$ that satisfies the following properties [8]:

1. **Capacity constraints:** The flow along an edge cannot exceed its capacity: For all $u, v \in V, f(u, v) \leq c(u, v)$.
2. **Skew symmetry:** The net flow from u to v must be the opposite of the net flow from v to u : For all $u, v \in V, f(u, v) = -f(v, u)$.

3. **Flow conservation:** unless $u=s$ or $v=t$. The net flow to a node is zero: For all $u \in V - \{s, t\}$, $\sum_{v \in V} f(u, v) = 0$.

The idea is that $f(u, v)$ describes a net flow along the edge (u, v) . In a flow network $G(V, E)$ the weight of each edge describes the capacity of that edge. Hence, each edge $(u, v) \in E$ has a nonnegative capacity $c(u, v) \geq 0$.

The maximum-flow problem is to find a flow (f) that maximizes the net flow in the directed graph [8] [21].

There is an important correspondence between flows and cuts in networks. The minimum s-t cut problem is equivalent to computing the maximum flow from the source to sink [8] [5].

The segmentation problem can be cast into a graph partitioning problem, where the pixels in the image correspond to nodes in the graph, the edges in the graph connect each pixel with its nearest neighbors and, associated with the edges, there is a weight function that measures the degree of similarity between two neighboring pixels. In this setting, the segmentation problem can be expressed as the optimal cut of the graph into a number of disjoint subsets of pixels that maximize the similarity (homogeneity, which is expected to be large if the gray levels of each pixel pair are similar) within each segment and the dissimilarity across segments [15].

Graph cut has several advantages. It allows a clean specification of the problem to be solved, as distinct from the algorithm used to solve it. In addition, energy minimization naturally allows the use of soft constraints, such as spatial coherence [4].

2.5 Related Work

We found various works related to image segmentation. Most of them work with the images represented as RGB values.

The author in [18] presented a framework for object segmentation in vector valued images. The first schema proposed is based on geometric active contours moving towards the objects to be detecting in the vector valued image. This technique is applicable to color and texture images. The second image processing algorithm used is image anisotropic diffusion and shock filters. This algorithm is a better approach for segmentation of images. The author shows that when boundary is roughly defined as a curve or surface separating homogeneous regions object segmentation can be associated with the problem of boundary detection and integration. In particular, the author shows that different approaches can be adopted to work on vector-valued images. The first approach is to process each plane separately, with the geodesic active contours and then integrate the results of this operation to obtain unique segmentation for the whole image. The second approach is to integrate the vector information from the very beginning, and deform a unique curve based on this information, directly obtaining unique object segmentation [18].

A graph cuts based active contours (GCBAC) approach to object segmentation is presented in [8]. First, they transformed a multi-source, multi-sink minimum cut problem into a single s-t minimum cut problem. In this context the problem of finding desired segmentation contour can be formulated as that of finding the closest contour that is a global minimum within its contour neighborhood (CN), given an initial contour. The authors of [8]

combine the active contours approach and the graph cuts approach. Their method differs fundamentally from traditional active contours in that they used graph cuts to deform the contour iteratively. The advantages of their approach are:

- (a) The algorithm has the ability to avoid the local minima and provide a more global result.
- (b) Graph cuts guarantee continuity, leading to smooth contours free of self-crossing and uneven spacing.

Another important technique was presented in [4], where the authors proposed graph cut methods by imposing connectivity constraints in the segmentation. The authors presented a new algorithm called DijkstraGC for computing segmentation of the image. The authors comment that graph cut is a popular technique for interactive image segmentation, but it has certain shortcomings. They further state that graph cuts have problems with segmenting thin elongated objects due to the “shrinking bias”. To overcome this problem, they proposed imposing an additional connectivity priority; this is a very natural assumption for real world objects.

The authors in [22] examined and tested a simple metric for classifying a pixel as either background or object material. They proposed and examined an object search that is well suited for hyper or multi spectral images. That search adapts the Mahalanobis distance metric, or Whitened Euclidean Distance (WED), and compares it to the Adaptive Cosine Estimator (ACE), and matched filter (MF). The matched filter (MF) provides a decision surface and helps extract objects from background. The WED (Whitened Euclidean Distance) is an enclosed surface surrounding the object. The Adaptive Cosine Estimator (ACE) is

another object identification algorithm. ACE confines object detection to a narrow cone. These of three methods were applied to independent data sets. ACE may detect objects with fewer false alarms than WED when, for example, the relative illumination of the object with respect to background during the training session varies from the actual detection collection. Only the object signature is needed to implement this algorithm and therefore no additional parameter optimization, testing, knowledge of the object's spectral signature or shape is required for WED and ACE. They remove the portion of the matched filter (MF) contributed by anomalous pixels, and only accept object pixels whose spectral vectors are oriented in the appropriate direction.

3 METHODOLOGY

In the previous chapter, we explained the principal concepts that help us to understand and develop the proposed application. This chapter presents the methodology used in our approach for object segmentation in hyperspectral images, and we show our approach and its variations in a detailed, step-by-step manner.

3.1 Proposed Algorithm Description

Hyperspectral images offer the possibility of characterizing materials and objects in the air, land and water on the basis of the unique reflectance patterns that result from the interaction of solar energy with the molecular structure of the material [23]. The proposed application segments all the desired objects in the hyperspectral image.

The proposed algorithm segments all the objects in an image and displays the contours of objects based on certain image characteristics or properties. The method uses both spectral and spatial information from images. Figure 3-1 shows the general block diagram for this approach.

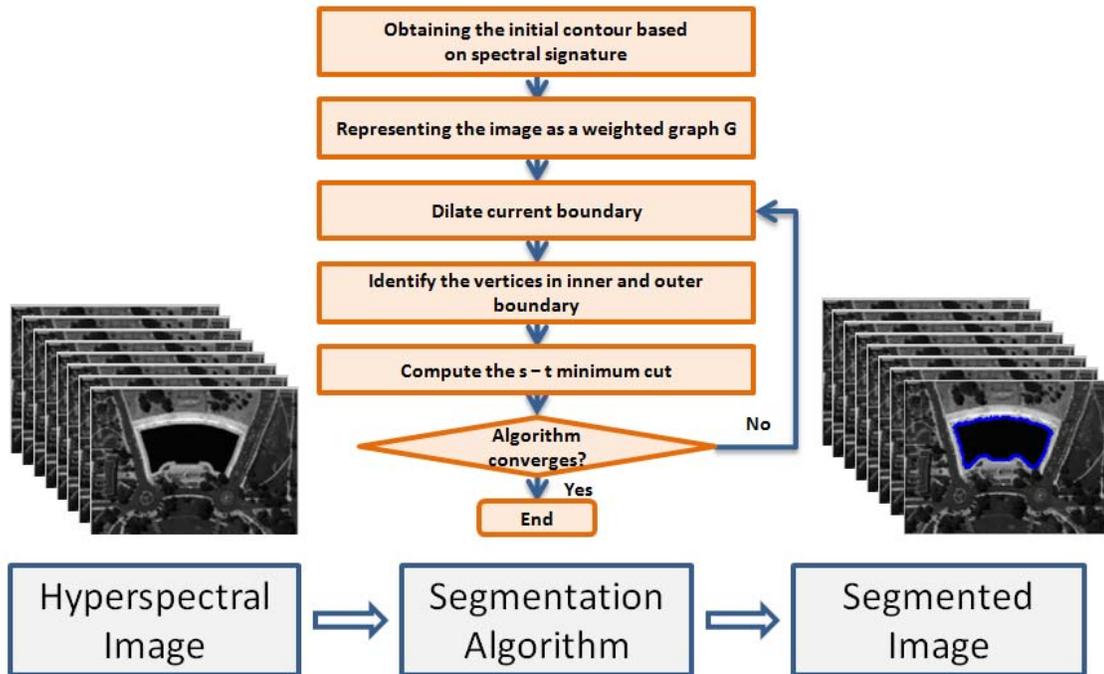


Figure 3-1: Block diagram for object segmentation in hyperspectral Images.

The work of this thesis is to modify the existing code presented in [8] to work with hyperspectral images and improve the results of the object segmentation. The description and modification that has been done is described below. Basically, steps 1 and 2 are different from the corresponding steps in [8]. Step 4 is different when the image is either small or has holes. Steps 3, 5 and 6 are quite similar. We begin with a brief and essential description. Details can be found in [8].

1. Obtaining the initial contour based on spectral signature.

Most of the methods based on active contour, require an initial contour. The basic idea is to start with initial boundary shapes represented in the form of closed curves (i.e. contours), and then to iteratively modify them by applying shrink/expansion operations according to the

constraints of the image [19]. It is important to give a good position for the initial contour. Most of the methods are sensitive to initial contour [20], and the segmentation results depend on it.

Classic active contour methods cannot detect more than one object, because they cannot split into multiple boundaries or merge multiple initial contours. Our approach avoids this restriction, allowing us to segment more than one object and place multiple initial contours.

Despite recent advances in hyperspectral image processing, automated object segmentation from hyperspectral images on non-homogeneous backgrounds is still an unsolved problem. With our approach, the initial contour is obtained in a supervised manner based on the spectral signature of the object. The spectra are obtained by selecting some points of the object.

Given a finite set of points selected by user and represented by $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$. Then vectors $P_x = \{x_1, x_2, \dots, x_n\}$ and $P_y = \{y_1, y_2, \dots, y_n\}$ represent the indices of those points in the X and Y direction, respectively. The set of points represented by *objPxs* correspond to real pixels from the hyperspectral images that represent the desired object:

$$objPxs = Img(Px, Py, :)$$

where *Img* represent the original hyperspectral image. Given a *pixel_{ij}* that represents a specific pixel of the image in the position on the *i* – row, and *j* – column respectively. The spectral angle distance between the desired object and *pixel_{ij}* is given by:

$$SAD(pixel_{ij}, objPxs) = \cos^{-1}\left(\frac{(pixel_{ij} * mean(objPxs))^T}{norm(mean(objPxs)) * norm(pixel_{ij})}\right)$$

Then, we calculate the spectral angle distances for all $pixel_{ij}$ in the hyperspectral image with the pixels of the desired object. With the results of this step we have a new image in two dimensions, in which the desired object is distinguished better in gray scale representation (Figure 3-2 (c)). Figure 3-2 (b) represents the normalized hyperspectral image.

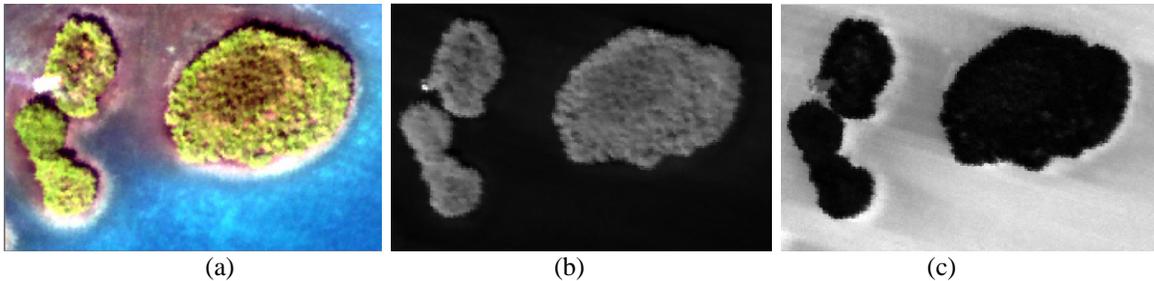


Figure 3-2: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color image from ENVI, (b) Gray scale of the hyperspectral image, and (c) Image formed by SAD.

