REGION-BASED GEOMETRIC ACTIVE CONTOUR FOR CLASSIFICATION USING HYPERSPECTRAL REMOTE SENSING IMAGES

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Lin Yan, M.S.

Graduate Program of Geodetic Science

The Ohio State University

2011

Dissertation Committee:

Rongxing Li, Advisor
Carolyn J. Merry
Alper Yilmaz
© Copyright by

Lin Yan

2011
ABSTRACT

The high spectral resolution of hyperspectral imaging (HSI) systems greatly
enhances the capabilities of discrimination, identification and quantification of objects of
different materials from remote sensing images, but they also bring challenges to the
processing and analysis of HSI data. One issue is the high computation cost and the curse
of dimensionality associated with the high dimensions of HSI data. A second issue is how
to effectively utilize the information including spectral and spatial information embedded
in HSI data.

Geometric Active Contour (GAC) is a widely used image segmentation method
that utilizes the geometric information of objects within images. One category of GAC
models, the region-based GAC models (RGAC), have good potential for remote sensing
image processing because they use both spectral and geometry information in images are
robust to initial contour placement. These models have been introduced to target
extractions and classifications on remote sensing images. However, there are some
restrictions on the applications of the RGAC models on remote sensing. First, the heavy
involvement of iterative contour evolutions makes GAC applications time-consuming and
inconvenient to use. Second, the current RGAC models must be based on a certain
distance metric and the performance of RGAC classifiers are restricted by the
performance of the employed distance metrics.
According to the key features of the RGAC models analyzed in this dissertation, a classification framework is developed for remote sensing image classifications using the RGAC models. This framework allows the RGAC models to be combined with conventional pixel-based classifiers to promote them to spectral-spatial classifiers and also greatly reduces the iterations of contour evolutions. An extended Chan-Vese (ECV) model is proposed that is able to incorporate the widely used distance metrics in remote sensing image processing. A new type of RGAC model, the edge-oriented RGAC model, is also discussed. The RGAC classifier utilizing this new model can be combined with any pixel-based classifier and is much more flexible and adaptive than the ECV models that must be based on a certain distance metric. Classification experiments were performed on two HSI datasets, and the proposed RGAC classifiers were evaluated and compared based on the results of the experiment.

In order to handle the curse of dimensionality of HSI, the nonlinear dimensionality reduction (DR) was tested in the research portion of this dissertation for its ability to distinguish the intrinsic nonlinear structures in HSI. An algorithm of the fast near neighbor search in high-dimensional spaces, locality-sensitive hashing (LSH), is introduced to the nonlinear DR method of Laplacian eigenmaps (LE) for speeding up the k-nearest neighbor search, which is highly computationally expensive and acts as a bottleneck for the nonlinear DR methods. Experimental results demonstrated that the LE combined with LSH perform nonlinear DR on relatively large HSI datasets, and the quality of the DR results are better than the DR results of PCA for hyperspectral image classifications.

The contributions of this dissertation include:
1) Proposes an extended Chan-Vese model (ECV) which is able to incorporate different distance metrics to process multiple-band remote sensing images;

2) Proposes a new type of edge-oriented RGAC model which can be combined with any pixel-based classifiers and has much more adaptability than the ECV models to be applied on remote sensing image classifications;

3) Develops a framework for the RGAC-based image classifications which enables RGAC classifiers to be easily combined with pixel-based classifiers;

4) Compares and evaluates the performance of different types of RGAC; classifiers introduced in this dissertation by experiments on HSI datasets.

5) Introduces the method of LSH-based Laplacian eigenmaps which makes it possible to perform non-linear dimensionality reductions on relatively large hyperspectral datasets.
ACKNOWLEDGMENTS

I would like to give my sincere appreciation to my advisor, Dr. Rongxing Li, who brought me to the Mapping and GIS laboratory at The Ohio State University which has significantly changed my life and career. I would like to acknowledge him for his guidance and unreserved patience throughout my study at OSU.

I would also like to thank Dr. Carolyn J. Merry and Dr. Alper Yilmaz for serving on my dissertation committee as reviewers and examiners, and thank them for their valuable comments and suggestions. I also would like to thank Dr. DeLiang Wang for his suggestions on my research.

I would like to give special thanks to Dr. Kaichang Di and Dr. Xutong Niu who have encouraged and helped me like elder brothers.

I would like to thank the faculty and staff in the Department of Civil and Environmental Engineering and Geodetic Science and the Graduate School who are always willing to provide help to students. They made me love OSU.

I dedicate this dissertation to my parents as a minimal return to what they have done for me. All my successes are devoted to them.

Finally, I wish to express my deepest thanks to my wife, Xin Wang, for all the love and peace she has brought to my life. It is fortune to have her in my life.
This dissertation was carried out at the Mapping and GIS Laboratory at the Ohio State University. The research was supported by the National Geospatial-Intelligence Agency (NGA).
VITA

July 2002 . . . . . . . . BS, Survey Engineering, Tongji University

March 2005 . . . . . . . . MS, Photogrammetry and Remote Sensing, Tongji University

2005 - 2010 . . . . . . . . . Graduate Research Associate at The Ohio State University

PREDEFENSE PUBLICATIONS


FIELDS OF STUDY

Major Field: Geodetic Science

# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>VITA</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xii</td>
</tr>
<tr>
<td>CHAPTER 1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Literature Review</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Issues and Expected Results</td>
<td>12</td>
</tr>
<tr>
<td>1.4 Overview of the Dissertation</td>
<td>17</td>
</tr>
<tr>
<td>CHAPTER 2 REGION-BASED GEOMETRIC ACTIVE CONTOUR</td>
<td>21</td>
</tr>
<tr>
<td>2.1 Geometric Active Contour</td>
<td>21</td>
</tr>
<tr>
<td>2.1.1 Snake Model</td>
<td>21</td>
</tr>
<tr>
<td>2.1.2 Level Set - Implicit Curve Representations</td>
<td>22</td>
</tr>
<tr>
<td>2.1.3 Edge-based GAC Model</td>
<td>25</td>
</tr>
<tr>
<td>2.2 Region-based GAC Model</td>
<td>29</td>
</tr>
<tr>
<td>2.2.1 Chan-Vese Model</td>
<td>29</td>
</tr>
<tr>
<td>2.2.2 Comparison of EGAC and RGAC models</td>
<td>33</td>
</tr>
<tr>
<td>2.3 Applications of RGAC Models on Remote Sensing Images</td>
<td>34</td>
</tr>
<tr>
<td>2.3.1 Two-phase RGAC for Target Detections</td>
<td>34</td>
</tr>
<tr>
<td>2.3.2 Multiphase RGAC for Classifications</td>
<td>38</td>
</tr>
<tr>
<td>2.3.3 Characteristics of the RGAC Classifiers</td>
<td>41</td>
</tr>
<tr>
<td>2.3.4 Summary</td>
<td>47</td>
</tr>
<tr>
<td>CHAPTER 3 THE RGAC CLASSIFIERS FOR HYPERSPECTRAL IMAGES</td>
<td>49</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>49</td>
</tr>
<tr>
<td>3.1.1 Two-stage Strategy for RGAC-based Classifications</td>
<td>50</td>
</tr>
<tr>
<td>3.1.2 A General RGAC Model</td>
<td>52</td>
</tr>
<tr>
<td>3.1.3 Image Force Map</td>
<td>55</td>
</tr>
<tr>
<td>3.2 Different Types of RGAC models for Remote Sensing Image Classifications</td>
<td>57</td>
</tr>
<tr>
<td>3.2.1 Extended Chan-Vese Model</td>
<td>58</td>
</tr>
<tr>
<td>3.2.2 Edge-Oriented RGAC Model</td>
<td>64</td>
</tr>
<tr>
<td>3.3 Classification Framework of RGAC Classifiers</td>
<td>67</td>
</tr>
<tr>
<td>3.3.1 Workflow of the Multiphase Approach</td>
<td>68</td>
</tr>
</tbody>
</table>
3.3.2 Relabeling of the Classification Results

CHAPTER 4 EXPERIMENTS AND DISCUSSIONS
4.1 HSI Datasets and Nonlinear Dimensionality Reduction with LSH-based Laplacian Engienmaps
4.1.1 Description of Datasets
4.1.2 Nonlinear Dimensionality Reduction with Laplacian Engienmaps
4.1.3 Dimensionality Reduction Experiments and Evaluations
4.2 Classification Experiments
4.2.1 Description of Classification Experiments
4.2.2 Experiments on Urban Dataset
4.2.3 Experiments on DC Dataset
4.3 Analysis of the Experimental Results
4.3.1 Overall Evaluations
4.3.2 Effects of the Distribution of Image Force Values
4.3.3 The Over-refinement Problem

CHAPTER 5 CONCLUSIONS

BIBLIOGRAPHY
LIST OF FIGURES

Figure 1.1 Illustration of hyperspectral remote sensing (AVIRIS) concept..................1
Figure 1.2 Comparison of the pixel spectrum of the Salic Fluvisol end member.........3
Figure 1.3 Example of GAC image segmentation..................................................8
Figure 1.4 Example of salt-and-pepper noise..........................................................14
Figure 2.1 Implicit representation of the curve $x^2+y^2=1$.....................................23
Figure 2.2 Illustration of curve curvature values....................................................25
Figure 2.3 Curve evolution with level set function...................................................25
Figure 2.4 An example of curve evolution with the edge-based GAC model..............27
Figure 2.5 An example of the leakage problem in edge-based GAC models.............29
Figure 2.6 An example of curve evolution with the Chan-Vese model....................32
Figure 2.7 Example of road detection using RGAC models.................................36
Figure 2.8 Representation of a partition of the image domain by subregions and their corresponding closed curves..................................................39
Figure 2.9 Comparison of RGAC classifier with conventional classifier.................45
Figure 2.10 Example of binary classification with the ECV-ED model.....................47
Figure 3.1 Procedure of the traditional RGAC classification using the Chan-Vese model.................................................................................................................50
Figure 3.2 Procedure of the two-stage RGAC classification strategy.......................51
Figure 3.3 Illustration of image force map on simulated image.............................56
Figure 3.4 Example of the binary classification on a single-band remote sensing image using the ECV-ED model.................................................................59
Figure 3.5 Illustration of the image force of the ECV-ED model.............................62
Figure 3.6 Image force strength distribution map with overlaid contours..................63
Figure 3.7 Example of binary classification using the EO model.............................67
Figure 3.8 Workflow of the RGAC classifiers.........................................................69
Figure 3.9 Illustration of the relabeling of the suppressed pixels............................73
Figure 3.10 Overall image term matrix and curvature term matrix in the 1st iteration..74
Figure 3.11 Final RGAC classification results.........................................................76
Figure 4.1 RGB image of the Urban dataset and the ground-truth.........................79
Figure 4.2 RGB image of the DC dataset and the ground-truth..............................80
Figure 4.3 Urban LE data.....................................................................................86
Figure 4.4 DC LE data.......................................................................................88
Figure 4.5 PCs of the Urban dataset.................................................................90
Figure 4.6 PCs of the DC dataset.........................................................................91
Figure 4.7 Classification results on the PCs of the Urban dataset..........................92
Figure 4.8 Classification results on the PCs of the DC dataset..............................93
Figure 4.9 Refinement of minimum-distance classification results by ECV-ED and EO-ED on the Urban LE data……………………………………………………………………98
Figure 4.10 Refinement of SAM classification results by ECV-SAM and EO-SAM on the Urban LE data……………………………………………………………………99
Figure 4.11 Refinement of maximum-likelihood classification results by ECV-ML and EO-ML on the Urban LE data……………………………………………………100
Figure 4.12 Refinement of SVM classification results by EO-SVM on the Urban LE data………………………………………………………………………………………………101
Figure 4.13 Refinement of minimum-distance classification results by ECV-ED and EO-ED on the DC LE data. ……………………………………………….………… 102
Figure 4.14 Refinement of SAM classification results by ECV-SAM and EO-SAM on the DC LE data……………………………………………………………….…….… 103
Figure 4.15 Refinement of maximum-likelihood classification results by ECV-ML and EO-ML on the DC LE data. ……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………
LIST OF TABLES

Table 3.1 Values of the elements at position (8, 2) in the first three iterations...........75
Table 3.2 Curvature values at position (8, 2) in Figure 3-2 (a) after three iterations.....75
Table 4.1 Numbers of training and test samples in Urban dataset.........................82
Table 4.2 Numbers of training and test samples in DC dataset............................82
Table 4.3 Running times of k-nearest neighbor search on the two HSI datasets........85
Table 4.4 OAs of the classification results from PCs and LE data of Urban dataset.....94
Table 4.5 OAs of the classification results from PCs and LE data of DC dataset.......94
Table 4.6 OAs of the pixel-based classifiers on the LE data of the two datasets........107
Table 4.7 Highest OAs and corresponding μ of the ECV classifiers on the LE data of the two datasets..........................................................109
Table 4.8 Highest OAs and corresponding μ of the EO classifiers on the LE data of the two datasets..........................................................109
CHAPTER 1
INTRODUCTION

1.1 Background

There have been remarkable advances in remote sensing technologies over the last few decades, particularly with regards to sensor improvements. The increases in spatial, spectral and temporal resolutions of satellite and airborne sensors have greatly enhanced the capabilities of data acquisition, and enabled remote sensing to play critical roles in such varied fields as military, agriculture, and environment monitoring (Borengasser et al., 2007; Lillesand et al., 2008; Li et al. 2010).

The spectral information contained in remote sensing images, recorded as image intensities, represents the reflectance of objects to sunlight measured by passive sensors within a particular spectral range, and is widely utilized for object identification and discrimination. The capabilities of remote sensing images are constrained by several factors; one of the most important is the spectral resolution of the sensor. Based on different spectral resolutions, the satellite and airborne sensors can be divided into three categories: monochrome sensors such as HiRISE (High Resolution Imaging Science Experiment) and LROC (Lunar Reconnaissance Orbiter Camera), multispectral sensors such as SPOT (Système Probatoire d'Observation de la Terre), IKONOS, Quickbird, WorldView, Landsat and AVHRR (Advanced Very High Resolution Radiometer), and
hyperspectral sensors such as EO-1 (Earth Observing-One) hyperion, CASI (Compact Airborne Spectrographic Imager), AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) and HYDICE (Hyperspectral Digital Imagery Collection Experiment).

Figure 1.1. Illustration of hyperspectral remote sensing (AVIRIS) concept (modified from http://aviris.jpl.nasa.gov).

Among the three categories of sensors, the hyperspectral imaging (HSI) system is one of the most valuable advancements in remote sensing technology in the last 20 years of the 20th century. What makes it remarkable is its ability to obtain images of continuous narrow spectral bandwidths, which form a data cube as illustrated in Figure 1.1. As shown in Figure 1.2, the spectrum of pixels as measured on HSI images can provide more subtle information than a multispectral image (Borengasser et al., 2007).
Figure 1.2. Comparison of the pixel spectrum of the Salic Fluvisol end member measured on a) DAIS hyperspectral images and b) ETM+ multispectral images (adopted from T. Schmid et al., 2004)

The high spectral resolution of HSI greatly enhances the capabilities of discrimination, identification and quantification of objects of different materials from remote sensing images, and makes finer classifications possible. However, such a large number of spectral channels create challenges for the processing and analysis of HSI.

1.2 Literature Review

One challenge in the processing of HSI is the curse of dimensionality. "It has been estimated that, as the number of dimensions increases, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities" (Jimenez and Landgrebe, 1998). So the training samples will hardly be adequate to determine reasonable estimates of the probability density functions for standard statistical classifiers. Some approaches have been used to overcome this problem. One approach is
to adaptively estimate the statistics of the classified samples. An initial classification is obtained with available training samples. Then statistics are iteratively updated using the classified samples and training samples until convergence is achieved (Jackson and Landgrebe, 2001). The algorithm of expectation–maximization (EM) (Moon, 1996) is usually employed, "which represents a general and powerful solution to the problem of ML estimation of statistics in the presence of incomplete data" (Melgani and Bruzzone, 2004). Note that although the required number of training samples is low in this approach, it still requires that the initial estimates of the classification statistics obtained from the training samples are accurate enough so as to avoid divergence of the iterative estimation. The main disadvantage of this approach is the time consuming nature of the iterative estimation procedure.

Another approach is to reduce the dimensionality of HSI by feature selection or feature extraction. 'Dimensionality reduction is the transformation of high-dimensional data into a meaningful representation of reduced dimensionality' (Maaten, 2009). The goal of feature selection is to select a subset of the features (bands) that optimally discriminate between some defined classes according to a certain measurement, such as Jeffries–Matusita distance, spectral angle mapper (SAM) and Euclidean minimum distance (Bruzzone et al. 1995; Keshava, 2004).

Feature extraction transforms the original data into an image with lower dimensions, which still contains most of the original information. Some methods, such as decision boundary feature extraction (DBFE) (Lee and Landgrebe, 1993) and projection pursuit (PP) (Jimenez and Landgrebe, 1999), have been proved to be effective for
supervised feature extraction of HSI, and are capable of reducing the number of features with negligible information loss. But these feature extraction methods, together with the feature selection methods mentioned above, have a distinct disadvantage which is that they are all case-specific due to the supervised manner.

On the other side, unsupervised feature extraction methods are more widely used. These methods include PCA (principle component analysis) and MNF (minimum noise fraction), which is based on PCA. These methods have been embedded in the commercial software for remote sensing image processing, such as ERDAS Imagine and ENVI. Although it has been reported that the linear mappings of the original HSI data in PCA can cause distortions that can possibly camouflage subtle differences between spectrally similar targets (Cheriyadat and Bruce, 2003), PCA and MNF are still widely used in the processing of HSI data (Mersereau, 2005; Du and Fowler, 2007; Fauvel et al., 2008; Huang and Zhang, 2008; Zhu et al., 2011).

Recently the non-linear dimensionality reduction (DR) methods have been receiving more attention for feature extractions of HSI. Non-linear DR methods, also known as manifold learning, search the manifold surfaces embedded in the high-dimensional dataset, and project these surfaces onto a lower-dimensional image to compress the original dataset. They can maintain the nonlinear properties of a high-dimensional dataset while preserving a high level of efficiency throughout the data compression process. Because of the nonlinear characteristics of HSI data due to the nature of the scattering, as described in the bidirectional distribution function (BRDF) (Goodin et al., 2004) and other nonlinearity sources such as water, which is a nonlinear
attenuating medium (Mobley, 1994), the non-linear DR methods are theoretically more appropriate for feature extraction of HSI compared to the linear DR methods, such as PCA and MNF.

