Air Force Institute of Technology

AFIT Scholar

Theses and Dissertations

Student Graduate Works

3-5-2009

Abstracting GIS Layers from Hyperspectral Imagery

Torsten E. Howard

Follow this and additional works at: https://scholar.afit.edu/etd

Part of the Physical and Environmental Geography Commons, and the Signal Processing Commons

Recommended Citation

Howard, Torsten E., "Abstracting GIS Layers from Hyperspectral Imagery" (2009). *Theses and Dissertations*. 2482.
https://scholar.afit.edu/etd/2482

This Thesis is brought to you for free and open access by the Student Graduate Works at AFIT Scholar. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

Abstracting GIS Layers
from Hyperspectral Imagery

THESIS

Torsten E. Howard, Captain, USAF

AFIT/GE/ENG/09-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/09-21

Abstracting GIS Layers from Hyperspectral Imagery

THESIS

Presented to the Faculty
Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Electrical Engineering

Torsten E. Howard, BSEE, MBA.
Captain, USAF

March 2009

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GE/ENG/09-21

Abstracting GIS Layers from Hyperspectral Imagery

Torsten E. Howard, BSEE, MBA.
Captain, USAF

Approved:

/signed/	5 Mar 2009
Maj Michael J. Mendenhall, Ph.D. (Chair)	date
/signed/	5 Mar 2009
Dr. Gilbert L. Peterson (Member)	date
/signed/	5 Mar 2009
Dr. Juan R. Vasquez (Member)	date

AFIT/GE/ENG/09-21

Abstract

Modern warfare methods in the urban environment necessitates the use of multiple layers of sensors to manage the battle space. *Hyperspectral imagers* are one possible sensor modality to provide remotely sensed images that can be converted into Geographic Information Systems (GIS) layers. GIS layers abstract knowledge of roads, buildings, and scene content and contain *shape files* that outline and highlight scene features. Creating shape files is a labor-intensive and time-consuming process. The availability of shape files that reflect the current configuration of an *area of interest* significantly enhances Intelligence Preparation of the Battlespace (IPB).

The solution presented in this thesis is a novel process to automate the creation of shape files by exploiting the spectral-spatial relationship of a *hyperspectral image* cube. It is assumed that "a-priori" *endmember spectra*, a spectral database, or specific scene knowledge is *not* available. The *topological neighborhood* of a Self Organizing Map (SOM) is segmented and used as a spectral filter to produce six initial *object maps* that are spatially processed with *logical* and *morphological operations*. A novel road finding algorithm connects road segments under significantly tree-occluded roadways into a contiguous road network. The manual abstraction of GIS shape files is improved into a semi-automated process. The resulting shape files are not susceptible to deviation from orthorectified imagery as they are produced directly from the hyperspectral imagery.

The results are eight separate high-quality GIS layers (Vegetation, Non-Tree Vegetation, Trees, Fields, Buildings, Major Buildings, Roadways, and Parking Areas) that follow the terrain of the hyperspectral image and are separately and automatically labeled. Spatial processing improves layer accuracy from 85% to 94%. Significant layer accuracies include the "road network" at 93%, "buildings"at 97%, and "major buildings" at 98%.

Acknowledgements

First and foremost, I am thankful to God for the desire and motivation to complete this thesis. May it be an attribute to His beautiful and good creation.

I am grateful to the Unites States Air Force for the opportunity to attend such an outstanding program from which I have learned and grown a great deal. The valuable investment will be returned many times over.

I am indebted to my thesis advisor, Maj Michael J. Mendenhall, and my thesis committee members, Dr. Bert Peterson and Dr. Juan Vasquez. Their continued support and contribution enabled me to accomplish this thesis and push myself to higher achievement.

Finally, thanks to Maj Dan Schmidt, Capt Rich Muster, and Capt Jose Gonzalez whose daily doses of humor helped to sustain the morale and camaraderie of the TecEdge lab.

Torsten E. Howard

Table of Contents

Page

Abstract iv

Acknowledgements v

List of Figures ix

List of Tables xii

List of Abbreviations xiii

I. Introduction 1-1
  1.1 A Brief History of Remote Sensing 1-1
  1.2 Hyperspectral Imagery 1-3
    1.2.1 Weaponizing Hyperspectral Imagery 1-3
    1.2.2 Collecting in the Urban Environment 1-4
    1.2.3 Augmenting Chromatic Imagery with Hyperspectral Imagery 1-6
  1.3 Geographic Information Systems (GIS) 1-6
    1.3.1 GIS Layers 1-6
    1.3.2 GIS Layers and Intelligence Preparation of the Battlespace 1-8
    1.3.3 Hyperspectral Images and GIS Layers 1-8
  1.4 Spectral and Spatial Analysis 1-9
  1.5 Research Goals 1-11
  1.6 Organization 1-11

II. Background Information 2-1
  2.1 Hyperspectral Images 2-1
    2.1.1 Collecting Hyperspectral Images 2-1
    2.1.2 Radiance and Reflectance 2-3
    2.1.3 Model of Light Transmission and Radiance Measurement 2-4
    2.1.4 Atmospheric Correction 2-4
    2.1.5 Spectral Features 2-6
    2.1.6 Spectral and Spatial Pre-Processing 2-6
  2.2 Spectral Analysis 2-8
    2.2.1 k-Means Clustering 2-9
    2.2.2 Self-Organizing Maps 2-11

2.2.3 SOM Learning Algorithm 2-13
2.2.4 SOM Learning Algorithm Example 2-15
2.2.5 Topological Neighborhood Preservation 2-16
2.2.6 SOM in Literature 2-16
2.2.7 Using a SOM in Spectral Analysis 2-19
2.2.8 The Spectral Database Tool 2-19
2.2.9 Normalized Difference Vegetative Index (NDVI) 2-20
2.2.10 Separating Trees from Grass 2-22
2.3 Spatial Analysis 2-23
2.4 From Spectral to Spatial Analysis 2-24
2.4.1 Spatial Analysis 2-26
2.4.2 Morphological Operations 2-27
2.4.3 Measuring Accuracy 2-31

III. Methodology 3-1
3.1 Process Inputs 3-3
3.2 Spectral Analysis 3-4
3.2.1 Self-Organizing Map (SOM) 3-6
3.2.2 Normalized Difference Vegetation Index 3-6
3.2.3 NIR Scatter Tree Identification 3-8
3.2.4 “Non-Tree Vegetation” *SOM Segmentations* . . 3-10
3.2.5 “Rooftop” and “Roadway” *SOM Segmentations* 3-11
3.3 Create the *Initial Object Maps* 3-13
3.4 Transitioning from Spectral to Spatial Analysis 3-16
3.5 Spatial Analysis 3-17
3.5.1 Step 1: The “Vegetation”, “Tree”, and “Fields” *Final Object Maps* 3-20
3.5.2 Step 2: The “Parking Areas” *Final Object Map* 3-21
3.5.3 Step 3: The “Major Buildings” *Final Object Map* 3-22
3.5.4 Step 4: The “Best Known Good Roadway” *Object Map* 3-24
3.5.5 Step 5: The “Road Network” *Final Object Map* 3-25
3.5.6 Step 6: The “Non-Tree Vegetation” and the “Buildings” *Final Object Maps* 3-29
3.6 Finalizing the GIS Layers 3-30

IV. Experimental Results and Analysis 4-1
4.1 Process Inputs 4-1
4.2 Spectral Analysis 4-4
4.2.1 Self-Organizing Map 4-4

4.2.2 Normalized Difference Vegetation Index 4-7
4.2.3 NIR Scatter Tree Identification 4-11
4.2.4 “Non-Tree Vegetation” *SOM Segmentations* . . 4-15
4.2.5 “Rooftop” and “Roadway” *SOM Segmentations* 4-16
4.3 Create the Initial Object Maps 4-19
4.4 Transitioning from Spectral to Spatial Analysis 4-19
4.5 Spatial Analysis 4-21
4.5.1 Step 1: The “Vegetation”, “Tree”, and “Fields” *Final Object Maps* 4-21
4.5.2 Step 2: The “Parking Areas” *Final Object Map* 4-24
4.5.3 Step 3: The “Major Buildings” *Final Object Map* 4-24
4.5.4 Step 4: The “Best Known Good Roadway” *Object Map* 4-28
4.5.5 Step 5: The “Road Network” *Final Object Map* 4-28
4.5.6 Step 6: The “Non-Tree Vegetation” and the “Buildings” *Final Object Maps* 4-38
4.6 Finalizing the GIS Layers 4-38
4.7 Summary 4-40
4.8 Conclusions 4-42

V. Future Work and Concluding Remarks 5-1
5.1 Supported Research 5-1
5.1.1 Vehicular Target Behavior Prediction 5-1
5.1.2 Target Detection with Shadows in SAR 5-1
5.2 Using the GIS Layers as Intelligence Products 5-3
5.3 Future Work 5-4
5.4 Concluding Remarks 5-5

Bibliography BIB-1

Vita VITA-1

List of Figures

Figure Page

1.1. Enhancing four USAF core competencies. 1-2

1.2. Weaponizing hyperspectral imagery. 1-5

1.3. Example of an RGB chromatic image. 1-7

1.4. Visualization of spectral and spatial analysis. 1-10

2.1. Example of a hyperspectral image cube. 2-2

2.2. Example of a hyperspectral image before *geocorrection*. 2-3

2.3. Model of light transmission and radiance measurement. 2-5

2.4. Example of wavelength correlation in hyperspectral imagery. . . 2-7

2.5. Examples of the *k*-means clustering algorithm. 2-10

2.6. Visualization of the dimensionality of hyperspectral images and self-organizing maps (SOM). 2-12

2.7. Visualization of the SOM learning process. 2-14

2.8. Example of the SOM learning process. 2-17

2.9. Example of *topological preservation* in a SOM. 2-18

2.10. Example of the Normalized Difference Vegetation Index (NDVI) applied to a hyperspectral image. 2-20

2.11. Example morphological operations. 2-29

2.12. Example of calculating the non-equal weighted accuracy. 2-30

3.1. The process flow diagram. 3-2

3.2. Example of a hyperspectral image data cube. 3-4

3.3. The process flow diagram for the *spectral analysis* step. 3-5

3.4. Example of verifying SOM convergence. 3-7

3.5. Example of applying NDVI to a SOM. 3-8

3.6. Visualization of the NIR scatter tree ID algorithm. 3-9

Figure Page

3.7. Example of applying the NIR scatter tree ID algorithm to a hyperspectral image. 3-9

3.8. Example of selecting exemplar "in-scene" spectra and a SOM segmentation threshold. 3-12

3.9. Example of SOM neighborhood topology. 3-14

3.10. The process flow diagram for the *create the initial object maps* step. 3-15

3.11. Example of creating an *initial object map* from a *SOM segmentation*. 3-16

3.12. The process flow diagram for the *spatial analysis* step. 3-18

3.13. The *road finding algorithm* pseudocode. 3-26

3.14. Visualization of the *road finding algorithm*. 3-27

3.15. The process flow diagram for the *finalizing the GIS layers* step. 3-31

4.1. Comparison of the satellite image and the synthetic scene. . . . 4-2

4.2. Visualization of the converged SOM showing density and "fences". 4-5

4.3. Visualization of NDVI applied to the converged SOM. 4-8

4.4. Plot of ordered NDVI values and NDVI thresholds. 4-9

4.5. The "vegetation" *SOM segmentations*. 4-10

4.6. Plot of ordered NIR scatter values and threshold. 4-12

4.7. The "tree" *SOM segmentations*. 4-14

4.8. The "non-tree vegetation" *SOM segmentations*. 4-15

4.9. The "rooftop" *SOM segmentation*. 4-17

4.10. The "roadway" *SOM segmentation*. 4-18

4.11. The six *initial object maps*. 4-20

4.12. The "vegetation" *final object map*. 4-21

4.13. Visualization of *morphological operations* on the "tree" *object map*. 4-23

4.14. The "tree" *final object map*. 4-24

Figure Page

4.15. Visualization of *iterative morphological operations* to create the intermediate "fields" *object map*. 4-25

4.16. The "fields" *final object map*. 4-26

4.17. Visualization of *iterative morphological operations* to create the "parking areas" *final object map*. 4-27

4.18. The "parking areas" *final object map*. 4-28

4.19. Visualization of the "buildings" *negative image mask*. 4-29

4.20. The intermediate "buildings" *object map*. 4-30

4.21. The "major buildings" *final object map*. 4-30

4.22. The "best known good roadway" *object map*. 4-31

4.23. Visualization of the *negative image mask* used in the *road finding algorithm*. 4-33

4.24. Visualization of the *road finding algorithm*. 4-34

4.25. The "road network" *final object map*. 4-35

4.26. Six "road network" *final object map* discrepancies. 4-36

4.27. Examples of *conservative* and *aggressive* road finding. 4-37

4.28. The "non-tree vegetation" *final object map*. 4-39

4.29. The "buildings" *final object map*. 4-39

4.30. Final GIS "vegetation" layer. 4-44

4.31. Final GIS "tree" layer. 4-45

4.32. Final GIS "grass" layer. 4-46

4.33. Final GIS "fields" layer. 4-47

4.34. Final GIS "buildings" layer. 4-48

4.35. Final GIS "major buildings" layer. 4-49

4.36. Final GIS "road network" layer. 4-50

4.37. Final GIS "parking lots" layer. 4-51

4.38. Examples of overlaying the final GIS layers. 4-52

5.1. Destination prediction using GIS road network and buildings. . 5-2

5.2. Target prediction with shadows in SAR imagery. 5-3

List of Tables

Table Page

2.1. Listing of NDVI thresholds. 2-22

3.1. Logical "deletion". 3-10

4.1. Listing of the SOM training parameters. 4-6
4.2. Listing of the "vegetation" *temporary object map* accuracies. . . 4-11
4.3. Listing of the "tree" *SOM segmentation* accuracies. 4-13
4.4. Listing of ℓ_2–norm thresholds for the "rooftop" and "roadway" *SOM segmentations*. 4-17
4.5. Listing of the *initial object map* accuracies. 4-19
4.6. Listing of the "tree" *object map* accuracies after morphological operations. 4-22
4.7. Listing of *finalized GIS layer* accuracies. 4-40

List of Abbreviations

Abbreviation		Page
GIS	Geographic Information Systems	iv
IPB	Intelligence Preparation of the Battlespace	iv
SOM	Self Organizing Map	iv
USAF	United States Air Force	1-1
IPB	Intelligence Preparation of the Battlespace	1-1
AFDD-1	Air Force Doctrine Document 1	1-1
GIS	Geographic Information Systems	1-1
VNIR	Visible and Near Infrared	1-2
LOS	Line of Sight	1-3
RGB	Red Green Blue	1-6
GPS	Global Positioning System	1-8
IPB	Intelligence Preparation of the Battlespace	1-8
RGB	Red Green Blue	2-1
SOM	Self-Organizing Map	2-11
PE	Processing Element	2-12
NDVI	Normalized Difference Vegetation Index	2-19
NIR	Near Infrared	2-21
NOAA	National Oceanic and Atmospheric Administration	2-21
NEWA	Non Equal Weighted Accuracy	2-31
GIS	Geographic Information Systems	3-1
SOM	Self-Organizing Map	3-1
PE	Processing Element	3-1
NDVI	Normalized Difference Vegetative Index	3-4
NIR	Near Infrared	3-4
SE	Structuring Element	3-20

Abbreviation		Page
NIM	Negative Image Mask	3-22
GIS	Geographic Information Systems	4-1
DIRSIG	Digital Imaging and Remote Sensing Image Generation	4-1
RIT	Rochester Institute of Technology	4-1
NASA	National Aeronautics and Space Administration	4-2
JPL	Jet Propulsion Laboratory	4-2
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer	4-2
SOM	Self-Organizing Map	4-4
NDVI	Normalized Difference Vegetation Index	4-4
NIR	Near Infared	4-4
PEs	Processing Elements	4-4
PLSOM	Parameter-Less Self-Organizing Map	4-4
NEWA	Non-Equal Weighted Accuracy	4-9
SOM	Self-Organizing Map	5-1
GIS	Geographic Information Systems	5-1
SAR	Synaptic Aperture Radar	5-1
PLSOM	Parameter-Less SOM	5-4
UAS	Unmanned Aerial System	5-4
DIRSIG	Digital Imaging and Remote Sensing Image Generation	5-4
GIS	Geographic Information Systems	VITA-1
IPB	Intelligence Preparation of the Battlespace	VITA-1
SOM	Self Organizing Map	VITA-1

ABSTRACTING GIS LAYERS FROM HYPERSPECTRAL IMAGERY

I. Introduction

At the dawn of the twenty-first century, the United States Air Force (USAF) is engaged in tactical urban warfare. According to the 2006 publication of the National Strategy for Combating Terrorism, "America is at war with a transnational terrorist movement fueled by a radical ideology of hatred, oppression, and murder." [28] Since the invasion of Iraq in 2003, the intervening years have been marked by house-to-house fighting and street clashes with terrorists and militants. Intelligence Preparation of the Battlespace (IPB) for this asymmetric style of warfare presents several challenges including providing current maps of the *area of operations*, continuously updating those maps to reflect new information and changes, and abstracting usable information from the maps. Managing the battlespace more effectively means possessing the most timely information and using technological means to increase situational awareness. Air Force Doctrine Document 1 (AFDD-1) states that as "... a leader in the military application of air, space, and intelligence, surveillance, and reconnaissance technology, the Air Force is committed to innovation to guide, research, development, and fielding of unsurpassed capabilities." [50] The work in this thesis directly enhances four USAF core competencies (see Figure 1.1).

This chapter provides a brief history of remote sensing, discusses hyperspectral image technology, Geographic Information Systems (GIS), spectral and spatial analysis, and the role of this thesis in using hyperspectral imagery to produce GIS layers.

1.1 A Brief History of Remote Sensing

The first black and white imagery of the earth's surface was made in 1858 [9]; but, it wasn't until 1972 that the first satellite dedicated to *remote sensing* of the

Figure 1.1: The work in this thesis enhances four USAF core competencies.

earth's surface was launched [9]. A series of Landsat satellites expanded the portion of the spectrum that is collected, enabling computer processing of images. Since the 1990's, *multi-spectral* remote *geosensing* satellites collect as many as ten wavelengths in the Visible and Near Infrared (VNIR) spectrum that are targeted at *geosensing* of the earth's natural environments. Geosensing satellites enable the automated classification and labeling of natural *materials*, e.g. vegetation, soils, clay, and other spectrally distinct materials. The natural environment consists of large areas of agriculture, forest, desert, and other uniform regions. To sense these large areas, geosensing satellites can have a resolution of tens of meters per pixel [9]. This sensor resolution precludes detailed analysis of "cluttered" urban scenes. Thus, the limited spectral content and sensor resolution of the multi-spectral images prevent the correct classification of most materials used in the construction of an urban city.

Hyperspectral imagers collect hundreds of wavelengths uniformly across the VNIR spectrum, and have a sensor resolution of meters per pixel. The spectral fidelity enables the classification of a broad range of natural and man-made materials. A $1m$/pixel sensor resolution allows detection of urban objects e.g. vehicles, houses, and trees. Hyperspectral imagers can be flown on board many types of air platforms [44] or on a satellite. Today, nations such as Russia, China, Japan, and India are launching satellites capable of sub–1m/pixel sensor resolution [36] and collect enough *spectral channels* to be considered hyperspectral imagers.

1.2 Hyperspectral Imagery

Hyperspectral imaging technology grew from remote and geosensing roots. With minimal adaptations, it can be weaponized for military applications. One of the primary military applications is imaging the urban environment. Hyperspectral imagery augments, rather than replaces, current chromatic imagers.

1.2.1 Weaponizing Hyperspectral Imagery. "Today, the Air Force is the major operator of sophisticated air- and space-based intelligence, surveillance, and reconnaissance systems and is the Service most able to quickly respond to the information they provide." [50] Hyperspectral imagers are able to collect hundreds of channels in the VNIR region of the spectrum. By collecting such a broad of the spectrum, similar materials be distinguished. Distinguishing natural materials, e.g. clay and vegetation types, has been the subject of many papers (see [17], [42], [42]) since the launch of the first remote-sensing satellite more than thirty years ago. What is new is the increasing resolution of the imagers. Where it would previously take acres of grassland to register a single pixel on the satellite image, today's imagers can register an automobile on multiple imager pixels. Recent remotely sensed hyperspectral images can be used to distinguish lawns, houses, cars, roads, and sidewalks – the features of interest in an urban environment.

Weaponizing hyperspectral imagers involves adapting a commercially available imager for use with a military air platform (e.g. a UAS) or using available satellite imagers. Hyperspectral images can be several hundred megabytes to gigabytes in size, and a Line of Sight (LOS) high-bandwidth data connection enables off-board processing of the data. There are two basic requirements for using a hyperspectral imager in military applications. First, the need to sense urban environments and objects found in the urban environment necessitates sub–$1m$/pixel sensor resolution. Second, the imager needs to collect hundreds of channels in the VNIR to enable automated exploitation of the image. Hyperspectral imagers that meet these conditions are commercially available [44].

One scenario for weaponizing hyperspectral imagery involves flying the imager, which weighs tens of pounds [44], on board a medium altitude UASs, illustrated in Figure 1.2. A medium altitude UAS, including the RQ-4 Global Hawk, has a $2,000lb$ payload, thirty-one hour loiter time, and a high bandwidth line of sight data link [32]. Flying a hyperspectral imager on board a medium altitude UAS provides near real-time persistent surveillance over an area of interest. The technology developed in this thesis could be used for continuous imaging of an area of interest and provide current GIS layer information. Other on-going research seeks to uniquely identify automobiles, track dismounts, monitor foot traffic patterns, and provide additional close-in intelligence to the battlespace commander. “Surveillance assets are now essential to national and theater defense and to the security of air, space, subsurface, and surface forces.” [50]

1.2.2 Collecting in the Urban Environment.

The urban environment is a worst-case scenario for putting together a cohesive, understandable scene abstraction. One reason for this is the presence of spectrally indistinct materials that makes spectral analysis problematic. Two such materials are asphalt shingles on rooftops and asphalt roadways. In [30], the authors concluded that urban streets and rooftops cannot be reliably distinguished without feature shape or context information. This highlights the need to exploit the spectral-spatial characteristics of a hyperspectral image.

Another challenge to spectral and spatial detection occurs when roadways and rooftops are partially or fully occluded by tree or other foliage, as pictured in Figure 1.3. The variety of materials and their orientation in the man-made environment leads to a large number of mixed pixels, which do not fare well in a winner-takes-all

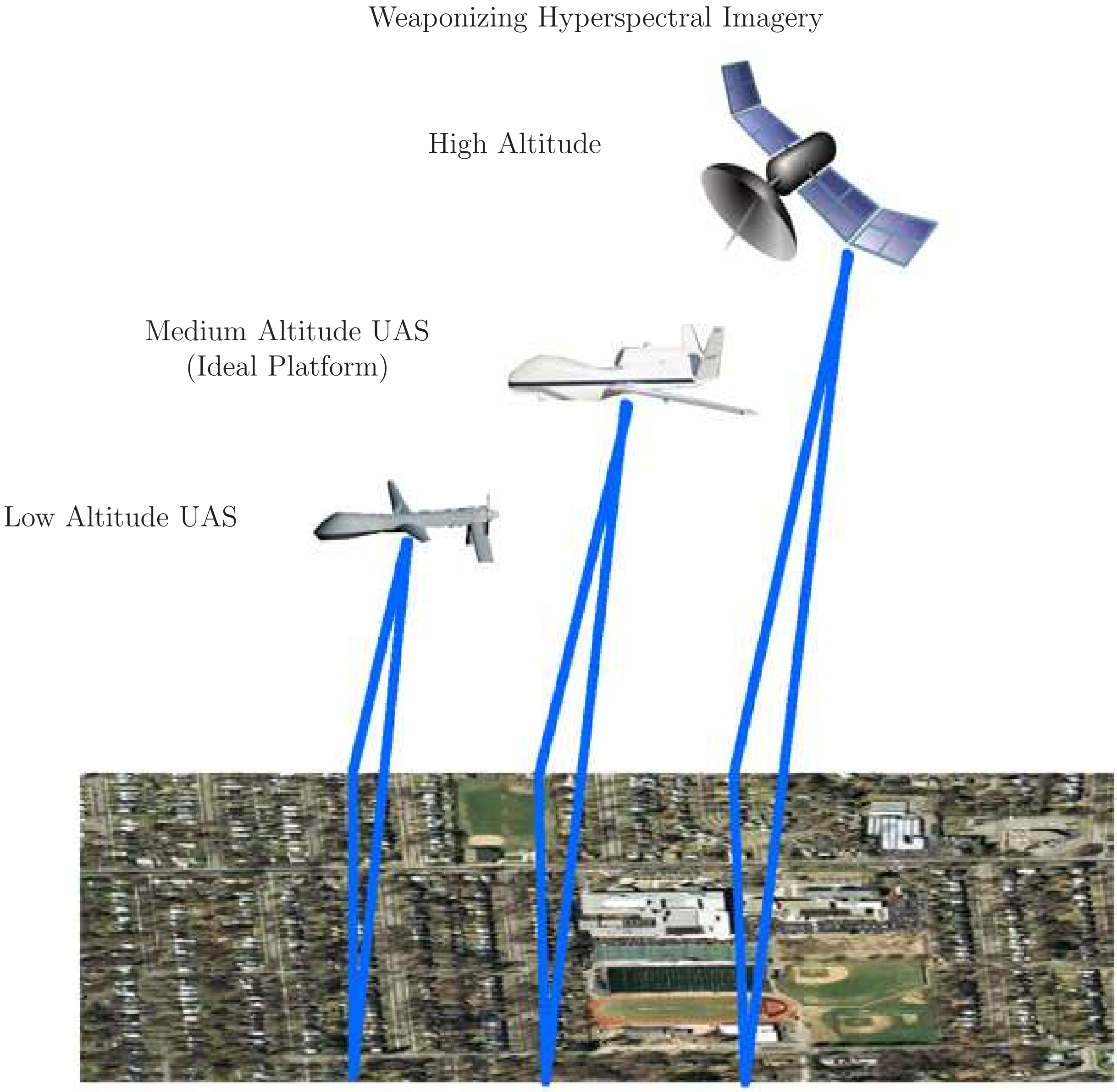


Figure 1.2: Possible method to weaponize hyperspectral imagery using a medium altitude UAS (middle picture) for persistent surveillance and sub-meter resolution. Hyperspectral imagers collect images line-by-line in a "pushbroom" mode (blue line).

approach to classification. Due to the natural growth and constant change in urban environments, consistent and rapid scene abstraction methods are needed [36].

1.2.3 Augmenting Chromatic Imagery with Hyperspectral Imagery. A *chromatic imager* collects three channels: red, green, and blue (RGB). *Multispectral imagers* collect more than three channels, but usually less than ten. A hyperspectral imager collects hundreds of channels across the visible spectrum (400nm to 700nm) and the near infrared spectrum (700 to 2400nm). Hyperspectral images require significantly more bandwidth and disk storage than RGB images. Most hyperspectral-capable satellites also collect RGB imagery.

RGB images, an example of which is pictured in Figure 1.3, are three channel (or three-color) images that cannot be effectively or reliably processed into GIS layers with automated or semi-automated means. Spatial processing of the image to identify features does not translate into identifying specific shapes such as buildings, roads, ponds, or other objects. The plurality of similar color information makes segmentation of the image by color difficult. This necessitates using hyperspectral imagery to automate scene segmentation and GIS layer creation.

1.3 Geographic Information Systems (GIS)

GISs provide simple outline or polygon identification of scene features in separately labeled GIS layers that can be viewed individually. GIS layers can be an excellent resource for intelligence preparation of the battlespace (IPB); however, automating GIS layer abstraction requires hyperspectral imagery.

1.3.1 GIS Layers. One common way to abstract usable information is with GIS layers. Two examples of this are the online Google Maps$^{\text{TM}}$ and Mapquest$^{\text{TM}}$ programs. GIS layers for buildings, major buildings, roadways, and vegetation can be layered over terrain, topographic, or satellite image data. The availability of this type of GIS information is foundational to current research into future technologies. For

RGB Chromatic Image

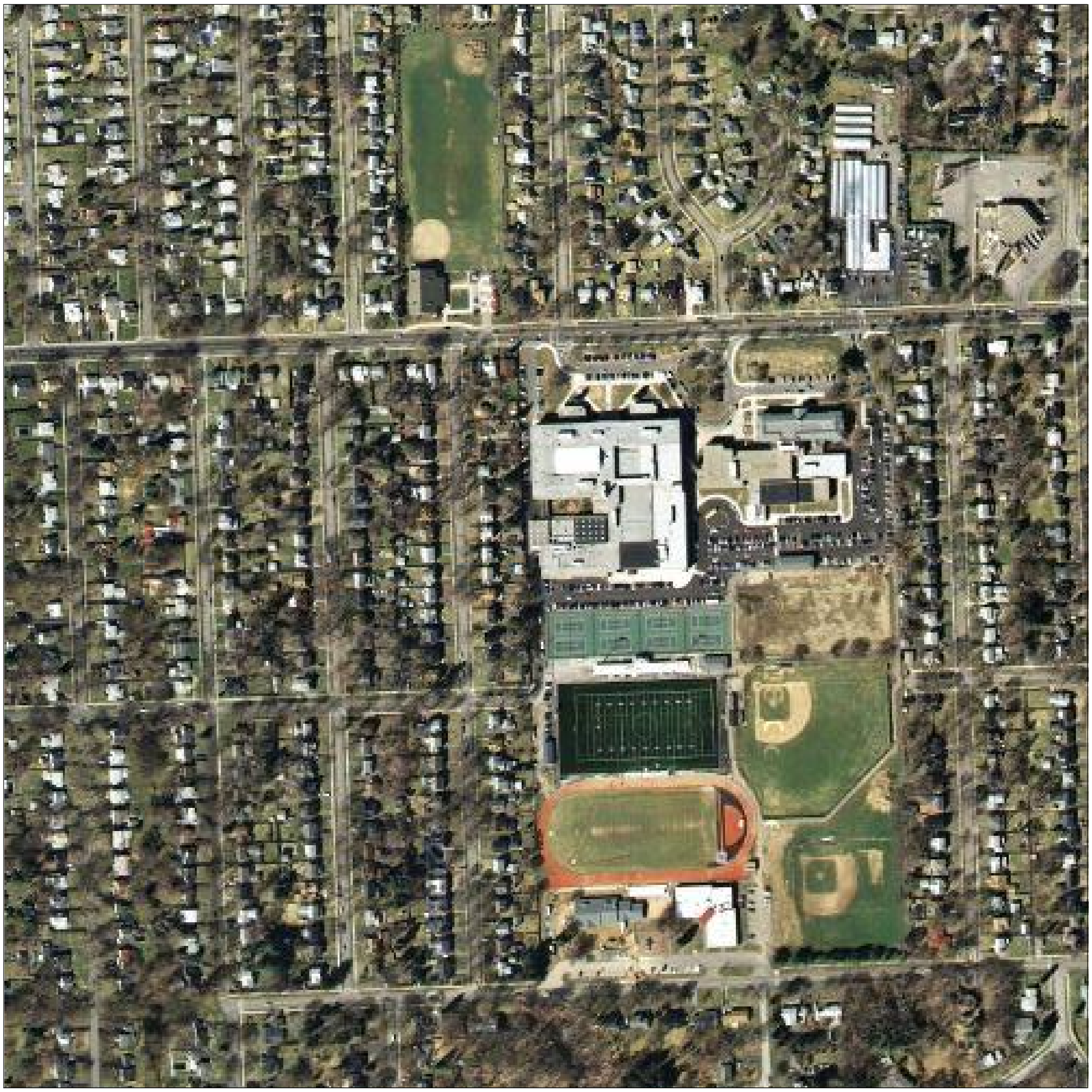


Figure 1.3: It is extremely difficult to reliably segment this RGB chromatic scene with a computer algorithm. It requires detailed manual processing of the image.

example, the work in [31] develops a target behavior prediction program based on a GIS representation of an urban scene, shown in Figure 5.1. In [12], the research depends on scene segmentation to drive the false alarm rate from 0.7 to 0.0, illustrated in Figure 5.2.

GIS road network maps are created through a labor-intensive process that can include manually marking road segments while viewing RGB images to using individuals driving vehicles around with Global Positioning System (GPS) units that create a digital representation of the road network. The street mapping project [1] uses the latter method. The maps are then *orthorectified* to longitude and lattitude, registered with information gathered by other driving teams, assembled, hand-matched to existing satellite terrain maps, coded into vector maps and published. GIS layers are used to outline buildings and other features, usually with generic polygon shapes.

This solution, developed by many commercial companies, does not satisfy the needs of the Combatant Commander in a hostile environment. The use of on-ground teams to acquire road network information and the time required to assemble high-quality digital maps hinders the effectiveness of the kill chain.

1.3.2 GIS Layers and Intelligence Preparation of the Battlespace.

Intelligence Preparation of the Battlespace (IPB) and battle management are organized around maps. Viewing high-resolution "cluttered" maps of urban scenes is a time-intensive process, and battle maps are best abstracted into separate layers that provide access to a single type of information at one time. GIS layers are ideally suited to this type of battle management by allowing one or more features to be considered at a time. The creation of GIS layers for a hostile environment is difficult and labor-intensive. Hyperspectral imagery allows the automation of the task.

1.3.3 Hyperspectral Images and GIS Layers.

Hyperspectral imagers with automated image processing techniques are the gateway to the future of surveillance because they acquire enough information to distinguish surface features. This thesis approaches the conceptual topic of creating abstracted, usable information from these images in an automated way. This information is then presented to the user in a format, the GIS layer, that is both familiar (as in Google Maps$^{\text{TM}}$) and actionable. GIS layers are *vector maps* that have the advantage of being highly compressible and can reduce bandwidth requirements by several orders of magnitude. One experiment performed by the this author compressed two gigaybes of vector graphic animations into less than twenty megabytes. This thesis demonstrates both the spectral analysis of a sample hyperspectral scene and the categorization into multiple, separate, GIS layers where each layer is labeled appropriately and can be overlaid over the original image.

By processing hyperspectral images in near real-time, it would be possible to provide continuously-updated digital maps that reflect minutes to hours-old information of the current ground configuration including navigable roads, waterways, obstructions, and complete and destroyed structures such as buildings. “The art of intelligence is rapidly turning information gathered through surveillance and reconnaissance into an accurate, predictive, and actionable format that can be used to assist planning, execution, and evaluation of air and space operations.” [50] The next critical step in converting imagery to maps is the process of escalating from feature abstraction to concept abstraction. Today, stitching together groups of pixels into buildings, roads, water bodies, trees, parks, fields, and other conceptual objects is completed through an almost entirely manual process using high-resolution chromatic imagery, driving cars around with GPS units to locate roads, and comparing results with old, out-of-date maps. This is a non-stop, on-going, manual process that is not suitable for real-time urban warfare because it yeilds maps that are months to years old. This is where the latest hyperspectral imagers provide new capabilities and possibilities.

1.4 Spectral and Spatial Analysis

The vast majority of authors choose either spectral or spatial analysis. In machine-learning, optimizing spectral analysis algorithms is the preferred approach,

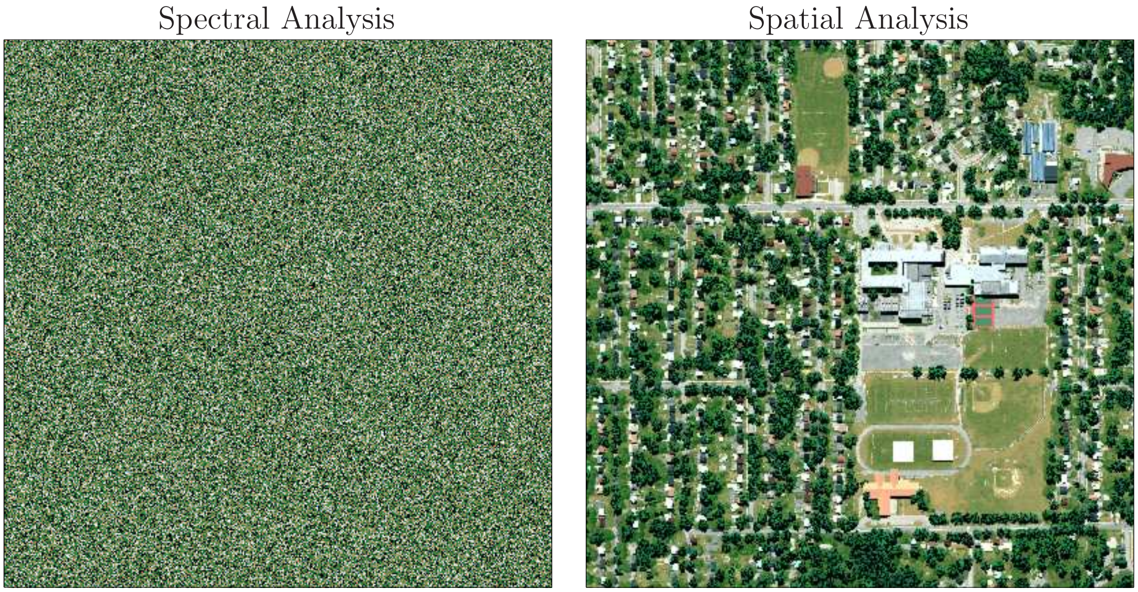


Figure 1.4: Machine learning optimizes spectral analysis by selecting sample spectra in random order (left image). The spatial information of a scene (right image), however, contains valuable information. This thesis exploits the spectral-spatial relationship.

considering the tremendous difficulties in the arena of spatial image processing. The spatial arrangement of the pixels is how the human brain interprets these abstract concepts, but computers are rarely tasked to do this kind of interpretation because of the extreme unreliability of automated spatial image processing algorithms. Simply put, spatial processing of images is tremendously difficult. What a human can discern from just a quick glance at an image, computers consistently fail at conceptualizing. Figure 1.4 illustrates these opposing concepts.

When spatial processing is used in conjunction with spectral analysis, it is not necessary to maximize the classification accuracy of the hyperspectral image. Spatial analysis can significantly improve the resulting accuracies. However, relying solely on accuracy as a measure of maps quality is unreliable. For example, a GIS layer may indicate the existence of a road where an RGB image shows only tree foliage.

1.5 Research Goals

The goal of this thesis is to automate the creation of GIS layer shape files by exploiting the spectral-spatial relationship of a *hyperspectral image* cube. It is assumed that an *a priori* endmember spectra, a spectral database, or specific scene knowledge is *not* available. This work furthermore seeks to connect the road network under significantly tree-occluded roadways. The design will be modular and allow for future extensions that increase the level of automation, robustness, resistance to error, and quantity and quality of the results.

This process is accomplished in two major phases: spectral analysis and spatial analysis. A hyperspectral image cube input is spectrally processed with a self-organizing map (SOM), which is then segmented into spectral regions corresponding to materials of interest. The *SOM segmentations* are then used to create binary-valued spatial maps that show the possible locations of the materials of interest. These *material maps* are processed and used in various combinations to produce the final GIS layers.

1.6 Organization

This thesis contains five chapters. Chapter I provides an introduction to hyperspectral images, the military uses of the technology, and what results to expect from the technology developed in this thesis.

Chapter II surveys the literature focusing on scene segmentation, spectral analysis, and spatial analysis. There are several foundational texts that contributed to the success of this paper. There are several articles published in the last few years that show a trend of expanding on spectral analysis with additional spatial analysis as done in this thesis. Going the extra step of processing an image spatially after it has been processed spectrally is seen as a way to improve both spectral analysis as well as scene segmentation. Some constituent technologies, e.g. morphological operations and self-organizing maps, are detailed.

Chapter III discusses process development of this thesis. A step-by-step presentation of the process, from the initial spectral analysis to the multiple levels of spatial analysis accomplished to produce the final products, is provided in detail.

Chapter IV presents the experimental results. This includes the multiple intervening steps leading to the final products. The steps needed to create the final GIS layers and segmented scene involve a mix of bottom-up and top-down approaches to analyzing the spatial scene. The foundational step is to identify what is known about the scene, then use that information to iteratively infer knowledge of the scene. By accounting for spatial arrangement of features, the inferred knowledge is refined and solidified, and then used again to infer additional knowledge. High level features, e.g. "fields", are abstracted from the spatial relationship of pixel-level features, e.g. grass pixels, and this knowledge becomes part of the refinement process.

Chapter V analyzes the results of the thesis and the final GIS layers are presented, visually, as GIS overlay products. It also includes future work and concluding remarks. This thesis is a proof-of-concept approach for a novel method of mixing spectral and spatial analysis.

II. Background Information

This chapter provides the background knowledge, including concept development and definitions, to support an exploration of hyperspectral image analysis. It examines the approaches used to address the problem, classify hyperspectral data, segment scenes, and provides a survey of recent literature encompassing the state-of-the-art.

Several examples from current literature demonstrate approaches to scene segmentation. Fewer articles deal explicitly with using hyperspectral data. This is likely due to the relatively recent interest in and the availability of hyperspectral imagery. Given these innovations, resources are being devoted to developing algorithms that are tuned to take advantage of the newer hyperspectral images and the expected future availability of such images.

2.1 Hyperspectral Images

Hyperspectral images are conceptually different from Red-Green-Blue (RGB) chromatic images by the number of channels that are collected. Where RGB images contain only three-channels, hyperspectral images can collect hundreds of channels. Before the development of hyperspectral imagers, multi-spectral imagers were available that typically collected the RGB channels plus an additional two to seven channels (channels are usually referred to by their wavelength).

Figure 2.1 is an example of a hyperspectral image. Because the data is presented with spatial dimensions intact and the channels as the third dimension, it can be referred to as a data cube.

2.1.1 Collecting Hyperspectral Images. The device used to collect hyperspectral images is an *Imaging Spectrometer* [9], but is referred to as a *hyperspectral imager*. Modern commercial versions often consist of three separate collectors. There is one for collecting bands in the visible spectrum and two for the near infrared. Ac-

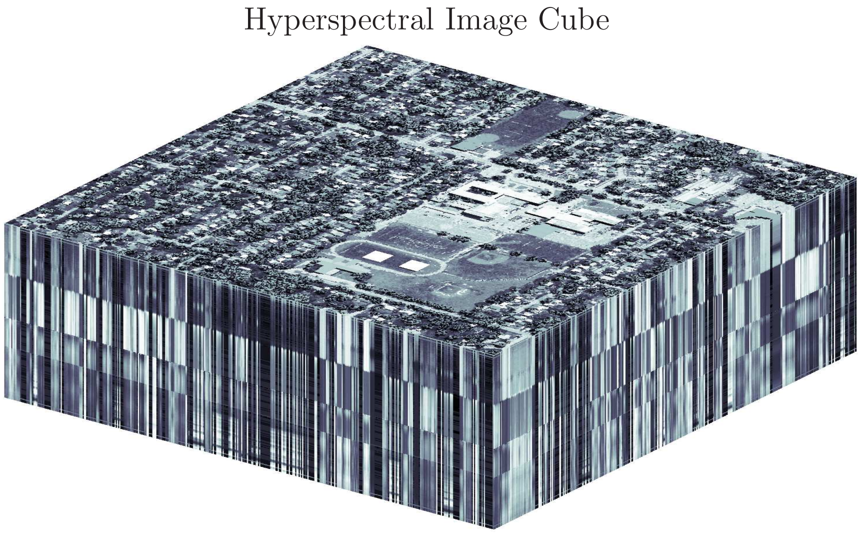
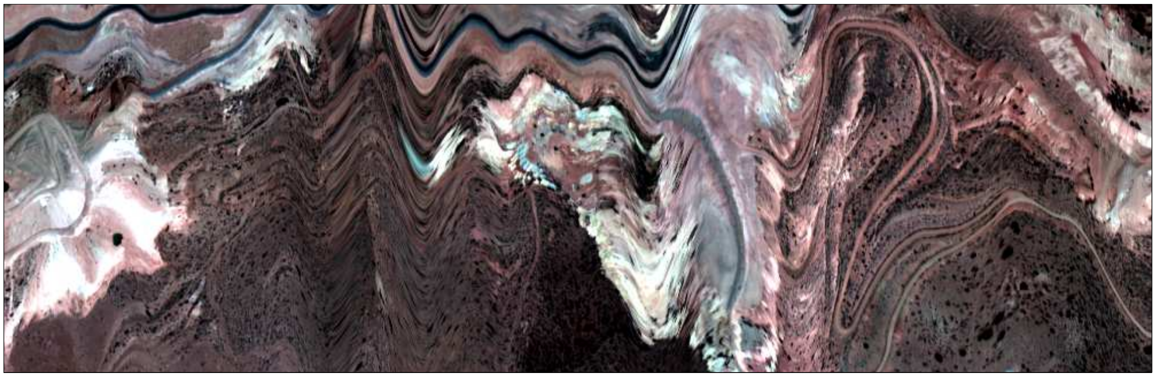


Figure 2.1: Example of a hyperspectral image cube. The ultraviolet band corresponding to wavelength $\lambda = 0.3975$ is seen on top, with each of the 194 bands below it.

cording to [43], the range is typically “at least 0.4 to 2.4 micrometers (visible through middle infrared wavelength ranges).”

One of the major advantages to hyperspectral imagers in a military context is that the cameras collect reflected light. Unlike radar, lidar, and other sensors, hyperspectral imagers do not radiate energy. This reduces the likelihood that the imager can be identified or tracked while it is operating. The hyperspectral camera can, in this way, be operational the entire time it is over the target area rather than operating in a detection-avoiding burst-mode.

The camera typically operates in a *push-broom* manner, simulated in Figure 1.2. “In many digital imagers, sequential measurements of small areas are made in a consistent geometric pattern as the imager platform moves and subsequent processing is required to assemble them into an image.” [43] The imager collects a single line at a time (see Figure 1.2, blue line), where the resolution of the line depends on the number of pixels the imager collects. These lines are oversampled and then processed into a continuous hyperspectral scene. Multiple strips are then stitched together to



An Uncorrected “Wavy” Hyperspectral Image

Figure 2.2: A hyperspectral image before geocorrection. This image is best used for spectral analysis. *Geocorrection* restores the spatial relationships but alters spectral information.

create a coverage scene. If the aircraft veers off-course, the variation in the flight path is recorded and the imagery is later corrected for the flightpath deviations. An example of an uncorrected image is presented in Figure 2.2. After the radiance data upwelling from the earth is collected, it is processed into radiances.

2.1.2 Radiance and Reflectance. Hyperspectral images are collected in radiance. This is an absolute number of photons hitting the sensor in a given wavelength. The number of photons that are returned from the earth’s surface can vary depending on a number of factors, including the amount of sunlight reaching the earth’s surface. Images collected at various times of the day exhibit varying radiances. In [20], the authors attempt to correct for the zenith angle of the sun to calibrate the sensor and adjust the radiance values.

Since hyperspectral images are collected in multiple runs over the target area, the variation in radiance can make it difficult to compare and stitch together multiple hyperspectral images taken over the same target area. The solution is to use a ratio, comparing the amount of incoming light to the amount of light returned to the imager. For a given material, the amount of light it reflects (ν_1) is proportional to the amount of light it receives(ν_2). These two values are used in the computation of the surface reflectance (ρ) in Equation 2.1.

$$\rho = \frac{\nu_1}{\nu_2} \tag{2.1}$$

2.1.3 Model of Light Transmission and Radiance Measurement.

The light source is the sun. The atmosphere absorbs radiation because of water content. Several bands are removed from the hyperspectral image during *atmospheric correction* because of this scattering. Light is then transmitted to the earth's surface, where it illuminates the targets. Some of the radiation absorbed by the atmosphere is re-transmitted to the earth's surface. The earth's surface scatters the majority of the light, but about three percent [36] radiates toward the imager. The atmosphere absorbs some of the radiated energy. The absorbed radiation as well as the light radiated from the earth's surface are collected on the sensor platform. Figure 2.3 demonstrates pictorally how the sun's light is ultimately collected by the hyperspectral imager. The imager can be located in a satellite or on an airborn platform. Before the data can be used, atmospheric correction is performed.

2.1.4 Atmospheric Correction.

The atmosphere plays a quantifiable role in collecting imagery. The atmosphere disrupts the path of radiation as it is reflected from the earth's surface, and post-processing must correct for these effects.