The initial boundary is obtained by applying the angle metric, plus a threshold value. All the angles less than the minimum angle plus the threshold are chosen as a desired object and are given the value of one, and the value for the others is zero (Figure 3-3(a)). Finally, we extract the contour of this binary image (Figure 3-3(b)).

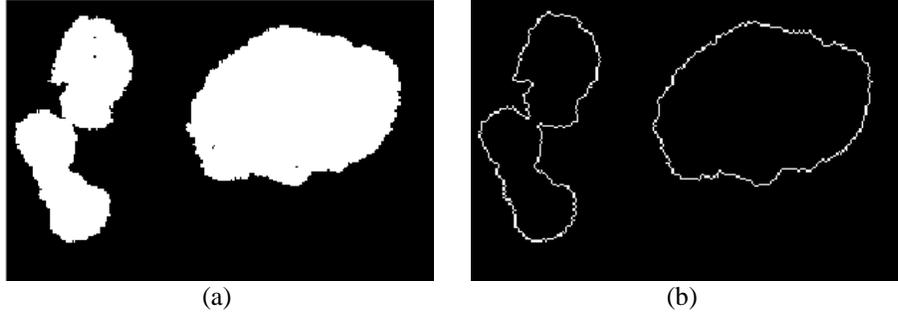


Figure 3-3: (a) Binary image obtained based on SAD with 15 thresholds value, and (b) Initial boundary for images 3.3 (a).

2. Representing the image as a weighted graph G .

The type of connectivity is an 8-connectivity graph, which means that each vertex in the graph corresponds to a pixel p and has edges connecting it to its 8 neighboring vertices, which correspond to the 8 neighboring pixels of p .

In this step, the hyperspectral image is represented as an edge weighted graph $G(V, E)$, where each pixel within the hyperspectral image is mapped to a vertex $v \in V$. If two pixels are adjacent, an edge exists $(u, v) \in E$ and has a nonnegative weight. If $(u, v) \notin E$ the weight is zero. The weights used in this approach are as follow:

$$W(i, j) = (g(i, j) + g(j, i))^2$$

where:

$$g(i, j) = \exp\left(-\frac{grad(i)}{\max(\max(grad))}\right)$$

Where $grad(i)$ is the norm of the gradient at location i , and $grad$ is the norm of the gradient of the normalized hyperspectral image plus norm of the gradients of the all SAD images obtained in the previous step.

$$grad = norm(grad_{gray}) + \sum_{Obj=0}^{NroObjects} norm(grad_{obj})$$

In this work, the internal energy is not used. The graph cuts guarantee the continuity of the resulting contour. The external energy is derived from the image and represented by edge weights on the corresponding graph.

3. Dilate current boundary.

The next step is to dilate the current boundary into its contour neighborhood (CN) with an inner boundary and an outer boundary. The size of the CN can be a fixed variable or specified by the user. The dilation size is an important parameter, as in some cases the accuracy of segmentation result depends on it. Note that for the first iteration the current boundary is the initial contour obtained in the first step.

According to [8], the dilation process has several objectives. As the dilation process generates a CN of the current contour, it makes the algorithm capable of jumping over local minima within this CN. The authors also explained that the size of the CN can be selected based on the size of the object to be segmented and the amount of noise in the data. In Figure 3-4 we display different sizes of CN.

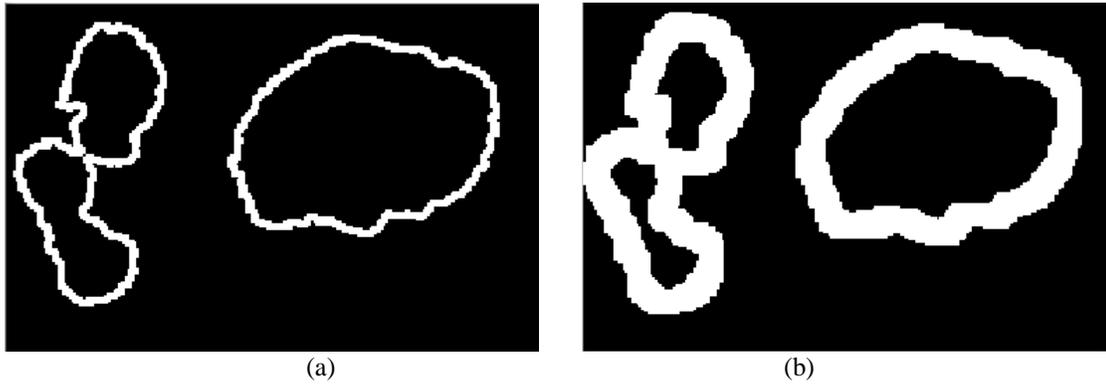


Figure 3-4: Dilation process. (a) Size of CN=3. (b) Size of CN=10.

4. Identify the vertices in inner and outer boundary.

Another objective of the dilation process is to generate an inner boundary that corresponds to the multiple sources and an outer boundary that corresponds to multiple sinks in the corresponding graph as shown in Figure 3-5. These multiple sources are identified as a single source, which is always contained in the S part of the resulting $s - t$ minimum cut.

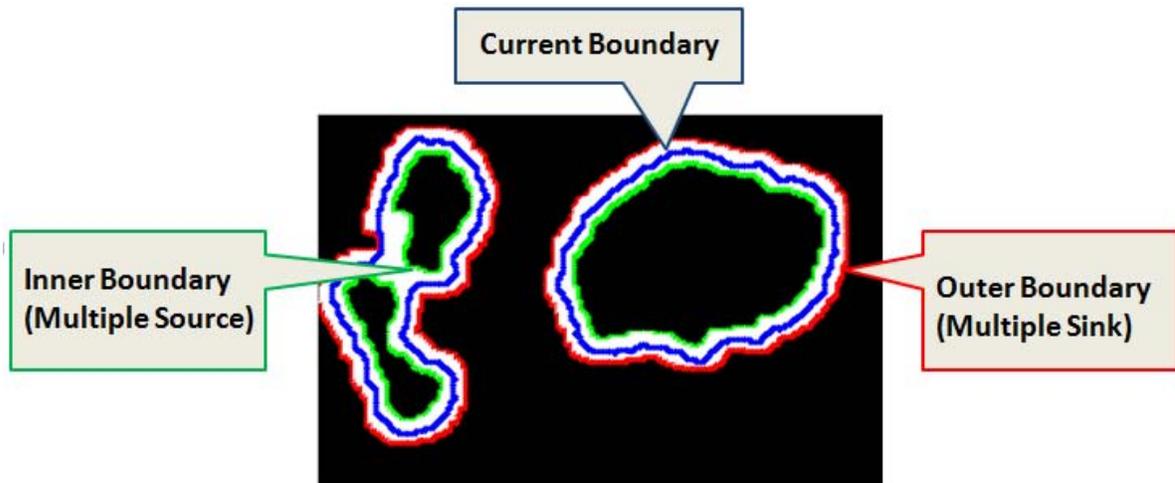


Figure 3-5: Vertices in inner, current, and outer boundaries.

5. Compute the $s - t$ minimum cut.

Treating the pixels on the inner boundary as multiple sources and the pixels on the outer boundary as multiple sinks as show in Figure 3-5, the goal of this step is to compute the best cut that would give an optimal segmentation (Figure 3-6). This is the problem of finding the global minimum contour within CN, formulated as a multi-source, multi-sink s-t minimum cut problem on the graph. The cost of a cut is defined as the sum of the costs of edges that it severs.

$$(S, T) = \sum_{u \in S, v \in T} c(u, v)$$

The s – t minimum cut problem is to find a cut in G that separates s and t as show in Figure 3-6, with the smallest weight [4] [5].



Figure 3-6: New Boundary Computed

There is an important correspondence between flows and cuts in networks, the max-flow min-cut theorem formulated in [24]. With this Theorem it is possible to solve the s-t minimum cut problem by using existing max-flow algorithms. The Theorem is as follows:

THEOREM 1 (Ford-Fulkerson Theorem): *The maximum flow from a vertex s to vertex t , $|f|$, is equal to the value of the weight $c(s, t)$ of the minimum cut separating s and t .*

An experimental comparison of several different max-flow min-cut algorithms can be seen in [25]. A simple operation on a graph of interest G in this regard is *node identification* which identifies a set of nodes $\{v_1, v_2, \dots, v_n\}$ as a single new node v , deleting self loops, if any, and merging parallel edges with cumulative capacity, as shown in Figure 3-7. In terms of this operation, there is a Theorem for the multi-source multi-sink $s-t$ minimum cut problem:

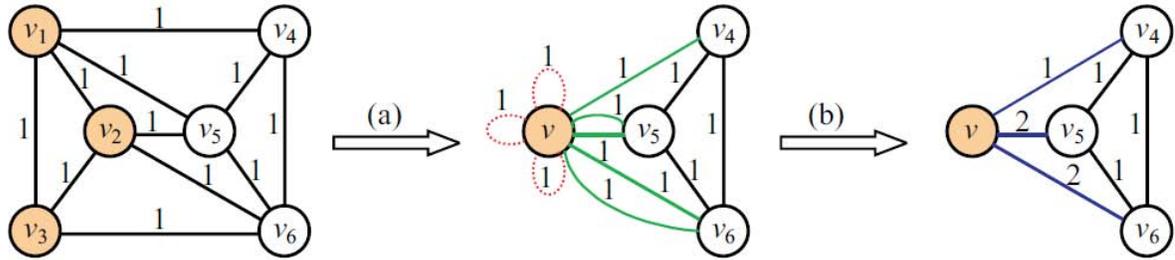


Figure 3-7: Node identification. (a) v_1, v_2, v_3 are merged into a new node v . (b) Self loops are deleted and parallel edges are replaced by a single edge. (Taken from [8]).

THEOREM 2 (Multi-source Multi-sink min-cut): *The minimum cut of graph G which separates a source set $\{s_1, s_2, \dots, s_n\}$ and a sink set $\{t_1, t_2, \dots, t_m\}$ is exactly the $s - t$ minimum cut of the graph that results after identifying s_1, s_2, \dots, s_n as a new source s and identifying t_1, t_2, \dots, t_m as a new sink t .*

With the help of this theorem, it is possible to use $s - t$ minimum cut algorithms to solve the multi-source, multi-sink minimum cut problem by simply identifying the multiple sources

as a single source and multiple sinks as a single sink, respectively. This theorem is demonstrated in [8].