Some non-linear DR methods, such as isometric mapping (ISOMAP) (Tenenbaum et al., 2000) and local linear embedding (LLE) (Roweis, 2000), have been applied to the feature extraction of HSI (Han and Goodenough 2005; Ainsworth and Fusina, 2005; Ainsworth and Fusina, 2006; Mohan et al. 2007; Wu et al., 2009; Zhou et al., 2009). Promising results were reported that these DR methods were more capable of preserving information and improving target discrimination compared to PCA in the experiments. However, the expensive computation cost, which is $O(n^2 \cdot d)$ ($n$ is the number of datapoints and $d$ is the number of dimensions), of the k-nearest neighbor search, which is required by all non-linear DR algorithms, has made it difficult to apply these methods to large HSI datasets. It has created a bottleneck that prevents them from being widely applied in practice use. So far the most of the HSI datasets that the non-linear DR algorithms directly apply to are limited to $100 \times 100$ pixels. Ainsworth and Fusina (2005, 200&) proposed an approach to utilize a strategy of dividing, conquering, and merging to overcome this problem. A large HSI dataset is split into small tiles ($75 \times 75$ pixels) on which the non-linear DR algorithms are applied, and then a procedure called alignment is conducted to align the manifold coordinates of different tiles into a consistent manifold coordinate system using a common small set of tie-points. Although this approach enables non-linear DR methods to be applied to large HSI datasets, its
effectiveness is restricted by the availability of the tie points, and the alignment using tie-points also causes errors in the final results for the entire dataset.

Many classifications have been proposed for HSI, such as support vector machines (SVM) (Melgani and Bruzzone, 2004; Camps-Valls and Bruzzone, 2005; Hsu et al., 2007) and statistical classifiers (Li et al., 2009). Although these approaches make good use of the rich spectral information in HSI, the spatial information is largely ignored. With the increasing availability of high spatial-resolution HSI acquired by airborne sensors, such as Airborne Visible InfraRed Imaging Spectrometer (AVIRIS, 4-20 m spatial resolution) and Hyperspectral Digital Imagery Collection Experiment (HYDICE, 1-4 m spatial resolution), the combination of spectral and spatial information in the applications of HSI images has become a critical issue.

Several aspects of this issue have been addressed by many researchers. Velasco-Forero and Manian preprocessed the original hyperspectral data by employing spatial smoothing with wavelet and anisotropic partial differential equations (Velasco-Forero and Manian, 2009). Mohan et al. introduced a spatial coherence defined on the local surrounding structure of pixels into the dimensionality reduction of HSI by LLE (Mohan et al. 2007). There are also various spectral-spatial classifiers proposed including Markov’s random fields (Jackson and Landgrebe, 2002; Li et al., 2009), morphological profiles (Palmason et al. 2002; Fauvel et al. 2008), mean-shift (Huang and Zhang, 2008), etc. By combining spectral and spatial information, these approaches have shown improved performance over methods which utilize only spectral information, particularly
in reducing the salt-and-pepper noise that is usually observed in the classification results of hyperspectral images.

Figure 1.3. Example of GAC image segmentation (adopted from Tsai et al., 2001). (a): initial contour, (b), (c): contour evolution, and (d): final segmenting contours.

Active contour is a commonly used image segmentation method that utilizes geometric information of objects in images (Kass et al., 1987; Caselles et al., 1993; Caselles et al., 1997; Malladi et al., 1995). It achieves image segmentation by evolving a curve according to an energy function whose minimum is obtained at the boundaries of the objects in the image. Figure 1.3 shows an example of segmentation on an image with a noisy background using an active contour. The advantages of active contour include its capabilities to suppress noise and obtain smooth segment boundaries, as well as the ease of implementation.

The original form of active contour, the Snake Model, was introduced by Kass et al. (1987), in which the curve is defined explicitly by the curve points. The implicit active contour model, Geodesic or Geometric Active Contour (GAC), implicitly represents the
contours with the zero-level set of a higher-dimension function referred to as the level set function, and the segmentation is achieved by evolving the level set function according to a partial differential equation (PDE) (Caselles et al., 1993). Compared to the Snake Model, GAC can handle the topological changes of the contours when breaking and merging occurs, and has been extensively used.

There are two main categories of GAC models, edge-based GAC (EGAC) and region-based GAC (RGAC). EGAC models feature a speed function to control the evolution speed that approaches zero at the boundaries (Caselles et al., 1993). The speed function is usually a decreasing function of the image gradient, which is calculated before the curve evolution starts, and the output is referred to as a speed map. The edge-based models have been used in a wide range of applications, such as visual tracking (Paragios and Deriche, 2000; Zhang and Freedman, 2003), and building and road extraction on satellite or aerial remote sensing images (Song and Shan, 2008; Niu, 2006). But in some cases the gradient is not a good indicator of boundaries. For example, when the adjacent targets are spectrally similar or there are spectral mixtures associated with the imaging sensors that do not have a very high spatial resolution, the EGAC models can encounter a leakage problem in which the evolving contours pass through the blurred boundaries (Niu, 2006).

The RGAC models define the energy functions based on the Mumford-Shah Functional (Mumford and Shah, 1989). The first RGAC model was proposed by Chan and Vese and is referred to as the Chan-Vese Model (Chan and Vese, 2001). These models have two target values that can be piece-wise constants (Chan et al., 2000; Chan
and Vese, 2001) or piece-wise smoothing functions (Tsai et al., 2001; Vese and Chan, 2002; Li et al., 2007; Lankton and Tannenbaum, 2008), which approximate the interior or exterior regions of the contours. The RGAC models utilize image intensity (spectral) information directly and are better at catching weak boundaries than the EGAC active contour models (Chan et al., 2000; Chan and Vese, 2001). In these models, both direction and speed of the curve evolution at a given pixel position are determined by certain distance measurements of the pixel’s intensity to the target values, and are more robust in the initial contour placement than the EGAC models, which mostly require the initial contours be completed inside or outside of objects in an image. Moreover, the target values are continually updated during the evolution and this self-adaptive mechanism makes the RGAC models have low requirements for the accuracies of the initial target values.

With the advantages of RGAC models, including robustness against initial curve displacement and lower requirements for training datasets, they have been introduced to remote sensing image processing including classifications and target extractions. Konstantinos and Demertre (2009) used a revised Chan-Vese model to extract the man-made objects in a single image, and Ke et al. (2010) adopted another revised Chan-Vese model proposed by Li et al. (2007) for the extraction of tree crowns in single-band images.

The RGAC models are two-phase, i.e., they can only separate an image into two parts, and therefore, are binary classifiers, which can be extended to multiphase. As discussed by Melgani and Bruzzone (2004), there are two approaches to combined binary
classifiers to be able to handle multiple-class issues. One is the hierarchical tree-based approach that uses the architecture of binary hierarchical trees (BHT) with each branch representing a binary classifier. The other is the parallel approach that utilizes a parallel architecture made up of multiple binary classifiers each providing a set of binary classification results.

In the hierarchical classification approach, binary classification is performed on each level of the binary hierarchical tree. The one-against-all (OAA) strategy (Melgani and Bruzzone, 2004) is usually used. On each level the objects of one class are extracted, and further binary separations are only conducted in the other sub-regions of lower levels. In this way the objects of one class are extracted on each level and the objects of different classes are extracted sequentially in different levels. Examples of this approach can be found in Tsai et al. (2002), Ball and Bruce (2006), and Ball and Bruce (2007). Although this approach is simple to implement, its disadvantage is that the binary separations in lower levels of the binary hierarchical trees are dependent upon the separation results occurring on upper levels, and the errors in the upper levels will be propagated to lower levels. Different orders of the extractions of multiple classes will also lead to different classification results.

The parallel level-set-based classification approach requires development of special methods to extend the two-phase RGAC models to multiphase. The method of multiple-level-set (MLS) has been proposed for the multiphase extension, in which multiple level set functions are used to represent multiple classes and usually certain rules are employed to enforce penalty with the goal of eliminating vacuum and overlap
Mansouri et al. (2006) proposed an MLS approach which has been adopted for classification of remote sensing images (Ma and Yang, 2009). In this approach, the ambiguity (vacuum and overlap) in the segmentations is eliminated without the use of any penalty terms by establishing an appropriate correspondence between the regions enclosed by closed plane curves and the regions represented by each of the multiple level set functions. Compared to the hierarchical approach, the implementation of the parallel approach is more complicated, but is more reliable. Moreover, its intrinsic competition mechanism to determine pixel labels is similar to those of the supervised classifiers, such as the maximum-likelihood classifier and the SAM classifier; and this makes it able to incorporate the multiphase RGAC classifiers into the conventional framework of supervised classifications.

1.3 Issues and Expected Results

Much research has been carried out on the processing of HSI data, but some issues still exist. The first is the difficulty in statistical estimations as described in Section 1.2. Another issue is the high computation cost of the high-volume HSI data. The third issue, which is the main issue to be addressed in this dissertation, is how to sufficiently utilize the information embedded in the HSI data. Most of the classifiers proposed for HSI data focus on the analysis of the pixel spectrum within the spectral domain and has been exclusively pixel-based, i.e. each pixel is treated independently without considering its relationship to the spatial content in the image. The thematic maps obtained through classification of the remote sensing images in which each pixel receives a unique label
identifying its ground cover class usually look noisy with a salt-and-pepper appearance in the image (in this dissertation salt-and-pepper noise refers to the small-sized segments in the classification results). Because the noisy pixels are indistinguishable in spectral space, they cannot be detected and eliminated with manipulations there.

An example is given in Figure 1.4. In a HYDICE HSI dataset, pixels of a 100 × 100 window (Figure 1.4 (a)) in three selected bands are projected in a 3-D spectral space to form a three dimensional scatter diagram. Namely, in this diagram, the 3-D coordinates of the pixels represent their intensities of the three bands. A point cluster, shown in red, in Figure 1.4 (b), is selected and its correspondences are shown in yellow in Figure 1.4 (c). It can be seen that the majority of the pixels belong to the surface of a building roof, but there are some noisy pixels that are the roofs of some vehicles in the parking lot. These noisy pixels are spectrally indistinguishable from the building roof pixels in these three bands, and will be classified as building roof with a pixel-based classifier using these three bands, which will result in the salt-and-pepper noise in the thematic map.

Although the noise in image are spectrally indistinguishable from useful pixels in spectral space, if the adjacency relationship of the pixels in image space is considered, they can be easily detected and eliminated because they are mostly small-sized isolated segments. Improvements in the classification system can be achieved by considering the relationship of adjacent pixels, i.e., by performing spectral-spatial classifications (Tarabalka et al, 2010).
Figure 1.4. Example of salt-and-pepper noise. (a) A 100 × 100 window from a HYDICE hyperspectral imagery dataset. The RGB channels are bands: 49, 36, and 18, (b) a 3-D scatter diagram displayed with ENVI n-D Visualizer. A point cluster, shown in red, is highlighted, and (c) corresponding pixels (shown in yellow) of the point cluster selected in spectral space.
As introduced in Section 1.2, region-based geometric active contour (RGAC) has good potential to perform spectral-spatial classifications on HSI because of its easy implementation and the usage of both spectral and geometry information in images. Although some RGAC models have been introduced into remote sensing image processing, most of these applications are restricted to binary classifications on a single-band image (Konstantinos and Demertre, 2009; Ke et al., 2010). The applications of the RGAC models for remote sensing image classifications are limited due to the following reasons.

First, the RGAC classifiers always follow the GAC-type framework that heavily relies on contour evolutions. For instance, the contour evolutions usually start from designated seed regions and expand outward gradually until reaching a stable state (Ball and Bruce, 2006; Ball and Bruce, 2007; Ma and Yang, 2009).

Second, applying the RGAC models on multiple-band remote sensing images require the incorporation of different distance metrics such as Euclidean distance, SAM (spectral angle mapper), Mahalanobis distance, and the likelihood function. But so far only a limited number of these metrics has been introduced into the RGAC models and there have been no research on the comparisons of the RGAC classifiers with the RGAC models based on different distance metrics.

Third, the currently existing RGAC models are all based on a certain distance metric. But because it is difficult to have accurate statistical estimates in HSI data, sometimes the RGAC classifiers suffer from the inaccurate statistical estimates and
therefore the requirement of using a certain distance metric limits the performance of RGAC classifiers on HSI data.

Besides the problem of contour evolutions which is time consuming, the issues above also bring uncertainties to the applications of the RGAC classifiers in remote sensing images. Although users are able to utilize an existent RGAC classifier, such as one based on the likelihood function (Ma and Yang, 2009), to perform the spectral-spatial classification of a multiple-band remote sensing image, the following questions remain. First, how does the performance of the likelihood-function-based RGAC classifier compare to the RGAC classifiers based on other distance metrics that have not been considered, such as Euclidean distance and SAM? Second, how does the performance of the current RGAC classifiers, which must be based on a certain distance metric, compare with the pixel-based classifiers whose discrimination functions are not based on certain distance metrics, such as support vector machines (Melgani and Bruzzone, 2004; Camps-Valls and Bruzzone, 2005; Hsu et al., 2007) and neural networks (Rumelhart et al., 1986)? Third, if the RGAC classifiers cannot outperform the other classifiers, can the RGAC classifiers be combined with them to obtain improved results?

With the goal of addressing the issues above, this dissertation has the following objectives:

1) To create an RGAC model that can incorporate the more used distance metrics in remote sensing image processing since different situations may require the usage of certain distance metrics, and also to develop a more flexible RGAC model that does not
rely on a distance metric and can refine the results of any pixel-based classifiers, including SAM, maximum-likelihood, support vector machines, neural networks, etc.;

2) To improve the classification framework for RGAC classifiers that enables different types of RGAC models to be easily combined with pixel-based classifiers to transform them to spectral-spatial classifiers and also enables the RGAC classifiers not to involve too many contour evolutions;

3) To evaluate the performance of the proposed RGAC classifiers on real HSI datasets to deepen understanding of their advantages and limitations with respect to their applications on remote sensing images including hyperspectral images.

1.4 Overview of the Dissertation

In this dissertation, an extended Chan-Vese model (ECV) is proposed to incorporate different distance metrics. Moreover, based on the analysis on the characteristics of the RGAC models and the remote sensing images, a new type of RGAC model is proposed that is totally independent on the selection of distance metrics and can be combined with any pixel-based classifiers. A classification framework is developed to use these RGAC models for remote sensing image classifications. It has a similar framework with conventional classifications and allows RGAC models to be combined with conventional pixel-based classifiers.

In order to reduce the computation cost and lower the requirements for training samples due to the high dimensions of HSI, the nonlinear dimensionality reduction (DR) of Laplacian eigenmaps (LE) is utilized to reduce the dimensions of HSI data. An
algorithm of fast nearest neighbor search in high-dimensional spaces, locality-sensitive hashing (LSH), is introduced to LE for speeding up the k-nearest neighbor search that is highly computationally expensive and is a bottleneck of the nonlinear DR methods. The LE combined with LSH can effectively perform nonlinear DR on relatively large HSI datasets, and its results are better than those of PCA for hyperspectral image classifications.

The rest of the dissertation is organized as follows.

Chapter 2 introduces general theories of the active contour model and level set, and the EGAC and RGAC models are described and compared with a focus on the latter. Then the applications of the RGAC models on remote sensing images, including target extractions (binary classification) and classifications are described. The characteristics of the RGAC models are analyzed to identify their unique features to indicate how the RGAC-based classifications can be improved.

Chapter 3 presents the methodology of the RGAC-based classifications on remote sensing images. The extended Chan-Vese model is proposed to incorporate different distance metrics to process multiple-band remote sensing images, and a new type of RGAC model is proposed that does not require the utilization of a certain distance metric and has much more adaptability than the CV models. The method revised from the MLS method (Mansouri et al., 2006) is used to extend the RGAC models to multiphase, and the two-stage classification strategy is used such that the RGAC-based classifications can have a similar framework with that of traditional image classifications.
Chapter 4 begins by introducing the HSI datasets used in the experiments and the preprocessing applied on them, including the nonlinear DR of LE combined with LSH. The DR results of LE are evaluated with a comparison with PCA. Then classifications with different RGAC classifiers and conventional pixel-based classifiers are performed, and evaluations and comparisons on these classifiers are conducted.

Chapter 5 summarizes the dissertation and points out the direction for future research.

In summary, the contributions of this dissertation include:

1) Proposes an extended Chan-Vese model (ECV) which is able to incorporate different distance metrics to process multiple-band remote sensing images;

2) Proposes a new type of edge-oriented RGAC model which can be combined with any pixel-based classifiers and has much more adaptability than the ECV models to be applied on remote sensing image classifications;

3) Develops a framework for the RGAC-based image classifications, which enables RGAC classifiers to be easily combined with pixel-based classifiers;

4) Compares and evaluates the performance of different types of RGAC classifiers introduced in this dissertation by experiments on HSI datasets;
5) Introduces the method of LSH-based Laplacian eigenmaps which makes it possible to perform non-linear dimensionality reductions on relatively large hyperspectral datasets.
CHAPTER 2

REGION-BASED GEOMETRIC ACTIVE CONTOUR

Two categories of GAC models, edge-based GAC (EGAC) models and region-based GAC (RGAC) models, are introduced and compared in this chapter. Focus will be placed on the RGAC models, which are more suitable for remote sensing image processing. This will be followed by an in depth analysis of both the RGAC models’ advantages and aspects that need improvement, particularly with regard to RGAC-based classifications.

2.1 Geometric Active Contour

2.1.1 Snake Model

The original form of active contour, the Snake Model, was introduced by Kass et al. (1988). Let \( C(p) = [x(p), y(p)], p \in [0,1] \) that represents a parameterized closed planar curve. The energy function of a curve \( C \) on a grayscale image \( I \) is defined as follows:

\[
E(C) = \alpha \int \|C'(p)\|^2 dp + \beta \int \|C''(p)\|^2 dp - \lambda \int \|\nabla I(C(p))\|dp
\] (2-1)

where \( \sigma, \beta \) and \( \lambda \) are positive constants and \( \nabla I \) is the gradient of the image intensity.

