WAVELET-DOMAIN HYPERSPECTRAL SOIL TEXTURE CLASSIFICATION

By

Xudong Zhang

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Master of Science
in Electrical Engineering
in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

May 2004
Copyright by

Xudong Zhang

2004
WAVELET-DOMAIN HYPERSPECTRAL SOIL TEXTURE CLASSIFICATION

By

Xudong Zhang

Approved:

Nicolas H. Younan
Professor of Electrical and Computer Engineering
(Major Professor and Graduate Coordinator)

Lori M. Bruce
Associate Professor of Electrical and Computer Engineering
(Committee Member)

Roger L. King
William L. Giles Distinguished Professor of Electrical and Computer Engineering
(Committee Member)

A. Wayne Bennett
Dean of the College of Engineering
This thesis presents an automatic soil texture classification system using hyperspectral soil signals and wavelet-based statistical models. Previous soil texture classification systems are closely related to texture classification methods, which use images for training and testing. Although using image-based algorithms is a straightforward way to conduct soil texture classification, our research shows that it does not provide reliable and consistent results. Rather, we develop a novel system using hyperspectral soil textures, better known as hyperspectral soil signals, which provide rich information and intrinsic properties about soil textures. Hyperspectral soil textures, in their very nature, are nonstationary and time-varying. Therefore, the wavelet transform, which is proven to be successful in such applications, is incorporated. In this study, we incorporate two wavelet-domain statistical models, namely, the maximum likelihood (ML) and the hidden Markov model (HMM) for the classification task. Experimental results show that this method is reliable.
and robust. It is also more effective and efficient in terms of practical implementation than the traditional image-based methods.
DEDICATION

To my family, including my wife, parents and sister, for their caring and love.
ACKNOWLEDGMENTS

First of all, I would like to thank my advisor, Dr. Nicolas H. Younan, for his continuous encouragement and support and efforts in reviewing this thesis. I would also like to thank Dr. Roger L. King and Dr. Lori M. Bruce for serving as my committee members, and I enjoyed being their students.

I have been enjoying conducting research at the Engineering Research Center (ERC), where students have access to the state-of-the-art computing facilities. I also thank the research staff at the ERC for their kindly help. I would like to thank Dr. Charles G. O’Hara for his helpful discussions on soil textures. I would also like to thank Mr. Louis Wasson, for his continuous and unreserved help in hyperspectral data collection. Special thanks go to the National Aeronautics and Space Administration (NASA), for providing financial support to this research. My time at the ERC would not be so comfortable and enjoyable without the support from NASA.

Last but not least, I would like to thank Dr. Ed Allen in the Department of Computer Science, for providing the \LaTeX template for typing this thesis so that it becomes enjoyable and fun.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEDICATION</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>vii</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>I.  INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction to Soil Texture</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Literature Review</td>
<td>4</td>
</tr>
<tr>
<td>1.3 Motivation</td>
<td>7</td>
</tr>
<tr>
<td>1.4 Thesis Organization</td>
<td>8</td>
</tr>
<tr>
<td>II. HYPERSPECTRAL SOIL TEXTURE &amp; WAVELET TRANSFORM</td>
<td>10</td>
</tr>
<tr>
<td>2.1 Hyperspectral Soil Texture</td>
<td>10</td>
</tr>
<tr>
<td>2.2 Linear Mixture Model</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Wavelet Transform</td>
<td>17</td>
</tr>
<tr>
<td>III. MAXIMUM LIKELIHOOD MODEL</td>
<td>22</td>
</tr>
<tr>
<td>3.1 Feature Extraction and Reduction</td>
<td>22</td>
</tr>
<tr>
<td>3.2 Maximum Likelihood Classification</td>
<td>25</td>
</tr>
<tr>
<td>IV. HIDDEN MARKOV MODELS</td>
<td>27</td>
</tr>
<tr>
<td>4.1 What are Hidden Markov Models</td>
<td>27</td>
</tr>
<tr>
<td>4.1.1 Definition</td>
<td>28</td>
</tr>
<tr>
<td>4.1.2 Three Basic Problems</td>
<td>30</td>
</tr>
<tr>
<td>4.2 Wavelet-Domain Hidden Markov Models</td>
<td>31</td>
</tr>
<tr>
<td>4.2.1 Gaussian Mixture Model</td>
<td>32</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>40</td>
</tr>
<tr>
<td>5.2</td>
<td>40</td>
</tr>
<tr>
<td>5.3</td>
<td>46</td>
</tr>
<tr>
<td>5.4</td>
<td>47</td>
</tr>
<tr>
<td>5.5</td>
<td>48</td>
</tr>
<tr>
<td>5.6</td>
<td>48</td>
</tr>
<tr>
<td>5.7</td>
<td>52</td>
</tr>
<tr>
<td>5.8</td>
<td>52</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>USDA Soil Texture Triangle. 3</td>
</tr>
<tr>
<td>1.2</td>
<td>A sand-textured soil sample. 4</td>
</tr>
<tr>
<td>1.3</td>
<td>A silt-textured soil sample. 5</td>
</tr>
<tr>
<td>1.4</td>
<td>A clay-textured soil sample. 5</td>
</tr>
<tr>
<td>2.1</td>
<td>Hyperspectral signals of sand-textured soil samples. 12</td>
</tr>
<tr>
<td>2.2</td>
<td>Hyperspectral signals of silt-textured soil samples. 12</td>
</tr>
<tr>
<td>2.3</td>
<td>Hyperspectral signals of clay-textured soil samples. 13</td>
</tr>
<tr>
<td>2.4</td>
<td>A mixed hyperspectral signal of a sand-textured soil sample. 16</td>
</tr>
<tr>
<td>2.5</td>
<td>A mixed hyperspectral signal of a silt-textured soil sample. 16</td>
</tr>
<tr>
<td>2.6</td>
<td>A mixed hyperspectral signal of a clay-textured soil sample. 17</td>
</tr>
<tr>
<td>2.7</td>
<td>Filter bank implementation of the dyadic tree algorithm. 20</td>
</tr>
<tr>
<td>2.8</td>
<td>The original signal of silt $f(t)$. 20</td>
</tr>
<tr>
<td>2.9</td>
<td>The approximation of $f(t)$ in decomposition level 7. 21</td>
</tr>
<tr>
<td>2.10</td>
<td>The detail of $f(t)$ in level 7. 21</td>
</tr>
<tr>
<td>4.1</td>
<td>The tree-structure of the wavelet coefficients in the time-frequency plane. 35</td>
</tr>
<tr>
<td>5.1</td>
<td>Hyperspectral Soil Textures. 42</td>
</tr>
<tr>
<td>5.2</td>
<td>Hyperspectral Soil Textures. 43</td>
</tr>
<tr>
<td>FIGURE</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>5.3</td>
<td>Hyperspectral Soil Textures.</td>
</tr>
<tr>
<td>5.4</td>
<td>The performance of the HMM classifier.</td>
</tr>
<tr>
<td>5.5</td>
<td>The performance of the ML classifier.</td>
</tr>
</tbody>
</table>
CHAPTER I

INTRODUCTION

Sand, silt, and clay are three major components found in a typical soil sample. Soil texture, which is one of the most important properties of soil, is determined by the three endmembers. That is, soil appears to have different textures with different proportions of the three endmembers. Since soil is the major cover of the earth and essential for agriculture, environment protection, and monitoring, etc., the analysis of soil texture plays an important role in applications where soil texture affects soil properties such as soil structure, aeration, water movement, etc. Thus, soil texture classification arises as an important task for such applications.

The primary goal of this thesis is to facilitate the process of automatic classification of soil textures in an efficient and effective manner based on the wavelet-domain classification algorithms, and it is presented with a hope that these methods can be put into practice in future applications.

1.1 Introduction to Soil Texture

Soil texture refers to the individual particle size in a given soil sample. Therefore, the definition of texture here is quite different from its definition in an image where texture
is usually defined as the local statistical property of a region with varying or constant patterns in a two-dimensional space. Soil texture is determined by the proportions of the three endmembers, e.g., sand, silt, and clay. Different types of soils have different types of textures. Sand feels gritty and has the largest particle size among the three endmembers, thus the individual particles can be clearly recognized by the naked eyes. A soil sample with a predominant portion of sand can be classified as sand-textured soil. The particle size of silt is much smaller than that of sand and can only be recognized clearly with the aid of a microscope. A soil sample that contains a large amount of silt is classified as silt-textured soil. The particle of clay has the smallest size among the three classes. Hence, the clay-textured soil is rich in clay and fine in texture. A more detailed definition of the soil texture can be found in Figure 1.1, which is the soil texture triangle defined by the United States Department of Agriculture (USDA).

The soil texture triangle provides a clear and easy way to determine the soil texture based on the proportions of the three endmembers. The triangle is equilateral, i.e., all angles are 60 degrees. Each side of the triangle is divided into 100 segments. A line extending from a side represents the percentage of an endmember. To determine a soil texture, we need to perform the following steps:

- Extend a horizontal line from the side labeled with “Clay”.
- Extend a line downward at 60 degrees from the side labeled with “Silt”.
- Extend a line upward at 120 degrees from the side labeled with “Sand”.