6. Return to step 2 until the algorithm converges.

THEOREM 3 (Convergence Theorem): *Within a finite data set, the graph cuts based active contour will either converge or oscillate between several results with the same weight after finite number of iterations.*

This algorithm iteratively replaces a contour with a global minimum within the CN of the contour until the objective is achieved. This approach is guaranteed to converge by the above theorem that was demonstrated in [6]. There are two criteria for convergence after finite number of iteration that are:

- If the cost of the last iteration is equal to or less than current cost.
- If the contour of the last iteration is the same as the current contour.

3.2 Run time analysis

The algorithm from [8] implemented the excess scaling preflow-push algorithm to solve the $s - t$ minimum cut problem. The algorithm achieves the run time of $O(nm + n^2 \log U)$, where n is the number of nodes, m is the number of edges, and U is the largest edge weight. As we are using the same algorithm the running time to obtain the segmentation in our approach is the same.

The additional time required to obtain the initial contour is the number of pixels in the hyperspectral image $O(\text{nroRows} * \text{nroColumns})$, where nroRows is the number of rows, and nroColumns is the number of columns in the hyperspectral image.

3.3 Justification of the change made

For the first step, we have chosen to use spectral information to obtain the initial contour because the spectral information can be very helpful in the object segmentation in hyperspectral images if it is known that a given object has distinctive spectral information. This distinctive spectral information or spectral signature of the object can be used to discriminate between other objects in the image since other objects will have different spectral signatures. Spectral signature can be seen as analogous to the fingerprint of a person; it can be used to differentiate between different objects in the image [10].

For the second step, we decided to work with the images based on the SAD computation by the spectral signature of the desired object and the normalized hyperspectral image. This is because by joining both images, we were able to obtain a good result with a short computation time. The high dimensionality of hyperspectral images demands high storage capacity and high computational efforts from segmentation algorithms [10]. Methods for reducing the image dimensionality are often applied. This reduction must be done without losing relevant information about objects of interest [1]. Of course, this process takes extra time. In next section we will compare the results with original images and preprocessing images.

4 TESTING AND EVALUATION

The previous chapter explained our approach step by step. This chapter presents results obtained by using different hyperspectral images using different input parameters and doing simple variations in some steps of the application.

4.1 Data Set Description

Data sets of hyperspectral images are typically very large because of the high dimensionality of the hyperpixels, and consequently, computational efficiency is a major factor in most hyperspectral image processing methods. Our approach has used a sub-region of the original image. Algorithm testing is done with hyperspectral images from the sensors given below:

Hyperspectral Data Imagery Collection Experiment (HYDICE), captures the information in 210 contiguous bandwidths from the visible to shortwave infrared (400-2500 nm) with a spatial resolution that varies between 1 to 4 meters depending on the aircraft's altitude above ground level.

The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) is a proven instrument in the realm of Earth Remote Sensing. It is a unique optical sensor that delivers calibrated images of the upwelling spectral radiance in 224 contiguous spectral channels (bands) with wavelengths from 400 to 2500 nanometers [6].

The SOC-700 hyperspectral camera has a spectral resolution of 4 nm with 120 bands and a spectral range from 400 to 900 nm.

Finally, hyperspectral images taken by AISA Eagle Sensor are also used. These images were taken at different spatial resolutions such as 1m and 4m, and they have 128 spectral bands.

4.2 Experimental Results

In this section, we present our experimental results with real and synthetic hyperspectral images. Different input parameters are used to compare the segmentation results. Points that represent the pixels of the desired object are chosen, values for the threshold to obtain the initial contour are set, and the size of dilation process is set. The algorithm is implemented in Matlab R2009b, and calls are made to some libraries implemented in Visual Studio 2008 C++. The resulting algorithm is robust and accurate. Choosing good input parameters is both simple and intuitive.

Our application provides a range for the threshold value based on the values of both the minimum angle and the maximum angle. The minimum value for the threshold is 0, and the maximum value for the threshold is the difference between the maximum angle and the minimum angle. If the value of the threshold is very small, the initial contour may have no objects as in Figure 4-1(b).

On the other hand, if the value of the threshold is much larger the initial contour will be the entire image Figure 4-1(c). In our experiments, we note that for images like those obtained by AISA sensor, the application works well with a value for the threshold equal to 10 as in Figure 4-1(d). Figure 4-1(a) shows the selected input points in the application.

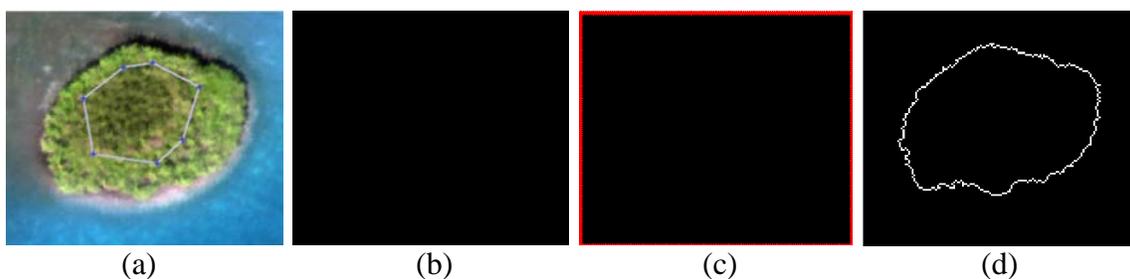


Figure 4-1: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) 7 points of the object of interest (bands 54 35 15 RGB), (b) Initial contour with threshold=0, (c) Initial contour with threshold=66, (d) Initial contour with threshold=10.

In some images, the size of the dilation process does not significantly change the results. Figure 4-2(a) and Figure 4-2(b) show the result using dilation size CN=3 and CN=9 respectively. Figure 4-2(b) shows the comparison between the two segmentation results, while the number of pixels found as an object for Figure 4-2(a) is 523, and the number of common pixels found as an object for both Figure 4-2(a) and Figure 4-2(b) is 517, representing a 98.85 % similarity between both results in Figure 4-2(a).

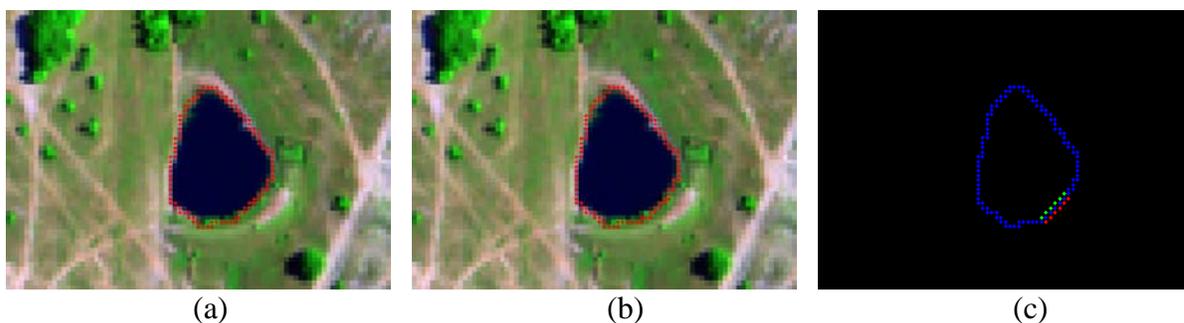


Figure 4-2: Region selected from HYDICE Terrain sample scene (bands 168 93 85 RGB). Segmentation generated using 5 points, threshold=9 for both images, and (a) CN=3; (b) CN=7 and c) Merging segmentation result for (a) and (b).

The input points selected for the application do not change the segmentation result in some images. Figure 4-3 shows different numbers of input points, which the segmentation

results are very similar. The Figure 4-3 ((a), (b) and (c)) shows the number of points selected by user and Figure 4-3 ((d), (e) and (f)) show the segmentation result for each of them respectively.

The number of pixels segmented as an object for Figure 4-3 ((d), (e) and (f)) are 977, 983, and 983 respectively. The total pixels common for three results are 977, which implies 100% similarity in comparison with segmentation result for the first image (Figure 4-3(d)), and 99.3896% similarity in comparison with the segmentation results for the second and third images (Figure 4-3(e) and Figure 4-3(f)).

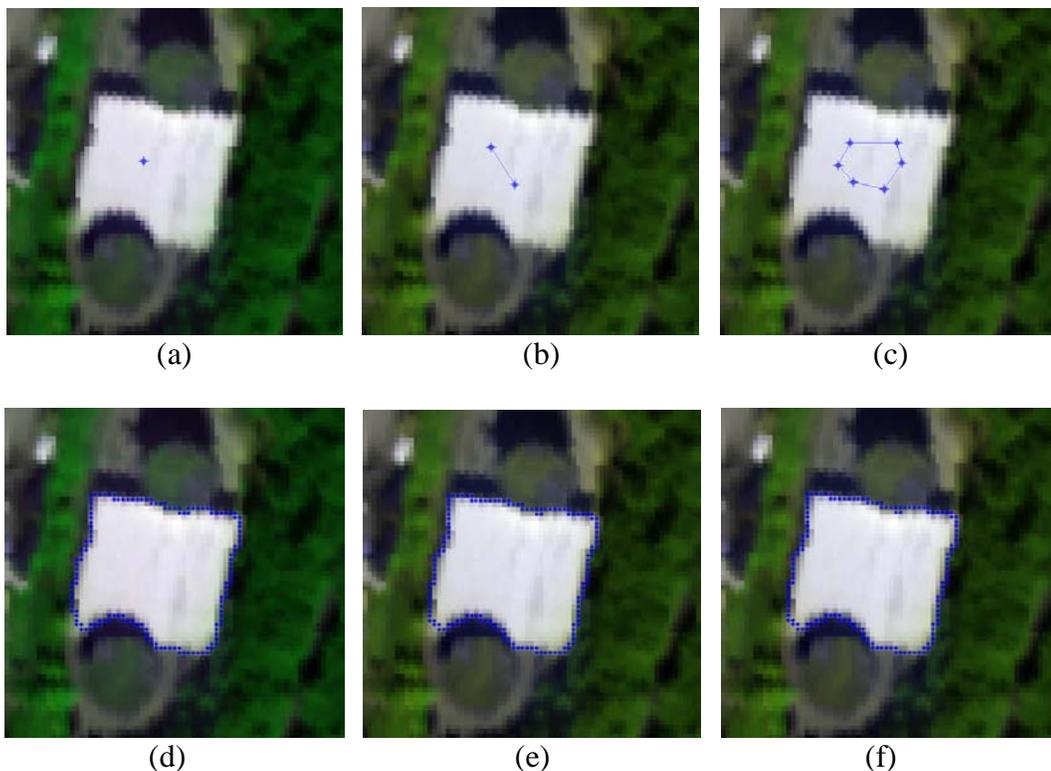


Figure 4-3: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 67 70 16 RGB) using threshold=10 and CN=3. (a), (b) and (c) Entry points; (d), (e) and (f) Segmentation results for (a), (b) and (c) respectively.

4.2.1 Results of segmenting images with holes

Figure 4-4 shows that our application works well with images that have holes while the method presented in [8] does not segment objects with holes. To work with images that have holes, our method finds the initial boundaries of the holes and then interchanges the object as a background and background as object.