In Equation (2-1), the first term represents the ‘tension’ of the curves as the 1st order derivative discourages stretching and causes the curve to behave like a membrane.
The second term denotes the 'rigidity' of the curve, since the 2\textsuperscript{nd} order derivative resists bending and causes the curve to act like a rigid rod. These two terms control the smoothness of the detected curve and are referred to as internal energy. The tension and rigidity of the curve are controlled by the two coefficients, \( \sigma \) and \( \beta \), respectively. The third term attracts the contour towards the object in the image and is referred to as external energy.

Solving the problem of snakes involves finding the curve \( C \) that minimizes \( E \) given an image \( I \) and a set of constants \( \sigma, \beta \) and \( \lambda \) by moving the vertices of the parameterized curve (Kass et al., 1988). The major weakness of the snake model is that it is unable to directly handle topology changes. When there is more than one object in an image, additional topology-handling procedures must be added, which can significantly increase the complexity of the snake model implementation. So Geometric Active Contour based on the method of level set was proposed to solve this problem.

2.1.2 Level Set– Implicit Curve Representations

In the Snake Model, the curves are explicitly represented by a set of points discretized from the parameterized curves. The contours, if defined as closed curves, can also be implicitly represented with a higher-dimensional function. In general, a closed \( n-1 \) dimensional interface can divide an \( n \)-dimensional space \( R^n \) into two separate subdomains consisting of the inside regions and outside regions of the interface, respectively. In the case of a two-dimensional space, the low-dimensional interface is a closed curve. Consider the function \( \phi(\overline{x}) = x^2 + y^2 - 1 \) where \( \phi(\overline{x}) = 0 \) isocontour is the
interface that is a unit circle defined by \( \partial \Omega = \{ \bar{x} | \bar{x} = 1 \} \). \( \phi \) is implemented as a signed distance function (see Osher and Fedkiw, 2003 for details). As illustrated in Figure 2.1, the interior region is the unit open disk \( \Omega = \{ \bar{x} | \bar{x} < 1 \} \), and the exterior region is \( \Omega^{+} = \{ \bar{x} | \bar{x} > 1 \} \).

![Figure 2.1. Implicit representation of the curve \( x^2 + y^2 = 1 \) (adapted from Osher and Fedkiw, 2003).](image)

The uniform Cartesian grid is usually used to implicitly represent a \( \phi(\bar{x}) = 0 \) isocontour within a bounded subdomain \( D \in \mathbb{R}^2 \). It is defined as \( \{(x_i, y_j) | 1 \leq i \leq m, 1 \leq n \leq n\} \) with equal intervals in \( x \) and \( y \) directions, where \( m \) and \( n \) are column and row numbers, respectively. This implicit interface representation introduces some very powerful geometric tools (Osher and Fedkiw, 2003). For example, whether a point \( \bar{x}_o \) is inside the interface or not can be determined by simply checking the sign \( \phi \) at that position. That is,
\( \vec{x}_o \) is inside the interface if \( \phi(\vec{x}_o) < 0 \), outside the interface if \( \phi(\vec{x}_o) > 0 \), and on the interface if \( \phi(\vec{x}_o) = 0 \).

In two-dimensional space, the unit (outward) normal of a point on the \( \phi(\vec{x}) = 0 \) isocontour is defined as (Osher and Fedkiw, 2003)

\[
\vec{N} = \frac{\nabla \phi}{|\nabla \phi|} \quad (2-2)
\]

where \( \nabla \phi \) is the gradient defined as \( \nabla \phi = \left( \frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y} \right) \).

The mean curvature of the interface is defined as the divergence of the normal

\[
\kappa = \nabla \cdot \vec{N} = \frac{\partial n_1}{\partial x} + \frac{\partial n_2}{\partial y} = \frac{\phi_{x}^{2} \phi_{yy} - 2 \phi_{x} \phi_{y} \phi_{yx} + \phi_{y}^{2} \phi_{xx}}{|\nabla \phi|^3} \quad (2-3)
\]

As shown in Figure 2.2, \( \kappa > 0 \) in convex regions, and \( \kappa < 0 \) in concave regions.

The evolution of the curve is implemented as follows. Let a time-varying level set function \( \phi(x, y; t) \) represent a curve \( C \) at different time \( t \), namely

\[
C(t) = \{(x, y) | \phi(x, y; t) = 0 \}, \text{ and the 0-level-set of } \phi \text{ is represented by }
\]

\[
\phi(C(t); t) = 0. \quad (2-4)
\]

The curve \( C \) evolves as \( \phi \) changes. An example of the curve evolution based on a time-varying level-set function can be seen in Figure 2.3, which shows the level set expanding outward at a constant speed in all directions, and as time passes, the level set function expands and the corresponding zero-level set curve transforms into a circle with a larger radius.
Figure 2.2. Illustration of curve curvature values (adapted from Osher and Fedkiw, 2003).

Figure 2.3. Curve evolution with level set function (adapted from Malladi et al., 1995).

2.1.3 Edge-based GAC Models

The method of Geodesic or Geometric Active Contour (GAC) was proposed by Caselles et al. (1993) and Malladi et al. (1995) independently. GAC implicitly represents the contours with the zero-level set of a higher-dimension function, which is usually a signed-distance function. There are two categories of active contour models: edge-based
GAC models (EGAC) and region-based models (RGAC). The GAC model proposed by Caselles et al. and Malladi et al., as well as the Snake model, are EGAC models, which means that the contour evolution speed is based on the edge information in images. The evolution function of a time-varying curve $C$ is as follows: (see Appendix B in Caselles et al. 1997 for detailed derivation)

$$\frac{\partial C(t)}{\partial t} = g(I)\kappa\vec{N} - (\nabla g(I) \cdot \vec{N})\vec{N}$$

(2-5)

where $g$ is a positive and decreasing function, such that $\lim_{x \to \infty} g(x) = 0$.

With the representation of the normal using the level set function $\phi$ (Equation (2-2)), the evolution function of $\phi$ becomes (Caselles et al. 1997)

$$\phi_t = g(I)\nabla \phi \kappa + \nabla g(I) \cdot \nabla \phi.$$  

(2-6)

By adding a constant velocity term to push the contour evolution, the following evolution function, which is the core of the EGAC model, is obtained (Caselles et al. 1997).

$$\phi_t = g(I)\nabla \phi (\kappa + c) + \nabla g(I) \cdot \nabla \phi$$

(2-7)

where $c$ is a positive constant. There are three terms on the right side of the evolution function of Equation (2-7). Detailed explanations concerning the roles of the three terms with examples can be found in Section 3.1 in Tsang (2004).

The function $g$ acts as an edge-detector that yields high values on homogeneous regions and approaches zero at the edges of images. It is usually defined on image gradient and the return values are $[0, 1]$. For instance,
\[ g(I) = \frac{1}{1 + \left| \nabla [G_\sigma \ast I(x, y)] \right|^p}, p=1, 2, 3\ldots \] (2-8)

where \( G_\sigma \) is a Gaussian filter to smooth the whole image in order to reduce noise.

An example of contour evolution with the EGAC model of Equation (2-7) is shown in Figure 2.4. Figure 2.4 (a) illustrates an initial contour bounding three objects in a grayscale image. Figures 2.4 (b)-(f) demonstrate the evolution of the contours until they reach a stable state (Figure 2.4 (f)). The inside regions of the contours, which are the elements with negative values in the matrix of the two-dimensional level set function on \( \phi \), are the segmented objects. In this example, the initial contour is a closed curve bounding all the objects in the image, and the contours evolve in the inward direction. The initial contours can also be placed inside the objects where they are able to evolve outward to the object’s boundaries. For example, when EGAC is used to detect highways on high-resolution aerial images (Niu, 2004), seed points are manually placed around the centerlines of the highway and propagate outwards to the edges of the highway in the images.

![Figure 2.4](image)

Figure 2.4. An example of curve evolution with the edge-based GAC model (adapted from Tsang, 2004).
The EGAC models have two main disadvantages. First, the contours can only evolve in one direction, either inward or outward. This means that the initial contours must be carefully placed so as to ensure that they will not intersect with any objects within the image. User interactions are usually required to help designate the initial contours either bounding or inside the objects in the applications on remote sensing images, and it is difficult to automatically generate initial contours without these intersections.

Second, the contour evolutions of EGAC models are vulnerable to weak edges because the edge-detector $g$ in EGAC models is mostly defined by image gradients. The gradient is sensitive to noise in an image and can possibly interfere with or even halt the contour evolution. So a smoothing filter, such as a Gaussian filter (see Equation (2-8)), is usually needed to preprocess the image in order to reduce the noise. Moreover, the gradient is not a good indicator of weak edges in an image, and this can cause a 'leakage' problem when the evolving contours pass through the weak edges (Niu, 2004; Tsang, 2004). This issue can be made even worse by the smoothing filter, which further blurs the edges. This explains the difficulties of applying the EGAC models to remote sensing images because blurred edges can always be expected in these types of images due to the phenomenon of spectral mixture. An example of the leakage problem is shown in Figure 2.5 where it can be seen that the leaking contours pass through the boundary and continue along until they are blocked by strong edges. So this problem, when it happens, is usually serious enough to fail the case.
Figure 2.5. An example of the leakage problem using edge-based GAC models (adapted from Niu, 2004). (a) The contours (red lines) emerge from a seed point (green crossing) selected within a highway in an image, (b) leakage appears on highway boundaries with weak edges, and (c) leaking contours continue propagating until being blocked by another strongly defined boundary.

2.2 Region-based GAC models

2.2.1 Chan-Vese Model

The first region-based GAC (RGAC) model was proposed by Chan and Vese, and therefore is referred to as the Chan-Vese model (Chan and Vese, 2001). The evolution function is as follows: (see Chan and Vese, 2001 for the details of the model derivation)

\[
\frac{\partial \phi}{\partial t} = \delta_x(\phi) \left[ \lambda_1 (I - c_1)^2 - \lambda_2 (I - c_2)^2 + \mu \cdot \kappa + v \right].
\]  (2-9)

where \( \mu \geq 0, v \geq 0, \lambda_1 > 0, \lambda_2 > 0 \) are fixed coefficients, \( c_1 \) and \( c_2 \) are two target values that are mean intensities of the image areas inside and outside the contours, respectively, \( \kappa \) is the mean curvature of the contours as defined by Equation (2-3), and \( \delta_x \) is the
following approximated Delta function to constrain the update of \( \phi \) occurring only within a narrow band of the current contours.

\[
\delta_{\varepsilon}(x) = \begin{cases} 
1, & \text{if } x = 0 \\
0, & \text{if } |x| < \varepsilon \\
\frac{1}{2\varepsilon} \left[ 1 + \cos \left( \frac{\pi x}{\varepsilon} \right) \right], & \text{otherwise}
\end{cases}
\] 

(2-10)

Equation (2-9) calculates the update of \( \phi \) at every pixel position, and the curve evolution is performed by calculating the update and applying it to \( \phi \) iteratively. The two target values \( c_1 \) and \( c_2 \) are updated during each iteration and are thus self-adaptive. The function of the term \( v \) is to reduce noise by pushing small segments into the background. Because this function can also be performed by the curvature term, in practice the following simplified evolution function that does not contain this term is usually used (Lankton and Tannenbaum, 2008; Karantzalos and Argilas, 2009; Ke et al., 2010).

\[
\frac{\partial \phi}{\partial t} = \delta_{\varepsilon}(\phi) \left[ (I - c_1)^2 - (I - c_2)^2 + \mu \cdot \kappa \right]
\] 

(2-11)

The first two terms on the right side of Equation (2-11) are referred to as the image term. At a given pixel position, the similarities (distances) of the pixel's intensity to the two target values \( c_1 \) and \( c_2 \) are calculated. The image term obtains the difference of the two distances, and the sign indicates the direction of evolution. If the image term is positive, it means the pixel is more similar to \( c_2 \) (averaging of the outside regions) because the positive sign of the image term implies \((I - c_1)^2 > (I - c_2)^2\). Then because the interior regions in \( \phi \) are negative and the exterior regions are positive, the positive image term attempts to push the contour at that position inward and exclude the pixel out of the
contours, i.e., push the pixel into the background segments. On the other hand, if the image term is negative, it pushes the contour outwards so the pixel is included in the foreground objects (regions inside the contours). When the contours reach object boundaries, the update of the regions inside $\phi$ are negative while the update of the regions outside $\phi$ are positive. Then the contours, which are the 0-level sets of $\phi$, will remain stationary, thus the evolution reaches a stable state.

The direction of evolution and the speed calculated from the image term are adjusted by the curvature term, which functions in the same way as it does in the EGAC models to control the smoothness of the contours and suppress noise. With the functions of the image term and curvature term, the Chan-Vese model separates an image into foreground (pixels inside the contours) and background (pixels outside the contours) based on their intensities. So it is a binary classifier from the perspective of image classifications. Figure 2.6 shows an example of binary classification using the Chan-Vese model. Figure 2.6 (a) is the third band of the dimensionality-reduced HYDICE HSI dataset (details on the dataset and the dimensionality reduction are described in Chapter 4). A set of evenly distributed circles are provided as initial contours on a grayscale image. Figures 2.6 (b) and (c) show an intermediate state and the final phase of the contour evolution, respectively. The final binary classification results, which correspond to the stable state of the contour evolution, are shown in Figure 2.6 (d).
Figure 2.6. An example of curve evolution with the Chan-Vese model. (a) Band 3 of the dimensionality-reduced HYDICE HSI dataset, (b) contours after 10 iterations, (c) stable state of the contour evolution after 110 iterations, and (d) final segmentation results that include the foreground objects (shown in brown) and background objects (shown in blue).
2.2.2 Comparison of EGAC and RGAC Models

As discussed above, EGAC models segment an image based on the detected edges, while RGAC models segment an image based on the image intensities and therefore are considered binary classifiers. The major differences between the two are as follows.

1) The RGAC models are more robust to initial contour placement and it is much easier to automatically generate initial contours. The classical EGAC models can only expand in one direction, either inward or outward. This leads to a strict constraint on the placement of the initial contours, which must not be allowed to intersect with objects in the image. It is difficult to automatically generate initial contours without these intersections, which are usually required to designate the initial contours manually. However, this constraint is not a problem for the RGAC models because they can evolve in both directions, so the initial contours can intersect with objects in an image, and simple algorithms, such as intensity thresholding, can be used to automatically obtain the initial contours.

2) The RGAC models do not have the leakage problem. As mentioned in Section 2.1.3, the EGAC models have this issue because the gradient does not effectively catch weak edges in an image. On the other hand, the image term in the RGAC models is based on the comparison of image intensities. So wherever there are two adjacent objects with different intensities, the contour evolution will always be stationary when the contours reach the boundaries between the two objects, no matter whether the edges are sharp or blurred. In the example shown in Figure 2.5, the leakage problem is fixed by adding the
constraint of regional intensity information (Niu, 2004).

3) The RGAC models are binary classifiers that directly utilize the intensity information in image. This flexibility allows for more applications in remote sensing image processing such as classifications, target detections, and anomaly detections.

2.3 Applications of RGAC Models for Remote Sensing Images

From the discussions in the previous section, it can be understood that the RGAC models have many advantages over the EGAC models and are more suitable for remote sensing applications. The two-phase Chan-Vese model and its variations, which are binary classifiers, can be used for target extractions, and can also be used for classifications when being extended to multiphase.

2.3.1 RGAC for Target Extractions

The current RGAC models, which include CV models, have been used for target extractions on multispectral and hyperspectral remote sensing images. Most of these models are designed to focus on only a single-band image (Karantzalos and Argilas, 2009; Ke et al., 2010). It can be a selected band from the original images, a luminance image generated from the RGB bands \(\text{luminance} = 0.299 \times \text{red} + 0.587 \times \text{green} + 0.114 \times \text{blue}\) (Hanbury, 2003), or data in the format of an image transformed by some discrimination functions such as NDVI (Normalized Difference Vegetation Index) (Lillesand et al., 2008).
When the CV models are applied to a selected band of an image or a luminance image, some requirements for the targets of interest need to be met. First, the targets to be extracted must be spectrally distinct from all other objects (Chan and Vese, 2001). Second, they should stay on one end of the histogram of the image (Karantzalos and Argilas, 2009). For example, the targets should be either the brightest or darkest objects in a grayscale image. Otherwise the targets brighter or darker than the targets of interest will quite likely be included in the extraction results. When these requirements cannot be met, some other constraints need to be enforced, or revisions to the Chan-Vese model are required.

An image depicting road extraction is shown in Figure 2.7, in which the RGAC model is applied to aerial remote sensing images with 1m spatial resolution. The image that the RGAC model works on is a luminance image computed from the original RGB images (Karantzalos and Argilas, 2009). Figure 2.7 (a) shows the final contours overlaid on the luminance image. Because not only roads but also some other man-made objects such as buildings have greater intensities, some non-road objects have been included within the contours.

In order to solve this problem, Karantzalos and Argilas revised the Chan-Vese model, thus creating the model below (Karantzalos and Argilas, 2009)

\[
\frac{\partial \phi}{\partial t} = \delta_{\varepsilon} (\phi) \left[ \frac{(I - c_1)^2}{\sigma_1^2} - \frac{(I - c_2)^2}{\sigma_2^2} + \mu \cdot \kappa \right]
\] (2-12)
Figure 2.7. Example of road detection using RGAC models (adopted from Karantzalos and Argilas, 2009). (a) Final state of contour evolution using the Chan-Vese model, and (b) final state of contour evolution using the revised Chan-Vese model.

where $\sigma_1^2$ and $\sigma_2^2$ are the variances of the intensities of the inner and outer regions, respectively.

The improved results using the revised model are shown in Figure 2.7 (b), in which most of the non-road objects are excluded from the contours. The reasons for the improvements are as follows. In this example the two target values $c_1$ and $c_2$ in the Chan-Vese model represent the mean intensities of the roads (regions inside the contours) and background (regions outside the contours), respectively. The reason that some non-road
objects are included in the contours when the Chan-Vese model is employed is that the intensities of these objects are more similar to $c_1$ than $c_2$, i.e. $|l(i, j) - c_1| < |l(i, j) - c_2|$, although they are somehow distant from $c_1$. In the revised model the intensities of the inside and outside regions are normalized. Because the background contains more objects with spectral variety, it can be expected that the intensity variance of the outside regions would be greater than that of the inside regions, i.e. $\sigma_1^2 < \sigma_2^2$. For some pixels that were originally included in the contours but are somehow distinct from $c_1$, after the normalization their similarity to the two target values changes to $\frac{|l(i, j) - c_1|}{\sigma_1^2} > \frac{|l(i, j) - c_2|}{\sigma_2^2}$, so they are ultimately excluded from the contours. This example indicates that if the background has greater spectral variety than the focus object, the revised model (Equation (2-12)) can help the detection results to be more precise by excluding some outlier pixels in the foreground objects that are more similar to $c_1$ and $c_2$ but are distant from $c_1$.