Some of the factors include *atmospheric scattering*, *refraction*, and *absorption*. Scattering is a result of particles (e.g. pollen, water, or dust) in the air between the imager and the source. Refraction is the result of light bending at the junction between two different materials, for example, atmospheric layers. Absorption occurs when energy is strongly attenuated, especially through gases (e.g. ozone), oxygen, carbon dioxide, and water vapor [43].

Reflectance is estimated through one of several methods: *flat field conversion*, *average relative reflectance conversion*, *empirical line method*, and *modeling methods.*

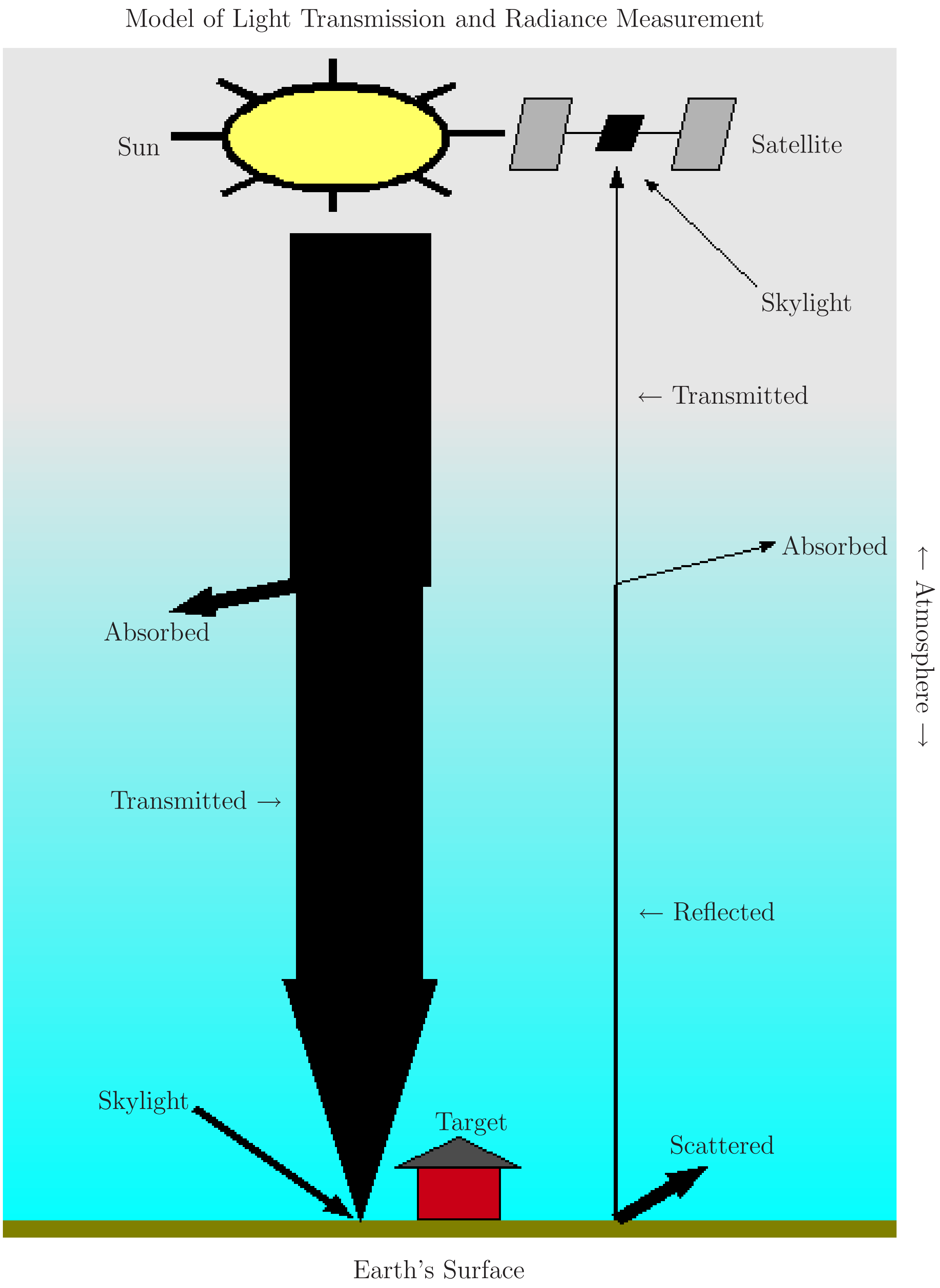


Figure 2.3: Model of light from incidence at the sun to reflectance at the earth's surface. Less than 10% of the sun's light is ultimately collected at the satellite [36].

The empirical line method is commonplace and is used to correct the hyperspectral image used in this thesis. The method uses linear regression to compare two or more known in-scene spectra with the spectra that were actually collected. All of the in-scene spectra are then corrected by an empirically determined gain and offset value. Atmosphere disturbances are both linear and non-linear in nature, but the linear effects dominate. Empirical line correction is therefore the preferred way to correct imagery because of its general acceptance and its ability to remove the dominant atmospheric effects.

$$\text{Reflectance} = \text{Gain} \times \text{Radiance} + \text{Offset} \tag{2.2}$$

2.1.5 Spectral Features.

Spectral features result from the *electronic* or *vibrational* properties of atoms [13]. Electronic features occur when electrons transition between valence shell energy levels. Vibrational features occur from the individual atoms vibrating against the lattice structure in which they are trapped. Different materials have characteristic electronic and vibrational features depending on their valence shells and how they arrange themselves in a lattice structure. A *composite* or *mixed* spectra results with multiple material types combined together [43].

The electronic and vibrational characteristic of atoms at certain frequencies also produce harmonics at multiples of the fundamental frequency. These harmonics lead to a high degree of correlation between the different wavelengths of a spectral signature (correlation measures the strength and direction of the linear relationship between two random variables [3]). Figure 2.4 illustrates correlation among the bands of a hyperspectral image.

2.1.6 Spectral and Spatial Pre-Processing.

Spectral pre-processing is accomplished on-sensor or prior to spectral analysis. The pre-processing steps are unique to each imager design and are assumed to be accomplished before image analysis. The calibration takes into account environmental effects. Some of the effects are angle of

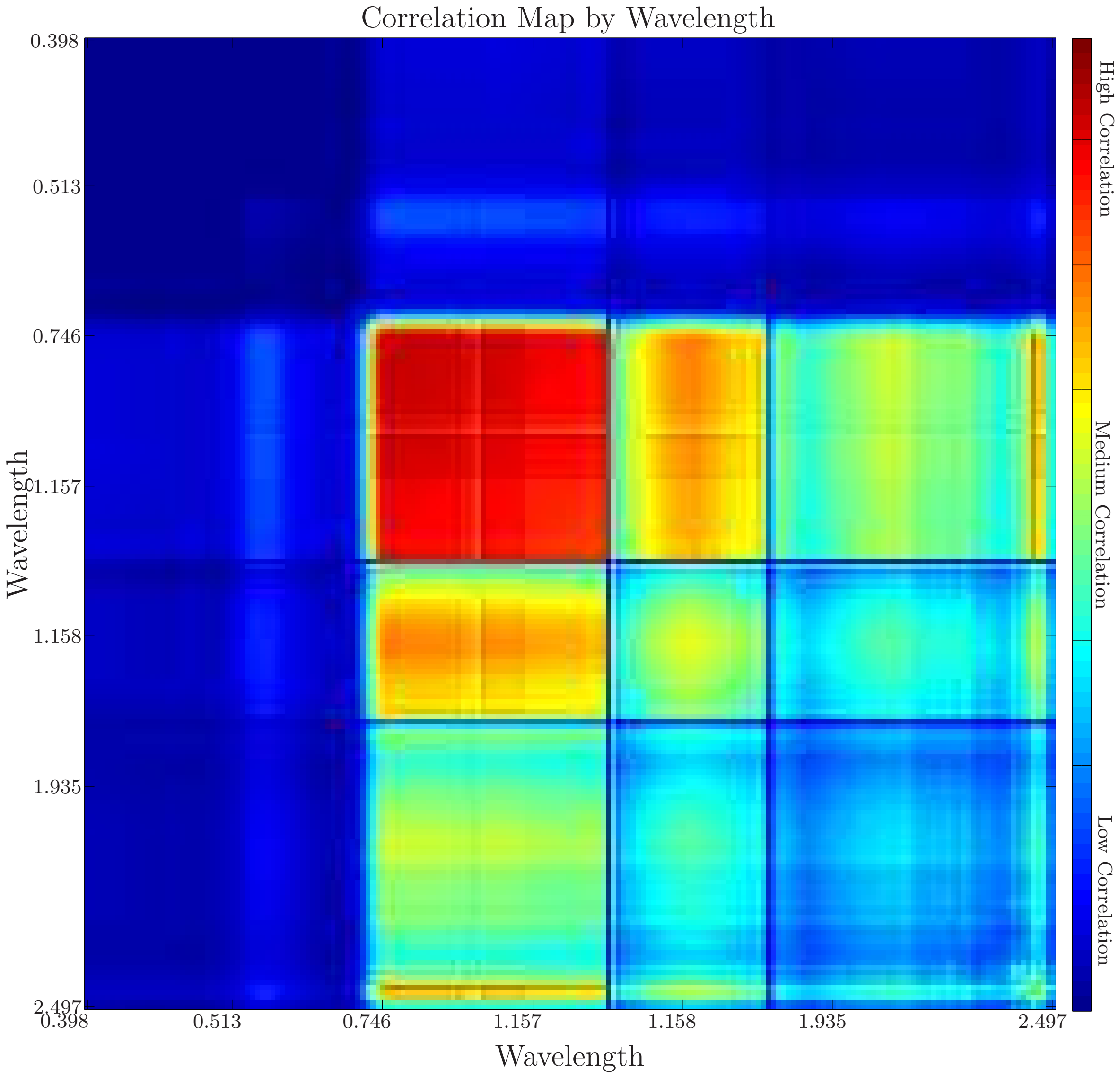


Figure 2.4: Hyperspectral images exhibit a high degree of correlation between the wavelengths because of vibrational and electronic effects.

the sun, viewing angle of the imager, light scattering, secondary illumination from reflected light, shadowing, solar radiation from scattering, and imager noise [42].

An airborn imager (as opposed to a satellite imager) can easily be vectored off course, for example, by a wind gust. These deviations in the flight path are recorded by the *inertial navigation system* and later used to *geocorrect* the image. Uncorrected images look “wavy”, as shown in Figure 2.2. By using the flight path geometry, the wavy image can be “straightened”. This process is called *geocorrecting* the image. Gecorrection restores the spatial relationship of the pixels, but alters spectral information because of the pixel averaging and interpolation that must be accomplished during the geocorrection process. Since images from multiple flight paths are stitched together along the edges, there may be gaps in the information where a second flight path may not overlap the deviation experienced in the previous collection. The geocorrection process may result in interesting spectra that do not actually exist on the ground [38].

In order to maximize the effect of spectral analysis and spatial analysis, spectral analysis and classification can be accomplished on the image before geocorrection. Spatial analysis is best done on the geocorrected image or else many spatial features (roadways, buildings, etc) may appear where no actual feature exists.

2.2 Spectral Analysis

Once a hyperspectral image is available for spectral analysis, it is ready to be classified. *Classification* is the process of assigning a label to an observation [42]. Practically, classification algorithms assign pixels to a given category. Multi-algorithm approaches as in [11] use majority-voting to improve the classification of individual pixels. Most of these algorithms, however, require supervised learning methods. Training samples are manually and laboriously classified. Increasing the number of training samples usually results in improved classification accuracy. In this thesis, the goal is to remove the manual labor element by using unsupervised classification methods which do not need manually classified training samples. Two popular methods considered here are k-Means Clustering [51] and the Self-Organizing Map [22].

2.2.1 k-Means Clustering. k-Means clustering is a general clustering tool that is often applied to spectral analysis. This tool assumes a particular statistical distribution of the data, usually a normal distribution. After random initialization of the means and standard deviations, the ℓ_2–norm is computed and the centroid of the distribution is updated. Through an iterative process of updating centroids, spectra are segmented and assigned to one of the existing k-means.

k-Means clustering is a promising method to segment a scene because it requires minimal user input and requires no training samples to accomplish segmentation. Choosing the number of cluster centers, k, can be a challenging task. Choosing k is in advance and without *a-priori* knowledge of the scene is challenging. If k is chosen too small, areas of the scene can be grouped together even though the materials of interest may be distinct (see Figure 2.5). If k is chosen too large, the segmentation can result in thousands of separate groups of materials and appear meaningless and noisy. A significant difficulty develops when similar materials are segmented differently and later need to be combined into a contiguous region. As noted in [51], "...once the clustering is done, it may be observed that the clustering is not entirely satisfactory in the sense that there may be too fine or too coarse a clustering". The subsequent spatial segmentation of the scene from the spectral clusters is necessary to identify the spatial structure. Due to the number of spatial segments and grouping of distinct materials, spatial analysis is an insurmountable task. An example of these difficulties is illustrated in Figure 2.5. Nonetheless, there have been some attempts to apply k-Means clustering to hyperspectral data with varying degrees of success.

In [29], k-Means clustering is used to separate a hyperspectral image into separate regions. A pixel is assigned to a cluster by minimizing the ℓ_2–norm, and convergence is determined when less than 5% of pixels change mean vectors during each iteration. The authors overcome some of the problems with random initialization by

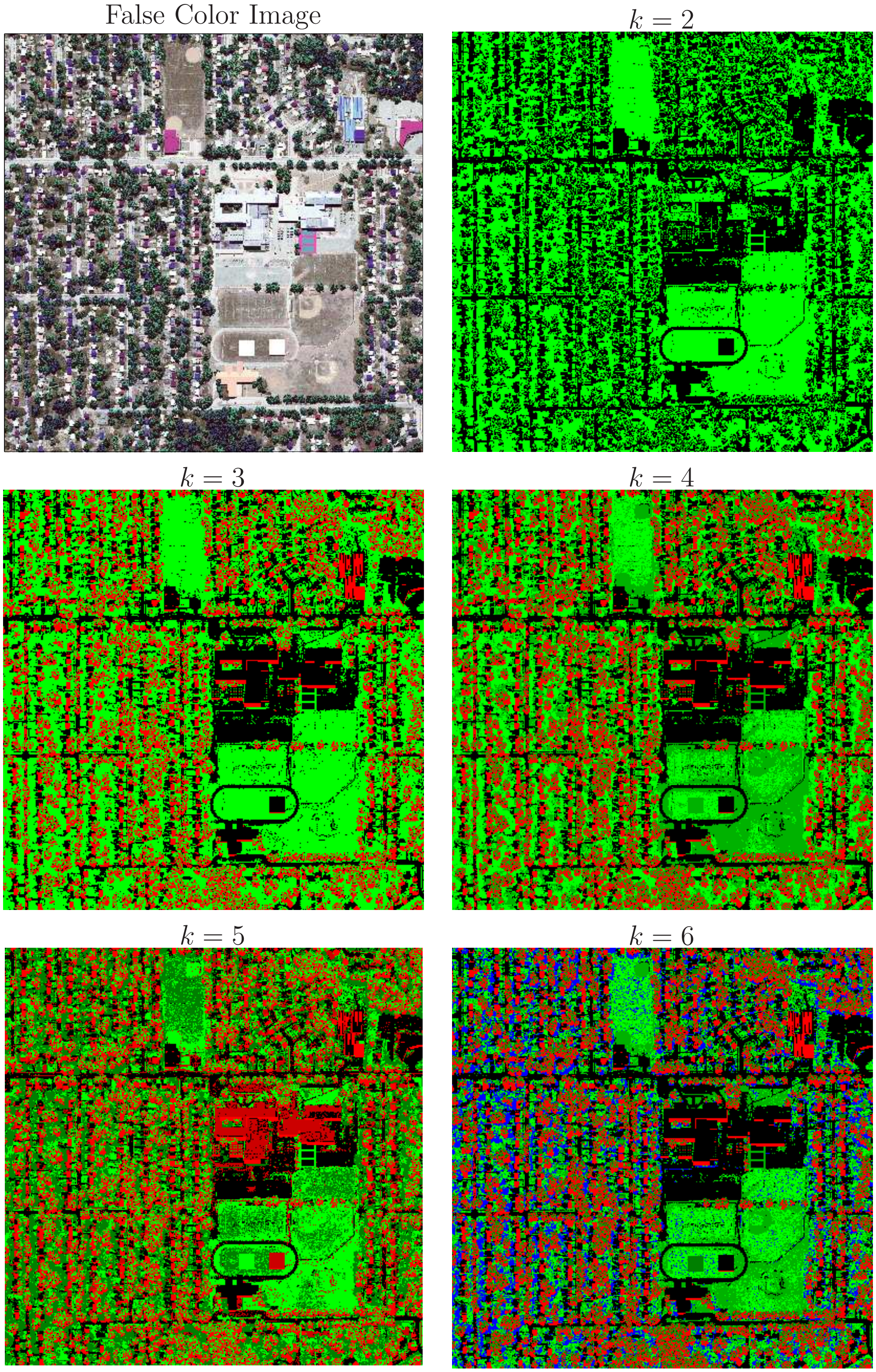
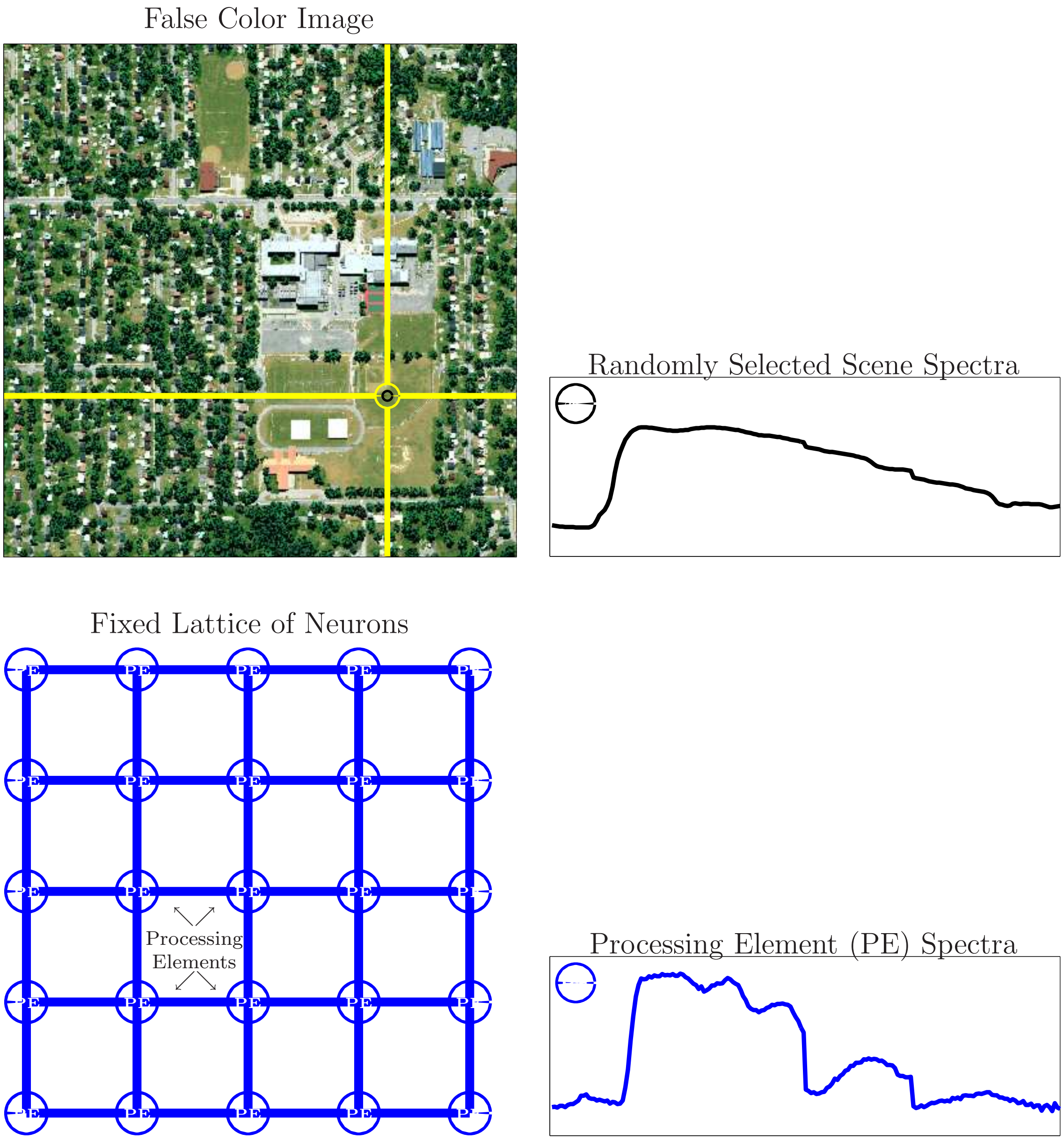


Figure 2.5: Illustrates the difficulties with k-Means clustering. Distinct materials (e.g. houses and trees or buildings and roads) are not being separated. Similar materials (e.g. grass) are not being grouped. Abstracting the spatial structure of this scene is an insurmountable task.

developing an informed initialization algorithm, which improves performance of the clustering; however, the informed initialization only shows improvement in the case where convergence is set at less than 2% of the pixels changing class. Informed initialization also increases the number of regions in the final region map. By varying K from 2 to 16, the image is segmented into 101 to 695 spatial regions.

In [25], the traditional k-means clustering is extended to allow kernel functions to replace the ℓ_2–norm usually used to assign pixels. This method allows pixels to belong to more than one cluster simultaneously. The authors use a Gaussian kernel, the normal clustering sensitivity to outliers is reduced. This work creates the spatial neighborhood of the pixel to improve assignment of pixels. The authors worked with RGB data, offering the theory needed to extend the process to multi and hyperspectral images.

2.2.2 Self-Organizing Maps. The human brain is capable of responding to sensory input by training specific areas of the brain. In the case that part of the brain is damaged, the "...recruitment of cells to different tasks depending on experience is well known" [22]. The self-organizing map (SOM) models this learning behavior by specifying an interconnected set of neurons. When new information comes in, the specific neuron that deals with the new information is identified. The neuron is modified to adapt to the new information, and then the surrounding neurons are modified in the same way, but with lesser impact. As a result of "experiencing" new information, "...a two-dimensional lattice of neurons can create ordered maps of any (metric) high-dimensional signal space, often in such a way that the main dimensions of the map thereby formed correspond to the most prominent features of the input signals" [22]. In response to high-dimensional hyperspectral images, the SOM organizes the spectra by the type of material that generated the spectra. The SOM then becomes a useful tool to cluster materials, making it an ideal choice for segmenting a hyperspectral scene into materials. SOM's have several desireable properties that support using this tool for this thesis.


False Color Image

Randomly Selected Scene Spectra

Fixed Lattice of Neurons

Processing Element (PE) Spectra

Figure 2.6: The scene spectra and the SOM PE's are both n–dimensional.

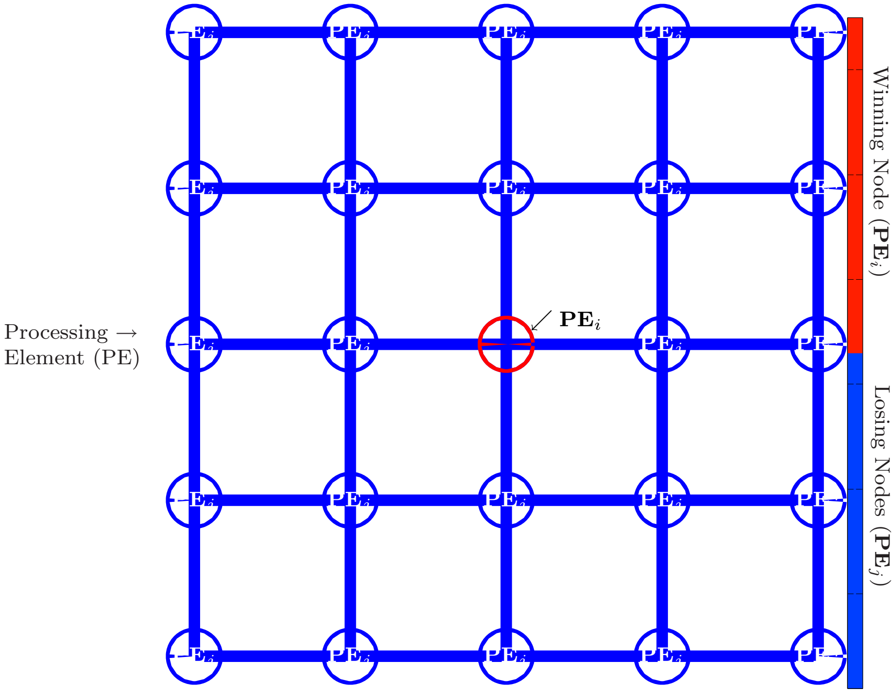
1. There is no user input required to do the spectral analysis. It is a completely unsupervised algorithm.
2. It does not require any training samples that have to be manually labeled by a user.
3. It organizes three-dimensional hyperspectral images into a two-dimensional abstraction of the information.
4. It reduces the abstracted knowledge by one or more orders of magnitude, reducing computation time.
5. The SOM learning method is an averaging process, making the SOM resilient to noise and outliers.
6. The SOM preserves the topology.

The SOM is specified as a fixed lattice of neurons, often a square or hexagonal lattice. Each neuron is also called a *processing element* (PE) that has associated with it a *weight*, w. The PE corresponds in number of dimensions to the hyperspectral image, as shown in Figure 2.6.

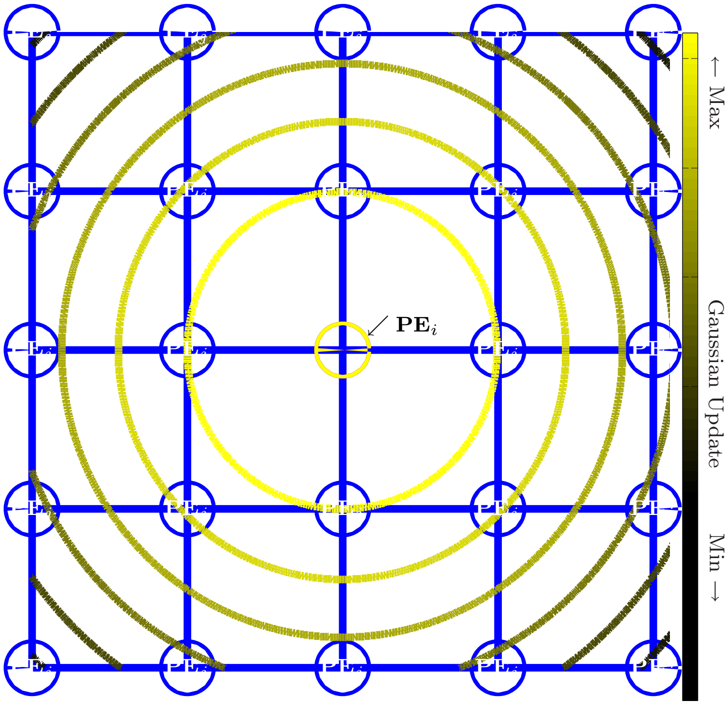
2.2.3 SOM Learning Algorithm. The SOM learns in three phases: *competitive* learning, *cooperative* learning, and *synaptic adaptation* as shown in Figure 2.7. Each of these phases is accomplished in response to an "experience" or input data. An n–dimensional spectra from a hyperspectral image is randomly selected and used as input to the learning process. The three phases of the learning process are iterated until convergence where convergence is usually defined as a fixed number of training steps.

During competitive learning, a "winning" neuron is selected. Each PE receives the same input information, and all of the PE's "compete" to select the winner. Different inputs determine different winners. The metric most often used to determine the winner is the minimum euclidean distance between the input spectra, x_n, and each

1. *Competitive* Learning



2. SOM *Cooperative* Learning



3. *Synaptic Adaptation*

$$w_j(t+1) = w_j(t) + \alpha(t)h_{ij}(t)\,(x_n - w_j)$$

Figure 2.7: SOM competitive learning is winner-takes-all. SOM cooperative learning depends on a Gaussian update rule. Synaptic adaptation updates all of the processing elements.

of the PE's weights, w_k (Equation 2.3). Another way to say this is that the SOM nearest neighbor in the ℓ_2–norm sense to the input vector is the winning PE.

$$\arg\min_k ||x_n - w_k||, \quad k = 1 \dots K \tag{2.3}$$

The cooperative learning phase follows the competitive learning phase. In this phase, the winning neuron activates its neighbors using a *neighborhood function*, $h_{ij}(t)$. One option is a non-normalized Guassian neighborhood function in Equation 2.4. As the variance, σ^2, decreases with time, the winning neuron neighbors are activated at a decreasing rate. Although many neighborhood functions can be specified, the discussion here is limited to the Gaussian neighborhood function because of its use in this thesis.

$$h_{ij}(t) = exp\left\{\frac{-||\text{PE}_i - \text{PE}_j||}{2\sigma^2(t)}\right\} \tag{2.4}$$

The third phase of SOM learning is synaptic adaption. During this phase, all of the PE's are updated. The learning rate, $\alpha(t)$, decreases according to a learning schedule that is specified before the training begins. The synaptic update rule in Equation 2.5 is the same for every PE except for the value of the neighborhood function, h_{ij}. The synaptic update operation averages the new information with the existing weight vector. This averaging makes the SOM resilient to noise and outliers.

$$w_j(t+1) = w_j(t) + \alpha(t)h_{ij}(t)\left(x - w_j\right), \quad i = 1 \dots I, \ j = 1 \dots J \tag{2.5}$$

2.2.4 SOM Learning Algorithm Example. Figure 2.8 is an example of the SOM learning algorithm. Before learning begins, the SOM weight vectors, are randomly initialized. A single input vector, x_n, is randomly selected and the "winning" neuron is identified. During the cooperative learning phase, the winning neuron activates the surrounding neurons according to the Gaussian neighborhood function, and then the synaptic adaptation is performed. After the SOM has learned the input data, the input information, x_{mn}, from data space can be assigned to the SOM in an ordered manner. The ability of the SOM to spatially order the high-dimensional input information is referred to as *topological neighborhood preservation.*

2.2.5 Topological Neighborhood Preservation.

Topological neighborhood perservation is one of the key benefits to using a SOM to organize spectra. According to [22], an "...intriguing result is that the various neurons develop into specific *decoders* or *detectors* of their respective signal domains in the input space." Since the SOM organizes hyperspectral images around their material types, the SOM becomes a detector that organizes materials with a meaningful order. Segmenting the SOM is the same as identifying material types in the original hyperspectral image. The results of Figure 2.7 shows a conceptual idea of how the final SOM may order the data space. If the input data is two-dimensional, then the topology of the SOM can be plotted in two-dimensions, yielding a visual representation of the meaning of preserving topology, illustrated in Figure 2.9. It does not follow that the SOM can label the material types of each PE. An additional spectral analysis step is required. While two-dimensional topology is explored in this thesis, in [15], the authors implement a three-dimensional SOM topology.

2.2.6 SOM in Literature.

The SOM is a very flexible tool that can be used in organizing many different types of data. In [23], the author uses a SOM to improve passive sonar tracking of multiple targets, resulting in an ability to triangulate target positions at a higher rate than traditional bearing information can provide. In [8], the authors apply a SOM to the trading behavior of the Australian Stock Exchange. The SOM is able to distinguish buyer-initiated from seller-initiated trades.

In [26], the authors use a SOM to perform classification on a data set. They are able to achieve an 80% classification rate. Finally, in [47] the authors explore in detail visualizing the SOM topology as it relates to the spatial scene. This work directly supports the approach to segmenting the SOM topology used in this thesis.

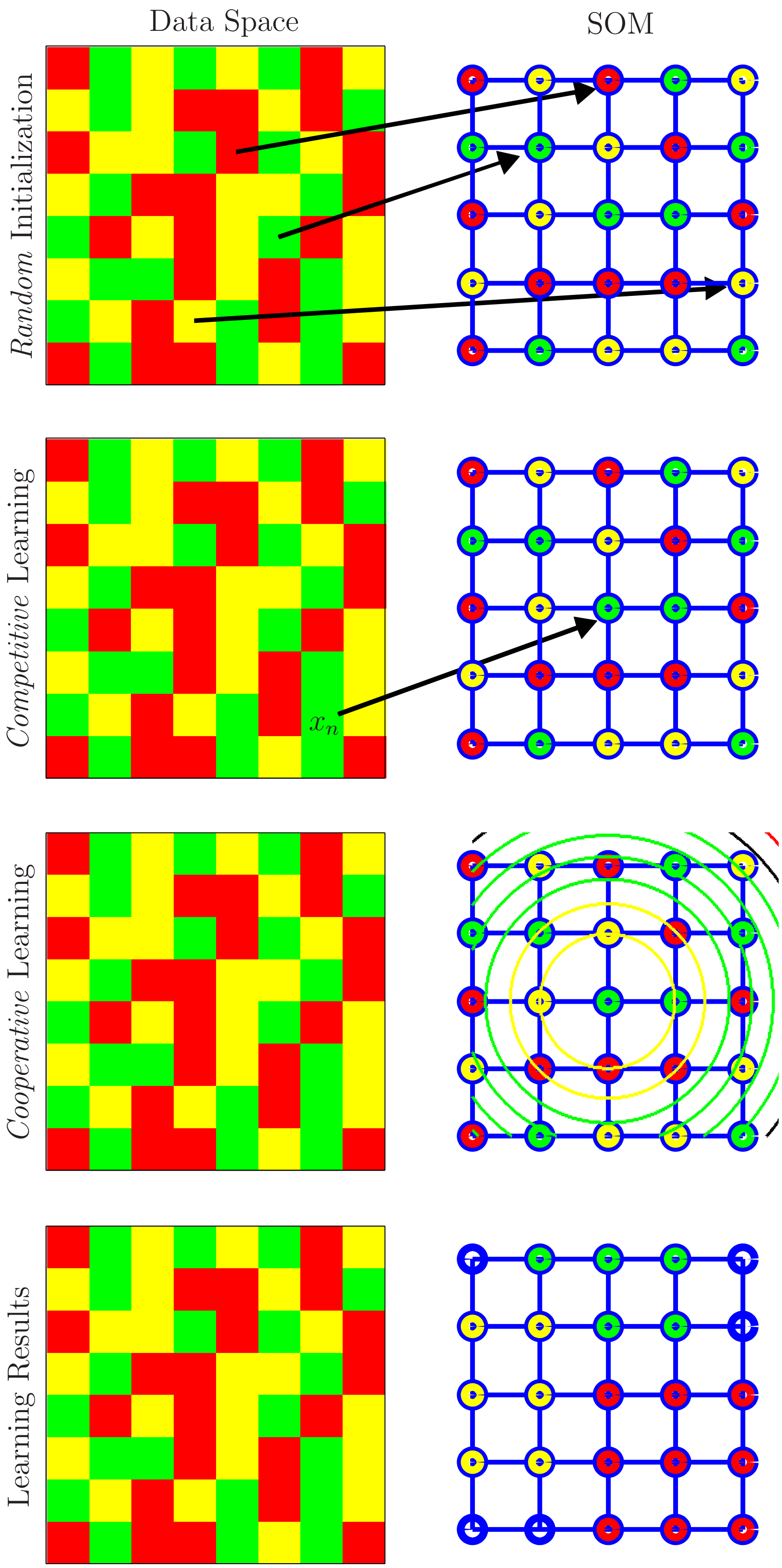


Figure 2.8: Pictorial view of the SOM learning algorithm. The result is a SOM that exhibits *topological preservation* of the high-dimensional data.

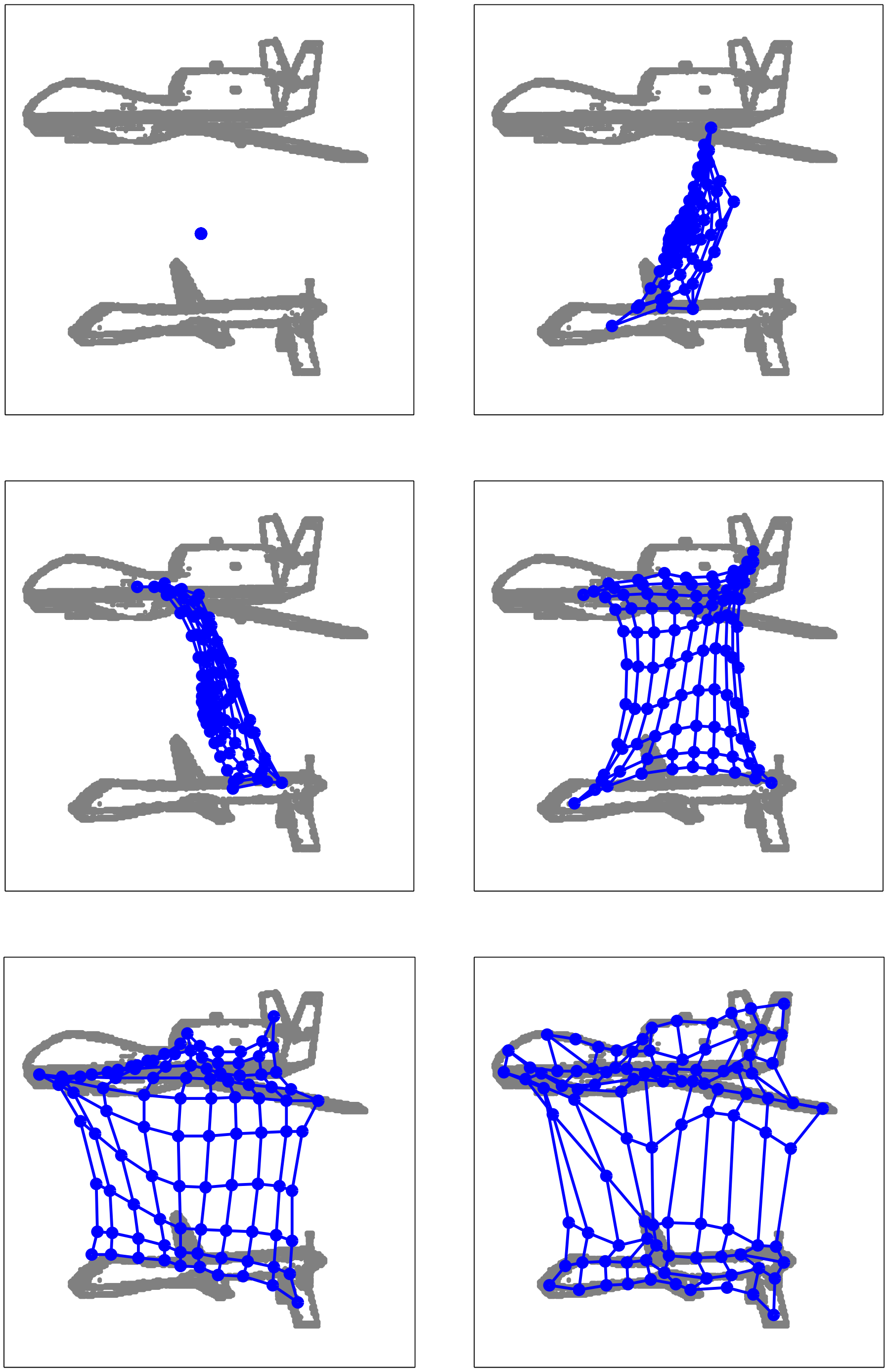


Figure 2.9: Two-dimensional input data (in gray) allows visualization of the preservation of the topological neighborhood (lower-right image). In topological neighborhood preservation, adjacent PE's are assigned to the same data set.

2.2.7 Using a SOM in Spectral Analysis. The key to exploiting the SOM for spectral analysis is segmenting the neighborhoods that have been topologically preserved. The number of PE's with a given label (or material type) is proportional to the number of pixels of that material that exist in the original hyperspectral image [22]. There are several ways to approach the process of segmenting the SOM neighborhoods.

One way is to make use of *a-priori* knowledge to segment what can be immediately known or determined. An example of this is the Normalized Difference Vegetation Index (NDVI) [36], presented in Section 2.2.9. It is a tool that is well studied and has been used in remote geosensing for decades. The NDVI is a simple ratio of visible and near infrared bands.

The *topological neighborhood preservation* characteristic of the SOM allows the SOM to be segmented into contiguous regions where the contiguous regions are made of adjacent SOM PEs. A SOM segment can be clustered around exemplar "seedling" pixels (e.g. user selected or from a spectral database). The nearest neighbor "winning" SOM PE to the the exemplar spectra in the ℓ_2–norm sense is located, and spatial analysis technique can be employed to segment the region around the "winning" PE. The region can be segmented by setting a threshold, using edge detection modified for high-dimensional data, region growing, or specifying a fixed number of nearest neighbors.

2.2.8 The Spectral Database Tool. The US Geological Survey provides hyperspectral endmember spectra for hundreds of materials. The materials are separated into categories: Man-Made, Mixture, Volatile (water), Plant, and Mineral. They are also sub-categorized to identify the common name for each material (e.g. Oak, Maple, Seawater, Asphalt, etc). This database is a valuable resource of endmember spectra that can be used for a quantitative assessment of an image. However, a spectral database is not used in this thesis for three reasons.

First, an endmember spectra that is desired may not exist in the database. Because the spectral analysis uses a nearest neigbor calculation based on the ℓ_2–

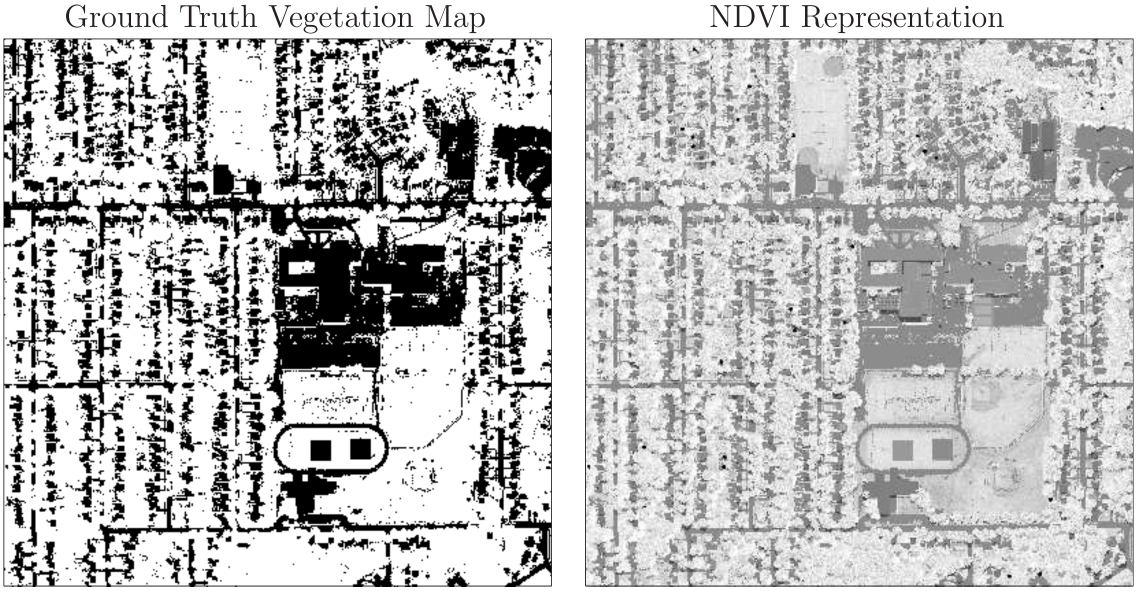


Figure 2.10: The ground truth vegetation map (left) compares favorably to the hyperspectral image after application of NDVI (right).

norm, it finds pixels in the scene that are the most similar to the desired spectra. For example, if searching for pixels similar to a US Geological Survey sample of concrete, some image pixels are returned that are geometrically close to concrete, even if there is no actual concrete spectra in the scene. There is no way to know for sure if these pixels are actually concrete. Second, in-scene materials may be composed, at the atomic level, of different materials than those used in the US Geological Survey Database. Concrete is a good example of this phenomenon. Concrete is often made of materials local to a region. The soil, soil content, minerals, ash, stone, pebbles, and other materials used in concrete vary distinctly from region to region and even field to field. Concrete buildings on different sides of a city may have different spectra. Third, the presence of dust, pollen, snow, water, or other contaminants change the spectral signature and render the spectral database information obsolete. For these reasons, in-scene spectra are used for spectral analysis.

2.2.9 Normalized Difference Vegetative Index (NDVI). The NDVI is a tool that was developed early in geosensing. It is used to estimate the vegetation in desert regions [55], determine vegetation type and coverage [54] and [56], determine the age of trees [35], and assess the quality of urban environments by their foliage cover [53].

In order to perform photosynthesis, plants collect energy in the visible region of the spectrum between 400 nm and 680 nm[1]. The Near Infrared (NIR) segment of the spectrum, from approximately 725 nm to 1100 nm, does not provide enough energy for photosynthesis and only acts to unnecessarily heat the vegetation. Plants have adapted to absorb most of the photosynthetic-friendly visible spectrum and scatter the NIR region of the spectrum. This creates a very sharp jump in the red region of the visible spectrum, somewhere between 680 nm and 725 nm. This sharp curve is caused by photosynthetic-active cells in the vegetation and is commonly called the "Red Edge" [36].

The NDVI compares the two sides of the red edge. In the visible region, the light is absorbed for photosynthesis and less than 20% of the light is reflected. There may be a small peak in the visible green region, which gives vegetation its characteristic green coloring. In the NIR, vegetation scatters up to 80% of the incident energy. By comparing these two regions in a ratio, an assessment can be made about how likely a particular hyperspectral pixel contains vegetation.

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{RED}}{\mathrm{NIR} + \mathrm{RED}} \tag{2.6}$$

The National Oceanic and Atmospheric Administration (NOAA) satellite bands for the RED and NIR regions, 400 nm to 680 nm and 725 nm to 1100 nm, are used to calculate the NDVI. Selecting a single sample band somewhere in the NOAA NIR or RED range might result in randomly selecting a noisy band and skew the results. The red edge also becomes less sharp when dealing with mixed spectra (e.g. grass and soil), and this can result in an NDVI value that is lower than it should be. In order to make the NDVI value robust to noise and vegetation type, an averaging operation

---

[1] The values 400 nm to 680 nm and 725 nm to 1100 nm were chosen because these are the ranges used by the National Oceanic and Atmospheric Administration (NOAA) satellites.

Table 2.1: Listing of six NDVI thresholds and their interpretations.

NDVI Value	Interpretation
1.0 to 0.8	Dense Vegetation
0.8 to 0.3	"Rule-of-Thumb" Vegetation Threshold
0.3 to 0.2	Mixed Spectra
0.2 to 0.1	Soil
0.1 to -0.1	Standing Water
-0.1 to -1.0	Water or Snow

was used. RED is the average of all the bands between 400 nm to 680 nm and the NIR value is the average of all the bands between 725 nm to 1100 nm.

The resulting NDVI values in Table 2.1 are between definitely not vegetation (-1), and definitely vegetation (+1). Some other things can be generally inferred from the NDVI, such as the health of vegetation, the soil content, and agriculture yields. In [54], spectral information and NDVI was used to separate trees from shrubs. In [35], the authors used NDVI to assess the age of trees. In [53], the authors used the water absorption analysis in combination with NDVI to distinguish trees from other vegetation in an urban environment. Figure 2.10 is an example of applying NDVI to a hyperspectral image.

NDVI measures the presence of chlorophyll. Chlorophyll content can depend on drought and the season. As chlorophyll content is lessened, the NDVI threshold decreases toward the soil range threshold. When analyzing hyperspectral images, it is necessary to take into consideration the season of the year in which the image was acquired.

2.2.10 Separating Trees from Grass.

In the urban classification problem, roadways may often be occluded by overhanging vegetation. This is usually broadleaf or needle-bearing trees. In order to segment both the roadway and the trees, it is necessary to separate tree spectra from other vegetation. Of interest are any kind of obstructions to movement in the urban environment, which is another justification for focusing energy on segmenting trees. This problem, in general, is a challenging one.