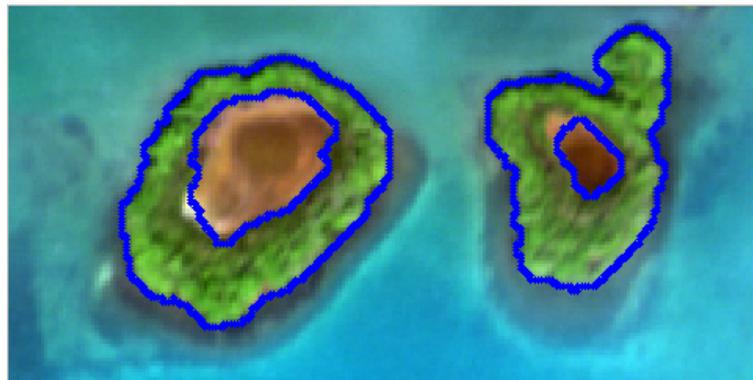


Figure 4-4: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 5 points of the object of interest, threshold=10 and CN=5.

4.2.2 Results of segmenting more than one object

As mentioned before, our approach can segment more than one object. This represents an important advantage for our approach. At least one point has to be selected for each object with distinct spectra. Then the application segments all the objects present in the images. Figure 4-5 and Figure 4-6 show the segmentation results for two different objects.



Figure 4-5: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 40 35 15 RGB) using 10 points for the first object(blue) and 5 point for the second object(red), for both threshold=10 and CN=3.

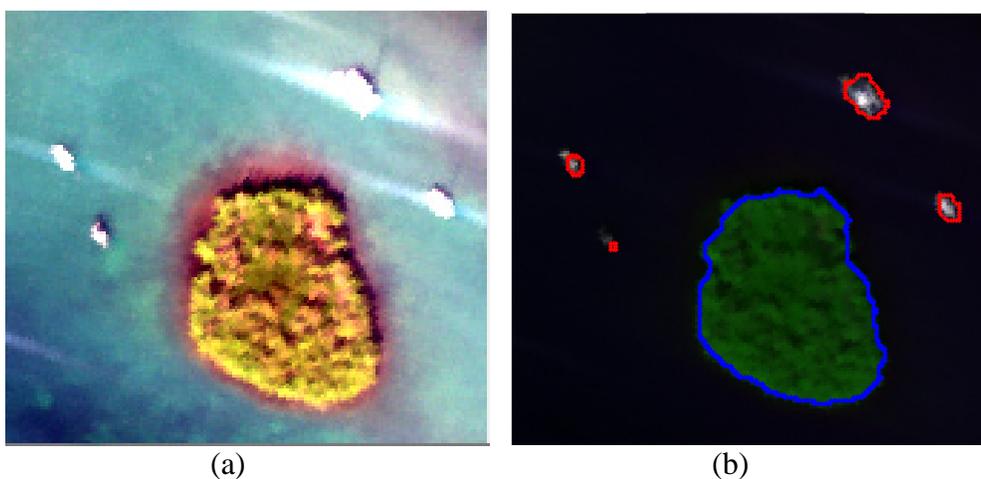


Figure 4-6: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color images from ENVI, and (b) Segmentation generated (bands 55 70 16 RGB) using Threshold=10; (a) 5 points and CN=5 for the first object (blue); and 4 point and CN=2 for the second object (red).

4.2.3 Results of segmenting small objects

The background is interchanged with the object while segmenting small objects (Figure 4-6(b) Figure 4-7(a), and Figure 4-8). All objects that have less than 40 points in the

initial contour are considered small objects. Another issue with respect to small objects is the size of the dilation process size. In order to work for a small object we need to use the CN value between 2 or 3. Figure 4-7(b) shows the result of segmentation using CN=5 as we can see the segmentation result is not correct, because the segmentation result is located far from the object of interest.

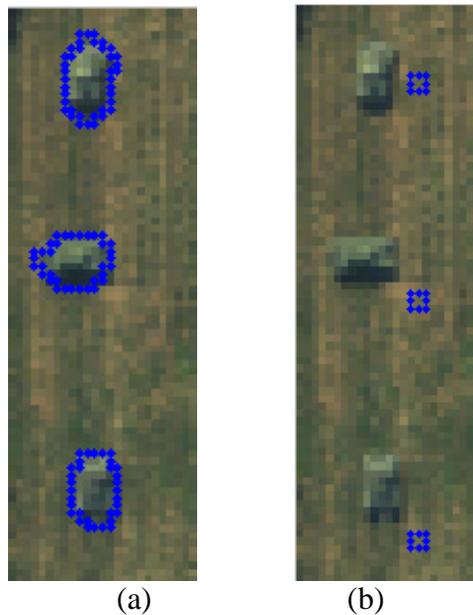


Figure 4-7: Region selected from HYDICE Forest. Segmentation generated (bands 49 37 18 RGB) using 6 points of the object of interest, threshold=10 and (a) CN=3. (b) CN=5.

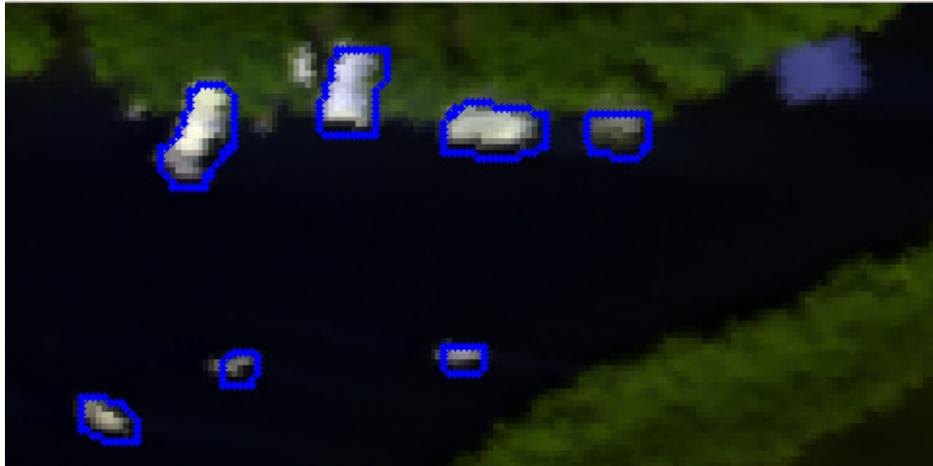


Figure 4-8: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 6 points of the object of interest, threshold=10 and CN=2.

4.2.4 Results of segmenting large images

The run time of this approach for Figure 4-9, the largest of the images, (number of rows is 400 and number of columns is 400, and the total nodes in the graph is 160000) is as follows: The CPU time used by the initialization process is 80.6143 seconds, the CPU time used to find the initial contour is 70.7146, and the CPU time of the segmentation algorithm is 0.3276 seconds.

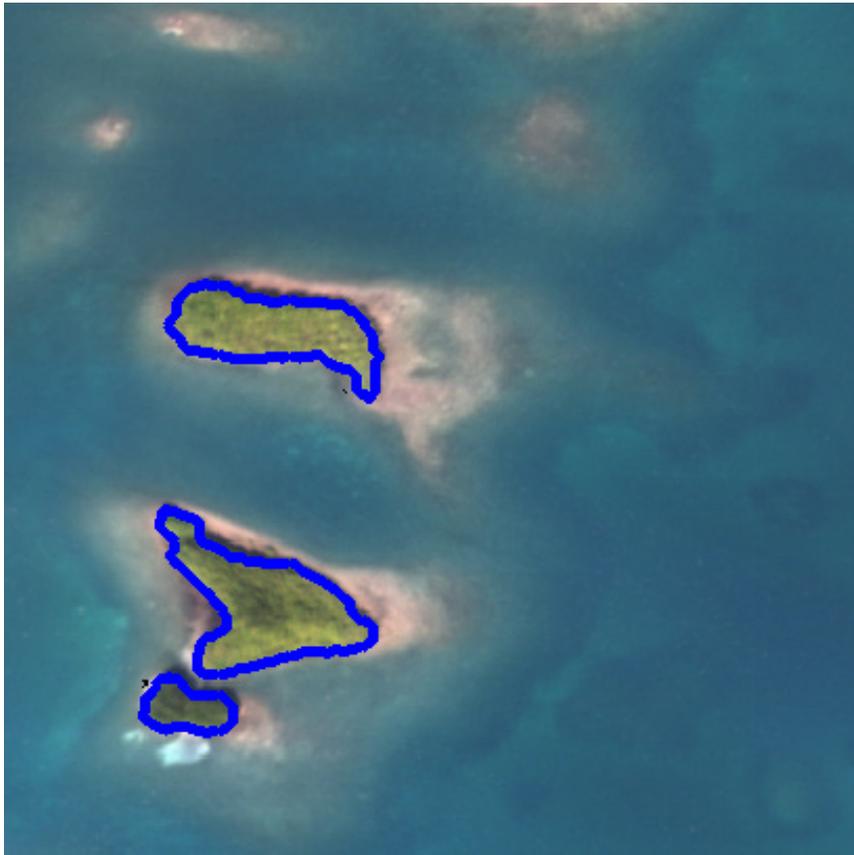


Figure 4-9: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 10 points of the object of interest, threshold=11 and CN=4.

For Figure 4-10 (number of rows is 320 and number of columns is 250, and the total nodes in the graph is 75000) the CPU time used by the initialization process is 11.6845 seconds, the CPU time used to find the initial contour is 21.9958, and the CPU time of the segmentation algorithm is 0.2028 seconds.



Figure 4-10: Regions selected from Washington D.C. Mall. Segmentation generated (bands 177 171 136 RGB) using 15 points of the object of interest, threshold=16 and CN=3.

For Figure 4-11 (number of rows is 167 and number of columns is 372, and the total nodes in the graph is 62124) the CPU time used by the initialization process is 3.1668 seconds, the CPU time used to find the initial contour is 2.9796 seconds, and the CPU time of the segmentation algorithm is 0.3900 seconds.

The results presented above show that our method is suitable for use in practice because the run time is not very large. Note that in our approach, the first step is the one that takes more time. This is because the method runs through all the pixels of the image to find the initial contour.

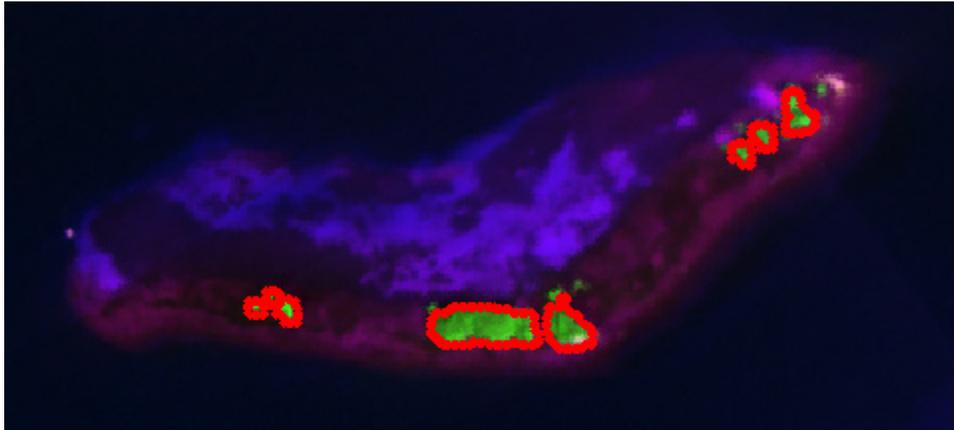


Figure 4-11: Regions selected from “Enrique Reef data” captured by AVIRIS. Segmentation generated (bands 50 100 16 RGB) using 5 points of the object of interest, threshold=10 and CN=2.