Another revised Chan-Vese model proposed by Li et al. (Li et al., 2007), shown below, is used for the detection of tree crowns (Ke, et al., 2010).

$$\frac{\partial \phi}{\partial t} = \delta_x (\phi)(e_1 - e_2 + \mu \cdot \kappa) + \mu(\nabla^2 \phi - \kappa)$$ (2-13)

where $e_1$ and $e_2$ are functions as below

$$e_1(x) = \int_\Omega K_\sigma(y - x)|l(x) - f_1(y)|^2 dy$$

$$e_2(x) = \int_\Omega K_\sigma(y - x)|l(x) - f_2(y)|^2 dy$$ (2-14)
In Equation (2-13), $K_u(u)$ is a Gaussian kernel function that decreases when $u$ increases, and $f_1(y)$ and $f_2(y)$ are two functions that fit the image intensities near the point $x$ (see Li et al. 2007 for details). The additional term $\mu(\nabla^2\phi - \kappa)$ is a penalty term that ensures the smoothness of the level set function (Li, et al., 2005). The major difference of this revised model from the original Chan-Vese model is that two locally fitted intensities replace the two constants $c_1$ than $c_2$, which are global means. It performs better at extracting tree crowns than the Chan-Vese model because of the intensity variances present in large tree crowns of the high-resolution remote sensing images, and the revised model is less like to split these crowns during the extraction process Ke et al. (2010).

Aside from the revisions to the model, some constraints are usually enforced in the applications of the RGAC models. The most commonly used constraint is the size of the extracted segments. This constraint is found in both of the two above examples in which certain thresholds on target sizes are provided to filter out smaller segments as a post-processing of the original RGAC extraction results.

2.3.2 Multiphase RGAC for Classifications

The RGAC models can be extended to multiphase for image classifications. As mentioned in Section 1.2, Mansouri et al. (2006) proposed an MLS approach which has been adopted for classification of remote sensing images (Ma and Yang, 2009). In this approach, the ambiguity (vacuum and overlap) in the segments is eliminated by establishing an appropriate relationship, shown below, between the regions enclosed by
closed plane curves and the regions represented by each of the multiple-level set functions.

Figure 2.8. Representation of a partition of the image domain by subregions and their corresponding closed curves (illustration for four regions) (adopted from Mansouri et al., 2006).

An image domain is split into a family of $N$ subregions $\{R_j\}_{j=1}^N$, corresponding to a family of $N$ parameterized time-varying closed curves $\tilde{\gamma}_j(x(s,t),y(s,t)):[0,1] \rightarrow \Omega$, $j = 1, 2, ..., N-1$. Let $R_{j-}$ and $R_{j+}$ denote the regions inside and outside the curve $\tilde{\gamma}_j$, respectively. For a region set $R_j$ ($1 < j < N$) is associated with the areas $R_{\tilde{\eta}}^c \cap R_{\tilde{\eta}}^c \cap \cdots \cap R_{\tilde{\eta}_j}^c \cap R_{\tilde{\eta}_j}$ that are represented by the curves. Then the family can be understood as $\{R_{\tilde{\eta}}^c \cap R_{\tilde{\eta}}^c \cap \cdots \cap R_{\tilde{\eta}_j}^c \cap R_{\tilde{\eta}_j} \cdots \}$ a section of the entire image domain. The relationships between regions and curves are illustrated by Figure 2.8. This representation avoids the possible ambiguities of overlaps and gaps among the
partitioning subregions. Equation (2-15) is the energy function composed of multiple level set functions each defined on its own region (Mansouri et al., 2006).

\[
E\left[\{\sum_{j=1}^{N} \xi_j(x, y)dx dy\right] = \int_{R_j} \xi_1(x, y)dx dy + \int_{R_j \cap R_{j+1}} \xi_2(x, y)dx dy + \cdots + \int_{R_j \cap R_{j+1} \cap \cdots \cap R_N} \xi_k(x, y)dx dy + \cdots + \int_{R_j \cap R_{j+1} \cap \cdots \cap R_N} \xi_N(x, y)dx dy + \mu \sum_{j=1}^{N-1} \int_{R_j} ds 
\]

\[
(2-15)
\]

\(\xi_j(x, y)\) is an image function expressing distances of pixels to the spectral statistics of region \(R_j\). \(\xi_j(x, y)\) can have different forms. For instance, assuming that the image follows a Gaussian distribution, then \(\xi_j(x, y)\) can be defined as Equation (2-16) or Equation (2-17) for the maximum-likelihood classifier:

\[
\xi_j(x, y) = -\frac{1}{2} \ln(|\text{cov}_j|) - \frac{1}{2} (I(x, y) - c_j)^{\top} \text{cov}_j^{-1} (I(x, y) - c_j) 
\]

\[
\xi_j(x, y) = \frac{1}{(2\pi)^{n/2} |\text{cov}_j|^{1/2}} \exp \left(\frac{1}{2} (I(x, y) - c_j)^{\top} \text{cov}_j^{-1} (I(x, y) - c_j)\right) 
\]

where \(c_j\) and \(\text{cov}_j\) are the mean and covariance matrix of class \(j\), respectively. The sum of the first \(N\) integral term in Equation (2-15) reflects the competition among \(N\) classes. The last term is the common curvature term. Each class \(j\) has an associated level set function \(\phi_j\), and the inside regions \(R_j^c\) and outside regions \(R_j^e\) correspond to areas represented by \(\phi_j < 0\) and \(\phi_j \geq 0\), respectively. Each \(\phi_j\) has its own evolution function as follows:
\[
\frac{\partial \phi_i}{\partial t} = \delta_i(\phi_i)\left[ -\xi_i + \Phi_i + \mu \cdot \kappa_i \right]
\]

\[
\frac{\partial \phi_j}{\partial t} = \delta_j(\phi_j)\left[ -\xi_j + \Phi_j + \mu \cdot \kappa_j \right]
\]

\[
\frac{\partial \phi_{N-1}}{\partial t} = \delta_{N-1}(\phi_{N-1})\left[ -\xi_{N-1} + \Phi_{N-1} + \mu \cdot \kappa_{N-1} \right]
\]

(2-18)

Let \( \chi \) be the indicator function of the regions denoted by \( R \), defined by

\[ \chi_R(x, y) = 1 \text{ if } (x, y) \in R, \text{ and } \chi_R(x, y) = 0 \text{ otherwise.} \]

\( \Phi_j \) is defined as follows and yields the distance \( \xi \) of each pixel to its currently associated class in the domain of \( R_{j+1} \) (Mansouri et al., 2006).

\[
\Phi_j(x, y) = \xi_{j+1}(x, y) \chi_{[\phi_j(x, y) < 0]}(x, y) + \xi_{j+2}(x, y) \chi_{[\phi_j(x, y) < 0]}(x, y) \chi_{[\phi_{j+1}(x, y) < 0]}(x, y) + \cdots + \xi_{N}(x, y) \chi_{[\phi_{j+1}(x, y) < 0]}(x, y) \cdots \chi_{[\phi_{N-2}(x, y) < 0]}(x, y) \chi_{[\phi_{N-1}(x, y) < 0]}(x, y) \]

(2-19)

According to the relationships of the regions of different classes defined above, \( \Phi_j \) traverses every pixel in \( R_j \), which are regions outside the curve \( \bar{y}_j \), and calculates the spectral statistical properties at every position. This approach was adopted by Ma and Yang (2009) for unsupervised classification of high-resolution remote sensing images.

2.3.3 Characteristics of the RGAC Classifiers

From the perspective of image classification, the RGAC models function as binary classifiers to separate an image into two subregions. In the case of binary classifications, compared to conventional pixel-based classifiers, there are two characteristics of the RGAC models that require further discussions.
First, the functionalities of smoothness, control of segment boundaries and noise suppression made possible by the trade-off between the image term and the curvature term are the key advantages of the RGAC models as spatial-spectral classifiers. The RGAC models have two commonly recognized advantages. The first is the self-adaptation mechanism of the two target values \( c_1 \) and \( c_2 \), which are the spectral statistics of the target objects. The second is the smoothness control of the segment boundaries and noise suppression. However, the self-adaptations of target spectral statistics can be regarded as a typical expectation–maximization (EM) strategy used in image classifications (Moon, 1996). As a matter of fact, if the initial estimates of the spectral statistics are accurate enough, the self-adaptations are not necessary; particularly, the updates of the two target values in the RGAC models do not have to be carried out during the contour evolutions. The unique aspects of the RGAC models are the smoothness control of the segment boundaries and noise suppression.

Second, for the CV classifiers (RGAC classifiers based on the CV models), their classification results are fundamentally determined by the selected distance metric in the image term, and the role of the curvature term is to refine the classification results. The image term can embed different distance metrics to represent the similarity of a pixel's intensity in relation to the two target values (spectral statistics of the foreground and background). The distance metric is a discrimination function such as Euclidean distance and likelihood function that are commonly used in conventional pixel-based classifiers.

The CV models that incorporate the distance metrics, such as Euclidean distance and SAM, to work on multiple-band images are considered as extended Chan-Vese
models (ECV), and the corresponding models are referred to as ECV-ED and ECV-SAM. The ECV models will be further discussed in Section 3.2.1. In this section the classifiers based on the ECV models are generally referred to as CV classifiers, or specifically referred to as ECV-ED classifier and ECV-SAM classifier.

Two examples are given below. The first one shows that by setting the curvature coefficient to $v = 0$, a CV classifier can perform equivalently as the conventional pixel-based classifier when the same distance metric is used. The second example shows that the results of the CV classifiers are fundamentally determined by the selected distance metric, and the role the RGAC models play is to refine the classification results.

Figure 2.9 (a) is a false color image which is composed of three dimensionality-reduction band of a HSI dataset (details of this dataset and DR processing are described in Section 4.1). First SAM is selected as the distance metric. Two classifiers are used: the CV classifier using SAM as the distance metric (ECV-SAM) and the conventional SAM classifier. The SAM of two vectors $\vec{v}_1$ and $\vec{v}_2$ is defined as follows:

$$\text{SAM} (\vec{v}_1, \vec{v}_2) = -\frac{\vec{v}_1 \cdot \vec{v}_2}{\|\vec{v}_1\| \|\vec{v}_2\|}$$  \hspace{1cm} (2-20)

Let $c_{k1}$ and $c_{k2}$ denote the averages of the regions inside and outside the contours, respectively, on the $k^{th}$ band of a multiple-band image. They form two vectors $\vec{c}_1 = (c_{11}, c_{21}, \cdots, c_{N1})$ and $\vec{c}_2 = (c_{12}, c_{22}, \cdots, c_{N2})$, referred to as target vectors. The two target vectors are the spectral signatures of the foreground and background segments, respectively. The corresponding RGAC model is

$$\frac{\partial \phi}{\partial t} = \delta_x (\phi) (\text{SAM}_1 - \text{SAM}_2 + \mu \cdot \kappa) .$$  \hspace{1cm} (2-21)
where \( \text{SAM}_1 \) and \( \text{SAM}_2 \) are the SAMs of the pixels in \( I \) to the two target vectors.

Training samples of the selected foreground objects and background objects are given, and the spectral signatures, which are means of the two types of samples, are calculated to feed both classifiers. The strategy of iteratively updating the two spectral signatures is chosen not to be used. The initial SAM classification results were obtained and are shown in Figure 2.9 (b). Then these initial classification results were used to set the initial contours for the CV classifier. The boundaries of the foreground segments were used to initialize a signed distance function to substantiate the level set function matrix \( \phi \). The signs of the elements corresponding to pixels belonging to the foreground class are set as negative (inside the contours) and the signs of other pixels are set as positive (outside the contours). Then \( v = 0 \) is set and the contour evolution is ready to begin. After 30 iterations no sign changes appeared in \( \phi \). This indicates the classification results of the CV classifier are identical to the original classification results. The reason is that when \( v = 0 \), for the elements inside the contours, which were initially negative, the update of \( \phi \) calculated by Equation (2-21) will always be negative because \( \text{SAM}_1 < \text{SAM}_2 \); the elements outside the contours that were positive will always receive positive updates because \( \text{SAM}_1 > \text{SAM}_2 \). So no sign changes take place during the process of contour evolution, and the results of the ECV-SAM classifier will be the same as the initial SAM classification results. When \( v = 1.2 \), the curvature term takes effect and the CV-SAM classifier generates the results that are less noisy and have smoother segment boundaries, as shown in Figure 2.9 (e).
Figure 2.9. Comparison of the ECV-SAM classifier to the conventional SAM classifier.

(a) False color image of DC dataset composed of bands 2, 3, 12 of the DR bands, (b) classification results of the SAM classifier, (c) ECV-SAM classification results when $v = 0$, and (d) classification results of ECV-SAM classifier when $\mu = 1.2$. 
In the following example, another commonly used distance metric, Euclidean distance (ED), is used in the RGAC model. The corresponding ECV-ED model is

\[
\frac{\partial \phi}{\partial t} = \delta_x (\phi) \left[ R \left( \sqrt{\sum_{k=1}^{N} (I_k - c_{k1})^2} - \sqrt{\sum_{k=1}^{N} (I_k - c_{k2})^2} \right) + \mu \cdot \kappa \right]
\]

(2-22)

where \( I_k \) denotes the pixel value on the \( k^{th} \) band of a multiple-band image. \( c_{k1} \) and \( c_{k2} \) are two constants that are the means of \( I_k \) inside and outside the contours, respectively, and \( R \) is the following normalization function.

\[
R(x) = \frac{x}{\max|x|}
\]

(2-23)

Given the same training samples, the initial classification results (Figure 2.10 (a)) of the minimum-distance classifier were obtained and used to initialize the contours for ECV-ED. Figure 2.10 (b) shows the results of the ECV-ED classifier after 35 iterations. Compared to Figure 2.9 (b) and (e), it can be seen that for both the conventional pixel-based classifiers and the CV classifiers, the classification results based on Euclidean distance are quite different from those based on SAM. The results of the ECV-SAM classifier are the refined results of the SAM classifier, and similarly, the results of the ECV-ED classifier are the refined results of the minimum-distance classifier.

The two examples above validate the two characteristics of the RGAC models. Namely, the trade-off between the two terms in the RGAC models refines the classification results and makes the RGAC classifiers as spectral-spatial classifiers. As for the CV classifiers, their classification results are fundamentally determined by the selected distance measurement embedded in the image term.
Classification results of the minimum-distance classifier, and (b) classification results of the ECV-ED classifier when $\mu = 1.2$.

2.3.4 Summary

As described in Sections 2.3.1 and 2.3.2, the current applications of RGAC for remote sensing image classifications and target extractions all utilize the traditional GAC framework that heavily relies on contour evolutions. The users are usually required to designate the initial contours, and the process of contour evolutions is time consuming. However, the discussions and experiments in Section 2.3.2 reveal that for the RGAC models, the key advantages are the abilities to control the smoothness of segment boundaries and suppress noise. This leads to a straightforward approach to RGAC-based image classifications, which is to obtain initial contours using pixel-based classifiers and then use RGAC models to refine the initial results. This approach has been demonstrated
in the examples shown in Figure 2.9 and Figure 2.10. It fully utilizes the key advantages of the RGAC models and can quickly reach a stable state. In the next chapter this approach will be generalized as a two-stage classifier strategy for RGAC classifiers.
CHAPTER 3

THE RGAC CLASSIFIERS FOR HYPERSPECTRAL IMAGES

This chapter presents a methodology for the RGAC classifier as it relates to classification of remote sensing images. In terms of classification, it refers to the supervised classifications in which the prior information is provided in the format of training samples. An extended Chan-Vese model is proposed to incorporate different distance metrics, and an edge-oriented RGAC model with more flexibility to be combined with pixel-based classifiers is also proposed. A classification framework has been developed for RGAC classifiers based on a multiphase RGAC approach revised from the MLS method described in Section 2.3.1.

3.1 Introduction

Methods that will be commonly used in conjunction with different RGAC models are introduced in this section. They include a two-stage image classification strategy for RGAC classifiers, a general RGAC model and the concept of the image force map that can be used to predict the performance of the refinement capacities of the RGAC models.
3.1.1 Two-stage Strategy for RGAC Classifiers

A method of contour initialization has been demonstrated in experiments in Section 2.3.3. A two-stage classification strategy is used here for RGAC-based classifications of remote sensing images. The comparison of this strategy with the traditional RGAC classification procedure is illustrated as follows.

Figure 3.1. Procedure of the traditional RGAC classification using the Chan-Vese model. (a) Initial contours are placed inside the target object, (b) an intermediate state of contour evolution occurs, and (c) final stable state of the contour evolution. The segment inside the contour is the foreground object.

For the RGAC-based classifications, the original information is provided in the form of initial contours. In the traditional RGAC classification procedure, the contours would usually begin as a rectangle or circle surrounding the targets (Figure 3.1 (a)), or as a series of circles evenly distributed throughout the image. The contours would evolve gradually as the foreground and background spectral statistics are updated (Figure 3.1 (b)) until they reach a stable state (Figure 3.1 (c)). Here the spectral statistics refer to the...
1\textsuperscript{st}-order and/or the 2\textsuperscript{nd}-order statistics of each class. The target values and target vectors are the 1\textsuperscript{st}-order statistics and can be considered to be the spectral signature of a class.

![Diagram](image)

Figure 3.2. Procedure of the two-stage RGAC classification strategy employing the Chan-Vese model. (a) Training samples selected from the image, (b) initial pixel-based classification results using the spectral statistics derived from the training samples, (c) initial contours obtained from the initial classification results, and (d) final stable state of the contour evolution that refines the initial classification results.

An initial classification is performed during the first stage of the two-stage strategy by a conventional pixel-based classifier given the spectral statistics (Figure 3.2 (a) and Figure 3.2 (b)). Then the contours are initialized using the initial classification results (Figure 3.2 (c)). In the second stage, these initial results will be refined as
illustrated in Figure 3.2 (d). If needed, the expectation–maximization (EM) strategy can be used to improve the estimates of the spectral statistics in the first stage.