Figure 1.1: USDA Soil Texture Triangle.
Accordingly the type of the texture is determined by the location where the intersect of the three lines falls. For instance, given a soil sample that contains 10% of sand, 45% of silt, and 45% of clay, it is clear that this soil texture is silty clay.

1.2 Literature Review

As a mixture of sand, silt, and clay, a soil sample may have different soil textures with different proportions of the three endmembers. Three typical soil samples are shown in Figure 1.2 to Figure 1.4. Since the most straightforward way to classify a soil texture is to use two-dimensional images of the soil samples and assign a specific soil sample to a class, texture classification arises as an important step for soil texture classification.

Figure 1.2: A sand-textured soil sample.
Figure 1.3: A silt-textured soil sample.

Figure 1.4: A clay-textured soil sample.
Texture is one of the important properties of an image that is usually defined as the local statistical property of a region with varying or constant patterns. Texture classification is an active yet difficult topic in image processing, although there have been a lot of research efforts on it over the past several decades. The methods applied to texture classification can be roughly divided into three categories. These categories are statistical, structural, and multiresolution methods. Among the former two, there are several algorithms that have resulted in promising results. In the 1970’s, Haralick et al. proposed a method based on second-order texture statistics, the co-occurrence matrix [1]. The typical features are energy, entropy, correlation, local homogeneity, and inertia. Later in the 1980’s, methods based on Gaussian Markov random fields (GMRF) and Gibbs distribution models were proposed in [2]. In these models, the gray levels between nearest neighboring pixels are characterized by a certain statistical relationship. However, all the above methods have one weakness in common: the features are not complete and only resulted from the analysis of a single scale. The multiresolution analysis (MRA) methods, which have shown to resemble the human visual system (HVS) behavior, have proven to give out better results [3, 4, 5, 6].

So far, the image-based soil texture classification algorithms have adopted the MRA methods. Sun used Gabor wavelets to do feature extraction and maximum likelihood (ML) classifier to do classification [7]. Zhang et al. used Daubechies orthonormal wavelets to do feature extraction and the ML classifier to do the classification [8]. Both of the algorithms yielded good classification accuracies.
Although image-based soil texture classification algorithms yield good classification results, they have some drawbacks that hinder their applications in real-world problems. First, the image-based algorithms usually have a large computational load. The soil images under study are divided into small blocks. Feature extraction and classification are performed block by block. Therefore, it involves a large amount of computation. Second, texture classification is still considered a difficult problem given the uncertainty of the structure of textures. Last but not least, the soil samples need to be pre-processed with special lab facilities, and soil images need to be taken with special-function cameras, which are not easily accessed by most of the users. Therefore, a more efficient, effective and practical method needs to be developed so that the task of soil texture classification can be easily accomplished.

1.3 Motivation

Combined with the above issues, a soil texture classification system using hyperspectral soil textures, better known as hyperspectral soil signals, is proposed in this thesis. This idea is motivated by the fact that different soil textures are determined by different proportions of the three endmembers, not by the structure of the soil surface. Thus, two soil samples with different surface structures may be in the same class. However, if they are classified using the image-based methods, they are probably ended up in two different soil textures. These kind of problems can be avoided by using the wavelet-based methods proposed in this thesis. Furthermore, hyperspectral signals provide more insightful infor-
mation of the objects. As a result, the classification task can be concentrated on the true properties of soil textures.

Soil texture classification is an interdisciplinary and challenging task. The primary contribution of this thesis is the application of wavelet-domain statistical signal processing methods, combined with the hyperspectral soil textures to tackle this task. The algorithms presented in this thesis are mature and proved to be successful, however, it is the first time to apply them to soil texture classification. We show that, soil texture classification can be successfully conducted using the proposed methods. As a result, the methods can be put into practice in future applications.

1.4 Thesis Organization

The remainder of the thesis is organized as follows.

Chapter II: The primary purpose of Chapter II is to provide an introduction to the hyperspectral soil texture and its wavelet transform. After a brief introduction to hyperspectral soil texture, the idea of using a linear mixture model (LMM) to simulate different hyperspectral soil textures is presented. Wavelet transform, in particular, the discrete wavelet transform (DWT), is briefly reviewed. The DWT is a mathematical tool that is used for MRA methods.

Chapter III: Chapter III covers the theoretical background of ML classifiers. It begins with feature extraction, where the energy of the wavelet coefficients in each scale is incorporated. Linear discriminant analysis, in particular, the Fisher’s linear discriminant
analysis (FLDA) is applied to the feature vectors as a tool for feature optimization and reduction.

**Chapter IV:** We present the wavelet-domain hidden Markov models (HMM’s) for the classification task in this chapter. It is primarily focused on the mathematical background of HMM’s. The general HMM’s are reviewed at first, then the wavelet-domain HMM’s is presented. In particular, the wavelet-domain Gaussian mixture model and hidden Markov tree model (HMTM) are described based on the wavelet coefficients of hyperspectral signals.

**Chapter V:** Experimental results and analysis are presented in this chapter. The input hyperspectral signatures are divided into two groups. One group contains three hyperspectral soil textures, which are sand dominant, silt dominant, and clay dominant. The other group contains twelve hyperspectral soil textures in accordance with the USDA soil texture triangle. The classification results of the ML and HMM classifiers are presented. The performance of the classifiers are analyzed using different mother wavelets and the optimal parameters are determined.

**Chapter VI:** Finally, a conclusion is presented. We conclude that the wavelet-domain statistical soil texture classification using the ML classifier and HMM is efficient and effective, and provides a method for future practical application.
CHAPTER II

HYPERSPECTRAL SOIL TEXTURE & WAVELET TRANSFORM

The primary goal of this chapter is to introduce hyperspectral soil textures and their wavelet transform. We first review hyperspectral signals, then introduce hyperspectral soil textures. Examples of the hyperspectral sand, silt, and clay textures are provided. A linear mixture model (LMM) is followed to simulate different soil textures in accordance with the USDA soil texture triangle. This chapter also provides a mathematical background of wavelets and presents the wavelet transform of the hyperspectral soil textures.

2.1 Hyperspectral Soil Texture

Target detection and classification have been the primary goals of remote sensing applications. Due to the limitations of spatial resolution, spectral information is playing an important role in these applications. In recent years, hyperspectral remotely sensed data have been applied to a variety of remote sensing applications. The hyperspectral data consists of electromagnetic reflectance in a series of contiguous spectral bands. The data is collected by hundreds of sensors, with each sensor being sensitive only to a narrow range of wavelengths. Since different materials have their unique electromagnetic radiance
characteristics, hyperspectral data are ideal candidate for target detection and classification on ground covers.

Hyperspectral soil textures, better known as hyperspectral soil signals, are the hyperspectral electromagnetic reflectance of soil samples. The idea of using hyperspectral soil signals instead of soil images arises from the hyperspectral signals’ providing more insightful and reliable information that represent the true nature of the soil samples. Since soil texture is determined by different proportions of the three major constituents, different soil textures have different spectral electromagnetic reflectance. Therefore, the hyperspectral soil texture provides a better perspective for the classification task. On the contrary, two-dimensional soil images may not provide reliable information of soil textures, primarily because the two-dimensional soil images only represent the surface structure of the soil samples. Moreover, it is possible that the same texture may have different surface structure. As a result, the classification results may be inaccurate. Therefore, hyperspectral soil textures are a better choice over two-dimensional image textures.

As an example, the hyperspectral signals of three different soil samples, which are sand-textured, silt-textured, and clay-textured, are shown in Figure 2.1 to Figure 2.3. These hyperspectral soil textures are taken by hand-held spectroradiometer from Analytical Spectral Devices (ASD), Inc. The electromagnetic reflectance taken by the instrument has a wavelength range from 350nm to 2500nm, which contains rich information of the soil samples.
Figure 2.1: Hyperspectral signals of sand-textured soil samples.

Figure 2.2: Hyperspectral signals of silt-textured soil samples.
Figure 2.3: Hyperspectral signals of clay-textured soil samples.

2.2 Linear Mixture Model

A linear mixture model (LMM) is usually used as an analysis tool in linear unmixing problems in many remote sensing applications [9, 10]. However in this thesis, the LMM is used as a synthesis tool to simulate the soil textures in accordance with the USDA soil texture triangle. The reason we use the LMM is twofold. First, the LMM is a general mathematical tool for modeling the ground-cover radiance with mixed materials or objects. Although it may be less optimal than the non-linear mixture model (NLMM), it has been widely applied in many applications such as hyperspectral unmixing and endmember estimation, primarily because of its simplicity in computation and its proven successes. The small tradeoff between optimality and computational efficiency is much better than the troubles caused by the uncertainties and large computational load of the NLMM. Sec-
ond, the LMM is a good model for soil texture, due to the fact that soil texture can be represented by linear mixture of three different endmembers.