In [54], the authors used a fuzzy classification method to separate trees from shrubs. Water absorption in combination with NDVI was used in [24] to identify trees from grass. The spectral features of broad-leaf trees and needle-leaf trees were used to estimate tree cover in [56]. The authors in [35] used NDVI to distinguish the age of trees, and in [53] to distinguish types of trees in an urban environment. Other papers have combined lidar height data with hyperspectral images to locate trees in an urban environment. However, no consistent method has been developed to separate trees from grass in all cases using spectral feature extraction alone.

2.3 Spatial Analysis

Spatial analysis with hyperspectral images is challenging. In [19], the authors attempt scene segmentation by applying a modified edge detection algorithm that can process high-dimensional data. Simple scenes with few mixed pixels and uniform objects are used as the test data. The results are not extended to complex hyperspectral scenes. In [45] complicated heuristics with convolution operations highlight road candidates and "suppress" background information. The resulting road network segmented through spatial processing is incomplete and unsatisfactory.

One technique used in [6], [10], and [14] are morphological profiles. These profiles are created by clustering and deleting pixels using standard morphological *open* and *close* operations (detailed below). The author selects a single representative band, performs edge detection, and then performs the morphological operations. The result is a "fingerprint" that can be compared to known fingerprints, usually from a GIS database. After the unknown and known fingerprints are matched, the GIS database is used to improve the quality of the spectral classification. While technically making use of the spatial arrangement of pixels in a certain band to register unknown images with *a-priori* knowledge, these approaches do not consider additional spatial information such as size of the objects, relative location, or quantity. True spatial analysis exploits all of these features.

2.4 From Spectral to Spatial Analysis

In traditional spectral analysis, every pixel in the original scene is labeled by a classifier. In "known" scenes where the label of each pixel is known (often an arduous manual task of identifying each pixel), the results from the algorithm can be compared to the "ground truth". The match between the ground truth and algorithm-produced class labels is the reported accuracy. The quality of an algorithm is normally assessed by its accuracy. In general, higher accuracies indicate better results. Nearly every paper that uses machine-learning to do classification of spectra uses this approach, as in [4], [6], and [52].

However, spatial information contains a wealth of information. Used in combination with spectral analysis, it can improve the algorithmic processing of a scene. In [30], the authors found that distinguishing rooftop asphalt shingles from asphalt road pixels was not possible without spatial size and shape context. In [47], the authors visualize the link between the topology of a self-organizing map and the spatial organization of the materials, but without using the spatial information to determine the scene layout. In [14], the classification accuracy was increased from 79% to 83% using spectral analysis combined with a spatial analysis technique called morphological profiles[2]. In [5], the authors increased accuracy from 72% to 100% with wavelet decomposition while analyzing very distinct vegetation types. In [21], the classification accuracy is increased using Markov Random Fields. These results are typical of spectral analysis. Starting with an initial accuracy, improvement comes from tuning

---

[2] The spatial analysis technique *morphological profiles* should not be confused with the spatial morphological operations used in this thesis. Morphological profiles is a registration method for hyperspectral images and known GIS databases. This thesis assumes no prior GIS database information is available.

the spectral analysis algorithm, doing feature selection, and making use of *a-priori* GIS database knowledge or endmember spectra.

This section is a unique contribution to the field of hyperspectral analysis. While minimal use has been made of the spatial arrangement of pixels through morphological profiles, there has been no notable research in the use of spatial analysis in combination with spectral analysis to abstract scene content. Part of the reason is the reduction in accuracy. In the higher-accuracy-is-better mentality, completing spatial arrangements, for example connecting roads under tree occlusions, would reduce accuracy. Another reason is the difficulty in applying traditional spatial scene analysis tools to high-dimensionsal data. Most papers, such as [4] and [6] limit the spatial analysis tools to one to three bands of the hyperspectral image to emulate using an RGB image. This approach does not exploit the value of having several hundred spectral dimensions. The unique contribution is to use spectral analysis to identify in-scene materials and create individual two-dimensional spatial material maps. By examining the spatial layers first individually, then in conjunction with all of the other material layers, knowledge is abstracted to form a complete set of information about the hyperspectral scene.

There are several challenges with this traditional idea of accuracy. For example, pixels classified as *asphalt* are rarely segmented into "road", "parking lot", "driveway", or even "rooftop" (for tar or asphalt-shingle rooftops). An image with every single pixel forced into one class or another also tends to appear, to the observer, just as "cluttered" as the original, unclassified image. The solution to this problem is to reduce the number of pixels that are classified, thereby reducing the information. This follows the natural process people use in segmenting a scene. The corollary to this solution is to separate the classified pixels into observable structures that make sense to an observer, e.g. "road", "buildings", "trees", and other major features of an image. The spatial analysis problem is complicated by obscurations. If trees cover majority of a roadway, the observer can sometimes mentally connect the roadway sections underneath the trees. This is something most spatial analysis algorithms have an enormous difficulty in accomplishing because there is no known way to teach a computer to conceptualize the location of non-existant objects.

Segmenting the SOM with overlapping neighborhoods results in PEs that are classified with multiple labels. The sample hyperspectral image spectra can then have multiple labels as well. By identifying tree vegetation that is mixed with roadway pixels, it may be possible to identify roadway or structures that are obscured by trees. The ability of the SOM to place mixed spectra between two groups of pure spectra, and by using thresholds that allow the pixel to be included in both tree and roadway analysis, brings about the ability to identify roadway underneath trees. Spatial processing of sets of classified pixels (asphalt) into human-level abstractions (rooftop, road) is a notoriously difficult problem.

2.4.1 Spatial Analysis.

While spectral analysis is not intuitive without the appropriate background, spatial analysis is instantly understood by most people. This is because spatial analysis is accomplished by people on a continual basis every day of their lives. When three bands of a spectral scene (usually from the red, green, and blue regions) are combined into a false-color RGB image, the spatial understanding of the scene by a human is usually instantaneous. The identification of buildings, roads, parks, fields, and other features happens almost without conscious thought. Missing information (roads under trees) is automatically filled in, and existing information (large areas of grass) are quickly abstracted (into baseball fields, soccer fields, etc).

This instantaneous understanding is not the case for computer algorithms. Many papers have approached the problem of spatial analysis, but the traditional difficulties exist. Most attempts to do spatial analysis use well-understood and standard image analysis techniques of edge detection [25] or region growing [25] to identify similar materials by setting thresholds. In [4], the authors limit the information by reducing the number of bands under analysis to three.

In [4], the authors use a hyperspectral image of the DC Mall. The DC Mall imagery is a 191 band hyperspectral image with a resolution of $1,280 \times 307$ pixels.

They use Principal Component Analysis to reduce the number of bands their algorithm operates on to the three, accounting for 99% of information variation in the image. By varying the size of the structuring element, image features were extracted corresponding to the size of buildings. Using an automated method based on [48] and starting with a given pixel, the spectral variations surrounding the neighborhood of the incident pixel were used to calculate a threshold used to distinguish buildings. The combination of the number of connected pixels, $C(n)$, and the spectral variation, $D(n)$, was used as a goodness measure to distinguish regions. The equation for the goodness, $M(n)$, is

$$M(n) = D\left(n, parent(n)\right) \times C(n). \tag{2.7}$$

In [52], the authors deal with the issue of using multispectral images to automate the update of GIS databases. By manually overlaying the multispectral images over GIS maps, the GIS map can be used to "classify" the pixels of the multispectral images. After classification, change detection on the multispectral images is used to determine new information, from which the GIS database is then updated.

Most spatial image analysis suffer the same issue. Once they identify a region, border, segment, or other feature, they are unable to then abstract it into human-concepts (e.g. house, road, or parking lot). It usually takes a human-in-the-loop to confirm what the spatial analysis has found. Spatial analysis then becomes a process of a human-dominant human-computer interaction to confirm or deny objects, and fine-tune the selection process. This is a very labor-intensive process that needs to be automated. The methodology developed in Chapter III addresses these challenges.

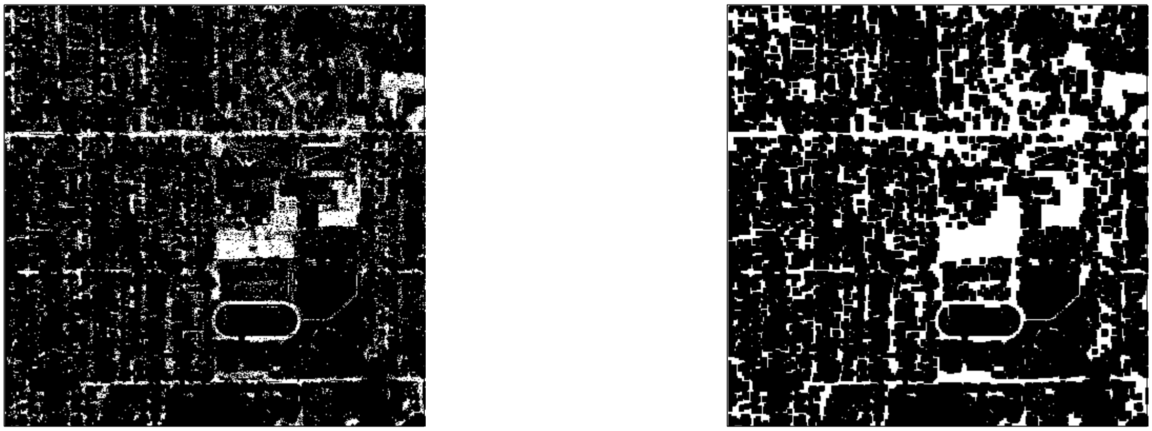
2.4.2 Morphological Operations. In [46], the author used a complicated heuristic to describe the physics of how roads are laid out. The resulting road networks are satisfactory, but not complete. The author suppressed scene information and extracted road candidates using sample spectra. In [39], the authors extract the "road skeleton" using spectral roadway identification and then growing the road segments together. As the probability of detecting true roadways increases, the completeness of the road network dramatically decreases. Alternate spatial operations are needed to construct complete roadway maps from a hyperspectral image. By specifying roadway maps in combination with many other materials, *morphological operations* may be used to accomplish spatial processing.

Morphological operations are uniquely suited to a human-level processing of images. By isolating specific materials using spectral analysis, features of the image become immediately obvious to humans. For example, identifying a few pixels of roadway results in a significant number of roadway pixels being identified (and some false alarms including asphalt shingles and parking lots). These pixels are separately displayed in a two-value image, usually black-and-white, where one value marks the pixels identified and the second value identifies everything else. For visual analysis, the desired feature (roadway) is colored white, and the undesire feature (everything else) is colored black. This highlights the road network to the user. The second thing that becomes obvious is the shape of the feature - usually square, rectangle, or with smooth-curves that signify man-made objects.

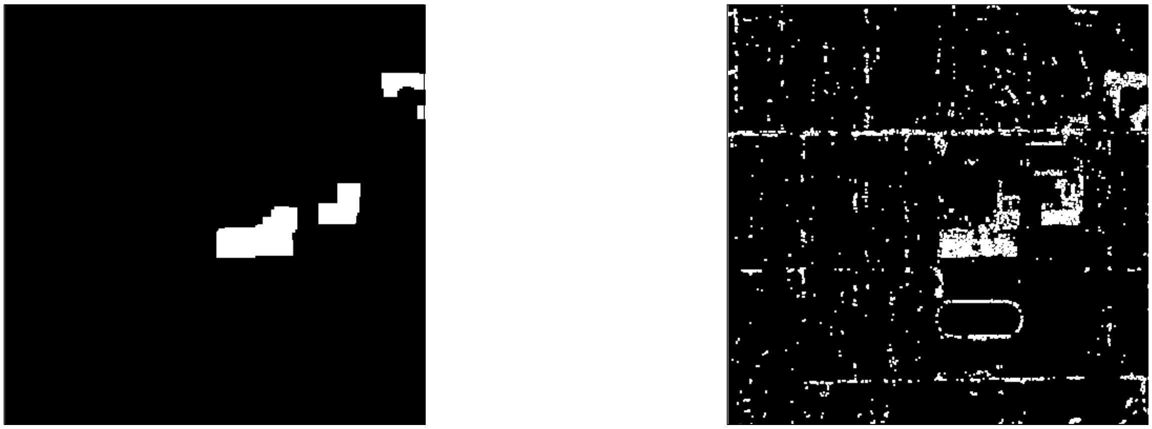
Once a particular material type is isolated into a binary spatial image, a morphological approach to analysis becomes available. "The morphological approach is generally based upon the analysis of a two-valued image in terms of some predetermined geometric shape known as a *structuring element*." [16] Choosing the geometric shape is considered a user-option, however a standard set of rules apply when it comes to grouping pixels in a spatial image.

These rules are as follows. It is possible to group pixels that are at least the same size as the *structuring element* in size. This meets with the previous criteria to assemble groups of pixels into complete abstract shapes. By starting with the smallest structural element possible, a two-by-two square, groups of pixels can be clustered. The pixels are automatically and iteratively grouped, even if the group of pixels exceeds a two-by-two spatial size. It is possible to throw away pixels smaller

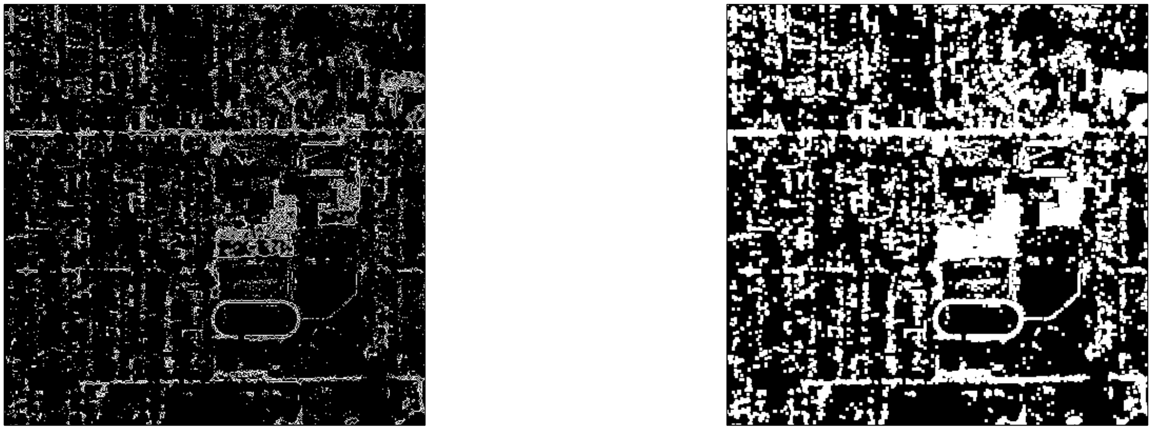
Examples of Morphological Operations


Original

Close, 5 × 5 SE


Iterative Open/Close

Open, 2 × 2 SE


Remove

Dilate, 3 × 3 SE

Figure 2.11: The original scene (upper left) is operated on with a close with a 5×5 SE (upper right), an iterative open/closes with an $n \times n$ SE where $n = 2, \dots 10$ (middle left), an open with a 2×2 SE (middle right), a remove that "outlines" the objects (lower left), and a dilation with a 3×3 SE that "fattens" the objects.

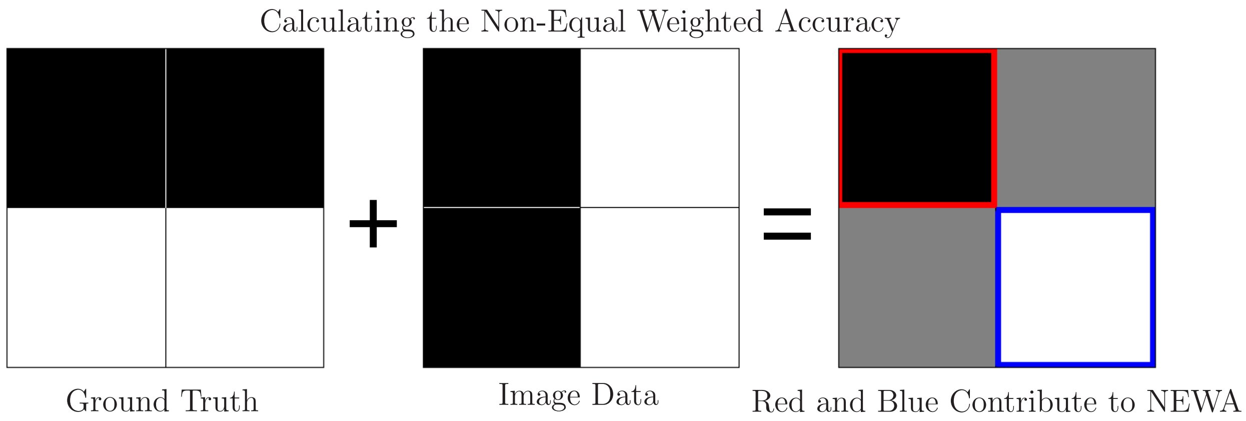

Calculating the Non-Equal Weighted Accuracy
+
=
Ground Truth
Image Data
Red and Blue Contribute to NEWA

Figure 2.12: This is the process to calculate the NEWA. The NEWA accuracy includes correctly classified pixels and does not include differences. In this example, the red and blue highlighted boxes, representing correctly classified pixels 0 → 0 and 1 → 1, results in NEWA = 0.50.

than the structuring element. This means stray pixels no larger than a grouping of one or two can be discarded.

The "grouping" operation is called an *image close* (Figure 2.11, upper right), and the "deletion" operation is called an *image open* (Figure 2.11, middle right). By increasing the size of the structuring element from two-by-two to some arbitrarily large square (or other shape), it is possible to isolate increasingly larger groupings of a particular material. For example, if all grass locations have been identified, one can iteratively close and open the grass map (Figure 2.11, middle left), resulting in the preservation of increasing areas of grass. Small grassy areas (urban lawns) are deleted, and large grassy areas (fields) are preserved. Using the morphological open and close operations, and changing the size and shape of the structuring element, image features can be abstracted with "detection maps" or "material maps" as its input. The "dilation" operation (Figure 2.11, lower right) adds a layer of pixels to each object, and makes them look "fatter". The "remove" operation (Figure 2.11, lower left) "outlines" the objects. Dilation is used to increase the visual appeal of the major buildings object map, and the remove operation is used to outline the objects to create visually appealing GIS layers.

2.4.3 Measuring Accuracy. In this thesis, the “ground truth” is a binary spatial *object map* indicating the locations of a particular material. It is tempting to consider it to be a single class where binary “true” is a success and binary “false” is a failure. This, however, fails to recognize that “false” is a success as well. The accuracy calculation will instead be thought of as a two class problem with classes “true” and “not true”. The Non Equal Weighted Accuracy (NEWA) is used to assess accuracy as shown in Figure 2.12.

III. Methodology

Chapter III details the process of converting a hyperspectral image into individual GIS layers illustrated in Figure 3.1. The four procedural steps are *spectral analysis* (Figure 3.3), *creating the initial object maps* (Figure 3.10), *spatial analysis* (Figure 3.12), and *finalizing the GIS layers* (Figure 3.15). An *object map* is a binary-valued spatial representation of the hyperspectral scene, where the binary "true" indicates presence of a particular *material* or *object* type. The *initial object map* is intended to show the spatial location of a particular *material* or *object type* where the material might be "vegetation", "asphalt", or any material that may contribute information to developing the final GIS layers. One material that may be of interest that is not included in this process is "water". It is not included because it does not exist in sufficient quantities in the hyperspectral scene to warrant a separate "water" GIS layer.

Spectral analysis is accomplished with a Self-Organizing Map (SOM) followed by *segmentation* of the SOM to determine its *topology.* Each time the SOM is segmented, the result is a *SOM segmentation.* A *SOM segmentation* is a binary-valued representation of the SOM that identifies the processing elements (PEs) that correspond to a particular material. The SOM segmentation is a partial classification of the SOM PEs. It is not necessary to completely classify the PEs because *not* all of the PEs contribute useful information to the final GIS layers. The difference between an *object map* and a *SOM segmentation* is their use. An *object map* has the spatial resolution of the hyperspectral image and shows the locations of in-scene materials. A *SOM segmentation* is a matrix with the same dimensions as the SOM PE lattice showing the PEs that are labeled with the particular material type. Each *SOM segmentation* is used to produce one *temporary object map.* The *temporary object map* is used for spatial scene verification (see Figure 3.1, *spectral analysis*). During spatial scene verification, the size of the SOM region being segmented is adjusted using a threshold until the material of interest is shown on the *temporary object map.*

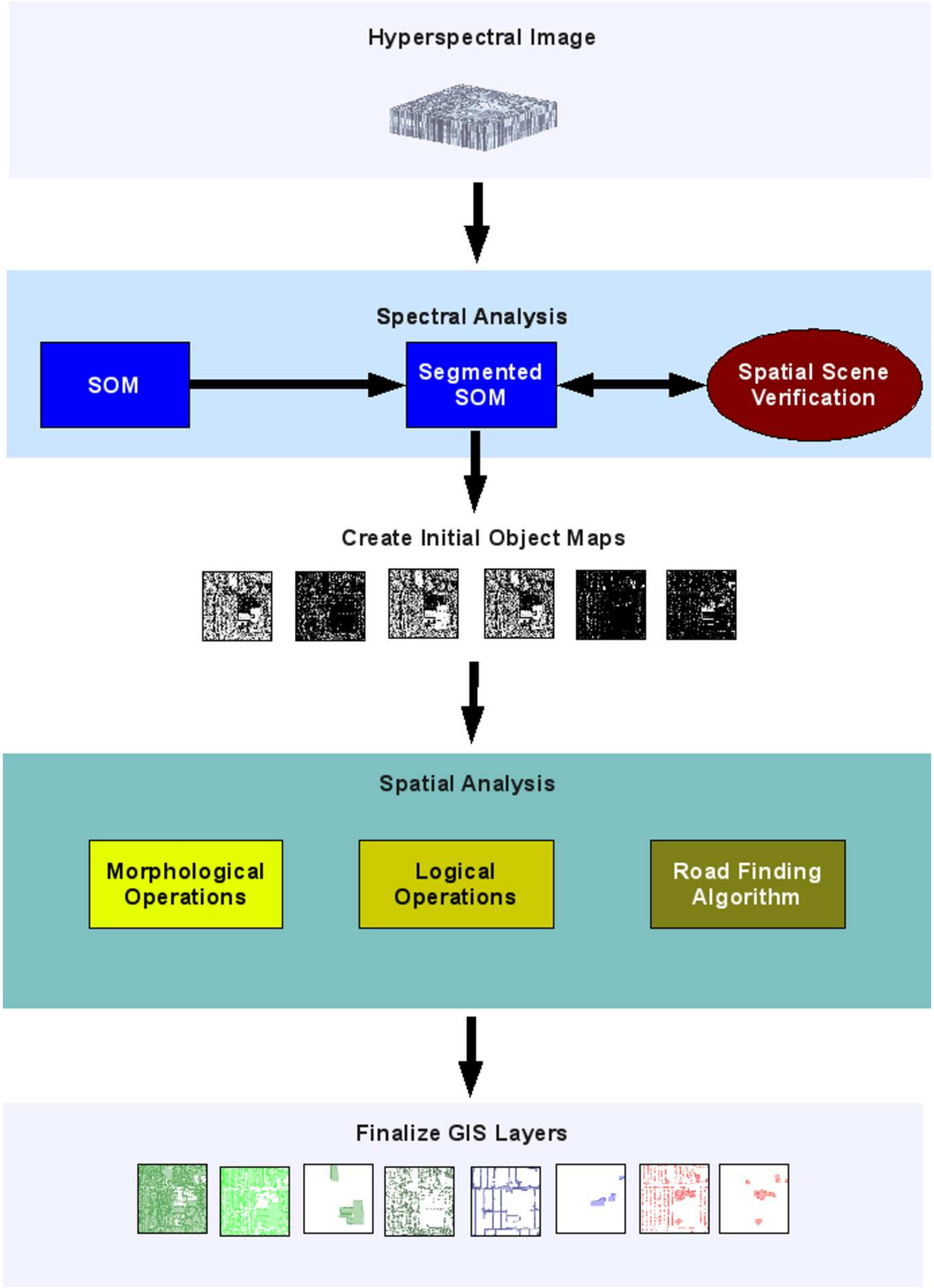


Figure 3.1: The process flow diagram to convert a hyperspectral image into GIS layers.

After spectral analysis, each of the resulting *SOM segmentations* is used to create an *initial object map*. This step is the transition step from spectral analysis to spatial analysis. The *initial object maps* are the inputs to the spatial analysis process.

Spatial analysis uses the *initial object maps* and performs *logical* and *morphological* operations, and the *road finding algorithm*. Spatial analysis results in a set of *final object maps*. It needs to be emphasized that the *final object maps* do not correspond one-to-one with the input initial object maps for two reasons. First, there are two more *final object maps* than *initial object maps* because the spatial analysis process abstracts two new *object maps* from the input *initial object maps*. Second, some of the *final object maps* are not labeled the same as the input object maps because the spatial analysis process abstracts the meaning of the map. For example, the "rooftop" *initial object map* becomes the "buildings" and "major buildings" *final object map*.

Once the *final object maps* are prepared, the GIS layers are finalized by applying a color, "outlining[1]" the objects to emphasize their shape, and creating a transparency mask so the GIS layers can be overlayed on a false color composite of the hyperspectral image.

3.1 Process Inputs

The process is designed to operate on a hyperspectral image cube. The image cube needs to contain both spectral and spatial information, as shown in Figure 3.2. It is assumed that the needed pre-processing steps to prepare the hyperspectral image for algorithmic analysis are complete. Some of the pre-processing steps might be resampling, atmospheric correction, and adjusting for the angle of the sun [20]. Pre-processing is dependent on the sensor platform, sensor noise, the hyperspectral imager, and multiple other variables. The process is unique in every situation, and is accomplished prior to spectral and spatial processing. Section 4.1.0.4 further de-

---

[1] "Outlining" is accomplished with a morphological *remove* operation.

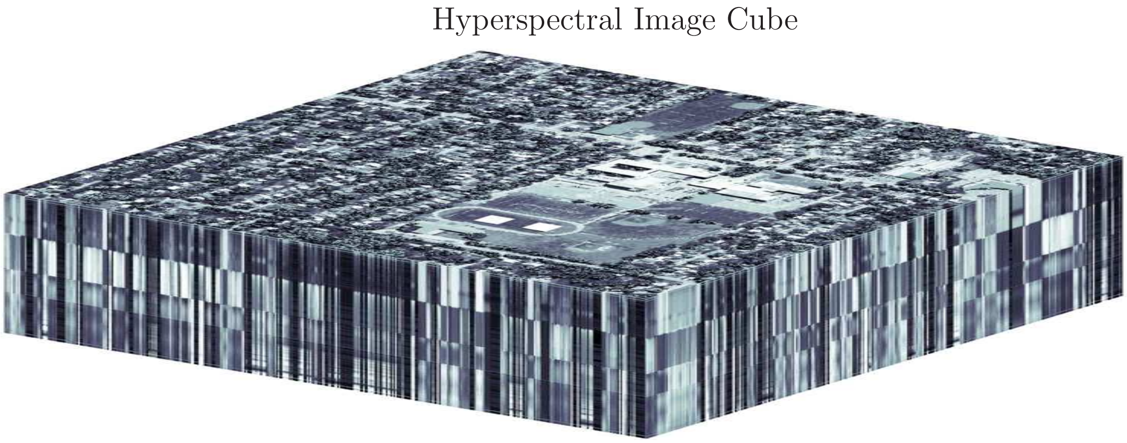


Figure 3.2: This is an example of a hyperspectral image cube used in the subsequently described process. It contains both spectral and spatial information.

tails the specific pre-processing steps used to create Figure 3.2. Once the image is prepared, it is used in the spectral analysis process that follows.

3.2 Spectral Analysis

The flow diagram for spectral analysis is shown in Figure 3.1. A SOM is used as the primary vehicle for spectral analysis. The SOM is then segmented to reveal its topology. First, the normalized difference vegetative index (NDVI) is used to create two *SOM segmentations* using two different thresholds. The two segmentations can be thought of as confidence limits. The NIR scatter tree identification algorithm is used in combination with NDVI to create one SOM segmentation, and user-selected exemplar spectra are used to create one *SOM segmentation* for "rooftop" and "roadway" materials. A *temporary object map* is created for each *SOM segmentation* and is used in spatial scene verification before proceeding. During the spatial scene verification process, a *temporary object map* is viewed by a user and the ℓ_2–norm threshold for the *SOM segmentation* is manually adjusted until the material of interest is maximized and false positives minimized.

It is useful during spatial analysis to have a "non-tree vegetation" *object map*. The "tree" *SOM segmentation* and the two "vegetation" *SOM segmentations* are combined to produce two "non-tree vegetation" *SOM segmentations*. The result is

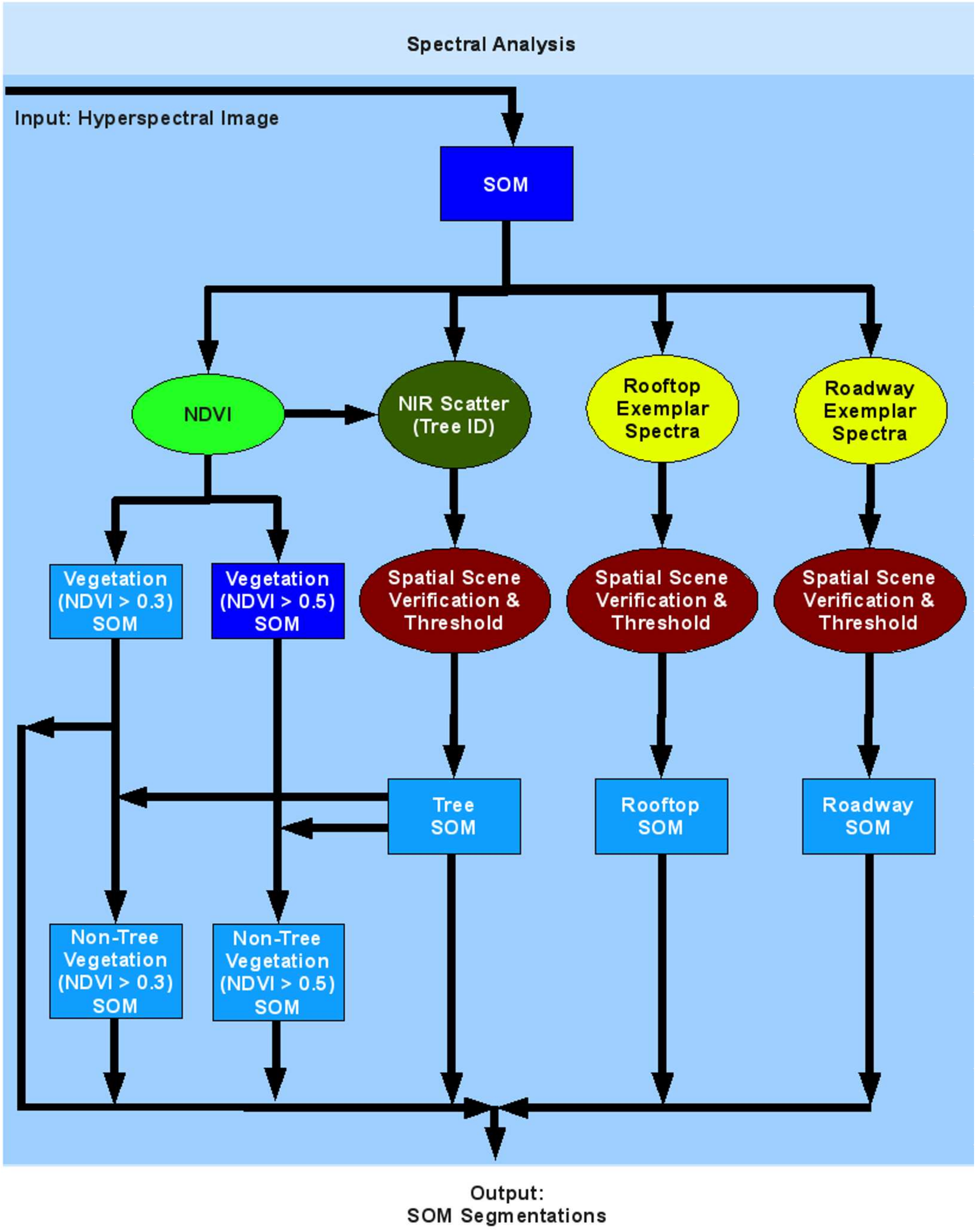

Spectral Analysis
Input: Hyperspectral Image
SOM
NDVI
NIR Scatter (Tree ID)
Rooftop Exemplar Spectra
Roadway Exemplar Spectra
Vegetation (NDVI > 0.3) SOM
Vegetation (NDVI > 0.5) SOM
Spatial Scene Verification & Threshold
Spatial Scene Verification & Threshold
Spatial Scene Verification & Threshold
Tree SOM
Rooftop SOM
Roadway SOM
Non-Tree Vegetation (NDVI > 0.3) SOM
Non-Tree Vegetation (NDVI > 0.5) SOM
Output: SOM Segmentations

Figure 3.3: This is the spatial flow diagram to convert a hyperspectral image into SOM segmentations. The six light blue SOM segmentations are outputs.

seven SOM segmentations, six of which are outputs of the spectral analysis process. In Figure 3.3, the six output *SOM segmentations* in light blue boxes are labeled "vegetation (NDVI $>$ 0.3) SOM", "non-tree vegetation (NDVI $>$ 0.3) SOM", "non-tree vegetation (NDVI $>$ 0.5) SOM", "tree SOM", "rooftop SOM", and "roadway SOM". The seventh *SOM segmentation* in a dark blue box that is not an output of the spectral process is labeled "vegetation (NDVI $>$ 0.5) SOM". It is not an output because it is not used during spatial analysis. It is only used to create the "non-tree vegetation (NDVI $>$ 0.5) SOM". The output is robust in the sense that minor variations in the size of the segmented regions is compensated for during spatial analyis.

3.2.1 Self-Organizing Map (SOM). A SOM organizes the spectra from a hyperspectral image. The SOM requires seven input training parameters: the size of the SOM PE lattice, the number of learning schedule decay steps, the learning rate max and min, the neighborhood function max and min, and the number of training steps. The SOM is inspected for convergence by randomly selecting one band, colorizing it according to PE value, and displaying the band. The SOM band should have regions of similar colors indicating the preservation of the topological neighborhood and the colors should have a gradation indicating the spectra are organized in an gradual ascending or descending pattern as shown in Figure 3.4. The topological neighborhood of the SOM is segmented into separate "SOM segmentations", one for each material of interest, and starting with NDVI.

3.2.2 Normalized Difference Vegetation Index. The NDVI is calculated for each of the processing elements in the SOM. Table 2.1 lists the range of possible NDVI values, and two of them (NDVI $>$ 0.3 and NDVI $>$ 0.5) are used as "confidence limits" to create two "vegetation" *SOM segmentations*. Figure 3.5 shows the results of applying NDVI to the SOM and setting a threshold to isolate one of the "vegetation" *SOM segmentations*. The NIR scatter algorithm is next used to segment the "tree" neighborhood of the SOM.

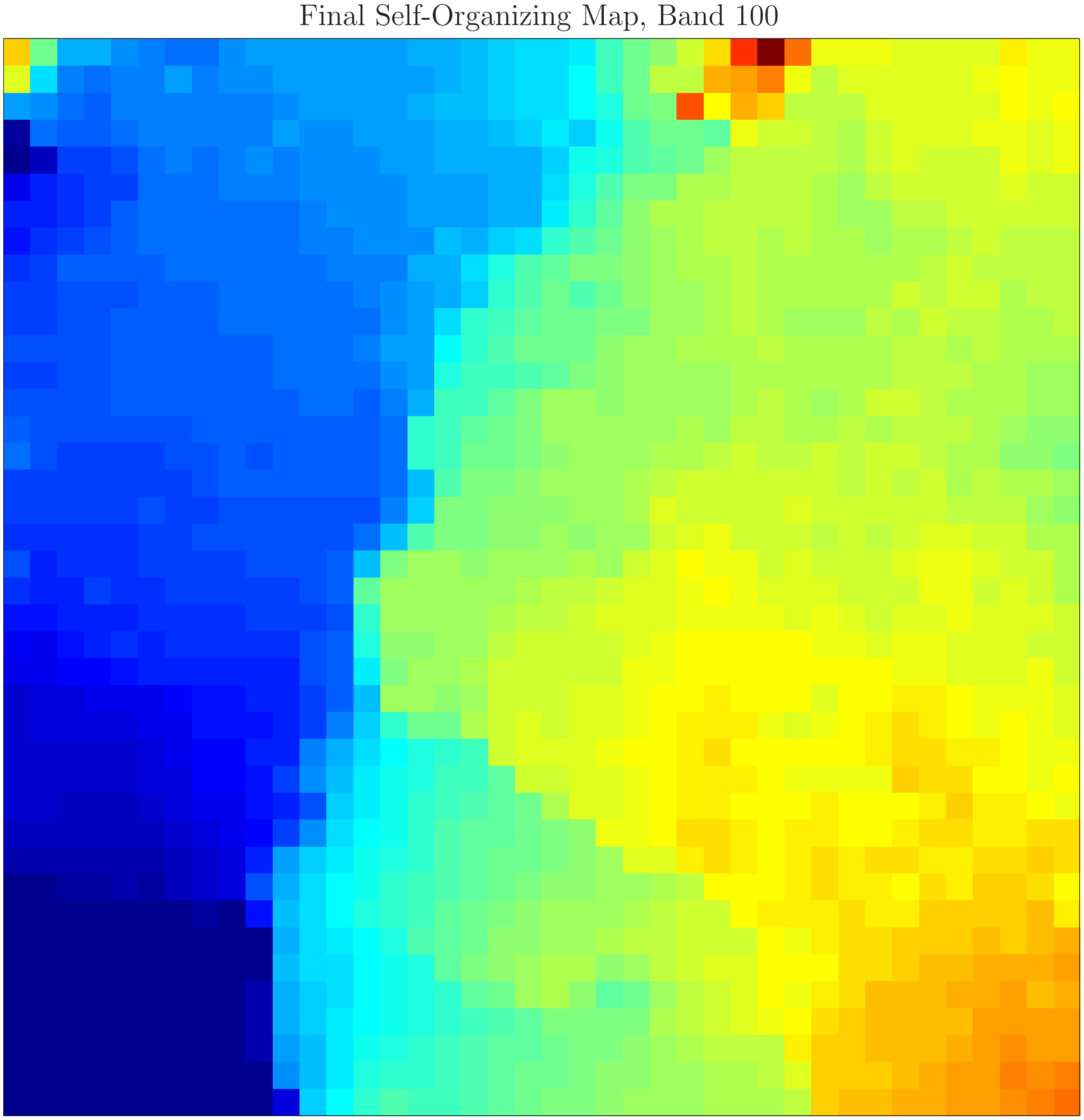


Figure 3.4: A self-organizing map (SOM) that reaches convergence is characterized by its *topological neighborhood preservation*, which is visualized by coloring the processing elements of a randomly chosen band of the SOM (above). Topological neighborhood preservation is evidenced by the “regions” and smooth gradations of color.

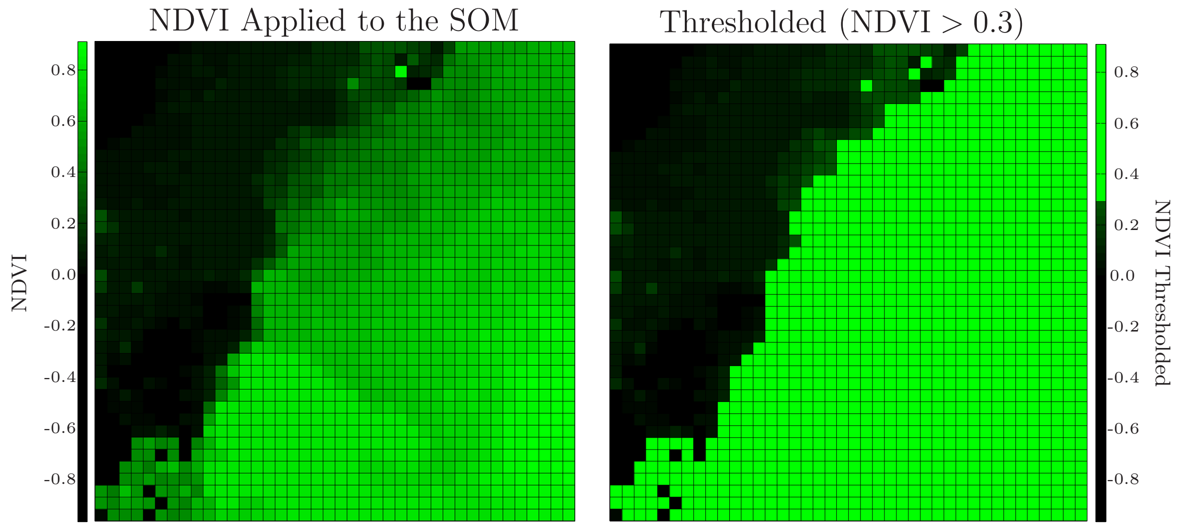


Figure 3.5: An example of NDVI applied to the SOM (left) and setting a threshold from Table 2.1 (right).

3.2.3 NIR Scatter Tree Identification. The process to locate trees is based on analysis of the NIR region of the spectrum. Spectral analysis as a means to identify trees is referenced in the literature, but this particular method is a novel approach that was developed for this thesis[2]. For this reason, it is explained in detail below. It needs to be noted that this technique works only in combination with NDVI, because this method also identifies non-tree PEs (in the SOM) and pixels (in the hyperspectral image).

Figure 3.6 shows the segment of the spectrum that is used to classify trees. The maple tree endmember spectra used to simulate the hyperspectral scene exhibit less NIR scatter than the urban grass. This difference in NIR scatter is exploited to separate tree from grass spectra. The spectral region 0.951–1.247μm is normalized over the range, and then the standard deviation of the result is calculated.

The NIR scatter is calculated for each PE in the SOM, and the NIR scatter values vary between 0 and 0.2059. NDVI is applied to each PE to remove "non-tree" PEs from consideration, and then the "tree" *SOM segmentation* is used in two ways.

---

[2] The analysis of the NIR scatter is motivated by a literature search that indicated this is possible, however subsequent detailed search has not revealed the original source of this method. The method developed for this thesis is novel because it was created without source material.

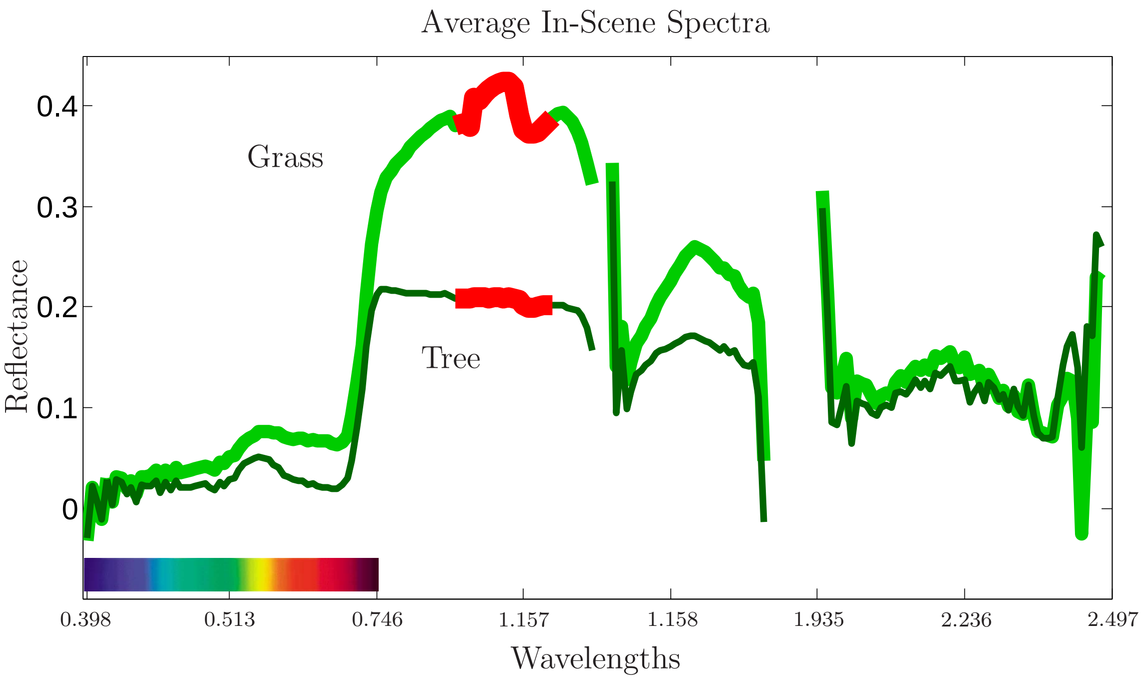


Figure 3.6: Trees are identified using the NIR scatter tree identification region (thick red line) between wavelengths 0.951–1.247μm. Urban grass (light green line) exhibits a larger amount of NIR scatter than maple tree leaves (dark green line). This variation is exploited to segment the "tree" neighborhood of the SOM.

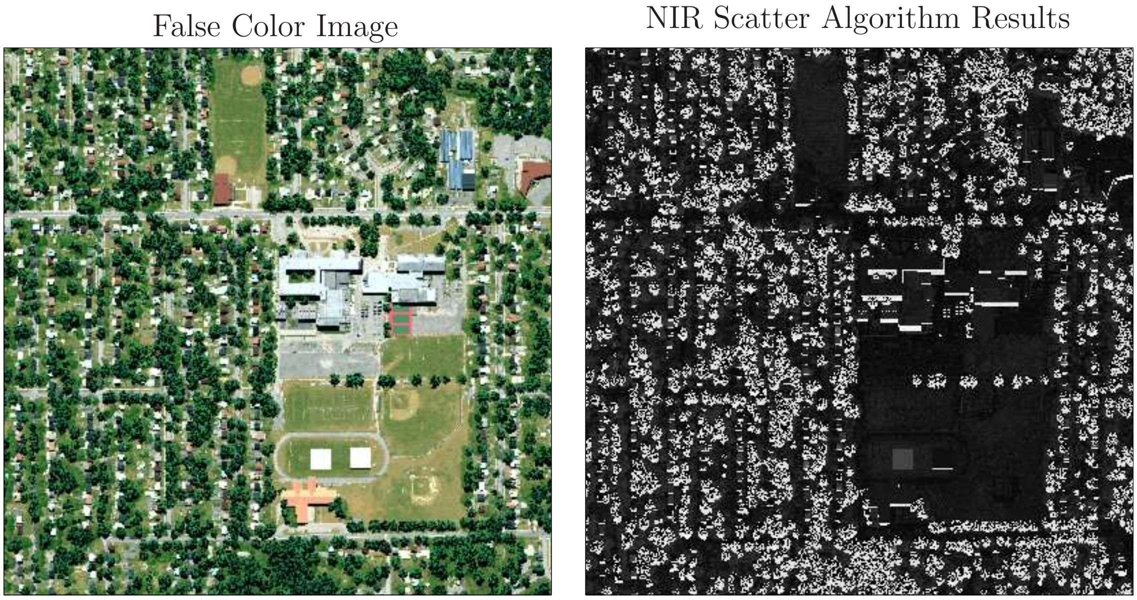


Figure 3.7: Applying NIR scatter to a hyperspectral image (right image) demonstrates the utility of the NIR scatter tree ID method. The false color composite (left image) of the hyperspectral image is shown for comparison. This comparison demonstrates the usefulness of the NIR scatter tree ID algorithm.

Table 3.1: Interpretation of logical deletion.

Roadway Object Map	Fields Object Map	*Final* Fields Object Map
false	false	false
false	true	true
true	false	false
true	true	false

First, the NIR scatter threshold to separate “tree” from “non-tree” is determined by plotting the ordered NIR scatter values and calculating the maximum of the derivative. A *temporary object map* is created from the “tree” *SOM segmentation* for visual scene verification. Visual scene verification can also be used to manually adjust the threshold to improve the tree *temporary object map.* To verify the results, the NIR scatter tree algorithm (and NDVI) are applied to an example hyperspectral image (Figure 3.7, left image), and a *temporary object map* used for spatial scene verification is created (Figure 3.7, right image). Figure 3.7 demonstrates the applicability of the NIR scatter tree algorithm. Next, the “vegetation” and “tree” *SOM segmentations* are used to create two “non-tree vegetation” *SOM segmentations* (see Figure 3.3).