There are two major differences to be noted in the two-stage strategy (Figure 3.2) when compared to the traditional RGAC classification method (Figure 3.1). First, the initial contours are obtained from the initial classification. Before the contour evolutions begin, each pixel is attributed to a class with associated spectral statistics. Second, the self-adaptation of the spectral statistics in the RGAC models is always performed in the traditional procedure, while in the two-stage strategy it is replaced by the EM strategy that is optional.

These changes are consistent with the characteristics of the RGAC models discussed in Section 2.3.3, and they have two major advantages. To begin with, the contour evolutions can quickly reach a stable state without a large number of iterations because the refined classification results are very consistent with the initial results. Secondly, this two-stage strategy enables the RGAC method to be combined with other pixel-based classifiers to elevate them to the advanced level of spectral-spatial classifiers, while not changing the framework of these classifiers since refinements of the RGAC models occur in the second stage, which does not require any user interactions.

3.1.2 A General RGAC Model

The following equation is meant to represent the traditional RGAC model as well as the CV models and the ones yet to be discovered.

\[
\frac{\partial \phi}{\partial t} = \delta_c(\phi)(IMG + \mu \cdot k)
\]  

(3-1)
where $IMG$ denotes the image term that can take different forms. For example, $IMG = (I - c_1)^2 - (I - c_2)^2$ for the Chan-Vese Model (Equation (2.11)) and $IMG = SAM_1 - SAM_2$ for the ECV-SAM model (Equation (2.21)). The range of $IMG$ is $[-1, 1]$, and so is the curvature $\kappa$ as defined by Equation (2.3).

With the contour initialization method described in the previous section, the sign of $IMG$ for every pixel is known, and specifically, $IMG$ is negative for the pixels inside the contours and positive for those outside the contours. Since the signs of $IMG$ are consistent with the signs of the level set matrix $\phi$, applying $IMG$ to $\phi$ does not change the signs of the elements in the level set matrix $\phi$ and thus will not change the contours that are 0-level set of $\phi$. The absolute of $IMG$, $|IMG|$, is referred to as the image force, which reflects the effort to keep the contours unchanged. If the curvature term at a pixel position has a different sign from the image term and also has a larger absolute value, then the update of the level set function, which is the sum of the two terms, will possibly change the sign of the corresponding element in $\phi$, and correspondingly, change the classification label of the pixel.

The refinement functions of RGAC models, which include smoothness control of segment boundaries and noise suppression, take effect where the curvature term dictates the trade-off of two terms in Equation (3.1). The curvature term has a large absolute value on irregular boundaries including small isolated segments, so it smoothes the segment boundaries and suppresses the salt-and-pepper noise in the classification results. Note that if the noise is caused by abnormal sensor response such as the impulse noise (Schowengerdt, 1983), the corresponding noisy pixels will have outlier spectra that are
thought to result in a weak image force, which further assists the curvature term to dictate the trade-off on these pixels and suppress noise.

In the ideal case that the intensities of the foreground and background pixels are homogenous without variations such as the image in Figure 3.1 (a), the image force (e.g., \( |IMG| = |I - c_1|^2 - (I - c_2)^2 | \) for the Chan-Vese Model) will be the same for every pixel. Then the refinement functions only affect the curvature term and the trade-off can be simply adjusted by the coefficient \( \nu \). However, when RGAC is applied to real remote sensing images, an essential factor must be taken into account, which is spectral mixture. This phenomenon exists in all remote sensing images. 'Mixed pixels result when a sensor’s IFOV includes more than one land cover type of feature on the ground' (Lillesand et al., 2008). Spectral mixture appears on all object boundaries in images including the small-sized objects that can be misinterpreted as noise in pixel-based classification results. For example, in a remote sensing image with 4 meters resolution, vehicles on roads can be classified as "non-road objects" and appear as salt-and-pepper noise in the classification results. The pixels representing the vehicles necessarily contain spectral mixture due to their small sizes with respect to the spatial resolution of the image.

A mixed spectrum is the blending of pure spectral signatures and causes deviations from pure spectral signatures (Lillesand et al., 2008). Mixed pixels should have a weak image force that tends to be overwhelmed by the curvature term. For example, spectral mixture creates a situation where pixel spectra are close to being equal-distant to multiple spectral signatures, and then, when the Chan-Vese model is used, the
image force, which is $\|IMG\| = \left| (I - c_1)^2 - (I - c_2)^2 \right|$, will be small. So the spectral mixture reflected in the image term also assists the curvature term to take domination of the trade-off.

Generally speaking, after each pixel is assigned to its nearest class in the first stage of the two-stage classification strategy, the RGAC models should have a strong image force on pixels well attached to their associated classes, according to certain criteria and weak image force on pixels that are loosely attached. More specifically, weak image force is associated with noisy pixels and mixed pixels whose spectra deviate from the spectral signatures of their associated classes. As will be presented in the following sections, how the image force is represented in the image term is an important factor that affects the performance of the refinement functions of the RGAC models and the RGAC-based classification results.

### 3.1.3 Image Force Map

Once the spectral statistics of each class are known, $IMG$ can be calculated for every pixel, and the calculated image force $|IMG|$ for the whole image domain creates a map referred to as the image force map. If the spectral statistics are not updated (the EM strategy is not used), the image force map will remain unchanged during the contour evolution and thus they can be obtained before the contour evolution begins. As mentioned in the previous section, in the ideal case the image force will be the same throughout the entire image domain. But in practice this is always complicated by the existence of spectral mixture, noise, and various types of objects. The image force map is
an important reference tool to indicate how the refinement functions of the RGAC models will take effect.

Figure 3.3. Illustration of image force map on simulated image. (a) A simulated grayscale image overlaid with the initial contour, (b) image force map of the Chan-Vese model, (c) stable state of contour evolution, and (d) classification results corresponding to (c).

Figure 3.3 (a) shows a simulated grayscale image that contains an object (overlaid with the initial contour) on a dark background. The image was rescaled to the range of [0, 1]. The right half of the object has a constant intensity, and the left half becomes gradually darker. $c_1$ and $c_2$ are given as the intensity values of the right half of the object and the background, respectively. Initial classification was performed and the initial contour bounding the object was obtained from the initial classification results, as shown in Figure 3.3 (a). When the Chan-Vese model was taken as an example,
\[ |IMG| = \left| (I - c_1)^2 - (I - c_2)^2 \right| \] was calculated on the entire image, and the image force map in the range of \([0, 1]\) was obtained as shown in Figure 3.3 (b).

The dark areas in the image force map of the Chan-Vese model indicate weak image force. They are the areas where the refinement functions tend to take effect because in the evolution function, the image term in those areas tend to be overwhelmed by the curvature term. As can be seen in the final stable state of the contour evolution (Figure 3.3 (c)) and the corresponding classification results (Figure 3.3 (d)), the left side of the object corresponding to the dark areas in the image force map is eroded more than the right side.

### 3.2 Different Types of RGAC models for Remote Sensing Image Classifications

In the Chan-Vese model and its variations referred to as CV models, the image term is defined as the difference of the distances of a pixel’s spectrum to two spectral signatures (target values or target vectors). In this dissertation an extended Chan-Vese model was proposed to incorporate different distance metrics. With the two-stage classification strategy, an edge-oriented RGAC model was proposed that does not require the utilization of a certain distance metric and as a result can be more freely combined with other classifiers.
3.2.1 Extended Chan-Vese Model

For an image $I$ with dimensions of $w$ (image width) $\times$ $h$ (image height) $\times$ $N$ (number of bands), the image term of the extended Chan-Vese models (ECV) in the general RGAC model of Equation (3-1) is expressed as follows:

$$IMG = R(D_1 - D_2)$$

(3-2)

where $D_1$ and $D_2$ denote two $w \times h$ matrixes storing distances of each pixel to the spectral signatures of the foreground and background segments, respectively, and $R$ is a virtual function to normalize the image term to $[-1, 0]$ and is implemented when needed by Equation (2-23). Note that if Euclidean distance is selected as the distance metric, then $\max(D_1 - D_2)$ in Equation (2-23) is the Euclidean distance between the two target vectors. Weak image force becomes apparent on pixels including mixed pixels whose spectra are close to being equal-distant to the two spectral signatures according to the selected distance metric.

As shown in Section 2.3.3, the distance metrics commonly used in the classification of remote sensing images can be incorporated into the RGAC models such as maximum likelihood (Equation (2-18)), SAM (Equation (2-21)) and Euclidean distance (Equation (2-22)). The ECV models embedding the above three distance metrics are referred to as ECV-ML, ECV-SAM and ECV-ED, respectively. These distance metrics are functions of image intensities and the 1st-order and 2nd order statistics of a class, and reflect the distance (similarity) of a pixel’s spectrum to a spectral signature. For example, if Euclidean distance is selected as the distance metric, then $D_1 = \sqrt{\sum_{k=1}^{N}(I_k - c_{k1})^2}$.
and $D_2 = \sqrt{\sum_{k=1}^{N}(I_k - c_{k2})^2}$, and Equation (3-1) becomes Equation (2-22). Other distance metrics can be adopted for use with the ECV models as well.

Figure 3.4. Example of a binary classification on a single-band remote sensing image using the ECV-ED model. (a) An image patch cut from band 27 of the DC HYDICE HSI dataset. The left figure is a false color image of the dataset composed of bands 60, 27 and 17. The right figure is a $57 \times 83$ image patch cut from band 27, (b) initial classification results, (c) image force map of the Chan-Vese model in range [0, 1], and (d) final classification results refined by the Chan-Vese model.
Figure 3.4 shows an example of the binary classification using the ECV-ED model on a real remote sensing image. Figure 3.4 (a) shows an image patch cut from a band of a HSI dataset. Initial classification was first conducted using the mean intensity of the image patch as the threshold, and the results are shown in Figure 3.4 (b). $c_1$ and $c_2$ were updated with the classification results. By using the equation 

$$|IMG| = \frac{|I - c_1| - |I - c_2|}{|c_1 - c_2|}$$

on each pixel, the image force map of the Chan-Vese model is obtained as shown in Figure 3.4 (c). Figure 3.4 (d) is the final refined classification.
results with smoothed boundaries and reduced noise.

Note the two road segments in the lower part of the image patch as marked in Figure 3.4 (b) are excluded from the foreground segments in the refined results, and the corresponding weak image force was observed on these segments in the image force map. A sample pixel at position (56, 48) as marked in Figure 3.4 (a) was taken to examine the cause of the weak image force. The intensity of that pixel was 5,905, and the means of the foreground and background segments were 7,367 and 3,213, respectively. So the image force at that position was calculated as follows:

\[ IMG(56,48) = \frac{\|I - c_1\| - \|I - c_2\|}{\|c_1 - c_2\|} = \frac{\|5905 - 7367\| - \|5905 - 3213\|}{\|7367 - 3213\|} = \frac{1462 - 2692}{4154} = \frac{1230}{4154} = 0.30. \]

The sample pixel's intensity (5,905) is considered to be close to the mean intensity (7,367) of that of the foreground pixels since the distance (1,492) is smaller than the standard deviation of the pixel intensities of the foreground pixels, which is 1,795. So the pixel should not be excluded from the foreground segments and its weak image force is inappropriate. Further analysis on the cause of this type of inappropriate weak image force is given below.

As mentioned in Section 3.1.2, the strength of the image force in RGAC models should reflect how well a pixel is attached to its associated class, and weak image force should appear on noisy pixels and mixed pixels. But these cannot be well realized in the ECV models because the image force does not reflect how strongly a point is attached to its associated class center. The following example illustrates this problem using synthetic data.
Figure 3.5. Illustration of the image force of the ECV-ED model. $D_{\text{max}}$ is the distance between the two class centers, and the distances ($D_1$ and $D_2$) of a randomly selected point to the two class centers are also illustrated.

Two synthetic data clusters were generated, made up of 100 points each, following a Gaussian distribution. One is centered by (2, 5) with the covariance matrix of (0.8, 0; 0, 0.8), and the other is centered by (7, 6) with the covariance matrix of (0.5, 0; 0, 0.5). As shown in Figure 3.5, a separation line divides the synthetic data into two classes based on their Euclidean distances, $D_1$ and $D_2$, of every point to the two cluster centers. The distance between the two cluster centers is the maximum of $|D_1-D_2|$, and is denoted by $D_{\text{max}}$. Assume these points are the pixels from two image bands displayed in 2-D spectral domain to form a 2-D scatter diagram. Meanwhile, for the ECV-ED model, there
exists a value of the image force, which is \( \frac{|D_1 - D_2|}{D_{\text{max}}} \), on every position in 2-D space. In order to illustrate the distribution of the image force strength in the spectral domain, a rectangle with the diagonal of (0, 0) and (8.5, 8.5) in the 2-D spectral space shown in Figure 3.5 is evenly sampled and the image force was calculated on every sampling position. The values of the image force are visualized in Figure 3.6 with a color scheme. Contours representing the image force values are overlaid on it to show the pattern of the strength distribution.

![Image](image.jpg)

Figure 3.6. Image force strength distribution map with overlaid contours in 2-D spectral space shown in Figure 3.5. Image force is calculated on the samples in range (0, 0) – (8.5, 8.5) which is equally sampled in \( x \) and \( y \) directions with the interval of 0.1.

Several inappropriate aspects can be observed in the image force distribution map
shown in Figure 3.6. First, as shown in Figure 3.6, imagine a circle centered by one cluster center (2, 5), with a radius of 0.5. The points on this circle are equal-distant to (2, 5) and are equivalently attached to it according to the probability of Gaussian distribution. However, the image force on these points is significantly different. Second, if there are noisy pixels distributed in the bright areas of the strength image force distribution map, they will have strong image force and can hardly be filtered out. Third, as can be seen in Figure 3.6, the pixels that fall in between the two cluster centers are likely to have a weak image force, but this does not mean that they are necessarily noisy pixels or mixed pixels. That explains why the sample pixel at position (56, 48) in the image patch in Figure 3.4 (a) has a weak image force because its intensity (5,905) is between the means (7,367 and 3,213)) of the two classes; therefore, this pixel, together with other pixels in the two road segments marked in Figure 3.4 (b), have been inappropriately excluded out of the refined classification results as shown in Figure 3.4 (d). The problems above can also be expected in the ECV models using distance measurements rather than Euclidean distance.

3.2.2 Edge-Oriented RGAC Model

A major limitation of the ECV models is that they require the usage of a distance metric, and therefore can only refine the classification results of the pixel-based classifiers based on certain distance metrics. For some other classifiers, such as support vector machines (SVM) and k-nearest-neighbor (KNN) that are not based on a distance metric, they cannot be combined with the ECV models. Another limitation is that because
the definitions of the image term are based on the distances of pixels’ spectrum to the spectral signatures of defined classes, the values of the image term are highly dependent on the designations of classes and the corresponding spectral statistics. The ECV classifiers can suffer from inaccurate estimates of the spectral statistics used in these distance metrics due to the curse of dimensionality in HSI data.

A new type of RGAC model is proposed in this section whose image term is independent on the defined classes and their spectral statistics. As mentioned in Section 3.1.2, the noisy pixels and mixed pixels found in remote sensing images should be reflected in the image term of the RGAC models and have a weak image force (absolute of the image term). For the noisy pixels and mixed pixels, as their spectra are different from the spectra of their surrounding pixels, they can be represented by measuring the spectral similarities to the surrounding pixels. Since the measured similarity can be considered to reflect the edge information of an image, this model is referred to as the edge-oriented RGAC model (EO).

Measuring the spectral similarities of a pixel to its neighboring pixels is a key part of the region-growing image segmentation algorithms, and this similarity can be endowed with different meanings under different scenarios. For example, in the biologically-inspired image segmentation method of Locally Excitatory Globally Inhibitory Oscillator Networks (LEGION) (Terman and Wang, 1995; Wang, 2005), each pixel is taken as an oscillator, and the similarity between adjacent pixels represents the stimulation that an active oscillator can pass on to its neighboring oscillators.
SAM is selected as the measurement of the spectral similarities among adjacent pixels in multiple-band images because of its unique properties of invariance to multiplicative scaling, nonadditivity, and nonmonotonicity compared with Euclidean distance (Keshava, 2004). Then the image force in the proposed EO model is represented by the mean of the SAMs of a pixel with its eight neighboring pixels. The image force in the image domain $\Omega$ is calculated by

$$IMG(i, j) = \frac{1}{N} \sum_{n=1}^{N} |\text{SAM}(\vec{I}(i, j), \vec{I}(i, j)_n)|, (i, j) \in \Omega$$

(3-3)

where $\vec{I}(i, j)$ is the pixel vector at position $(i, j)$ and $\vec{I}(i, j)_n$ is the pixel vector of its $n^{th}$ neighboring pixel ($N=8, 5$ or 3 depending on the position $(i, j)$).

Because SAM does not apply to 1-D data, if the similarity is measured on a single band, the image force must be calculated in other ways. General edge detectors including those used in EGAC models, as seen in Equation (2-8) can all be considered. Because of the 16-bit color depth of the study image, when the edge detectors similar to Equation (2-8) are used, a large portion of the data will have the output crowded into a small range close to 0 in range $[0, 1]$. For this reason the image force was calculated in the following way in this case study.

$$IMG = e^{-\frac{\nabla I}{2\sigma}}$$

(3-4)

where $\nabla I$ is the gradient of $I$ and $\sigma$ is the standard deviation of $\nabla I$.

Figure 3.7 (a) shows the image force map calculated with Equations (3-4) on the image patch seen in Figure 3.4 (a). With the initial classification results of Figure 3.4 (b), the refined results using the EO model are shown in Figure 3.7 (b). Because the EO
model is not dependant on the designations of classes and the selections of distance metrics, the performance of its refinement functions is more much robust, and it has much more flexibility to be combined with other pixel-based classifiers including unsupervised classifiers. Further experiments with this model will be described in the next chapter.

![Image](image.png)

(a)                                                            (b)

Figure 3.7. Example of binary classification using the EO model on the single-band image patch shown in Figure 3.4 (a). (a) Image force map calculated with Equation (3-6), and (b) final classification results refined by the EO model.

3.3 Classification Framework of RGAC Classifiers

As described in Section 2.3.2, several multiphase approaches have been suggested for RGAC classifiers. The Multiple Level Set (MLS) method proposed by Mansouri et al. (2006) has been adopted in the research of the dissertation with revisions to reduce the complexity of model implementation.
3.3.1 Workflow of the Multiphase Approach for RGAC Models

The two-stage classification strategy, described in Section 3.1.1, makes it possible to revise the implementations of the MLS method in order to simplify it for the study; for example, only one level set function is used to represent the multiple classes. This multiphase approach, associated with the two-stage classification strategy, can combine the RGAC models with pixel-based classifiers. Initial classification is conducted with a pixel-based classifier, and the results are refined by the selected multiphase RGAC model.