An LMM assumes that the electromagnetic (EM) reflectance of a mixed material is the linear combination of that of each endmember. In general, the LMM is described as [9]

\[ y = Ax + e, \]  

(2.1)

where

\[ y = [y_1, y_2, \ldots, y_N]^T \]  

(2.2)

represents the electromagnetic reflectance of the mixed material; \( N \) is the number of the spectral bands;

\[ x = [x_1, x_2, \ldots, x_M]^T \]  

(2.3)

represents the fractions or abundances of the endmembers; \( M \) is the number of the endmembers;

\[ e = [e_1, e_2, \ldots, e_N]^T \]  

(2.4)

represents the random errors caused by various noises; and

\[ A = [a_1, a_2, \ldots, a_N] \]  

(2.5)

is a matrix representing the spectra of the pure endmembers. Each column of \( A \) represents an endmember spectrum as follows

\[ a_i = [a_{i1}, a_{i2}, \ldots, a_{iN}]^T, \]  

(2.6)
where $i = 1, 2, \ldots, M$. Clearly the LMM has a simple mathematical expression that is easy to implement and understand, while on the contrary, the NLMM does not have a general mathematical expression. Although experiments have shown that the NLMM slightly outperforms the LMM, it usually involves a large amount of calculation. Moreover, the LMM is easily combined with other mature mathematical methods in linear unmixing estimation, which is not the case for the NLMM.

Although the LMM is usually used for linear unmixing problems as an analysis tool, it is used in our research as a synthesis tool to mix the spectra of the endmembers. For simplicity, the LMM in our system is described as

$$y = \alpha x_1 + \beta x_2 + \gamma x_3,$$

(2.7)

where $y$ is the mixed spectrum, $x_i (i = 1, 2, 3)$ is the spectrum of sand, silt, and clay. The coefficients $\alpha, \beta, \text{and} \gamma$ are the abundances of each of the endmember satisfying the following constraint

$$\alpha + \beta + \gamma = 1.$$  

(2.8)

This simple synthesis model is in accordance with the USDA soil texture triangle, where a specific soil texture is determined by the mixture of sand, silt, and clay with different proportions. For example, in a typical clay dominant soil sample, the proportions of the three endmembers are 20%, 10%, and 70%, respectively. In a typical sand dominant soil sample, they are 70%, 10%, and 20%, and in a typical silt dominant soil sample, they are 20%, 70%, and 10%, respectively. To illustrate the mixtures in the hyperspectral
domain, three hyperspectral signals of three different soil textures are shown in Figure 2.4 to Figure 2.6.

Figure 2.4: A mixed hyperspectral signal of a sand-textured soil sample.

Figure 2.5: A mixed hyperspectral signal of a silt-textured soil sample.
Figure 2.6: A mixed hyperspectral signal of a clay-textured soil sample.

2.3 Wavelet Transform

Wavelets are both old and new mathematical tools. The primitive idea of wavelets existed at the beginning of the twentieth century, when Haar first published his paper on the simplest scaling and wavelet functions. Not until the 1980s did wavelets began their roles in signal processing, especially in the analysis of time-varying and nonstationary signals [11, 12]. Over the past twenty years, wavelets have evolved to a mature and systematic mathematical tool for many fields. Wavelets have found their applications not only in signal processing, but also in computer vision, statistics, computer graphics, etc.

In general, when we say wavelet transform, it goes to wavelet analysis. Wavelet analysis decomposes a signal into a set of sub-signals by wavelet functions generated by dilation
and translation of a basis function known as a mother wavelet $\psi(t)$. The wavelet functions at scales $j$ and translation $k$ is obtained via [13]

$$
\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k),
$$

(2.9)

where $j$ is the decomposition scale and $k$ is the translation along the time axis. The wavelet functions must satisfy the multiresolution analysis (MRA) constraint as

$$
\psi(t) = \sum_n g[n] \sqrt{2} \phi(2t - n),
$$

(2.10)

where $\phi(t)$ is the scaling function, and $g[n]$ is a set of coefficients. The scaling function $\phi(t)$ must also satisfy the MRA condition via

$$
\phi(t) = \sum_n h[n] \sqrt{2} \phi(2t - n),
$$

(2.11)

where $h[n]$ is a set of coefficients different than $g[n]$. $h[n]$ and $g[n]$ are related to each other by

$$
g[n] = (-1)^n h[1 - n].
$$

(2.12)

With the above conditions, any function $f(t) \in L^2(\mathbb{R})$ could be written as

$$
f(t) = \sum_{k=-\infty}^{\infty} c(k) \phi_k(t) + \sum_{j=0}^{\infty} \sum_{k=-\infty}^{\infty} d(j,k) \psi_{j,k}(t),
$$

(2.13)

where $c(k)$ and $d(j,k)$ are called scaling coefficients and wavelet coefficients respectively.

Like Fourier transform, the wavelet transform also has its continuous and discrete counterparts. In this thesis, the discrete wavelet transform (DWT) is incorporated. In the DWT, (2.13) becomes

$$
f(t) = \sum_k c_{J_0}[k] \phi_{J_0,k}(t) + \sum_{j=J_0}^{J_1-1} \sum_k d_{j}[k] \psi_{j,k}[t],
$$

(2.14)
where the choice of \( J_0 \) sets the coarsest scale of the expansion and \( J_1 \) is the maximum decomposition level to a given discrete signal. The coefficients \( c_{J_0}[k] \) and \( d_{J_1}[k] \) in (2.14) are called the discrete wavelet transform of \( f(t) \). If the wavelet system is orthonormal, that is, if the scaling and wavelet functions are orthonormal to themselves, the scaling and wavelet coefficients can be obtained via

\[
c_j[k] = \langle \phi_{j,k}, f \rangle \tag{2.15}
\]

\[
d_j[k] = \langle \psi_{j,k}, f \rangle. \tag{2.16}
\]

In practical applications, the DWT is usually implemented by a fast algorithm called a dyadic filter tree. In the dyadic filter tree algorithm, the signal \( f(t) \) is convolved with a high-pass and low-pass filters simultaneously, and the filtered signals are down sampled by a factor of two. The outputs of the low-pass filter is called wavelet approximation coefficients, and the outputs of the high-pass filters are called wavelet detail coefficients. The decomposition process can be recursively applied to the output of the low-pass filter until it reaches the maximum level of \( J_1 \), where \( J_1 = \log_2 N \) and \( N \) is the length of the signal. Figure 2.7 illustrates the process of the wavelet decomposition in one node of the dyadic tree.

As an example of the wavelet decomposition, the hyperspectral signal of silt and its wavelet decomposition using the Haar wavelet in the 7th level are shown in Figure 2.8 to Figure 2.10. The silt signal consists of 2048 bands of electromagnetic reflectance. The figures clearly show that the approximation looks much alike the original signal, whereas the detail part consists of the high frequency components of the original signal. The reason
is that the approximation and detail are outputs of convolution of the original signal with the low and high-pass Haar wavelet filters.

Figure 2.7: Filter bank implementation of the dyadic tree algorithm.

Figure 2.8: The original signal of silt $f(t)$.
Figure 2.9: The approximation of $f(t)$ in decomposition level 7.

Figure 2.10: The detail of $f(t)$ in level 7.
CHAPTER III

MAXIMUM LIKELIHOOD MODEL

The Maximum Likelihood (ML) classifier is one of the most widely used classifiers in the remote sensing community due to its good performance and simplicity in implementation. This chapter presents the theoretical background of a classification method with the ML classifier. It starts with feature extraction, where the wavelet energy feature is incorporated. Next is the feature reduction and optimization steps, where the Fisher’s linear discriminant analysis (FLDA) method is applied to the wavelet energy features. The ML classification algorithm is then described in detail for the hyperspectral soil texture classification task.

3.1 Feature Extraction and Reduction

The wavelet energy feature is widely used in many remote sensing applications [14, 15]. In the wavelet energy feature vector, each item is obtained by calculating the root mean square (RMS) value of the wavelet and scaling coefficients. In other words, the feature vector is expressed as

\[ \mathbf{F} = [F_0, F_1, F_2, \ldots, F_J]^T, \]

(3.1)
where \( J_1 \) is the maximum wavelet decomposition level, and \( T \) is the transpose operation to a vector. The \( j^{th} \) element of the feature vector is obtained via

\[
F_j = \sqrt{\frac{1}{N_j} \sum_{k=1}^{N_j} |d_{j,k}|^2}, \quad j \in \{1, 2, \ldots, J_1\}, \quad k \in \{1, 2, \ldots, N_j\},
\]

(3.2)

where \( d_{j,k} \) are the wavelet coefficients in level \( j \) and \( N_j \) is the number of coefficients in this level. The energy feature of the scaling coefficients is obtained via

\[
F_0 = \sqrt{\frac{1}{N_0} \sum_{k=1}^{N_0} |c_{0,k}|^2}, \quad k \in \{1, 2, \ldots, N_0\},
\]

(3.3)

where \( c_{0,k} \) are the scaling coefficients in the coarsest scale and \( N_0 \) is the number of the coefficients in this scale.

