NOVEL PATTERN RECOGNITION TECHNIQUES FOR IMPROVED TARGET DETECTION IN HYPERSPECTRAL IMAGERY

A Dissertation

by

WESAM ADEL SAKLA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2009

Major Subject: Electrical Engineering

NOVEL PATTERN RECOGNITION TECHNIQUES FOR IMPROVED TARGET DETECTION IN HYPERSPECTRAL IMAGERY

A Dissertation

by

WESAM ADEL SAKLA

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Andrew Chan
Committee Members,	Jim Ji
	Deepa Kundur
	Sheng-Jen Hsieh
Head of Department,	Costas Georghiades

December 2009

Major Subject: Electrical Engineering

ABSTRACT

Novel Pattern Recognition Techniques for Improved Target Detection in Hyperspectral Imagery. (December 2009)

Wesam Adel Sakla, B.S., The University of South Alabama;

M.S., The University of South Alabama

Chair of Advisory Committee: Dr. Andrew Chan

A fundamental challenge in target detection in hyperspectral imagery is spectral variability. In target detection applications, we are provided with a pure target signature; we do not have a collection of samples that characterize the spectral variability of the target. Another problem is that the performance of stochastic detection algorithms such as the spectral matched filter can be detrimentally affected by the assumptions of multivariate normality of the data, which are often violated in practical situations.

We address the challenge of lack of training samples by creating two models to characterize the target class spectral variability—the first model makes no assumptions regarding inter-band correlation, while the second model uses a first-order Markov-based scheme to exploit correlation between bands. Using these models, we present two techniques for meeting these challenges—the kernel-based support vector data description (SVDD) and spectral fringe-adjusted joint transform correlation (SFJTC).

We have developed an algorithm that uses the kernel-based SVDD for use in full-pixel target detection scenarios. We have addressed optimization of the SVDD kernel-width parameter using the golden-section search algorithm for unconstrained optimization. We investigated a proper number of signatures N to generate for the SVDD target class and found that only a small number of training samples is required relative to the dimensionality (number of bands). We have extended decision-level fusion techniques using the majority vote rule for the purpose of alleviating the problem of selecting a proper value of σ^2 for either of our target variability models. We have shown that heavy spectral variability may cause SFJTC-based detection to suffer and have addressed this by developing an algorithm that selects an optimal combination of the discrete wavelet transform (DWT) coefficients of the signatures for use as features for detection.

For most scenarios, our results show that our SVDD-based detection scheme provides low false positive rates while maintaining higher true positive rates than popular stochastic detection algorithms. Our results also show that our SFJTC-based detection scheme using the DWT coefficients can yield significant detection improvement compared to use of SFJTC using the original signatures and traditional stochastic and deterministic algorithms.

To my Lord, God, and Savior, Jesus Christ

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Andrew Chan, and my committee member, Dr. Jim Ji, for their humble guidance and support throughout the course of my graduate career at Texas A&M. I also want to thank my other committee members, Dr. Kundur and Dr. Hsieh, for their support and time.

I would also like to thank my father, Dr. Adel Sakla, for his invaluable support, guidance, and assistance with my research. I want to thank my mother, Soheir Sakla, for her constant encouragement, prayers, and love. My parents' mentorship and character have molded me spiritually, socially, and academically. Without their influence, this work would not be possible.

I thank my colleagues and the department faculty and staff for making my time at Texas A&M University a rewarding experience. I also want to extend my gratitude to the Department of Defense (DoD) for selecting me as a recipient for the SMART Scholarship Program and sponsoring me through the majority of my graduate program at Texas A&M.

I want to thank my dear friends for their fellowship, love, and support throughout my academic endeavors at Texas A&M. My friends kept me going at times when I did not feel like going. I want to give a special thanks to my best friend, Wendell Pugh, for his endless support and encouragement in the Lord. I want to extend my sincere thanks to Elizabeth and Rafeik Abouelkheir for hosting our weekly youth Bible studies and always keeping their home open for me.

Above all, I thank my Lord, God, and Savior, Jesus Christ, for His help, comfort, and peace that passes all understanding throughout this work and in my life. It is through Him that this work manifests any significance in my life, *for by Him all things were created, both in the heavens and on earth, visible and invisible, whether thrones or dominions or rulers or authorities—all things have been created through Him and for Him (Colossians 1:16).*

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

LIST OF TABLES

TABLE Page

TABLE Page

TABLE Page

TABLE Page

TABLE Page

TABLE Page

TABLE Page

TABLE Page

CHAPTER I

INTRODUCTION

A. HSI Overview

Automatic target recognition (ATR) is a vital and complex step in image processing and exploitation. ATR has experienced significant strides with the advent of hyperspectral imaging (HSI) sensors. ATR systems should be able to detect, classify, recognize, and/or identify targets in an environment where the background is cluttered and targets are at long distances and may be partially occluded, degraded by weather, or camouflaged.[1] HSI may be defined as imagery taken over many spectrally contiguous and spatially co-registered bands. HSI sensors provide plenty of spectral information to uniquely identify materials by their reflectance spectra. A material's reflectance spectrum contains the reflectance values of the material as a function of wavelength. HSI sensors generate images containing both spatial and spectral information that can be used in remote sensing detection and classification applications. Although it is theoretically possible for two completely different materials to exhibit the same spectral signature, targets in ATR applications are typically man-made objects with spectra that differ considerably from the spectra of natural background materials.[2] In contrast to multispectral sensors, which measure reflectance values at wide wavelength bands,

---

This dissertation follows the style of *Optical Engineering*.

hyperspectral sensors measure reflectance values at narrow, contiguous wavelength bands. Consequently, the richer information in hyperspectral imagery has better potential in ATR applications than multispectral imagery.

B. Target Detection Problem

The goal of target detection is to locate pixels containing a material whose spectral composition is known. In HSI target detection applications, the targets are present sparsely throughout an image—they may account for less than 1% of the total pixels in a hyperspectral scene, rendering traditional spatial processing techniques impractical. Consequently, most HSI detection algorithms exploit the spectral information of the scene, an approach otherwise known as *nonliteral exploitation* in the HSI literature.[3] One of the main challenges in HSI processing is *spectral variability*, which refers to the phenomenon that spectra observed from samples of the same material will never be identical.

While many detection algorithms have been developed over the years, spectral variability poses challenges for these algorithms. Current popular detection algorithms such as the spectral matched filter (MF) and its variations stochastically model the background by using first and second order spectral statistics (i.e., mean and covariance) estimated from the scene data. Although these stochastic detectors are mathematically tractable and can work well in some situations, they are only optimal under the assumption of the multivariate normality of the data. The quadratic Neyman-Pearson detector requires the covariance matrix of the target class, which is not available if one is given a single spectral signature obtained from a library.[2,3] The MF and adaptive matched filter (AMF) algorithms assume that the target and background covariance matrices are identical. In real-life scenarios, the multivariate normality assumption is often violated because a hyperspectral image may contain multiple types of terrain, thus causing detection performance to suffer.[4] To increase detection performance with these techniques, improvements to background models can be achieved by increasing the multivariate normality of the data used to characterize the background.[5]

C. Proposed Work

1. The support vector data description

Kernel methods[6] have become increasingly popular in a variety of pattern recognition (PR) applications. The recently-developed support vector data description (SVDD) has its roots in statistical learning theory and is an emerging non-parametric approach for describing a set of data.[7,8] The SVDD is connected with support vector machines (SVMs) and is capable of providing accurate descriptions of a dataset via the use of kernels. The SVDD differs from the SVM in that it only considers samples belonging to the class of interest in order to provide a tight boundary around the data.

The SVDD has been successfully applied in the areas of facial expression analysis[9], gene expression data clustering[10], image retrieval[11], remote sensing image classification[12,13], and HSI anomaly detection[14,15]. To our knowledge, the SVDD has not been used in full-pixel target detection scenarios because no training samples are available to characterize the target class. This work will explore two different models for generation of the target class training data in conjunction with the SVDD to perform target detection in hyperspectral imagery and will investigate the corresponding variables and parameters critical to successful SVDD target detection.

2. *Spectral fringe-adjusted joint transform correlation*

Joint transform correlation has proven to be an effective detection technique in optical PR applications.[16] The fringe-adjusted JTC (FJTC) technique[17] yields better correlation peaks compared to other existing JTC techniques. Recently, the 1-D spectral fringe-adjusted joint transform correlation (SFJTC) technique has been used effectively for performing deterministic target detection in hyperspectral imagery.[18] However, our preliminary experiments show that the technique suffers when significant spectral variability is present in the target signatures.[19] Our work will focus on application of the discrete wavelet transform (DWT) coefficients as features for HSI target detection using the SFJTC technique.

3. Organization of this dissertation

Chapter II discusses the basics of HSI processing and the associated challenges. Chapter III presents a collection of popular HSI target detection algorithms that are found in the literature. In Chapter IV, we describe the hyperspectral scenery used for the experiments and provide the two models that we have used to generate the target signatures used for our work. Chapter V provides formulation of the SVDD and the associated algorithms we have developed in accordance with properly training an SVDD for HSI target detection applications. The concepts of decision-level fusion are also studied in connection with the SVDD. In Chapter VI, we review the wavelet transform and present the SFJTC algorithm. In connection with the wavelet transform and the SFJTC algorithm, we provide the methodology and results obtained when using the DWT coefficients for SFJTC-based detection. This includes our supervised training algorithm for selecting a proper set of DWT coefficients to use prior to detection. Chapter VII presents the final results of our SVDD-based and SFJTC-based detection algorithms on both data models. For comparison, we have generated the detection results of the traditional detection algorithms presented in Chapter III. We summarize our findings and provide conclusions and future directions for research in Chapter VIII.

CHAPTER II

HSI PROCESSING

A. Components of an HSI System

As shown in Figure 1, the basic parts of a hyperspectral imaging system include the illumination source, the atmosphere, the imaged surface, and the sensor. The illumination source may be either passive or active; the sun is the source of illumination in a passive system. In the mode of operation shown in Figure 1, solar energy propagates through the atmosphere, where its intensity and spectral distributions are modified. The energy then interacts with the imaged surface materials and is reflected and/or absorbed by these materials. The reflected energy passes back through the atmosphere where it may be further modified by the atmosphere before reaching the sensor, where it is captured and stored digitally.

HSI sensors, also called *imaging spectrometers*, are typically placed on satellites or aircrafts and acquire digital images in a large number of narrow, adjacent wavelength (frequency) bands that typically span the visible, near-infrared, and mid-infrared portions of the electromagnetic spectrum (0.4 – 2.5 μm). HSI sensors capture the energy reflected by the surface materials and measure the intensity of energy in different parts of the spectrum. The HSI sensor divides the imaged surface into a number of contiguous pixels. The size of each pixel, or *ground resolution cell*, is determined by the optics of the sensor and the sensor altitude. The spatial resolution of the sensor determines the size of the smallest object that can be seen on the surface of the earth by the sensor as a distinct object separate from its surroundings.[2] Figure 2 illustrates the construction of a typical hyperspectral image.

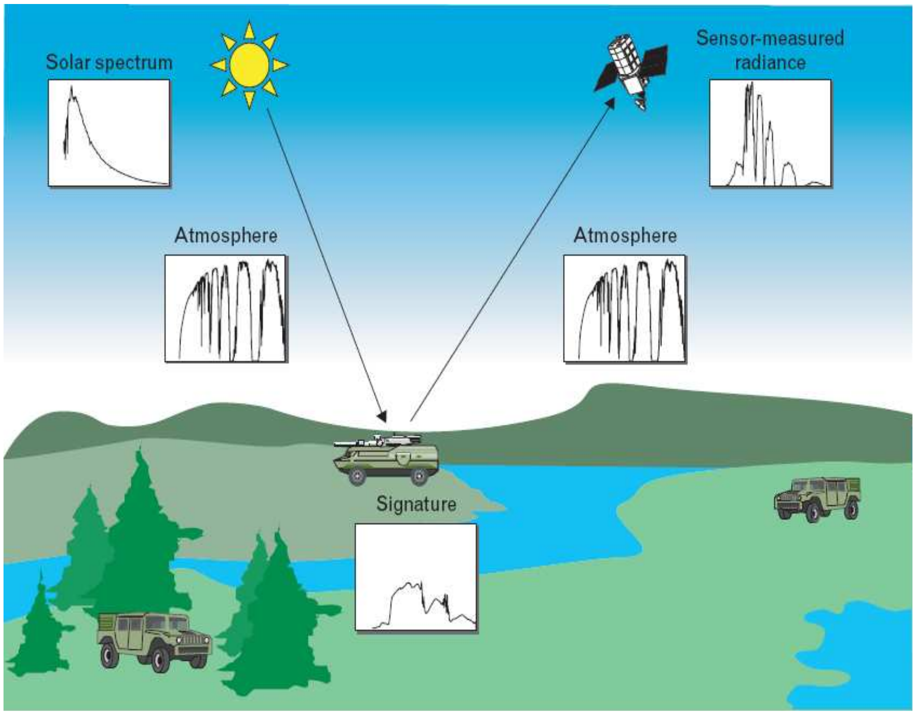


Figure 1. General concept of hyperspectral imaging.[3]

B. Radiance Domain Versus Reflectance Domain

It is vital to stress the difference between *radiance* domain spectra and *reflectance* domain spectra. Although the raw measurements seen by the HSI sensor are radiance values, we desire to obtain the spectral reflectance values—the ratio of reflected energy to incident energy at each wavelength band. The reflectance spectrum, or *spectral signature*, shows the fraction of incident solar energy that is reflected by a material as a function of the wavelength λ of the energy.[20] Conversion to the reflectance domain normalizes the data since the properties of the illumination source and the atmospheric effects are removed. In practice, sophisticated atmospheric compensation codes are used to obtain the reflectance spectrum from the radiance spectrum. The atmosphere also limits which bands in the data are useful because it selectively absorbs radiation at particular wavelengths due to the presence of oxygen and water vapor. The signal-to-noise ratio (SNR) at these absorption bands is very low; as a result, any useful information about the reflectance spectrum is lost.[3] Reflectance is a dimensionless quantity that ranges in value from zero to one. After the data has been atmospherically corrected, the resulting reflectance spectrum for each pixel can be compared to spectra of known materials available in “spectral libraries”[2]. In this work, hyperspectral signatures will be spoken of in the context of their reflectance values.

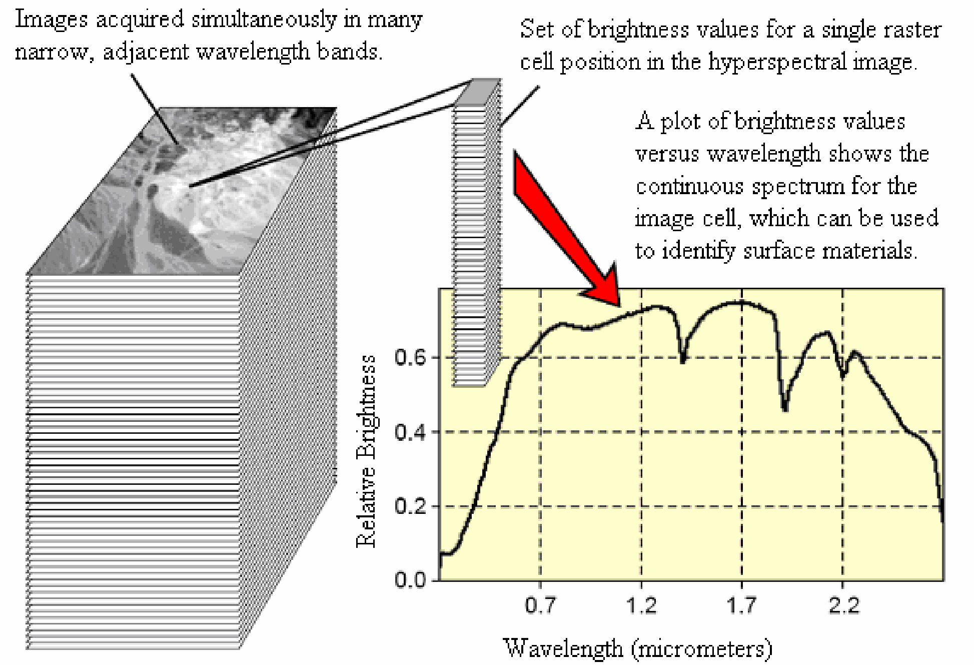

Images acquired simultaneously in many narrow, adjacent wavelength bands.
Set of brightness values for a single raster cell position in the hyperspectral image.
A plot of brightness values versus wavelength shows the continuous spectrum for the image cell, which can be used to identify surface materials.
Relative Brightness
0.0
0.2
0.4
0.6
0.7
1.2
1.7
2.2
Wavelength (micrometers)

Figure 2. Construction of a typical hyperspectral image.[21]

C. Spatial Domain Versus Spectral Domain

Given that HSI sensors generate images containing both spatial and spectral information, it is worthwhile to note some basic differences between image processing exclusively in either of the domains. With spatial domain processing, information is embedded in the spatial arrangement of pixels in a two-dimensional (2-D) image. With spectral domain processing, each pixel has a corresponding spectrum that can be used to identify materials in that ground resolution cell. Hence, spatial processing exploits geometrical shape information, while the spectral domain allows for processing one pixel at a time. Also, spatial domain processing requires very high spatial resolution to identify objects by their shape, whereas spectral domain processing only requires a single pixel since the spectral signature is used to identify the object. The advantages of spectral processing were evident in the early applications of remote sensing that involved mineral exploration, ground cover classification, and crop health characterization.[3] In these applications, morphological information is not useful since the various natural materials of interest do not have predetermined shapes. In practice, hyperspectral sensors represent a deliberate trade-off in which spatial resolution is compromised for improved spectral resolution. In fact, the data volume is greatly reduced since data volume increases with the square of spatial resolution, but only linearly with the number of spectral bands.[22]

D. The HSI Data Cube

Because both spectral and spatial information are obtained, HSI sensors provide a three-dimensional (3-D) data structure known as a *data cube*. If one extracts all the pixels in a single ground resolution cell and plots the spectral values as a function of wavelength, one obtains the spectral signature for that ground resolution cell. However, if one extracts all the pixels in the same spectral band, the result is a 2-D intensity image showing the spatial distribution of reflectance values of the scene for that particular wavelength. Figure 3 gives an example of a data cube.

The spectral signatures in either the radiance or reflectance domain can be viewed as a scattering of points in a K-dimensional Euclidean space, denoted by $\Re^K$, where K is the number of spectral bands. Hence, the spectral signature at each spatial location can be seen as a column vector $\boldsymbol{x} = [x_1, x_2, \ldots, x_K]^T$. To be physically meaningful, radiance or reflectance components can only take nonnegative values.