3.2.4 “Non-Tree Vegetation” SOM Segmentations.

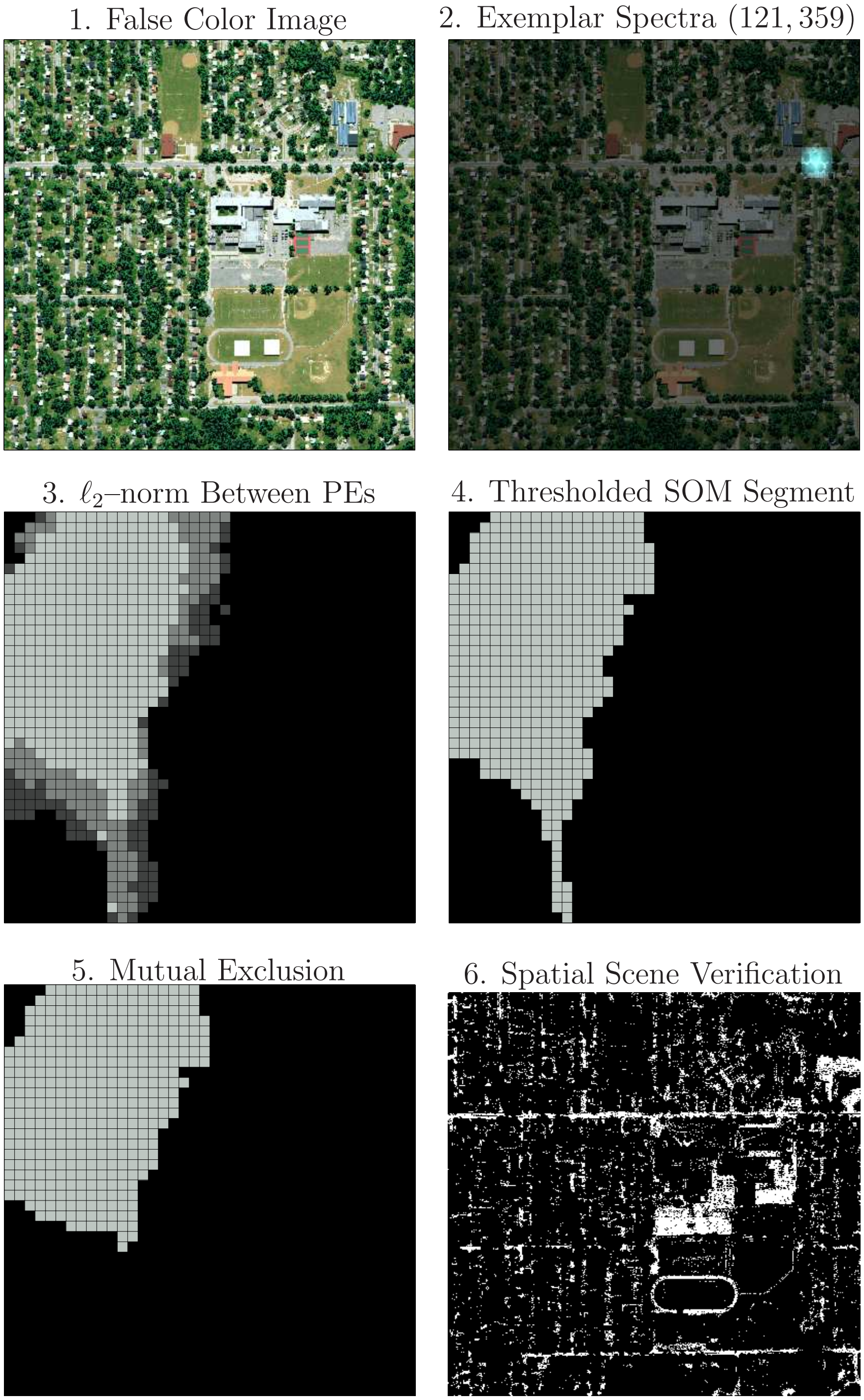
In Figure 3.3), the two “non-tree vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)” *SOM segmentations* are created by logically “deleting” (see Table 3.1) the “tree” *SOM segmentation* from each of the “vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)” *SOM segmentations.* This process is shown in Equation 3.1.

$$\text{(Non-Tree Vegetation SOM)} = \text{(Vegetation SOM)} - \text{(Tree SOM)} \tag{3.1}$$

This completes the segmentation of the vegetative topology of the SOM. The next step is to segment non-vegetative materials of interest. The approach to segmenting the non-vegetative topology is slightly different than that for the vegetative topology. In the new approach, the user selects exemplar spectra from the false color composite of the hyperspectral image, and the exemplar spectra are used to locate the SOM topological neighborhood for surrounding the exemplar spectra. The methodology is modular and any materials of interest may be selected by the user and developed into GIS layers. In this thesis, the scene content limits the needed material selection to “rooftop” and “roadway” materials.

3.2.5 “Rooftop” and “Roadway” SOM Segmentations.

It is assumed that *a-priori* knoweldge of endmember spectra, a spectral database, and specific scene knowledge is *not* available. User input is used to select exemplar spectra for materials of interest to further segment the SOM. The process to create a *SOM segmentation* from an exemplar spectra is illustrated in Figure 3.8. The false color composite of the hyperspectral image is displayed (Figure 3.8 step 1) and the one or more exemplar spectra are selected (Figure 3.8 step 2). The ℓ_2–norm is calculated between the exemplar spectra (or average of multiple exemplar spectra) and each SOM PE (Figure 3.8 step 3). A maximum allowable ℓ_2–norm distance threshold is set resulting in an initial *SOM segmentation* (Figure 3.8 step 4). The initial *SOM segmentation* is expected to be a contiguous region of adjacent PEs because of the topological neighborhood preservation property the SOM. The contiguous region may overlap PEs that are included in other *SOM segmentations*, which results in PEs with multiple labels. However, in this thesis the choice was made to make “roadway”, “rooftop”, “tree”, and “non-tree vegetation” *SOM segmentations* mutually exclusive by logically “deleting” (see Table 3.1) each of the previous SOM segmentations (Figure 3.8 step 5). A *temporary* object map (Figure 3.8 step 6) is used to manually adjust the ℓ_2–norm threshold, and steps 3-6 are repeated until the *temporary oject map* maximizes the material identification and minimizes the false positives. The results are the two “rooftop” and the “roadway” *SOM segmentations* shown in Figure 3.3.



1. False Color Image

2. Exemplar Spectra $(121, 359)$

3. ℓ_2–norm Between PEs

4. Thresholded SOM Segment

5. Mutual Exclusion

6. Spatial Scene Verification

Figure 3.8: This is the process to create a SOM segmentation from user input. 1. The user views a false color image of the hyperspectral scene. 2. User selects an exemplar spectra. 3. The ℓ_2–norm between the exemplar spectra and each SOM PE is calculated. 4. The user sets an initial threshold. 5. Previous SOM segmentations are removed to make them mutually exclusive. 6. The temporary object map is viewed by the user and used to adjust the threshold. Steps (2-6) are repeated until a satisfactory object map is created.

3.2.5.1 Summary of the Spectral Analysis Process. The spectral analysis process used NDVI to create two "vegetation" *SOM segmentations*. The NIR scatter tree identification algorithm, in combination with NDVI, produced a "tree" *SOM segmentation*, which is then logically "deleted" from the two "vegetation" *SOM segmentations* to produce two "non-tree vegetation" *SOM segmentations*. Four of these five vegetation-related SOM segmentations, the "vegetation (NDVI $>$ 0.3)" *SOM segmentation*, "non-tree vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)" *SOM segmentations*, and the "tree" *SOM segmentation* are outputs of the spectral analysis process. Non-vegetative SOM segmentation is accomplished using user-selected exemplar spectra. The ℓ_2–norm is calculated between the SOM PEs and the exemplar spectra and the ℓ_2–norm threshold used to isolate a material neighborhood is manually adjusted as in Figure 3.8. The results is the "rooftop" and "roadway" *SOM Segmentations* that are outputs of the spectral analysis process.

There are six *SOM segmentations* that result from the spectral analysis process, as shown in Figure 3.3. They are labeled "vegetation (NDVI $>$ 0.3)", "non-tree vegetation (NDVI $>$ 0.3)", "non-tree vegetation SOM (NDVI $>$ 0.5)", "tree", "rooftop", and "roadway" *SOM segmentations*. One way to visualize the SOM topological neighborhood is to assign a color to four of the SOM segmentations and overlay them as shown in Figure 3.9. In the next step, the six SOM segmentations are used to create six *initial object maps*.

3.3 Create the Initial Object Maps

Each of the six SOM segmentations that are output from spectral analysis are used as inputs to the *create the initial object maps* step. The process is illustrated in Figures 3.10 and 3.11. Six object maps with the same spatial resolution as the hyperspectral image are created, one for each SOM segmentation.

Each pixel of the hyperspectral image (Figure 3.11 step 1) is sampled in order, and the nearest neighbor PE in the ℓ_2–norm sense is located in each of the six SOM segmentations (Figure 3.11 step 2). If the sample lies in the binary "true" region of the

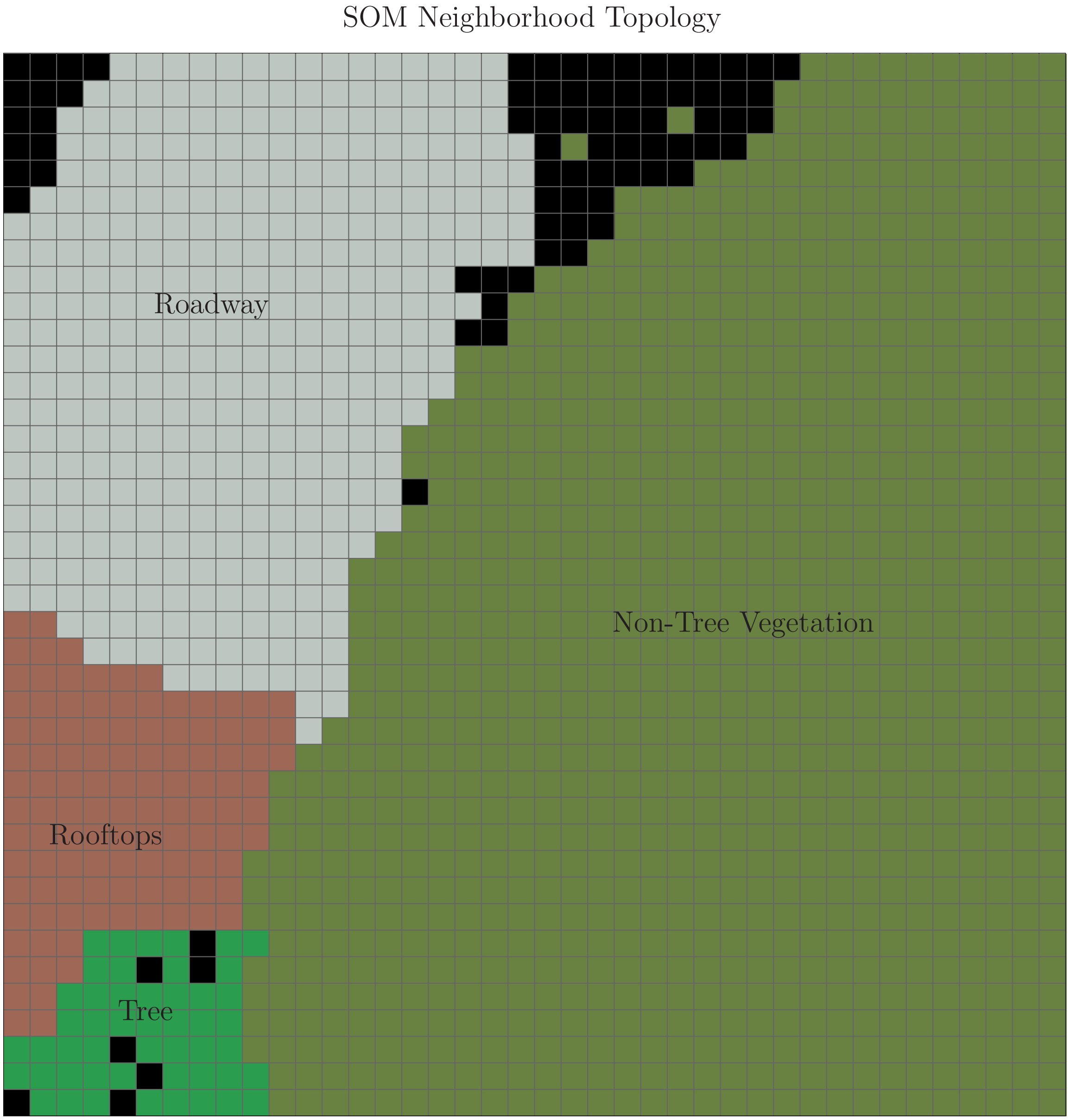


Figure 3.9: Four of the *SOM segmentations* resulting from spectral analysis are mutually exclusive: “roadway”, “rooftop”, “tree”, and “non-tree vegetation (NDVI $>$ 0.3)”. They are each given a color and overlaid, producing this topological view of the SOM neighborhoods.

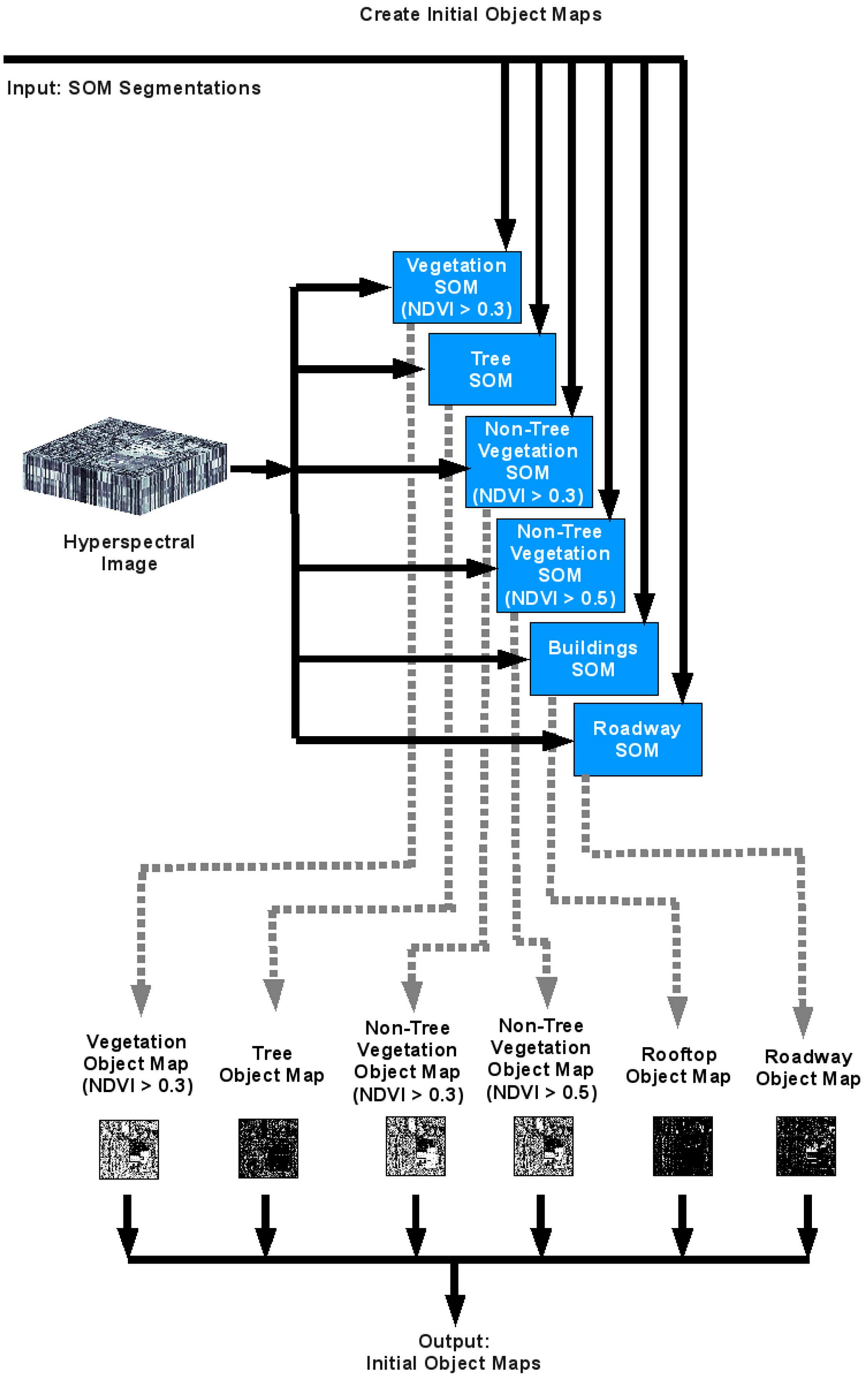

Create Initial Object Maps
Input: SOM Segmentations
Vegetation SOM (NDVI > 0.3)
Tree SOM
Non-Tree Vegetation SOM (NDVI > 0.3)
Non-Tree Vegetation SOM (NDVI > 0.5)
Buildings SOM
Roadway SOM
Hyperspectral Image
Vegetation Object Map (NDVI > 0.3)
Tree Object Map
Non-Tree Vegetation Object Map (NDVI > 0.3)
Non-Tree Vegetation Object Map (NDVI > 0.5)
Rooftop Object Map
Roadway Object Map
Output: Initial Object Maps

Figure 3.10: This is the process flow diagram for the *create the initial object maps* step. The nearest neighbor PE (in the ℓ_2–norm sense) for each *SOM segmentation* is located for each sample pixel of the hyperspectral image. The spatial location of the sample pixel is marked with the binary "true" or "false" in the corresponding *initial object map*, depending on the binary state of the *SOM segmentation*. One example SOM segmentation is shown in Figure 3.11.

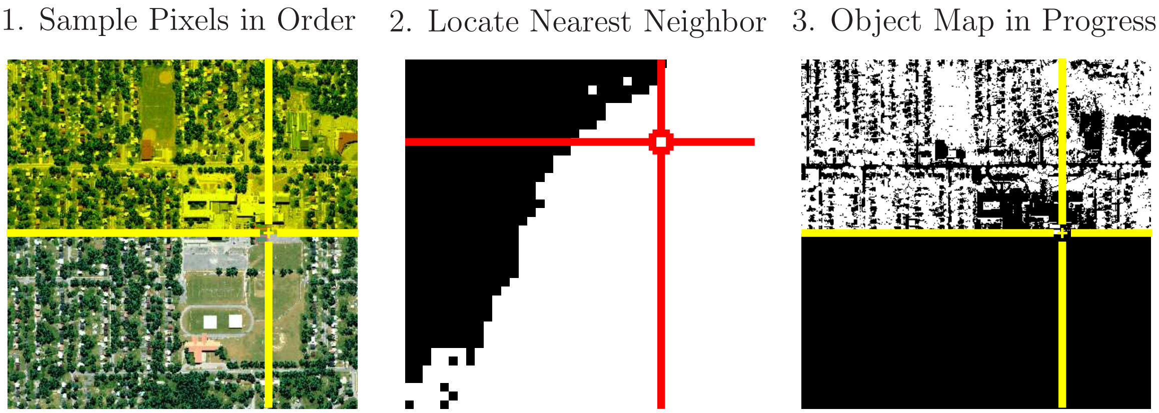

1. Sample Pixels in Order
2. Locate Nearest Neighbor
3. Object Map in Progress

Figure 3.11: An object map is created by selecting each pixel (1) and locating its nearest neighbor in the ℓ_2–norm sense in the SOM segmentation (2). If the nearest neighbor is a binary "true", then the corresponding object map is marked "true" (3). The result is a complete object map showing the location of a particular material type. Steps 1-3 are repeated for every pixel of the hyperspectral image.

SOM segmentation, the sample's spatial location is set to binary "true" in the object map that corresponds to the SOM segmentation (Figure 3.11 step 3). Steps 1-3 are accomplished for each of the six input SOM segmentations. The results are six *initial object maps* where each object map corresponds to one input SOM segmentation. The *initial object maps* are the outputs of the *create the initial object maps* step. They are next used in the spatial analysis process.

In summary, six *initial object maps* are created from six SOM segmentations. The *initial object maps* are the inputs to and form the basis of the spatial analysis process. Before spatial analysis is discussed, it is important to emphasize why the transition from spectral to spatial analysis is critical.

3.4 Transitioning from Spectral to Spatial Analysis

In traditional "pure" spectral analysis where minimal or no spatial analysis is accomplished, this is the end of the classification process. The goal of traditional spectral analysis is maximizing the correct labeling of hyperspectral image pixels as compared with the "ground truth" labeling provided with test images. Once each pixel in the hyperspectral image has a label, the accuracy of the labeling is calculated. The quality of the algorithm is assessed to be proportional to the accuracy. A higher classification accuracy insinuates a better algorithm. There are two problems with this understanding of accuracy. First, the "ground truth" hyperspectral image labeling rarely gives pixels more than one label. *Mixed pixels* are usually labeled with their dominant spectra. A partially tree-occluded building pixel might be labeled "tree" or "asphalt shingle", or dual-labeled "tree/asphalt shingle". The significant number of mixed pixels in an urban environment can lead to significant reductions in accuracy calculations despite correct labeling. Second, the dominant spectra labeling approach is in contrast to an abstract understanding of the scene content that requires some pixels to have multiple labels. Roadways that are occluded by tree foliage need to be labeled both "roadway" and "tree". Multiple labels reduces a pixel-level accuracy calculation as well. Third, correctly labeling individual pixels may not yield an abstract understanding of the scene. A "grass" pixel may be best described as a "field pixel".

In contrast to the traditional spectral analysis, this thesis continues the process with spatial processing of the images. Since the spatial analysis results in an improved pixel-level accuracy calculation, it is not necessary to maximize the accuracy during spectral analysis. An empirically-determined estimate is that a 75% accuracy at the end of spectral analysis yields excellent to outstanding results after spatial analysis. The benchmark 75% accuracy also justifies the use of the SOM, which is itself an abstraction of the high-dimensional hyperspectral image. It should be emphasized that the significant spatial processing with *logical* and *morphological* operations after spectral analysis is *the* unique contribution of this thesis to the state-of-the-art.

3.5 Spatial Analysis

The spatial analysis process shown in Figure 3.1 follows the *create the initial object maps* step. A hyperspectral image input as to the spectral analysis process results in six *SOM segmentation* maps that are processed into six *initial object maps* in the *create the initial object maps* step. The six *initial object maps* are used during

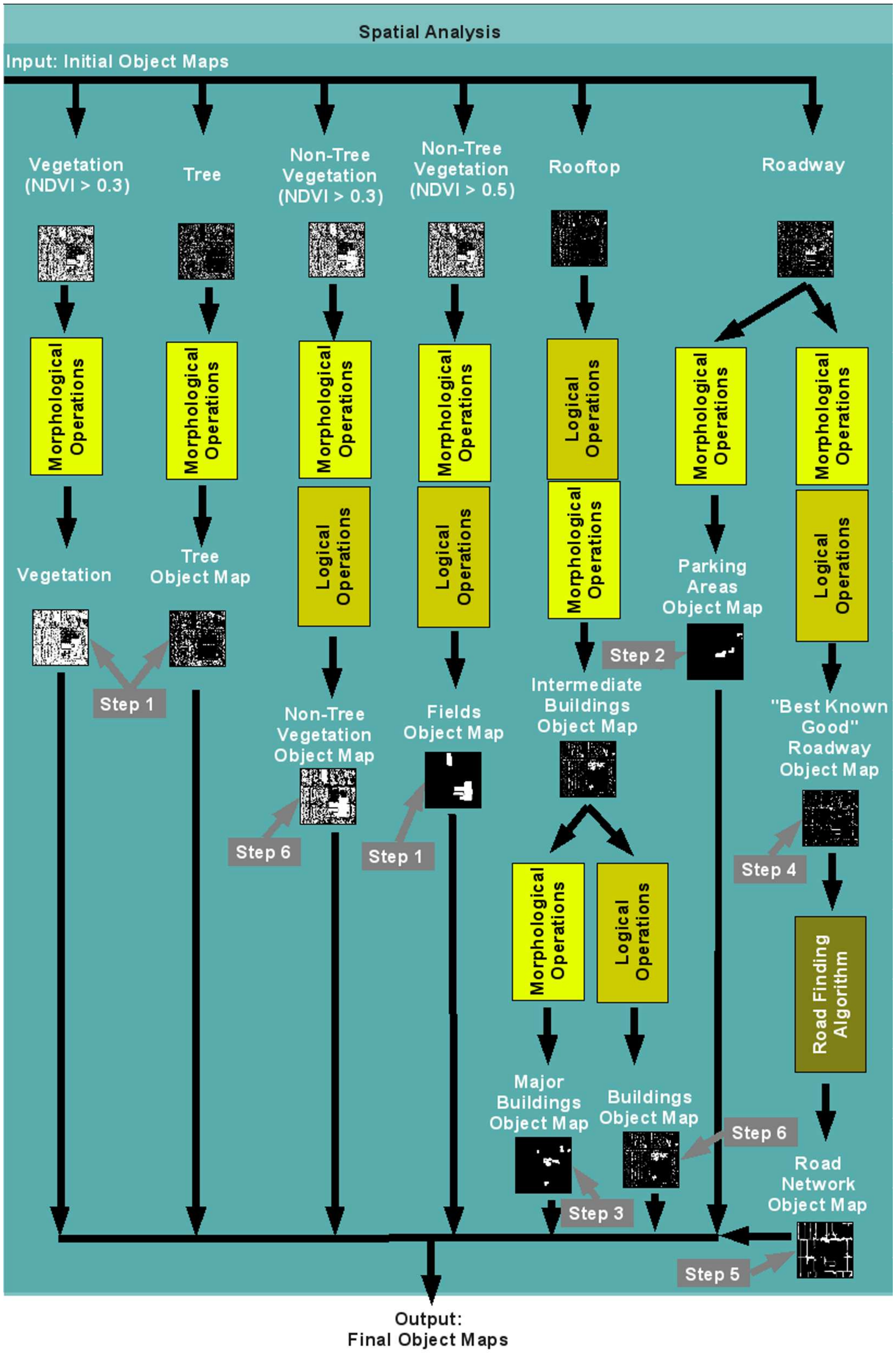


Figure 3.12: The process flow diagram for the spatial analysis process. The primary flow is columnar, with steps 1-6 specifying the order of accomplishment. Six input *initial object maps* (top row) are processed with a series of *logical* and *morphological* operations into eight *final object maps* (labeled steps 1-6). A novel *road finding algorithm* connects possible road segments under a significantly tree-occluded road network.

the spatial analysis process to produce eight *final object maps*. The *final object maps* do not correspond one-to-one with the *initial object maps* because some entirely new *object maps* are created from the *initial object maps*. The process of creating the eight *final object maps* is shown in Figure 3.12. The ordering of the spatial operations is important; however, the introduction of a massive set of dependency arrows in Figure 3.12 makes the visualization of the process impossible to follow. The spatial process steps are therefore enumerated in Figure 3.12 with labels "step 1", "step 2", ..., "step 6".

In step 1, the "vegetation (NDVI $>$ 0.3)" and the "tree" *initial object maps* undergo morphological operations to produce the "vegetation" and "tree" *final object maps*. The "non-tree vegetation (NDVI $>$ 0.5)" *initial object map* undergoes morphological operations and then logical operations to create the "fields" *final object map*.

In step 2, the "parking areas" *final object map* is derived from the "roadway" *initial object map* with iterative morphological operations. The "parking areas" *final object map* is used in creating the "buildings", "major buildings", "road network", and "non-tree vegetation" *final object maps* in steps 3-6.

In step 3, the "rooftops" *initial object map* undergoes logical and morphological operations to produce an intermediate "buildings" *object map*. The intermediate "buildings" *object map* is operated on by morphological operations to produce the "major buildings" *final object map*. The intermediate "buildings" *object map* is also used in step 6 to create the "buildings" *final object map*. The change in the label from "rooftops" to "buildings" is significant because the buildings object map contains several buildings not identified during spectral analysis.

In step 4, the "roadway" *initial object map* undergoes morphological operations and then logical operations to produce a "best known good roadway" *object map* which is only used as in input to the *road finding algorithm*. The "best known good roadway" *object map* is a minimal set of roadway pixels with a high probability of being correctly labeled.

In step 5, the *road finding algorithm* locates possible road connections despite a significantly tree-occluded road network. The result is the "road network" *final object map.*

In step 6, the "non-tree vegetation (NDVI > 0.3)" *initial object map* is processed with morphological and logical operations to produce the "non-tree vegetation" *final object map.* The intermediate "buildings" *object map* undergoes logical operations to produce the "buildings" *final object map.*

The spatial analysis process converts the six input *initial object maps* into eight *final object maps* in six steps that are labeled in Figure 3.12. Steps 1-6 are detailed below.

3.5.1 Step 1: The "Vegetation", "Tree", and "Fields" Final Object Maps.

In step 1, the "vegetation (NDVI > 0.3)", "tree", and "non-tree vegetation (NDVI > 0.5)" *initial object maps* are processed into the "vegetation", "tree", and "non-tree vegetation" *final object maps* as shown in Figure 3.12. The morphological and logical operations used to create each of the three *final object maps* are detailed below.

The "vegetation (NDVI > 0.3)" *initial object map* undergoes a morphological close operation with a 2×2 structuring element (SE). The size of the SE is chosen to preserve roadways. Roadways are a key distinguishing feature of the scene that a person might use to mentally "line-up" two maps. The "in-scene" roadways are, on average, 3 to 5 pixels wide. Therefore, an SE that is smaller than a typcal roadway in length and width does not remove a roadway, and the basic structure of the roadways are present after the morphological close operation. The result is the "vegetation" *final object map.*

The "tree" *initial object map* is closed with SEs that vary in size to correspond to the minimum and maximum expected size of an "in-scene" maple tree. The trees are estimated to be at least three pixels in diameter. The accuracy of the "tree" *object map* as compared to "ground truth" is calculated for an $n \times n$ SE size where $n = 1, 2, \ldots, 5$. The results are presented in Table 4.6. A 3×3 SE is selected because it yields the highest accuracy as compared to "ground truth" and because it provides a visually appealing *final object map.* The "tree" *initial object map* undergoes a morphological close with a 3×3 SE resulting in the "tree" *final object map.*

The "non-tree vegetation (NDVI > 0.5)" *initial object map* undergoes iterative morphological open and close operations, with an $n \times n$ SE where $n = 5, 6, \ldots, 20$. The starting and ending size of the SE are chosen in relation to the spatial size of "in-scene" fields. The urban backyard or front yard lawn is less than or equal to $n = 5$ pixels. The largest field is larger than $n = 20$ pixels. If the SE size is specified smaller than 5 pixels on a side, all of the urban lawns that are clustered together turn the scene into one very large field. This is not a desirable situation. The "fields" *object map* is then processed with a logical operation. The "roadway" *initial object map* is logically *deleted* (see Table 3.1) from the "fields" *object map* resulting in the "fields" *final object map.*

The results of step 1 are three *final object maps.* They are the "vegetation", "tree", and "fields" *final object maps.* Step 2 creates the "parking areas" *final object map* from the "roadway" *initial object map* using iterative morphological operations in a similar fashion to creating the "fields" *object map.*

3.5.2 Step 2: The "Parking Areas" Final Object Map.

In Figure 3.12 step 2, the "roadway" *initial object map* is processed into the "parking areas" *final object map.* The morphological operations used to create the map are detailed below.

The "roadway" *initial object map* undergoes iterative morphological open and close operations, with an $n \times n$ SE where $n = 2, 3, \ldots, 10$. The starting size SE, $n = 2$, is the smallest SE that may be used in morphological operations. The ending size of the SE is chosen in consideration of the spatial size of the "in-scene" parking areas.

The parking areas are larger than $n = 10$ pixels on a side. The result of this process is the "parking areas" *final object map.*

The "parking areas" *final object map* is created prior to the "road network" *final object map* assuming that the two object maps are mutually exclusive. Mutually exclusive here means that the road finding algorithm assumes that roadways do not cross parking areas. If a road segment passes through a parking area, the road finding algorithm (as implemented) will not find the road segment.

3.5.3 Step 3: The "Major Buildings" Final Object Map.

In Figure 3.12 step 3, one *initial object map* is processed into one *final object map.* An intermediate *object map* used during step 6 is created. Figure 3.12 shows the conceptual splitting of step three into two pieces. First, the "rooftop" *initial object map* is processed with logical operations and then morphological operations to produce an intermediate "buildings" *object map.* In this step, the intermediate "buildings" *object map* is processed with morphological operations to produce the "major buildings" *final object map.* In step 6, the intermediate "buildings" *object map* is processed with logical operations to produce the "buildings" *final object map.* The morphological and logical operations used to create the intermediate "buildings" *object map* and the "major buildings" *final object map* are detailed below.

3.5.3.1 The Intermediate "Buildings" Object Map.

The intermediate "buildings" *object map* is used to create the "buildings" and "major buildings" *final object maps* in steps 3 and 6, respectively. The intermediate "buildings" *object map* is created through the use of a Negative Image Mask (NIM) in conjunction with logical and morphological operations.

3.5.3.2 The Negative Image Mask.

The NIM is, conceptually, where one would not expect to find a building. It is constructed with logical operations followed by a morphological operation. A logical "or" operation is used to "add" the "vegetation (NDVI > 0.3)" and "roadway" *initial object maps*, and the "fields" and "parking areas" *final object maps.* The result of the logical "or" operation is morphologically closed with a 3×3 SE, resulting in the NIM. The size of the SE is chosen based on the approximate minimum size of an "in-scene" house. The smallest size house is estimated at 3×3 pixels. In the final step, the NIM is logically "inverted" and "added" to the "rooftops" *initial object map* with a logical "or" operation. The result is the intermediate "buildings" *object map.*

The reason a NIM is used rather than operating directly on the "rooftops" *initial object map* is to preserve information about buildings. For example, logically "deleting" (see Table 3.1) vegetation from the "rooftop" *initial object map* would partially remove some tree-occluded rooftops. This is the reason the "tree" *final object map* is not "added" to the NIM. Another example is the morphological operation. If the morphological close on the NIM is replaced instead with a morphological open on the "rooftop" *initial object map*, the result is again the deletion of some rooftop pixels. This happens because some of the rooftop pixel clusters are smaller than the SE, a situation that happens when a rooftop is occluded by tree foliage. The effect of using the NIM concept is that the logical and morphological operations do *not* occur on the *object map* and valuable rooftop information is retained.

By combining the inverse of the non-building object maps with the "rooftop" *initial object map*, the "rooftop" *initial object map* now has all of the information that is known about in-scene buildings. It is therefore relabeled "rooftop" to "buildings". The result is next processed with morphological operations.

3.5.3.3 Intermediate "Buildings" Object Map.

The last step to complete the intermediate "buildings" *object map* is to perform one morphological operation on the map. The map is morphologically opened with a 3×3 SE. The size of SE is based on the "in-scene" minimum size of a building where that size is at least three pixels on a side. The result is the intermediate "buildings" *object map.* This map is used next to create the "major buildings" *final object map*, and in step 6 to create the "buildings" *final object map.*

3.5.3.4 Creating the Final Object Map.

The intermediate "buildings" *object map* is processed with three morphological operations. First, the map is morphologically opened with a 7 × 7 SE. This size is chosen to delete the "in-scene" houses. The maximum house size is estimated to be six pixels on a side. The SE is guaranteed to delete houses while leaving any major buildings that are larger than the SE. Second, the map is morphologically dilated with a 5 × 5 SE. This SE size is chosen because it adds two pixels to the edges of the remaining major buildings. This operation is purely to improve the visual appeal of the major buildings map. The "fatter" buildings show up better when overlaid on a scene. The third operation is a morphological close with a 3 × 3 SE. The size is chosen to be significantly smaller then the first open operation, in order to least affect the shape of the buildings. Its effect is to "close" any "holes" in the major buildings objects that are smaller than the SE in size, and resuls in a building object that does not have "holes" in it. The purpose is also purely for aesthetic appeal. Complete and uniform objects are visually more appealing and less disruptive to visual continuity. The result of these logical and morphological operations is the "major buildings" *final object map.*

3.5.4 Step 4: The "Best Known Good Roadway" Object Map.

In Figure 3.12 step 4, the "roadway" *initial object map* is processed into the "best known good roadway" *object map.* The "best known good roadway" *object map* is used only in step 4 with the *road finding algorithm.* The *road finding algorithm* finds possible road connections and adds them to the "best known good roadway" *object map.* The algorithm, as implemented, never removes non-road segments. For this reason, if the "best known good roadway" *object map* contains too many non-roadway pixels (false positives), the result is that many road connections are made that do not actually exist. For this reason, the "best known good roadway" *object map* is a significant reduction of the "roadway" *initial object map* using morphological operations followed by logical operations.

3.5.4.1 Morphological Operations. The initial roadway object map is reduced using one morphological open operation for each SE where the SE sizes are 2×1, 1×2, 2×2 northwest-southeast diagonal, 2×2 northeast-southwest diagonal, and then 3×3. Previously created object maps are next removed with logical operations.

3.5.4.2 Logical Operations. The "best known good roadway" *object map* is operated on by a logical "deletion" (see Table 3.1) to further remove non-roadway pixels. The "parking areas" and "major buildings" *final object maps* are logically "deleted" from the object map. The result is labeled the "best known good roadway" *object map* in Figure 3.12. The "best known good roadway" *object map* is a minimal roadway surface object map. It is used directly in the next step to make possible roadway connections.

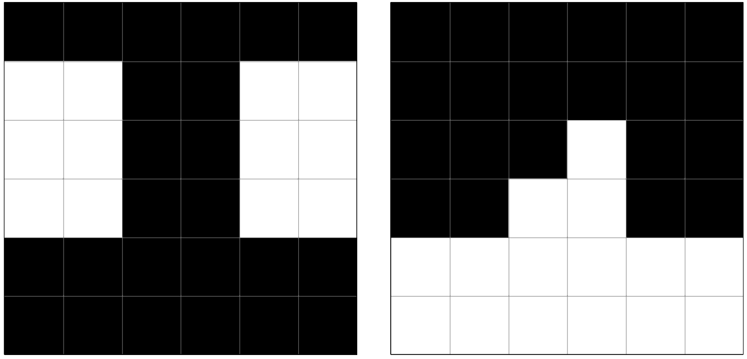
3.5.5 Step 5: The "Road Network" Final Object Map*.* In Figure 3.12 step 5, the "best known good roadway" *object map* is processed into the "road network" *final object map.* The *road finding algorithm* first constructs a NIM as in Section 3.5.3.2. The NIM is gradually "pealed" to reveal possible road connections, and the possible road connections are added to the "best known good roadway" *object map.*

3.5.5.1 The Negative Image Mask. The details of constructing a NIM are discussed in Section 3.5.3.2. The concept is to build the NIM from non-roadway object maps that have been created in steps 1-3. The NIM is constructed with the "non-tree vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)" *initial object maps*, "fields", "parking areas", "major buildings" *final object maps* and the intermediate "buildings" *object map.* The reason they are listed in this order is that they are logically "deleted" (see Table 3.1) in the reverse order during the road finding portion of the algorithm. The last step is to morphologically close the NIM with a 2×2 SE. The size of the SE is selected so that roadways, which are approximately three to four pixels in width, are not removed from consideration.

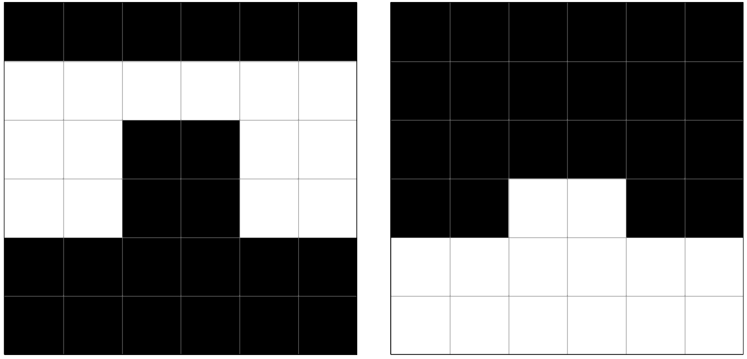
```
Inputs:
    ‘‘Best Known Good’’ Roadway Object Map
    NIM
    ‘‘NIM Threshold Max’’

Start:
    roadway  = ‘‘Best Known Good’’ Roadway Object Map
    [roadway_columns,roadway_roads] = size(roadway)
    NIM_threshold = ‘‘NIM Threshold Max’’

    % 15 is experimentally determined number of iterations
    % to uncover all roads.
    for index = 1:15

        % threshold is experimentally determined
        for NIM _threshold = 0:NIM_threshold_max

            % iterate through all M,N pairs of pixels in the roadway map
            for M = 1:roadway_columns
            for N = 1:roadway_rows

                % If the pixel is a roadway
                if roadway(M,N) == ‘‘true’’

                    % find the next roadway pixel
                    [M_next, N_next] = find_next_roadway_pixel(roadway,M,N)

                    % if there is at least one pixel distance exists
                    if sqrt((M-M_next)^2 + (N-N_next)^2) >  1

                        % Compare the SUM to threshold
                        if sum(NIM(M,N,M_next,N_next)) < NIM_theshold
                            draw_roadway(roadway,M,N,M_next,N_next)
                        end
                    end
                end
            end
            end
        end
    end
```

Figure 3.13: The pseudocode for the road finding algorithm. This algorithm uses a negative image mask and a “best known good” roadway object map to produce the road network final object map.

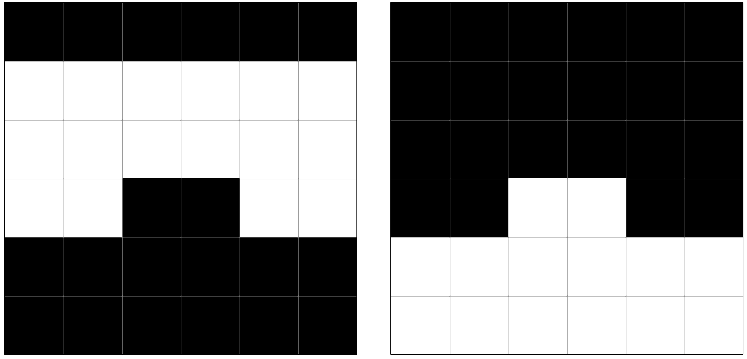
1. Initial Roadway Map 2. NIM Threshold = 0



3. Possible Roads 4. Open 2 × 2 SE



5. Possible Roads 6. NIM Threshold = 2



7. Possible Roads 8. Open 3 × 3 SE

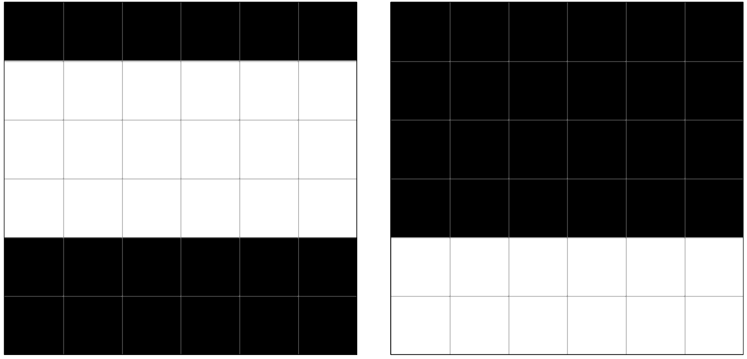


Figure 3.14: This sequence illustrates the iterative process the road finding algorithm uses to build the roadway network layer. The negative image mask (right column) is gradually "pealed" while the possible road connections are made on the "best known good roadway" *object map* (left column).

3.5.5.2 The Road Finding Algorithm.

The road finding algorithm operates iteratively. It takes as inputs the "best known good roadway" *object map*, the NIM, and a NIM threshold max. All pixels marked "true" in the "best known good roadway" *object map* are assumed roadway pixels. All the pixels marked as "true" in the NIM are assumed non-roadway pixels. The pseudocode is presented in Figure 3.13. A pictorial view of the process is in Figure 3.14.

First, the road finding algorithm considers every pair of roadway pixels that are separated by a minimum euclidean distance of at least one pixel. For each pair, it calculates the number of non-roadway pixels between those two same locations from the NIM[3]. Second, the algorithm compares the sum of the non-roadway pixels to a *NIM threshold.* Based on the results of the comparison, the algorithm draws a possible road connection between the roadway pixels on the "best known good roadway" *object map.* Third, the algorithm increases the NIM threshold by one. If the maximum NIM threshold is reached, then the algorithm resets the NIM threshold count and "peals" information from the NIM.

The "pealing" process of removing information from the NIM warrants a brief explanation. Once the NIM is constructed, "pealing" is accomplished in two ways. First, a morphological open operation with the smallest possible SE (2×1 or 1×2) reduces stray pixels, and increases the number of possible road segments. The largest size SE is experimentally determined to be 3×3 before the *road finding algorithm* makes too many non-road network connections. Second, the NIM is re-created, but with one less *object map* than before. The intermediate "buildings" *object map* is one of the first not to be included when the NIM is re-created because it contains several roadway pixels. The inclusion of the roadway pixels in the "buildings" *object map* prevents the *road finding algorithm* from finding some possible road network segments. Using morphological operations on the NIM and re-creating the NIM with

---

[3] An expansion of this algorithm is to calculate the number of roadway pixels between the pair of pixels that the algorithm is considering. This could be used in a momentum calculation.

less *object maps* "peals" the NIM, revealing an increasing number of possible road segment connections to the *road finding algorithm.*

In summary, step 5 of the spatial analysis process results in a "road network" *final object map.* It is derived from the "roadway" *initial object map.* The *road network algorithm* has the largest number of parameters to tune than any part of the spatial analysis process. The tunable parameters are the construction of the NIM, the NIM threshold, and the order that information is "pealed" from the NIM. The road algorithm is modular and lends itself well to improving the quality of the road network produced. Some extension ideas include using a ruleset of how roads can connect, morphological profiles, edge detection, linear regression, and multiple other spatial processing techniques.

3.5.6 Step 6: The "Non-Tree Vegetation" and the "Buildings" Final Object Maps.

In Figure 3.12 step 6, the "non-tree vegetation (NDVI > 0.3)" *initial object map* and the intermediate "buildings" *object map* are processed into their final forms. The "non-tree vegetation (NDVI > 0.3)" *initial object map* undergoes two morphological operations followed by four logical operations. The two logical operations are a morphological close with a 3×3 SE and then a morphological open with a 3×3 SE. The close operation "fills in" the objects to make it them visually appealing, and the open operation deletes stray clusters of pixels that distract the eye. The four logical operations are deletion operations (see Table 3.1). The "road network", "buildings", "parking areas", and "fields" *final object maps* are deleted from the "non-tree vegetation (NDVI > 0.3)" *initial object map.*

The intermediate "buildings" *object map* undergoes one deletion operation. The "road network" *final object map* is deleted, resulting in the "building" *final object map.* The reason the road network is deleted from the buildings map is that they are not mutually exclusive. The spectral similarity between asphalt rooftop shingles and asphalt roadway resulted in some roadway pixels being included in the "rooftop" *initial object map.* The "road network" *final object map* is used to delete these stray roadway pixels that also exist in the "buildings" *final object map.*

3.5.6.1 The Final Object Maps. The results of the spatial analysis process and the six separate processing steps are eight *final object maps*: the "vegetation", "tree", "non-tree vegetation", "fields", "buildings", "major buildings", "parking areas", and "road network" *final object maps.* These eight *final object maps* are produced from six input *initial object maps*: "vegetation (NDVI > 0.3)", "tree", "non-tree vegetation (NDVI > 0.3 and NDVI > 0.5)", "rooftop", and the "roadway" *initial object maps.* The six steps from initial to final object maps are a phased approach to accomplishing the spatial analysis process. The last task to complete is to convert the eight final object maps into eight GIS layers.

3.6 Finalizing the GIS Layers

This is the last step in converting a hyperspectral image into abstract GIS layers. The hyperspectral image from Figure 3.1 is processed spectrally to produce six initial object maps, which are then processed spatially to produce eight final object maps. Figure 3.15 illustrates the process of creating the eight finalized GIS layers from the eight final object maps.