Feature optimization is an important step in classification algorithms because it reduces the dimensionality of the feature vectors and provides a good class separation. The Fisher’s Linear discriminant analysis (FLDA) is a frequently used technique for optimization. The FLDA chooses a weight vector such that the projections of patterns belonging to different classes are well separated. In other words, the FLDA tries to increase the between-class variance and simultaneously reduce the within-class variance. The criterion function for the FLDA is as follows [16]

\[
J(w) = \frac{w^T S_b w}{w^T S_w w},
\]

(3.4)
where $\mathbf{w}$ is the weight vector, $\mathbf{S}_b$ is the between-class scatter matrix, $\mathbf{S}_w$ is the within-class scatter matrix, and $T$ is the transpose operation to a vector. In a multiclass case, the between-class scatter matrix $\mathbf{S}_b$ and the within-class scatter matrix $\mathbf{S}_w$ are as follows [17]

$$
\mathbf{S}_b = \sum_{i=1}^{C} (\hat{\mathbf{m}}_i - \mathbf{m})(\hat{\mathbf{m}}_i - \mathbf{m})^T,
$$

(3.5)

and

$$
\mathbf{S}_w = \sum_{i=1}^{C} \hat{\Sigma}_i,
$$

(3.6)

where $\hat{\mathbf{m}}_i$ and $\hat{\Sigma}_i$ are the sample mean and covariance matrix of each class, $\mathbf{m}$ is the mean of all classes, $C$ is the number of classes, and $n_i$ is the number of samples of each class.

The weight vector $\mathbf{w}$ that meets the goal of the FLDA is the one that maximizes (3.4), at the same time, it should satisfy the following condition

$$
\mathbf{S}_b \mathbf{w} = \lambda \mathbf{S}_w \mathbf{w},
$$

(3.7)

where $\lambda$ is the eigenvalue of (3.7). Thus, $\mathbf{w}$ can be obtained via the eigen decomposition of the equation known as the generalized symmetric eigenvector equation as follows

$$
\mathbf{S}_w^{-1} \mathbf{S}_b \mathbf{w} = \lambda \mathbf{w}.
$$

(3.8)

After we obtain the weight matrix $\mathbf{w}$, the feature reduction and optimization can be implemented as

$$
\mathbf{F}_r = \mathbf{w}^T \mathbf{F},
$$

(3.9)

where $\mathbf{F}_r$ is the optimized feature vector, $\mathbf{F}$ is the input feature vector, and $T$ is the transpose operation to a vector.
3.2 Maximum Likelihood Classification

The ML classifier is one of the most popular classification, detection, and segmentation methods used in remote sensing applications. The idea of the ML classification method is based on Bayesian decision theory [17, 18], which finds the *a posteriori* probability of a random variable using its priori probability and likelihood function. A decision is made based upon the value of the *a posteriori*, which is expressed as follows

\[
p(\theta_j|x) = \frac{p(x|\theta_j)p(\theta_j)}{p(x)},
\]

where \( p(\theta_j|x) \) is the *a posteriori*, that is, the probability of the observed data being in class \( j \) given the value of the observation; \( p(x|\theta_j) \) is the likelihood of \( \theta_j \) with respect to \( x \), that is, given the parameter vector \( \theta_j \) of class \( j \), the probability of the observation \( x \) being in this class; \( p(\theta_j) \) is the priori probability of class \( j \); and \( p(x) \) is the probability of \( x \), which is also called *evidence* in the pattern recognition literature. Practical applications assume that the priori probability and evidence of \( x \) are equal for each class. Hence, we only need to consider the likelihood function in the Bayesian formula (3.10). Therefore, the classification task is similar to comparing the likelihood function of each class \( j \) with respect to \( x \), and assign \( x \) to the class which has the maximum likelihood.

Moreover, the ML classifier assumes a multivariate Gaussian distribution for each labeled class and unknown observation. That is, given the parameter vector \( \theta_j \) of class \( j \), we have
\[
p(x|\theta_j) = \frac{1}{(2\pi)^{n/2} |\Sigma_j|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu_j)^T \Sigma_k^{-1}(x - \mu_j)\right],
\]

(3.11)

where \(\Sigma_j\) and \(\mu_j\) are the covariance matrix and mean vector of the corresponding Gaussian function, respectively, and \(n\) is the number of entries in the feature vector. The parameter vector \(\theta_j\) for a class \(j\) is obtained using training data via the ML parameter estimation. With the multivariate Gaussian assumption, the ML parameter estimation for \(\Sigma\) and \(\mu\) is as follows

\[
\hat{\mu} = \frac{1}{n} \sum_{k=1}^{n} x_k,
\]

(3.12)

and

\[
\hat{\Sigma} = \frac{1}{n} \sum_{k=1}^{n} (x_k - \hat{\mu})(x_k - \hat{\mu})^T,
\]

(3.13)

where \(n\) is the number of training data and \(T\) is the transpose operation to a vector. In our case, \(x_k\) is the wavelet energy feature vector.

In summary, performing ML classification requires the following steps:

- **ML training phase**—The training phase is to build ML models using the training data, which consist of ground truth data, via the ML parameter estimation. As a result, the parameter vector \(\theta_j\) is obtained, where \(j \in \{1, 2, \ldots, N\}\) and \(N\) is the number of class.

- **Likelihood determination phase**—Given an observation \(x\), the likelihood function \(p(x|\theta_j)\) is calculated. The observation \(x\) is assigned to a class that has the maximum likelihood.
CHAPTER IV

HIDDEN MARKOV MODELS

Statistical models play an essential role in classification, segmentation, and denoising tasks. We presented the maximum likelihood model in Chapter III. In this chapter, we introduce other models, namely, the Hidden Markov Modes (HMM’s). We first introduce the background of the general HMM’s, then present the wavelet-domain HMM’s. In the wavelet-domain HMM’s, we first present the Gaussian Mixture to model the individual wavelet coefficient, then present the hidden Markov tree model (HMTM) to model the Markov property over the entire wavelet tree in the time-frequency plane. A significant part of this chapter is focused on the mathematical background of the HMM’s.

4.1 What are Hidden Markov Models

Hidden Markov Models are models that describe the statistical behaviors of an event and its underlying associated state probabilities. The basic ideas and theories of HMM’s were first introduced by Baum et al. in the 1960s [19, 20]. However, it is not until the late 1980s that the HMM’s began their applications in the signal processing areas. The HMM’s are most widely known for their applications in speech processing applications, in particular, speech recognition [21]. In recent years, the applications of HMM’s in signal
and image processing for the tasks of classification and segmentation were proposed by Li et al. [22, 23, 24]. Furthermore, the HMM’s were extended to the wavelet domain by Crouse et al. [25]. Based on the Markov property of the tree structure of the wavelet coefficients, the HMM’s provide a robust and efficient model to capture the intra and inter-scale correlation of the wavelet coefficients. Fan et al. extended the wavelet-domain HMM’s to texture classification, segmentation, and synthesis [26, 27]. These wavelet-based HMM algorithms, combined with the efficient parameters estimation algorithms, make this model a promising tool for statistical signal and image processing applications.

Motivated by the ideas of the wavelet-domain HMM’s, we incorporate this model in our hyperspectral soil texture classification task. We present the wavelet-domain HMM’s with a hope that they can be applied to other remote sensing applications, such as signal/image classification, segmentation, target detection, etc., in our future research works.

4.1.1 Definition

HMM’s are models that are used to describe the stochastic behavior of an event that contains two random processes, both of which are associated with each other by the underlying statistical functions. One of the random processes is called observation, which is visible; the other random process is called state, which is hidden and contains a finite number of states. The outcome, or the observation, is dictated by its associated hidden state via a probability distribution. Transitions among different hidden states are dictated
by a finite set of probabilities, known as transition probabilities. The following elements are necessary to make the concept and idea of the HMM’s clear to understand [21].

- The number of the states in the model, $N$. There is not a fixed value for the hidden states since they are “hidden”. However, for a specific application, we can associate the significant parameters with the hidden states as long as this association can approximate the model in an accurate manner. The individual state is denoted by $S = \{ S_1, S_2, \ldots, S_N \}$, and the state at time $t Q_t$.

- The number of the outcome or observation symbols in each state, $M$. The individual symbol is denoted by $V = \{ v_1, v_2, \ldots, v_M \}$.

- A finite set of the state transition probabilities $A = \{ a_{i,j} \}$, where

$$a_{i,j} = p\{ Q_{t+1} = S_j | Q_t = S_i \}. \quad (4.1)$$

The transition probabilities should satisfy the stochastic constraints, that is

$$a_{i,j} \geq 0, \quad 1 \leq i, j \leq N.$$

and

$$\sum_{j=1}^{N} a_{i,j} = 1, \quad 1 \leq i \leq N.$$

The transition probabilities’ independence of the time index is assumed, that is, it is a stationary Markov chain, i. e.,

$$p\{ Q_{t_1+1} = S_j | Q_{t_1} = S_i \} = p\{ Q_{t_2+1} = S_j | Q_{t_2} = S_i \}. \quad (4.2)$$

- The probability distribution of the symbols in each state $j$, $B = \{ b_j(k) \}$, where

$$b_j(k) = p\{ O_t = v_k | q_t = S_j \}, \quad (4.3)$$

with $O_t$ being the observation at the instant of $t$. $b_j(k)$ must satisfy the stochastic constraints as follows

$$b_j(k) \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$

and

$$\sum_{k=1}^{M} b_j(k) = 1, \quad 1 \leq j \leq N.$$

- The initial state distribution $\pi = \{ \pi_i \}$, where

$$\pi_i = p\{ Q_1 = S_i \}, \quad 1 \leq i \leq N. \quad (4.4)$$
As can be seen from the above elements, we can use a compact and convenient expression to denote an HMM, that is

$$\lambda = (A, B, \pi). \quad (4.5)$$

### 4.1.2 Three Basic Problems

Given an HMM, three problems of interest are encountered in most real-world applications; Likelihood evaluation, state sequence estimation, and HMM training. A brief discussion of each problem follows.