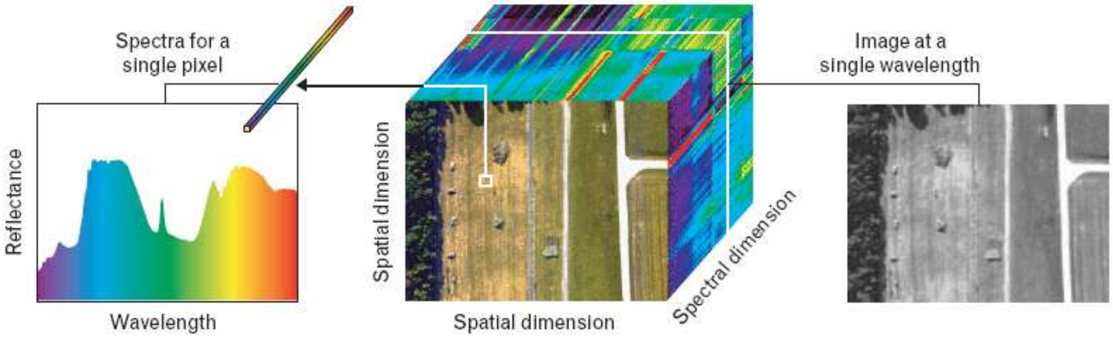


Figure 3. Data cube visualization showing spatial and spectral dimensions.[3]

E. Classification Versus Detection

In the remote sensing community, hyperspectral image processing has applications in *classification* and *target detection*. The goal of classification is to assign the pixels in an input scene to one of K classes. As with any pattern classification algorithm, there must be a sufficient amount of training data for each of the classes. In the context of classification, the natural criterion of performance is the minimization of the probability of misclassification errors.[2] The goal in target detection is to search all the pixels in the scene for the presence of a target. Target detection can be formulated as a binary classification problem, where each pixel is labeled as *target* or *background*. The *target* class will be sparsely populated since the targets that may be present in a scene will only account for a very small fraction of the total pixels. In contrast, the *background* class refers to all the nontarget pixels in the scene and represents all of the specific nontarget classes present in the scene.

F. Challenges in HSI Processing

1. Spectral variability

As discussed in the introduction, one of the significant challenges in HSI processing is spectral variability, which refers to the phenomenon that spectra observed from samples of the same material will never be identical. In other words, spectra of the same material are not fixed due to the inherent variations present in the material. Further spectral variability is introduced by external factors such as atmospheric conditions, sensor noise, and illumination variations.[2,3,20]

Interactions between incoming and reflected solar energy in the atmosphere can contribute to spectral variability. For certain wavelengths, these interactions reduce the amount of incoming energy reaching the ground and further reduce the amount of reflected energy reaching the hyperspectral sensor—a phenomenon known as *absorption*.[20] For example, the presence of water vapor and carbon dioxide causes a significant reduction in energy at 1.4 μm and 1.9 μm, rendering these bands useless. The effect of *scattering*, whereby light is scattered upward by the atmosphere, also contributes to variability by distorting the energy incident on the HSI sensor. Also, variations between sensors and temporal changes within sensors can introduce noise in the measurements, leading to variability.

For hyperspectral imaging, two factors related to illumination that lead to variability include illumination geometry and shadowing. The amount of energy reflected by an area on the ground depends on the amount of solar energy illuminating the area, which in turn depends on the *angle of incidence*: the angle between the path of the incoming energy and a line perpendicular to the ground surface.[20] The angle of incidence can vary as the sun's height changes with the time of day and season or with differences in the terrain surface (e.g., flat or sloped). Shadows in an area can also distort the reflected energy from a surface and thus decrease the amount and intensity of illumination for the affected pixels. Shadows can be caused by clouds or other nearby objects on the terrain.

2. Spectral mixing

Each pixel in a hyperspectral image corresponds to an area of the earth's surface as captured by the imaging spectrometer. The size of the area depends on the sensor design and its height above the earth's surface. This size, which can range from several meters to several kilometers, defines the *spatial resolution* of the hyperspectral image. If the size of a chunk of the imaged surface is large enough, it is possible for the spectrum of a target to be combined with the spectrum of the local background, leading to a *mixed pixel*. This is in contrast to a *pure pixel*, whose spectrum is determined by only one material, whether it is background or target. Hence, we have the distinction between a full pixel target whose spectrum completely occupies a pixel in an HSI image and a subpixel target whose spectrum occupies only a portion of the pixel. The challenge presented by subpixel targets justifies nonliteral exploitation of the HSI data because subpixel targets have no chance of being detected spatially but they do have a chance of being detected spectrally. While the detectability of full pixel targets depends on the spectral contrast between target and background, the detectability of subpixel targets is also dependent on the portion of the pixel occupied by the target (fill factor), because it determines the amount of background interference on the observed target spectrum.[23] Figure 4 shows the spectra of three materials where the third material's spectrum is a linear combination of the first two spectra.

3. Volume of data

The first two challenges that have been discussed are manifested during data acquisition and affect both multispectral and hyperspectral scenarios. As is the case, these challenges can significantly hinder classification and detection rates. A third problem in HSI is simply the volume of data present in hyperspectral datasets. In essence, this challenge manifests itself after the data has been acquired. Although the increase in spectral resolution introduced by HSI sensors can improve the rate of classification and detection, the sheer volume of data heavily increases the transmission bandwidth and the computational complexity of processing the data.[24] Furthermore, the abundance of data poses a constraint on real-time processing of HSI data in military detection applications. In military detection scenarios involving ground targets, real-time processing of HSI data is utilized for improving battlefield situation awareness, decreasing sensor-to-shooter cycle time, and providing timely battle damage assessments.[25]

To illustrate the magnitude of data involved in hyperspectral imagery, consider the following example. Suppose we have an HSI data cube with a spatial resolution of 512 x 512 pixels and a spectral resolution of 200 bands. Assuming that 8 bits are used to store the values of each band, this data cube is approximately 52 megabytes (MB). Storing, transmitting, and processing data of such magnitude for each hyperspectral data cube inevitably poses concerns for the designers of HSI classification and detection systems.

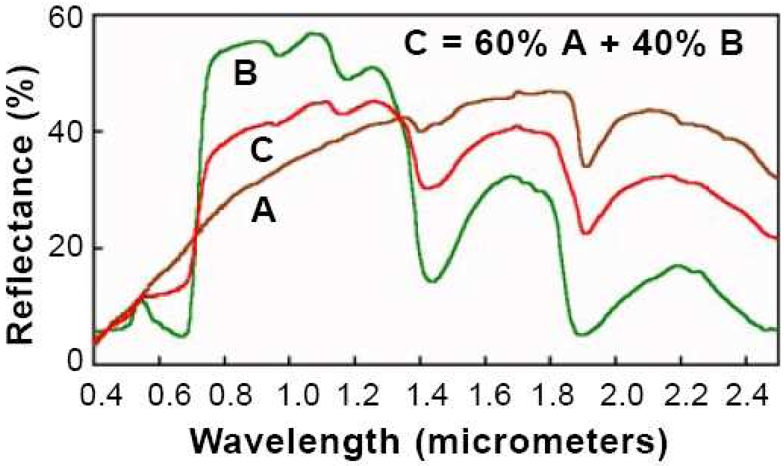


Figure 4. Illustration of spectral mixing.[21]

G. Types of Target Detection Algorithms

Spectral target detection algorithms can be broadly classified into *matching* algorithms and *anomaly detection* algorithms. In the spectral matching approach, the spectral signature of the target is required *a priori*, and the algorithm determines whether a match exists between the spectral signatures of the scene pixels and the target spectral signature. The target signature is usually obtained from a spectral library and is sometimes referred to as the *reference* signature. In anomaly detection scenarios, no *a priori* information is used; the algorithm identifies as a potential target, each pixel whose spectral signature does not fit a model of the local or global background.[23] Anomaly detection algorithms are more susceptible to false alarms and less reliable than matching algorithms since their performance is completely determined by the background model parameters.[26]

Hyperspectral target detection algorithms have been developed with *stochastic* and *deterministic* approaches. Stochastic approaches typically model the hyperspectral data with multivariate normal distributions and are the most prevalent in the literature because of their mathematical tractability and reasonable performance in practical scenarios. Deterministic approaches make no assumptions regarding the nature of the data and can also perform well, depending on the spectral contrast between target and background objects.

CHAPTER III

TRADITIONAL DETECTION ALGORITHMS

A. Stochastic Algorithms

Stochastic detection approaches must decide whether a target is present in a pixel based on its observed spectrum $\boldsymbol{x}$. Let $\boldsymbol{x} = [x(\lambda_1), x(\lambda_2), \ldots, x(\lambda_L)]$, where L is the number of spectral bands. In fact, $\boldsymbol{x}$ can be treated as a multidimensional random vector with a probability distribution where its number of dimensions equals the number of spectral bands. The binary hypothesis test for the observed spectrum $\boldsymbol{x}$ is as follows:

H_0: target absent (background),

H_1: target present (target),

where H_0 is commonly known as the *null* hypothesis and H_1 as the *alternative* hypothesis. The binary nature of the problem formulation can be viewed as a two-class classification problem in the context of pattern recognition where the two classes are *background* and *target*. In Bayesian decision theory, given knowledge of the conditional probability density functions (PDFs) of both classes, the reasonable decision rule would be to choose the class that is most probable or likely, given the observed spectrum $\boldsymbol{x}$.[27] This concept is quantified by the *likelihood ratio test* (LRT) $\Lambda(\boldsymbol{x})$, the ratio of the conditional PDFs of both classes:

$$\Lambda(\boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid H_1)}{p(\boldsymbol{x} \mid H_0)} \tag{3.1}$$

If $\Lambda(\boldsymbol{x})$ is larger than the threshold η, the pixel is declared to be a target, otherwise, it is declared background. For any given problem, the minimum probability of error is achieved by the LRT decision rule. This probability of error is called the *Bayes error rate* and is the best that any LRT-based classifier can achieve. Unfortunately, in practice, knowledge of the theoretical conditional class densities $p(\boldsymbol{x} \mid H_0)$ and $p(\boldsymbol{x} \mid H_1)$ is not available. Thus, stochastic approaches to detection typically involve the replacement of unknown density parameters with their maximum likelihood (ML) estimates which are obtained directly from the data.[23]

1. Quadratic Neyman-Pearson detector

The most prevalent stochastic detector in the literature operates under the assumption that the class conditional PDFs of the background and target classes follow multivariate normal distributions. In a hyperspectral image, an observed spectrum $\boldsymbol{x}$ follows a multivariate normal distribution with mean vector $\boldsymbol{\mu} = E(\boldsymbol{x})$ and covariance matrix $\boldsymbol{\Gamma} = E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]$ denoted by $\boldsymbol{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ if its PDF is given by the following:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{L/2} |\boldsymbol{\Gamma}|^{1/2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Gamma}^{-1} (\boldsymbol{x}-\boldsymbol{\mu})} \tag{3.2}$$

where $|\boldsymbol{\Gamma}|$ denotes the determinant of the covariance matrix $\boldsymbol{\Gamma}$ and the term $(\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Gamma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$ is known as the *Mahalanobis* distance[27]. Under the multivariate normal assumption of the background and target classes, the likelihood ratio may be expressed as follows:

$$\Lambda(\boldsymbol{x})=\frac{(2\pi)^{-L/2}\left|\boldsymbol{\Gamma}_t\right|^{-1/2}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_t)^T\boldsymbol{\Gamma}_t^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_t)}}{(2\pi)^{-L/2}\left|\boldsymbol{\Gamma}_b\right|^{-1/2}e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_b)^T\boldsymbol{\Gamma}_b^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_b)}} \tag{3.3}$$

where $\boldsymbol{\mu}_b$ and $\boldsymbol{\Gamma}_b$ and $\boldsymbol{\mu}_t$ and $\boldsymbol{\Gamma}_t$ denote the mean vectors and covariance matrices of the background and target classes, respectively. After simplifying the expression in equation (3.3) and rearranging terms, the likelihood ratio may be expressed as follows:

$$\Lambda(\boldsymbol{x})=\frac{\left|\boldsymbol{\Gamma}_b\right|^{1/2}}{\left|\boldsymbol{\Gamma}_t\right|^{1/2}}\exp\left[\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_b)^T\boldsymbol{\Gamma}_b^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_b)-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_t)^T\boldsymbol{\Gamma}_t^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_t)\right] \tag{3.4}$$

After taking the natural logarithm of both sides and eliminating constant terms, equation (3.4) may be simplified to yield the following quadratic Neyman-Pearson (NP) detector[23]:

$$D(\boldsymbol{x})=\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_b)^T\boldsymbol{\Gamma}_b^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_b)-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_t)^T\boldsymbol{\Gamma}_t^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_t) \tag{3.5}$$

which compares the Mahalanobis distances of the input spectrum $\boldsymbol{x}$ from the means of the two classes. The constant at the beginning of equation (3.5) is often omitted in the literature since it is only a scaling factor.

2. Matched filter

A further assumption imposed on the data leads to the widely known *matched filter* (MF). If one assumes that the background and target covariance matrices are identical, i.e., $\mathbf{\Gamma}_b = \mathbf{\Gamma}_t = \mathbf{\Gamma}$, the quadratic terms in equation (3.5) disappear, and the expression can be simplified to the following:

$$D(\boldsymbol{x}) = \kappa(\boldsymbol{\mu}_t - \boldsymbol{\mu}_b)^T \mathbf{\Gamma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_b), \tag{3.6}$$

where κ is a normalization constant. While the choice of κ does not alter the performance of the MF, it is typically chosen such that $D(\boldsymbol{\mu}_t) = 1$.[23] Thus, the MF detector of equation (3.6) can be expressed as follows:

$$D_{MF}(\mathbf{x}) = \frac{(\boldsymbol{\mu}_t - \boldsymbol{\mu}_b)^T \Gamma^{-1}(\mathbf{x} - \boldsymbol{\mu}_b)}{(\boldsymbol{\mu}_t - \boldsymbol{\mu}_b)^T \Gamma^{-1}(\boldsymbol{\mu}_t - \boldsymbol{\mu}_b)} \tag{3.7}$$

3. Adaptive matched filter

The matched filter detector of equation (3.7) is optimal in the Bayes sense when the target and background classes follow multivariate normal distributions with a common covariance matrix. While these constraints lead to ease of mathematical tractability, they are highly unlikely scenarios for practical HSI data.[2] Practically speaking, the true mean and covariance matrices are unknown and must be estimated from the actual data. Under the assumption of low-probability of occurrence of targets, we can compute the ML estimates of the background parameters from the entire HSI data cube. Let $\boldsymbol{x}(n), n = 1, 2, \ldots, N$ denote the N spectral vectors in an HSI data cube. The ML estimates of the mean vector and covariance matrix of the background are given as follows:

$$\overset{\wedge}{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}(n) \cong \overset{\wedge}{\boldsymbol{\mu}}_b \tag{3.8}$$

$$\hat{\boldsymbol{\Gamma}} = \frac{1}{N}\sum_{n=1}^{N}\left[\boldsymbol{x}(n)-\hat{\boldsymbol{\mu}}\right]\left[\boldsymbol{x}(n)-\hat{\boldsymbol{\mu}}\right]^{T} \cong \hat{\boldsymbol{\Gamma}}_{b} \tag{3.9}$$

Concerning the parameters of the target class, we typically use a target spectral signature $\mathbf{t}$ from a library in place of $\boldsymbol{\mu}_t$ since there is no training data to calculate $\boldsymbol{\mu}_t$. As stated before, the target and background classes share a common covariance matrix whose ML estimate is given by equation (3.9). The resulting modifications of the matched filter lead to the *adaptive matched filter* (AMF) given by the following expression:

$$D_{AMF}(\boldsymbol{x}) = \frac{(\boldsymbol{t}-\hat{\boldsymbol{\mu}}_b)^T\,\hat{\boldsymbol{\Gamma}}_b^{-1}\,(\boldsymbol{x}-\hat{\boldsymbol{\mu}}_b)}{(\boldsymbol{t}-\hat{\boldsymbol{\mu}}_b)^T\,\hat{\boldsymbol{\Gamma}}_b^{-1}\,(\boldsymbol{t}-\hat{\boldsymbol{\mu}}_b)} \tag{3.10}$$

4. Constrained energy minimization

Another popular stochastic detection algorithm, akin to the MF detector, is the *constrained energy minimization* (CEM) algorithm. Suppose we wish to design a linear filter of the form $\boldsymbol{y} = c^T\boldsymbol{x}$. The CEM filter operates by minimizing the total energy of $\boldsymbol{y}$ subject to the constraint that $\boldsymbol{y}(\boldsymbol{t}) = c^T\boldsymbol{t} = 1$. The total energy of an HSI data cube is given by the following:

$$\begin{aligned} E &= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}^2(n) \\ &= c^T\left[\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}(n)\boldsymbol{x}^T(n)\right]c \\ &= c^T\,\hat{\mathbf{R}}\,c, \end{aligned} \tag{3.11}$$

where $\mathbf{R}$ is the sample correlation matrix of the data cube. The solution of this optimization problem leads to the following CEM detector:[3]

$$D_{CEM}(\boldsymbol{x}) = \frac{\boldsymbol{t}^T \hat{\mathbf{R}}^{-1} \boldsymbol{x}}{\boldsymbol{t}^T \hat{\mathbf{R}}^{-1} \boldsymbol{t}} \qquad (3.12)$$

Note that the CEM detector of equation (3.12) becomes the AMF detector of equation (3.10) if we remove the mean of the data from the target and input signatures and use the covariance matrix of the data instead of the correlation matrix. Unlike the NP and MF detectors, the CEM detector is not optimum in any Bayes sense since it is not derived using a hypothesis test.[23]

B. Deterministic Algorithms

1. Spectral angle mapper

One of the earliest and most basic algorithms for deterministic HSI detection is known as the spectral angle mapper (SAM) algorithm. Let $\boldsymbol{t}$ denote the target spectral signature and $\boldsymbol{x}$ denote the input spectral signature. The SAM metric measures the cosine of the angle between these two vectors and is defined by the following:

$$D_{SAM}(\boldsymbol{x},\boldsymbol{t}) = \frac{\langle \boldsymbol{x},\boldsymbol{t} \rangle}{\|\boldsymbol{x}\| \|\boldsymbol{t}\|}, \qquad (3.13)$$

where $\langle \cdot,\cdot \rangle$ is the dot product operator, and $\|\cdot\|$ is the Euclidean length, or L^2 norm of a vector. The values for D_{SAM} will range between zero and one since all the components of the spectral vectors are positive. While the SAM algorithm is very convenient and straightforward in target detection applications, it only performs well for targets having well separated distributions with small dispersions.[2]

2. Spectral information divergence

Spectral information divergence (SID) is an information-theoretic measure which compares the similarity between two hyperspectral signatures by measuring the probabilistic discrepancy between them.[28] It models each pixel signature as a multi-dimensional random variable and creates an associated probability distribution for the signature by normalizing its spectral histogram to unity. Let $\boldsymbol{x} = [x_1, x_2, \ldots, x_K]^T$ be a hyperspectral signature with probability vector $\boldsymbol{p} = [p_1, p_2, \ldots, p_K]^T$ where $p_j = x_j / \sum_{l=1}^{K} x_l$. SID uses the *Kullback-Leibler* information, or *relative entropy*, measure between two hyperspectral signatures $\boldsymbol{x}$ and $\boldsymbol{y}$ with corresponding probability vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ given by the following:

$$D(\boldsymbol{x} \| \boldsymbol{y}) = \sum_{l=1}^{K} p_l \log\left(\frac{p_l}{q_l}\right) \tag{3.14}$$

Using the relative entropy measure defined by equation (3.14), the SID between two signatures $\boldsymbol{x}$ and $\boldsymbol{y}$ is defined by the following:

$$SID(\boldsymbol{x}, \boldsymbol{y}) = D(\boldsymbol{x} \| \boldsymbol{y}) + D(\boldsymbol{y} \| \boldsymbol{x}) \tag{3.15}$$

Hence, SID uses the relative entropy between two signatures to capture the spectral information provided by each pixel. Experimental results show that SID can characterize spectral variability more effectively than the SAM metric.[29]

CHAPTER IV

EXPERIMENTAL DATA AND TARGET VARIABILITY MODELS

C. Raw HSI Data

Two of the data cubes in our experiments have been acquired using a CASI[30] sensor that produces 36 spectral bands ranging from 433 nm to 965 nm with a spectral resolution of 15 nm. The first data cube is vegetative scenery with a spatial resolution of 200 x 200, and the second data cube is urban scenery with a spatial resolution of 200 x 200. Visual range images of both scenes are shown in Figure 5 and Figure 6, respectively.