4.2.5 Results of segmenting images with noise

Another test to validate the performance of the algorithm was conducted. It consists of adding noise to the original image to see how sensitive the proposed algorithm is to noise. Mathematically, the process of adding noise can be expressed as: $imgNoise = Img + n * W$, where Img is the original image; n is a multiplication factor and W is a structure the same size as Img containing Gaussian noise. Figure 4-12 shows the result of the original images, with Gaussian noise for $n=10$ and $n=100$.

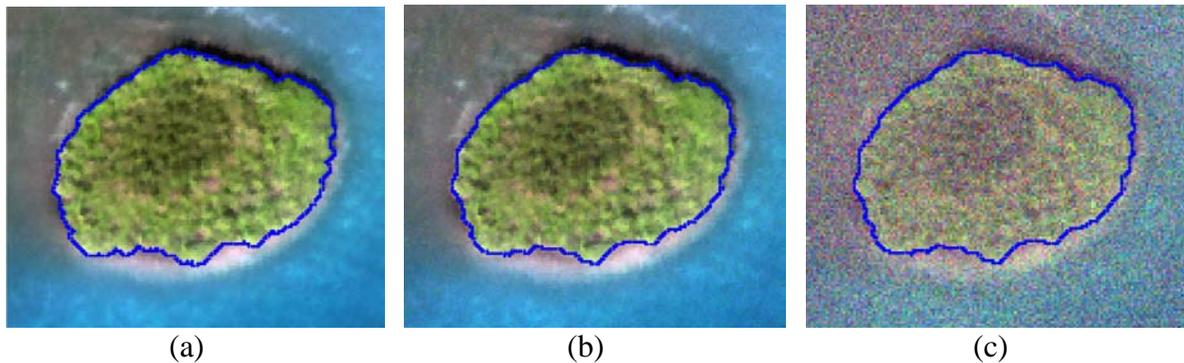


Figure 4-12: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 5 points of the object of interest, threshold=10 and CN=5. (a) Result of the original images, (b) Result adding noise with $n=10$, (c) Result adding noise with $n=100$.

The number of pixels that represent the object obtained by segmentation of the original image without noise is 7350; adding noise with $n = 10$ the total pixels belonging to the object after segmentation is 7344, which represents 99.82 % similarity in comparison with the segmentation result for the original image. The number of pixels belonging to the boundary found in the original image is 426 pixels, and the common pixels boundary for both images is 407, which implies 95.76 % similarity for the boundary detection result.

Adding noise with $n = 100$ the total pixels common to both (original and noisy image) segmentation results is 7327, which represents 99.69 % similarity in comparison with segmentation result of the original image. The common pixels in the boundary for both images are 397, which implies 95.76 % similarity.

Figure 4-13 shows the results combining the original result and the results with noise. The blue points are the common boundary point results for both the original image and the image with noise. The green points are the result of the segmentation of the original image, and the red points are the result of segmentation of the image with noise.

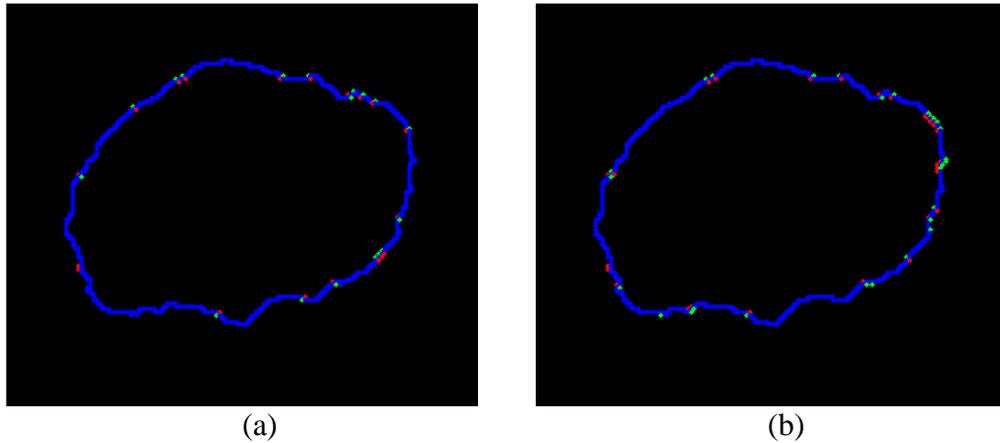


Figure 4-13: (a) Combination of segmentation results from Figure 4-12 (a, and b), (b) Combination of segmentation result from Figure 4-12 (a and c)

Figure 4-14 shows the result of the original images and with Gaussian noise $n=100$. Visually the segmentation result for both images is very close. The percentages presented above as well as Figure 4-13 and Figure 4-14 show no significant change in segmentation result, demonstrating that the proposed application can work very well with images that contain noise.

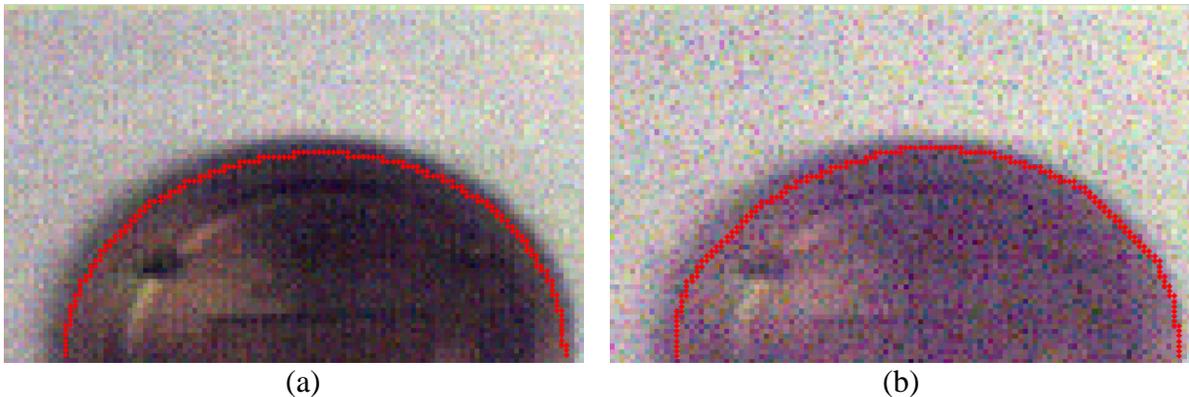


Figure 4-14: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 40 35 15 RGB) using 5 points, threshold=3 and CN=3 (a) Result of the original images, (b) Result adding Gaussian noise with $n=100$.

4.2.6 Results of segmentation after preprocessing the image

Many datasets of hyperspectral images are contaminated with noise because of the data acquisition process, naturally occurring phenomena, poor resolution of sensor, or other factors. In some cases it makes the segmentation results more difficult and causes inaccurate results. Several algorithms have been developed to reduce noise and improve the signal to noise ratio of images, and this task is called denoising. Many researchers have shown more acceptable and accurate results when the initial steps are both resolution enhancement and feature extraction [15] [1] [10].

Resolution enhancement and dimensionality reduction algorithms are used in the HIAT (The Hyperspectral Image Analysis Toolbox) [26] as pre-processing processes. There are several different methods available for dimensionality reduction. Here, the SVDSS (Singular Value Decomposition Band Subset Selection) has been used (It is available in the HIAT). Subset selection refers to selecting the most independent columns of a matrix, and an important advantage of this method is the retention of the physical meaning of the data; that is, there are no data transformations.

Figure 4-15(a) shows the segmentation result using original image, which the CPU time used by the initialization process is 0.2656 seconds, the CPU time used to find the initial contour is 0.2352 seconds, and the CPU time of the segmentation algorithm is 0.2040 seconds. Figure 4-15(b) shows the segmentation result using dimensionality reduction with 20 bands from original image. The CPU time used by the initialization process is 0.2328 seconds, the CPU time used to find the initial contour is 0.3484, and the CPU time of the segmentation algorithm is 0.1936 seconds. Figure 4-15(c) shows the segmentation result

using dimensionality reduction with 10 bands from the original image. The CPU time used by the initialization process is 0.2340 seconds, the CPU time used to find the initial contour is 0.6588, and the CPU time of the segmentation algorithm is 0.0936 seconds. In summary, the total times for segmentation results are 0.6348, 0.5348 and 0.9553 seconds respectively.

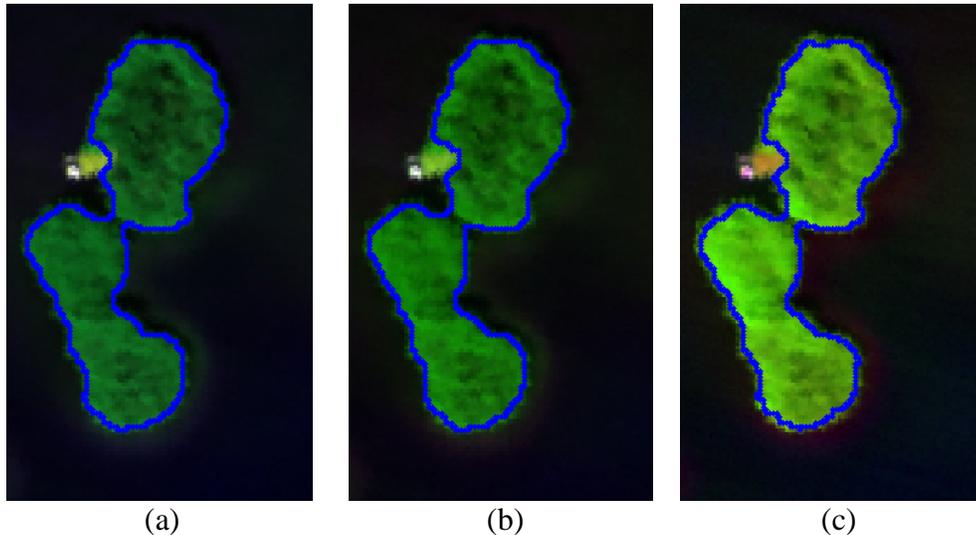


Figure 4-15: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated using 8 points, threshold=8 and CN=3. (a) Result for original images with 128, (b) Result for images with 20 bands, and (c) Result for images with 10 bands.