**Likelihood Evaluation**: This problem arises when we need to determine the probability of the observation $O = \{O_1, O_2, \ldots, O_T\}$, given the model $\lambda = (A, B, \pi)$. That is, calculate $p\{O|\lambda\}$. It is often used in the applications of signal classification, image segmentation, and speech recognition. However, the direct calculation of the likelihood requires huge computational load and thus is unrealistic. Rather, we can use a fast algorithm, known as Upward-Downward, to implement the likelihood calculation in a reasonable amount of time [21].

**State Sequence Estimation**: Given a sequence of observation $O = \{O_1, O_2, \ldots, O_T\}$ and the model parameters, determine the most likely sequence of the hidden states. The well-known Viterbi Algorithm can be used to efficiently calculate this likelihood of the hidden states and determine the most likely state sequence [21].
**HMM Training**: This problem comes into consideration when we need to build an HMM given a set of training data. More specifically, we need to estimate the parameters $\lambda = (A, B, \pi)$ in a maximum likelihood (ML) sense. That is

$$\hat{\lambda} = \arg \max_\lambda p\{O|\lambda\}.$$  

(4.6)

If we are given the hidden states, the estimate of $\lambda$ is as simple as the ML estimate. However, the states are unknown, therefore, a direct estimate of the parameters is intractable. The *Expectation-Maximization* (EM) algorithm tackles this problem in an iterative manner [21]. It consists of two steps, the E step and the M step. In the E step, we start with an initial guess of $\lambda^0$, then calculate the expectation value of the log-likelihood $E[\log p\{O, S|O, \lambda^t\}]$ for each iteration $l$. In the maximization step, or M step, we update the current parameter $\lambda^{t+1}$ using $\lambda^t$ from the previous iteration. The EM algorithm is guaranteed to converge to a local maximum of the likelihood function $p\{O|\lambda\}$ in an efficient manner. Hence, an HMM can be built using the EM algorithm. A detailed description of the wavelet-domain EM algorithm is presented in [25].

### 4.2 Wavelet-Domain Hidden Markov Models

HMMs are widely applied to speech recognition tasks due to their superior performance. However, direct applications of HMMs in other applications like signal and image classification and denoising are prohibitively complicated due to the existence of many possible states in the HMMs. For instance, if we associate a gray value of a pixel with a
hidden state, there can be more than 200 states in a natural image. As a result, a direct modeling of this kind of signals using HMMs is unrealistic.

The wavelet transform makes the modeling of HMMs for the problems heretofore considered a realistic issue because of its unique properties. One of them is compression, which states that a large amount of the wavelet coefficients has small values, and a small amount of the wavelet coefficients has large values. Hence, it is possible to partition the wavelet coefficients into two groups, in which one is a large-value group, and the other is a small-value group, and associate the hidden states with each class. That is, we have two states; large and small states. The other properties are known as clustering and persistence. They show that there are dependencies between the wavelet coefficients in the same and different scales although the wavelet transform nearly decorrelates a signal. Combined with the tree structure of the wavelet coefficients, a transition probability distribution can be built upon the states of the coefficients across different scales. In the following two subsections, two basic models, which are the Gaussian Mixture Model and Hidden Markov Tree Model, are presented based on the ideas given above.

4.2.1 Gaussian Mixture Model

In general, the statistical distribution of the wavelet coefficients is not Gaussian due to the compression property. That is, most of the significant information of the original signal is concentrated in a few large coefficients. As a result, the probability density function (PDF) of the wavelet coefficients, \( f_W(w) \), can be modeled by a peak and heavy-
tailed non-Gaussian distribution [27], where $W$ represents the random variable of $w$. More specifically, Crouse et al. has proposed Gaussian mixture model (GMM) to closely model the PDF the wavelet coefficients [25].

In general, an $N$-state GMM for a random variable $W$ consists of the following two elements.

- A discrete random state variable $S$ taking its value from a finite set of states, $s = \{1, 2, \ldots, N\}$, according to the probability mass function (pmf) $p_S(n)$.
- The conditional Gaussian distribution $f_{W|S}(w|S = s)$, $s = \{1, 2, \ldots, N\}$. That is, each wavelet coefficient is modeled as conditional Gaussian. Given its hidden state, it is a Gaussian distribution.

Given the above two elements, the PDF of the wavelet coefficients can be modeled as a mixture of $N$ conditional Gaussian distributions as follows

$$f_W(w) = \sum_{n=1}^{N} p_S(n) f_{W|S}(w|S = n), \quad (4.7)$$

where

$$f_{W|S}(w|S = n) = \frac{1}{\sqrt{2\pi}\sigma_n^2} \exp\left(-\frac{(w - \mu_n)^2}{2\sigma_n^2}\right) \triangleq \mathcal{N}(w; \mu_n, \sigma_n^2). \quad (4.8)$$

In practice, a two-state GMM can closely fit the distribution of the wavelet coefficients and keep the computational load comfortable to deal with [27, 25]. Although increasing the number of states can arbitrarily increase the accuracy of the GMM, we cannot afford the computational complexity. Therefore, we model the GMM in our system as a two-state GMM as follows

$$f_W(w) = \sum_{n=1}^{2} p_S(n) f_{W|S}(w|S = n). \quad (4.9)$$
4.2.2 Hidden Markov Tree Model

It is well known that although the wavelet transform tries to decorrelate a signal, there are still residual dependencies, which are called clustering and persistence, among the wavelet coefficients. Clustering states that a wavelet coefficient and its adjacent neighbors tend to be in the same state with high probability. Persistence states that the magnitude of a wavelet coefficient tend to propagate across scales, that is, a wavelet coefficient and its ancestor and descendents tend to be in the same state with high probability. Both of the properties play critical roles in modeling the statistical behaviors of a signal in the wavelet domain.

The hidden Markov tree (HMT) is a model that captures the underlying dependencies among the wavelet coefficients across different scales. It is built based on the binary tree-structure of the wavelet coefficients in the time-frequency plane. To make it clear, a binary HMT in the time-frequency plane is shown in Figure 4.1. In the HMT, each wavelet coefficient is associated with its hidden state. The white nodes represent the hidden states and the black ones represent the corresponding wavelet coefficients. A state is usually called a node in the HMM’s literature. The dependency between a pair of the states of the wavelet coefficients is described by the link between the two state nodes as shown in Figure 4.1. More specifically, it can be described by a transition probability $\epsilon_{i,\rho(i)}^{mr}$, which is defined as [25]

$$
\epsilon_{i,\rho(i)}^{mr} = p_{S_i|S_{\rho(i)}}[S_i = m|S_{\rho(i)} = r].
$$

(4.10)
Figure 4.1: The tree-structure of the wavelet coefficients in the time-frequency plane.
That is, the conditional probability that $S_i$ is in state $m$ given its parent $S_{\rho(i)}$ is in state $r$. Since there are two hidden states in our system, the parent-to-child transition probability matrix can be specifically described as

$$A_i = \begin{bmatrix} p_{i}^{S \rightarrow S} & p_{i}^{S \rightarrow L} \\ p_{i}^{L \rightarrow S} & p_{i}^{L \rightarrow L} \end{bmatrix}.$$ \hfill (4.11)

The entry of the matrix $p_{i}^{S \rightarrow S} (p_{i}^{L \rightarrow L})$ denotes the probability that node $i$ is in small (large) state given its parent node is in small (large) state. Similarly, $p_{i}^{L \rightarrow S} (p_{i}^{S \rightarrow L})$ denotes the probability that node $i$ is in small (large) state given its parent node is in large (small) state. It is expected that $p_{i}^{S \rightarrow S} > p_{i}^{S \rightarrow L}$ and $p_{i}^{L \rightarrow L} > p_{i}^{L \rightarrow S}$, which is dictated by the persistence property.

In addition to the dependencies between the state of the wavelet coefficients, there are also conditional independent relationships between the hidden states, the Markov property. That is, given its parent state $S_{\rho(i)}$, node $i$ is only related to its descent nodes, and given its child’s state $S_j$, it is independent of $S_j$’s descents. Furthermore, given its parent and child states, node $i$ is independent of the entire tree. For instance, node 2 in Figure 4.1 is independent of the subtree on the right given the state of node 1, and is independent of node 8 and 9 given the state of node 4. Given the states of node 2, node 4, and node 5 simultaneously, then node 2 is independent of the entire tree.

Combined with the two-state Gaussian mixture model and the properties of the HMT, the following items are necessary to model an HMT.

- $p_{S_1}(m)$, the pmf of the root node being in state $m$. 

• The state transition probability matrix, $A$, as described by Equation 4.11.

• $\mu_{i,m}$ and $\sigma_{i,m}$, the mean and variance of the wavelet coefficient $i$ given its hidden state $S_i$ is in state $m$.

These parameters are obtained via the HMM training process, which is better known as the \textit{Expectation-Maximization} (EM) algorithm [25]. The likelihood of an observation is obtained via the Upward-Downward algorithm. In summary, the classification task using wavelet-domain HMMs consists of the following steps:

• Training phase—The HMM training estimates a parameter vector $\hat{\theta}_j$ for class $j$, where $j \in \{1, 2, \ldots, N\}$ and $N$ is the number of class.