Two other data cubes have also been acquired using a HYDICE[31] sensor. This sensor operates in the Visible to Near Infrared (VNIR) and Short-Wave Infrared (SWIR) range of 400 nm to 2500 nm and is sampled to yield 203 spectral bands. As before, we have both vegetative and urban scenes with spatial resolutions of 200 x 200 whose visual range images are shown in Figure 7 and Figure 8, respectively. The low SNR and water absorption bands have been removed, leaving a total of 193 bands and 172 bands for the vegetative and urban scenes, respectively.

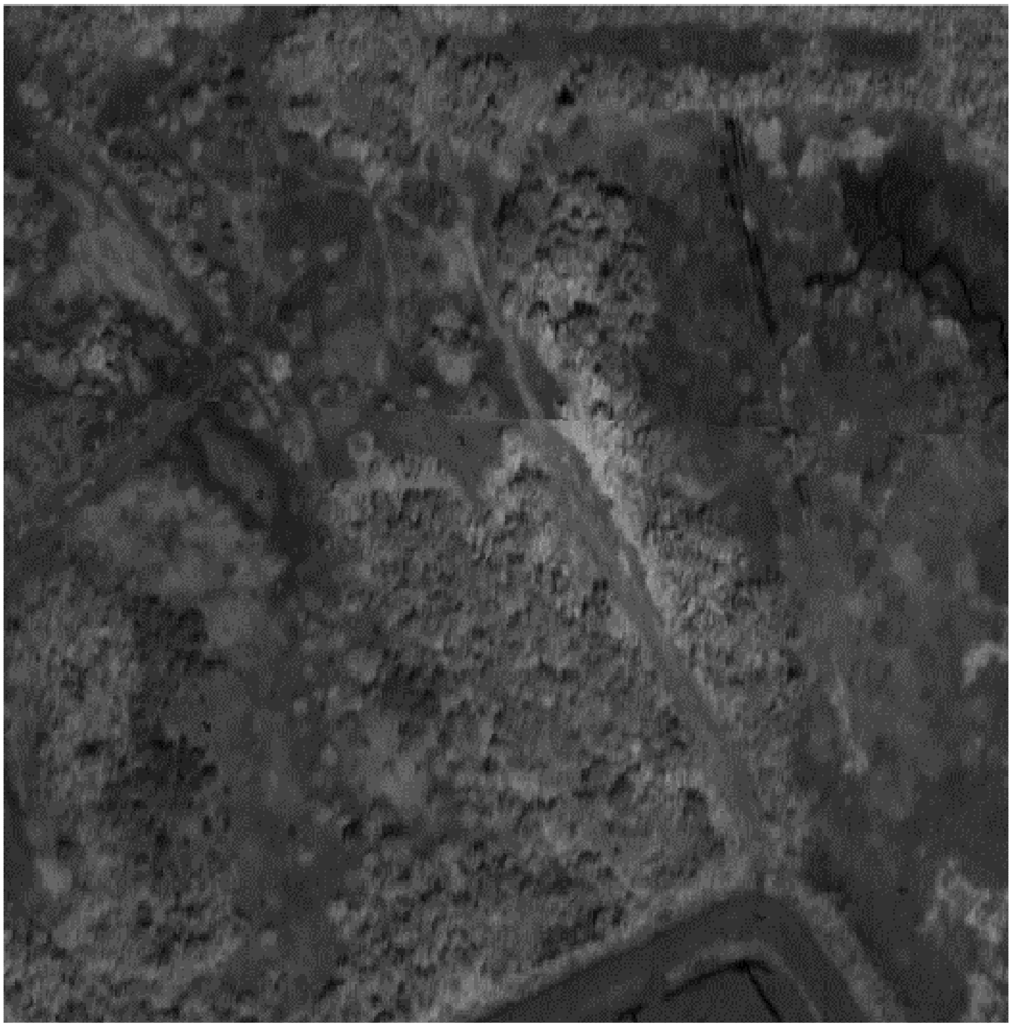


Figure 5. CASI vegetative scenery.

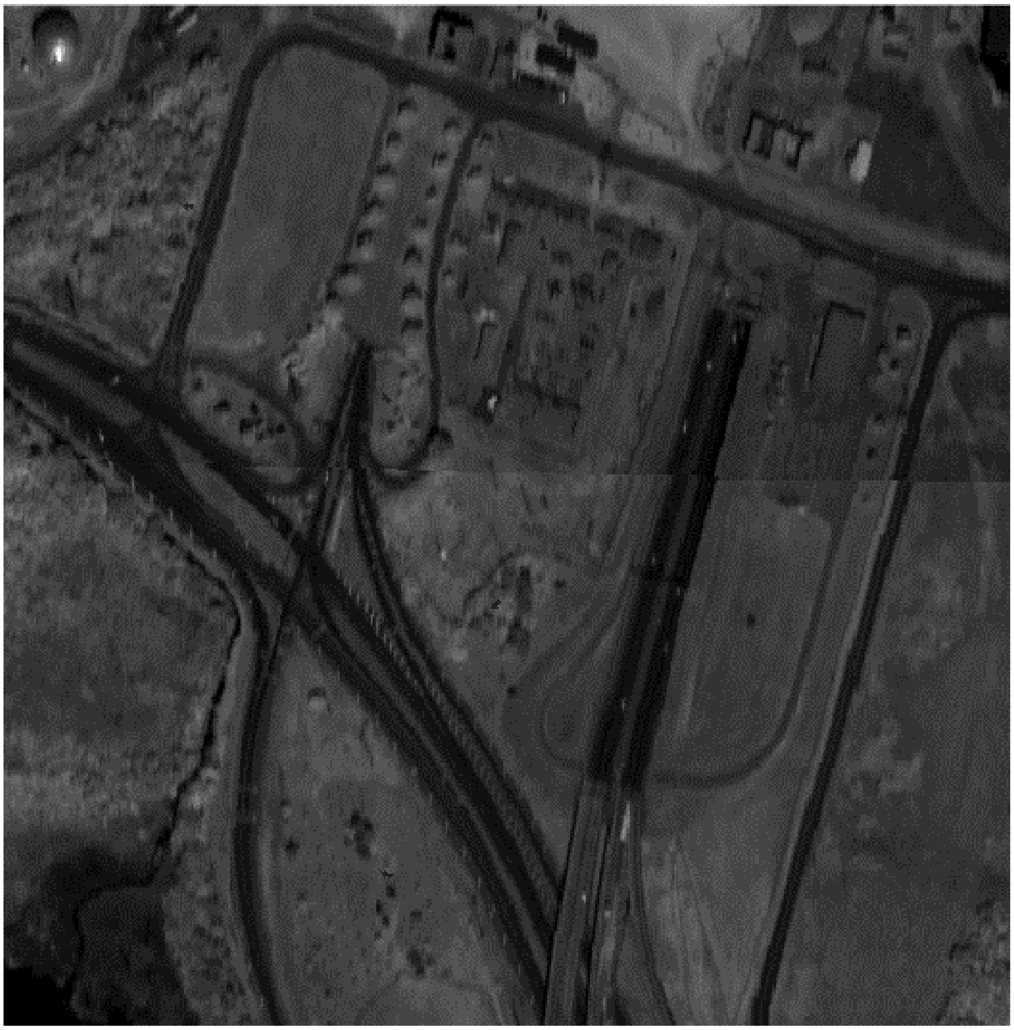


Figure 6. CASI urban scenery.

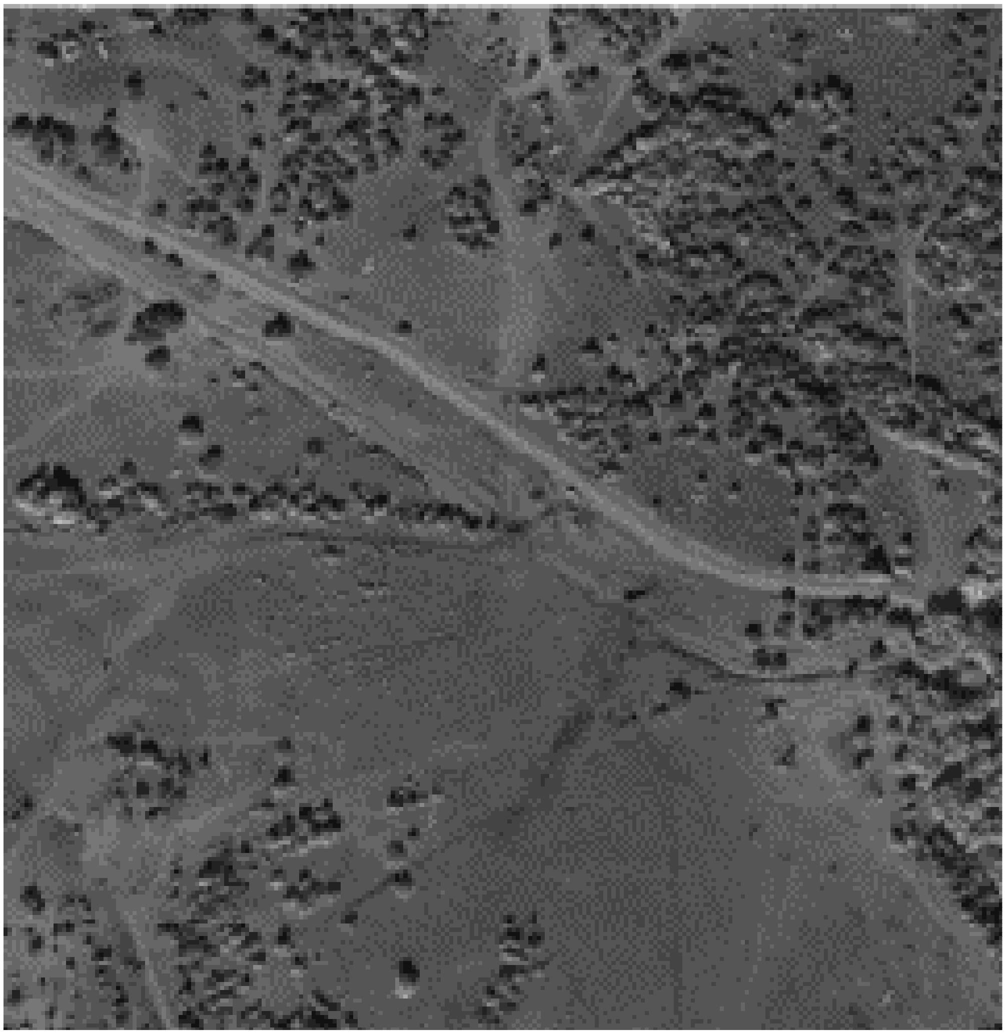


Figure 7. HYDICE vegetative scenery.

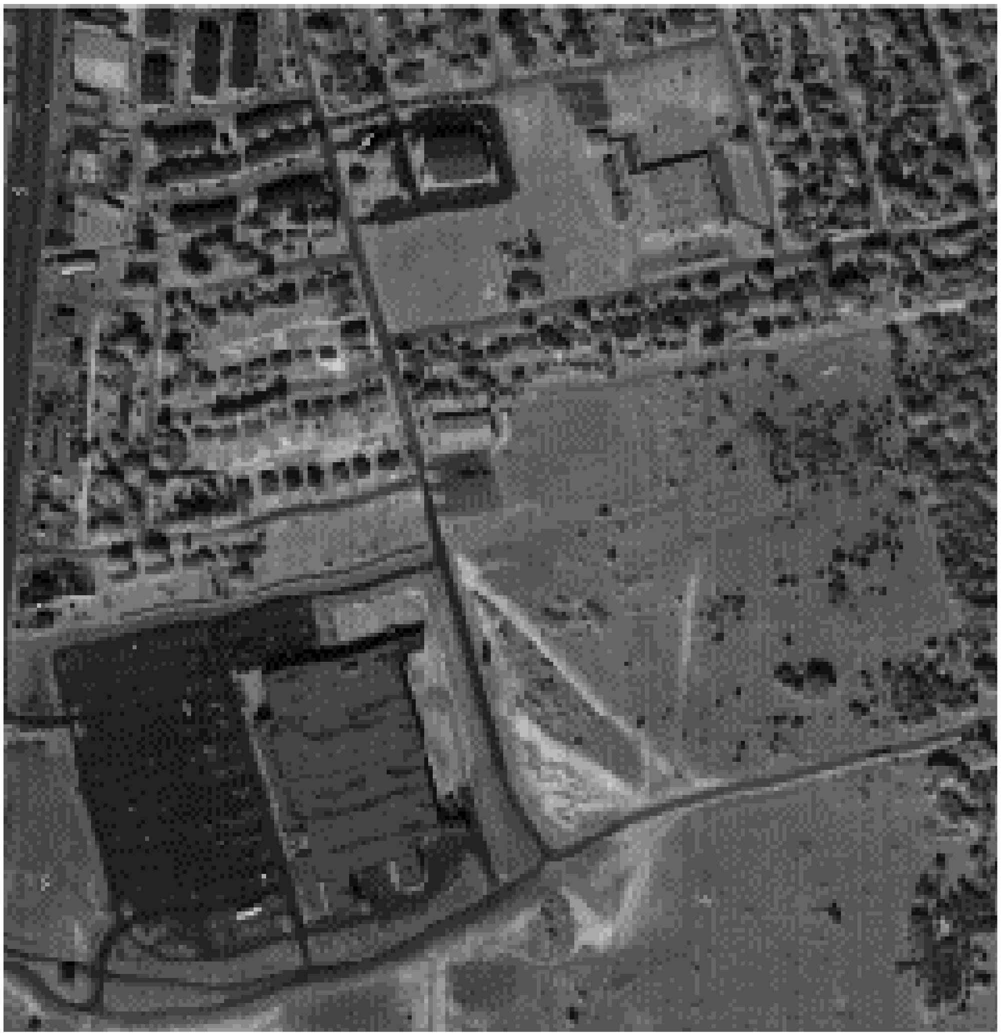


Figure 8. HYDICE urban scenery.

D. Synthetic Data Generation

In practice, it is difficult to obtain ground-truthed HSI imagery that contains multiple real targets in low-probability target detection scenarios. The alternative is to insert simulated targets in real HSI imagery. For the purposes of testing the proposed work in this dissertation, we have randomly selected 200 pixels from each of the scenes and replaced their signatures with "corrupted" spectral signatures of a particular target material. The randomly selected locations of the targets are fixed for all four scenes. Figure 9 provides the truth mask showing the locations of the inserted targets. Note that 200 pixels equals only half a percent of the 40000 total scene pixels in each data cube, thus simulating a low-probability target detection scenario.

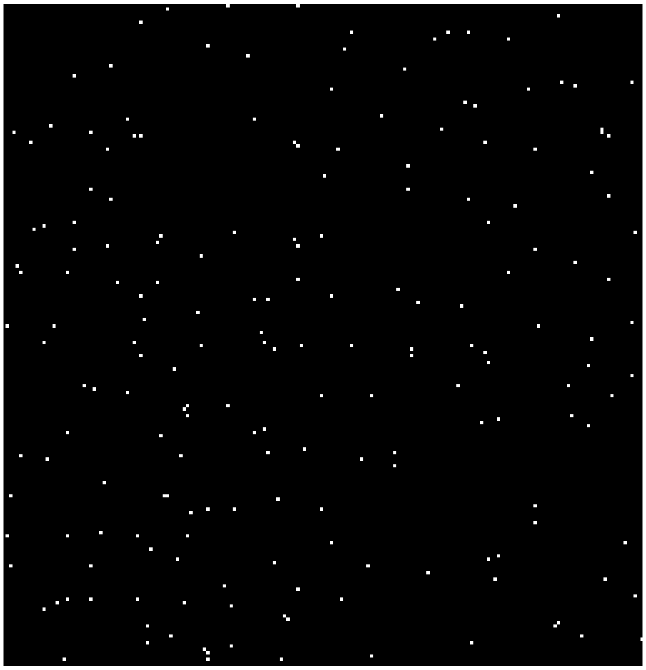


Figure 9. Truth mask showing locations of inserted targets.

How are these targets generated? In this dissertation, we have investigated the use of two unique models for generating the targets. These models will be discussed in the next section.

To account for the possibility of mixed pixel spectra corresponding to the boundaries of targets in the scene, twenty (10%) of the 200 signatures have been mixed in abundance ratios from 50% to 95% using the *linear mixing model* (LMM). According to the LMM, a mixed pixel spectrum is represented as a linear combination of endmember spectra. The weight, or fractional abundance, of each component endmember spectrum is proportional to the area that the endmember covers in a mixed pixel. For a hyperspectral scene with L spectral bands, the scene signatures can be represented as L-dimensional feature vectors. The general LMM equation is given by the following:[32]

$$\boldsymbol{x} = \sum_{k=1}^{M} \tau_k \boldsymbol{s}_k + \boldsymbol{w} = \boldsymbol{S\tau} + \boldsymbol{w}, \tag{4.1}$$

where $\boldsymbol{x}$ is a mixed pixel spectrum, $\boldsymbol{s}_k$, $k = 1, \ldots, M$, are the endmember spectra, τ_k are the fractional abundances of the endmember spectra, M is the number of endmembers, and $\boldsymbol{w}$ is an L-dimensional error vector that accounts for either noise or model-error. To make the LMM physically meaningful, the nonnegativity and sum-to-one additivity constraints are imposed on the endmember abundances:

$$\tau_k \geq 0\,(nonnegativity), \sum_{k=1}^{M} \tau_k = 1(additivity) \tag{4.2}$$

The LMM in (4.1) with the constraints in (4.2) is known as the *constrained linear mixing model* (CLMM).