Figure 4-16(c) shows the result combining the segmentation results for the figures in Figure 4-15. The blue points are the common boundary's points for three images. The yellow points are the result of the segmentation for the original image. The green points are the result of the segmentation for the image with 20 bands. Finally, the red points are the result of the segmentation of the image with 10 bands.

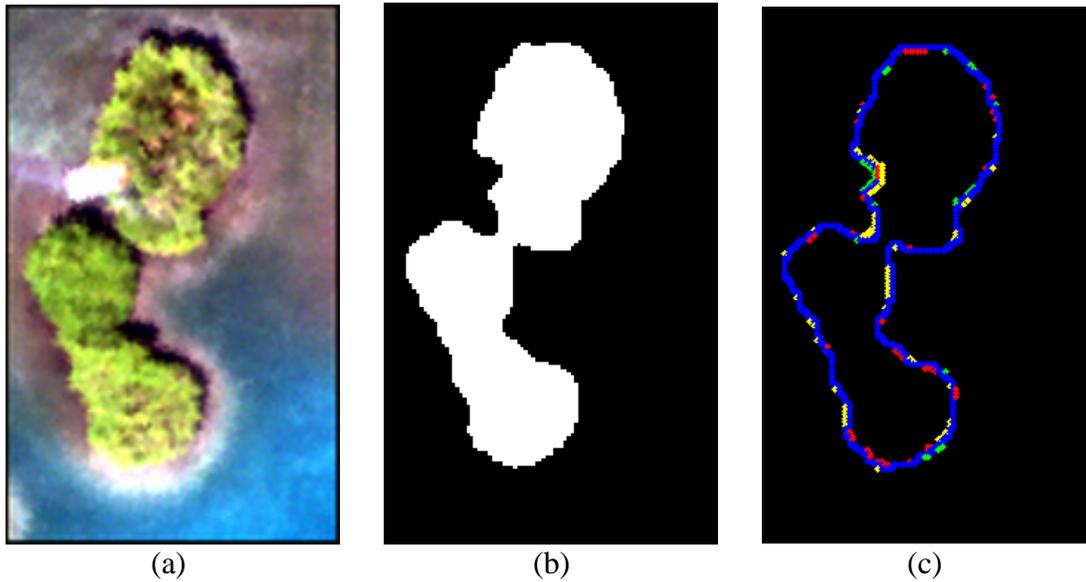


Figure 4-16: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. (a) True color from ENVI, (b) Common pixels belonging to segmentation result for images from Figure 4-15, and (c) Combination of segmentation result from Figure 4-15.

The total pixels for each segmentation result in Figure 4-15 are 3530, 3541 and 3509 respectively. The total pixels common for the three segmentation results (Figure 4-16(b)) are 3453, which represent 97.8187%, 97.5148% and 98.4041% similarity in comparison with the segmentation result of the original image, images with 20 bands and images with 10 bands respectively.

The total pixels for each segmentation result in Figure 4-17((a), (b), and (c)) are 1756, 1751 and 1774 respectively. The total pixels common for three segmentation results are 1751, which represent 99.7153%, 100% and 99.8290% similarity in comparison with the segmentation result of the original image, images with 10 bands and images with 5 bands respectively.

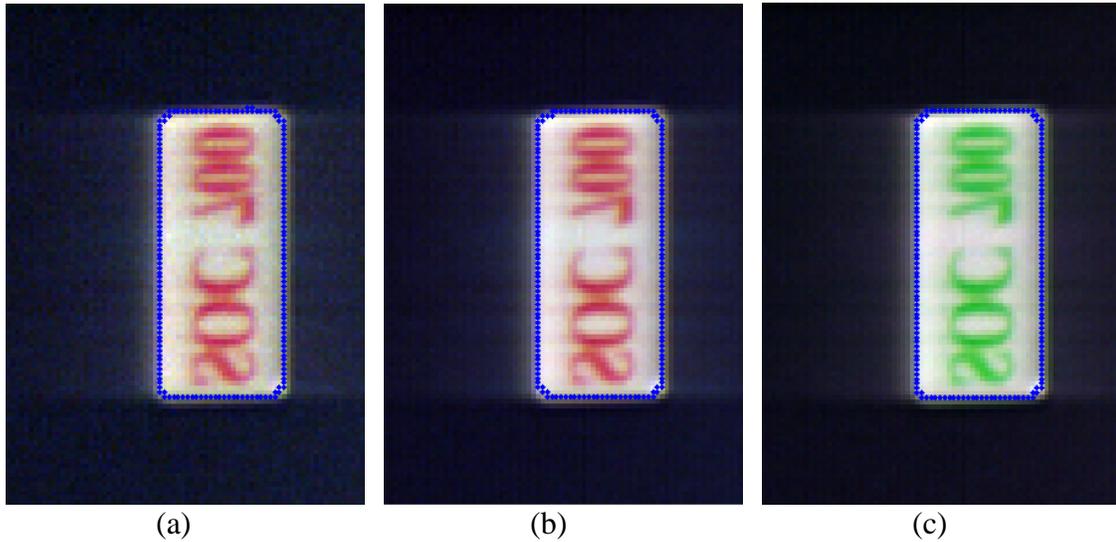


Figure 4-17: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated using 8 points, threshold=10 and CN=3 (a) Result for original images with 120, (b) Result for images with 10 bands, and (c) Result for images with 5 bands.

The total run times presented above show that our approach is quite fast, even without preprocessing images. The percentages presented above show no significant change in segmentation result, indicating that it is not necessary to spend time doing the additional step of dimensionality reduction and thus saving preprocessing time.

4.2.7 Comparison with original algorithm

To compare our approach with the original algorithm presented in [8], we represented the hyperspectral image in RGB scale. The Figure 4-18 (a) and Figure 4-18 (b) represent a RGB images using the band 40 for red color, 35 for green color and 15 for blue color. Figure 4-19(a) and Figure 4-19 (b) represent a RGB images using the band 181 for red color, 189 for green color and 179 for blue color.

Figure 4-18(a) and Figure 4-19(a) show that if an initial boundary is far from the object boundary, it is still difficult for the original algorithm to find the real object boundary

(Figure 4-18(d) and Figure 4-19(d)). It is necessary to give a reasonably accurate initial boundary (Figure 4-18(b) and Figure 4-19(b)) in order to obtain good results (Figure 4-18(e) and Figure 4-19(e)) in the original algorithm.

Our approach can segment a desired object by choosing at least one point (Figure 4-18(c) and Figure 4-19(c)). The CPU times of the segmentation algorithm are 1.180, 0.9672 and 0.3120 seconds for Figure 4-18((d), (e) and (f)) respectively, and The CPU times of the segmentation algorithm are 0.2496, 0.2184, 0.1952 seconds for Figure 4-19((d), (e) and (f)) respectively. We can appreciate when the initial contour is closer to the desired object and the time of the segmentation algorithm is less. Note that in calculating the total CPU time for our approach it is necessary to add the CPU time for our first step.

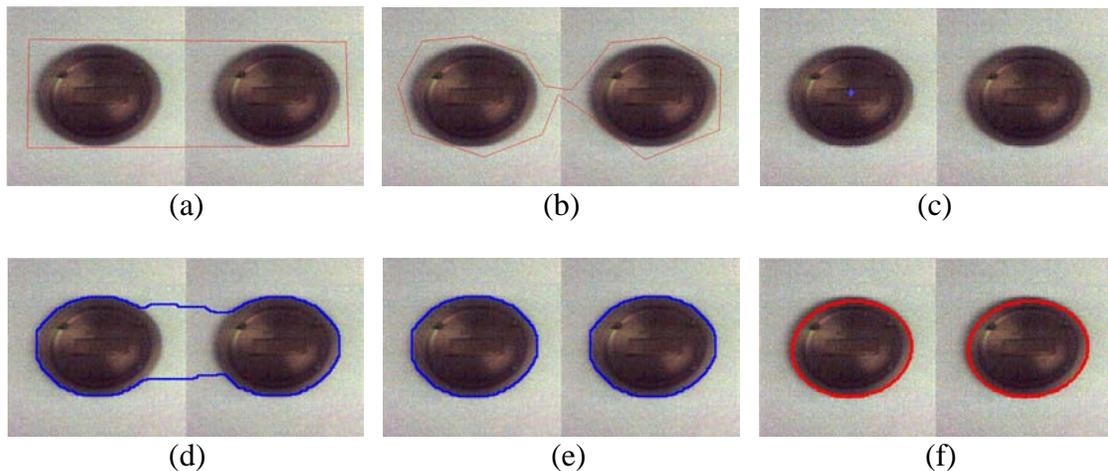


Figure 4-18: Region selected from Fake Leaves hyperspectral image captured by SOC-700 (bands 40 35 15 RGB). (a) and (b) Initial contour, (c) Input point; (d) and (e) Segmentation result using original algorithm for Figure 4-18 ((a) and (b)) respectively; (f) Segmentation result using our approach with threshold=10 and CN=2.

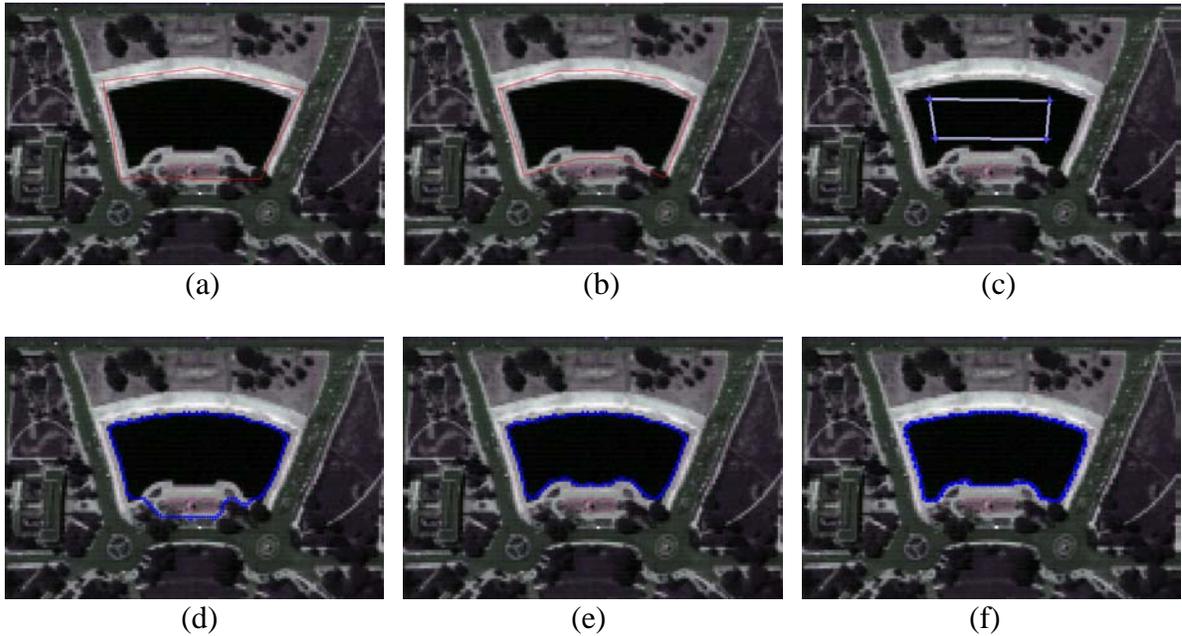


Figure 4-19: Regions selected from Washington D.C. Mall (bands 181 189 179 RGB). (a) and (b) Initial contour, (c) Input points; (d) and (e) Segmentation result using original algorithm for Figure 4-19 ((a) and (b)) respectively; (f) Segmentation result using our approach with threshold=16 and CN=5.