Each of the final object maps resulting from the spatial analysis process is turned into a GIS layer with three steps. These steps increase the visual appeal of the object maps, but do not change the content or improve the accuracy. First, a color is selected for each map. Second, the object map undergoes a morphological *remove*[4] operation followed by a morphological *dilation* operation with a 3×3 SE. The remove operation creates an "outline" of any objects in the object map, and the dilation operation "thickens" the outline by one pixel on each side of the outline. An RGB "mask" is created with the final object map and a mask color is selected.

---

[4] The morphological *remove* operation creates an outline of the objects in the image.

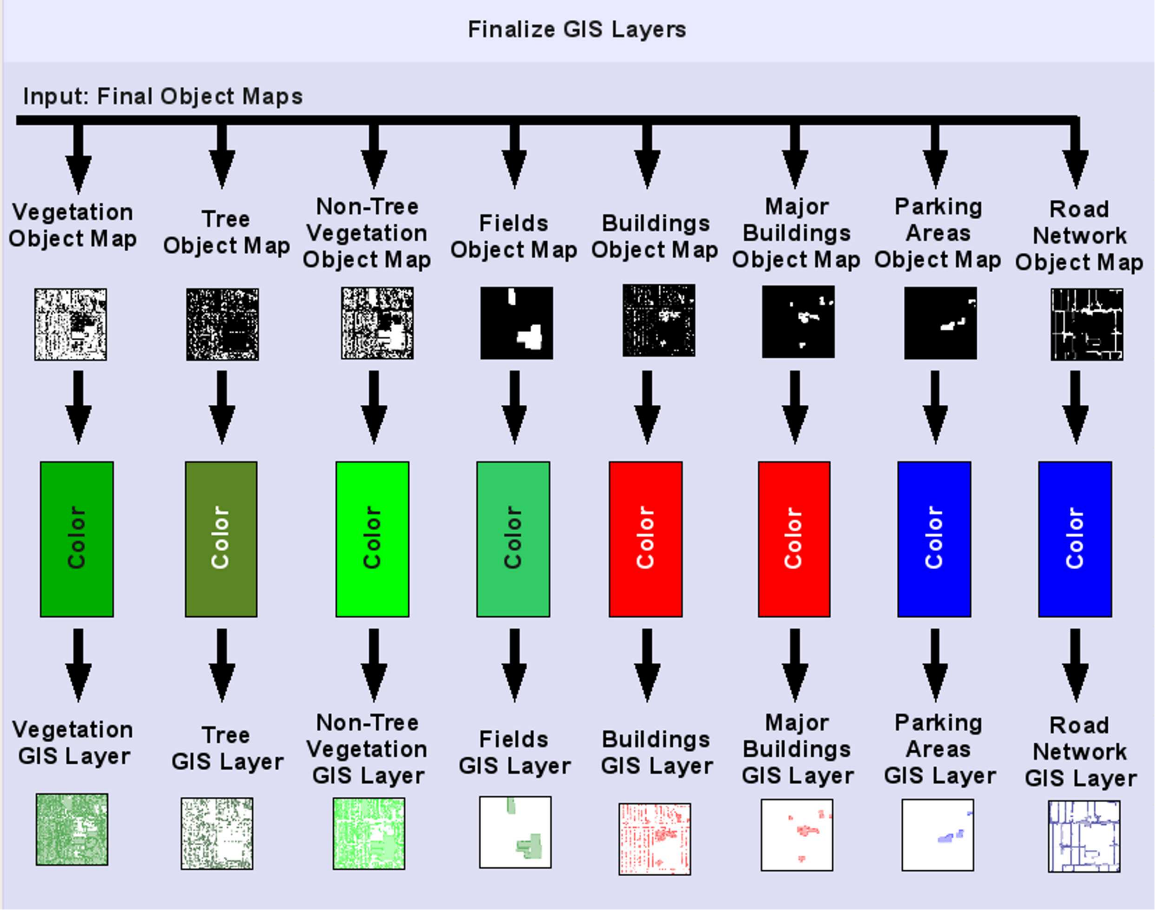


Figure 3.15: This is the process flow diagram to convert a hyperspectral image into multiple GIS layers (identified with yellow outlined boxes).

When the mask and outline are used in conjunction with a transparency setting, the maps can be overlaid over the original scene to show the locations of objects. These overlays are the *final GIS layers*.

In summary, the process from Figure 3.1 to convert a hyperspectral image into GIS layers is completely described. A hyperspectral image (Figure 3.2) is processed spectrally (Figure 3.3) to produce six *SOM segmentations*. Spectral analysis is accomplished with a SOM (Figure 3.4). The SOM is segmented using NDVI (Figure 3.5), the NIR scatter tree ID (Figure 3.6), and user input (Figure 3.8). The six *SOM segmentations* are converted into six *initial object maps* (Figures 3.10 and 3.11). The *initial object maps* undergo spatial processing (Figure 3.12) to yield eight *final object maps*. Spatial analysis is accomplished using logical and morphological operations. A novel road finding algorithm (Figures 3.13 and 3.14) is used to assemble the "road network" *final object map*. The *final object maps* are then finalized into eight GIS layers (Figure 3.15). With the process from hyperspectral image to GIS layers thus described, it is applied to a synthetic hyperspectral image in Chapter IV.

IV. Experimental Results and Analysis

Chapter IV presents the experimental results when the methodology developed in Chapter III is applied to a synthetic hyperspectral image. The methodology to convert the hyperspectral image into GIS layers is illustrated in Figure 3.1. Eight GIS layers are created from the hyperspectral image in four steps. The four steps are spectral analysis (Figure 3.3), creating the initial object maps (Figure 3.10), spatial analysis (Figure 3.12), and finalizing the GIS layers (Figure 3.15). Each step of the process presents the results generated in that step and an analysis of the results.

4.1 Process Inputs

The process input is the synthetic hyperspectral image represented in Figure 3.2. The hyperspectral image is generated with a custom simulation software program called the Digital Imaging and Remote Sensing Image Generation (DIRSIG) [37]. A satellite Red Green Blue (RGB) image (Figure 4.1, left image) was used as a layout template to create a synthetic scene called Megascene1. The synthetic scene is compared to a false color composite image (Figure 4.1, right image) in Figure 4.1. The Megascene1 is simulated with the DIRSIG software to produce the hyperspectral image (Figure 3.2). The simulation models atmospheric distortion, which necessitates *atmospheric correction.*

4.1.0.2 The Digital Imaging and Remote Sensing Image Generation (DIRSIG) Software. DIRSIG is a software program that was started in the late 1980's at the Rochester Institute of Technology (RIT) in Rochester, NY. The program predicts imager radiance values using a physics model and ray tracing. The physics model takes into account thermal characteristics, atmosphere properties and distortion, hyperspectral imager features and noise, the sensor platform, and the material types [37]. The DIRSIG modeler uses actual end-member spectra for in-scene materials, and calculates the spectral response. The software is used by civilian, academic, and defense agencies. The resolution of the hyperspectral image data is estimated to be approximately $1.5 \times 1.5m$ per pixel and is 400×400 pixels in size. The hyperspectral image

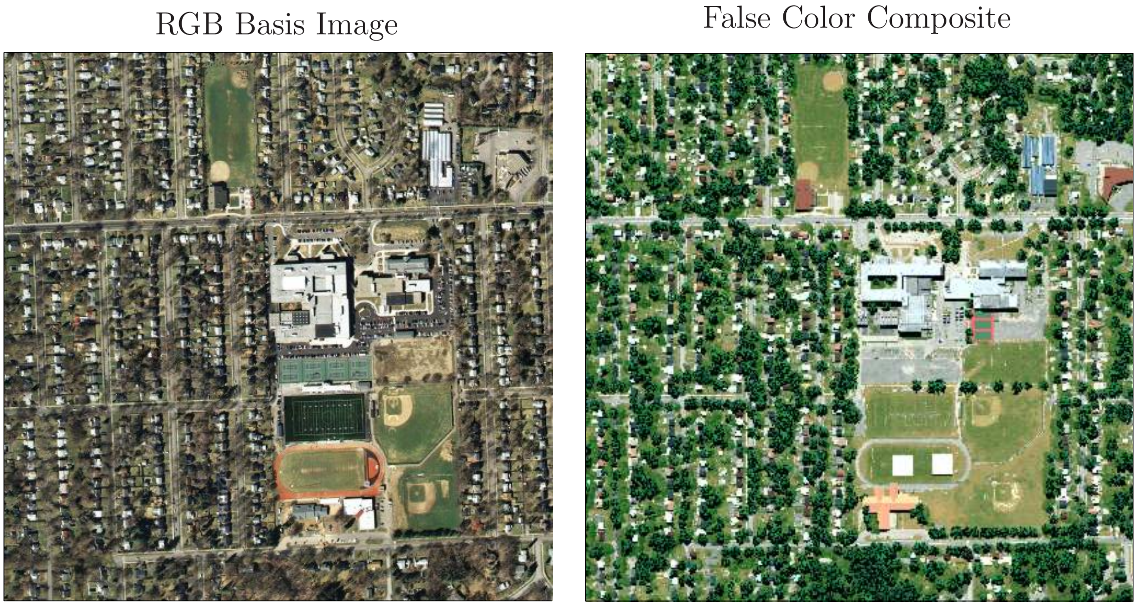


Figure 4.1: The satellite RGB image (left) is compared to the false color composite image (right) of the hyperspectral image. The satellite image was used as a layout reference to create the layout of Megascene1, from which the hyperspectral image is generated. The similarities and differences are discussed in Section 4.1.0.3.

is spectrally resampled to reflect a NASA/JPL AVIRIS acquired image. The layout of Megascene1 scene differs slightly from the satellite image as shown in Figure 4.1.

4.1.0.3 Comparing the Satellite Image and Megascene1. The layout of Megascene1 is based on the actual area surrounding the RIT campus. The RIT campus, located at approximately lattitude $43^\circ\,12'\,50''$ and longitude $-77^\circ\,12'\,60''$, is shown in the left image of Figure 4.1. The right image of Figure 4.1 is a false color composite of the hyperspectral image.

There are two significant discrepancies between the two images. First, the running track is the traditional orange rubber material in the satellite image and it is asphalt in Megascene1. Second, the large, light-colored building in the center of the image does not have a courtyard in the satellite image, but it does in the Megascene1. These discrepancies are significant because they affect the results when the results are compared with the satellite image. In the final GIS layer for trees, the courtyard of the light-colored building contains trees. In the final GIS layer for the road network, the running track is included as a roadway because it is the same asphalt material as the roadways. It would normally not be included in the GIS layer if it were the orange rubber material.

There are two notable similarities. First, the number of roads connecting to the edges of the image (seven on the top, four on the right, five on the bottom, and five on the left) are the same. Second, thre are five major buildings (six if the light-colored central building is counted as two), two large field areas, and four separate parking areas. These similarities are notable because they are used to verify the results of the process.

The satellite RGB image (Figure 4.1, left image) is not used as an input to the process, nor is it directly related to the data content of the hyperspectral image. Before the process can begin, the synthetic hyperspectral image that results from simulating the Megascene1 with the DIRSIG software needs to be atmospherically corrected.

4.1.0.4 Atmospheric Correction. Empirical line correction is a technique to correct for atmospheric disturbances, which are dominated by linear distortion effects. The DIRSIG software simulates atmospheric disturbances as part of its physics model, and these disturbances removed prior to processing. Two materials with known spectra were placed in the Megascene1 prior to simulation. The two materials are the two light-colored tarps that can be seen in the middle of the running track in the left image of Figure 4.1.

Once the Megascene1 is simulated, linear regression is performed on the resulting spectra for the two tarps and their known spectra. The result of the linear regression is a gain and an offset value. Equation 4.1 is used to correct the hyperspectral image spectra. Bands 105 to 107 and 140 to 150 are deleted due to scattering from atmoshperic water. These bands correspond to wavelengths $1.38\mu m$ to $1.40\mu m$ and $1.82\mu m$ to $1.94\mu m$. The last two bands, 209 and 210, with corresponding wavelengths

$2.48\mu m$ and $2.50\mu m$, are removed because of high noise. Therefore, 194 bands remain for analysis.

$$\text{Reflectance} = \text{Gain} \times \text{Radiance} + \text{Offset} \tag{4.1}$$

4.2 Spectral Analysis

The spectral analysis process flow is illustrated in Figure 3.3 and discussed in Section 3.2. The hyperspectral input image (Figure 3.2) is processed with a self-organizing map (SOM) to organize the image spectra. The normalized difference vegetation index (NDVI), near infrared (NIR) scatter tree identification algorithm, and user input are used to segment the converged SOM. The results of the segmentation process are six *SOM segmentations*. The six *SOM segmentations* are propogated from the spectral analysis process to the next step, the *create the initial object maps* step.

4.2.1 Self-Organizing Map. The SOM requires seven input training parameters before it can organize the spectra from the hyperspectral scene: the SOM PE lattice size, the number of learning schedule decay steps, the learning rate max and min, the neighborhood function max and min, and the number of training steps. These input parameters are listed in Table 4.1. The processing elements (PEs) lattice size (40×40 PEs) contains two orders of magnitude fewer processing elements than the spatial size of the hyperspectral image (400×400 pixels). This PE lattice size exponentially reduces processing time from days to hours. The number of decay steps, initial and final learning rate (α), and the initial and final neighborhood function update (σ) are "rule-of-thumb[1]" values that are known to provide good experimental results [22]. The number of training steps is suitably large to ensure that the SOM reaches convergence.

---

[1] In [7], the authors developed a parameter-less SOM (PLSOM). Choosing parameters is prone to error, and their PLSOM may allow the SOM input parameter selection to be fully automated.

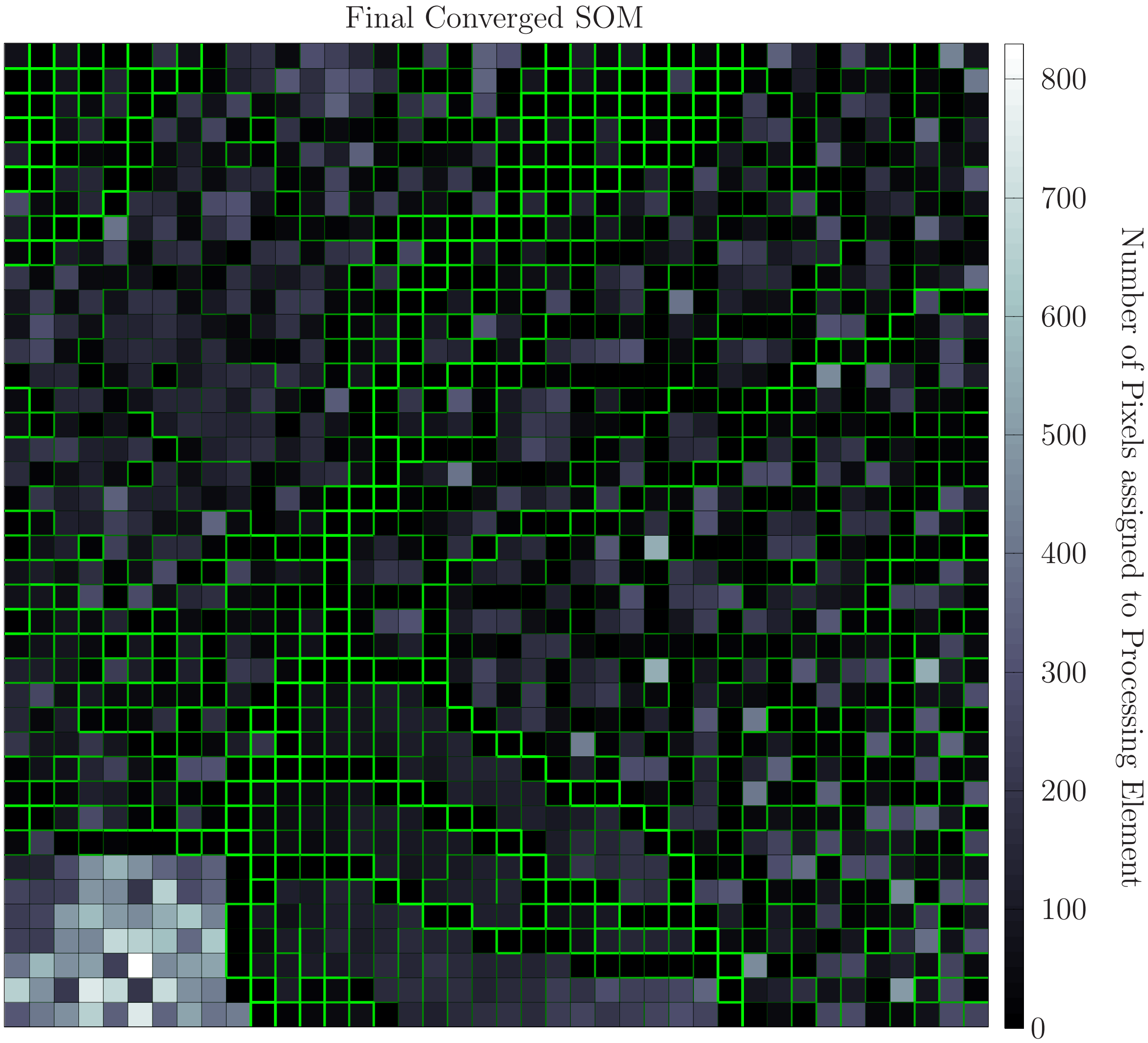


Figure 4.2: The converged SOM presented in alternative U-matrix form showing the "density" of pixels assigned to processing element (PE) where the density is proportional to the brightness of the PE. The green "fences" are proportional in brightness and thickness to the ℓ_2–norm distance between adjacent PEs.

Table 4.1: Listing of the SOM training parameters.

Variable	Value
SOM Lattice Size	40×40 PEs
Learning Schedule Decay Steps	20
Initial Learning Rate	$\alpha_{\text{initial}} = 0.8$
Final Learning Rate	$\alpha_{\text{final}} = 0.5 \times 10^{-4}$
Initial Neighborhood Function Update	$\sigma_{\text{initial}} = 1$
Final Neighborhood Function Update	$\sigma_{\text{final}} = 0.5 \times 10^{-4}$
Training Steps	6×10^6

Band 100 of the SOM is randomly chosen to be examined, and each PE value is assigned a color, resulting is Figure 3.4. The gradation of PE values (i.e. PE colors) is a positive sign that the SOM has completed successfully. If the SOM were to converge in a "twisted" state, the gradation of values would be significantly altered and not appear uniform as in Figure 3.4. The converged SOM is presented in Figure 4.2 in an alternative U-matrix form showing the "density" of pixels where the brightness of a PE is proportional to density. The density is the number of pixels assigned to the PE. The SOM has 40×40 or $1,600$ PEs. The hyperspectral image has 400×400 or $160,000$ pixels. There is, on average, 100 pixels/PE. 987 PEs have less than the average number of pixels assigned, and 613 PEs have greater than (or equal to) the average number of pxiels assigned. The maximum number of pixels assigned to a PE is 830, and the minimum is 0. These SOM statistics indicate the SOM has converged to a steady state. The green "fences" in Figure 4.2 are proportional in brightness to the ℓ_2–norm distance between adjacent PEs.

The gradation of PE values in band 100 of the SOM, the distribution of the density, and the fences that outline contiguous topological neighborhoods indicate that the SOM has converged successfully and is ready for segmentation. The segmentation of the SOM topology involves NDVI, NIR scatter tree ID, and user input. The results are several SOM *segmentations*. A SOM segmentation is a binary valued matrix with the same size as the SOM PE lattice where a logical "true" indicates presence of the material of interest. For each material of interest, one or more segmentations are created. First, vegetation is segmented using the NDVI.

4.2.2 Normalized Difference Vegetation Index. NDVI (see Section 2.2.9) is applied to the converged SOM from Figure 4.2 resulting in Figure 4.3. Two NDVI thresholds (NDVI $>$ 0.3 and NDVI $>$ 0.5) from Table 2.1 are used to create two *SOM segmentations*. Table 2.1 indicates that the threshold (NDVI $>$ 0.3) is the lower confidence limit of the presence of vegetation, and the threshold (NDVI $>$ 0.5) is the "safe" confidence limit for the presence of vegetation. The ordered NDVI values of the PEs in Figure 4.3 are plotted in Figure 4.4. The purpose of Figure 4.4 is to visually inspect and the NDVI thresholds from Table 2.1 and support the two NDVI thresholds (NDVI $>$ 0.3 and NDVI $>$ 0.5). Figure 4.4 is analyzed in relation to the NDVI thresholds from Table 2.1.

1. NDVI $>$ 0.8 a plateau in the plot at "definitely vegetation".
2. NDVI $>$ 0.5 a natural break that is indicates a possible dividing point between healthy vegetation and grass that is starting to seness.
3. NDVI $>$ 0.3 the "rule-of-thumb" for vegetation.
4. NDVI $>$ 0.1 the soil break indicates that there are likely mixed grass and soil spectra.

The results of this analysis point to the two thresholds (NDVI $>$ 0.3 and NDVI $>$ 0.5). The next step is to create a temporary object map using the process from Figure 3.11 to visually verify the segmentations using each of the four *SOM segmentation* that result from each of the four NDVI thresholds under consideration. This is accomplished in Figure 4.5.

The *temporary object map* for the threshold NDVI $>$ 0.8 in the top right image of Figure 4.5 does not encompass the vegetation content of the scene. The *temporary*

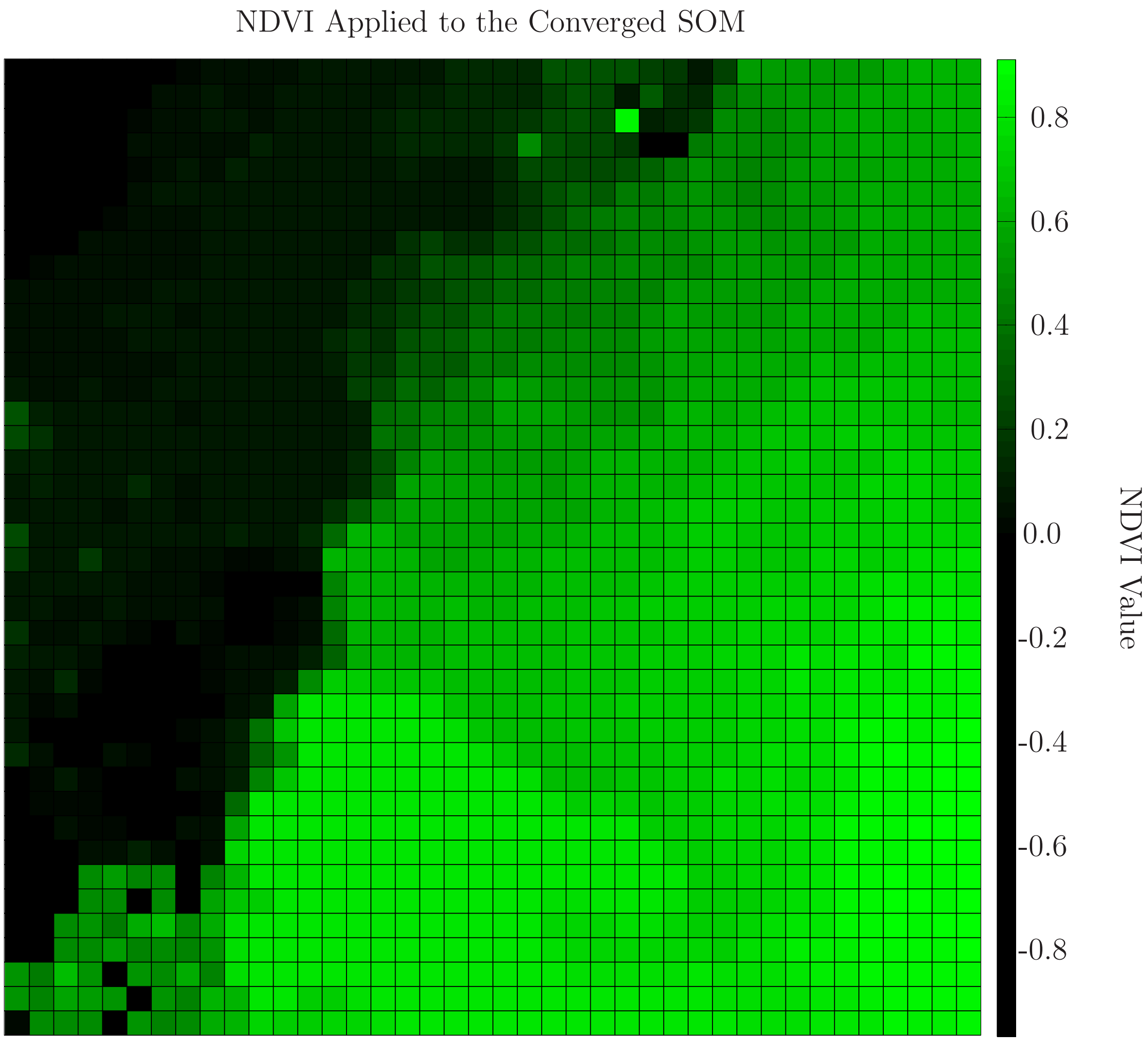


Figure 4.3: The result when NDVI is applied to the converged SOM from Figure 4.2. The color map has been adjusted to fade from green to black at the lower limit of confidence, NDVI > 0.3. The brighter the pixel, the higher the confidence that it is vegetation.

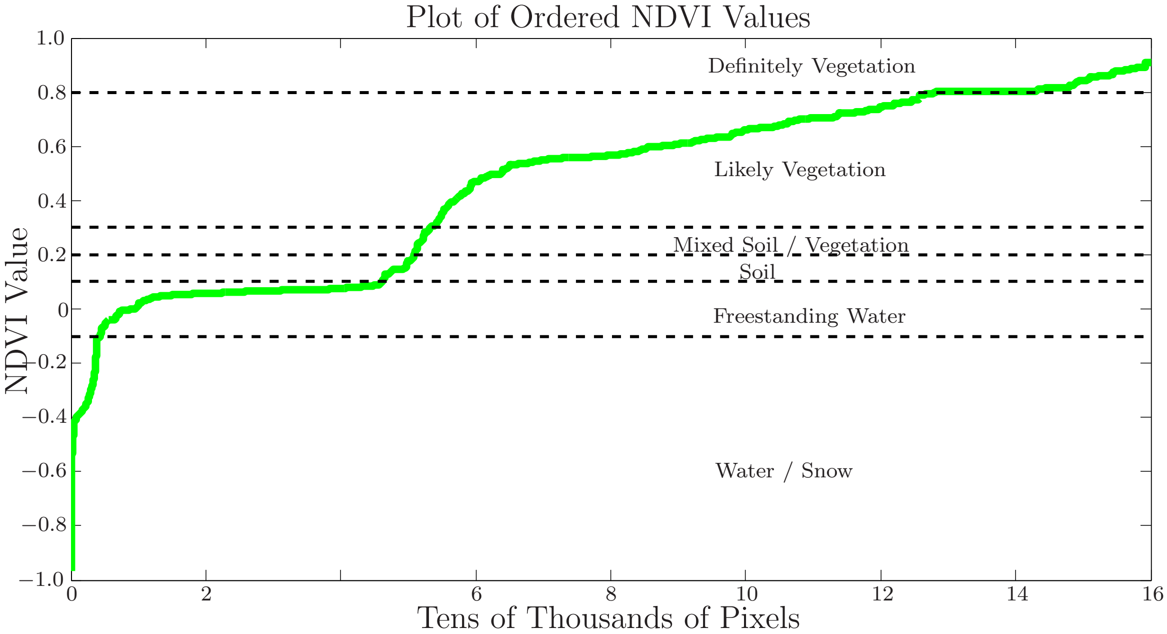


Figure 4.4: This plot of the ordered NDVI values from Figure 4.3 is plotted with each of the thresholds and theshold labels from Table 2.1.

object map for the threshold NDVI > 0.1 in the lower right image of Figure 4.5 marks the large central building in the middle of the scene as vegetation. These two NDVI thresholds (NDVI > 0.8 and NDVI > 0.1) are not good choices. The Non-Equal Weighted Accuracy (NEWA) (see Section 2.4.3) for each of the four *temporary object maps* in the right column of Figure 4.5 is calculated as compared to the “ground truth” vegetation map. The results are in the middle column of Table 4.2. For comparison, the NDVI is applied directly to the hyperspectral image. For each NDVI threshold, the NEWA is calculated and listed in the right column of Table 4.2.

There are three conclusions drawn from Table 4.2.

1. The “rule-of-thumb” threshold, NDVI > 0.3, provides a lower confidence limit of the presence of vegetation.
2. Using the SOM to identify vegetation improved the NEWA accuracy from 0.9608 to 0.9637 for NDVI > 0.8 and from 0.9806 to 0.9949 for NDVI > 0.5. Abstracting information through a SOM was expected to lower the NEWA slightly.

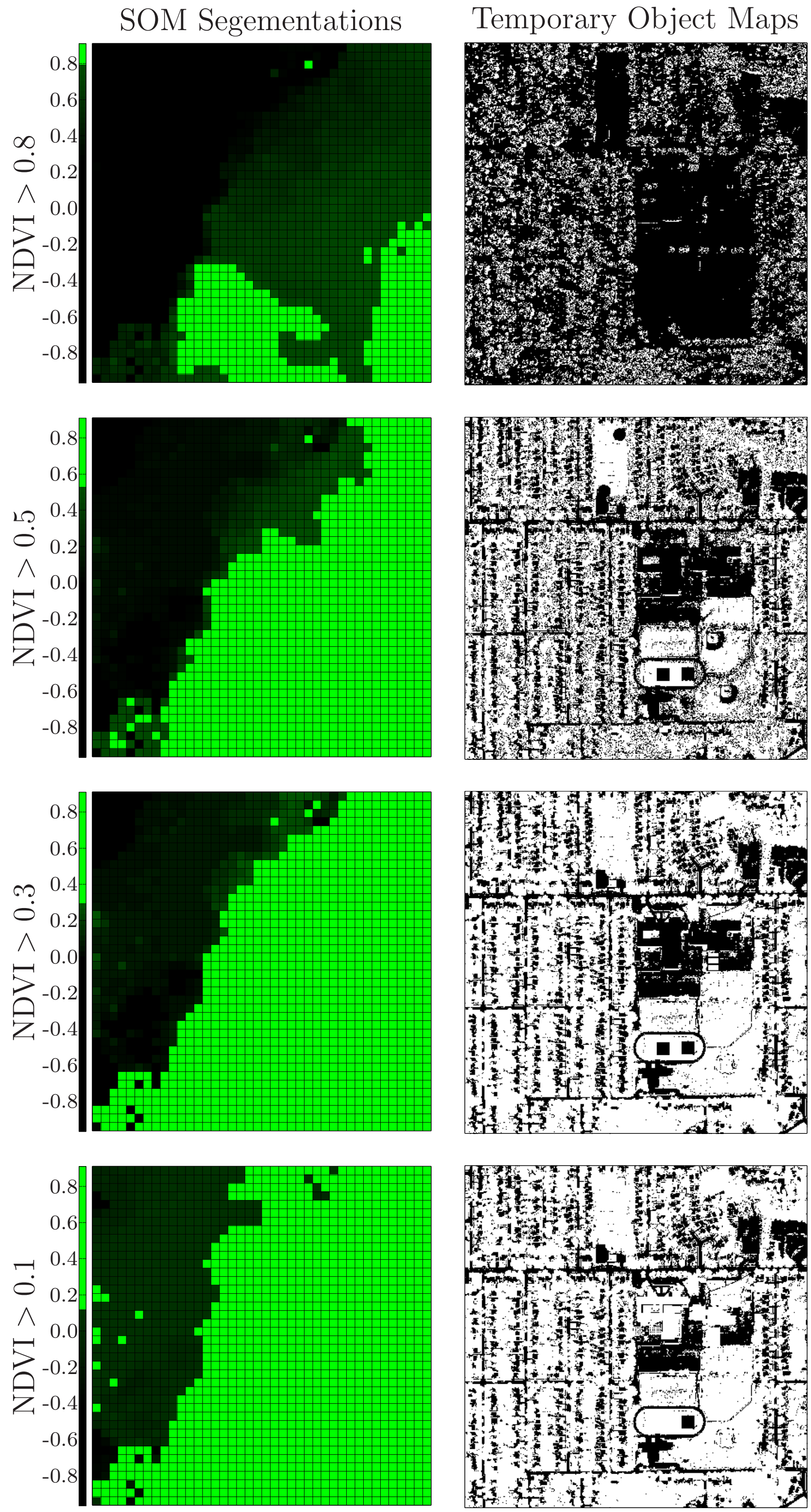


Figure 4.5: Each of the vegetation SOM segmentations (left column) are processed into temporary object maps (right column) using the process from Figure 3.11. Green PEs and white pixels are binary "true" indicating the presence of vegetation. The top right object map does not encompass the vegetation, and the bottom right object map marks the large building in the center of the image as vegetation.

Table 4.2: Listing of the NEWA for the *temporary object maps* from Figure 4.5 (right column) as compared to ground truth. NDVI is applied to the hyperspectral image resulting in the accuracies in the right column.

Threshold	Figure 4.5 (right column) Temporary Object Maps	NDVI applied to the Hyperspectral Image
NDVI > 0.8	0.9637	0.9608
NDVI > 0.5	0.9949	0.9806
NDVI > 0.3	0.8801	0.9175
NDVI > 0.1	0.5141	0.5155

3. The hyperspectral scene analyzed is simulated data, and these values need to be cross-correlated with results from a real-data analysis.

The threshold NDVI > 0.1 does not meet the recommendation to have at least a 75% accuracy as described in Section 3.4. Visual inspection of the upper right *temporary object map* from Figure 4.5 negates using NDVI > 0.8 as a threshold. The conclusion is that the thresholds NDVI > 0.3 and NDVI > 0.5 are reasonable values and are therefore selected. The thresholds are used to create the vegetation NDVI > 0.3 and NDVI > 0.5 SOM segmentations shown in the middle two images of the left column of Figure 4.5. The *SOM segmentations* are later used during spectral analysis and propogated to the *create the initial object maps* step.

Application of NDVI to the converged SOM (Figure 4.3) to produce two *SOM segmentations* is complete. The two *SOM segmentations* are labeled "vegetation (NDVI > 0.3 and NDVI > 0.5)" *SOM segmentations*, in the spectral process flow diagram in Figure 3.3. The next step of the spectral analysis is to apply the NIR scatter tree ID algorithm to segment the "tree" neighborhood of the SOM.

4.2.3 NIR Scatter Tree Identification. The NIR scatter tree ID algorithm is described in Section 3.2.3. The NIR scatter is calculated for the hyperspectral

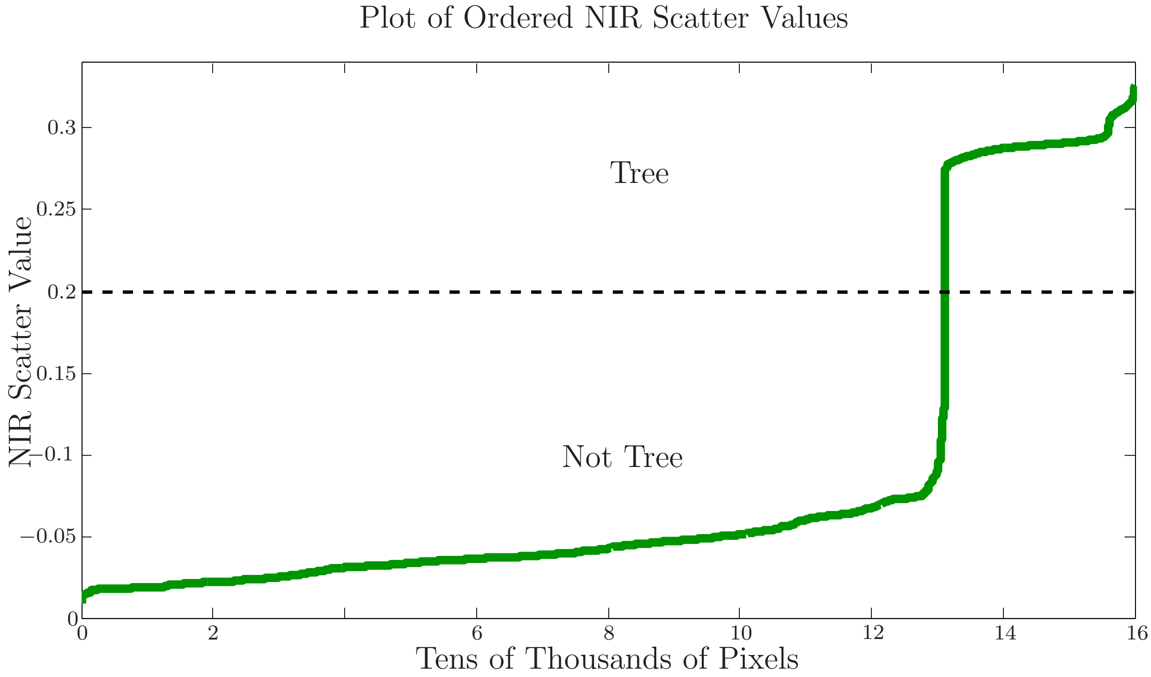


Figure 4.6: The NIR scatter tree identification algorithm from Section 3.2.3 is applied to the hyperspectral image and the resulting ordered NIR scatter values are then plotted. The tree threshold is set at the maximum of the derivative of this plot at NIR scatter value = 0.2.

image from Figure 3.2 and then the ordered NIR scatter values are plotted as shown in Figure 4.6. The maximum of the derivative of Figure 4.6 is the tree threshold (Tree_Threshold = 0.2).

The NIR scatter tree ID algorithm is applied to the converged SOM from Figure 4.2 and thresholded with the NIR scatter tree threshold. NDVI with thresholds NDVI $>$ 0.3 and NDVI $>$ 0.5 are applied in order to separate the vegetative from the non-vegetative PEs. This results in the two *SOM segmentations* shown in the left two images of Figure 4.7. Using the process from Figure 3.11 to create a *temporary object map* for spatial scene verification results in the right two images in Figure 4.7.

The *temporary object map* images in the right column of Figure 4.7 are very similar. Both images correctly identify trees, and both contain the same non-tree artifacts. The small rectangles in the right images of Figure 4.7 are actually grass located in shadow of a building. The lower right image (corresponding to NDVI $>$ 0.3) appears to have identified more trees than the upper right image. Table 4.3 shows the NEWA calculation for each of the tree *temporary objects maps* in Figure 4.7.

Table 4.3: This table compares the NEWA for each of the "tree" *SOM segmentations* in the right column of Figure 4.7.

Threshold	NEWA	Related Figure
NDVI $>$ 0.5	0.7867	Figure 4.7, upper right image
NDVI $>$ 0.3	0.8201	Figure 4.7, lower right image

Based on the accuracies in Table 4.3 and the qualitative assessment of the temporary object maps in the right column of Figure 4.7, the "tree" *SOM segmentation* corresponding to an NDVI threshold of NDVI $>$ 0.3 shown in the lower right image of Figure 4.7 is selected to be the final "tree" *SOM segmentation* used later in spectral analysis and propogated to the *create the initial object maps* step. With the "tree" *SOM segmentation* complete, the next step is to create the "non-tree vegetation"

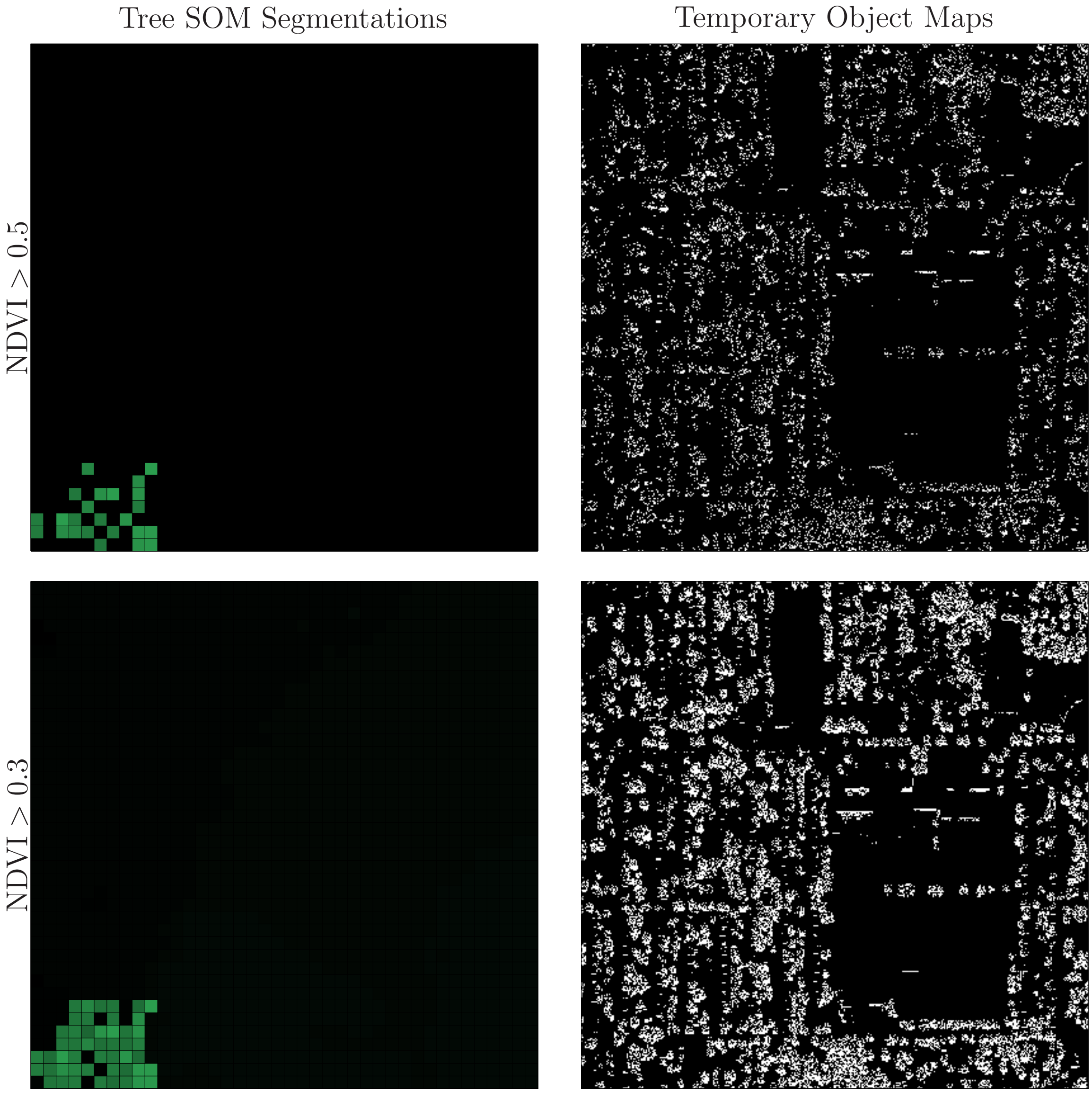


Figure 4.7: Green PEs and white pixels are labeled “tree”. The NIR scatter tree ID algorithm from Section 3.2.3 is applied to the converged SOM (Figure 4.2), and then thresholded based on Figure 4.6. NDVI is used to separate the vegetative and non-vegetative SOM PEs using thresholds NDVI > 0.3 and NDVI > 0.5 resulting in two SOM segmentations (left column). The process in Figure 3.11 is used to create the two temporary object maps for spatial scene verification (right column). Green PEs and white pixels are “true” and black PEs and pixels are “false” for “trees”.

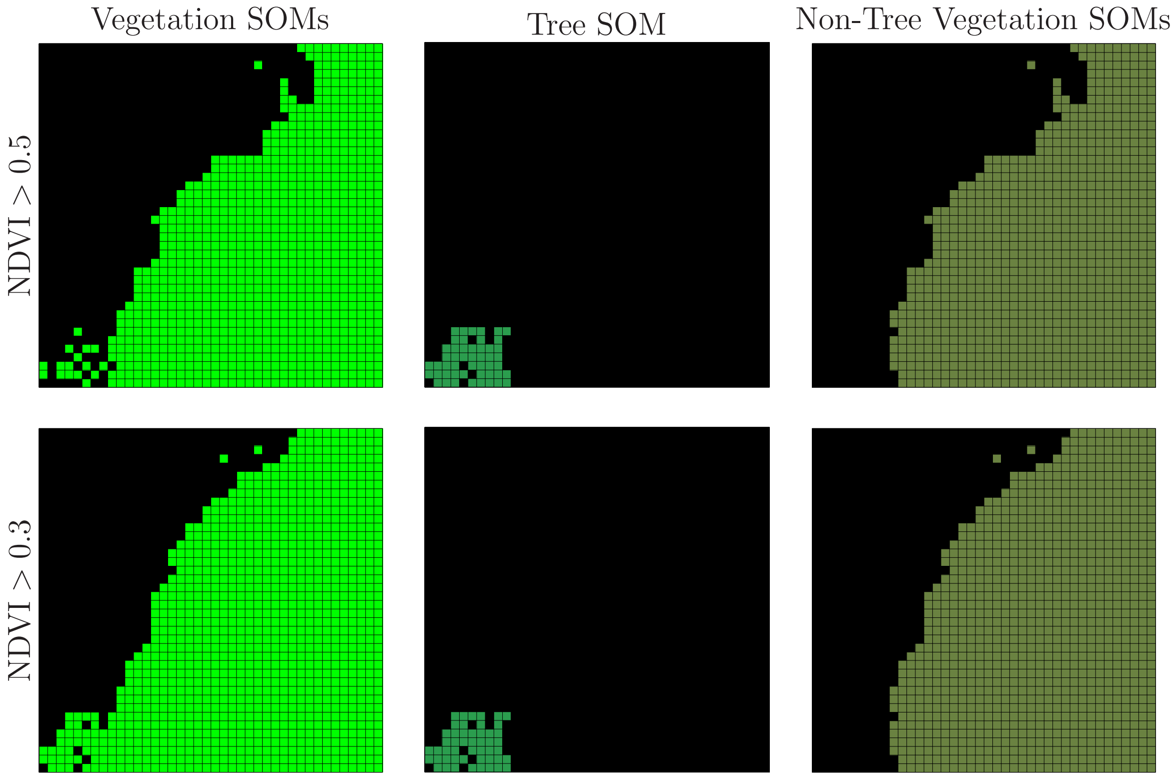


Figure 4.8: This is the graphical representation of Equation 3.1. The "tree" *SOM segmentation* (middle column, repeated) is logically "deleted" from the the two "vegetation" *SOM segmentations* (NDVI $>$ 0.3 and NDVI $>$ 0.5) (left column). The results are the two "non-tree vegetation" (NDVI $>$ 0.3 and NDVI $>$ 0.5) *SOM segmentations* (right column). Colored PEs are "true" and black PEs are "not true" for the indicated material of interest.

SOM segments using the two "vegetation" *SOM segmentations* from the previous step and the "tree" *SOM segmentation* from thsi step.

4.2.4 "Non-Tree Vegetation" SOM Segmentations. The two "non-tree vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)" *SOM segmentations* shown in Figure 3.3 are created using Equation 3.1. The process is shown graphically in Figure 4.8. The "tree" *SOM segmentation* (Figure 4.8, middle column, repeated) is logically "deleted" (see Table 3.1) from the two "vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)" *SOM segmentations* (Figure 4.8, left column) resulting in two "non-tree vegetation (NDVI $>$ 0.3 and NDVI $>$ 0.5)" *SOM segmentations* (Figure 4.8, right column).

The “non-tree vegetation” *SOM segmentations* do not need spatial scene verification because the three *SOM segmentations* from which they are created are previously verified. The two “non-tree vegetation” *SOM segmentations* are propogated to the *create the initial object maps* step. The next step is to have the user select exemplar “rooftop” and “roadway” exemplar spectra, and use them to create the “rooftop” and “roadway” *SOM segmentations*.

4.2.5 “Rooftop” and “Roadway” SOM Segmentations.