E. Simple Gaussian Model

In the first model that we have used for the generation of inserted scene targets, the signatures have been corrupted by drawing them from a multivariate Gaussian distribution and are of the following form:

$$\boldsymbol{y}_c = \boldsymbol{t} + \mathbf{n} \sim N_K[\boldsymbol{t}, \sigma^2 \mathbf{I}] \tag{4.3}$$

where $\boldsymbol{y}_c$ is a corrupted target signature, $\boldsymbol{t}$ is the pure target signature, $\mathbf{n}$ is additive stochastic noise, and σ^2 is the per-band variance. The notation $\boldsymbol{x} \sim N_K[\boldsymbol{\mu}, \boldsymbol{\Gamma}]$ denotes a K-dimensional Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Gamma}$. By inserting the corrupted signatures that have been generated in this fashion, we have introduced spectral variability into the scenes. The variance σ^2 that has been used in our experiments has been varied to achieve SNRs of 8 dB, 10 dB, 12 dB, and 15 dB. The SNR is defined here as the root mean square (RMS) of the pure target signature divided by the standard deviation of the noise and is given by the following:

$$SNR = \frac{\sqrt{\frac{1}{K}\sum_{i=1}^{K} \boldsymbol{t}_i^2}}{\sigma} \tag{4.4}$$

Thus, for each original data cube, four data cubes have been generated with each containing corrupted targets with a specific SNR. Hence, we have introduced differing levels of spectral variability ranging from light to heavy variability—the SNR decreases as the variability increases since the SNR is inversely proportional to σ. In future chapters, the data cubes will be referenced by the acquisition sensor, the type of scenery, and the SNR of the targets that have been inserted into them. For instance, for the vegetative scene acquired using the CASI sensor where we have inserted targets with a SNR of 10 dB, we will refer to this data cube as *CASI_veg_10*. As another example, for the urban scene acquired using the HYDICE sensor where we have inserted targets with a SNR of 10 dB, we will refer to this data cube as *HYDICE_urban_10*. Figure 10 – Figure 25 show the differing levels of spectral variability of the targets that have been inserted into the scenes using the simple Gaussian model. In each of the figures, the pure target signature is shown in black, while the generated target signatures are shown in blue. The pure target signatures used in the CASI data cubes belong to a particular material, while the pure target signatures used in the HYDICE data cubes belong to another material.

F. Adaptive Gaussian Model based on First-order Markov Model

In the simple model introduced above, the covariance matrix in (4.3) is diagonal, thus ignoring any band-to-band correlation. The second model that we will use in this dissertation will also be based on a multivariate Gaussian distribution; however, the covariance matrix $\mathbf{\Gamma}$ is not diagonal. The signatures in this model are of the following form:

$$\boldsymbol{y}_c = \boldsymbol{t} + \mathbf{n} \sim N_K[\boldsymbol{t}, \mathbf{\Gamma}] \tag{4.5}$$

where $\boldsymbol{y}_c$ is a corrupted target signature, $\boldsymbol{t}$ is the pure target signature, $\mathbf{n}$ is additive stochastic noise, and $\mathbf{\Gamma}$ is the covariance matrix.

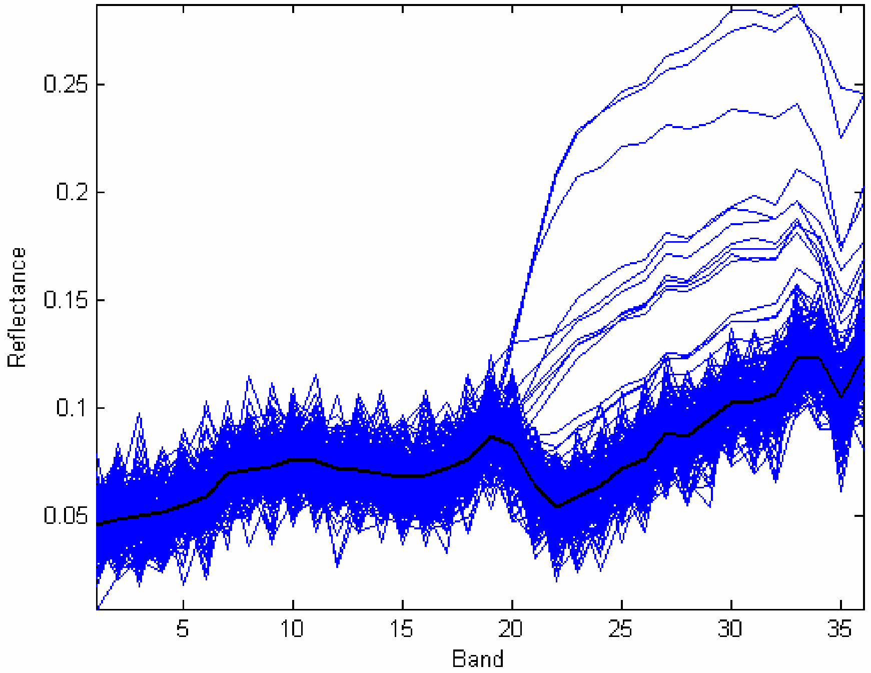


Figure 10. ***CASI_veg_8*** **scene targets generated using simple Gaussian model.**

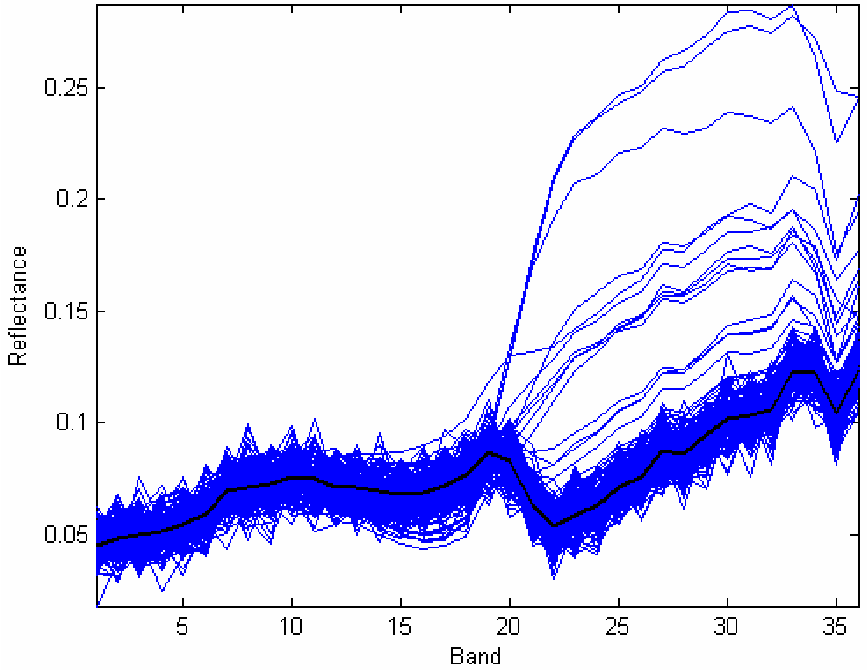


Figure 11. ***CASI_veg_10*** **scene targets generated using simple Gaussian model.**

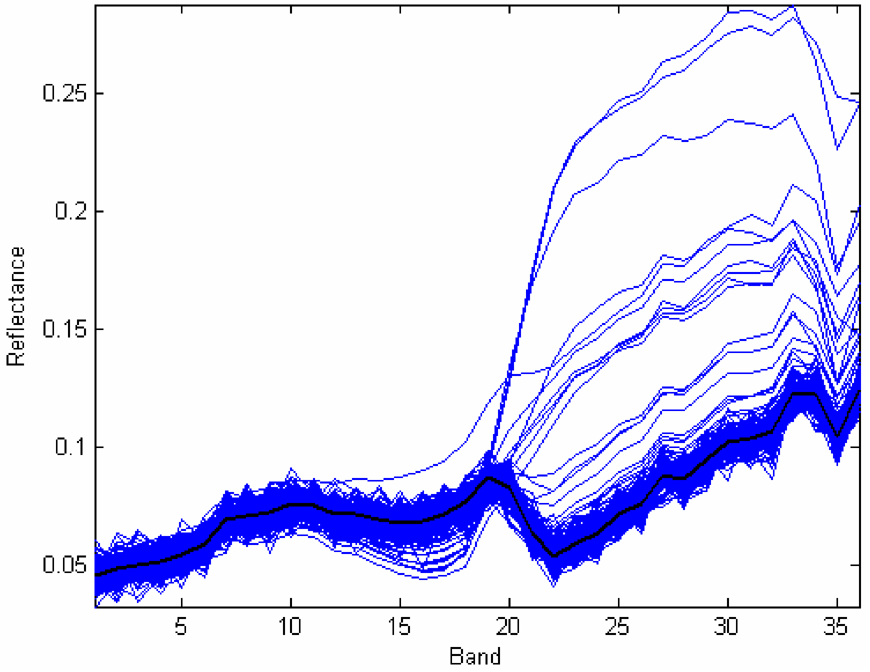


Figure 12. *CASI_veg_12* scene targets generated using simple Gaussian model.

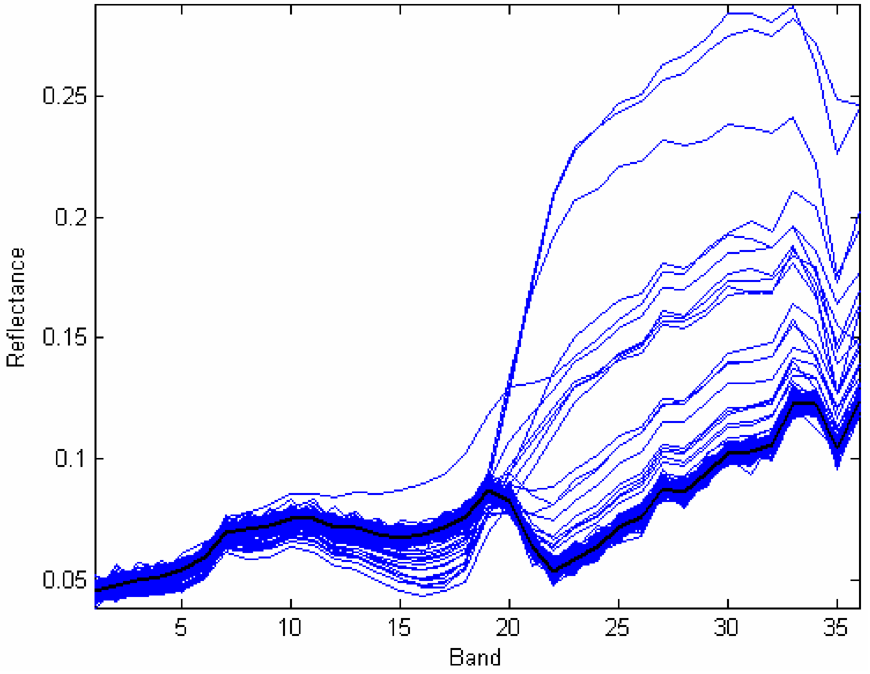


Figure 13. *CASI_veg_15* scene targets generated using simple Gaussian model.

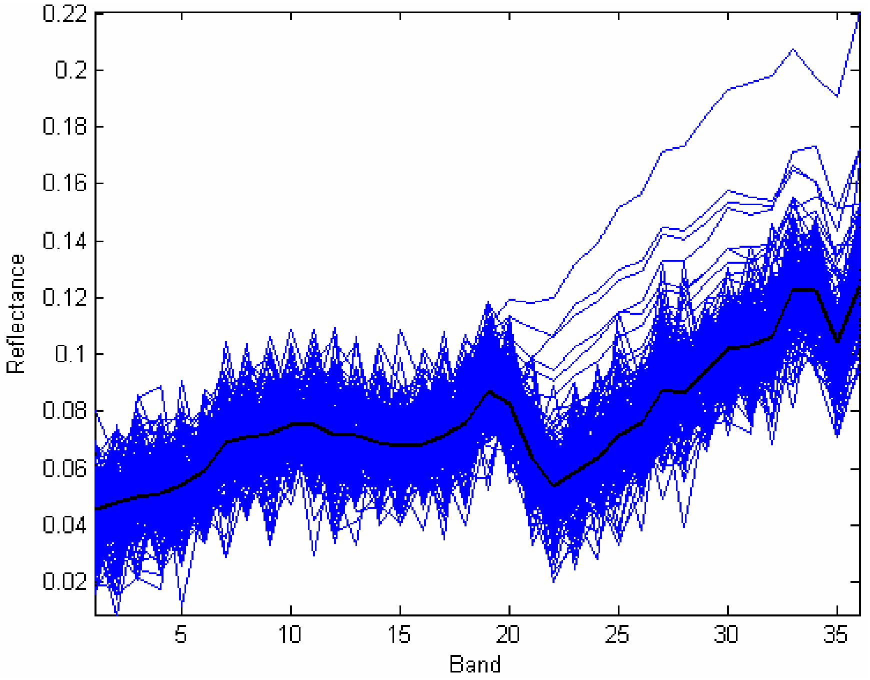


Figure 14. ***CASI_urban_8*** **scene targets generated using simple Gaussian model.**

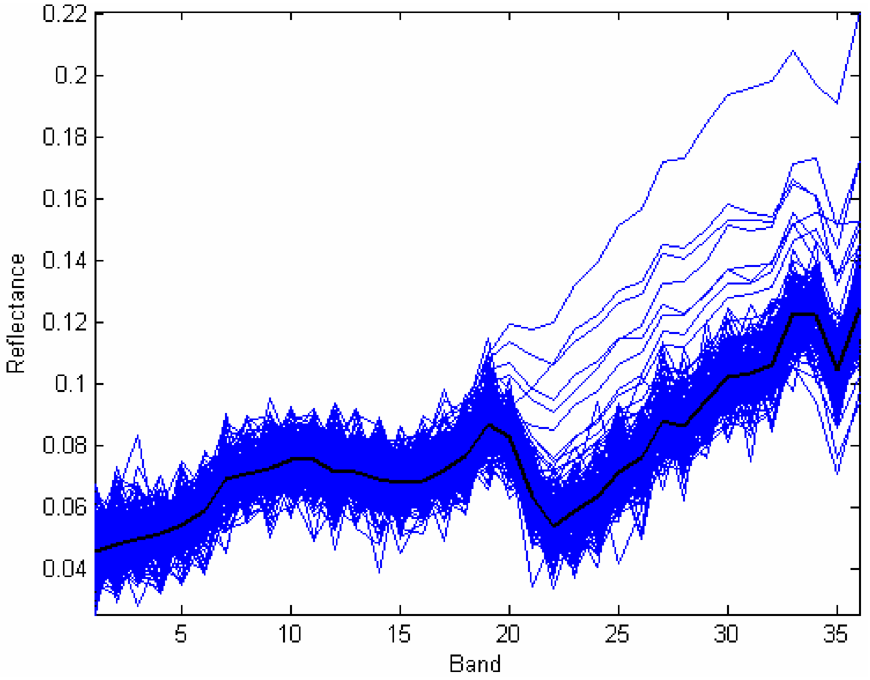


Figure 15. ***CASI_urban_10*** **scene targets generated using simple Gaussian model.**

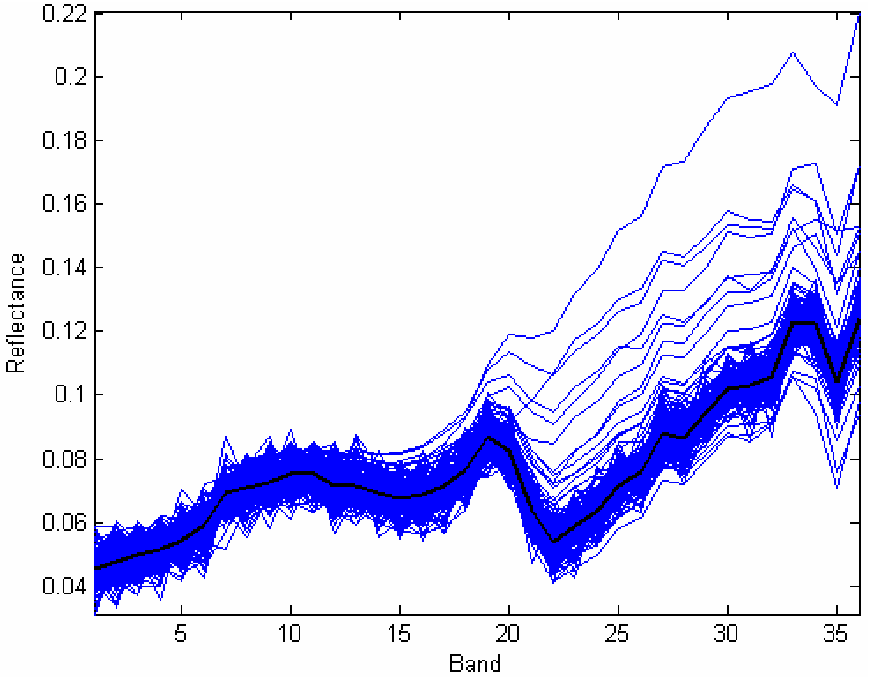


Figure 16. ***CASI_urban_12*** **scene targets generated using simple Gaussian model.**

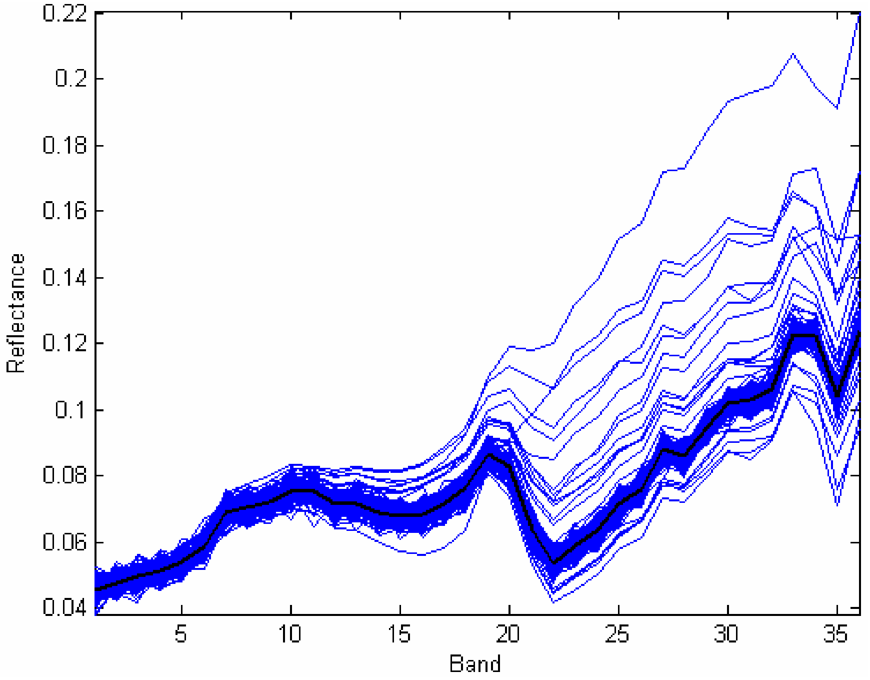


Figure 17. ***CASI_urban_15*** **scene targets generated using simple Gaussian model.**