• Likelihood determination phase—Given the wavelet transform $w$ of an observation $x$, the likelihood function $p(w|\hat{\theta}_j)$ is calculated. The observation $x$ is assigned to a class that has the maximum likelihood.
CHAPTER V

EXPERIMENTAL RESULTS & ANALYSIS

The development of the hyperspectral soil texture classification system in the wavelet domain has been motivated by the effective and efficient analysis capability of wavelets, which have a unifying role in signal processing. The hyperspectral signals, in their very nature, are random and time-varying, and therefore no simple statistical model can capture their underlying properties. The complex stochastic models proposed in this thesis, with their proven successes in modeling real-world statistical behaviors, are expected to produce promising results. However, to realize our goal, we need to put them into practice. This chapter presents the experimental results of our system using hyperspectral soil textures. The classification accuracies for both the ML and HMM are presented. Comparisons between the results and critical analysis are also presented. To validate the applicability of the ML and HMM algorithms, two experiments are conducted.

5.1 Experiment I

In this experiment, the soil samples used for training and testing are purchased online from WARD’s Natural Science, (http://www.wardsci.com). The purchase codes for the soil samples are 45V1982 (Pure Black Clay), 45V1982 (Pure Black Silt), and
45V1983 (Pure Fine Sand), respectively. This ensures that the soil samples are pure or nearly pure, making the experimental results reliable.

5.1.1 Data Preparation

The data preparation consists of two steps. In the first step, the hyperspectral signals of pure sand, silt, and clay are recorded using a hand-held specroradiometer from ASD Inc. In the second step, the linear mixture model (LMM) is used to synthesize the hyperspectral soil textures that are used in training and testing. For a comprehensive study, two groups of data are prepared. In the first group, we have twelve hyperspectral soil textures. That is, in addition to the three classes obtained in step 1, there are other nine classes that are synthesized using the LMM. In the second group, there are three classes, which are sand dominant hyperspectral soil texture, silt dominant hyperspectral soil texture, and clay dominant hyperspectral soil texture. They are also synthesized using the data obtained from step 1 via the LMM. This group is used to test the performance of the system in some cases where a soil sample is classified as being one of the sand dominant, silt dominant, or clay dominant texture. This may be more practical since we do not necessarily need to classify a soil sample into one of the twelve classes in many applications.

The coefficients $\alpha$, $\beta$, and $\gamma$ of the LMM as in Equation (2.7), which represent the abundances of the three endmembers in soil samples, are selected in accordance to the USDA soil texture triangle of Figure 1.1 and shown in Table 5.1 and Table 5.2 for group 1 and group 2, respectively. In the experiment, we took 200 hyperspectral soil signals of
Table 5.1: Endmember Abundances of the Hyperspectral Soil Textures in group 1

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Endmembers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sand</td>
<td>Silt</td>
<td>Clay</td>
<td></td>
</tr>
<tr>
<td>Sand</td>
<td>1.00</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Sandy Clay</td>
<td>0.55</td>
<td>0.05</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>Sandy Clay Loam</td>
<td>0.60</td>
<td>0.15</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>Sandy Loam</td>
<td>0.65</td>
<td>0.25</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>Silt</td>
<td>0</td>
<td>1.00</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Silt Loam</td>
<td>0.25</td>
<td>0.60</td>
<td>0.15</td>
<td></td>
</tr>
<tr>
<td>Silty Clay</td>
<td>0.5</td>
<td>0.45</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>Silty Clay Loam</td>
<td>0.10</td>
<td>0.55</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>Clay</td>
<td>0</td>
<td>0</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>Clay Loam</td>
<td>0.30</td>
<td>0.35</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>Loam</td>
<td>0.40</td>
<td>0.40</td>
<td>0.20</td>
<td></td>
</tr>
<tr>
<td>Loamy Sand</td>
<td>0.85</td>
<td>0.05</td>
<td>0.10</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Endmember Abundances of the Hyperspectral Soil Textures in group 2

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Endmembers</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sand</td>
<td>Silt</td>
<td>Clay</td>
<td></td>
</tr>
<tr>
<td>Sand dominant</td>
<td>0.7</td>
<td>0.1</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Silt dominant</td>
<td>0.1</td>
<td>0.7</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>Clay dominant</td>
<td>0.1</td>
<td>0.2</td>
<td>0.7</td>
<td></td>
</tr>
</tbody>
</table>
the sand, silt, and clay soil samples, then synthesize the other hyperspectral soil textures in group 1 and group 2 using the coefficients in Table 5.1 and Table 5.2. The hyperspectral signals of sand, silt, and clay, together with the synthesized hyperspectral signals, are shown in Figure 5.1, Figure 5.2, and Figure 5.3. Please note that only a small amount of signals in each class are shown in these figures.

It is observed from the hyperspectral soil textures that there are slight noise around the lower spectral bands. The noise presence may cause negative effects on the classification performance. To overcome this problem, we can take the following two methods into consideration. One method is to apply some kind of filtering to the hyperspectral signals. However, this will only increase the system’s complexity and does not contribute to the classification performance. Because the lower bands of all of the hyperspectral soil signals vary slightly and look very similar to each other, therefore there will be only a minor difference between the statistical behaviors in the wavelet domain. The other method is to use the 1024 samples in the higher spectral bands for training and classification. A major benefit of this method is the reduction in computational complexity. Moreover, the higher spectral bands of the soil hyperspectral signals contain most of the major features. Therefore, the higher 1024 samples, in particular, the samples from band 1477 to band 2500, are used in this experiment.
Figure 5.1: Hyperspectral Soil Textures.
Figure 5.2: Hyperspectral Soil Textures.

Figure 5.3: Hyperspectral Soil Textures.
5.1.2 Experimental Results and Analysis

First we need to partition the data of each class into training and testing data with equal number of samples. As for the wavelet transform, we incorporate the Haar, Daubechies orthonormal wavelets with filter length $N = 4, 6, \text{and} 8$, Coiflet wavelets with filter length $N = 12, 18, \text{and} 24$, and Symlet wavelets with filter length $N = 8, 10, \text{and} 12$. In the wavelet filter tree algorithm, we apply the maximum number of decomposition levels, which is $L = \log_2 1024 = 10$, to the hyperspectral signals. Before we start testing our systems, the first task at hand is to train the ML and HMM classifiers using the wavelet transform of the hyperspectral soil signals. The training process is simple and straightforward for the ML classifier as described in Equations (3.12) and (3.13). We train the ML classifier using the wavelet energy signals of the twelve hyperspectral soil textures. As a result, we obtain a mean vector and a covariance matrix for each soil class. Unlike the ML training, the training process for the HMM does not use the wavelet energy signals of the hyperspectral soil signals. Rather, it applies the Expectation-Maximization (EM) algorithm to the wavelet coefficients. As a result, we obtain a parameter vector for each wavelet coefficient, as stated in Chapter IV.

The testing results of the ML and HMM classifiers for the data in group 1 are shown in Table 5.3 and Table 5.4, respectively. It is clear that the classification accuracy of sand is the highest one, which is as high as 97% (the best case). This is primarily because the statistical behaviors of the features of the hyperspectral sand texture have the most prominent difference than those of the remaining hyperspectral soil textures. On the other
hand, the classification accuracy for some other hyperspectral soil texture, such as sandy clay, is as low as 30% (the worst case). This is because the statistical behavior of the features of the hyperspectral sandy clay has much overlaps with those of others, such as sandy clay loam. Thus we end up with poor classification result.

The testing results of the ML and HMM classifiers for the data in group 2 are shown in Table 5.5 and Table 5.6, respectively. It is clear that both the ML and HMM classifiers have very promising classification accuracies, which are much better than the results obtained from the data of group 1. They even achieve perfect result in some cases. This is reasonable since there are less overlaps in group 2 data than those of group 1 data.

We also notice that, from the classification results alone, the ML classifier slightly outperforms the HMM classifier. Thus, we may want to choose the ML classifier over the HMM classifier. However, from the practical perspective, we should take the following items into consideration:

- The ML classifier needs enough training samples so that the sample mean and variance are as accurate as possible. This requirement is not a problem in our experiment, however, it may be a big problem in other cases, where the training data is very limited.
- There should be an enough amount of variance among the training data to ensure the inverse of the covariance matrix is not ill conditioned. In other words, there should not exist high correlation among the training data.
- The Gaussian assumption is not always applicable.