Figure 3.8 shows the workflow of the classification procedure using the multiphase approach on a dimensionality-reduced HSI dataset with the dimensions of \(w\) (image width) \(\times h\) (image height) \(\times d\) (number of DR bands). A number of \(m\) classes on the image are defined and the estimated spectral statistics of each class, including the spectral signatures, are calculated from the provided training samples.

First, the initial classification using a pixel-based classifier is conducted. If the applied RGAC model is an ECV model, the pixel-based classifier is restricted to the classifier based on a certain distance metric, such as Euclidean distance, SAM, Mahalanobis distance or the likelihood function, and the classification procedure is as follows. The distances of the pixels to the spectral signature of each class are calculated and form a \(w \times h \times m\) matrix \(D\). \(D = \{D_1, D_2, \ldots D_m\}\) where \(D_i\) is the \(w \times h\) matrix of the distances of the pixels in class \(i\). For this reason, the classification results are represented by a \(w \times h\) matrix \(CLS\) in which every element is labeled according to the minimum distance at each pixel position. The label \(c\) at position \((i, j)\) in \(CLS\) is determined by
\[ D(i, j, c) = D(i, j) = \min(D(i, j, 1), D(i, j, 2), \ldots, D(i, j, m)) \]  \hspace{1cm} (3-5)

If the selected RGAC model is an EO model, initial classification can be made with any type of pixel-based classifiers.

---

Figure 3.8. Workflow of the RGAC classifiers.
After the initial classification, for each class \( c \) there is a domain \( CLS_c \) defined by 
\[ CLS_c = \{(i, j) | CLS_y = c \}. \]
Let \( \phi \) denote the level set matrix \((w \times h)\). In the initialization of \( \phi \), a temporary level set matrix \( \phi_c \) is obtained for each domain \( CLS_c \) and the corresponding elements in \( \phi \) are initialized as 
\[ \phi(i, j) = \phi_c(i, j), (i, j) \in CLS_c \]  
(3-6)

Because the elements in \( \phi_c \) corresponding to the domain \( CLS_c \) are negative, all elements in \( \phi \) are negative. In the contour evolution of the second stage, a temporary level set matrix \( \phi_c \) is also formed as follows:
\[ \phi_c(i, j) = \begin{cases} \phi(i, j), (i, j) \in CLS_c \\ -\phi(i, j), (i, j) \notin CLS_c \end{cases} \]  
(3-7)

Then the update of the level set matrix in the domain \( CLS_c \) is calculated by
\[ \frac{\partial \phi_c(i, j)}{\partial t} = \delta (\phi_c(i, j)) \left[ IMG(i, j) + \nabla \cdot \text{div} \frac{\nabla \phi_c(i, j)}{\nabla \phi_c(i, j)} \right] (i, j) \in CLS_c. \]  
(3-8)

For the ECV models, \( D_c \) in Equation (3-8) is the \( c^{th} \) band of the matrix \( D \) and is a \( w \times h \) matrix storing the distances of pixels to the class \( c \). \( D_c \) is a \( w \times h \) matrix storing the second smallest distances of the pixels in class \( c \) and is obtained by
\[ D_c(i, j) = \min \{ D(i, j, c') | c' \in [1, m], c' \neq c, ij \in CLS_c \}. \]  
(3-9)

Since \( D_c \) stores the smallest distance of each pixel, \( D_c(i, j) - D_c(i, j) < 0 \) is always true for \((i, j) \in CLS_c \). So the image term is always negative in Equation (3-8) for the ECV models.
Similarly, for the RGAC-CO model and the EO model, the image term in Equation (3-8) is also always negative according to the definitions of their image terms.

Let $IMG_c$ and $CVT_c$ denote the image term and curvature term in Equation (3-8), respectively. For example,

$$IMG_c = \{IMG(i, j), (i, j) \in CLS_c\}$$  \hspace{1cm} (3-10)

$$CVT_c = \left\{ \delta_c(\phi_c(i, j)) \left[ \mu \cdot \text{div} \frac{\nabla \phi_c(i, j)}{\nabla \phi_c(i, j)} \right], (i, j) \in CLS_c \right\}.$$  \hspace{1cm} (3-11)

At the same time, the curvature term calculated with $\phi_c$ for the entire domain is stored in a $w \times h \times m$ matrix $CVT_m$, i.e.,

$$CVT_m(i, j, c) = CVT_m(i, j) = \delta_c(\phi_c(i, j)) \left[ \mu \cdot \text{div} \frac{\nabla \phi_c(i, j)}{\nabla \phi_c(i, j)} \right], (i, j) \in \Omega.$$  \hspace{1cm} (3-12)

After all classes are traversed, an overall image term matrix $IMG$ and an overall curvature term matrix $CVT$ can be obtained to derive the overall update $U$ as follows:

$$IMG = IMG_1 \cup IMG_2 \cup \ldots \cup IMG_m$$  \hspace{1cm} (3-13)

$$CVT = CVT_1 \cup CVT_2 \cup \ldots \cup CVT_m$$  \hspace{1cm} (3-14)

$$U = IMG + CVT$$  \hspace{1cm} (3-15)

$U$ must be rescaled to maintain a steady set function, and then Equation (3-15) becomes

$$U = \alpha \frac{IMG + CVT}{\max |IMG + CVT|}$$  \hspace{1cm} (3-16)
The scale coefficient $\alpha$ is always set as 0.49 in the experiments in this dissertation, which means the values of $U$ are always within $[-0.49, 0.49]$. Then the following temporary level set matrix $\phi'$ is obtained.

$$\phi' = \phi + U$$  \hspace{1cm} (3-17)

In the level set theory, if the sign of an element in the level set matrix changes, that means that element’s spatial relationship with respect to the contours is also changed, by either being included into the contours from outside (changing from positive to negative) or being excluded out of the contours from inside (changing from negative to positive). As $\phi$ is always negative, if the sign of an element in $\phi'$ changes from negative to positive, its correspondence in $CLS$ should be relabeled, and the updated level set matrix $\phi_{updated}$ is:

$$\phi_{updated}(i, j) = \begin{cases} 
\phi'(i, j) \geq 0, & (i, j) \in \Omega, \\
-\phi'(i, j) < 0, & (i, j) \in \Omega.
\end{cases}$$  \hspace{1cm} (3-18)

### 3.3.2 Relabeling of the Suppressed Pixels

As noted above, the update of $\phi$ is the trade-off between the image term and curvature term. As the image term $IMG$ is always negative, when an element $(i, j)$ changes from negative to positive, it indicates the curvature term at that position must be positive and larger than the sum of the absolutes of the corresponding image term and the element in the current level set matrix $\phi$, i.e., $CVT(i, j) > |IMG(i, j)| + |\phi(i, j)|$. As a result,
that element in $CLS$ needs to be relabeled. An example is given below to illustrate how the new label is determined using the ECV-ED model.

Figure 3.9. Illustration of the relabeling of the suppressed pixels. (a) A gray scale image with three classes, and (b) the initial minimum-distance classification results.

Figure 3.9 (a) is a $10 \times 10$ grayscale image with three classes, $A$, $B$ and $C$. Trainings sample were provided to obtain the initial classification results shown in Figure 3.9 (b), where each color represents a particular class. The temporary level set matrixes were formed with Equation (3-7) for each class, and the corresponding image term and curvature term were calculated with Equations (3-10) and (3-11). Then the overall image term $IMG$ and overall curvature term $CVT$ were obtained using Equation (3-13) and Equation (3-14), respectively.
Figure 3.10. The overall image term matrix and curvature term matrix in the 1st iteration overlaid on the initial classification results. (a) The overall image term matrix. (b) The overall curvature term matrix.

Figure 3.10 (a) and Figure 3.10 (b) show the non-zero elements in \(IMG\) and \(CVT\) overlaid on the matrix of the initial classification results, respectively. The negative values in \(IMG\) (Figure 3.10 (a)) reflect the force that attempts to keep the current contours unchanged, and the positive values in \(CVT\) reflect the force that attempts to pull the contours back. For example, the element in position (8, 2) is a convex class \(B\). The corresponding values in \(IMG\) and \(CVT\) were -0.94 and 1.0, respectively. The corresponding update calculated by Equation (3-16) was 0.03 which means the update at (8, 2) was attempting to pull the convex back, resulting in a smoother boundary of class
B. This change was not to happen in this iteration because $\phi$ was initialized as $\phi = -0.5$, and the updated value calculated by Equation (3-17) was -0.47, which is still negative. In the next iterations, its curvature term would remain unchanged at -0.94, while the curvature term would change due to the adjustments in $\phi$. Table 3.1 shows the values of $IMG, CVT, \phi, U, \phi'$ and $\phi_{updated}$ in the first three iterations.

From Table 3-1, it can be seen that after the 3rd iteration, the value in the level set matrix $\phi$ at position (8, 2) changed from negative to positive, which indicated the corresponding element in $CLS$ should be relabeled. The $w \times h \times m$ matrix $CVT_m$ that stores the curvature term of each class for the entire image domain $\Omega$ was used to determine the new label. Table 3-2 lists the three values in $CVT_m$ at position (8, 2), the curvature values of the three classes at that position.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$IMG$</th>
<th>$CVT$</th>
<th>$\phi$</th>
<th>$U$</th>
<th>$\phi'$</th>
<th>$\phi_{updated}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.94</td>
<td>1.00</td>
<td>-0.50</td>
<td>0.03</td>
<td>-0.47</td>
<td>-0.47</td>
</tr>
<tr>
<td>2</td>
<td>-0.94</td>
<td>1.31</td>
<td>-0.47</td>
<td>0.18</td>
<td>-0.29</td>
<td>-0.29</td>
</tr>
<tr>
<td>3</td>
<td>-0.94</td>
<td>1.44</td>
<td>-0.29</td>
<td>0.39</td>
<td>0.10</td>
<td>-0.10</td>
</tr>
</tbody>
</table>

Table 3.2. Curvature values at position (8, 2) in Figure 3.2 (a) after three iterations.

<table>
<thead>
<tr>
<th>Class</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>curvature</td>
<td>0.76</td>
<td>1.44</td>
<td>-1.41</td>
</tr>
</tbody>
</table>
Contrary to positive curvature values that pull the convex contours back, the negative curvature values reflect the force that attempts to push the contours to fill in the concave spaces. In the initial classification results shown in Figure 3.2 (b), the element (8, 2) is a convex for class $B$, while it is also a concave for class $C$. So at this position when the positive curvature value of 1.44 was derived for class $B$, the negative curvature value of -1.41 was obtained for class $C$. For class $A$, the element (8, 2) can be considered to be a convex of it in an 8-connection graph which explains why the positive curvature value of 0.76 is obtained for class $A$.

The rule of element relabeling is as follows. For an element $(i, j)$ in $CLS$ whose corresponding element in $\phi'$ calculated in Equation (3-17) is positive, its new label $c_1$ is the class whose curvature is at the lowest position.

$$C_{c_1}(i, j) = \min[C_1(i, j), C_2(i, j), \ldots, C_m(i, j)]$$  \hspace{1cm} (3-19)

![Figure 3.11](image)

Figure 3.11. The final RGAC classification results. (a) Classification results after the $3^{rd}$ iteration. (b) Classification results after the $4^{th}$ iteration (stable state).
From Equation (3-19), \( c_i = 3 \) was obtained because
\[
C_3(8,2) = \min\{C_1(8,2), C_2(8,2), C_3(8,2)\} = \min[0.76, 1.44, -1.41].
\]
So CLS(8, 2) is relabeled as 3, which is class C. Also in the 3\(^{\text{rd}}\) iteration, the element at position (4, 6) in \( \phi \) was 0.26, and the corresponding curvatures of the three classes were -1.57, 1.57 and 1.11, respectively. So CLS(4, 6) was relabeled as 1. Figure 3.11 (a) shows the classification results after relabeling. Figure 3.5 (b) shows the classification results after the 4\(^{\text{th}}\) iteration, which were the final results because no positive elements appeared in \( \phi \) in further iterations and the contour evolution is considered to have reached a stable state.
CHAPTER 4

EXPERIMENTS AND DISCUSSIONS

After the nonlinear dimensionality reduction of the two HSI datasets from LE, two types of RGAC classifiers that included three extended Chan-Vese (ECV) classifiers (ECV-ED, ECV-SAM, and ECV-ML) and four edge-oriented (EO) classifiers (EO-ED, EO-SAM, EO-ML, and EO-SVM) were tested using the dimensionality-reduced data. Classification experiments were also conducted to evaluate the performance of these RGAC classifiers.

4.1. HSI Datasets and Nonlinear Dimensionality Reduction with LSH-based Laplacian Eigenmaps

4.1.1 Description of the Datasets

Two HSI datasets, both taken by HYDICE, were used in the experiments. HYDICE collects data of 210 bands over the range of 399-2,499 nanometers with a spatial resolution of 1 to 4 meters depending on the aircraft altitude and ground speed.
Figure 4.1. RGB image of the *Urban* dataset and the ground-truth. (a) RGB image composed of bands 49, 36 and 18, and (b) manually selected ground-truth of 10 classes overlaid on the RGB image.
Figure 4.2. False-color image of the DC dataset and the ground-truth. (a) False-color image of the DC dataset composed of bands 60, 27 and 17, and (b) samples of seven classes in the DC dataset overlaid on the RGB image (bands 49, 36, 18).
The first dataset was taken at Copperas Cove, an urban area in Texas, and is referred to as the *Urban* dataset. The spatial resolution of this dataset is approximately 4 meters. It contains 307 scan lines with 307 pixels in each scan line. 162 bands were manually selected by removing the corrupted bands and the bands that contained noise. Figure 4.1 (a) shows the RGB image of the *Urban* dataset, and Figure 4.1 (b) shows the manually selected ground-truth of ten classes within this dataset. That dataset was obtained from (http://www.agc.army.mil/hypercube/).

The second dataset was taken at The Mall in Washington DC, and is referred to as the *DC* dataset. The spatial resolution of this dataset is approximately 3.5 meters. It contains 317 scan lines made up of 307 pixels in each scan line. 191 bands were manually selected by removing the corrupted bands and the noisy bands. Figure 4.2 (a) shows the false-color image of the *DC* dataset. Figure 4.2 (b) shows the manually selected ground-truth of seven classes in this dataset. That dataset was obtained from (https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html).

Of all the experiments described in this chapter, approximately 20% of the ground-truth in each class or a maximum of 500 pixels were used as training samples and the other are test samples to assess the accuracies of the classification results. The numbers of training and test samples in the two datasets are listed in Table 4.1 and Table 4.2.
Table 4.1. Numbers of training and test samples in *Urban* dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>Training samples</th>
<th>Test samples</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>building A</em></td>
<td>489</td>
<td>1904</td>
</tr>
<tr>
<td><em>building B</em></td>
<td>369</td>
<td>1475</td>
</tr>
<tr>
<td><em>concrete</em></td>
<td>180</td>
<td>719</td>
</tr>
<tr>
<td><em>grass</em></td>
<td>468</td>
<td>6544</td>
</tr>
<tr>
<td><em>parking lot</em></td>
<td>458</td>
<td>2288</td>
</tr>
<tr>
<td><em>road A</em></td>
<td>463</td>
<td>2315</td>
</tr>
<tr>
<td><em>road B</em></td>
<td>470</td>
<td>1878</td>
</tr>
<tr>
<td><em>soil</em></td>
<td>464</td>
<td>1854</td>
</tr>
<tr>
<td><em>tree</em></td>
<td>498</td>
<td>7459</td>
</tr>
<tr>
<td><em>white soil</em></td>
<td>397</td>
<td>1585</td>
</tr>
</tbody>
</table>

Table 4.2. Numbers of training and test samples in *DC* dataset.

<table>
<thead>
<tr>
<th>Class</th>
<th>Training samples</th>
<th>Test samples</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>grass</em></td>
<td>472</td>
<td>3298</td>
</tr>
<tr>
<td><em>road</em></td>
<td>481</td>
<td>2882</td>
</tr>
<tr>
<td><em>shade</em></td>
<td>309</td>
<td>1234</td>
</tr>
<tr>
<td><em>street</em></td>
<td>270</td>
<td>1078</td>
</tr>
<tr>
<td><em>tree</em></td>
<td>493</td>
<td>2460</td>
</tr>
<tr>
<td><em>water</em></td>
<td>338</td>
<td>1349</td>
</tr>
<tr>
<td><em>roof</em></td>
<td>471</td>
<td>6112</td>
</tr>
</tbody>
</table>

4.1.2 Nonlinear Dimensionality Reduction with Laplacian Eigenmaps

Laplacian eigenmaps (LE) is a nonlinear dimensionality reduction method that attempts to preserve the local properties of the embedded manifold in the high-dimensional data through low-dimensional representation (Belkin and Niyogi, 2002). The local properties of the original data are represented based on the pairwise distances between near neighbors. For the Laplacian eigenmaps algorithm, a neighborhood graph is first constructed in which every data point is connected to its $k$ nearest neighbors. For any two points in the neighborhood graph that are connected by an edge, the weight of the
edge is computed using a kernel function that assigns a higher weight to closer points, and the results form a sparse adjacency matrix $W$. With the degree matrix $M$ and the graph Laplacian $L$ of the sparse adjacency matrix $W$, the Laplacian eigenmaps algorithm finds the low-dimensional representation of the original data by solving the following eigenvector problem:

$$L \nu = \lambda M \nu$$

where $\nu$ represents the eigenvectors. The eigenvectors corresponding to the smallest nonzero eigenvalues form the low-dimensional data representation of the original high-dimensional data (details can be found in Belkin and Niyogi, 2002 and Maaten et al., 2009). Compared with other nonlinear dimensionality reduction methods such as LLE and ISOMAP, the advantages of LE are the simplicity of its algorithm and the low computation cost, which is why LE was selected to be part of the research in this dissertation.