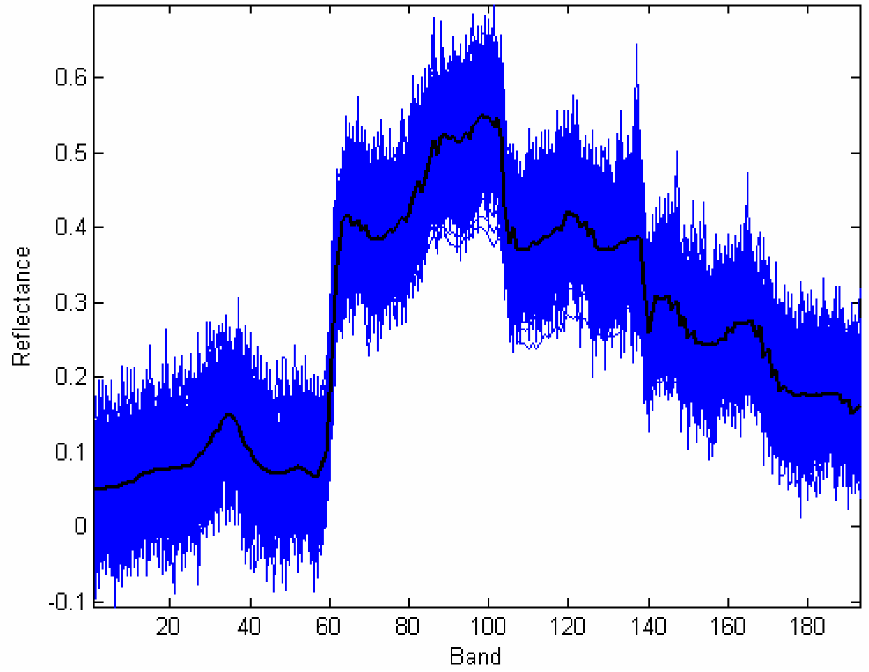


Figure 18. ***HYDICE_veg_8*** **scene targets generated using simple Gaussian model.**

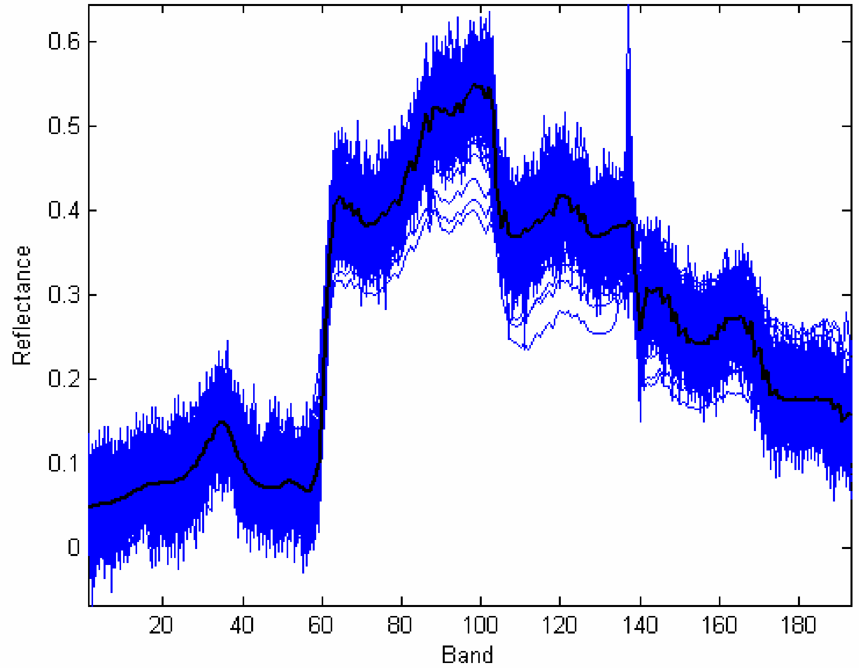


Figure 19. ***HYDICE_veg_10*** **scene targets generated using simple Gaussian model.**

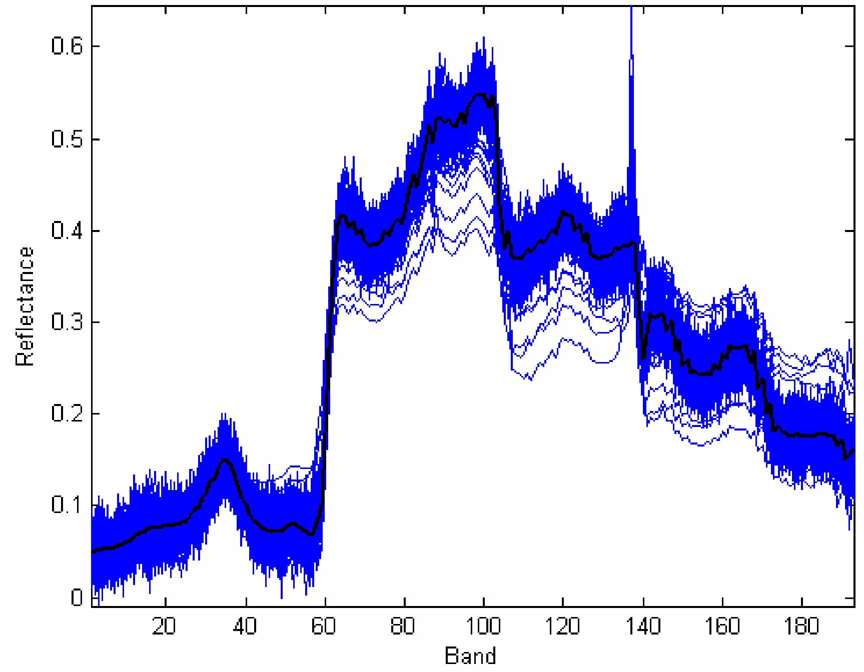


Figure 20. ***HYDICE_veg_12*** **scene targets generated using simple Gaussian model.**

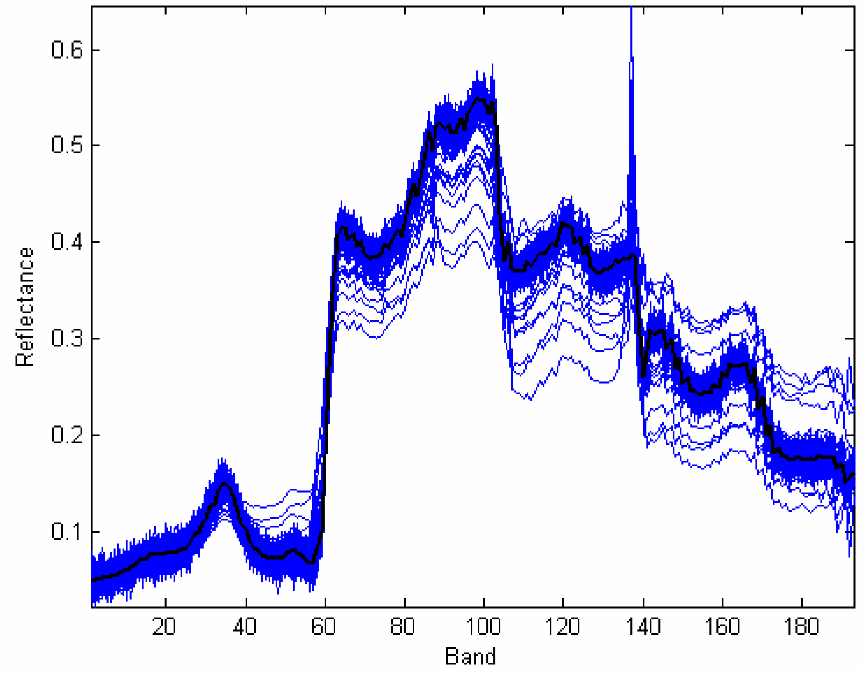


Figure 21. ***HYDICE_veg_15*** **scene targets generated using simple Gaussian model.**

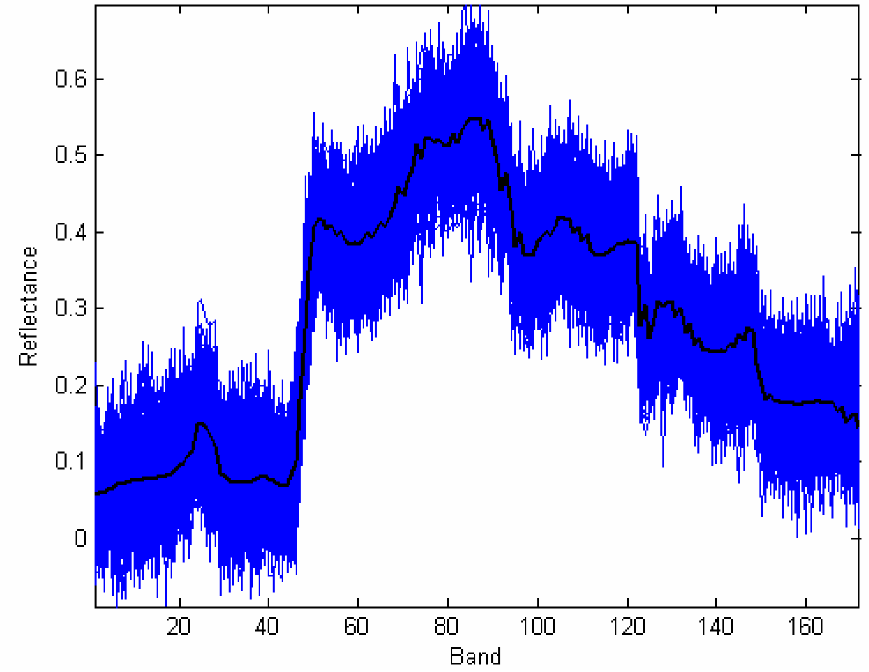


Figure 22. ***HYDICE_urban_8*** **scene targets generated using simple Gaussian model.**

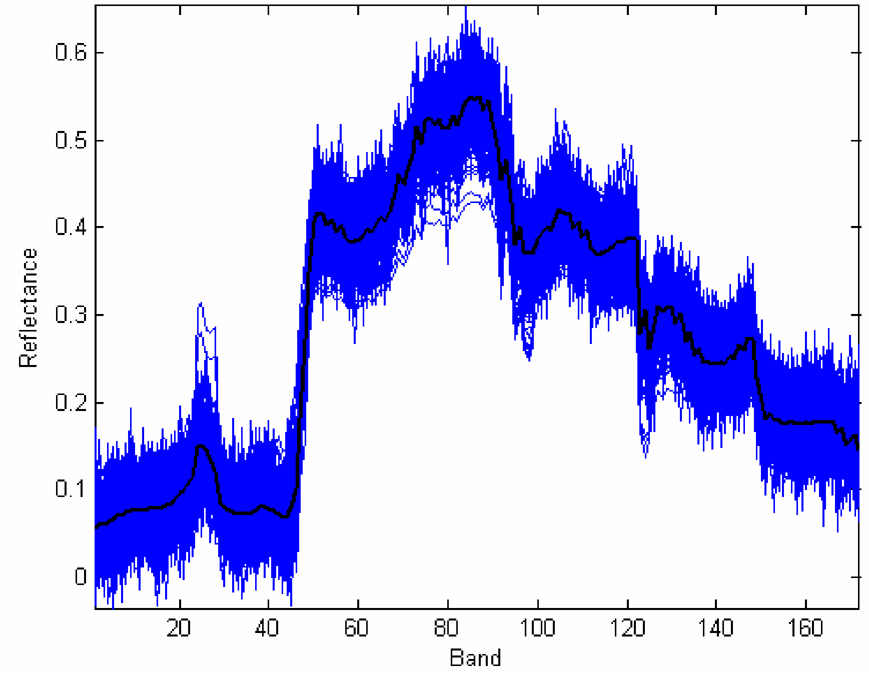


Figure 23. ***HYDICE_urban_10*** **scene targets generated using simple Gaussian model.**

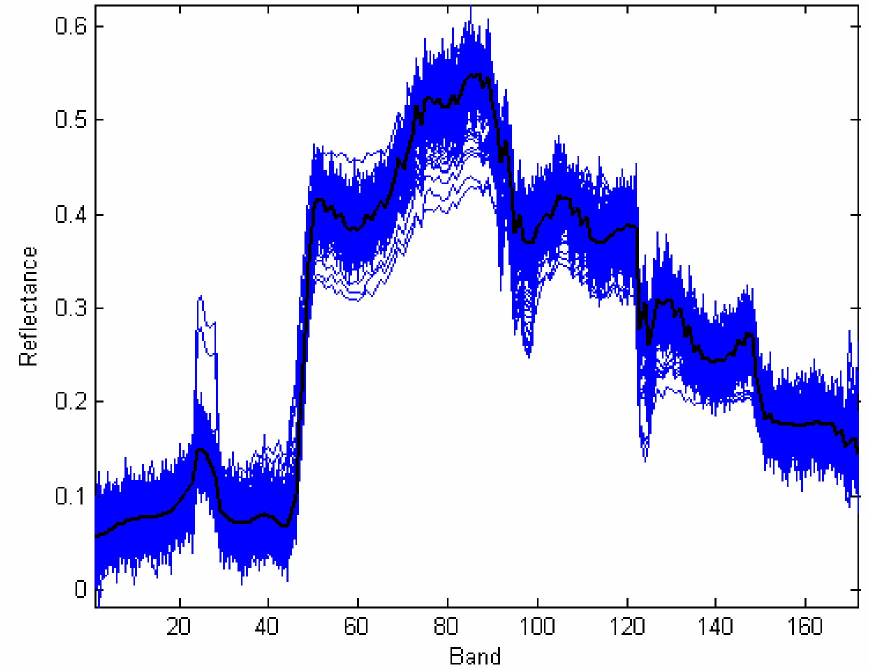


Figure 24. ***HYDICE_urban_12*** **scene targets generated using simple Gaussian model.**

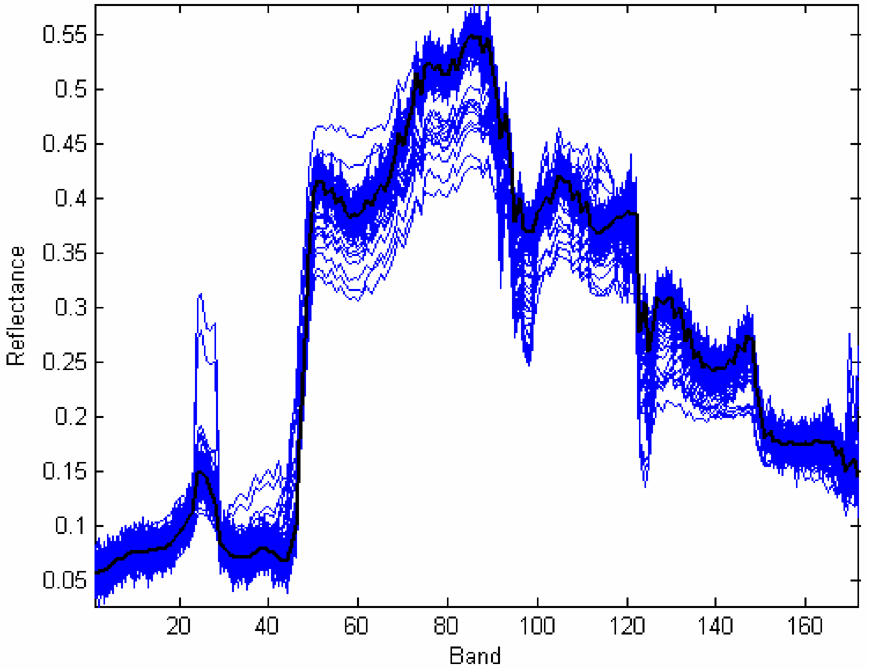


Figure 25. ***HYDICE_urban_15*** **scene targets generated using simple Gaussian model.**

In this model, we present a novel application of Markov theory to the generation of target signatures. A random sequence $\boldsymbol{x(n)}$ is called a *p*-order Markov sequence if the conditional probability of $\boldsymbol{x(n)}$ given the entire past is equal to the conditional probability of the last *p* samples. This can be stated more formally as follows:

$$P[x(n)\,|\,x(n-1),x(n-2),...]=P[x(n)\,|\,x(n-1),...,x(n-p)],\forall n \tag{4.6}$$

To exploit spectral band-to-band correlation, we will use a first-order Markov model to generate the covariance matrix $\boldsymbol{\Gamma}$ in (4.5). The pure target signature $\boldsymbol{t}$ is treated as a first-order stationary Markov sequence. According to this model, the covariance matrix is defined as follows:

$$\boldsymbol{\Gamma}=\sigma^2\cdot\mathbf{R} \tag{4.7}$$

where **R** is the Toeplitz correlation matrix defined according to the first-order Markov model:[33]

$$\mathbf{R}=\begin{bmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{K-1} \\ \rho & 1 & \rho & \rho^2 & \vdots \\ \rho^2 & \rho & 1 & \rho & \rho^2 \\ \vdots & \rho^2 & \rho & 1 & \rho \\ \rho^{K-1} & \cdots & \rho^2 & \rho & 1 \end{bmatrix} \tag{4.8}$$

In (4.7), the constant σ^2 is the per-band variance that is introduced to control the level of variability in the generated target signatures as we did in (4.3). As with the first model, we have varied σ^2 to yield SNRs of 8 dB, 10 dB, 12 dB, and 15 dB.

In (4.8), the constant ρ represents the first-order correlation coefficient. In our work, we have estimated ρ using the correlation coefficients between adjacent bands of the scene. This is done in the following manner:

1. Compute the correlation coefficient matrix of the data cube. This yields a K x K matrix, where K is the number of spectral bands.
2. Generate a vector **r** that contains the correlation coefficients between band *j* and the adjacent band *j+1*. Hence, the vector has K-1 components.
3. Estimate ρ by computing the mean of **r**.

Intuitively speaking, the reasoning behind using a first-order Markov model to generate the covariance matrix is that it is a reasonably safe assumption that the reflectance values at band *j* will closely resemble the reflectance value at *j-1* because of the increased spectral resolution of the HSI sensor. Because the only information we have concerning the target class is the pure target signature $\boldsymbol{t}$, we have adopted an adaptive approach that uses the scene to estimate ρ. Figure 26 – Figure 41 show the differing levels of spectral variability of the targets that have been inserted into the scenes using the first-order Markov-based Gaussian model. Notice how the generated signatures in this model are “smoother” than those generated according to the simple model in (4.3). Once again, this is due to the presence of band-to-band correlation using the non-diagonal correlation matrix structure given in (4.8).