In such cases, the HMM classifier is a more practical alternative, since it does not need much training data to be stable. Rather, the EM algorithm for the HMM training finds the statistical parameters for each wavelet coefficient in an iterative manner, which
Table 5.3: Classification results of the ML classifier for the twelve-class case

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Wavelet Filters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
<td>Sym12</td>
</tr>
<tr>
<td>Sand</td>
<td>0.90</td>
<td>0.89</td>
<td>0.95</td>
<td>0.94</td>
<td>0.87</td>
<td>0.80</td>
<td>0.89</td>
<td>0.97</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td>Sandy Clay</td>
<td>0.46</td>
<td>0.50</td>
<td>0.88</td>
<td>0.75</td>
<td>0.78</td>
<td>0.65</td>
<td>0.82</td>
<td>0.85</td>
<td>0.88</td>
<td>0.85</td>
</tr>
<tr>
<td>Sandy Clay Loam</td>
<td>0.84</td>
<td>0.75</td>
<td>0.85</td>
<td>0.86</td>
<td>0.90</td>
<td>0.80</td>
<td>0.87</td>
<td>0.78</td>
<td>0.73</td>
<td>0.73</td>
</tr>
<tr>
<td>Sandy Loam</td>
<td>0.50</td>
<td>0.65</td>
<td>0.70</td>
<td>0.75</td>
<td>0.73</td>
<td>0.74</td>
<td>0.76</td>
<td>0.67</td>
<td>0.67</td>
<td>0.65</td>
</tr>
<tr>
<td>Silt</td>
<td>0.64</td>
<td>0.68</td>
<td>0.78</td>
<td>0.70</td>
<td>0.65</td>
<td>0.75</td>
<td>0.76</td>
<td>0.63</td>
<td>0.83</td>
<td>0.84</td>
</tr>
<tr>
<td>Silt Loam</td>
<td>0.55</td>
<td>0.75</td>
<td>0.70</td>
<td>0.76</td>
<td>0.70</td>
<td>0.72</td>
<td>0.86</td>
<td>0.77</td>
<td>0.56</td>
<td>0.50</td>
</tr>
<tr>
<td>Silty Clay</td>
<td>0.43</td>
<td>0.38</td>
<td>0.55</td>
<td>0.60</td>
<td>0.72</td>
<td>0.65</td>
<td>0.70</td>
<td>0.54</td>
<td>0.62</td>
<td>0.60</td>
</tr>
<tr>
<td>Silty Clay Loam</td>
<td>0.55</td>
<td>0.70</td>
<td>0.80</td>
<td>0.75</td>
<td>0.77</td>
<td>0.50</td>
<td>0.54</td>
<td>0.67</td>
<td>0.71</td>
<td>0.66</td>
</tr>
<tr>
<td>Clay</td>
<td>0.50</td>
<td>0.65</td>
<td>0.70</td>
<td>0.75</td>
<td>0.70</td>
<td>0.55</td>
<td>0.45</td>
<td>0.50</td>
<td>0.62</td>
<td>0.54</td>
</tr>
<tr>
<td>Clay Loam</td>
<td>0.66</td>
<td>0.60</td>
<td>0.55</td>
<td>0.50</td>
<td>0.55</td>
<td>0.51</td>
<td>0.78</td>
<td>0.68</td>
<td>0.84</td>
<td>0.80</td>
</tr>
<tr>
<td>Loam</td>
<td>0.55</td>
<td>0.45</td>
<td>0.60</td>
<td>0.64</td>
<td>0.56</td>
<td>0.63</td>
<td>0.66</td>
<td>0.84</td>
<td>0.89</td>
<td>0.87</td>
</tr>
<tr>
<td>Loamy Sand</td>
<td>0.44</td>
<td>0.60</td>
<td>0.70</td>
<td>0.75</td>
<td>0.65</td>
<td>0.54</td>
<td>0.76</td>
<td>0.81</td>
<td>0.87</td>
<td>0.83</td>
</tr>
<tr>
<td>Overall</td>
<td>0.585</td>
<td>0.633</td>
<td>0.73</td>
<td>0.729</td>
<td>0.715</td>
<td>0.653</td>
<td>0.737</td>
<td>0.726</td>
<td>0.765</td>
<td>0.735</td>
</tr>
<tr>
<td>Confidence Interval</td>
<td>± 0.04</td>
<td>± 0.04</td>
<td>± 0.04</td>
<td>± 0.03</td>
<td>± 0.04</td>
<td>± 0.04</td>
<td>± 0.03</td>
<td>± 0.03</td>
<td>± 0.03</td>
<td>± 0.03</td>
</tr>
<tr>
<td>Soil Textures</td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
<td>Sym12</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>Sand</td>
<td>0.95</td>
<td>0.93</td>
<td>0.93</td>
<td>0.95</td>
<td>0.78</td>
<td>0.60</td>
<td>0.60</td>
<td>0.84</td>
<td>0.97</td>
<td>0.73</td>
</tr>
<tr>
<td>Sandy Clay</td>
<td>0.30</td>
<td>0.35</td>
<td>0.64</td>
<td>0.45</td>
<td>0.58</td>
<td>0.65</td>
<td>0.70</td>
<td>0.45</td>
<td>0.61</td>
<td>0.40</td>
</tr>
<tr>
<td>Sandy Clay Loam</td>
<td>0.51</td>
<td>0.46</td>
<td>0.44</td>
<td>0.44</td>
<td>0.40</td>
<td>0.35</td>
<td>0.45</td>
<td>0.57</td>
<td>0.33</td>
<td>0.38</td>
</tr>
<tr>
<td>Sandy Loam</td>
<td>0.44</td>
<td>0.40</td>
<td>0.43</td>
<td>0.44</td>
<td>0.40</td>
<td>0.35</td>
<td>0.30</td>
<td>0.40</td>
<td>0.42</td>
<td>0.69</td>
</tr>
<tr>
<td>Silt</td>
<td>0.66</td>
<td>0.64</td>
<td>0.75</td>
<td>0.73</td>
<td>0.65</td>
<td>0.70</td>
<td>0.60</td>
<td>0.55</td>
<td>0.30</td>
<td>0.65</td>
</tr>
<tr>
<td>Silt Loam</td>
<td>0.56</td>
<td>0.50</td>
<td>0.54</td>
<td>0.50</td>
<td>0.65</td>
<td>0.70</td>
<td>0.68</td>
<td>0.86</td>
<td>0.30</td>
<td>0.82</td>
</tr>
<tr>
<td>Silty Clay</td>
<td>0.40</td>
<td>0.53</td>
<td>0.50</td>
<td>0.48</td>
<td>0.55</td>
<td>0.80</td>
<td>0.75</td>
<td>0.80</td>
<td>0.59</td>
<td>0.91</td>
</tr>
<tr>
<td>Silty Clay Loam</td>
<td>0.56</td>
<td>0.55</td>
<td>0.56</td>
<td>0.45</td>
<td>0.47</td>
<td>0.55</td>
<td>0.50</td>
<td>0.55</td>
<td>0.36</td>
<td>0.59</td>
</tr>
<tr>
<td>Clay</td>
<td>0.53</td>
<td>0.72</td>
<td>0.87</td>
<td>0.86</td>
<td>0.95</td>
<td>0.96</td>
<td>0.96</td>
<td>0.97</td>
<td>0.55</td>
<td>0.96</td>
</tr>
<tr>
<td>Clay Loam</td>
<td>0.63</td>
<td>0.77</td>
<td>0.75</td>
<td>0.70</td>
<td>0.45</td>
<td>0.40</td>
<td>0.45</td>
<td>0.40</td>
<td>0.51</td>
<td>0.37</td>
</tr>
<tr>
<td>Loam</td>
<td>0.40</td>
<td>0.42</td>
<td>0.40</td>
<td>0.44</td>
<td>0.35</td>
<td>0.55</td>
<td>0.40</td>
<td>0.45</td>
<td>0.49</td>
<td>0.33</td>
</tr>
<tr>
<td>Loamy Sand</td>
<td>0.38</td>
<td>0.53</td>
<td>0.55</td>
<td>0.50</td>
<td>0.40</td>
<td>0.45</td>
<td>0.40</td>
<td>0.56</td>
<td>0.51</td>
<td>0.59</td>
</tr>
<tr>
<td>Overall</td>
<td>0.526</td>
<td>0.567</td>
<td>0.613</td>
<td>0.578</td>
<td>0.553</td>
<td>0.588</td>
<td>0.565</td>
<td>0.616</td>
<td>0.495</td>
<td>0.618</td>
</tr>
<tr>
<td>Confidence Interval</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.04</td>
</tr>
</tbody>
</table>
Table 5.5: Classification results of the ML classifier for the three-class case

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Wavelet Filters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
<td>Sym12</td>
</tr>
<tr>
<td>Sand dominant</td>
<td>0.95</td>
<td>0.95</td>
<td>0.90</td>
<td>1.00</td>
<td>0.92</td>
<td>0.94</td>
<td>0.95</td>
<td>0.95</td>
<td>0.97</td>
<td>0.95</td>
</tr>
<tr>
<td>Silt dominant</td>
<td>0.98</td>
<td>1.00</td>
<td>0.96</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
<td>0.99</td>
<td>1.00</td>
<td>0.97</td>
<td>0.96</td>
</tr>
<tr>
<td>Clay dominant</td>
<td>0.95</td>
<td>0.94</td>
<td>0.97</td>
<td>0.96</td>
<td>0.98</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>1.00</td>
</tr>
<tr>
<td>Overall</td>
<td>0.96</td>
<td>0.943</td>
<td>0.943</td>
<td>0.97</td>
<td>0.953</td>
<td>0.967</td>
<td>0.98</td>
<td>0.983</td>
<td>0.967</td>
<td>0.97</td>
</tr>
<tr>
<td>Confidence Interval (95%)</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.01</td>
<td>±0.02</td>
<td>±0.01</td>
<td>±0.01</td>
<td>±0.01</td>
<td>±0.01</td>
<td>±0.01</td>
</tr>
</tbody>
</table>