The nearest neighbor (NN) problem is defined as: given a set of $n$ points $P = \{p_1, ..., p_n\}$ in a $d$-dimensional space $X$, find the point in $P$ closest to a query point $q \in X$. All nonlinear dimensionality reduction methods require a $k$-nearest neighbor search, whose goal is to find the $k$ nearest neighbors of every point as the basis to represent the manifold. The negative aspect of this method is that it creates a bottleneck of the nonlinear dimensionality reduction methods due to the high computational cost, which is $O(n^2 d)$. Corresponding to NN, the problem of approximate nearest neighbor (ANN) is defined as: finding a point $q \in P$ that is an $\epsilon$-approximate nearest neighbor of the query point $q$ such that for all $p' \in P$, $D(p,q) \leq (1+\epsilon)D(p',q)$ where $\epsilon$ is a small positive constant.
representing the approximation error (Indyk and Motwani, 1998). Locality-sensitive hashing (LSH) is an algorithm of approximate nearest neighbor search, which is able to speed up the nearest neighbor search, particularly in high-dimensional spaces.

The key idea of LSH is to use hash functions to place data points into buckets of different hash tables, such that the probability of collision is higher for points in the same bucket. The LSH algorithm used in the research of this dissertation was adopted from the algorithm described in Georgescu et al. (2003). The approximate k-nearest neighbor search with LSH works as follows. Suppose a d-dimensional HSI dataset has n pixels each corresponding to a d x 1 pixel vector. These pixels are ‘tessellated L times with random partitions, each defined by K inequalities' (Georgescu et al. 2003). First, K pairs of random numbers, dk and vk, are generated L times. dk denotes the band number that is an integer between 1 and d, and vk is a value within the range of the dk -th band. Let xi denote the intensity of a pixel vector x on the i-th band. Each pair of the random numbers, dk and vk, creates the following inequality for this pixel.

\[ x_{dk} \leq v_k \]  

(4-2)

With Equation (4-2), the K pairs of random numbers generate a K-dimensional boolean vector. With a hashing function, pixels with the same boolean vector are placed in the same bucket of a hash table. Since each pixel has L corresponding K-dimensional boolean vectors (the K pairs of random numbers are generated L times), it simultaneously belongs to L buckets in L hash tables. All pixels are assigned to L buckets in different hash tables. Then for a query pixel q, the pixels belonging to the L corresponding buckets of q are returned as the query results, and the k-nearest neighbors of q are searched only
within these pixels. For a certain dataset, given the approximation error threshold \( \varepsilon \), the optimal values of \( K \) and \( L \) can be estimated to provide the shortest running time. Details of this parameter selection can be found in Georgescu et al. (2003).

Table 4.3. Running times of \( k \)-nearest neighbor search on the two HSI datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Size (pixels ( \times ) bands)</th>
<th>Traditional</th>
<th>LSH (seconds)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urban</td>
<td>94,249 ( \times ) 162</td>
<td>3828</td>
<td>339</td>
<td>11.3</td>
</tr>
<tr>
<td>DC</td>
<td>97,319 ( \times ) 191</td>
<td>5224</td>
<td>445</td>
<td>11.7</td>
</tr>
</tbody>
</table>

Table 4-3 shows the achieved speedups of the \( k \)-nearest neighbor search on the two datasets. The number of nearest neighbors was \( k = 40 \). The approximation error \( \varepsilon = 0.05 \). The optimal values of \( K \) and \( L \) were selected as \( K = 30, L = 6 \) for Urban dataset and \( K = 35, L = 7 \) for DC dataset. The results were obtained with programs written in C programming language on a computer with a single 3 GHz CPU.

4.1.3 Dimensionality Reduction Experiments and Evaluations

Dimensionality reductions were performed on the two HSI datasets using the LSH-based Laplacian eigenmaps (LE) using the Matlab Toolbox for Dimensionality Reduction (Maaten et al., 2009). LSH replaced the method of nearest neighbor search in LE, and the outputs of LSH were created as inputs to LE. The algorithm of LE combined with LSH can provide DR results for the two datasets in an acceptable time (about 10 minutes).
Because the methods of intrinsic dimensionality estimation such as eigenvalue-based estimation and maximum-likelihood estimator (Maaten et al., 2009) always provide different results, the number of DR bands of LE for each dataset was selected as 20, a relatively large number. After the DR of LE, the DR bands in lower orders that contained noise or negligible information were manually selected to be discarded. The first 16 bands of the Urban dataset and the first 12 bands of the DC dataset out of 20 DR bands were kept, and are shown in Figures 4.3 and 4.4, respectively. Throughout the rest of this dissertation, the terms 'LE Urban data' and 'LE DC data' refer to the above selected DR bands of LE in the two datasets.

Figure 4.3. Urban LE data. (a)-(p): the first 16 DR bands.
Figure 4.3 continued
Figure 4.3 continued

Figure 4.4. DC LE data. (a)-(l): the first 12 DR bands.
Figure 4.4 continued
In order to evaluate the quality of the dimensionality-reduced data by LE, the widely used data compression method PCA was performed on the two datasets, while classifications were conducted on the LE data and the PCA data with three commonly used pixel-based classifiers, minimum-distance, SAM and maximum-likelihood. PCA was conducted using ENVI software, version 4.7, from ITT Visual Information Solutions. After the removal of the noisy PCs, the first four PCs of the Urban dataset that retained 99.31% of the variance criterion are shown in Figure 4.5, and the first four PCs of the DC dataset that retained 99.54% of the variance criterion are shown in Figure 4.6.

Figure 4.5. PCs of the Urban dataset. (a)-(d): the first four PCs.
Figure 4.6. PCs of the $DC$ dataset. (a)-(d): the first four PCs.
Figure 4.7. Classification results on the PCs of the *Urban* dataset. (a) Minimum-distance classification results (OA = 82.0%), (b) SAM classification results (OA = 79.9%), and (c) Maximum-likelihood classification results (OA = 94.8%).
Figure 4.8. Classification results on the PCs of the DC dataset. (a) Minimum-distance classification results (OA = 84.8%), (b) SAM classification results (OA = 79.8%), and (c) Maximum-likelihood classification results (OA = 95.6%).

The classification results on the PCs of the two datasets with the three pixel-based classifiers are shown in Figures 4.7 and 4.8. As to the classification results obtained on
the LE data of the two datasets, because they will be used as initial classification results for RGAC classifiers, they are shown in Figures 4.9 (a), 4.10 (a), 4.11 (a), 4.13 (a), 4.14 (a) and 4.15 (a) in the next section in order to have a direct visual comparison with the classification results refined by the RGAC classifiers.

Tables 4.4 and 4.5 list the OAs of the pixel-based classifications on the PCs and LE data from the two datasets. For both datasets, higher OAs were obtained on the LE data with the three classifiers. For the PCs, the maximum-likelihood classifier using both 1\textsuperscript{st} order and 2\textsuperscript{nd} order statistics obtained substantially higher OAs than the minimum-distance and SAM classifiers, which used only 1\textsuperscript{st} order statistics. However, for the LE data the two later classifiers obtained OAs comparable to those of the maximum-likelihood classifier. These results indicate LE performs well in reducing the dimensions of HSI data, while preserving the discrimination information of the original high-dimensional data.

Table 4.4. OAs of the classification results from PCs and LE data of Urban dataset.

<table>
<thead>
<tr>
<th>DR method</th>
<th>minimum-distance</th>
<th>SAM</th>
<th>maximum-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>82.0%</td>
<td>79.9%</td>
<td>94.8%</td>
</tr>
<tr>
<td>LE</td>
<td>92.0%</td>
<td>94.3%</td>
<td>95.9%</td>
</tr>
</tbody>
</table>

Table 4.5. OAs of the classification results from PCs and LE data of DC dataset.

<table>
<thead>
<tr>
<th>DR method</th>
<th>minimum-distance</th>
<th>SAM</th>
<th>maximum-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA</td>
<td>84.8%</td>
<td>79.8%</td>
<td>95.6%</td>
</tr>
<tr>
<td>LE</td>
<td>95.4%</td>
<td>95.0%</td>
<td>97.9%</td>
</tr>
</tbody>
</table>
4.2 RGAC Classification Experiments

4.2.1 Description of Classification Experiments

Following the RGAC classification framework described in Section 3.3, an initial classification is first performed with a pixel-based classifier and then refined by an RGAC classifier. Four pixel-based classifiers are used to provide the initial classification results. Besides the three commonly used classifiers, minimum-distance, SAM, and maximum-likelihood, another pixel-based classifier, support vector machines (SVM), which is generally considered to be better than the maximum-likelihood for HSI data (Melgani and Bruzzone, 2004; Camps-Valls and Bruzzone, 2005), was used with the SVM function provided by ENVI 4.7. The ENVI implementation of SVM uses the pairwise (one-against-one) classification strategy for multiclass classification. Radial Basis Function (RBF) was selected as the kernel type (Hsu et al., 2007) in the SVM classifications in this dissertation.

Two types of RGAC classifiers that included three extended Chan-Vese (ECV) classifiers (ECV-ED, ECV-SAM, and ECV-ML) and four edge-oriented (EO) classifiers (EO-ED, EO-SAM, EO-ML, and EO-SVM) were combined with the above pixel-based classifiers to refine the initial classification results. Note that the ECV classifiers are based on a certain distance metric, but the classifications of SVM are based on the marginal data. There is not a distance for each data point, so its results cannot be refined by an ECV classifier. So the initial results of the first three pixel-based classifiers were refined by their corresponding ECV classifiers, and the initial results of all four pixel-based classifiers were refined by their corresponding EO classifiers, respectively.
For all the RGAC classifiers, the only parameter to be set is the curvature coefficient $\mu$. The effects of the refinement functions of the RGAC classifiers are stronger when $\mu$ is larger. In order to evaluate the effects of the common refinement functions for different RGAC classifiers, $\mu$ was tuned from 0.2 to 2 with an interval of 0.2 for every RGAC classifier. So the following four groups of experiments were conducted:

1) minimum-distance classification, whose results were refined by ECV-ED and EO-ED with $\mu = 0.2, 0.4, \ldots, 2.0$, respectively;

2) SAM classification, whose results were refined by ECV-SAM and EO-SAM with $\mu = 0.2, 0.4, \ldots, 2.0$, respectively;

3) maximum-likelihood classification, whose results were refined by ECV-ML and EO-ML with $\mu = 0.2, 0.4, \ldots, 2.0$, respectively;

4) SVM classification, whose results were refined by EO-SVM with $\mu = 0.2, 0.4, \ldots, 2.0$.

Note that the initial classification results of any pixel-based classifiers can be refined by the EO classifier. But for an ECV classifier that is based on a certain distance metric, the initial classification results should be provided by the pixel-based classifier using the same distance metric. Otherwise the initial classification results can have much inconsistency with the final refined results and it will take much more iterations for the contour evolutions to reach a stable state.

With the training samples and test samples obtained from the ground-truth, the above classifications were conducted with Matlab programs on a computer with a single 3 GHz CPU. The running time of the RGAC classifiers reflects how quickly the contour
evolutions reach the stable state. In this dissertation, the stable state of contour evolutions is defined as: in 10 consecutive iterations, the number of pixels that change labels is less than 0.5% of the number of total pixels in the image. In general, the larger \( \mu \) is, the more pixel label changes will take place and the longer it will take for the contour evolutions to reach a stable state. The running time of all the RGAC classifiers was typically 1 - 2 minutes on either of the two datasets, and the longest running time on each dataset was 124 seconds on the *Urban* dataset and 89 seconds on the *DC* dataset. The OAs (overall accuracies) of the classification results were calculated as well. In Sections 4.2.2 and 4.2.3, the initial classification results and the best corresponding RGAC classification results are presented. Detailed analysis will be made in Section 4.3.

In order to find out the reasons for the differences in the performance of different RGAC classifiers, the image force maps, which are intermediate results in RGAC classifications as described in Section 3.1.3, were obtained, and the histograms of the image force values of the boundary pixels were calculated and analyzed in Section 4.3.2. The reason why only the boundary pixels are counted is that the refinement functions only apply to the neighboring pixels within a narrow band of the segment boundaries. In this dissertation the boundary pixels are defined as the pixels within the distance of 1.2 pixels to the segment boundaries in the initial classification results. Namely, they are the pixels corresponding to the elements in the initialized level set matrix with the absolute values less than 1.2.
4.2.2 RGAC Classifications on the *Urban* LE Data

Figure 4.9. Refinement of minimum-distance classification results by ECV-ED and EO-ED on the *Urban* LE data. (a) Minimum-distance classification results (OA = 92.0%), (b) ECV-ED classification results with the highest OA (95.5%, \( \mu = 1.4 \)), and (c) EO-ED classification results with the highest OA (94.7%, \( \mu = 1.8 \)).
Figure 4.10. Refinement of SAM classification results by ECV-SAM and EO-SAM on Urban LE data. (a) SAM classification results (OA = 94.3%), (b) ECV-SAM classification results with the highest OA (97.4%, $\mu = 1.4$), and (c) EO-SAM classification results with the highest OA (96.9%, $\mu = 1.6$).
Figure 4.11. Refinement of maximum-likelihood classification results by ECV-ML and EO-ML on the *Urban* LE data. (a) Maximum-likelihood classification results (OA = 95.9%), (b) ECV-ML classification results with the highest OA (98.2%, $\mu = 0.2$), and (c) EO-ML classification results with the highest OA (98.0%, $\mu = 1.4$).
Figure 4.12. Refinement of SVM classification results by EO-SVM on the Urban LE data. (a) SVM classification results (OA = 99.4%), and (b) EO-SVM classification results with the highest OA (99.8%, μ = 1.0).

The initial classification results and the corresponding RGAC classification results with highest OAs obtained on the Urban LE data are shown in Figures 4.9 - 4.12. Compared with the initial pixel-based classification results, the classification results of the RGAC classifiers are visually less noisy and generally have higher OAs (detailed analysis is in Section 4.3). Note that some segments such as those marked in Figures 4.9 (b) and (c) are broken and thus misclassified. This indicates the problem of over-refinement of the RGAC classifiers and will be discussed in Section 4.3.3.

4.2.3 RGAC Experiments on the DC LE data

The initial classification results and the corresponding typical RGAC classification results obtained on the DC LE data are shown in Figures 4.13 - 4.16.
Figure 4.13. Refinement of minimum-distance classification results by ECV-ED and EO-ED on the DC LE data. (a) Minimum-distance classification results (OA = 95.4%), (b) ECV-ED classification results with the highest OA (96.7%, μ = 0.8), and (c) EO-ED classification results with the highest OA (96.4%, μ = 1.6).
Figure 4.14. Refinement of SAM classification results by ECV-SAM and EO-SAM on the DC LE data. (a) SAM classification results (OA = 95.0%), (b) ECV-SAM classification results with the highest OA (96.2%, $\mu = 1.0$), and (c) EO-SAM classification results with the highest OA (96.0%, $\mu = 1.6$).
Figure 4.15. Refinement of maximum-likelihood classification results by ECV-ML and EO-ML on the DC LE data. (a) Maximum-likelihood classification results (OA = 97.9%), (b) ECV-ML classification results with the highest OA (98.9%, μ = 0.2), and (c) EO-ML classification results with the highest OA (99.0%, μ = 1.6).
4.3 Analysis of the RGAC Classification Results

4.3.1 Evaluations on Overall Performance

Figures 4.17 and 4.18 plot the OAs and segment numbers of tested RGAC classifiers on the LE data of the two datasets with changing $\mu$. For a plotted curve of an RGAC classifier, the value (OA and segment number) at $\mu = 0$ is the value of the corresponding initial classification results. For example, in Figure 4.17 (a) and Figure 4.17 (b), the value of the blue lines at $\mu = 0$ is 92.0% for both lines, which is the OA of the minimum-distance classification results on Urban LE data.

From Figures 4.17 and 4.18, it can be observed that the OAs of most RGAC classification results were higher than the OAs of their corresponding initial classification results.
results (exceptions were found for the ECV-ML classifier, which will be explained in Section 4.3.2). At the same time, the refined classification results contained fewer segments, which resulted in less-noisy thematic maps as shown in the previous section.

![Figure 4.1](image)

Figure 4.17. OAs of the classification results on the *Urban* LE data with changing $\mu$. (a) OAs of the three ECV classifiers, (b) OAs of the four EO classifiers, (c) segment numbers of the classification results of the three ECV classifiers, and (d) segment numbers in the classification results of the four EO classifiers.
Figure 4.18. OAs of the classification results on the DC LE data with changing $\mu$. (a) OAs of the three ECV classifiers, (b) OAs of the four EO classifiers, (c) segment numbers of the classification results of the three ECV classifiers, and (d) segment numbers in the classification results of the four EO classifiers.

Table 4.6. OAs of the pixel-based classifiers on the LE data of the two datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>minimum-distance</th>
<th>SAM</th>
<th>maximum-likelihood</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urban</td>
<td>92.0%</td>
<td>94.3%</td>
<td>95.9%</td>
<td>99.4%</td>
</tr>
<tr>
<td>DC</td>
<td>95.4%</td>
<td>95.0%</td>
<td>97.9%</td>
<td>97.9%</td>
</tr>
</tbody>
</table>

Because an RGAC classifier refines the initial classification results, its accuracy is dependent on the pixel-based classifier that provides the initial classification results.
The OAs of the four pixel-based classifiers (minimum-distance, SAM, maximum-likelihood and SVM) are listed in Table 4.3. SVM provided the highest OA (99.4%) on the Urban LE data, and also provided the highest OA (97.9%) that is the same as the OA of the maximum-likelihood classifier on the DC LE data. The maximum-likelihood classifier also provided the second highest OA (95.9%) on the Urban LE data. So overall, SVM performed the best out of the four pixel-based classifiers, and the maximum-likelihood classifier which makes use of 2nd-order statistics of the samples also outperformed the minimum-distance and SAM classifiers that only consider 1st-order statistics. This information indicates that SVM and maximum-likelihood classifiers are both able to provide accurate initial classification results from the LE data of HSI datasets to be refined by RGAC classifiers, and SVM is more likely to provide better results due to its lower sensitivity to the curse of dimensionality (Melgani and Bruzzone, 2004; Camps-Valls and Bruzzone, 2005). However, it should be noted that the results of SVM can only be refined by the corresponding EO classifier, EO-SVM. So the better performance of SVM also demonstrates the advantage of the high adaptability.