4.2.8 Comparisons with segmentation band by band

Some approaches use each band separately to work with hyperspectral images and then integrate the results to obtain a unique result for the whole image. These kinds of approaches take more processing time and sometimes the results are not good.

Figure 4-20((a), (b) and (c)) show the segmentation result band by band. The CPU times of the segmentation algorithm are 12.1057, 12.0277 and 12.7297 respectively. Visually the Figure 4-20(f) shows that the resulting contour does not close properly the region of water to be segmented. Figure 4-20(f) shows the segmentation result with our approach. The CPU time of the segmentation algorithm is 0.1240. With these results we show some disadvantages of working band by band.

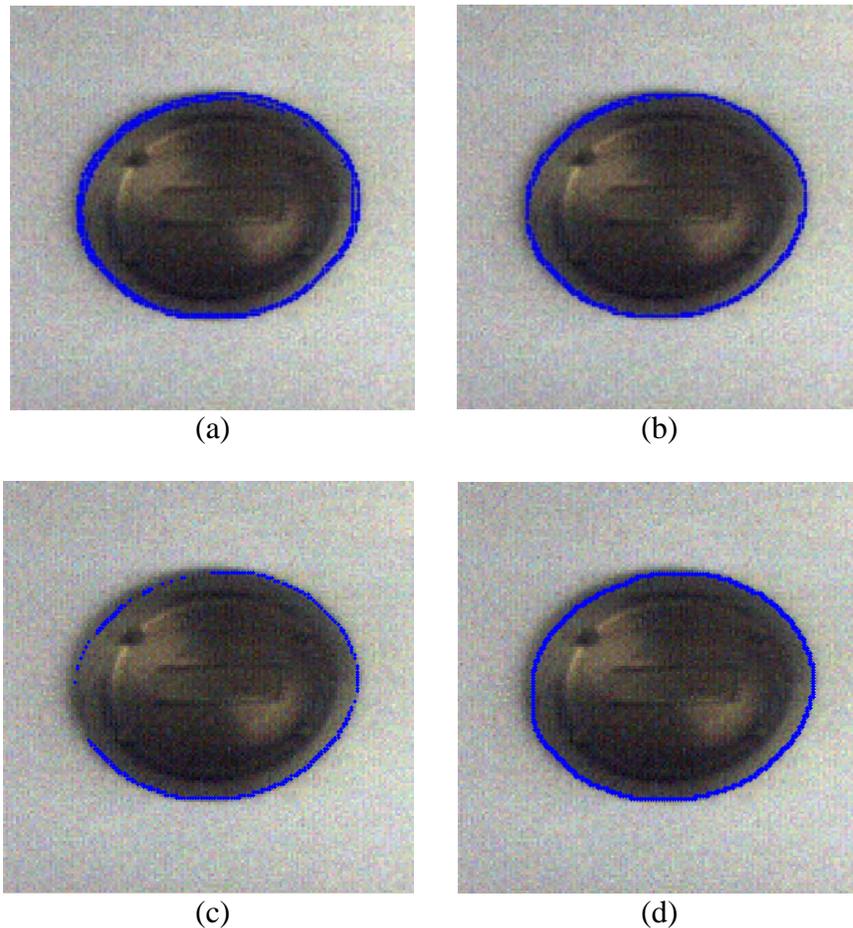


Figure 4-20: Region selected from Fake Leaves hyperspectral image captured by SOC-700 (bands 54 36 16 RGB) Segmentation generated using 8 points, threshold=11 and CN=4 for all images. (a) Result for more than 25% fitting in the all segmentation results. (b) Result for more than 35% fitting in the all segmentation results. (c) Result for more than 50% fitting in the all segmentation results. (d) Segmentation result for our approach.

4.2.9 Comparisons between segmentation and classification results

We do not have images with ground truth, and to see if our approach has acceptable results we compare our results using the process of classification. Image classification is the process of partitioning the pixels in a digital image into classes according to their characteristics [1]. Two main classification schemes are unsupervised and supervised

classification. Unsupervised classification can be defined as the identification of natural groups or structures within the data. This technique does not require the user to specify any information about the features contained in the images. Supervised classification can be defined as the process of using samples of known identity (training data) to classify pixels of unknown identity. The training data are used to train the classifier which is tested with testing samples to evaluate the accuracy of the classifier [1] [15].

We use supervised classification. The supervised angle detection classification and Mahalanobis distance classification have been used (These methods are available in the toolbox HIAT, that support this kind of classifications). The parameters for regularization used here were the default values that appear in the toolbox.



Figure 4-21: Regions selected from “PR Hyperspectral Science Areas” captured by AISA. Segmentation generated (bands 54 35 15 RGB) using 10 points of the object of interest, threshold=10 and CN=11.



Figure 4-22: Training classes for the classification.

Figure 4-21 shows the segmentation result of our approach. Figure 4-22 shows the training classes for supervised classifications. Figure 4-23 and Figure 4-24 show the classification results using angle detection and Mahalanobis distance respectively.

The number of pixels that represent the object obtained by segmentation for Figure 4-21 is 3107. The total pixels belonging to the object after the classification for Figure 4-23 is 3149. The total common pixels for both results are 3024, which implies 96.0305% similarity in comparison with the classification result, and 97.3286% similarity in comparison with the segmentation result.

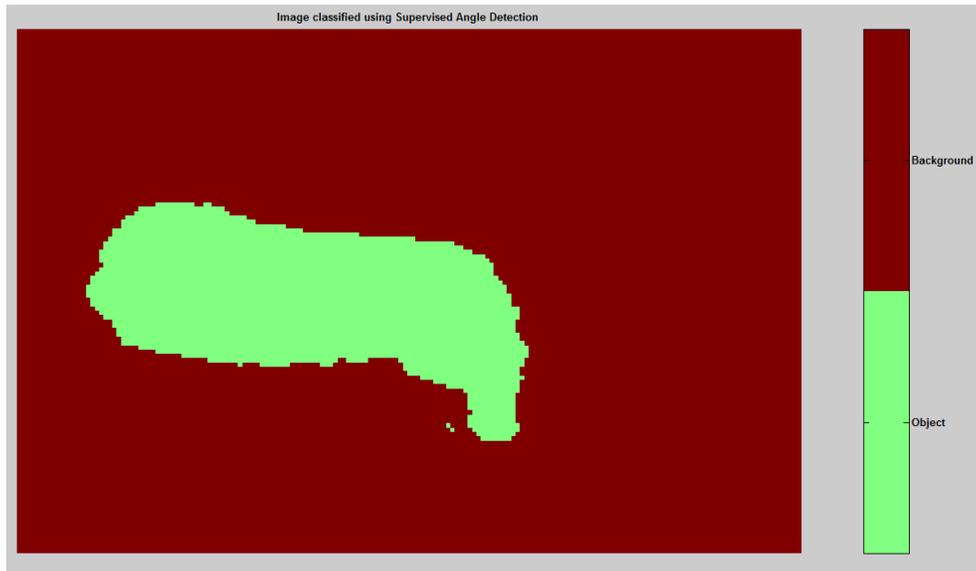


Figure 4-23: Result of the image classification using supervised angle detection for Figure 4-21.

The number of pixels that represent the object obtained by segmentation for Figure 4-21 is 3107. The total pixels belonging to the object after the classification for Figure 4-24 is 3077. The total common pixels for both results are 2999, which implies 97.4651% similarity in comparison with classification result, and 96.5240% similarity in comparison with segmentation result.

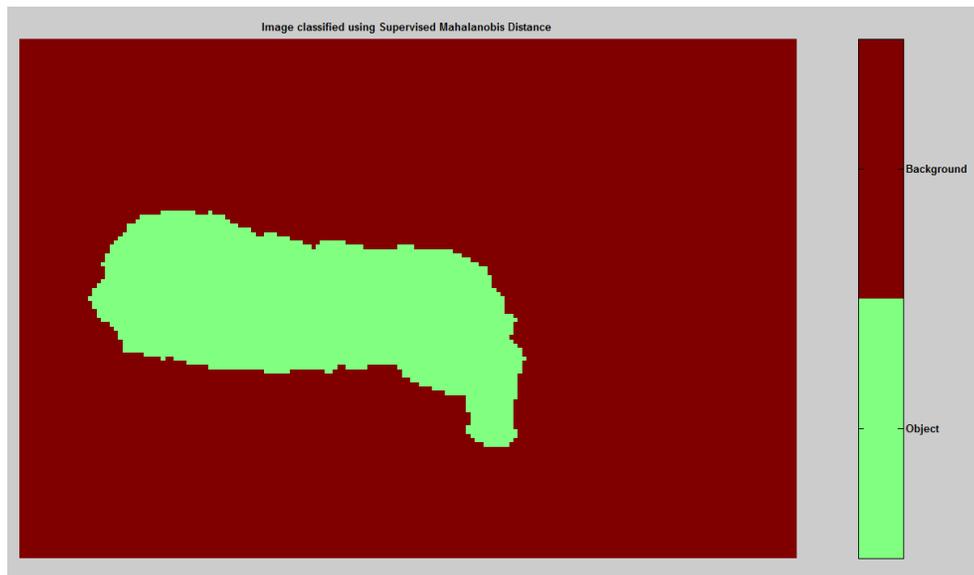


Figure 4-24: Result of the image classification using Mahalanobis distance for Figure 4-21.

The CPU time for classification using the angle detection is 126.2048 seconds. The CPU time for classification using Mahalanobis distance is 128.2172. For segmentation the CPU time used by the initialization process is 0.8580 seconds, the CPU time used to find the initial contour is 1.2792, and the CPU time of the segmentation algorithm is 0.2184 seconds.

Figure 4-25 shows the segmentation result of our approach. Figure 4-26 shows the training classes for supervised classifications. Figure 4-27 and Figure 4-28 show the classification results using angle detection and Mahalanobis distance respectively. As we can see, visually there is not much difference between classification results and the segmentation result.



Figure 4-25: Region selected from Fake Leaves hyperspectral image captured by SOC-700. Segmentation generated (bands 57 36 13 RGB) using 8 points for the first object(round) and 7 point for the second object(rectangular), for both threshold=10 and CN=3.

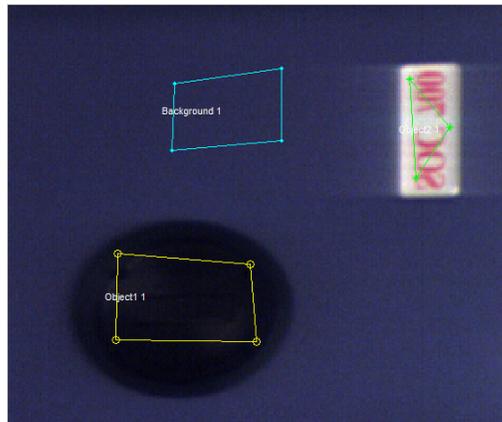


Figure 4-26: Training classes for the classification.