Table 5.6: Classification results of the HMM classifier for the three-class case

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Wavelet Filters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
<td>Sym12</td>
</tr>
<tr>
<td>Sand dominant</td>
<td>0.95</td>
<td>0.65</td>
<td>0.83</td>
<td>1.00</td>
<td>0.83</td>
<td>0.81</td>
<td>0.79</td>
<td>0.88</td>
<td>1.00</td>
<td>0.78</td>
</tr>
<tr>
<td>Silt dominant</td>
<td>1.00</td>
<td>1.00</td>
<td>0.97</td>
<td>0.97</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.94</td>
<td>0.97</td>
<td></td>
</tr>
<tr>
<td>Clay dominant</td>
<td>0.75</td>
<td>0.84</td>
<td>0.96</td>
<td>0.89</td>
<td>0.94</td>
<td>0.97</td>
<td>0.99</td>
<td>1.00</td>
<td>0.90</td>
<td>1.00</td>
</tr>
<tr>
<td>Overall</td>
<td>0.90</td>
<td>0.83</td>
<td>0.92</td>
<td>0.953</td>
<td>0.9133</td>
<td>0.923</td>
<td>0.923</td>
<td>0.953</td>
<td>0.947</td>
<td>0.917</td>
</tr>
<tr>
<td>Confidence Interval (95%)</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.03</td>
</tr>
</tbody>
</table>
means it does not necessarily need much training data. Furthermore, it is not necessary that there should be enough variance among the training data, and the conditional Gaussian distribution in the HMM classifier is applicable for most of the applications. These make the HMM classifier more robust in the situations considered above. Thus, we should be careful as to which classifier to choose in the practical situation.

To have a better idea on the robustness of the classifiers, we examine the relationship between the number of training samples and the classification accuracy. This is illustrated in Figure 5.4 and Figure 5.5. For the HMM classifier, we choose the best case in the three-class mode, which has the classification accuracies of 100%, 97%, and 89% for sand dominant, silt dominant, and clay dominant textures, respectively. The wavelet filter is Daubechies with length $N = 8$. For the ML classifier, we also choose the best case in the three-class mode, which has the classification accuracies of 95%, 100%, and 100% for sand dominant, silt dominant, and clay dominant textures, respectively. The wavelet filter is Symlets with length $N = 8$. Figure 5.4 shows that as the number of training samples decreases, the classification accuracies vary slightly. The classification accuracy is still good even when we have only one training sample. This clearly shows that the HMM classifier is very robust. Figure 5.5 shows that as the number of training samples decreases from 100 to 10, the classification accuracy also varies slightly. However, when the number of training samples is 7 or less, the inverse of the covariance matrix becomes ill conditioned. In this case, the result is not reliable and consequently the ML classifier is
Figure 5.4: The performance of the HMM classifier.

not applicable. We thus come to a conclusion that the HMM classifier can be applied to some cases where we only have a very limited number of training samples at hand.

5.2 Experiment II

This experimental analysis is conducted using hyperspectral signals collected by analytical spectral devices (ASD) spectroradiometer at the Mississippi State University north farm. In this case study, 25 hyper spectral data of sand, silt, and clay are used. These data are different from those in Experiment I in that they are not pure and are collected in the real field. The training and testing data are obtained by mixing the hyperspectral signals
of sand, silt, and clay with different abundances, which are the same as the ones used in Experiment I, group 2 (see Table 5.2). In the classification process, we use the same methods as those in Experiment I. The classification results of the ML and HMM classifiers are shown in Table 5.7 and Table 5.8.

The results show very promising performance of the classifiers. The ML classifier also slightly outperforms the HMM classifier. This case study shows that the methods can be applied to the real soil texture classification task.
Table 5.7: Classification results of the ML classifier in Experiment II

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Wavelet Filters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
</tr>
<tr>
<td>Sand dominant</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
<td>1.00</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Silt dominant</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td>0.96</td>
<td>0.96</td>
</tr>
<tr>
<td>Clay dominant</td>
<td>1.00</td>
<td>0.96</td>
<td>0.84</td>
<td>0.98</td>
<td>0.96</td>
<td>1.00</td>
<td>0.98</td>
<td>0.98</td>
<td>0.96</td>
</tr>
<tr>
<td>Overall</td>
<td>1.00</td>
<td>0.987</td>
<td>0.933</td>
<td>0.993</td>
<td>0.983</td>
<td>0.99</td>
<td>0.983</td>
<td>0.98</td>
<td>0.973</td>
</tr>
<tr>
<td>Confidence Interval (95%)</td>
<td>0</td>
<td>±0.02</td>
<td>±0.04</td>
<td>±0.01</td>
<td>±0.02</td>
<td>±0.01</td>
<td>±0.02</td>
<td>±0.02</td>
<td>±0.02</td>
</tr>
</tbody>
</table>

Table 5.8: Classification results of the HMM classifier in Experiment II

<table>
<thead>
<tr>
<th>Soil Textures</th>
<th>Wavelet Filters</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Haar</td>
<td>Db4</td>
<td>Db6</td>
<td>Db8</td>
<td>Coif12</td>
<td>Coif18</td>
<td>Coif24</td>
<td>Sym8</td>
<td>Sym10</td>
</tr>
<tr>
<td>Sand dominant</td>
<td>1.00</td>
<td>0.85</td>
<td>0.90</td>
<td>1.00</td>
<td>0.90</td>
<td>0.85</td>
<td>0.85</td>
<td>0.92</td>
<td>1.00</td>
</tr>
<tr>
<td>Silt dominant</td>
<td>1.00</td>
<td>1.00</td>
<td>0.95</td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
</tr>
<tr>
<td>Clay dominant</td>
<td>0.85</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.96</td>
</tr>
<tr>
<td>Overall</td>
<td>0.95</td>
<td>0.917</td>
<td>0.917</td>
<td>0.967</td>
<td>0.96</td>
<td>0.95</td>
<td>0.97</td>
<td>0.987</td>
<td>0.947</td>
</tr>
<tr>
<td>Confidence Interval (95%)</td>
<td>±0.03</td>
<td>±0.04</td>
<td>±0.04</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.03</td>
<td>±0.02</td>
<td>±0.04</td>
</tr>
</tbody>
</table>
CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This thesis presents a novel system that facilitates the soil texture classification process, which is time consuming and requires a lot of lab work using traditional methods. We incorporate hyperspectral soil textures, better known as hyperspectral soil signals, into our system and conduct the classification task in the wavelet domain. To investigate the performance of different mother wavelets, we incorporate ten different mother wavelets, which are Haar, Daubechies\( N \) \((N = 4, 6, 8)\), Coiflets\( N \) \((N = 12, 18, 24)\), Symlets\( N \) \((N = 8, 10, 12)\), where \(N\) is the length of the wavelet filters. We exploit the effectiveness, robustness, and applicability of two statistical classifiers. One of them is the maximum likelihood (ML) classifier, and the other is the hidden Markov model (HMM) classifier. The system can work in two modes. One is the twelve-class mode, where the system is designed to classify twelve soil textures defined in the USDA soil texture triangle. The other is the three-class mode, where the system is designed to classify three typical types of soil textures, namely, the sand dominant, silt dominant, and clay dominant soil textures.

The data preparation roughly consists of two steps. In the first step, we took the hyperspectral signals of pure sand, silt, and clay using a hand-held spectroradiometer from ASD Inc. In the second step, we synthesized other hyperspectral soil textures using the linear
mixture model (LMM). We partition the data into two groups. The first group contains the twelve hyperspectral soil textures for the system working under a twelve-class mode. The second group contains three types of hyperspectral soil textures for the system working under a three-class mode, which is usually a more practical case.

6.1 Conclusions

The experimental results show that the ML classifier outperforms the HMM classifier both in the twelve-class mode and the three-class mode. The ML classifier has the best performance in the twelve-class mode when we choose “Sym12” as the wavelet filter, and has the best performance in the three-class mode when we choose “Sym8” as the wavelet filters. However we have observed that although the performance of the HMM classifier is slightly inferior to that of the ML classifier, it has also achieved good classification results, especially in the three-class mode. Furthermore, in some practical applications, we would prefer the HMM classifier due to its robustness. We have shown that as the number of training samples decreases, the outcomes of the HMM classifier vary slightly whereas it is not always the case for the ML classifier. Specifically, it has been shown that the ML classifier fails for training samples of 7 or less. This is of particular interest to some remote sensing applications where only a limited number of training data are available, which often arises in remote sensing image classification, segmentation, and target recognition.
6.2 Suggestions for Future Works

Soil texture classification is a challenging task. One of the reasons is that there are many types of soils even in a specific soil texture. For instance, there are “black clay” and “red clay” in the clay texture. Hence, it is necessary to build a comprehensive soil texture library that consists of most of the soil types we have ever known. Thus provides a better source for the training phase. User options on what kind of soil texture he/she wants to classify will also be added in the future.

We have shown that the HMM classifier is a promising tool. However, there are some further works to be incorporated to improve its performance. One of them is to increase the number of the hidden states, thus achieving better classification results. Another one is to improve the efficiency of the HMM training process. We have observed that the computational time in the training phase of the HMM classifier is almost 10 times that of the ML classifier. Therefore we need to simplify the parameter estimation procedure without a degradation in the performance.
REFERENCES