The next step in the spectral analysis process flow diagram shown in Figure 3.3 is to have the user select exemplar spectra for rooftops and roadways as shown in Figure 3.8 and outlined in Section 3.2.5. User input is simulated to select pixels $(187, 305)$ and $(155, 95)$ for the “rooftop” exemplars, and $(121, 359)$ for the “roadway” exemplar. Spatial scene verification from Figure 3.11 is used to adjust the ℓ_2–norm distance threshold for each of the *SOM segmentations* starting with the “rooftop” *SOM segmentation* and ending with the “roadway” *SOM segmentation*. The final “rooftop” *SOM segmentation* is shown in the left image of Figure 4.9, and the corresponding verified spatial scene in the right image of Figure 4.9. The final “roadway” *SOM segmentation* is shown in the left image of Figure 4.10 and the corresponding verified spatial scene in the right image of Figure 4.10.

It is important to adjust the ℓ_2–norm threshold to encompass the majority of the material without introducing extensive error. This is currently a “best judgement” process; however, there is plenty of room for error since the spatial analysis phase significantly improves the initial accuracies. The final rooftop and roadway ℓ_2–norm thresholds are listed in Table 4.4.

The results are the “rooftop” *SOM segmentation* and the “roadway” *SOM segmentation* labeled in light-blue boxes in Figure 3.3 and shown in the left images of Figures 4.9 and 4.10. The other light-blue boxes correspond to the previously created “vegetation (NDVI > 0.3)”, “non-tree vegetation (NDVI > 0.3 and NDVI > 0.5)”, and “tree” *SOM segmentations*. These six *SOM segmentations* are the final spectral

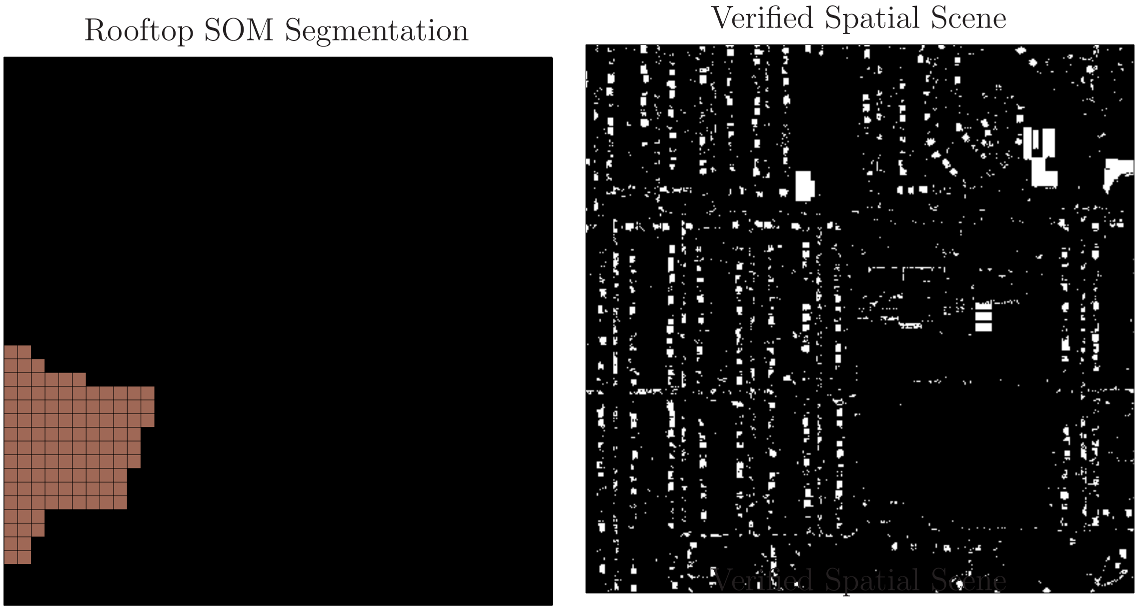
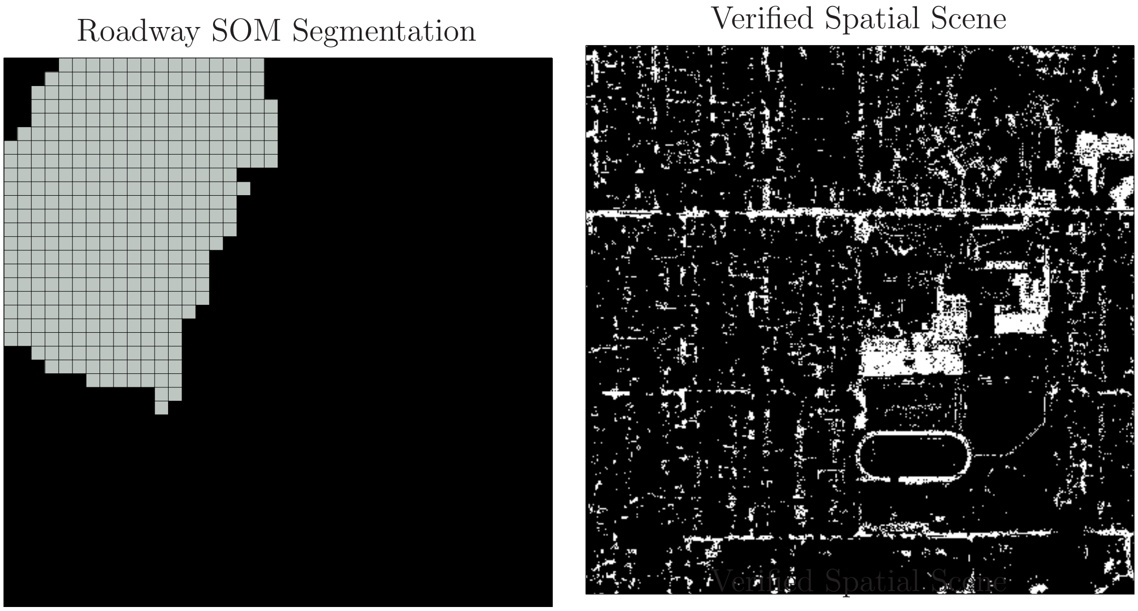


Figure 4.9: Exemplar spectra chosen with the process shown in Figure 3.8 in combination with spatial scene verification are used to adjust the ℓ_2–norm threshold resulting in the "buildings" *SOM segmentation* (left image). During spatial scene verification, the threshold is adjusted while verifying the results in the *temporary object map* (right image). The brown PEs and white pixels are "rooftop" and the black PEs and pixels are "not rooftop".

Table 4.4: Listing of ℓ_2–norm thresholds for the "rooftop" and "roadway" *SOM segmentations*

SOM Segmentation	ℓ_2–norm Threshold
Rooftop	0.1
Roadway	0.2


Roadway SOM Segmentation

Verified Spatial Scene

Figure 4.10: Exemplar spectra chosen with the process shown in Figure 3.8 in combination with spatial scene verification are used to set a ℓ_2–norm threshold resulting in the roadway SOM segmentation (left image). During spatial scene verification, the threshold is adjusted while verifying the results in the spatial object map (right image). The grey PEs and white pixels are "roadway" and the black PEs and pixels are "not roadway".

analysis outputs that are propogated to the *create the initial object maps* step shown in Figure 3.1. Four of the *SOM segmentations* were previously shown in Figure 3.9 to demonstrate the SOM characteristic of preserving the topological neighborhood.

4.3 Create the Initial Object Maps

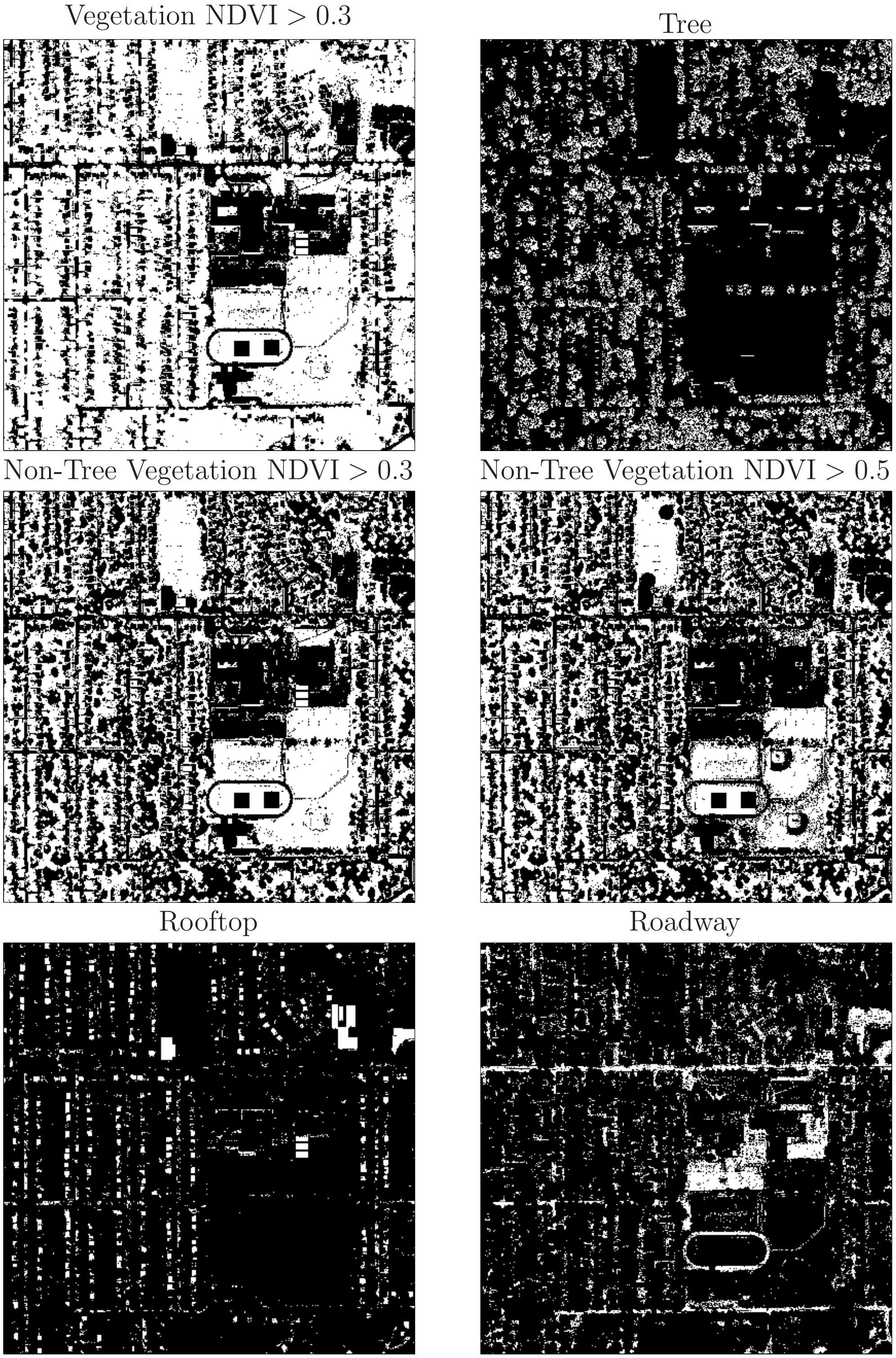
Once spectral analysis is complete, the six *SOM segmentations* resulting from spectral analyis are processed as shown in Figures 3.1 and 3.11 into six *initial object maps*. The *six initial object maps* shown in Figure 4.11 are the outputs of the *create the initial object maps* step. Once the *initial object maps* are created, the process proceeds into the spatial analysis process of Figure 3.1.

4.4 Transitioning from Spectral to Spatial Analysis

Table 4.5: Listing of the *initial object map* accuracies.

Initial Object Mask	NEWA	Reference Figure
Vegetation NDVI > 0.3	0.9949	Figure 4.11, upper left
Tree NDVI > 0.5	0.8201	Figure 4.11, upper right
Non-Tree Vegetation NDVI > 0.3	0.7796	Figure 4.11, middle left
Non-Tree Vegetation NDVI > 0.5	0.8159	Figure 4.11, middle right
Rooftop	0.7783	Figure 4.11, bottom left
Roadway	0.8673	Figure 4.11, bottom right

Section 3.4 states that the recommended accuracy from the spectral analysis process is 75%. Table 4.5 calculates the accuracy for each of the *initial object maps* from Figure 4.11. In traditional spectral analysis, this would be the stopping point and these accuracies would be the final reported accuracies. In [26], the author segments an urban scene with a SOM and achieves an overall 80% accuracy. The accuracies in Table 4.5 confirm these findings, with an 84.2% average accuracy. In order to create the GIS layers, however, the *initial object maps* must be spatially processed. As a by-product of the spatial analysis process shown in Figure 3.1, the "vegetation"



Vegetation NDVI > 0.3

Tree

Non-Tree Vegetation NDVI > 0.3

Non-Tree Vegetation NDVI > 0.5

Rooftop

Roadway

Figure 4.11: The six *initial object maps* that result from the *create the initial object maps* step of Figure 3.1. They are created using the process shown in Figure 3.11 and discussed in Section 3.3. They are used during spatial analysis to create eight *final object maps*. White pixels are “true” and black pixels are “not true” for the indicated object type.

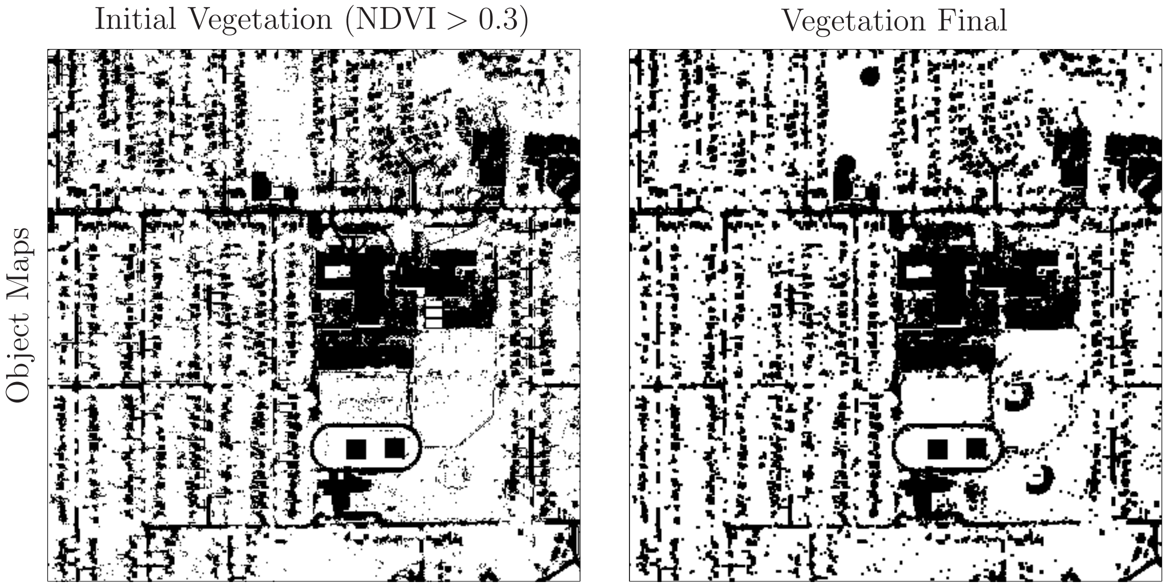


Figure 4.12: A comparison between the “vegetation (NDVI $>$ 0.3)” *initial object map* (left image), which undergoes a morphological close operation with a 2×2 SE to produce the “vegetation” *final object map* (right image). White pixels represent “vegetation” and black pixels represent “not vegetation”.

final object map accuracy decreases. This is because the “vegetation” *final object map* (Figure 4.11, upper left image) is “noisy” and “cluttered” and undergoes spatial operations that sacrifice accuracy to increase its visual appeal and subsequently its utility. The other five *object maps* show an improvement in the accuracy calculation. The six *initial object maps* from Figure 4.11 are propogated to the spatial analyis step of Figure 3.1. Spatial analysis results in eight *final object maps*.

4.5 Spatial Analysis

The spatial analysis process shown in Figures 3.1 and 3.12 result in eight *final object maps* from the six input *initial object maps* shown in Figure 4.11. Spatial analysis is accomplished in six steps, itemized below.

4.5.1 Step 1: The “Vegetation”, “Tree”, and “Fields” Final Object Maps.

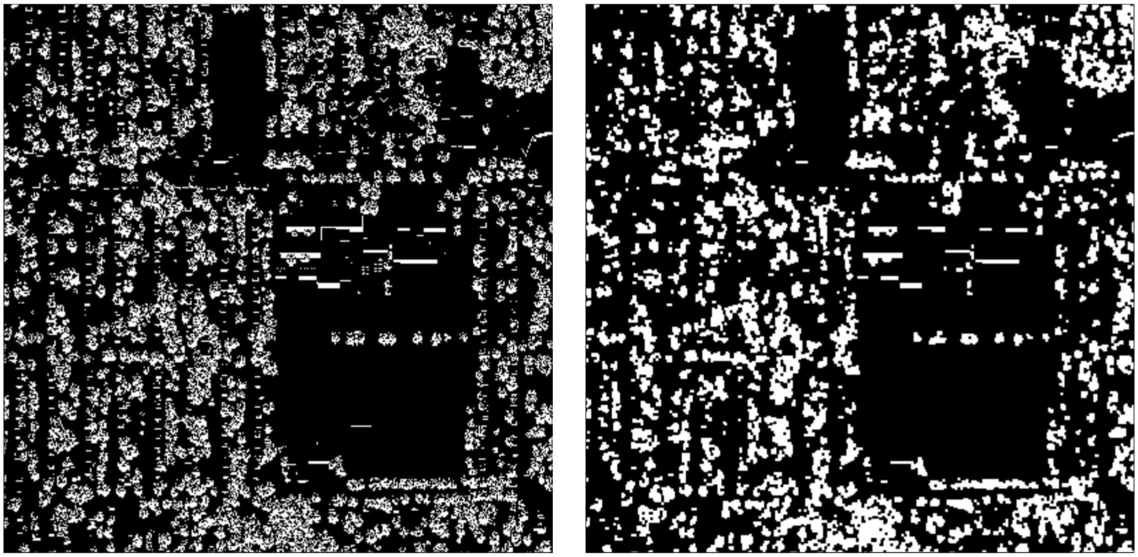
Section 3.5.1 describes the process used in step 1 to operate on the “vegetation NDVI $>$ 0.3” *initial object map* (Figure 4.12, left image) to produce the “vegetation” *final object map* (Figure 4.12, right image). The "vegetation" *final object map* is propogated forward to the *finalize GIS layers* step.

Section 3.5.1 describes the process used to operate on the "tree" *initial object map* to produce the "tree" *final object map.* Figure 4.13 demonstrates the results of applying three different-sized SEs (2×2, 3×3, and 4×4) on the "tree" *initial object map* (Figure 4.13, upper right image). Ten different-sized SEs are used with a morphological close operation, and the accuracy calculation of the resulting *object maps* are listed in Table 4.6. The maximum achieved experimental accuracy from Table 4.6 corresponds to the 3×3 SE. This SE size is selected to process the "tree" *initial object map* into the "tree" *final object map.* Figure 4.14 compares the "tree" *initial object map* (Figure 4.14, left image) with the "tree" *final object map* after morphological processing with a 3×3 SE (Figure 4.14 right image). The "tree" *final object map* is propogated forward to the *finalize GIS layers* step.

Table 4.6: Listing of the "tree" *object map* accuracies after morphological operations.

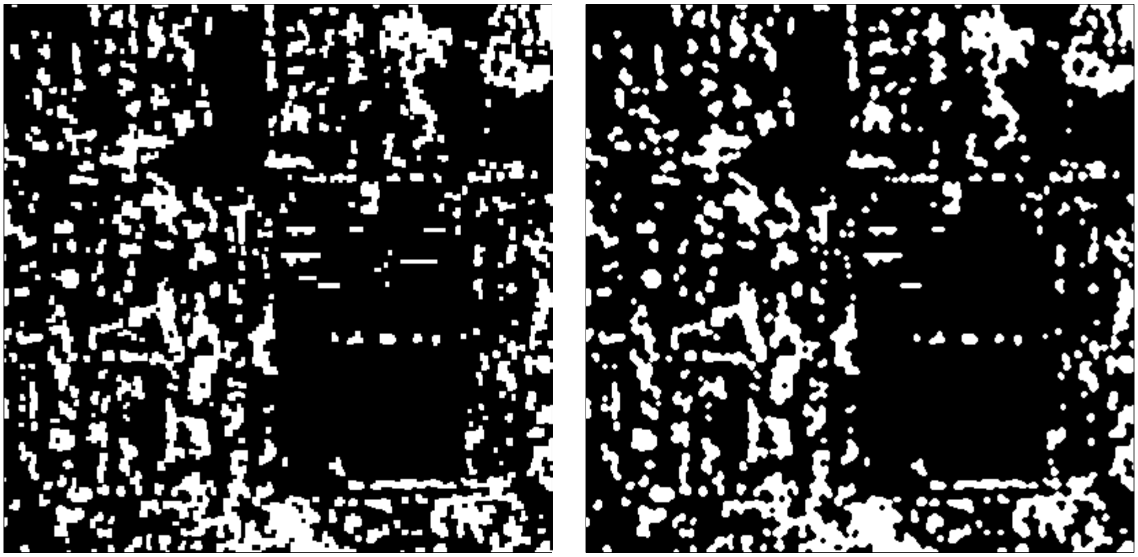
Structuring Element	Tree Final Object Map Accuracy
2×2 Square	0.860
3×3 Plus	0.866
3×3 Square	0.885
3×3 Diamond	0.875
4×4 Square	0.877
4×4 Plus	0.859
4×4 Diamond	0.852
5×5 Square	0.858
5×5 Ball	0.835
5×5 Diamond	0.815

Section 3.5.1 describes the process used to operate on the "non-tree vegetation (NDVI > 0.5)" *initial object map* to produe the "fields" *final object map.* The "non-tree vegetation (NDVI > 0.5)" *initial object map* undergoes iterative morphological



Tree Initial Object Map

2 × 2 SE



3 × 3 SE

4 × 4 SE

Figure 4.13: The "tree" *initial object map* (upper left image) is operated on with morphological close operations (left) and (bottom left/right) with the indicated SE size in order to make the object map more visually appealing. In this sequence, white pixels represent "trees" and black pixels represent "not trees".

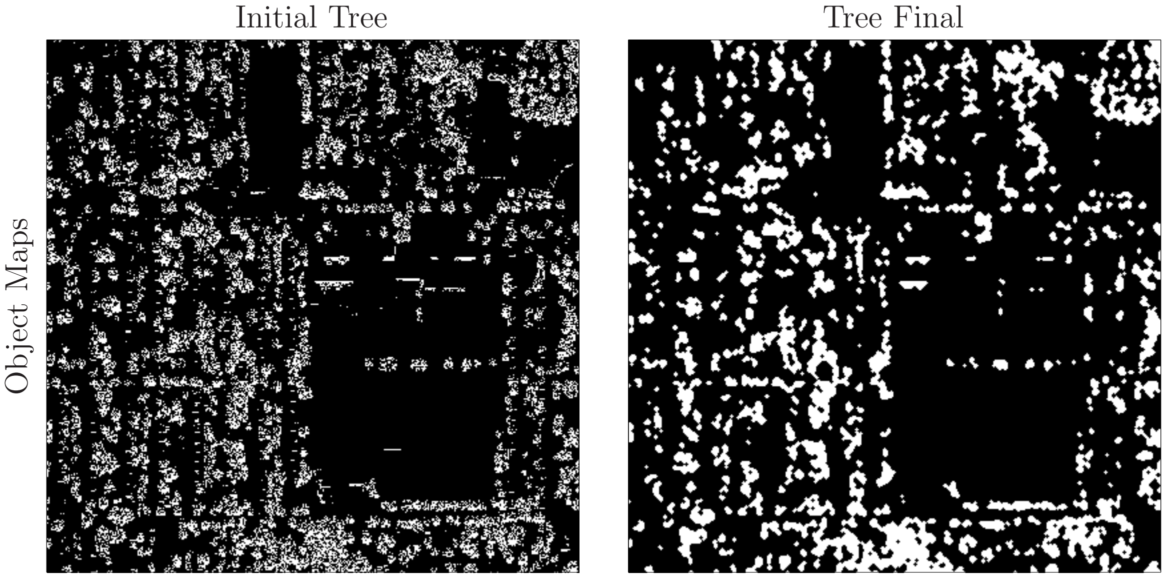


Figure 4.14: A comparison between the "tree" *initial object map* (left image), which is operated on with a morphological closed with a 3 × 3 SE to produce the "tree" *final object map* (right image). White pixels represent "trees" and black pixels represent "not trees".

operations with an $n \times n$ SE where $n = 5, 6, \ldots, 20$ as demonstrated in Figure 4.15, resulting in an intermediate "fields" *object map.* The "roadway" *initial object map* (Figure 4.11, lower right image) is logically "deleted" (see Table 3.1) from the intermediate "fields" *object map* to create the "fields" *final object map* as shown in the right image of Figure 4.18. The "fields" *final object map* is propogated forward to the *finalize GIS layers* step.

4.5.2 Step 2: The "Parking Areas" Final Object Map. Section 3.5.2 describes the process used in step 2 to process the "roadway" *initial object map* (Figure 4.11, lower right image) into the "parking areas" *final object map.* The "roadway" *initial object map* undergoes iterative morphological open and close operations, with an $n \times n$ SE where $n = 2, 3, \ldots, 10$. The process is illustrated in Figure 4.17. The "parking areas" *final object map* is propogated forward to the *finalize GIS layers* step.

4.5.3 Step 3: The "Major Buildings" Final Object Map. Section 3.5.3 describes the process used in step 3 to create the "major buildings" *final object map.*

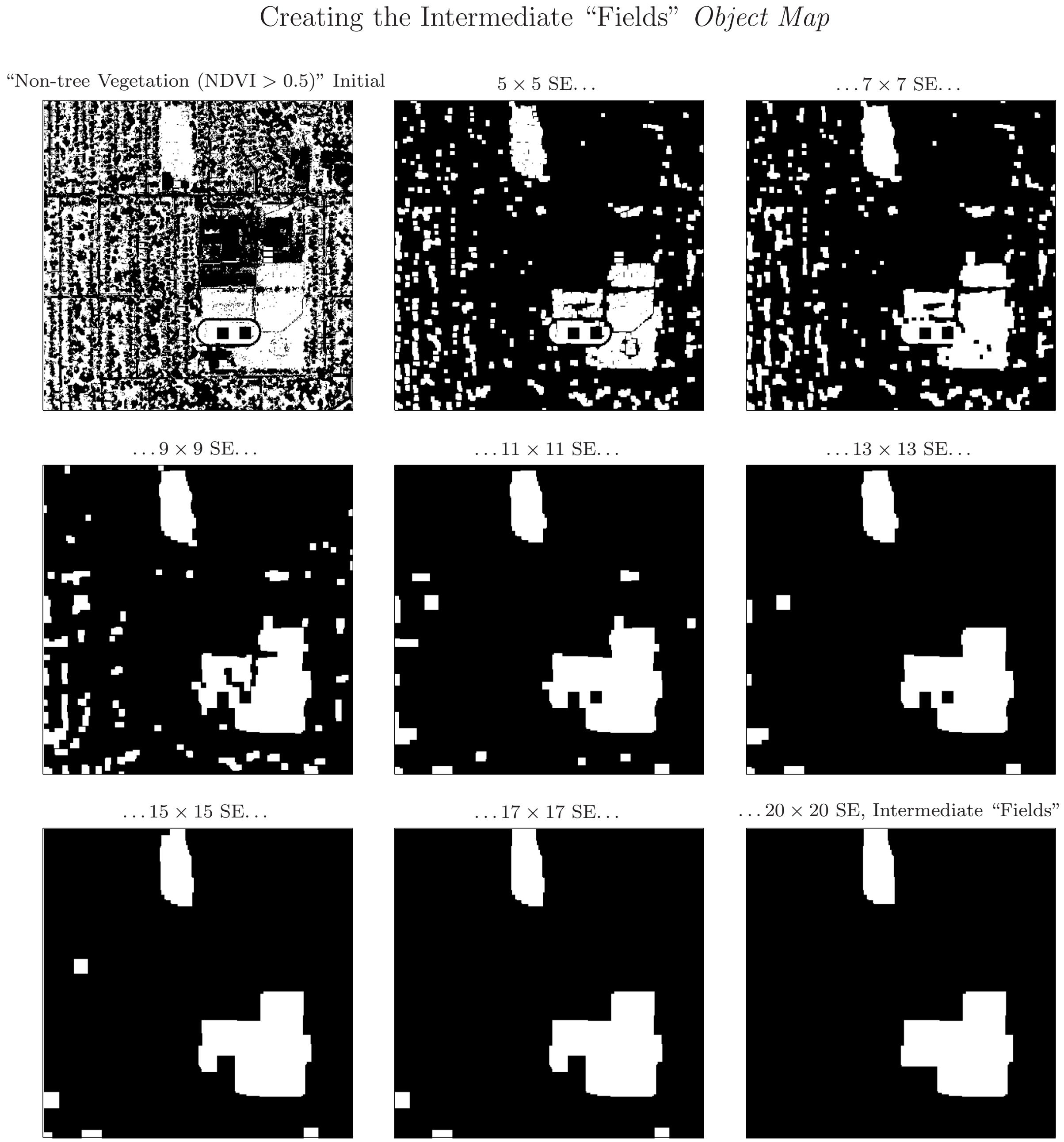

Creating the Intermediate "Fields" *Object Map*
"Non-tree Vegetation (NDVI > 0.5)" Initial
5 × 5 SE...
...7 × 7 SE...
...9 × 9 SE...
...11 × 11 SE...
...13 × 13 SE...
...15 × 15 SE...
...17 × 17 SE...
...20 × 20 SE, Intermediate "Fields"

Figure 4.15: Iterative morphological open/close operations are used on the "non-tree vegetation (NDVI $>$ 0.5)" *initial object map* (upper left image) with an $n \times n$ SE where $n = 5, 6, \ldots, 20$. Eight of the iterations are displayed. The result is the intermediate "fields" *object map* (lower right image).

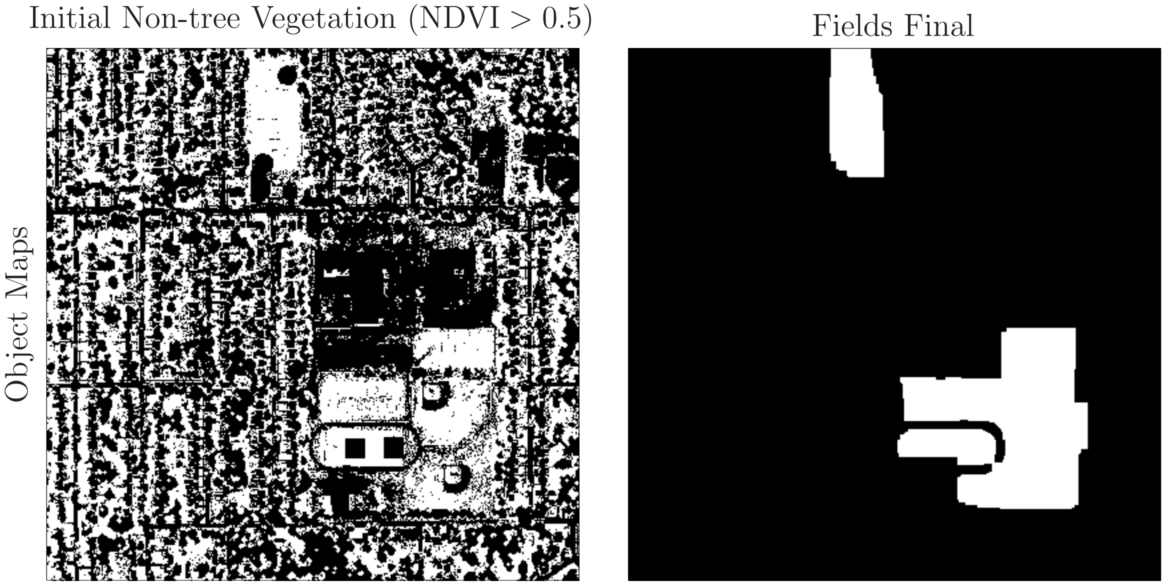


Figure 4.16: This is a comparison between the "non-tree vegetation (NDVI > 0.5)" *initial object map* (left image) which is iteratively open/closed with morphological as shown in Figure 4.15 to produce the "fields" *final object map* (right image). White pixels represent "fields" and black pixels represent "not fields".

The "rooftop" *initial object map* (Figure 4.11, lower left image) is processed into the intermediate "buildings" *object map*. This is done by first constructing a negative image mask (NIM) with logical "or" operations and the "initial vegetation (NDVI > 0.3)" and "roadway" *initial object maps*, and the "fields" and "parking areas" *final object maps*. The NIM is morphologically closed with a 3×3 SE. The NIM construction is illustrated in Figure 4.19. The logical inverse of the NIM in the lower-right image of Figure 4.19 is logically "added" to the "rooftop" *initial object map* with a logical "or" operation. The intermediate "buildings" *object map* is shown in the right image of Figure 4.20.

The intermediate "buildings" *object map* is morphologically opened with a 7×7 SE, dilated with a 5×5 SE, and then closed with a 3×3 SE to produce the "major buildings" *final object map* (Figure 4.21, right image). The "major buildings" *final object map* is propogated forward to the *finalize GIS layers* step.

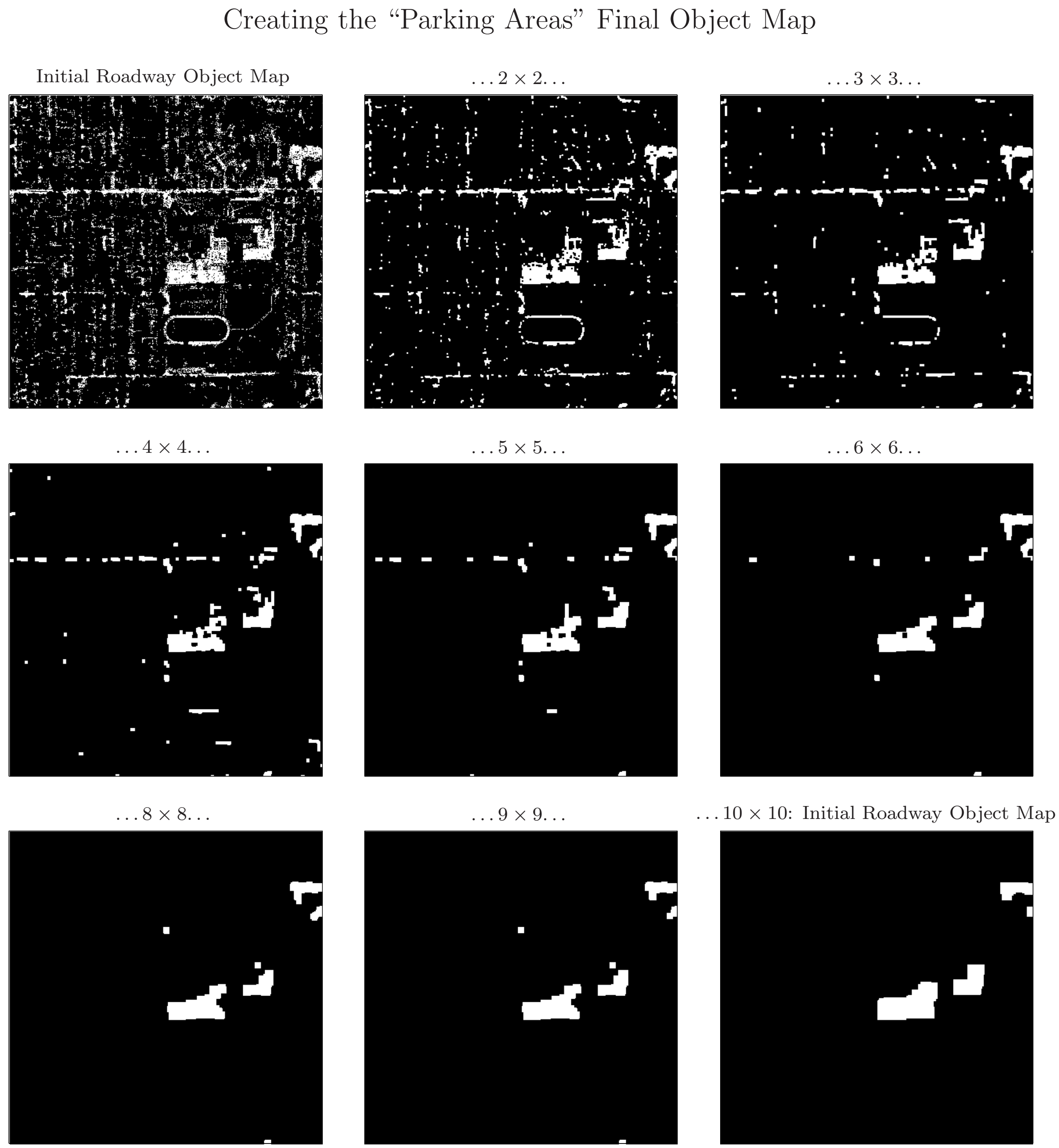

Creating the "Parking Areas" Final Object Map
Initial Roadway Object Map
...2 × 2...
...3 × 3...
...4 × 4...
...5 × 5...
...6 × 6...
...8 × 8...
...9 × 9...
...10 × 10: Initial Roadway Object Map

Figure 4.17: Iterative morphological open/close operations are used on the "roadway" *initial object map* (upper left image) with an $n \times n$ SE where $n = 2, 3, \ldots, 10$. Eight of the iterations are displayed. The result is the "parking areas" *final object map* (lower right image).

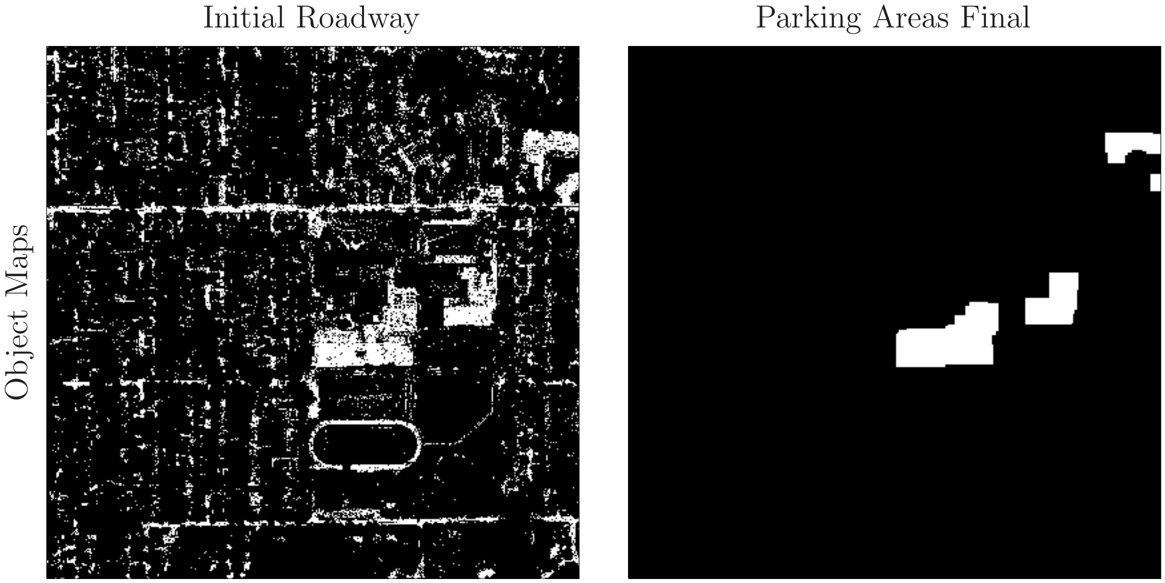


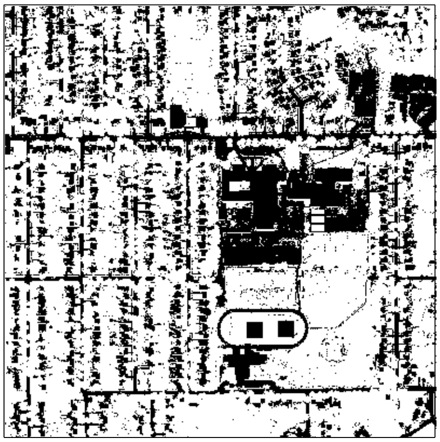
Figure 4.18: This is a comparison between the “roadway” *initial object map* (left image) which is iteratively open/closed with morphological as shown in Figure 4.17 to produce the “parking areas” *final object map* (right image). White pixels represent “parking areas” and black pixels represent “not parking areas”.

4.5.4 Step 4: The “Best Known Good Roadway” Object Map*.* Section 3.5.4 describes the process used in step 4 to create the “best known good roadway” *object map.* The “roadway” *initial object map* (Figure 4.22, left image) is operated on by morphological open operations with SE sizes 2 × 2 and 3 × 3, and then by logically “deleting” (see Table 3.1) the “parking areas” and “major buildings” *final object maps.* The result is the “best known good roadway” *object map* (Figure 4.22, right image). The “best known good roadway” *object map* is used as an input to the *road finding algorithm.* The *road finding algorithm* adds possible road connections. For this reason, it is important to ensure the “best known good roadway” *object map* does not include excessive non-roadway segments, or the *road finding algorithm* may add road connections that do not exist. Future improvements in the algorithm are intended to remove this caveat.

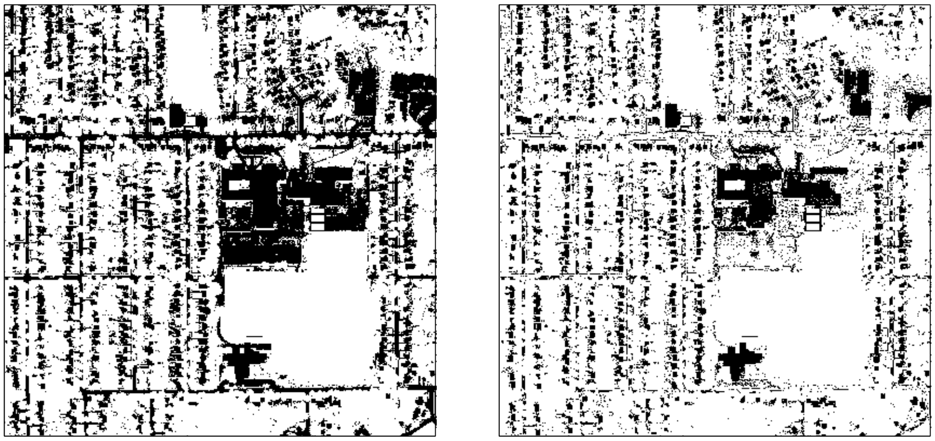
4.5.5 Step 5: The “Road Network” Final Object Map*.* Section 3.5.5 describes the process used in step 5 to create the “road network” *final object map.* The *road finding algorithm* uses the “best known good roadway” *object map* and a NIM

Constructing the Buildings Negative Image Mask

1. Initial Vegetation (NDVI > 0.3) Object Map



2. “or” Fields Final Object Map 3. “or” Initial Roadway Object Map



4. “or” Parking Final Object Map 5. Close with 3 × 3 SE Object Map

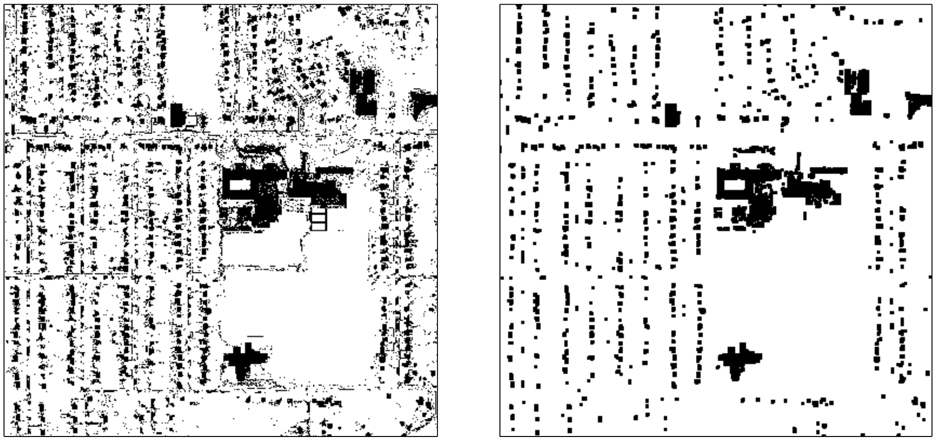


Figure 4.19: The negative image mask (NIM) is created by logically “adding” four *object maps* with a logical “or” operation and then morphologically closing the NIM. The logical inverse of this map is logically “added” to the “rooftop” *initial object map* (Figure 4.11, lower left image) to create the intermediate “buildings” *object map*.

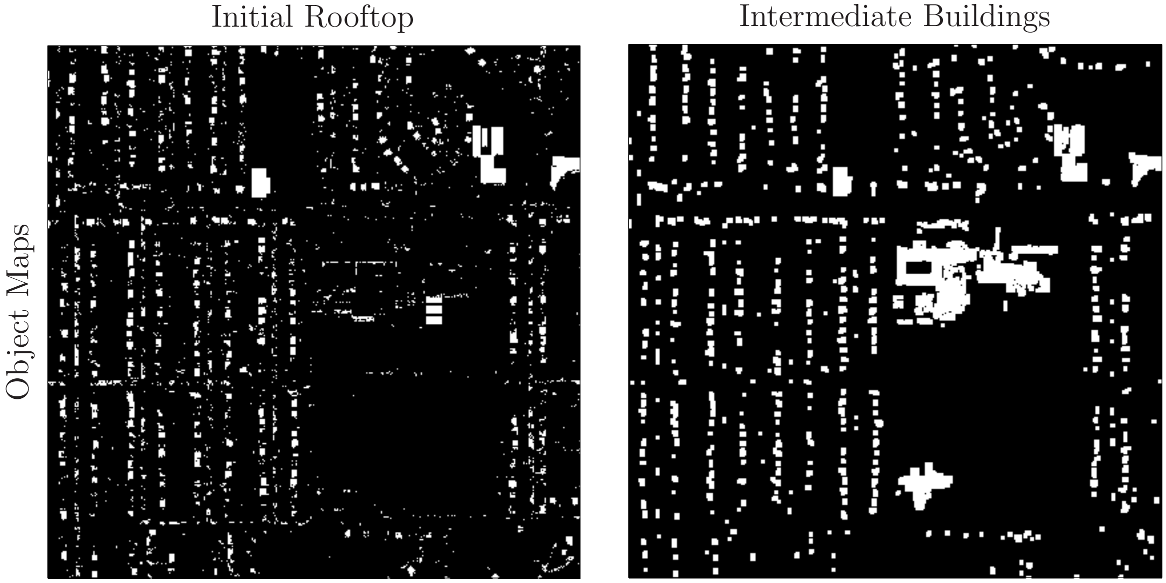


Figure 4.20: The logical inverse of the negative image mask (NIM) in the lower right image of Figure 4.19 is logically "added" to the "rooftop" *initial object map* with a logical "or" operation (left image) followed by a morphological open with a 3×3 SE to produce the intermediate "buildings" *object map* (right image). White pixels represent "buildings" and black pixels represent "not buildings areas".