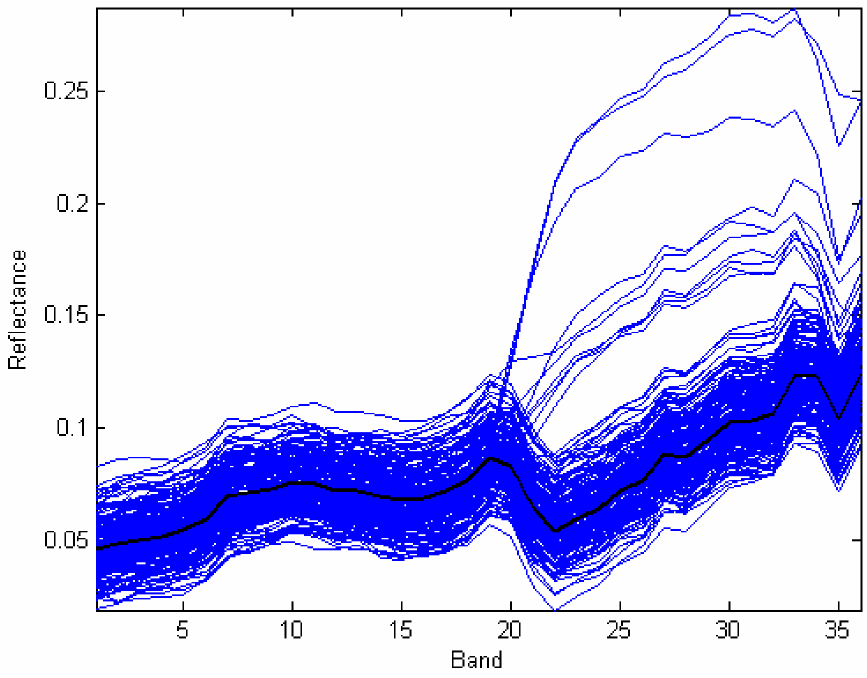


Figure 26. ***CASI_veg_8*** **scene targets generated using adaptive Gaussian model.**

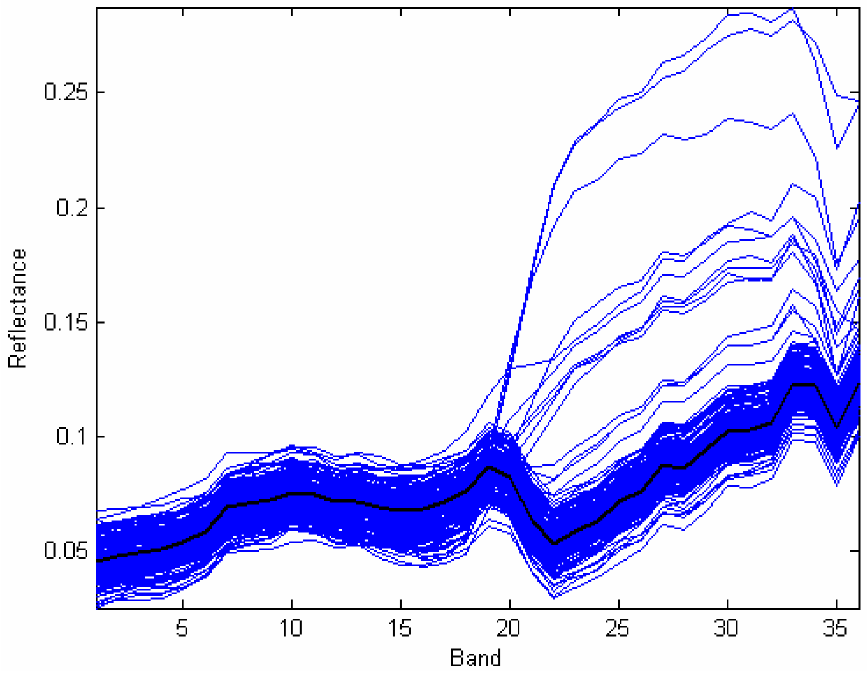


Figure 27. ***CASI_veg_10*** **scene targets generated using adaptive Gaussian model.**

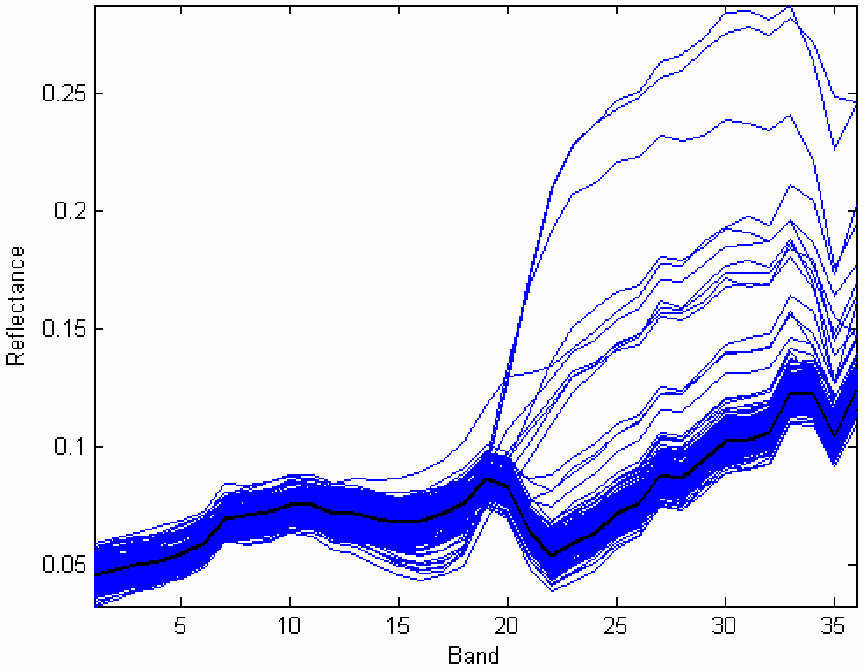


Figure 28. *CASI_veg_12* scene targets generated using adaptive Gaussian model.

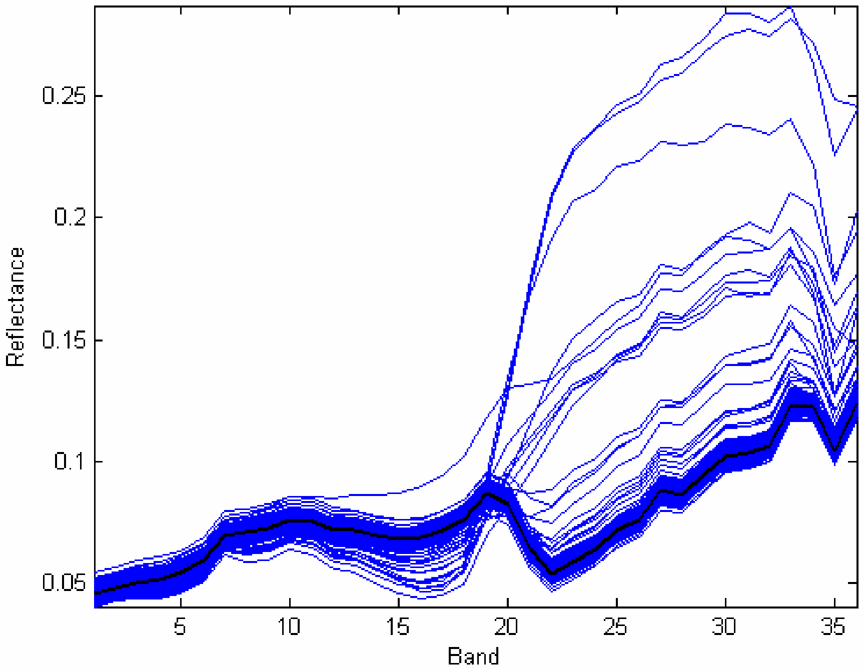


Figure 29. *CASI_veg_15* scene targets generated using adaptive Gaussian model.

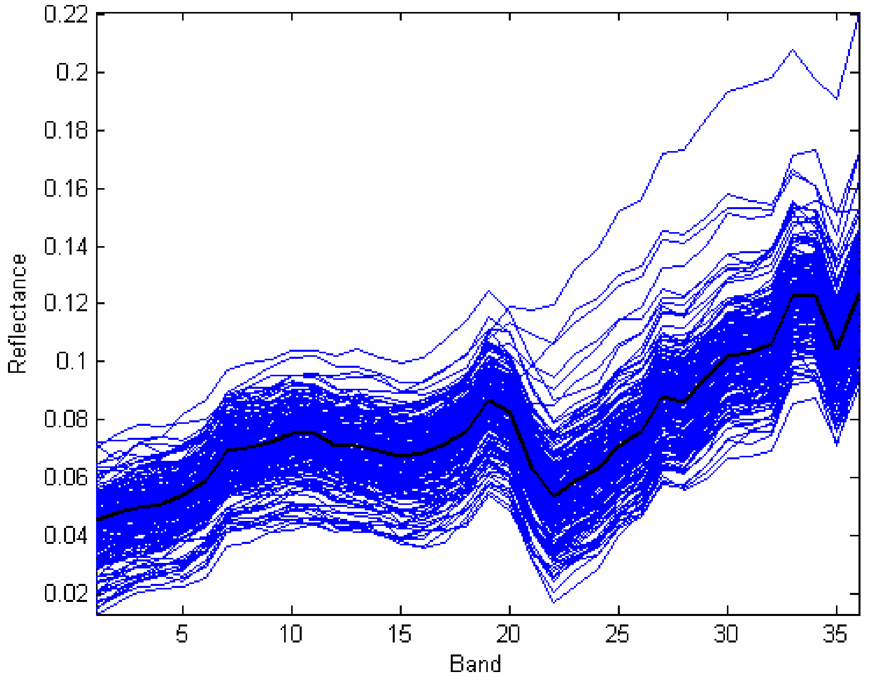


Figure 30. ***CASI_urban_8*** **scene targets generated using adaptive Gaussian model.**

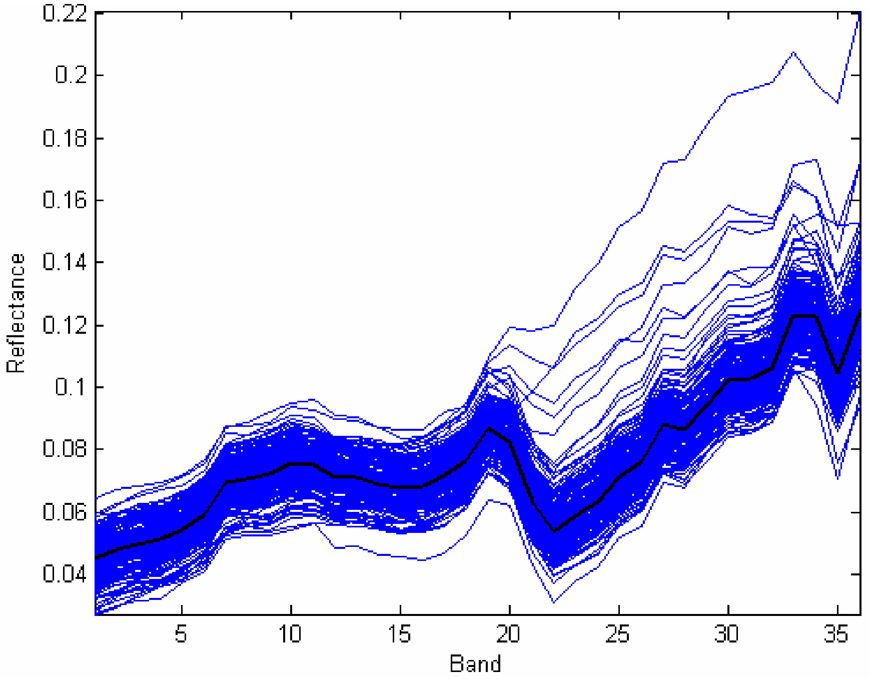


Figure 31. ***CASI_urban_10*** **scene targets generated using adaptive Gaussian model.**

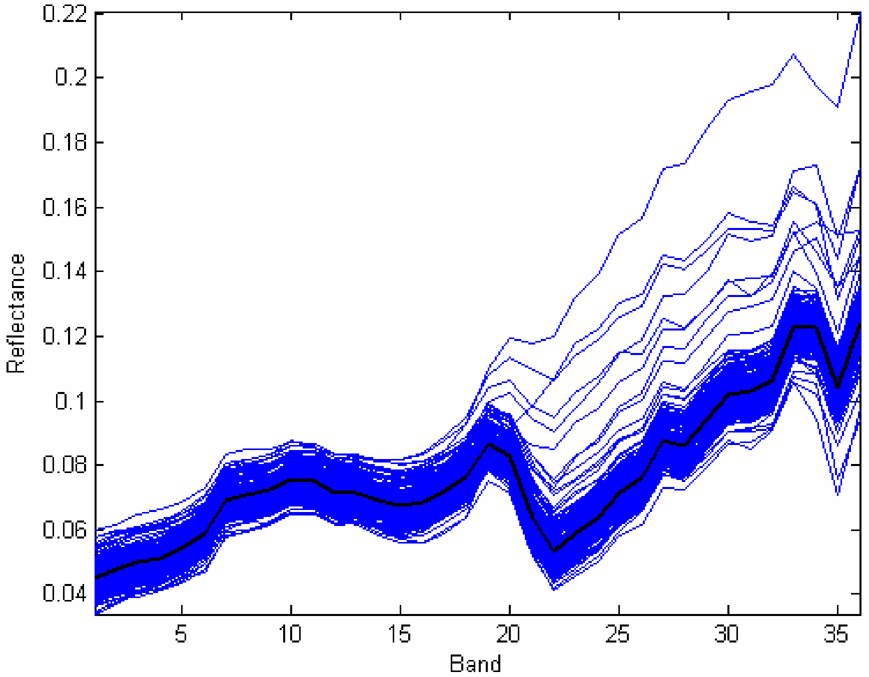


Figure 32. ***CASI_urban_12*** **scene targets generated using adaptive Gaussian model.**

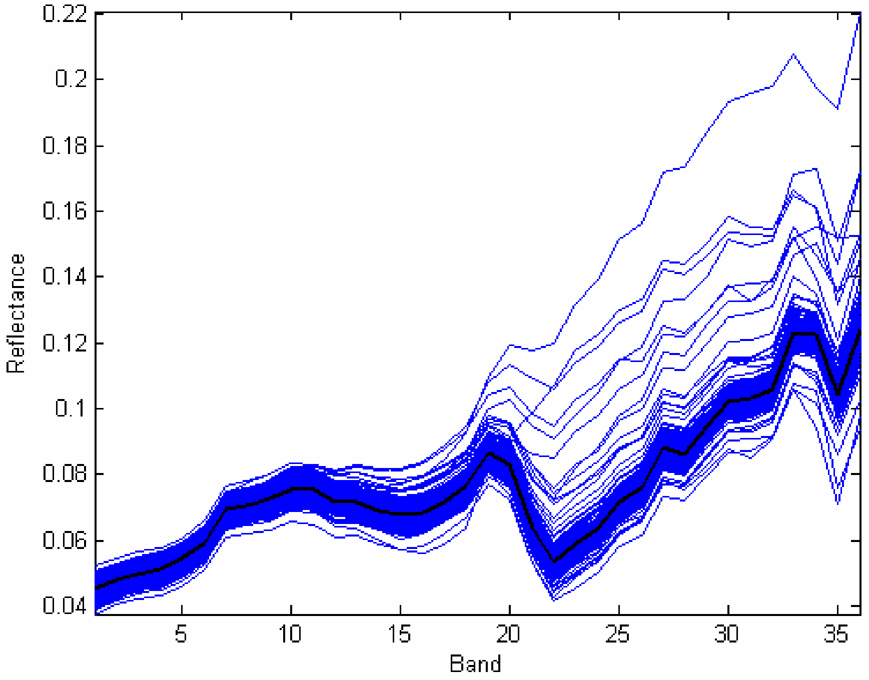


Figure 33. ***CASI_urban_15*** **scene targets generated using adaptive Gaussian model.**

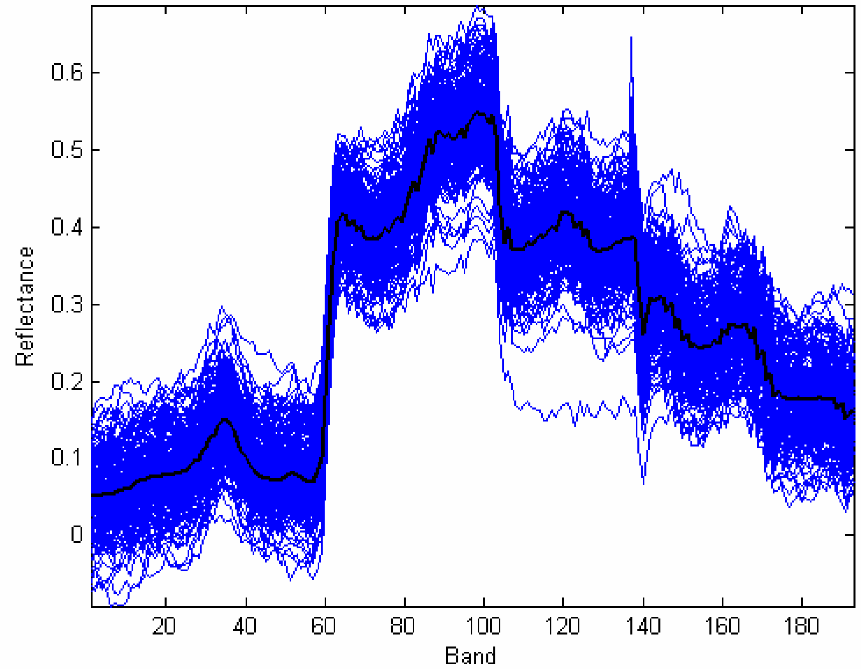


Figure 34. ***HYDICE_veg_8*** **scene targets generated using adaptive Gaussian model.**

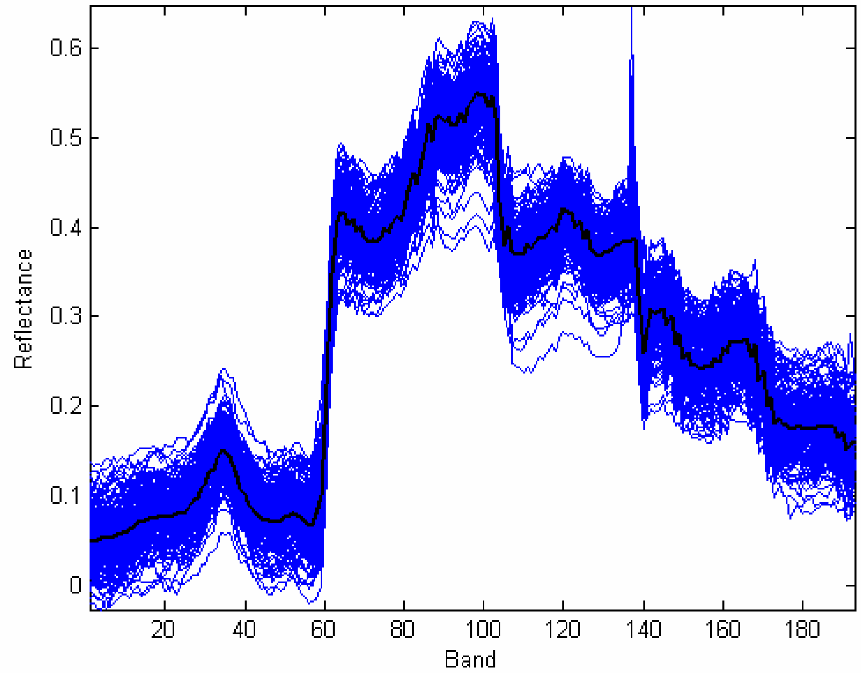


Figure 35. ***HYDICE_veg_10*** **scene targets generated using adaptive Gaussian model.**