From Figures 4.17 (a), (b) and Figure 4.18 (a), (b), it can be observed that when an ECV classifier and an EO classifier are applied to the same initial classification results, the OAs of the latter are generally slightly lower than those of the former. Table 4.4 and Table 4.5 show the highest OAs and corresponding values of \( \mu \) of the tested RGAC classifiers on the LE data of the two datasets (in Table 4.4 the values of the highest OA and \( \mu \) of the ECV-ML classifier are obtained from the experiments described in Section 4.3.2). By comparing the corresponding OAs in the two tables, it is found that
the highest OAs of the EO classifiers are generally lower than those of the corresponding ECV classifiers. The maximum, minimum and mean of the differences are 0.8\%, 0\%, and 0.3\%, respectively. Note that when the initial classification results are provided by the maximum-likelihood classifier, the differences in the highest OAs of the two types of classifiers are very small, which are 0.2\% on the *Urban* LE data and 0 on the *DC* LE data. So when an EO classifier and an ECV classifier are both applicable on the same initial classification results, the performance of the ECV classifier is slightly better, but the difference is very small, especially when the initial classification results are provided by a highly accurate classifier such as maximum-likelihood.

Table 4.7. Highest OAs and corresponding $\mu$ of the ECV classifiers on the LE data of the two datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>ECV-ED</th>
<th>ECV-SAM</th>
<th>ECV-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Highest OA</td>
<td>$\mu$</td>
<td>Highest OA</td>
</tr>
<tr>
<td><em>Urban</em></td>
<td>95.5%</td>
<td>1.4</td>
<td>97.4%</td>
</tr>
<tr>
<td><em>DC</em></td>
<td>96.7%</td>
<td>0.8</td>
<td>96.2%</td>
</tr>
</tbody>
</table>

Table 4.8. Highest OAs and corresponding $\mu$ of the EO classifiers on the LE data of the two datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>EO-ED</th>
<th>EO-SAM</th>
<th>EO-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Highest OA</td>
<td>$\mu$</td>
<td>Highest OA</td>
</tr>
<tr>
<td><em>Urban</em></td>
<td>94.7%</td>
<td>1.8</td>
<td>96.9%</td>
</tr>
<tr>
<td><em>DC</em></td>
<td>96.4%</td>
<td>1.6</td>
<td>96.0%</td>
</tr>
</tbody>
</table>
From Table 4.7 and Table 4.8, it can also be observed that for the ECV classifiers, the values of µ with which the highest OAs were obtained vary from 0.18 to 1.4, but for the EO classifiers, the values of µ only vary from 1.4 to 1.8. This indicates the optimal values of µ are different for different ECV classifiers and different datasets. But for the EO classifiers, the optimal value of µ can be safely set as an experience value of 1.6. The reasons for the differences in the optimal values of µ between the ECV and EO classifiers will be discussed in the next section.

4.3.2 Effects of the Distribution of Image Force Values

As mentioned in Section 3.1.2, given a µ, the curvature term tends to dominate the trade-off with the image term when the value of the later is relatively small. This can be confirmed by the experimental results. The pixels belonging to a certain class in the initial classification results that are relabeled during the RGAC classification process can be considered to be suppressed into other classes. By comparing the initial classification results to the RGAC-refined classification results, the suppressed pixels can be extracted and certain statistics can be calculated. Figure 4.19 shows an example of the means of the image force values of the suppressed pixels of the ECV-ED classifier on the DC LE data. It can be seen that these means increase with µ, which indicates that larger curvature terms are needed to suppress a larger image force. Equivalently, the smaller the image force, the more likely it is to be suppressed by the curvature term.

Image force maps are generated during the RGAC classification procedures. Figure 4.20 shows the image force maps of the tested RGAC classifiers on the DC LE
data with the same color scheme, where dark colors indicate weak image force (close to 0) and brighter colors indicate a strong image force (close to 1). Figure 4.21 shows the histograms of the boundary pixels’ image force values of the RGAC classifiers.

![Figure 4.19](image.png)

Figure 4.19. Means of the image force values of the pixels suppressed by the ECV-ED classifier with different $\mu$ on the DC LE data.

As shown in Figures 4.21 (a), (b) and (c), the histograms of the boundary pixels’ image force values of the three ECV classifiers are quite different. That means the image terms in these models have different sensitivities to $\mu$ and the optimal values of $\mu$ are different. For example, a significant portion of the image force values of the ECV-ML classifier are distributed in the region close to 0. Correspondingly, the image force map of the ECV-ML classifier as shown in Figure 4.20 (c) is overall much darker than the other three image force maps. This indicates that for the ECV-ML classifier, the curvature term can easily dominate the trade-off with the image term to have the refinement functionalities take effect even with a small $\mu$. This is why in the experiments on the LE data of both datasets, the OA plots of the ECV-ML classifier both reach
maxima when $\mu = 0.2$ and then drop dramatically as shown in Figure 4.17 (a) and Figure 4.18 (a).

Figure 4.20. Image force maps of the RGAC classifiers on the DC LE data: (a) ECV-ED; (b) ECV-SAM; (c) ECV-ML; and (d) EO.
Figure 4.21. Histograms of the boundaries pixels’ image force values of RGAC classifiers on the DC LE data: (a) ECV-ED; (b) ECV-SAM; (c) ECV-ML; (d) EO-ED; (e) EO-SAM; (f) EO-ML; and (g) EO-SVM.
Figure 4.21 continued
Figure 4.22. OAs of the ECV-ML classification results with $\mu$ increasing from 0 to 0.3 with a 0.03 interval on the LE data of two datasets: (a) Urban; (b) DC.

So for the ECV-ML classifier, $\mu$ should be selected in a smaller range rather than $[0, 2]$. Figure 4.22 shows the OAs of the ECV-ML classifier on the LE data of the two datasets while $\mu$ increases from 0 to 0.3 with a 0.03 interval. The highest OA is achieved on the Urban LE data when $\mu = 0.18$ (OA = 98.2%) and on the DC LE data when $\mu = 0.18$ (OA = 99.0%).

On the other hand, the ECV-ML classifier's image force map for the EO classifiers (Figure 4.20 (d)) are generally brighter, and the image force histograms (Figure 4.21 (d), (e), (f), (g)) show that the image force values are mostly distributed in the range [0.5, 1]. That means $\mu$ must be relatively large to have the curvature term
dominate its trade-off with the image term. It also explains why the changing of the OAs and segment numbers of the EO classifiers (Figures 4.17 (b), (d), Figures 4.18 (b), (d)) leads to inertia at the beginning in response to the increase of \( \mu \).

For the EO classifiers, there is only one unique image force map on one image dataset. Because the boundaries of various initial classification results are different, there will be slight differences among the histograms of the boundary pixels' image force values given different initial classification results. As shown in Figures 4.21 (d), (e), (f), (g), there are only slight differences in the histograms of the boundary pixels' image force values of the four EO classifiers. So for the EO classifiers, no matter what the initial classification results are, the optimal values of \( \mu \) are constant in different situations and \( \mu \) can be safely selected as an experience value.

### 4.3.3 The Over-refinement Problem

The problem of over-refinement can be expected for some type of objects when the selected \( \mu \) is large. A typical example is that the stretching force in the active contour models tends to break the narrow linear segments. Specifically, the curvature embedded in the active contour models generates negative values for concaves, and therefore for the narrow linear segments with unsmooth boundaries, the negative values of the curvature result in the force that tends to break these linear features. In the experiments by Ball and Bruce (2005), it was reported that the proposed level-set classifier 'had trouble finding the small paths' in a HYDICE dataset.
This problem also exists for RGAC classifiers. Besides the curvature term, the representations of the image term in the RGAC model also contribute to this problem. The narrow linear features on remote sensing images usually contain spectral mixture, which results in a weak image force for RGAC classifiers. With the weak image force and the negative curvature force (curvature term value multiplied by $\mu$), the sum of the two forces tends to be negative when $\mu$ is large, so the corresponding segment pixels can be suppressed to other classes and the segments of the narrow paths are often misclassified in the corresponding RGAC classification results.

The problem of over-refinement was found in the performed experiments. One example is the missed linear segments as marked in Figure 4.9. Another example would be the class street in the experiments on the DC LE data. Take the EO-ED classifier on the DC LE data as example. By plotting the UAs (user's accuracy, which is the ratio of the number of pixels correctly classified as a class over the total number of pixels classified as that class) of all seven classes, as shown in Figure 4.23, an apparent decrease in the UAs of the class street (the yellow curve) can be observed after the UA curve reaches the maxima at $\mu = 0.4$. A similar pattern of decrease can also be seen in the UAs of the class street in the other RGAC classification results.
Figures 4.23. UAs of the seven classes on the LE DC data in classification results of EO-ED.

Figures 4.24 (a) and (b) show the segments of the class *street* with \( \mu = 0.4 \) (UA = 92.8\%) and \( \mu = 1.6 \) (UA = 90.3\%), respectively, in the EO-ED classification results on the DC LE data. It can be seen that some linear segments in the \( u = 0.4 \) segments (marked in Figure 4.24 (a)) were missed or broken in the \( \mu = 1.6 \) segments. Since these segments were all selected in the ground-truth of the class *street*, the missing or breaking of them caused the lower UA of that class as shown by the yellow curve in Figure 4.23.

The widths of some horizontal streets, including the four missed ones, are approximately 4 meters (measured on GoogleEarth), which suggests a high probability of spectral mixture in this HSI dataset whose spatial resolution is 3.5 meters. By checking the spectra of some *street* sample pixels in the original HSI data and LE data, it was found that the four missed street segments contained a higher level of spectral mixture.
which caused the weaker image force in the EO model. For this reason, some pixels on these segments were suppressed to other classes. Note that this issue is a trade-off between over-refinement and noise reduction. Some small segments that appear noisy in the $\mu = 0.4$ segments were eliminated in the $\mu = 1.6$ segments, but this improvement in noise reduction was not reflected in the statistical accuracies because most of these segments were not selected from the ground-truth.

![Figure 4.24. Segments of the class street in the EO-ED classification results obtained on the DC LE data. (a) street segments with $\mu = 0.4$, and (b) street segments with $\mu = 1.6$.](image)
Figure 4.25. Comparison of classification results with uniform µ and multiple µ. (a) Classification results EO-ED on DC LE data with µ = 1.6. (b) Classification results EO-ED on DC LE data with µ = 0.4 for class street and µ = 1.6 for other six classes.

Due to the stretching force embedded in the RGAC models and the weak image force caused by the spectral mixture, over-refinement can be a problem for linear features with widths similar to the spatial resolution of the images. A solution to this problem is the use of multiple curvature coefficient values for different classes. If it is known that some classes include such narrow linear features in the image and it is more desirable to save these features than to suppress noise, a smaller µ can be selected for these classes. The value of µ for such a special class can be an experience value, or experiments with changing µ can be performed on a small training area and the µ would be selected to be the value for which the highest UAs of these classes are obtained. An example of this would be the EO-ED classification on DC LE data. By setting µ = 0.4 for
class street and setting $\mu = 1.6$ for the other six classes, the classification results shown in Figure 4.25 (b) can be obtained. Compared with the EO-ED classification results where $\mu = 1.6$ for all classes as shown in Figure 4.25 (a), the classification results with multiple $\mu$ values would be able to secure more narrow street segments. The UA of the street class would increase from 90.3% to 93.6%.

In order to illustrate the effects of the over-refinement problem on different types of objects, the UAs of the seven classes in the DC datasets and the ten classes in the Urban dataset in the RGAC classification results are shown in Figures 4.26 and 4.27. In Figures 4.26 and 4.27, each figure represents a class and the seven curves in each figure are the UA curves of that class from the seven RGAC classifiers. All the figures share the same legend to represent the seven RGAC classifiers as shown in Figures 4.26 (a) and 4.27 (a). Note that as discussed in Section 4.3.2 and shown in Figure 4.22, the ECV-ML classifier has different sensitivity to $\mu$ compared with the other RGAC classifiers, so the UA curves of the ECV-ML classifier is stretched to be shown together with the UA curves of the other RGAC classifiers. In Figures 4.26 and 4.27, the $x$-axis $\mu'$ represents the range $[0, 0.3]$ with a 0.03 interval for the ECV-ML classifier, and represents the range $[0, 2]$ with a 0.2 interval for the other six RGAC classifiers.
Figure 4.26. UAs of the ten classes in the *DC* dataset in the classification results of the seven RGAC classifiers. All figures share the same legend as the one shown in (a).
Figure 4.26 continued

- (e) tree (DC)
- (f) water (DC)
- (g) roof (DC)
Figure 4.27. UAs of the ten classes in the Urban dataset in the classification results of the seven RGAC classifiers. All figures share the same legend as the one shown in (a).
Figure 4.27 continued

(e) parking lot (Urban)

(f) road A (Urban)

(g) road B (Urban)

(h) soil (Urban)

continued
According to the trend of the UA curves in Figures 4.26 and 4.27, the classes in the two datasets can be roughly categorized into three types. The first type contains the classes that include the narrow linear features and are sensitive to the over-refinement problem such as the class street in the DC dataset whose UA curves drop with the increasing of $\mu$, as shown in Figure 4.26 (d).

The second type includes the targets that are relatively unaffected by the over-refinement problem such as the classes grass (Figure 4.26 (a)) and water (Figure 4.26 (f)) in the DC dataset, as well as the classes concrete (Figure 4.27 (c)) grass (Figure 4.27 (d)), parking lot (Figure 4.27 (e)), soil (Figure 4.27 (h)) and tree (Figure 4.27 (i)) in the Urban dataset. Most of the objects in those classes have relatively larger sizes compared to objects of other classes, and most of the UA curves of those classes do not have significant decreases with the increase of $\mu$. 
The third type includes all other classes in the two datasets. The sensitivity of these classes to increasing $\mu$ is between the two types mentioned above. With the increase of $\mu$, their UAs increase when $\mu$ is relatively small, and start to drop when $\mu$ increases. This trend of the UA changes with increasing $\mu$ is reflected in the OA curves, as shown in Figures 4.17 (a), (b) and Figures 4.18 (a), (b).

The above discussions indicate that different types of objects have different levels of sensitivity to the over-refinement problem due to their relative size in the images and their geometric characteristics. With an appropriate value of $\mu$, improvements can be expected to be made on most (second and third of the above three types) of the objects. With regard to the first type of objects which are comprised of narrow linear features, selecting a small $\mu$ can help achieve an optimal compromise between noise suppression and over-refinement. This further demonstrates that the strategy of applying multiple values of $\mu$ to different classes, with a smaller $\mu$ to the classes which contain narrow linear features and a larger $\mu$ to other classes, can be a suitable solution to the over-refinement problem.
CHAPTER 5

CONCLUSIONS

This dissertation addresses the issue of how to utilize region-based active contour (RGAC) models for classifications of hyperspectral remote sensing images. As introduced in Chapter 2, as a binary spectral-spatial classifier, the RGAC models have the advantages of easy implementation and low requirements for initial contour placements, and as a result have good potential for classifications in remote sensing images. The analysis and experiments in Chapter 2 revealed that for RGAC models, the key feature is the abilities of the refinement functions to smooth segment boundaries and suppress noise. As a result, a two-stage classification strategy is used for classifications of RGAC models, and a corresponding classification framework is developed.

Two types of RGAC classifiers were proposed and studied in this dissertation. One is based on an extended Chan-Vese model (ECV), which enables the RGAC models to incorporate the commonly used distance metrics. The other is based on a new type of RGAC model, the edge-oriented (EO) model. Unlike the ECV classifiers that can only be applied to classification results based on certain distance metrics, the EO classifiers can be used to refine any classification results and thus have much more flexibility.

The RGAC classifiers including the ECV and EO classifiers were tested in the classification experiments. More specifically, experiments were conducted on two HSI
datasets. In order to overcome the problem of the curse of dimensionality, the HSI data were reduced in order to lower the dimensions via the nonlinear dimensionality reduction method of Laplacian eigenmaps (LE) using the algorithm of LSH to speed up the step of \( k \)-nearest neighbor search in LE. Evaluations of the quality of the LE DR data were performed and compared to the DR data of PCA. From the experimental results and discussions in Chapter 4, the following observations can be made.

1) With the help of LSH, the nonlinear DR method of LE is able to perform dimensionality reductions on relatively large HSI datasets within an acceptable time; and the DR data of LE are better than those of PCA in terms of providing more accurate classification results using the commonly used pixel-based classifiers, including minimum-distance, SAM and maximum-likelihood classifiers. In particular, the classification accuracy of the minimum-distance and SAM classifiers can be significantly improved at the level of the maximum-likelihood classifier;

2) The RGAC classifiers can effectively refine the initial pixel-based classification results by increasing the OAs and producing less noisy thematic maps.

3) The accuracies of the RGAC classifiers are dependent on the initial classification results. Since some well-recognized classifiers such as SVM can be used to provide initial results only for EO classifiers, the EO classifiers has much more adaptability for classifications of hyperspectral images;
4) When an ECV classifier and an EO classifier are both applicable on the same initial classification results, the former can provide better accuracy, but the difference is small, especially when the initial results are provided by a high-accuracy classifier such as the maximum-likelihood classifier;

5) The difference in the distribution of image force values among different ECV models cause difficulty in selection of the optimal value of μ, which varies among different ECV classifiers and different datasets;

6) The problem of over-refinement affects all RGAC classifiers. It is an issue related to not only the size of the objects, but the spatial resolution of the image sensor. A solution to this problem may be the use of multiple values of μ for difference classes, but this is a compromise between over-refinement and noise suppression.

In summary, the contributions of this dissertation include:

1) Introduction of the method of LSH-based Laplacian eigenmaps making it possible to perform non-linear dimensionality reductions on relatively large hyperspectral datasets;

2) Development of a framework for the RGAC-based image enabling RGAC classifiers to be easily combined with pixel-based classifiers;

3) Proposing an extended Chan-Vese model (ECV) enabling incorporation of different distance measurements to process multiple-band remote sensing images;
4) Proposing a new type of edge-oriented RGAC model which can combine with any pixel-based classifier and has much more adaptability than the ECV models to be applied on remote sensing image classifications;

5) Evaluation of the performance of the proposed RGAC classifiers and classification experiments on two HSI datasets.

The following work will be performed in the future:

1) Modification of the edge-oriented RGAC model to help overcome the over-refinement problem, e.g. by adopting the additional information of linear connectivity into the model to prevent narrow linear feature to be broken;

2) Further evaluations on the effects of the approximate error in the LSH algorithm on the dimensionality reduction results;

3) More tests and evaluations on the combinations of LSH with other nonlinear dimensionality reduction algorithms such as local linear embedding, and utilization of appropriate techniques to estimate the intrinsic dimensions of HSI data to automatically determine the number of dimensionality reduction bands.
BIBLIOGRAPHY