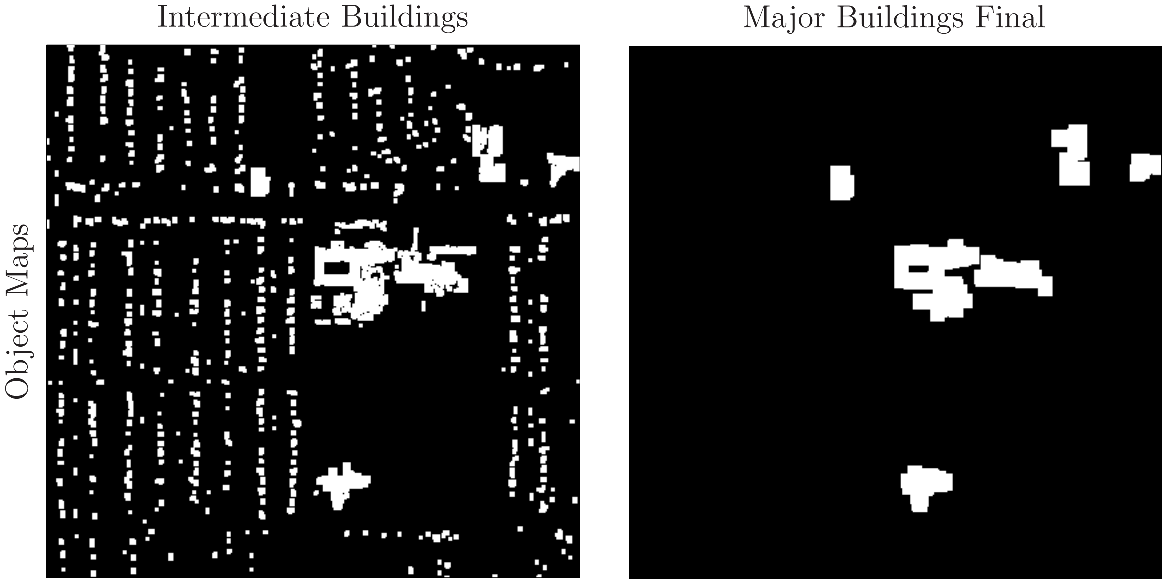


Figure 4.21: The intermediate "building" *object map* (left image) is morphologically opened with a 7×7 SE, dilated with a 5×5 SE, and then close with a 3×3 SE to produce the "major buildings" *final object map* (right image). White pixels represent "major buildings" and black pixels represent "not major buildings areas".

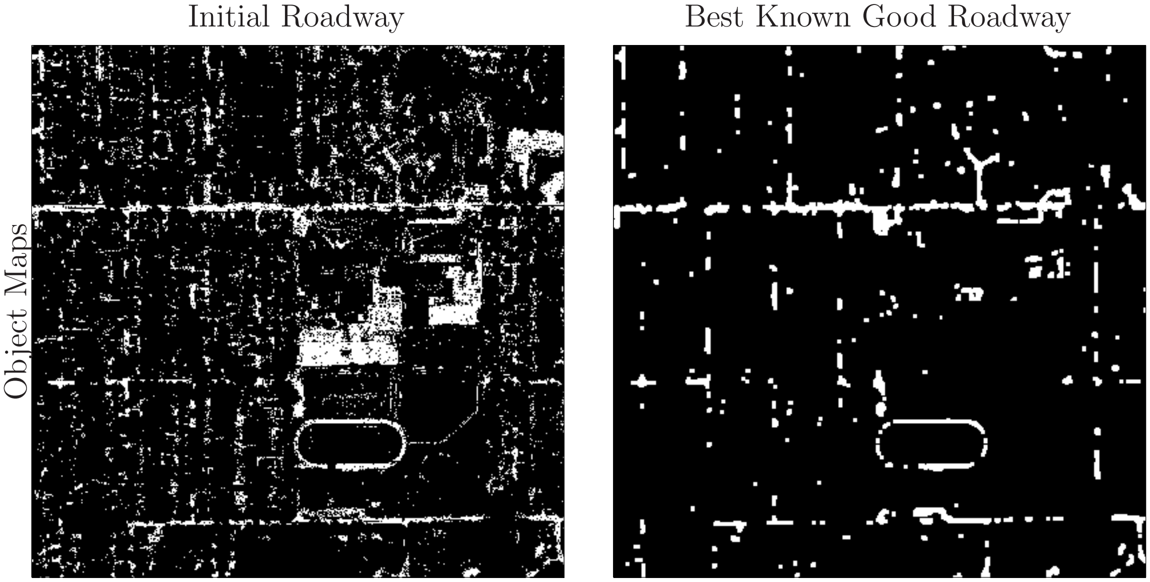


Figure 4.22: The “roadway” *initial object map* (left image) is processed with a morphological open with an SE up to 3×3, and then the “parking areas” and “major buildings” *final object maps* are logically “deleted”. The result is the “best known good roadway” *object map* (left image). The “best known good roadway” *object map* is used by the *road finding algorithm* to locate possible road connetions. White pixels are “known roadway” and the black pixels are “known not roadway”.

to make possible road connections. The NIM is constructed in the same manner as in Figure 4.19 except with different object maps. Six object maps used to create the road finding algorithm NIM: the "non-tree vegetation (NDVI > 0.3 and NDVI > 0.5)" *initial object maps*, the "fields", "parking areas", "major buildings" *final object maps*, and the intermediate "buildings" *object map.*

Once the NIM is constructed, the *road finding algorithm* from Figure 3.13 is used to find possible road connections on the "best known good roadway" *object map.* Each of the NIM layers is "pealed", and simultaneously possible road connections are created. This simultaneous process is shown in Figures 4.23 and 4.24. Figures 4.23 is the NIM being "pealed", and Figure 4.24 shows the possible road connections being created. The *road finding algorithm* produces the "road network" *final object map* (Figure 4.25, right image), which is analyzed for discrepancies in Figure 4.26. The road network that results from allowing the road finding algorithm to have more aggressive thresholds is considered in Figure 4.27. The "road network" *final object map* is propogated forward to the *finalize GIS layers* step.

4.5.5.1 Road Network Discrepancies. Figure 4.26 shows the original RGB model with roadways labeled, the false color image of the hyperspectral image, the ground truth roadways map, and a list of discrepancies between them and the final spatial roadway map. There are six notable discrepancies. When the "road network" *final object map* is compared to "ground truth", there are six discrepancies to make note of.

1. These areas are actually parking lots and are legitimate roadways. They were not part of the parking lot overlay because of their small size, and they are attached to the primary roadway.
2. These two corner sections have significant segments of roadway next to them with no vegetation between them. The algorithm connected these pieces. The upper one encompasses driveways, and the lower one may be a small parking area. Fine-tuning the algorithm with a Kalman filter or an embodiment of

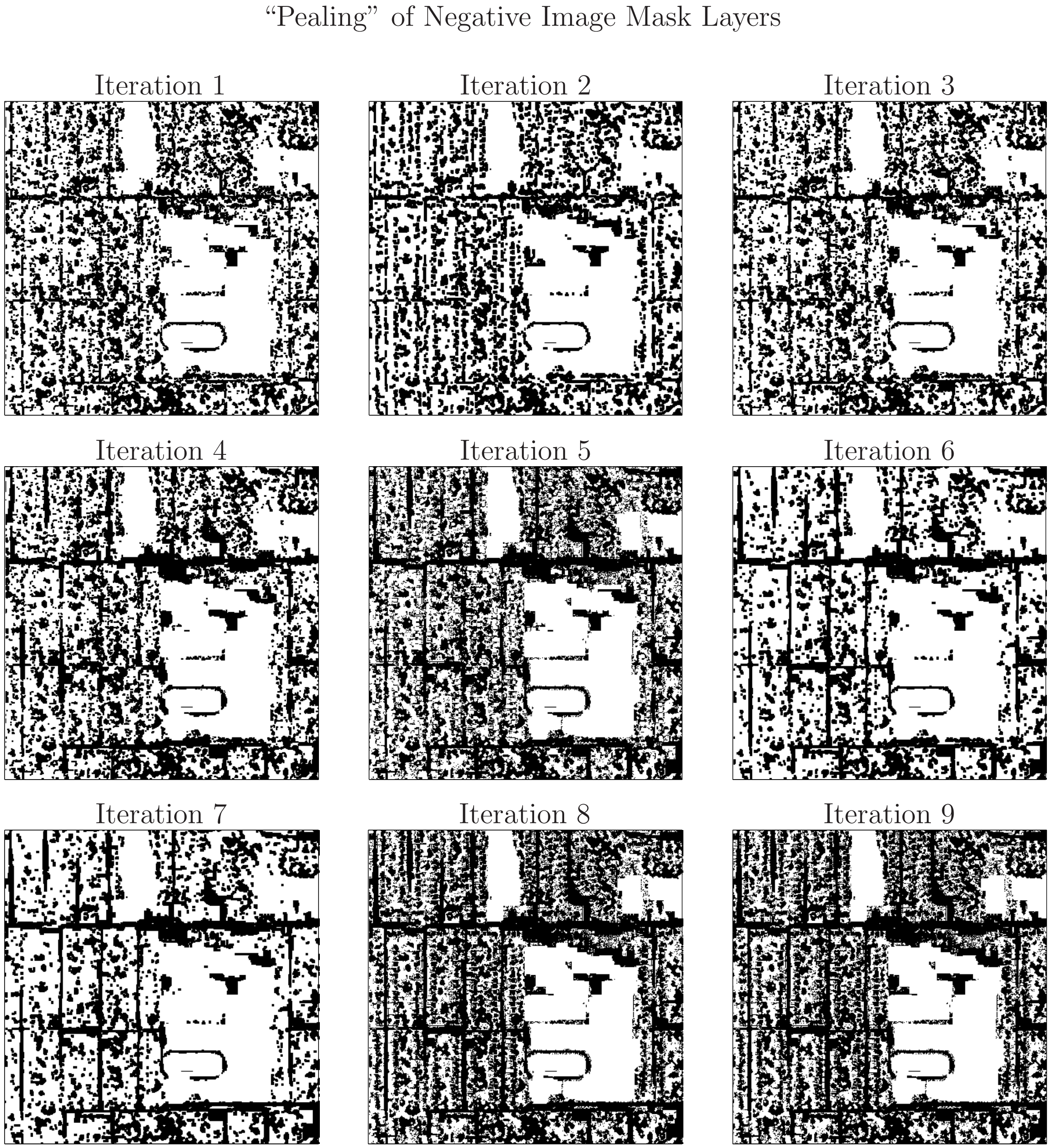


Figure 4.23: The *road finding algorithm* compares the number of negative image mask (NIM) pixels with a binary "true" between the pairs of roadway pixels from the "best known good roadway" *object map* to a NIM threshold when deciding on a possible road connection (Figure 4.24).

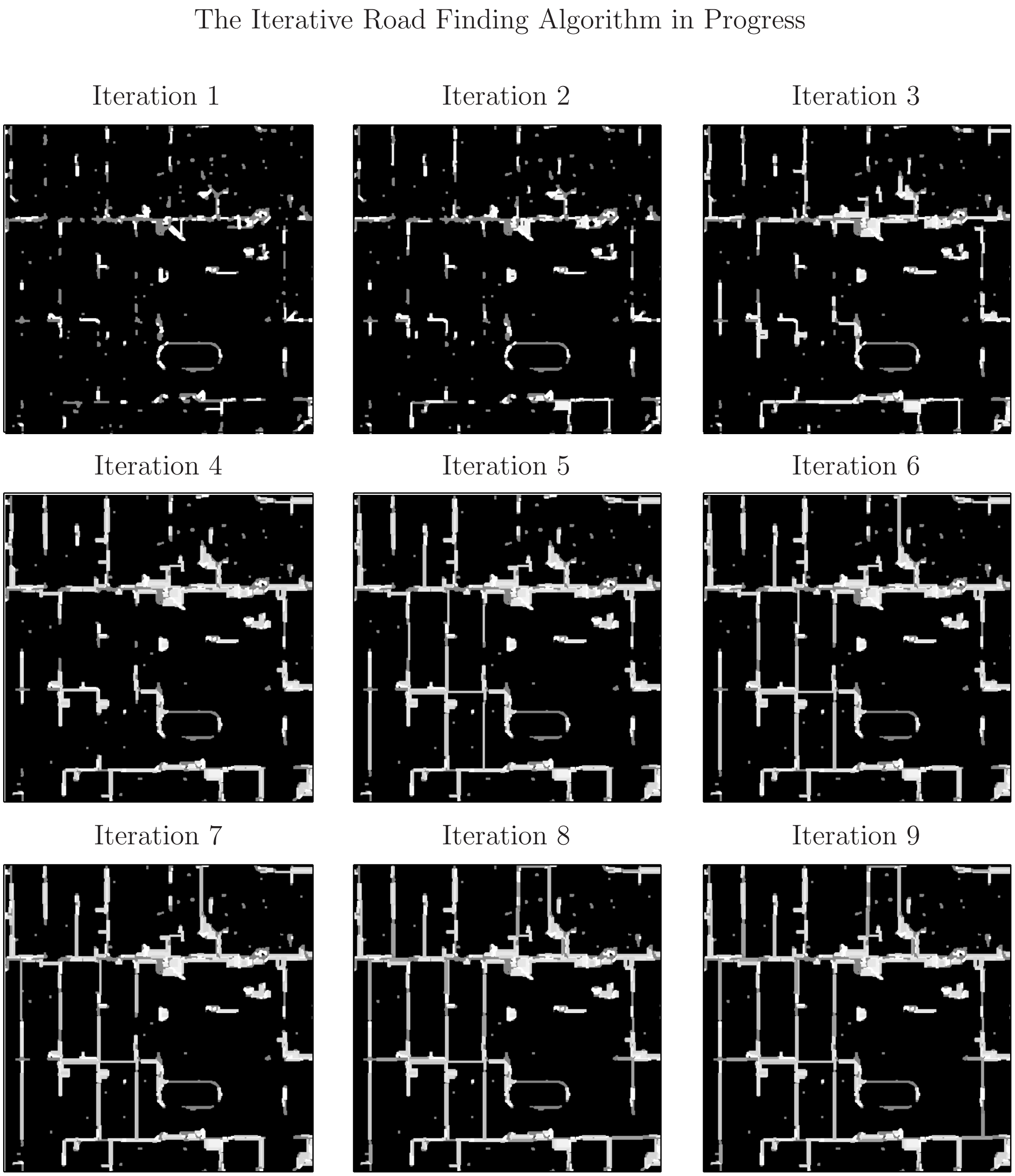


Figure 4.24: The NIM from Figure 4.23 is used in conjunction with the “best known good roadway” *object map* to decide if a possible roadway connection should be drawn between a pair of roadway pixels.

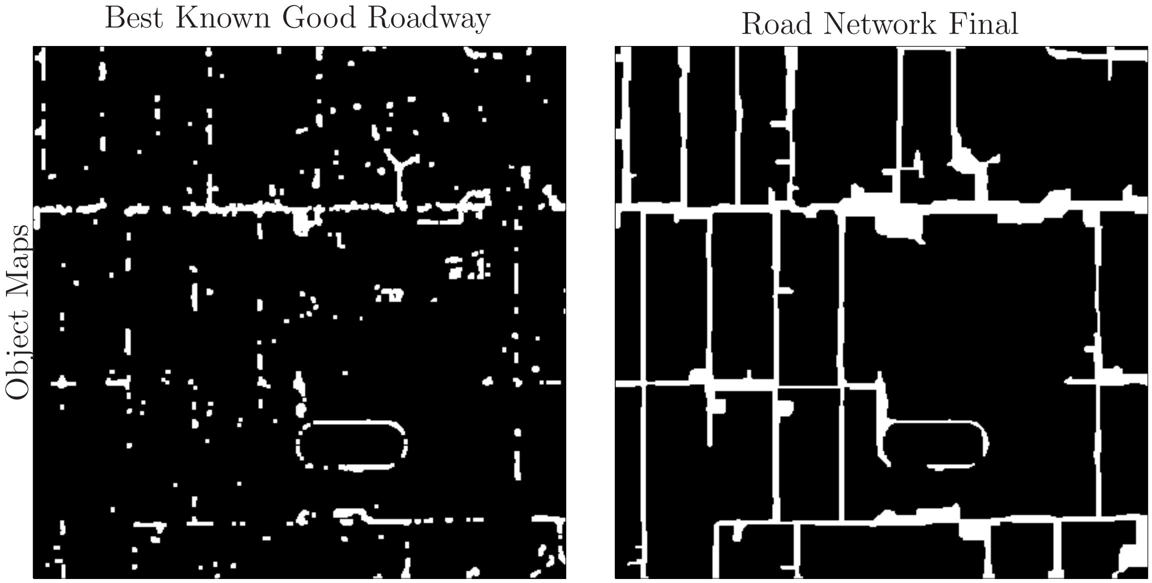
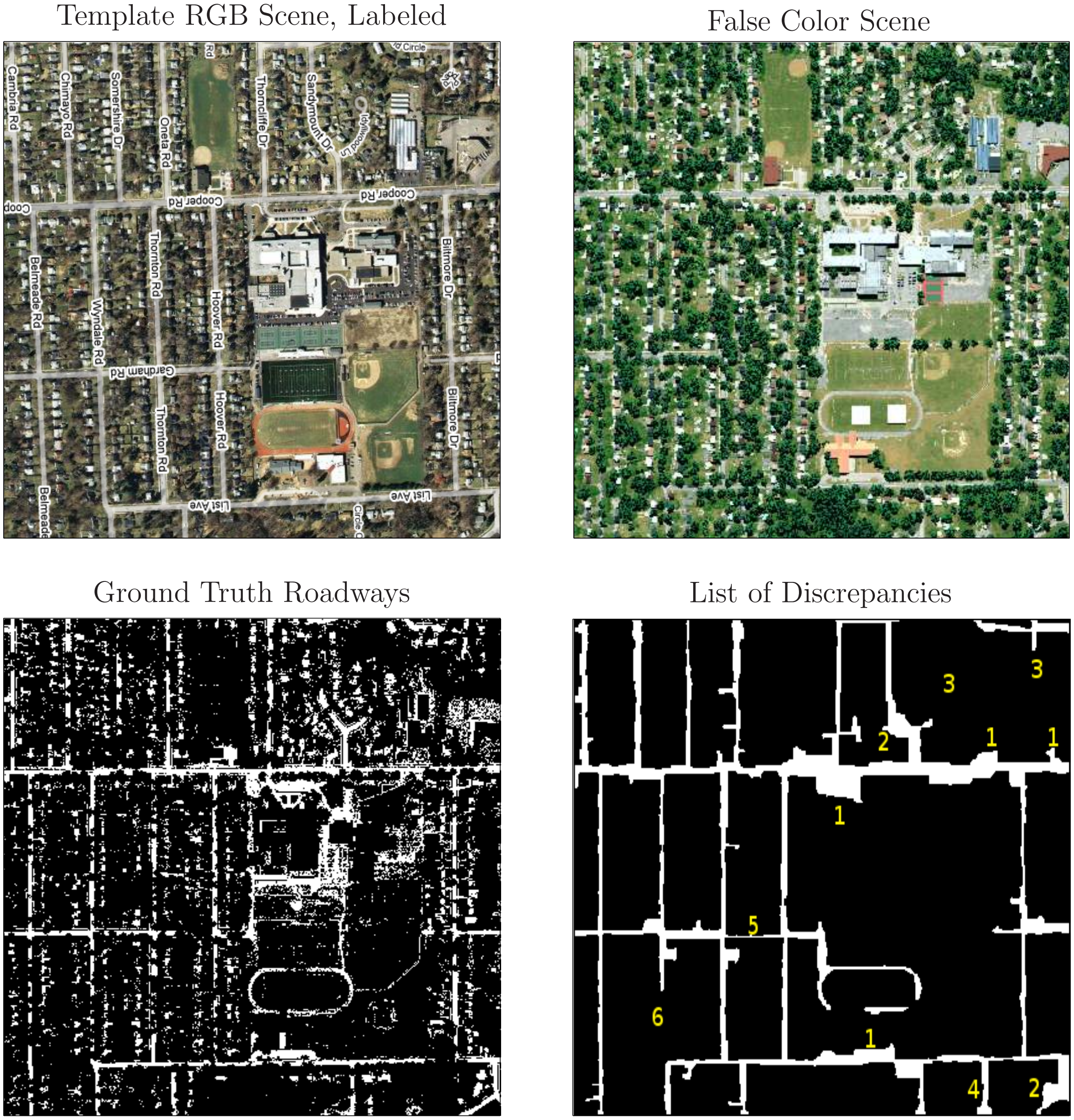


Figure 4.25: The “best known good roadway” *object map* (left image) is processed with the *road finding algorithm* (Figure 4.24) to produce the “road network” *final object map* (right image).

the expected roadway paths would improve the probability of not attaching wayward segments of asphalt if they lie off the primary path.

3. These two cul-de-sacs do not exist in the ground truth roadway, are completely obscured in the hyperspectral false color image, and are only visible in the original RGB template scene when the roadway labels are overlaid.
4. This is an errant roadway. The algorithm looks for possible road connections between existing road segments, and prioritizes the connection if it lies under trees. In this case, both of these conditions are met. An improvement to the algorithm might be to consider how many road pixels actually exist, check previous roadway maps for in-between pixels, and refuse the connection if it is primarily under trees.
5. This segment of roadway may or not may exist, it is unclear. In the RGB template scene, it is there. In the false color image, it may not be there. The ground truth roadway indicates that a road segments may exist.

Analyzing Roadway Discrepancies



Template RGB Scene, Labeled

False Color Scene

Ground Truth Roadways

List of Discrepancies

Figure 4.26: When compared to the original satellite image (upper left), the false color composite of the hyperspectral scene (upper right), and the "ground truth" roadway surface object map (lower left), there are six notable road network discrepancies (lower right). 1. These are parking areas identified as road network. 2. The road curves are not handled well. 3. There are two cul-de-sacs that are completely occluded by tree foliage. 4. This roadway does not exist in the "ground truth". 5. It is undetermined if this roadway exists or not. It exists in the satellite image (upper left), does not exist in the false color image (upper right), and may exist in the ground truth roadway object map (lower left). 6. Road segment is missing.

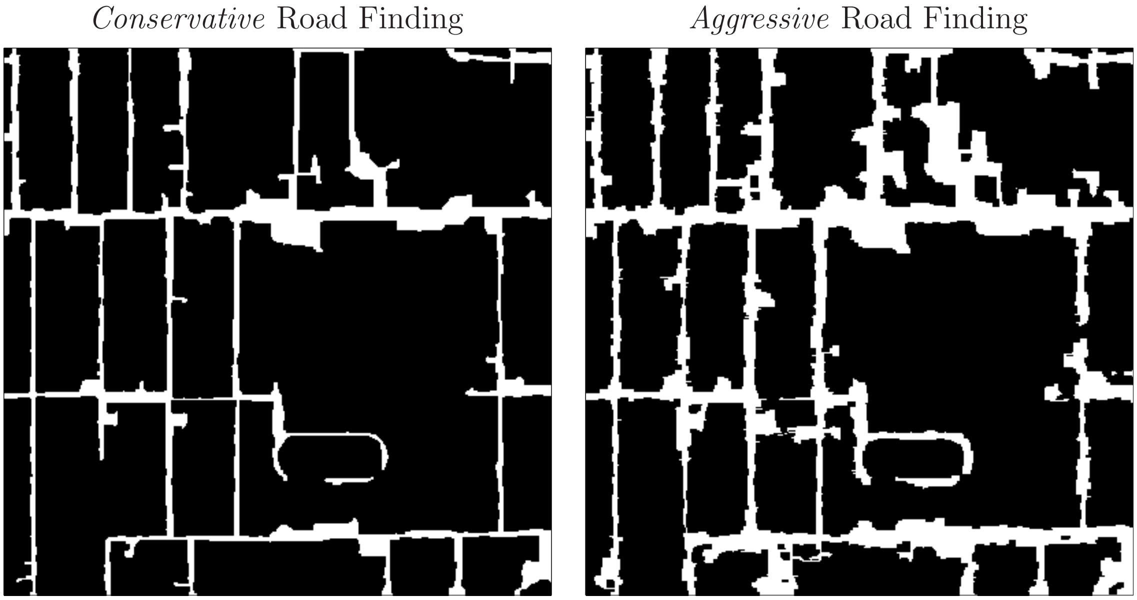


Figure 4.27: A conservative and aggressive approach to find possible road connections.

6. This missing segment is due to the fact that there does not exist a straight-line path between the two end pieces. The algorithm looks for road segments in the eight cardinal directions (every forty-five degrees of the circle). Improving the algorithm to consider the entire circle would significantly improve the probability of finding the connection.

4.5.5.2 Aggressive vs. Conservative Road Finding. The *road abstraction algorithm* has the option of an aggressive (Figure 4.27, left image) or conservative (Figure 4.27, right image) approach to locating possible roads. Both approaches are illustrated in Figure 4.27. The aggressive approach relaxes the NIM threshold shown in Figure 3.13. The danger of the aggressive approach is that it results in the inclusion of less significant "roadway" pixels (e.g. asphalt driveways) in the "road network" *final object map.* Furthermore, the aggressive approach connects over small patches of grass representing urban front or back lawns[2].

---

[2] Linear processing may remove some of these artifacts

4.5.6 Step 6: The "Non-Tree Vegetation" and the "Buildings" Final Object Maps. Section 3.5.6 describes the process used in step 6 to create the "non-tree vegetation (NDVI > 0.3)" and "buildings" *final object maps.* The reasons these two maps are processed last are unique to each object map. The "non-tree vegetation (NDVI > 0.3)" *initial object map* (Figure 4.28, left image) exhibits vegetation pixels where roadways exist. The "road network" *final object map* is created in Step 5, and it is logically "deleted" from the "non-tree vegetation" *initial object map.* The map is then processed for visual appeal with a morphological close using a 2×2 SE to create the "non-tree vegetation" *final object map* (Figure 4.28, right image). The "non-tree vegetation" *final object map* is propogated forward to the *finalize GIS layers* step.

Due to the spectral similarity of the asphalt rooftop shingles and the asphalt roadway surface, the "rooftop" *initial object map* contains a small number of roadway pixels that persist into the intermediate "buildings" *object map.* Now that the "road network" *final object map* is known, the roadway pixels are removed from the intermediate "buildings" *object map* (Figure 4.29, left image) using a logical "deletion" resulting in the "buildings" *final object map* (Figure 4.29, right image). The "buildings" *final object map* is propogated forward to the *finalize GIS layers* step.

The spectral analysis process began with six *initial object maps* that processed in six steps into the eight *final object maps.* The *eight final object maps* are used as the basis to finalize eight GIS layers in the next step.

4.6 Finalizing the GIS Layers

The *finalizing the GIS layers* step is shown in Figures 3.15. This is a procedural step similar to the *create initial object maps* because no analyis is accomplished. To convert a *final object map* into a GIS layer, the objects in the object maps are "outlined" with a morphological *remove* operation. The object map is used as an object transparency mask. By applying a color to the transparency mask and separately varying the opacity of the object transparency mask and the shape outlines produce a visually appealing GIS layer that can also be overlaid over the false color composite

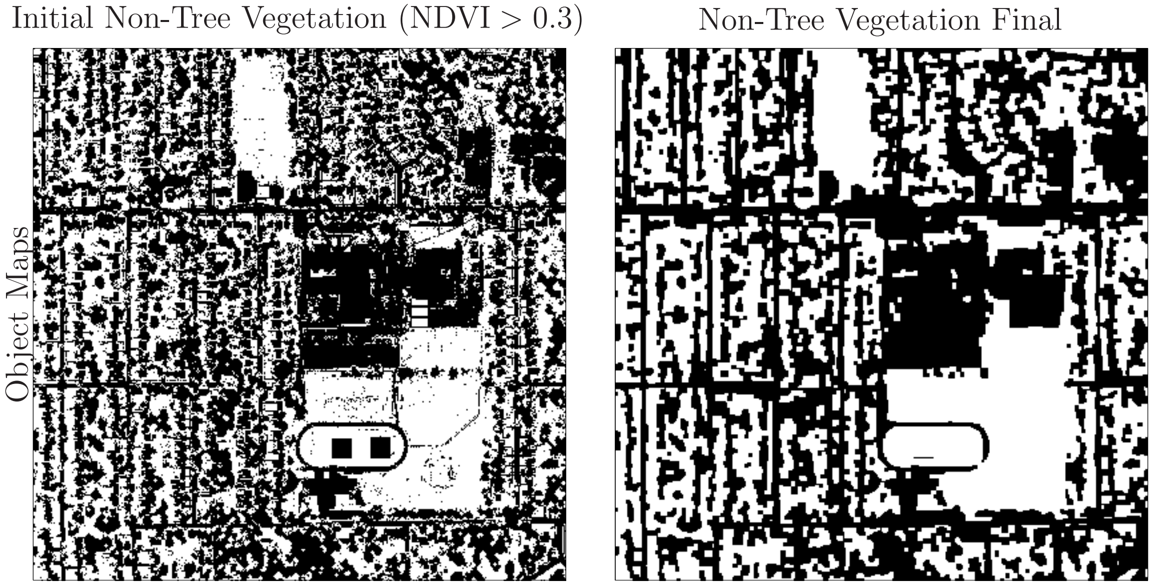

Initial Non-Tree Vegetation (NDVI > 0.3)
Non-Tree Vegetation Final
Object Maps

Figure 4.28: The "road network" *final object map* is logically "deleted" from the "non-tree vegetation (NDVI > 0.3)" *object map* (left image), is processed with a morphological close operation with 2×2 for visual appeal, and results in the "non-tree vegetation" *final object map* (right image). White pixels are "non-tree vegetation" and the black pixels are "not non-tree vegetation".

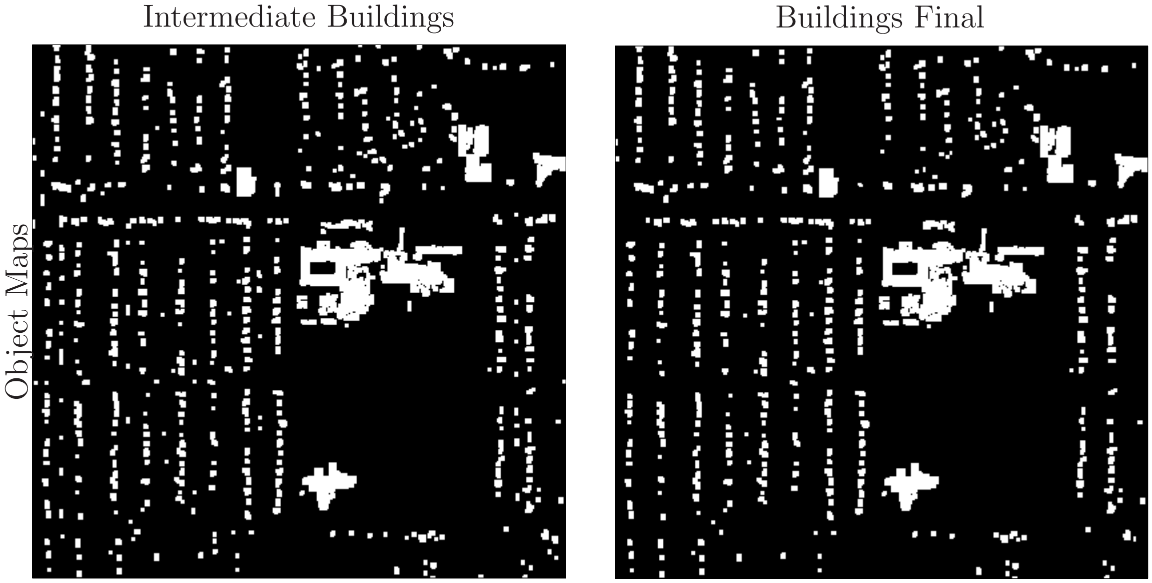


Figure 4.29: The "road network" *final object map* is logically "deleted" from the intermediate "buildings" *object map* (left image) and results in the "buildings" *final object map* (right image). White pixels are "buildings" and the black pixels are "not buildings".

image of the hyperspectral scene for visual verification. This is done in Figures 4.30 through 4.36.

4.7 Summary

Figures 4.30 through 4.36 show the final GIS layers. Table 4.7 summarizes the accuracies resulting from the spectral analyis and from the spectral followed by spatial analysis. The GIS layers are compared to the “ground truth” information. The three spatial maps that are both abstractions and have no corresponding spectral analysis result (Parking Areas, Fields, and Major Buildings) had the highest accuracies. The buildings, road network, tree, and grass GIS layers all improved significantly. The vegetation GIS layer, however, is the only one that is reduced in accuracy.

Table 4.7: Listing of final GIS layer accuracies. The table is ordered from high to low in the third column.

Final GIS Layer	Spectral Analysis	Spectral & Spatial Analysis	Reference Figure
Parking Areas		0.9935	Figure 4.37
Fields		0.9851	Figure 4.33
Major Buildings		0.9825	Figure 4.35
Buildings	0.7783	0.9668	Figure 4.34
Road Network	0.8673	0.9284	Figure 4.36
Vegetation	0.9949	0.9238	Figure 4.30
Tree	0.8201	0.8684	Figure 4.31
Non-Tree Vegetation	0.8159	0.8669	Figure 4.32

The accuracy for each of the *object maps* was expected not to vary significantly from the spectral analysis process because of the significant morphological operations used to improve the visual appeal of the object maps. The “vegetation”, “non-tree vegetation”, and “tree” GIS layers are derived directly from *SOM segmentations* using only morphological operations and some logical operations, and they improved the least.

The "buildings" map improved the most, which is a significant demonstration of the negative image mask (NIM) concept used to determine the most likely building locations. By using the inverse of the NIM with the "rooftop" *initial object map*, the final GIS layer buildings accuracy is increased. Another contributor to increased buildings accuracy is the removal of non-building pixels by logically "deleting" the "road network" *final object map.*

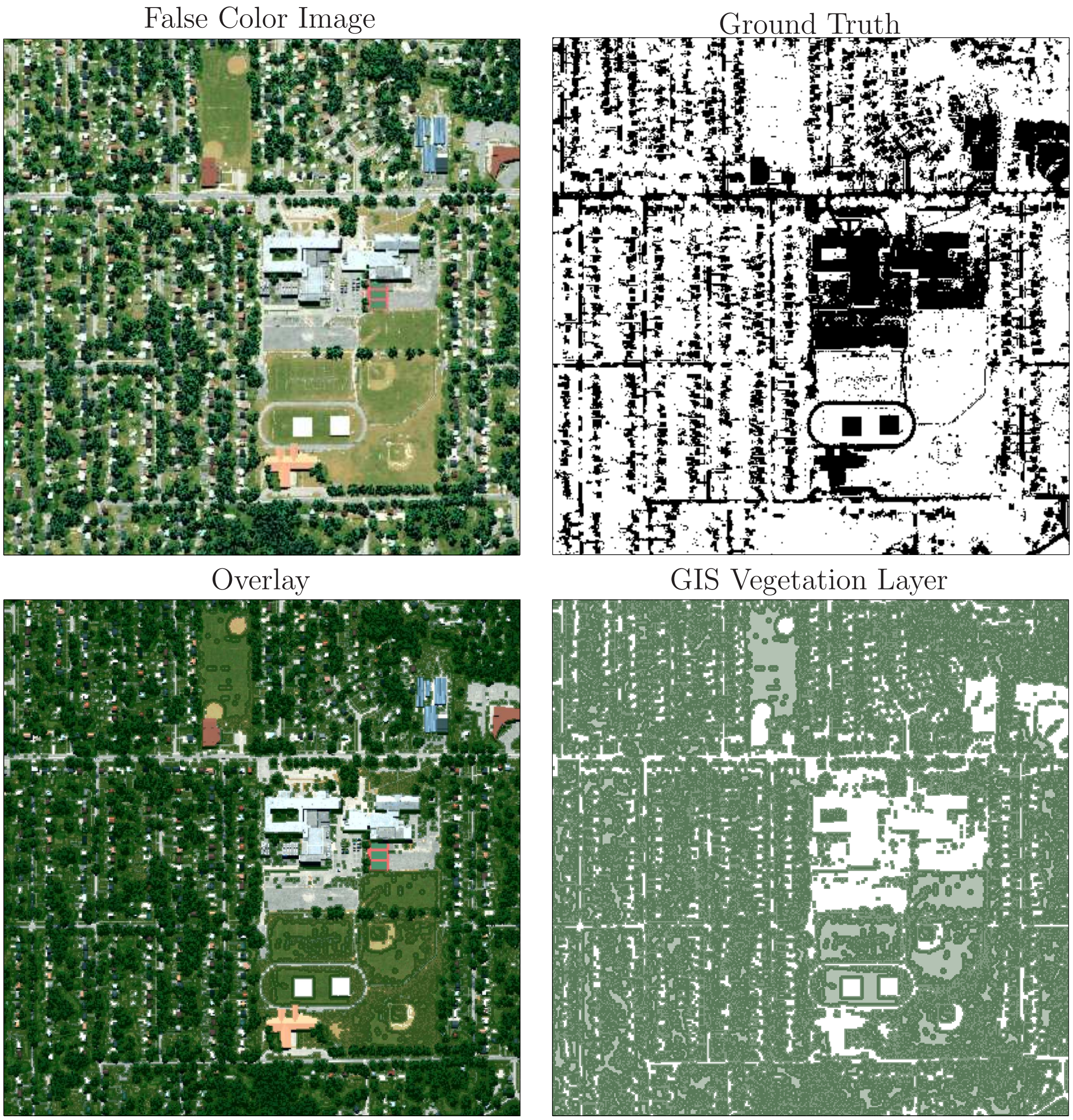
Finally, the "vegetation" GIS layer in the lower right image of Figure 4.30 in contrast to the "vegetation" *initial object map* in the left image of Figure 4.12 shows significant deviations that were used to make the GIS layer more aesthetically appealing through morphological operations. The knowledge contributed by the GIS layer is not decreased despite the decrease in the pixel-level accuracy calculation of Table 4.7. This decrease in accuracy highlights the challenges of relying on pixel-level accuracy calculations when abstracted scene object knowledge is prioritized. The "ground truth" labels for the hyperspectral scene do not contain abstract knoweldge of the scene. For example, there is not label "parking areas". The "ground truth" contains the labels "asphalt roadway old" and "asphalt roadway new".

Despite the challenges of using pixel-level accuracy calculations, tweaking road network parameters, and adjusting the SE sizes to match the in-scene object sizes, the results are compelling. The "tree" GIS layer in Figure 4.31 provides the "big picture" locations of the tree locations and compares favorably to "ground truth". The "non-vegetation" GIS layer in Figure 4.32 clearly shows the road network, and locations of the large contiguous grassy areas. The "fields" GIS layer in Figure 4.33 outlines the two major contiguous grassy areas. The "buildings" GIS layer in Figure 4.34 correctly identifies the urban houses and major buildings, and the "major buildings" GIS layer in Figure 4.35 correctly identifies the five major buildings. The road network is completely described by the "road network" GIS layer in Figure 4.36. The road network is significantly tree-occluded in some cases, it is difficult to visually verify (see Figure 4.36, upper left image) the existance of some road network segments. The *road finding algorithm* overcame all of the tree occlusions to abstract the road network. The four significant parking areas are correctly identified in the "parking areas" GIS layer shown in Figure 4.37. In every case, the object shapes in the GIS layers match the feature shapes in the false color composite of the hyperspectral image. Finally, overlaying the road network, parking areas, major buildings, fields, all buildings, and trees in Figure 4.38 demonstrates that the shapes from the individual layers "fit together" in the same manner as in the original scene.

The GIS layer results have surpassed all expectation. Using a SOM to segment an urban environment in [26] resulted in an overall 80% accuracy, and corresponds to the "spectral-procesing-only" accuracies from Table 4.7. The value of morphological spatial operations is demonstrated. The morphological operations are also robust and operate in a similar manner to human perception. The morphological open operation deletes stray pixels that a person ignores as "noise" or "scene clutter", and the morphological close operation clusters pixels that a person would mentally group together. The average classification resulting from spectral analysis (84.3%, see Table 4.5) increased after spatial processing to an average of 93.9% (see Table 4.7), surpassing previous attempts by other authors.

4.8 Conclusions

Chapter IV covers a detailed step-by-step implementation of the process outlined in Chapter III and Figure 3.1 to convert a synthetic hyperspectral image (Figure 3.2) into eight GIS layers (Figures 4.30 through 4.36). It takes four steps to complete the process. The first step is spectral analysis (Figure 3.3), creating the initial object maps (Figure 3.10), spatial analysis (Figure 3.12), and finalizing the GIS layers (Figure 3.15). The pixel-level accuracy calculations in Table 4.7 show improvement and significant improvement in all but one case through the use of spectral and spatial analysis in combination. The end product has, on average, an accuracy of 93.9%, improving over previous attempts in abstracting scene content.



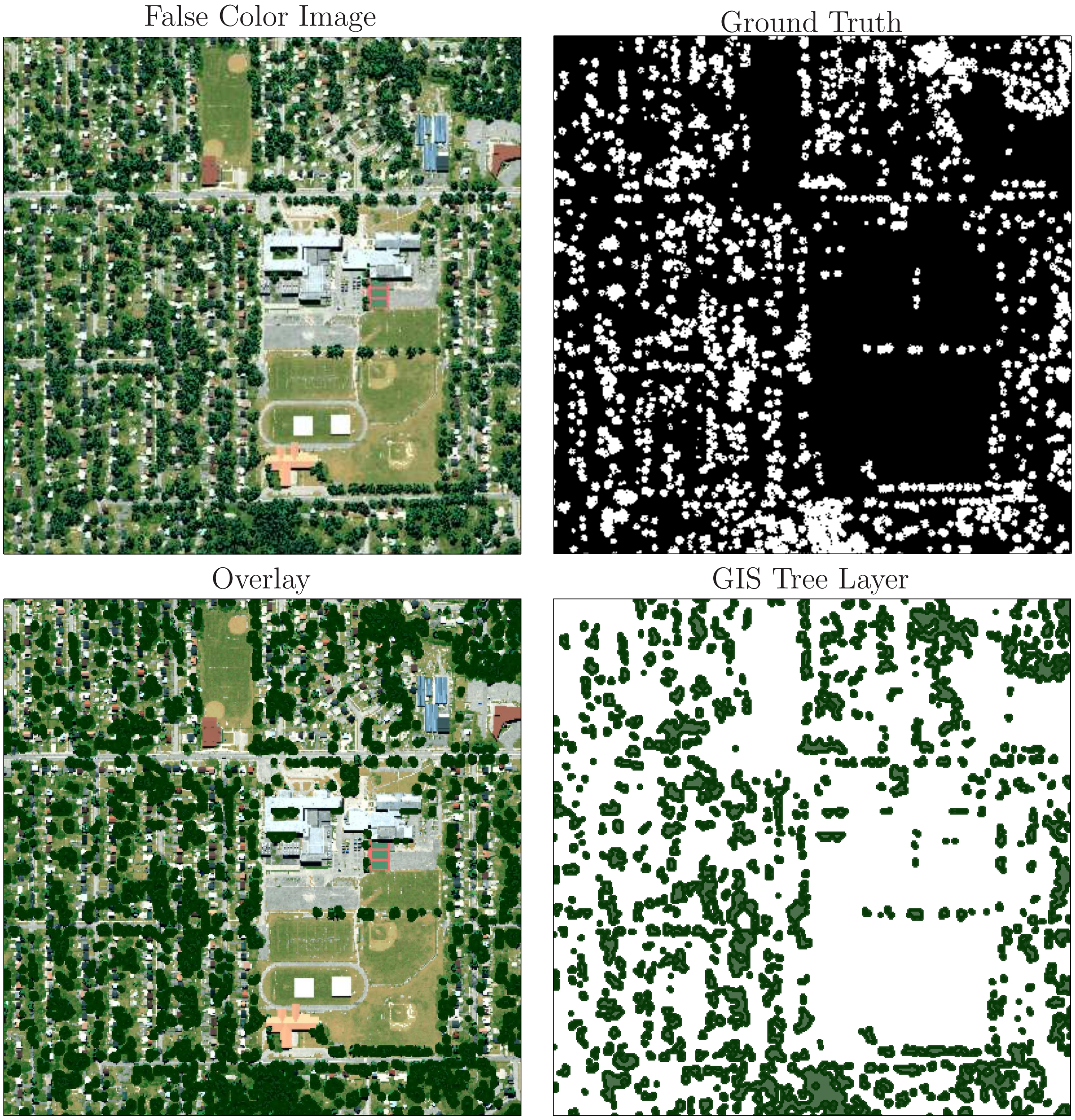
False Color Image

Ground Truth

Overlay

GIS Vegetation Layer

Figure 4.30: Comparing the vegetation results.



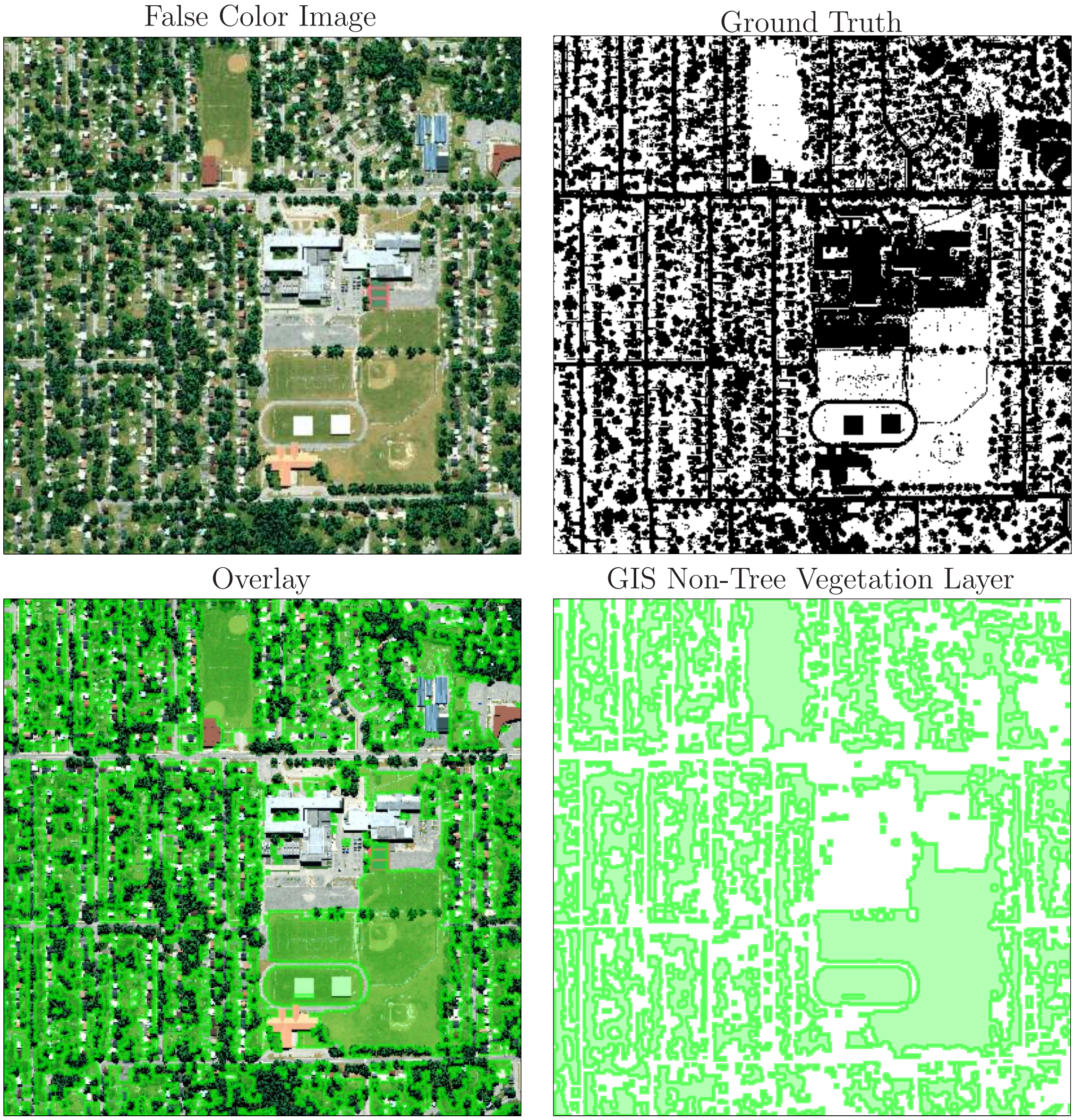
False Color Image

Ground Truth

Overlay

GIS Tree Layer

Figure 4.31: Comparing the tree results.