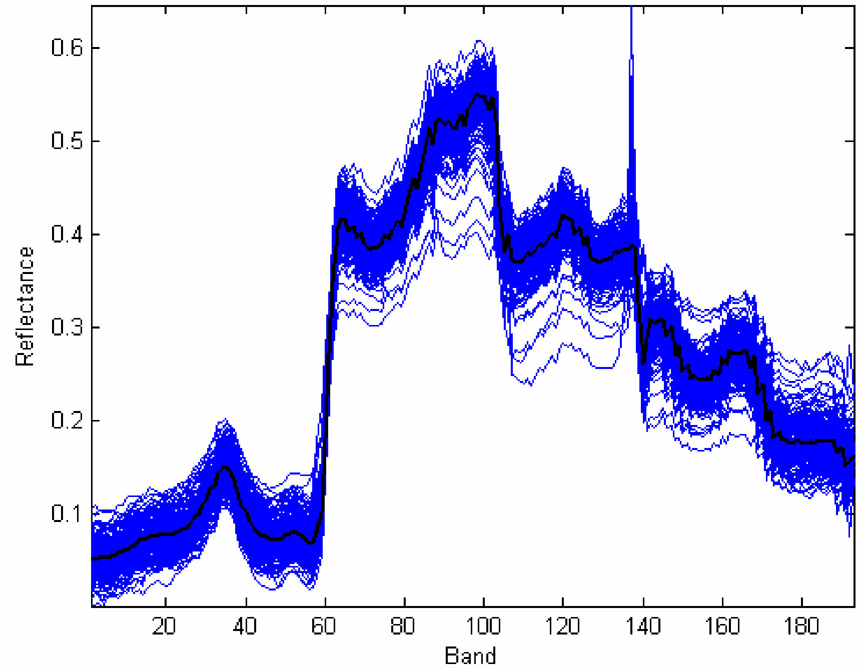


Figure 36. ***HYDICE_veg_12*** **scene targets generated using adaptive Gaussian model.**

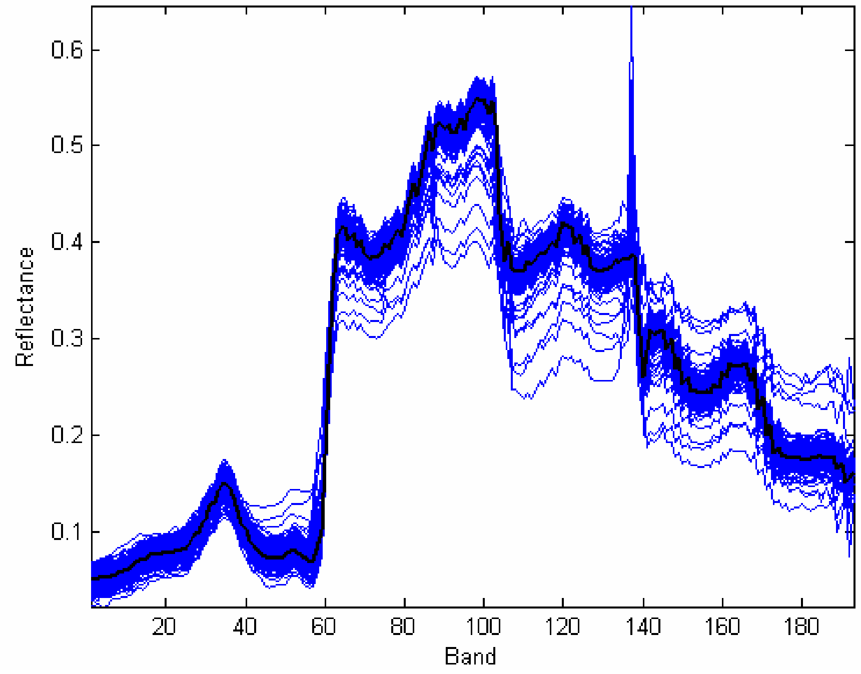


Figure 37. ***HYDICE_veg_15*** **scene targets generated using adaptive Gaussian model.**

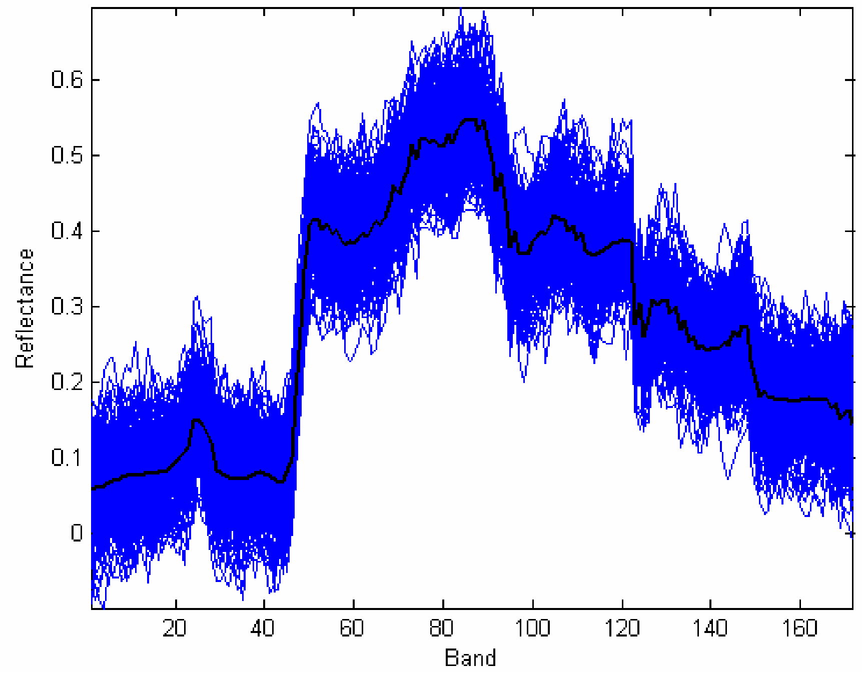


Figure 38. ***HYDICE_urban_8*** **scene targets generated using adaptive Gaussian model.**

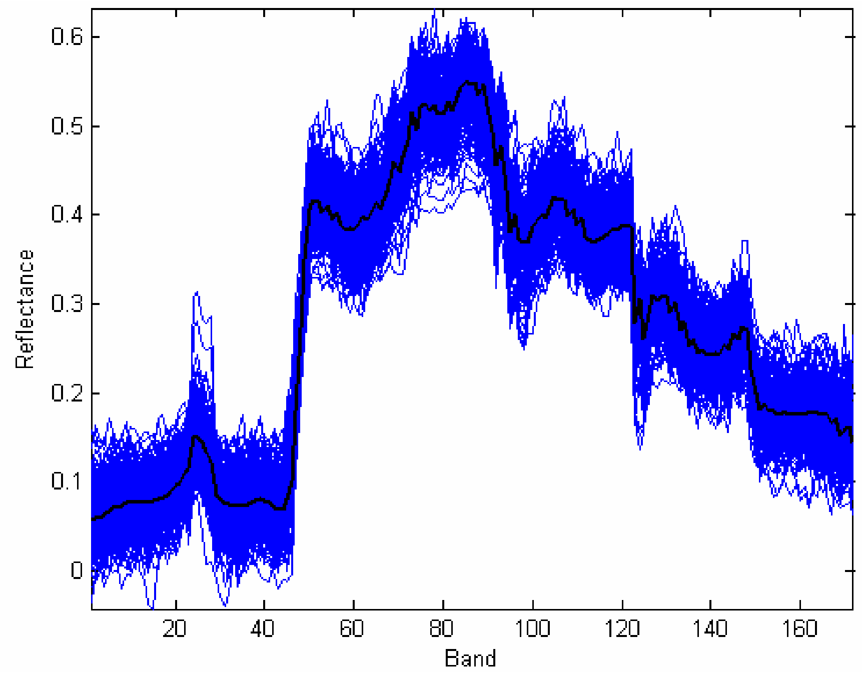


Figure 39. ***HYDICE_urban_10*** **scene targets generated using adaptive Gaussian model.**

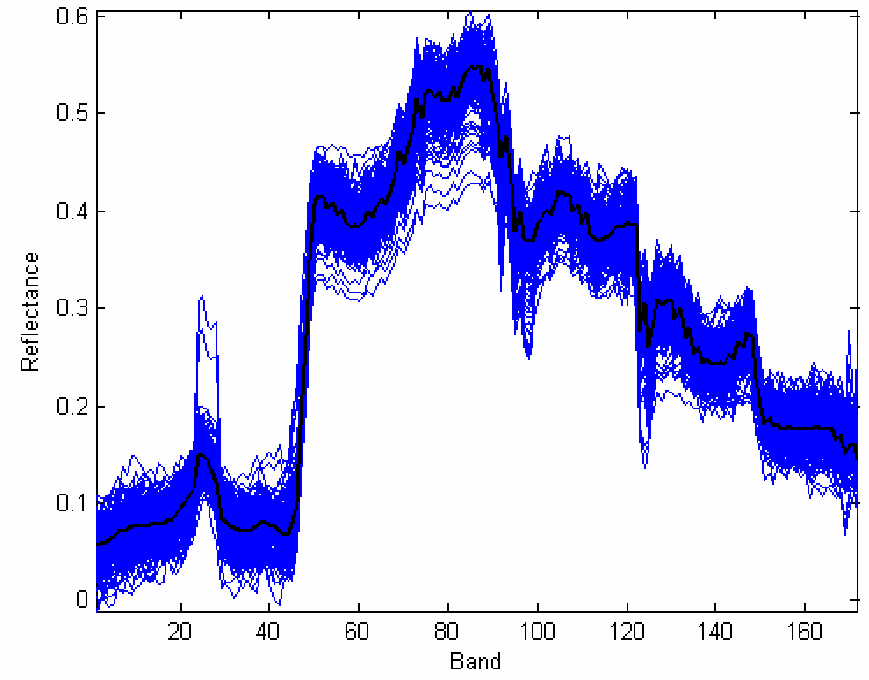


Figure 40. ***HYDICE_urban_12*** **scene targets generated using adaptive Gaussian model.**

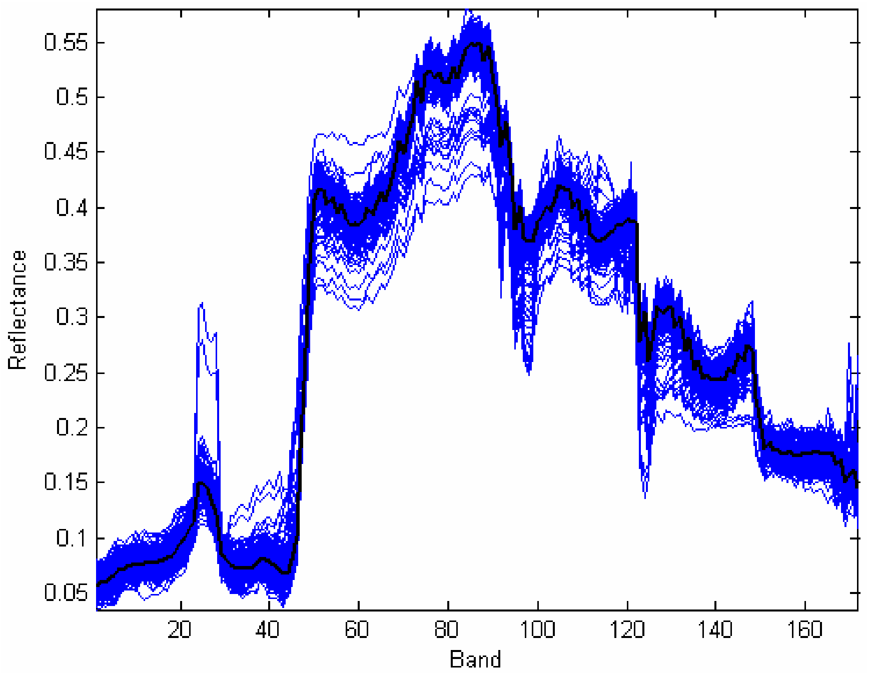


Figure 41. ***HYDICE_urban_15*** **scene targets generated using adaptive Gaussian model.**

CHAPTER V

THE SUPPORT VECTOR DATA DESCRIPTION

A. Introduction

In this chapter, we provide the theoretical framework for the kernel-based SVDD and discuss the various parameters that should be tuned for successful operation of the SVDD. Section B provides the mathematical background for the kernel-based SVDD. Section C provides our algorithm for optimizing the inherent SVDD parameter *s*.

In any supervised PR system, one should have access to a set of training samples that are representative of the class to be detected or classified. According to the well known *Hughes phenomenon*, as the number of dimensions increases, the number of training samples should increase exponentially in order to obtain reliable estimates of the multivariate statistics.[34] This has long been a challenge with the use of the likelihood-ratio based stochastic detectors presented in Chapter III. In target detection scenarios, we do not have access to a collection of samples characterizing the target class; we are typically given a pure target signature that is obtained from a spectral library. In our work, we will investigate the creation of *N* training samples pertaining to the target class according to both target variability models discussed in Chapter IV. The number of training samples *N* and the variance σ^2 used in the generation of the training samples are free parameters that will have an influence on the trained SVDD. These are free parameters that are connected with our proposed target training class algorithm and will be studied in sections D and E, respectively. Section F will present the concepts of decision-level fusion that will be utilized to alleviate the problem of choosing an appropriate σ^2 value in our SVDD-based detection scheme.

B. Theoretical Background

The support vector data description (SVDD) models a class of data by fitting a hypersphere with center $\boldsymbol{a}$ and radius R around all or most of the samples. Assume that we are given a set of training samples $\{\boldsymbol{x}_i, i = 1 \ldots N\}$. The SVDD aims to minimize the volume of the hypersphere by minimizing R^2. The task then becomes minimization of the following error function:[8]

$$F(R, \boldsymbol{a}, \xi_{\mathrm{i}}) = R^2 + C\sum_i \xi_i, \tag{5.1}$$

with the added constraints that *most* of the training samples $\boldsymbol{x}_i$ lie within the hypersphere. These constraints are postulated as follows:

$$\left\|\boldsymbol{x}_i - \boldsymbol{a}\right\|^2 \leq R^2 + \xi_i,\ i = 1, \ldots, N \tag{5.2}$$

The C parameter in (5.1) controls the tradeoff between the volume of the hypersphere and the number of target objects rejected.[35] Since the training data may contain outliers, the ξ_i in (5.1) represent *slack variables* that relax the constraints. The solution of (5.1) is obtained by solving the Lagrangian dual problem:

$$\max_{a_i}\left\{\sum_i a_i(\boldsymbol{x}_i \cdot \boldsymbol{x}_i) - \sum_{i,j} a_i a_j(\boldsymbol{x}_i \cdot \boldsymbol{x}_j)\right\} \tag{5.3}$$

subject to $0 \le a_i \le C$. After solving (5.3), only a subset of training samples will satisfy the equality given by (5.1). These are the $\boldsymbol{x}_i$ with corresponding nonzero a_i and are called the *support vectors* since they are the only samples needed to provide the hypersphere boundary around the data.

For a new object $\boldsymbol{y}$ to be considered as belonging to the SVDD, the square of the distance from the center of the sphere to $\boldsymbol{y}$ must be less than or equal R^2. Hence, $\boldsymbol{y}$ is deemed as belonging to the class when the following inequality is satisfied:[35]

$$(\boldsymbol{y}\cdot\boldsymbol{y}) - 2\sum_i a_i(\boldsymbol{y}\cdot\boldsymbol{x}_i) + \sum_{i,j} a_i a_j(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) \le R^2 \tag{5.4}$$

In many cases, fitting a hypersphere around the data in the original feature space does not provide a tight boundary. The nonlinear version of the SVDD implicitly maps the data from the input space to a higher-dimensional Hilbert feature space through a mapping function $\Phi(\boldsymbol{x})$. As a result, the problem becomes fitting a hypersphere around the data in the higher-dimensional feature space, which translates to a tighter, more accurate description of the boundary in the original feature space.[8] In the nonlinear SVDD, the inner products $(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)$ found in (5.3) are replaced by a kernel function $K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j)$ satisfying Mercer’s theorem[6]. Accordingly, equation (5.4) becomes the following:

$$K(\boldsymbol{y}\cdot\boldsymbol{y}) - 2\sum_i a_i K(\boldsymbol{y}\cdot\boldsymbol{x}_i) + \sum_{i,j} a_i a_j K(\boldsymbol{x}_i\cdot\boldsymbol{x}_j) \le R^2 \tag{5.5}$$

Several different choices of kernel functions exist. We will use the well-known Gaussian radial basis function (RBF) kernel. The RBF kernel only has one free parameter to be tuned and is shown to yield tighter boundaries than other kernel choices.[8,12,35] The RBF kernel is given by the following:

$$K(\boldsymbol{x},\boldsymbol{y}) = \exp(-\|\boldsymbol{x}-\boldsymbol{y}\|^2 / s^2) \tag{5.6}$$

In (5.6), *s* is the free parameter that is adjusted to control the tightness of the boundary and is typically optimized through cross-validation.[12,14]

Using the fact that $K(\boldsymbol{y} \cdot \boldsymbol{y}) = 1$ for the RBF kernel in (5.6), we can define a bias term that incorporates all constant terms in equation (5.5). The bias term is given by the following:

$$b = 1 + \sum_{i,j} a_i a_j K(\boldsymbol{x}_i \cdot \boldsymbol{x}_j) - R^2 \tag{5.7}$$

After incorporating the bias term of (5.7) into equation (5.5) and some algebraic manipulation, we have the following SVDD decision function:

$$SVDD(\boldsymbol{y}) = \operatorname{sgn}\left(\sum_i a_i K(\boldsymbol{y} \cdot \boldsymbol{x}_i) - \frac{b}{2} \right) \tag{5.8}$$

Thus, an input signature $\boldsymbol{y}$ is predicted to be a target if its output is positive and predicted to be background if its output is negative.

C. Optimization of the RBF Kernel Parameter *s*

As shown in the previous section, use of the SVDD with the RBF kernel requires selection of the free parameters *C* and *s*. In empirical trials of varying the value of C between 0.001 and 0.1, we found that all values yielded identical results. In this dissertation, we have used a value of $C = 0.01$. Proper tuning of *C* is not critical in practical applications of the SVDD.[35] On the other hand, the kernel parameter *s* has to be carefully chosen for successful SVDD performance. If *s* is chosen too small, a large number of support vectors will be selected, thus over fitting the training samples. In contrast, if *s* is chosen too large, a relatively small number of support vectors will be selected, thus under fitting the training samples and allowing for a loose boundary for the target class.[8,14] In this section, we present a supervised PR algorithm for the optimization of *s* and investigate the feasibility of efficient search methods for its optimization.

To begin, we constructed a set of *s* values ranging from 0.01 - 2 in increments of 0.002, yielding 1000 total candidate *s* values. Values above 2 were found to yield an insufficiently small number of support vectors, leading to a poor description of the target class. Thus, values above 2 are not considered. An SVDD was trained using each *s* value and applied to an independent validation set consisting of 100 target signatures and 8000 background signatures. The 100 target signatures are generated according to the adaptive Gaussian model. 8000 pixels, 20% of the total scene pixels, are randomly selected from the scene and used as background signatures since targets occur with such low probability. The ratio of 100 target signatures to 8000 background signatures was chosen to maintain a low probability scenario.