The CPU time for classification using the angle detection is 133.4901 seconds. The CPU time for classification using Mahalanobis distance is 134.1609. For segmentation the CPU time used by the initialization process for first object (round) is 2.0124 seconds, and for second object (rectangular) it is 1.9968 seconds. The CPU time used to find the initial contour for the first object is 2.0280, and for the second object is 1.8096 seconds. The CPU

time of the segmentation algorithm is 0.1872 seconds. Therefore, the total time for our approach is 8.034 seconds, which is significantly faster than the classification approach.

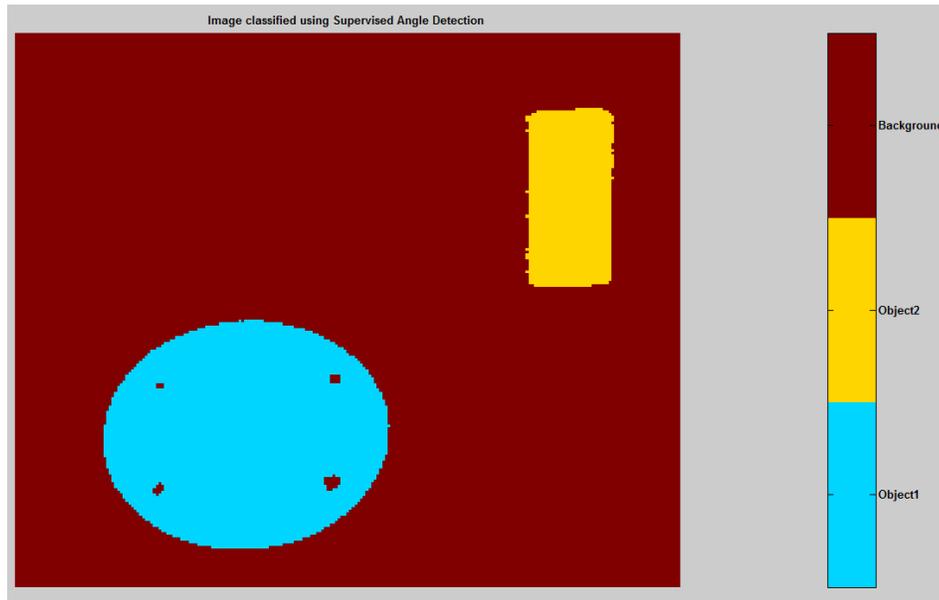


Figure 4-27: Result of the image classification using supervised angle detection for Figure 4-25.

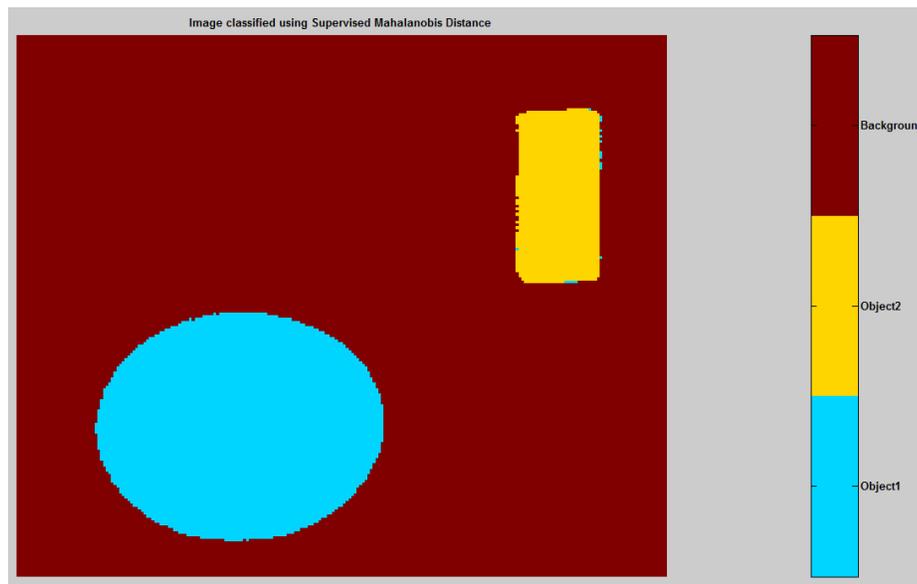


Figure 4-28: Result of the image classification using Mahalanobis distance for Figure 4-25.

5 CONCLUSIONS AND FUTURE WORK

Despite recent advances in hyperspectral image processing, automated object segmentation from hyperspectral image data is still an area of great interest. We have presented a robust method that combines graph cut and active contour methods to segment objects in hyperspectral images. This method produces good object segmentation results. When appropriate input parameters are used, accurate results are obtained.

The results with different sets of hyperspectral images have confirmed the effectiveness of the method for segmenting both large and small objects. It also works well for objects with holes, and objects in images with noise. In addition, the algorithm also runs comparatively quickly.

We developed an application that is easy for the user to use and understand. The algorithm implemented here allows researchers from our laboratory to utilize our existing modules in order to produce new applications, giving them new opportunities to develop better applications.

Up to now, our efforts have been dedicated to the development and testing of the segmentation algorithm, rather than to developing a structured approach to the GUI design and evaluation. This is another direction to be followed in order to increase the usefulness of the project for real-world applications.

Another good future project would be an implementation of this project in the C++ language. This would address the issues of time efficiency, and would also be helpful because it could be used in every platform.

Another future task might be finding another approach to define a cost for cutting mixed pixels of the hyperspectral images; the idea is to decide which metric makes the most sense for this algorithm.

Contributions

This work was presented at the follow events:

1. Poster Session presented at the 10th Annual Research and Industry Collaboration Conference on October 27th 2009, at Northeastern University.
2. Poster Session at the “Computing Alliance of Hispanic-Serving Institutions - CAHSI's 4th Annual Meeting” on April 5-7, 2010, at Redmond, Washington.
3. Paper accepted to “The 2010 International Symposium on Spectral Sensing Research (2010 ISSSR)” venue during the week of June 21-24, 2010, Springfield, Missouri.
4. Paper submitted to “International Journal of Remote Sensing”.

REFERENCES

- [1] S.M. Cruz, "Hyperspectral Image Classification Using Spectral Histograms and Semi-Supervised Learning," University of Puerto Rico - Mayaguez Campus, 2008.
- [2] A.G.D.S. Filho, A.C. Frery, C.C.D. Araújo, H. Alice, J. Cerqueira, J.A. Loureiro, M.E.D. Lima, M.D.G.S. Oliveira, and M.M. Horta, "Hyperspectral Images Clustering on Reconfigurable Hardware Using the K-Means Algorithm," *Proceedings of the 16th symposium on Integrated circuits and systems design*, IEEE Computer Society, 2003, p. 99.
- [3] A. Erturkerturk and S. Erturkerturk, "Unsupervised Segmentation of Hyperspectral Images Using Modified Phase Correlation," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, 2006, pp. 527-531.
- [4] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, USA: 2008, pp. 1-8.
- [5] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation," *Int. J. Comput. Vision*, vol. 70, 2006, pp. 109-131.
- [6] P. Hong, L. Kaplan, and M. Smith, "Hyperspectral image segmentation using filter banks for texture augmentation," *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, 2003*, Greenbelt, Maryland, USA: 2003, pp. 254-258.
- [7] C.P. Lee, "Hyperspectral image segmentation using active contours," *Proceedings of SPIE*, Orlando, FL, USA: 2004, pp. 159-169.
- [8] Ning Xu, R. Bansal, and N. Ahuja, "Object segmentation using graph cuts based active contours," *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, Madison, WI, USA: 2003, pp. II-46-53.
- [9] J. Marot, S. Bourennane, and M. Adel, "Array Processing Approach for Object Segmentation in Images," *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, Honolulu, HI, USA: 2007, pp. I-621-I-624.

- [10] C. Rivera, "The Development of Ground Truth Data and Accuracy Assessment of Hyperspectral Image Classification and Spectral Unmixing," University of Puerto Rico - Mayaguez Campus, 2007.
- [11] Hyperspectral imaging - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Hyperspectral_imaging, April 2010.
- [12] G. Shaw and H. Burke, "Spectral Imaging for Remote Sensing," *Lincoln Laboratory Journal*, vol. Vol. 14, no.1, 2003.
- [13] ENVI Software - Image Processing & Analysis Solutions. <http://www.itvis.com/ProductServices/ENVI/tabid/119/language/en-US/Default.aspx>. March 2010.
- [14] Purdue/LARS MultiSpec. <http://cobweb.ecn.purdue.edu/%7Ebiehl/MultiSpec/>. March 2010.
- [15] J. Duarte, "Geometric Scale-Space Framework for the Analysis of Hyperspectral Imagery," PHD, University of Puerto Rico - Mayaguez Campus, 2007.
- [16] V. Manjit, "Parameter Optimization For Segmenting Structures In CT images," ME in Electronic Instrumentation and Control Engineering, Thapar University, 2008.
- [17] C. Lee, W. Snyder, and C. Wang, "Supervised Multispectral Image Segmentation Using Active Contours," *International Conference on Automation and Robotics, Barcelona*, Apr. 2005.
- [18] G. Sapiro, "Vector-valued active contours," *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, USA: , pp. 680-685.
- [19] P.L. Cheolha, "Robust Image Segmentation using Active Contours: Level Set Approaches," North Carolina State University, 2005.
- [20] Z. Ying, L. Guangyao, S. Xiehua, and Z. Xinmin, "Geometric active contours without re-initialization for image segmentation," *Pattern Recogn.*, vol. 42, 2009, pp. 1970-1976.
- [21] Glossary of graph theory - Wikipedia, the free encyclopedia http://en.wikipedia.org/wiki/Induced_subgraph#Subgraphs, March 2010.

- [22] J.E. R. Mayer, "Segmentation Approach and Comparison to Hyperspectral Object Detection Algorithms," *34th Applied Imagery and Pattern Recognition Workshop (AIPR'05)*, Washington, DC, USA: , pp. 36-41.
- [23] A. Plaza, J. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gaultieri, and J. Tilton, "Advanced Processing of Hyperspectral Images," *2006 IEEE International Symposium on Geoscience and Remote Sensing*, Denver, CO, USA: 2006, pp. 1974-1978.
- [24] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
- [25] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, 2004, pp. 1124-1137.
- [26] E. Arzuaga-Cruz, L. Jimenez-Rodriguez, M. Velez-Reyes, D. Kaeli, E. Rodriguez-Diaz, H. Velazquez-Santana, A. Castrodad-Carrau, L. Santos-Campis, and C. Santiago, "A MATLAB toolbox for hyperspectral image analysis," *IEEE International IEEE International IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004*, Anchorage, AK, USA: , pp. 4839-4842.