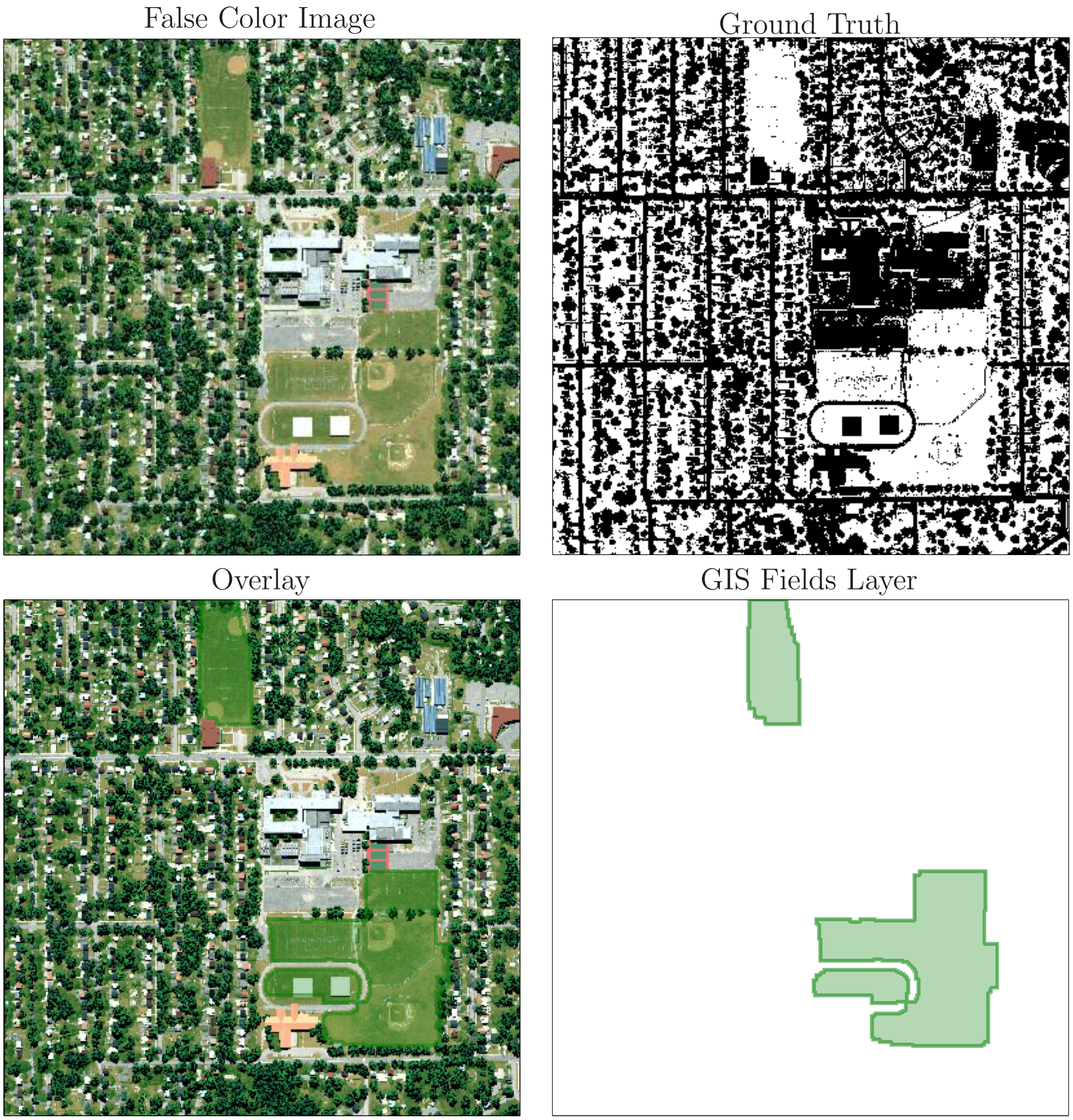
False Color Image

Ground Truth

Overlay

GIS Non-Tree Vegetation Layer

Figure 4.32: Comparing the grass results.



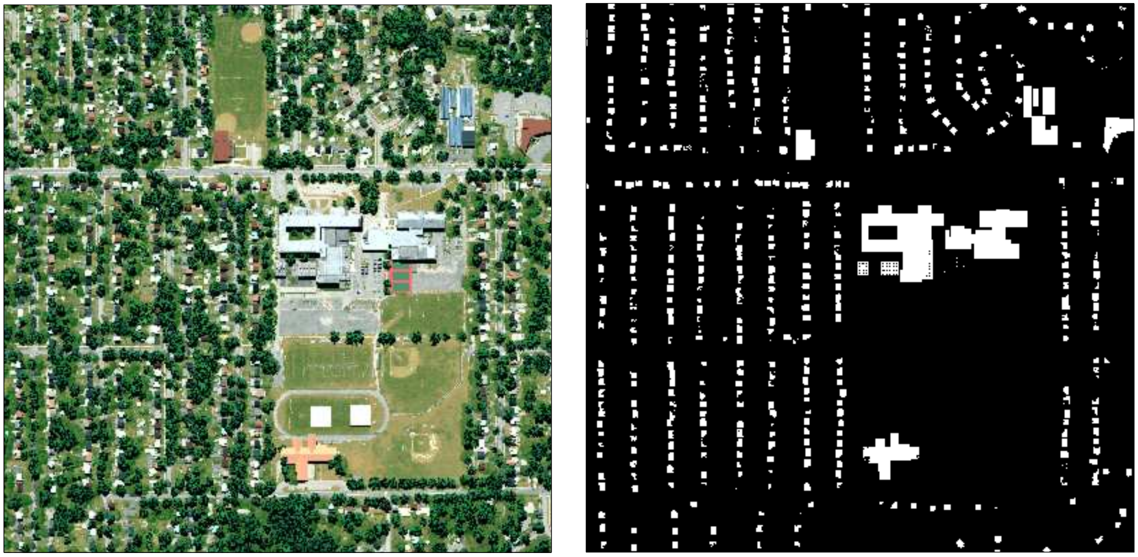
False Color Image

Ground Truth

Overlay

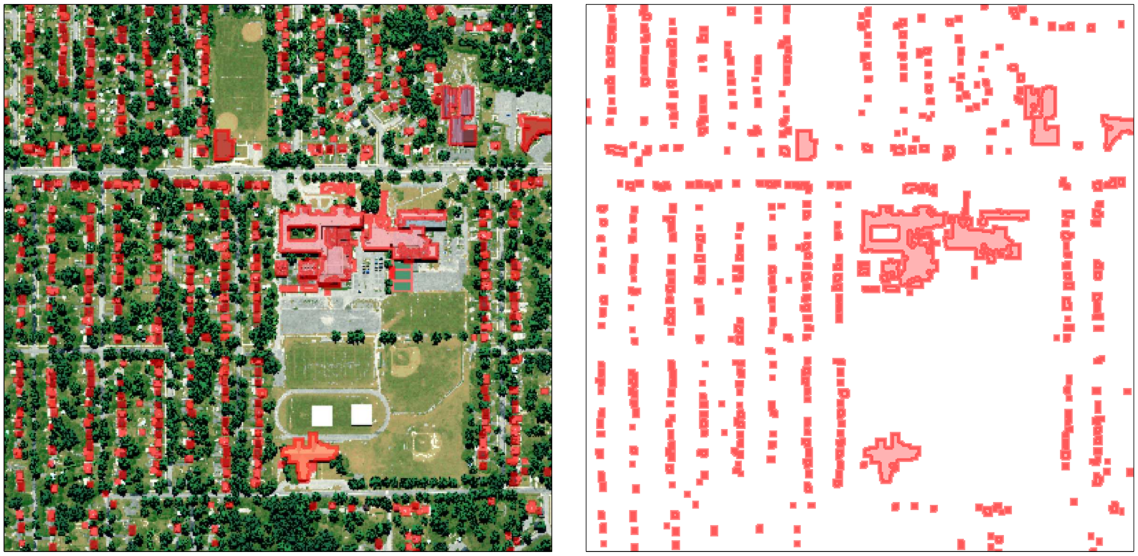
GIS Fields Layer

Figure 4.33: Comparing the fields results.



False Color Image

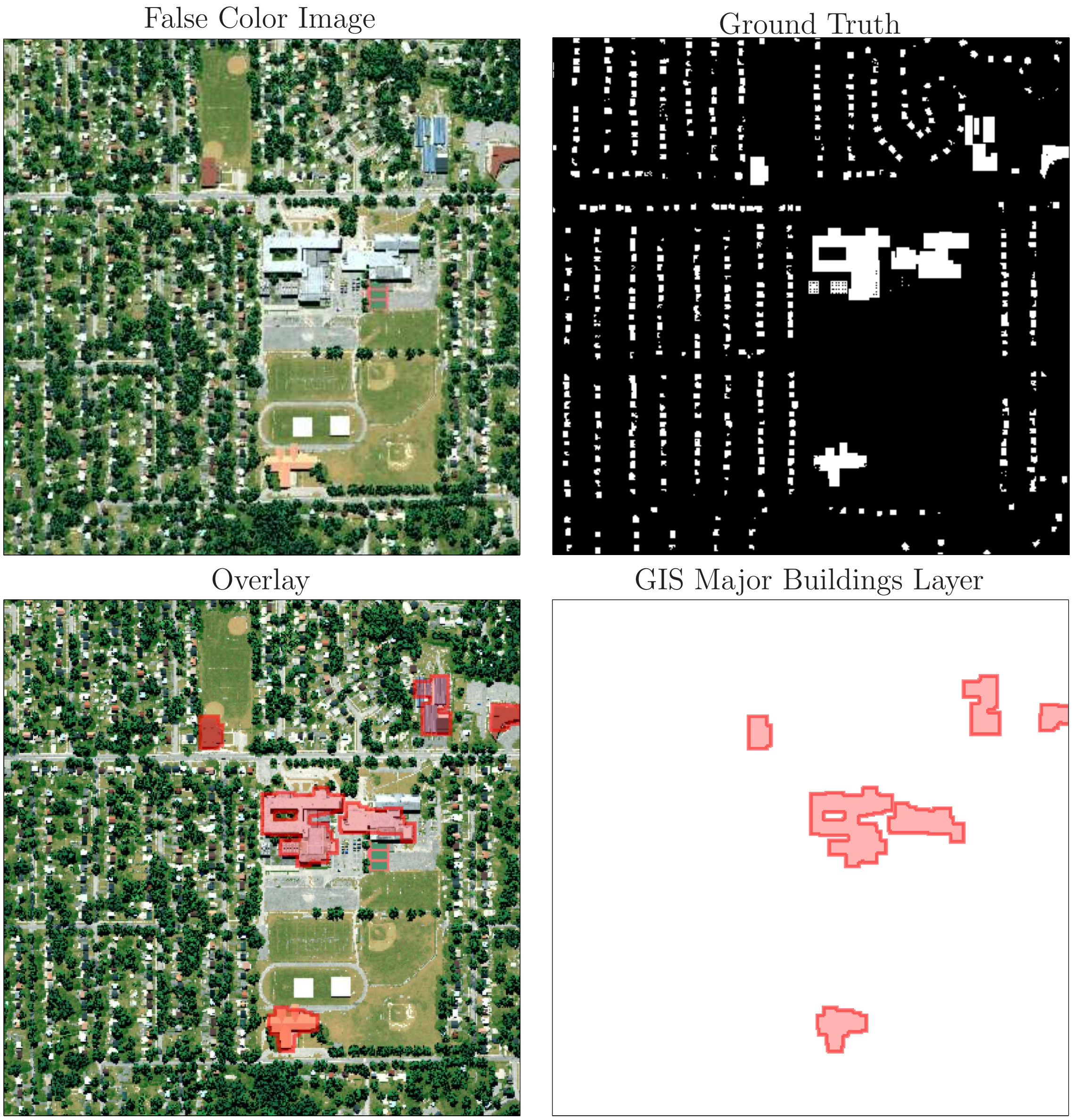
Ground Truth



Overlay

GIS Buildings Layer

Figure 4.34: Comparing the buildings results.



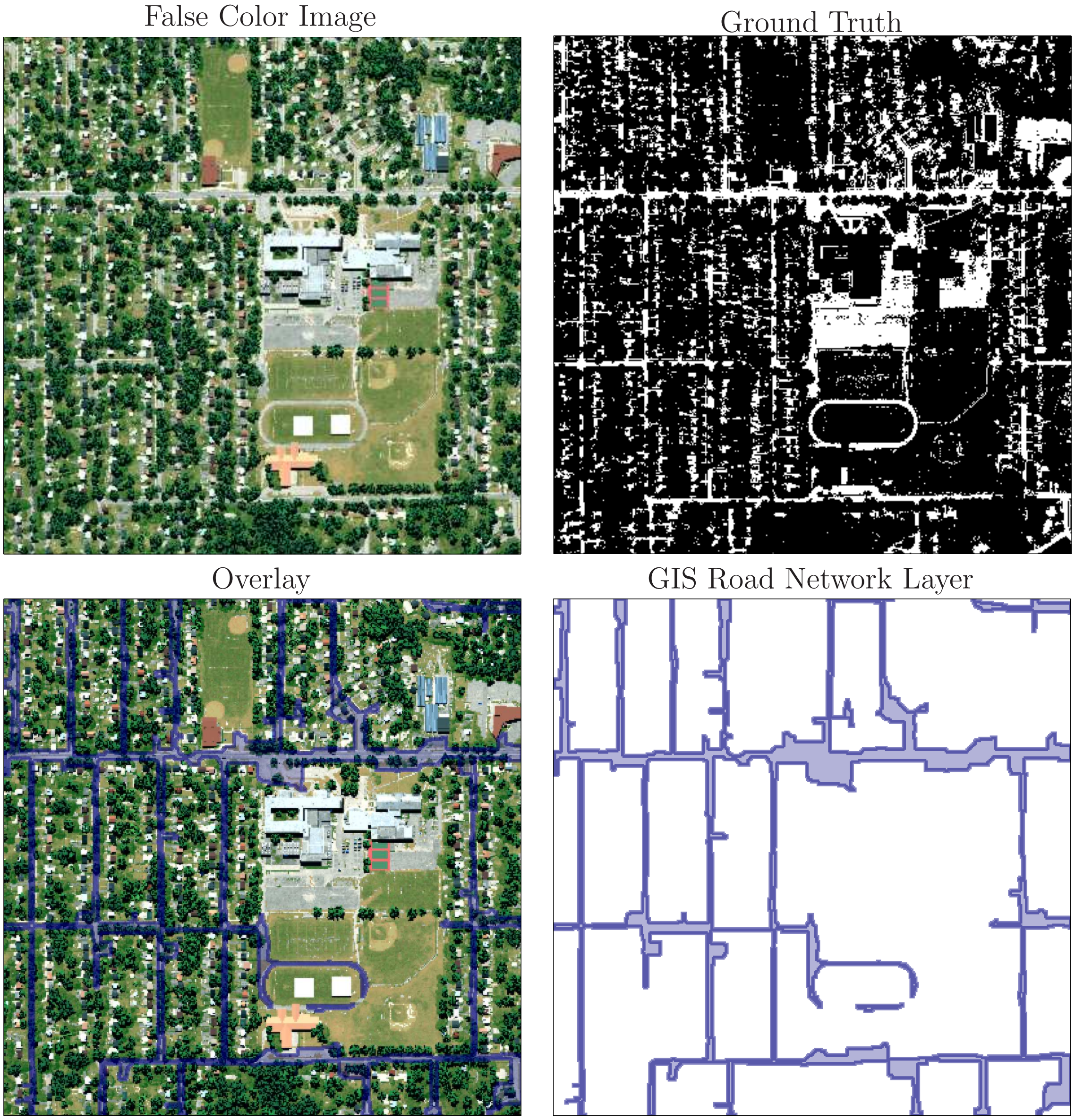
False Color Image

Ground Truth

Overlay

GIS Major Buildings Layer

Figure 4.35: Comparing the major buildings results.



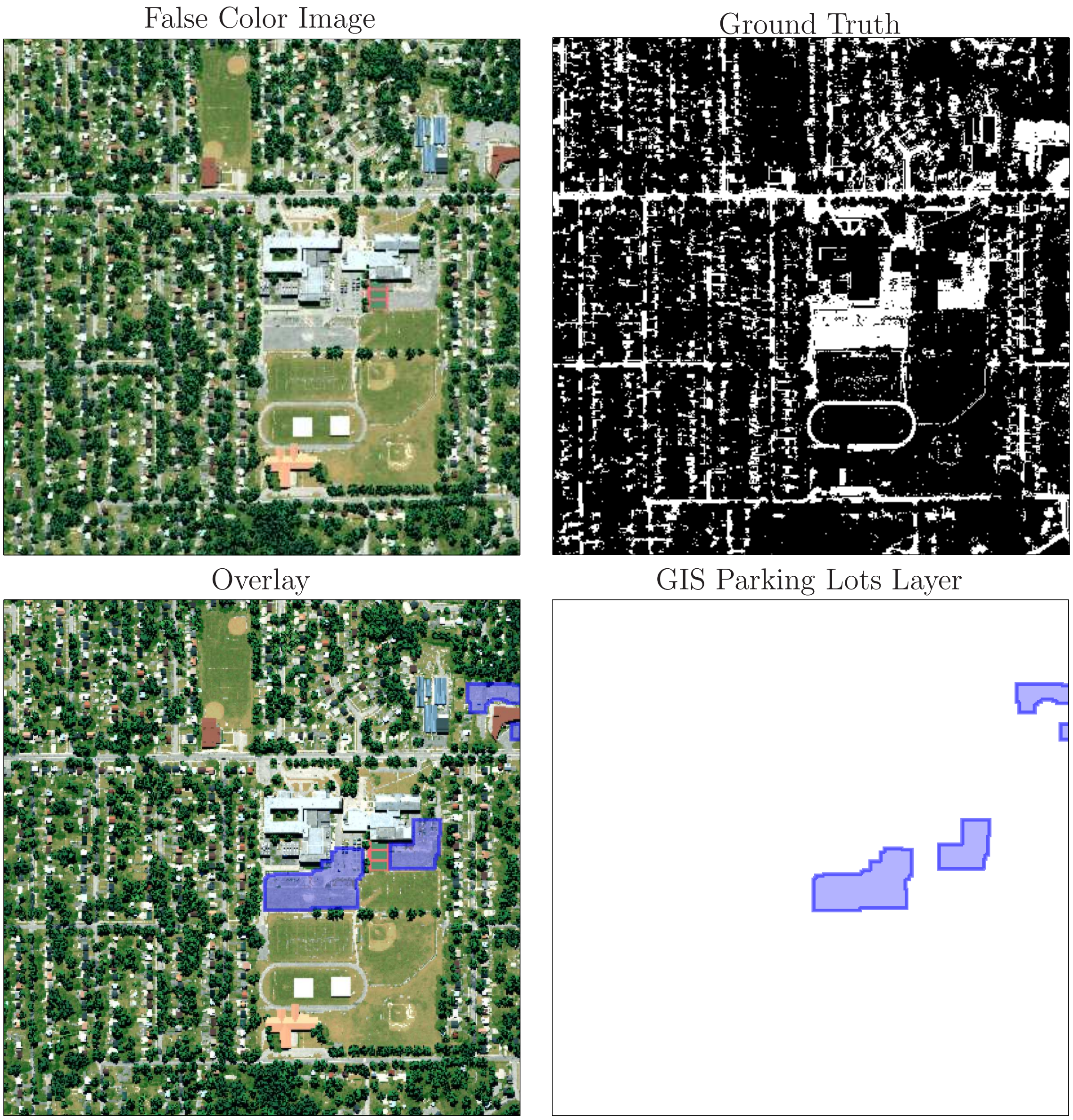
False Color Image

Ground Truth

Overlay

GIS Road Network Layer

Figure 4.36: Comparing the road network results.



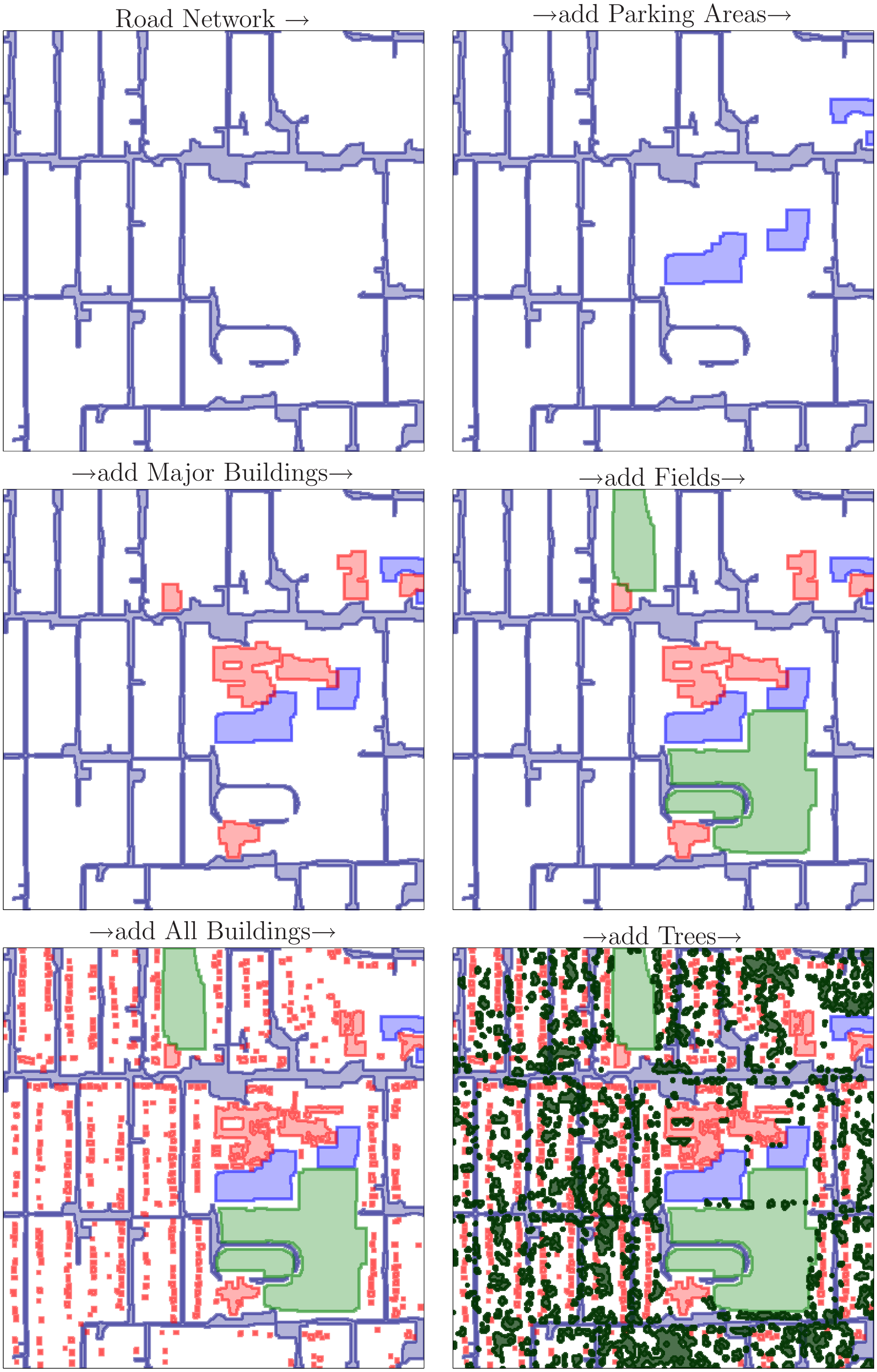
False Color Image

Ground Truth

Overlay

GIS Parking Lots Layer

Figure 4.37: Comparing the parking lot results.



Road Network →

→add Parking Areas→

→add Major Buildings→

→add Fields→

→add All Buildings→

→add Trees→

Figure 4.38: Combining GIS layers demonstrates the fit of shapes.

V. Future Work and Concluding Remarks

The results of using a Self-Organizing Map (SOM) to abstract GIS shape files are excellent. The GIS layers are derived from a *hyperspectral image* using a *semi-automated* predictable process that is resistant to human error. *Minimal* interpretation of the hyperspectral scene is necessary because of the available spectral information that was analyzed to generate the final results. Furthermore ground truth information was *not* collected. The shape files *follow* the features of the hyperspectral scene in two dimensions and provide an excellent abstraction of the hyperspectral image information.

By using morphological operations to cluster pixels together and delete non-clustered pixels, Geographic Information Systems layers were successively abstracted from a hyperspectral image cube. No *a-priori* knowledge of the endmember spectra, spectral databases, or any other information was brought to the analysis.

5.1 Supported Research

This thesis supports emerging technologies that rely heavily on quality GIS layers being available. Abstracting features and identifying objects is foundational to several in-progress research projects. Two these technologies are vehicular behavior prediction and context-related automated target recognition.

5.1.1 Vehicular Target Behavior Prediction.

The work in [31] is developing a behavior prediction system that depends heavily on available roadway and building GIS layers (see Figure 5.1). The GIS information used in [31] was obtained from US municipal governments. Obtaining similar information for hostile territory or from a hostile government may not be possible or advantageous. Therefore, an automated way to extract these features, as accomplished in this thesis, is necessary.

5.1.2 Target Detection with Shadows in SAR.

The work in [12] uses Synaptic Aperture Radar (SAR) imagery to detect targets using shadows. The results currently has a high false alarm rate associated with buildings and trees. Incorporating accurate

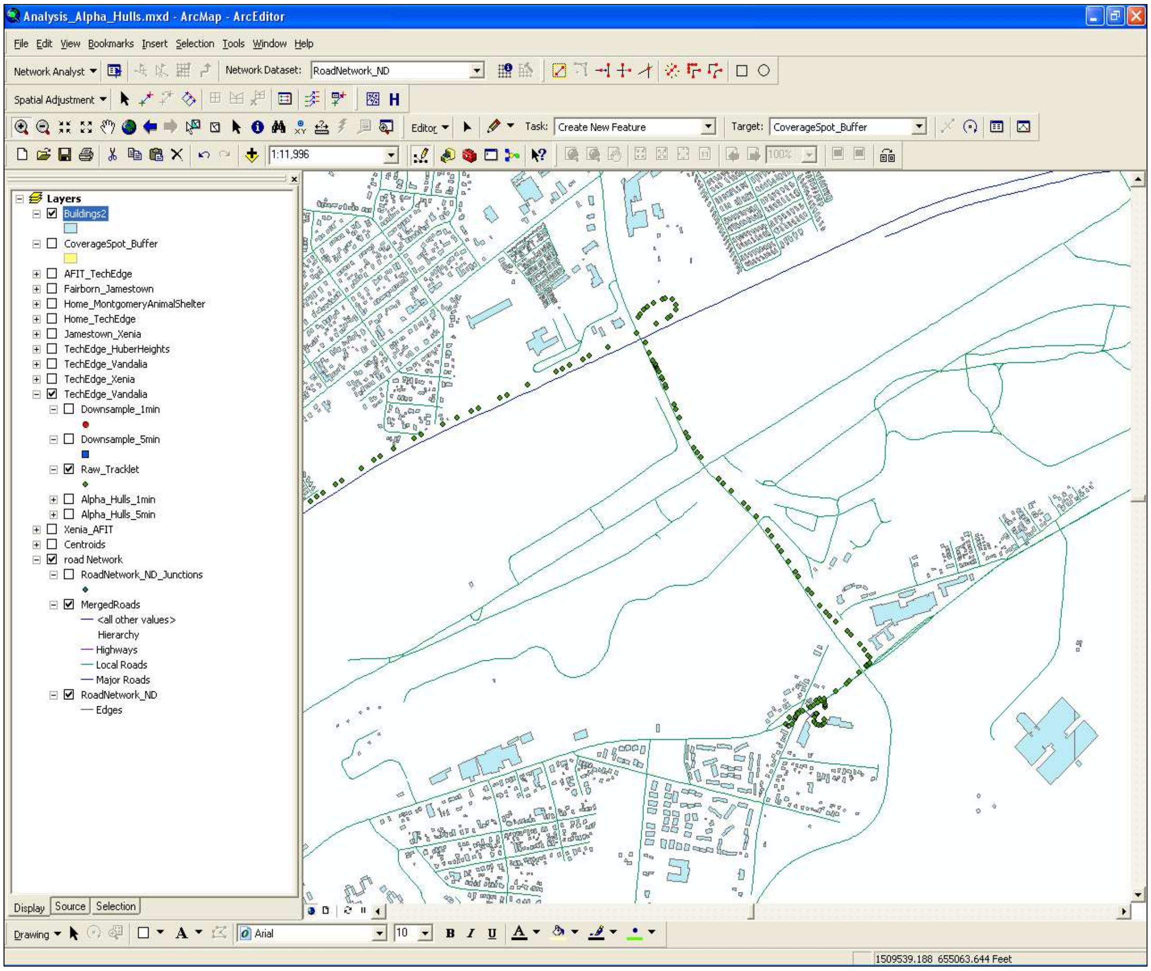


Figure 5.1: This thesis work uses available GIS road network and building layers to predict vehicle behavior. The GIS layers were obtained from US municipal governments, an option that may not be possible from hostile territories. This justifies the approach used in this thesis, and the GIS layer results that can be obtained through remote and passive surveillance.

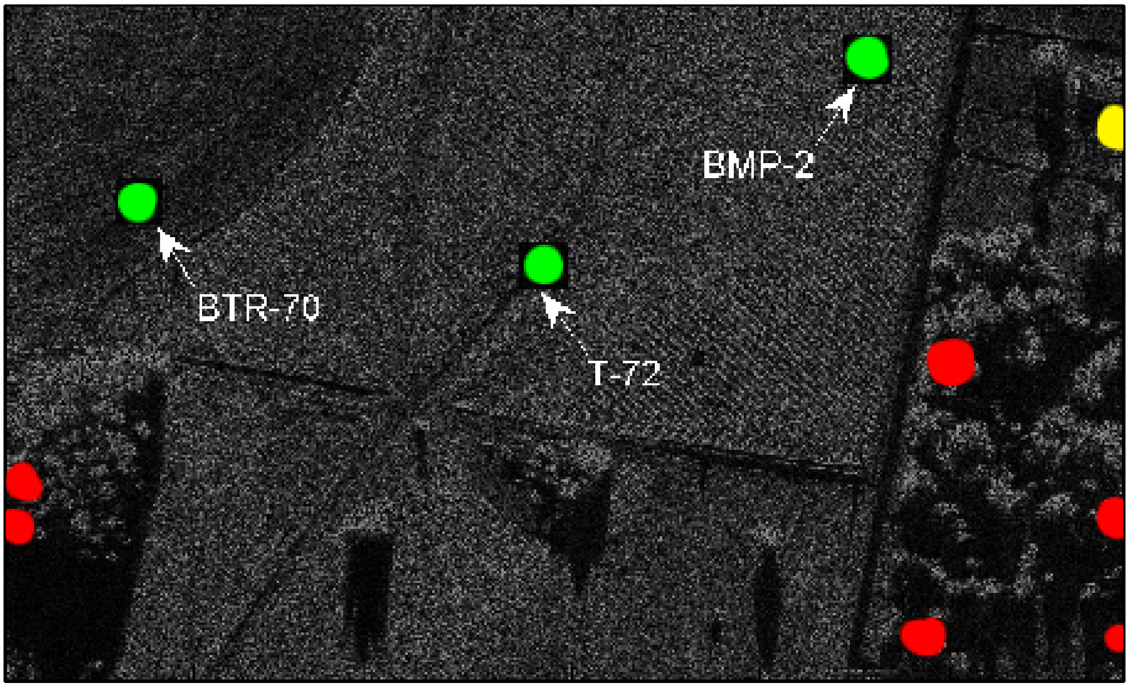


Figure 5.2: The green dots are positive detections, the yellow dot is a false detection due to a building, and the red dots are false detections due to trees. Having the tree and building coverage information would drive the false alarm rate of this SAR detection algorithm to zero.

building and tree coverage drives the false alarm target detection rate to near zero (see Figure 5.2). The techniques developed in this thesis provide building and tree coverage information adequate to improve the process in [12].

5.2 Using the GIS Layers as Intelligence Products

Intelligence products need abstract, yet precise, information to be actionable by commanders for battle planning and execution. Imagery is normally delivered as-is, or is marked by intelligence analysts after some delay. The human-in-the-loop intelligence product is still required to make final targeting decision, but knowledge of roadways, fields, parking lots, major and minor buildings, and tree locations may improve the ability to make better informed and faster decisions.

5.3 Future Work

The first area to address is the SOM. In [7], the author creates a "Parameter-Less SOM" (PLSOM) which removes the need to provide SOM training parameters. Automating the segmentation of the SOM would futher reduce the probability of human error. The tree segmentation process needs to be validated with real data and improved. The building and roadway segmentation process depends on user input, and the inclusion of a spectral database may alleviate this manual labor.

Second, setting *SOM segmentation* thresholds is another area of interest. By setting an expectation for the *object map* with an urban environment layout ruleset, the threshold might be automated. Another approach to SOM segmentation is to perform a spatial edge detection on the SOM to reveal the neighborhood topology, use region growing, or other spatial processing techniques to isolate the neighborhoods.

Third, applying the SOM as a filter for a different hyperspectral image needs to be examined. The SOM should act as a "memory" for similar scenes. An online spectral database created from in-scene spectra could be created that updates on a continual basis. The database could be used after each scene segmentation and also act as a "memory" for future segmentations.

Fourth, combining prior intelligence products, GIS layers, maps and existing knowledge into the spatial analysis process may significantly improve the resulting accuracy. Another interesting source of data to combin in the process is lidar data because a hyperspectral imager flown on board an Unmanned Aerial System (UAS) could easily be coupled with a lidar sensor.

Fifth, persistent surveillance can be simulated with the Digital Imaging and Remote Sensing Image Generation (DIRSIG) [37] software by modifying the camera viewing angle. This can be done by collecting several hyperspectral images over a period time while simulating changing weather, atmospheric conditions, sun angle, and the possible destruction of buildings or obstruction of roadways. The resulting images can be segmented, registered, and may demonstrate an increased "real-world" use of the scene segmentation technology.

Sixth, the scene segmentation can be used in target detection. By simulating moving targets with the DIRSIG software, one can attempt to locate and track specific targets. The author in [34] uses anomaly detection with a SOM to discover rare targets. After scene segmentation, another processing step in this thesis may be to locate on of several vehicles present in-scene. One possible goal is to create a history of vehicle tracks that can be consulted when an in-scene vehicle becomes a *target of interest* in response to some stimulus (e.g. an in-scene explosion, possible from an improvised explosive device).

Finally, the *road finding algorithm*, which is accomplished with only spatial processing techniques, can be improved. Using a Kalman Filter, embedding road connection information, identifying junctions and intersections, and exploring the spatial analysis realm in combination with the negative image mask concept may improve road the abstraction.

5.4 Concluding Remarks

Military warfare at the turn of the millenia is focused on urban combat. As the Air Force augments its capabilities with intelligence, surveillance, and reconnaissance systems geared toward the urban environment, technologies that significantly increase the effectiveness of human labor and battlespace awareness, while taking advantage of existing technologies, are in demand. The technology developed in this thesis exceeds these priorities by providing an on-demand, real-time method of gauging the area of operations. Scene segmentation and the resulting GIS shape files is an enabling technology that fills a critical need for high-quality digital information of the threat environment.

Bibliography

1. *Open Street Map Project*, 2008. URL `http://www.openstreetmap.com/`.
2. *Open Clip Art Library*, 2009. URL `http://www.openclipart.org/`. All clip art used is free and open for unlimited re-distribution.
3. Air Force Institute of Technology, WPAFB OH. *Statistical Machine Learning, Class Notes*, 2008.
4. Akcay, Gokhan and Selim Aksoy. "Morphological Segmentation of Urban Structures". *Urban Remote Sensing Joint Event*, (11-13):1–6, 2007.
5. Almog, O., M. Shoshany, and V. Alchanatis. "Improving hyperspectral classification based on wavelet decomposition 1Ophir Almog". *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, 3806–3809. Barcelona, July 2007.
6. Benediktsson, J. A., J. A. Palmason, and J. R. Sveinsson. "Classification of Hyperspectral Data From Urban Areas Based on Extended Morphological Profiles". *IEEE Trans. Geoscience and Remote Sensing*, 43(3):480–491, March 2005. URL `http://ieeexplore.ieee.org`.
7. Berglund, E. and J. Sitte. "The parameterless self-organizing map algorithm". *IEEE Transactions on Neural Networks*, 17(2):305–316, March 2006. ISSN 1045-9227.
8. Blazejewsk, Adam and Richard Coggins. "Application of self-organizing maps to clustering of high-frequency financial data". *ACM International Conference Proceeding Series*, 2004.
9. Campbell, James B. *Introduction to Remote Sensing*. The Guilford Press, New York, 4 edition, 2007.
10. Dell'Acqua, F., P. Gamba, A. Ferari, J. A. Palmason, J. A. Benediktsson, and K. Arnason. "Exploiting Spectral and Spatial Information in Hyperspectral Urban Data With High Resolution". *IEEE Trans. Geoscience and Remote Sensing Letters*, 1(4):322–326, Mar 2004.
11. Dell'Acqua, Fabio, Paolo Gamba, and Alessio Ferrari. "Exploiting spectral and spatial information for classifying hyperspectral data in urban areas". *Geoscience and Remote Sensing Symposium*, 1(21-25):464–466, 2003.
12. Donnell, Capt Brian P. *Using Shadows to Detect Targets in Synthetic Aperture Radar*. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, Mar 2009.
13. Elachi, Charles and Jakob Van Zyl. *Introduction to the Physics and Techniques of Remote Sensing*. John Wiley and Sons, Hoboken, New Jersey, 2 edition, 2006.

14. Fauvel, M., J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson. "Spectral and Spatial Classification of Hyperspectral Data Using SVMs and Morphological Profiles". *IEEE Transactions on Geoscience and Remote Sensing*, 46:3804–3814, November 2008. ISSN 0196-2892.

15. Fujimura, K., K. Masuda, and Y. Fukui. "A Consideration on the multi-dimensional topology in Self-Organizing Maps". *Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on*, 825–828. Tottori, December 2006.

16. Giardina, Charles R. and Edward R. Dougherty. *Morphological Methods in Image and Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

17. Hunt, Graham R. and John W. Salisbury. "Visible and Near-Infrared Spectra of Minerals and Rocks: I Silicate Materials". *Lunar and Planetary Branch, Air Force Cambridge Research Laboratories, Hanscom Field, Bedford, Mass.*, (01730), 1970.

18. Jahne, Bernd. *Practical Handbook on Image Processing for Scientific and Technical Applications*. CRC Press, Heidelberg, Germany, 2 edition, 2004.

19. Juang, R.R., P. Burlina, and A. Banerjee. "Level set segmentation of hyperspectral images using joint spectral edge and signature information". *11th Conference on Information Fusion*, 7, 2008.

20. Kaufmann, R. K., L. Zhou, Y. Knyazikhin, V. Shabanov, R. B. Myneni, and C. J. Tucker. "Effect of orbital drift and sensor changes on the time series of AVHRR vegetation index data". *IEEE Transactions on Geoscience and Remote Sensing*, 38(6):2584–2597, November 2000.

21. Keshavarz, A. and H. Ghassemian. "Hyperspectral classification using Markov random field and spatial probability density function". *Telecommunications, 2008. IST 2008. International Symposium on*, 677–681. Tehran, August 2008.

22. Kohonen, T. *Self-Organizing Maps*. Springer-Verlag, Berlin, Germany, 1995.

23. Lai, Hoe Chee, Rong Yang, and Gee Wah Ng. "Enhanced self-organizing map for passive sonar tracking to improve situation awareness". *Information Fusion, 2007 10th International Conference on*, 1–7. Quebec, Que., July 2007. Histograms fail to separate out common trading behavior.

24. Lewis, M. "Mapping arid vegetation associations with HyMap imagery". *Geoscience and Remote Sensing Symposium, 2002. IGARSS '02. 2002 IEEE International*, volume 5, 2805–2807. 2002.

25. Li, Q., N. Mitianoudis, and Stathaki T. "Spatial Kernel K-harmonic Means Clustering for Multi-Spectral Image Segmentation". *Institution of Engineering and Technology Image Processing*, 1(2):156–167, September 2006.

26. Martinez, P., P.L. Aguilar, R.M. Perez, M. Linaje, J.C. Preciado, and A Plaza. "Self-organizing map for hyperspectral image analysis". *Springer-Verlag*, 208–218, 2001.

27. The MathWorks, 3 Apple Hill Drive, Natick, MA 01760-2098. *Matlab 7.4.0.287 (R2007a)*, 1984-2007. URL `http://www.mathworks.com`.

28. The White House. *National Strategy for Combating Terrorism*. In-house, United States Government, Washington DC, September 2006.

29. Meyer, A. W., D. W. Paglieroni, and C. Astaneh. "K-Means Re-Clustering Algorithmic Options with Quantifiable Performance Comparisons". *The International Society for Optical Engineering, Photonics*, January 2003.

30. Mueller, M., K. Segl, and H. Kaufmann. "Discrimination between roofing materials and streets within urban areas based on hyperspectral, shape, and context information". *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*, 196–200. May 2003.

31. Muster, Capt Richard T. *Vehicular Destination Prediction with Persistent Video Surveillance*. Master's thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, Mar 2009.

32. NASA. *National Aeronautics and Space Administration*, 2009. URL `http://www.nasa.gov`.

33. Olsen, R.C. *Remote Sensing from Air and Space*. SPIE – The Society for Optical Engineering, Bellingham, WA, 2007.

34. Penn, B. S. "Using self-organizing maps for anomaly detection in hyperspectral imagery". *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, 3–1531. 2002.

35. Ratnayake, Ranitha L. "Identification of age groups of managed Pine plantation using remote sensing data". *Proceedings of SPIE*, 152–161, April 2006.

36. Ridd, Merrill K. and James D. Hipple (editors). *Remote Sensing of Human Settlements*, volume 5. American Society for Photogrammetry and Remote Sensing, 3 edition, 2006.

37. Rochester Institute of Technology, Rochester, NY. *Digital Imaging and Remote Sensing Image Generation*. URL `http://www.dirsig.org/`.

38. RSI ITT Industries. *ENVI Tutorial: Georeferencing Images Using Input Geometry*, Retrieved Jan 2009. URL `http://www.ittvis.com`.

39. Shackelford, A. K. and C. H. Davis. "Urban road network extraction from high-resolution multispectral data". *Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on*, 142–146. May 2003.

40. Shaw, Gary and Dimitris Manolakis. “Detection Algorithms for Hyperspectral Imaging Applications”. *IEEE Signal Processing Magazine*, (1053-5888), January 2002.

41. Shaw, Gary and Dimitris Manolakis. “Hyperspectral Image Data Analysis”. *IEEE Signal Processing Magazine*, (1053-5888), January 2002.

42. Shaw, Gary and Dimitris Manolakis. “Signal Processing for Hyperspectral Image Exploitation”. *IEEE Signal Processing Magazine*, (1053-5888), January 2002.

43. Smith, Randall B. “Introduction to Hyperspectral Imaging”. 2006. URL `http://www.microimages.com`.

44. SpectIR. *Spectrometer Training Class*. Nevada City, NV, Oct 2008.

45. Sun, Tzu-Lung. “A detection algorithm for road feature extraction using EO-1 hyperspectral images”. *Security Technology, 2003. Proceedings. IEEE 37th Annual 2003 International Carnahan Conference on*, 87–95. October 2003.

46. Sun, Tzu-Lung. “Morphological Segmentation of Urban Structures”. *37th Annual International Carnahan Conference Security Technology*, 87–95, Oct 2003.

47. Tasdemir, K. and E. Merenyi. “Exploiting Data Topology in Visualization and Clustering of Self-Organizing Maps”. *IEEE Transactions on : Accepted for future publication Neural Networks*.

48. Tilton, J. C. “Analysis of Hierarchically Related Image Segmentations”. *Proceedings of IEEE GRSS Workshop on Advances in Techniques*, October 2003.

49. Ubuntu. *Ubuntu 8.10 Intrepid Ibex, 64-bit*, 2008. URL `http://www.ubuntu.com/`.

50. United States Air Force. *Air Force Doctrine Document 1*. In-house, United States Department of Defense, November 2003.

51. Varshney, Pramod K. and Majoj K. Arora. *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data*. Springer, Heidelberg, Germany, 2004.

52. Walter, Volker and Dieter Fritsch. “Automated revision of GIS databases”. Ki-Joune Li, Kia Makki, Niki Pisinou, and Siva Ravada (editors), *Proceedings of the 8th ACM Symposium on Advances in Geographic Information Systems (GIS-00)*, 129–134. ACM Press, N. Y., November 10–11 2000.

53. Wania, A. and C. Weber. “Hyperspectral imagery and urban green observation”. *Urban Remote Sensing Joint Event, 2007*, 1–8. April 2007.

54. Waser, L. T., C. Ginzler, M. Kuechler, E. Baltsavias, and H. Eisenbeiss. “Modeling fractional shrub/tree cover and multi-temporal changes in mire ecosystems using high-resolution digital surface models and CIR aerial images”. *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, 2288–2293. July 2007.

55. Youhao, E., Wang Jihe, Gao Shangyu, Yan Ping, and Yang Zihui. “Monitoring of vegetation changes using multi-temporal NDVI in peripheral regions around minqin oasis, northwest china”. *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, 3448–3451. July 2007.

56. Zhan, Xiwu, R. S. DeFries, S. O. Los, and Zong-Liang Yang. “Application of vegetation continuous fields data in atmosphere-biosphere interaction models”. *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, volume 5, 1948–1950. July 2000.

Vita

Torsten E. Howard graduated from the University of Florida in Gainesville, FL, in 2003 with a B.S. in Electrical Engineering. His first assignment was at Beale AFB, CA, where he worked as the Chief of Systems Integrations for the Distributed Ground Station-II (DGS-II), part of the Air Force's Distributed Common Ground Station (DCGS) weapon system. During this time, he also completed his Master of Business Administration in 2006. He was selected for and enrolled in the prestigious Air Force Institute of Technology in 2007 at Wright-Patterson AFB, OH, where he received his Master of Electrical Engineering in March of 2009. Captain Howard follows this assignment at the Automated Target Recognition Group, Air Force Research Laboratory (AFRL/RYAT), also located at Wright-Patterson AFB, OH.

Permanent address: 2950 Hobson Way
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*
21–03–2009

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
July 2008 — Mar 2009

4. TITLE AND SUBTITLE
Abstracting GIS Layers
from Hyperspectral Imagery

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Howard, Torsten E., Capt, USAF

5d. PROJECT NUMBER
JON# 08-240

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/09-21

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
(MR. DEVERT WICKER)
Air Force Research Labs
BLDG 620, 2241 AVIONICS CIRCLE
WPAFB, OH 45433-7333
(937-904-9871 devert.wicker@wpafb.af.mil)

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RYAT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Modern warfare methods in the urban environment necessitates the use of multiple layers of sensors to manage the battle space. *Hyperspectral imagers* are one possible sensor modality to provide remotely sensed images that can be converted into Geographic Information Systems (GIS) layers. GIS layers abstract knowledge of roads, buildings, and scene content and contain *shape files* that outline and highlight scene features. Creating shape files is a labor-intensive and time-consuming process. The availability of shape files that reflect the current configuration of the *area of interest* significantly enhances Intelligence Preparation of the Battlespace (IPB). The solution presented in this thesis is a novel process to automate the creation of shape files by exploiting the spectral-spatial relationship of a *hyperspectral image* cube. It is assumed that "a-priori" endmember spectra, a spectral database, or specific scene knowledge is *not* available. The *topological neighborhood* of a Self Organizing Map (SOM) is segmented and used as a spectral filter to produce six initial *object maps* that are spatially processed with *logical* and *morphological operations*. A novel road finding algorithm connects road segments under significantly tree-occluded roadways into a contiguous road network. The manual abstraction of GIS shape files is improved into a semi-automated process. The resulting shape files are not susceptible to deviation from orthorectified imagery as they are produced directly from the hyperspectral imagery. The results are eight separate high-quality GIS layers (Vegetation, Non-Tree Vegetation, Trees, Fields, Buildings, Major Buildings, Roadways, and Parking Areas) that follow the terrain of the hyperspectral image and are separately and automatically labeled. Spatial processing improves layer accuracy from 85% to 94%. Significant layer accuracies include the "road network" at 93%, "buildings"at 97%, and "major buildings" at 98%. .

15. SUBJECT TERMS
GIS, Hyperspectral, Scene Segmentation

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
U	U	U	U	154

19a. NAME OF RESPONSIBLE PERSON
Major Michael Mendenhall

19b. TELEPHONE NUMBER *(include area code)*
(937) 255-3636, ext 4614 email: michael.mendenhall afit.edu