The detection results on the validation set were recorded for each trained SVDD. Because target detection scenarios are essentially binary decision problems, we have recorded the SVDD detection results via a *confusion matrix*[36]. Recall that the confusion matrix has four entries: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Once the detection results were obtained for each trained SVDD, the *F-stat* was calculated. The F-stat is a measure of a detection test's accuracy that considers both the *precision* and *recall* of the test and is defined as follows:[36]

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.9}$$

with the precision and recall given by the following:

$$precision = \frac{TP}{TP + FP} \tag{5.10}$$

$$recall = \frac{TP}{TP + FN} \tag{5.11}$$

In essence, we will be using the F-stat as our objective function to assess the detection results. Exhaustively searching for the optimal *s* from 1000 total values is a cumbersome process. In this section, we will study the behavior of the F-stat values as a function of *s* in various scenarios to assess the feasibility of using an efficient search algorithm for optimizing *s*. Our approach will be the following:

1. Load one of the four possible HSI data cubes (e.g., *CASI_urban_XX*, *CASI_veg_XX*, *HYDICE_urban_XX*, *HYDICE_veg_XX*). The SNR of the inserted scene targets has no impact on the results. Scenes with different target SNRs (i.e., *CASI_urban_10* versus *CASI_urban_15*) will yield identical results since the selection of *s* is not affected by the targets in the scenes—it is affected by the variability (SNR) of the generated training set and the validation set.

2. Generate 100 SVDD training set signatures according to the adaptive Gaussian model. The variance σ^2 will be varied to achieve SNRs of 8 dB, 10 dB, 12 dB, and 15 dB.
3. Train an SVDD using each value of s and plot the F-stat as a function of *s*.

1. Linear search results for s on HSI scenes

In Figure 42 – Figure 45, we provide the results of the linear search for *s* on the urban CASI scenery. It appears that in each case, the F-stat reaches a global maximum. However, in the SNR = 15 dB and SNR = 12 dB cases, several values of *s* yield the maximum. Notice that as the SNR of the target training set decreases (i.e., increasing variability in the training set), the maximum value of the F-stat decreases. For each of the target training set SNRs, Table 1 provides the maximum value of F-stat, the number of *s* values that yield the maximum value of F-stat, and the corresponding range of *s* values.

Figure 46 - Figure 49 provide the results of the linear search for *s* on the vegetative CASI scenery. Once again, the F-stat reaches a plateau for certain values of *s* and declines afterwards. Table 2 summarizes the behavior of the F-stat values as a function of s for each target training set SNR. Table 3 and Table 4 provide the summaries of results for the urban and vegetative HYDICE scenery, respectively, while Figure 50 - Figure 57 show the results of the linear search for *s* on these scenes.

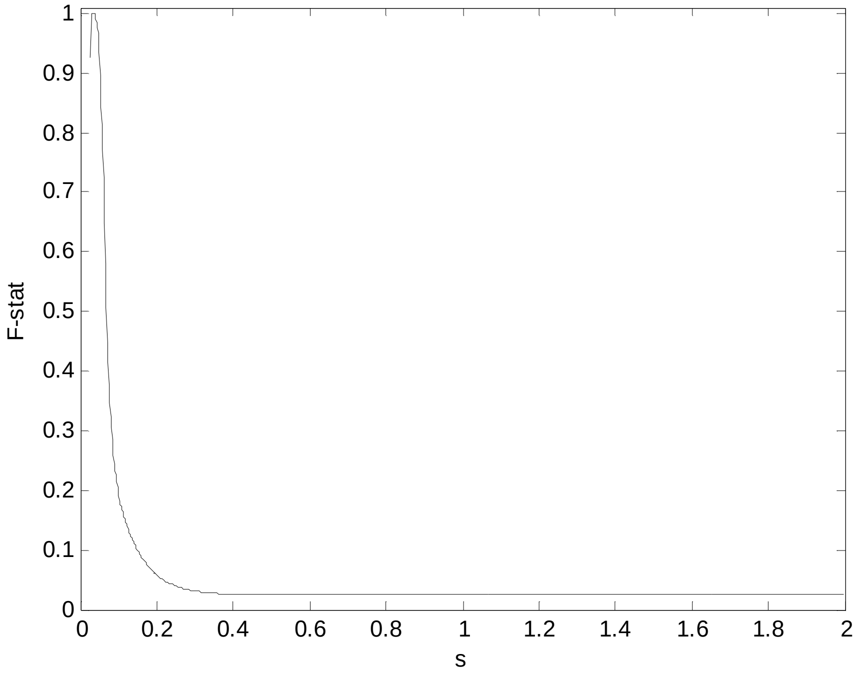


Figure 42. Linear search for *s* for urban CASI scenery with training set SNR = 15 dB.

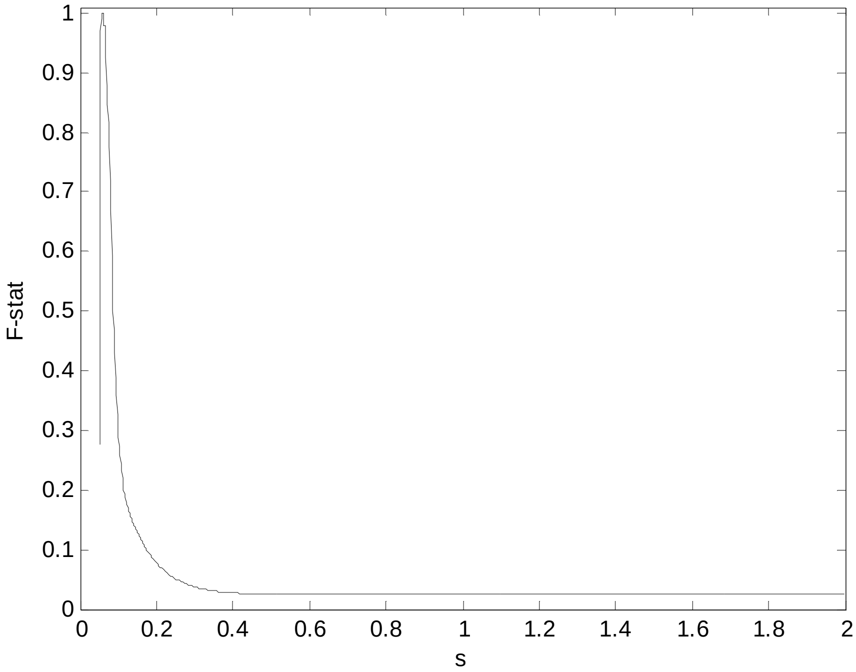


Figure 43. Linear search for *s* for urban CASI scenery with training set SNR = 12 dB.

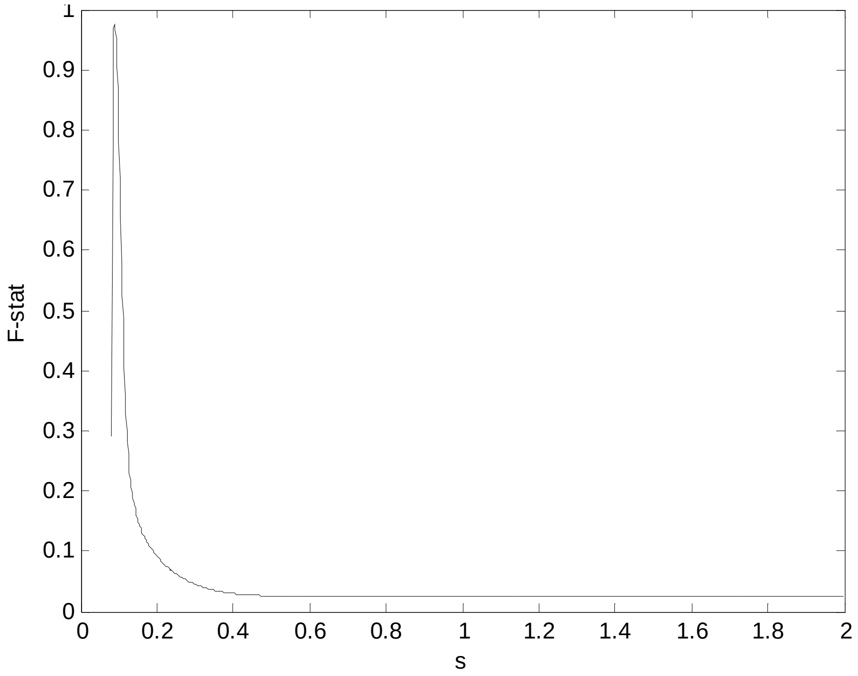


Figure 44. Linear search for *s* for urban CASI scenery with training set SNR = 10 dB.

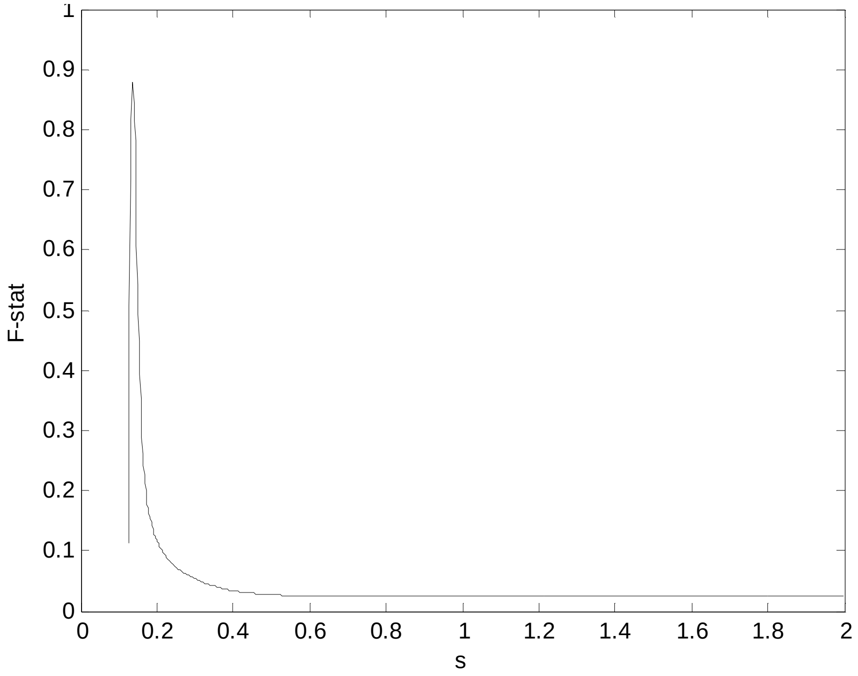


Figure 45. Linear search for *s* for urban CASI scenery with training set SNR = 8 dB.

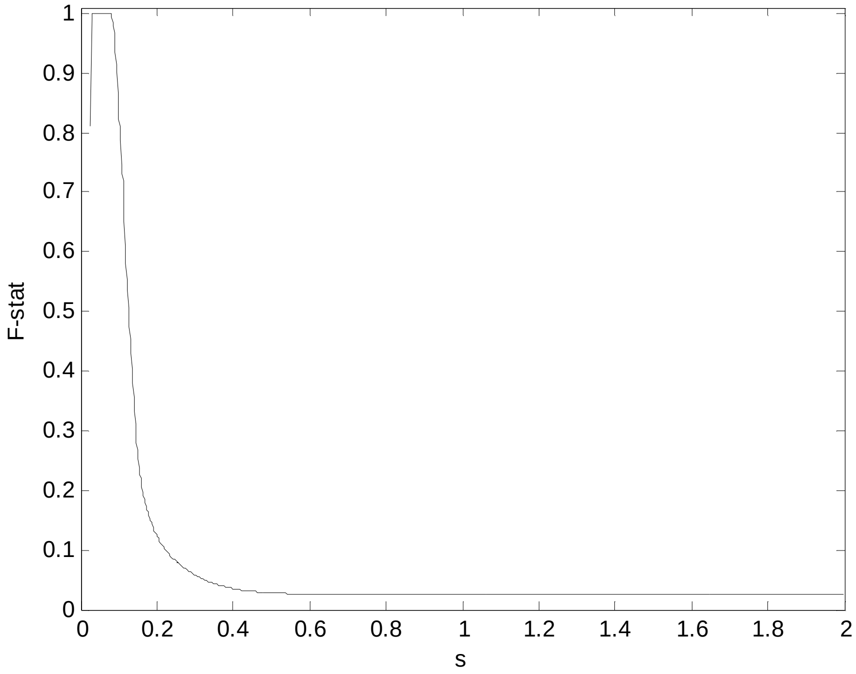


Figure 46. Linear search for *s* for vegetative CASI scenery with training set SNR = 15 dB.

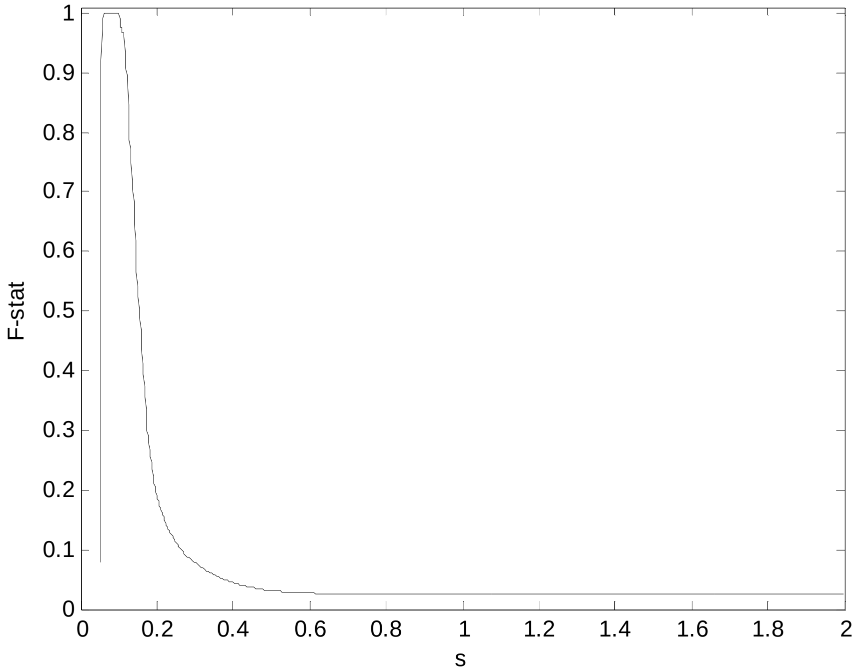


Figure 47. Linear search for *s* for vegetative CASI scenery with training set SNR = 12 dB.

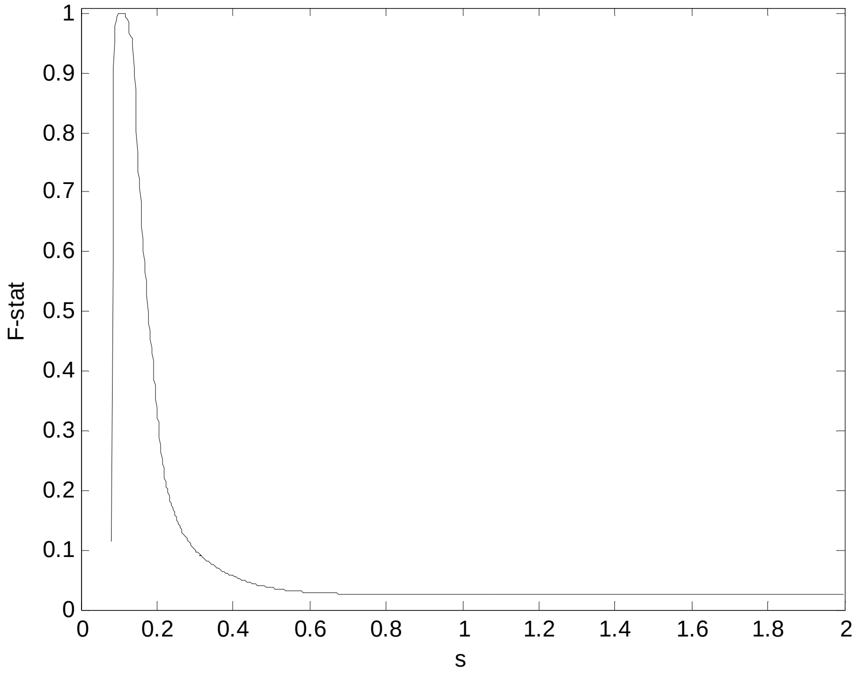


Figure 48. Linear search for *s* for vegetative CASI scenery with training set SNR = 10 dB.

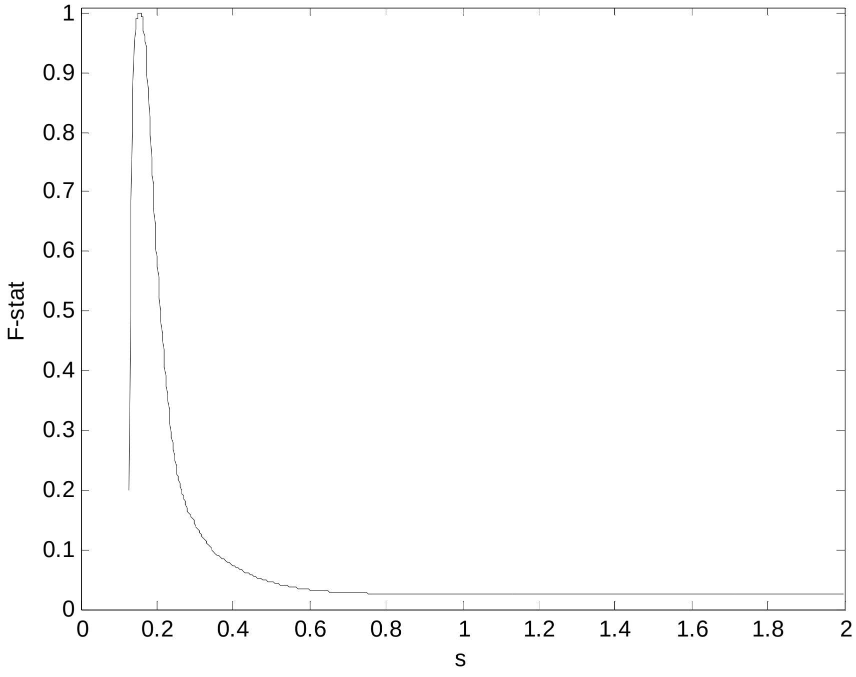


Figure 49. Linear search for *s* for vegetative CASI scenery with training set SNR = 8 dB.

Table 1. Behavior of F-stat values as a function of *s* for urban CASI scenery.

Training Set SNR (dB)	Maximum F-stat Value	Number of Maximums	Range of *s*
15	1	5	0.031 – 0.039
12	1	2	0.059, 0.061
10	0.97512	1	0.089
8	0.87923	1	0.137

Table 2. Behavior of F-stat values as a function of *s* for vegetative CASI scenery.

Training Set SNR (dB)	Maximum F-stat Value	Number of Maximums	Range of *s*
15	1	25	0.031 – 0.079
12	1	21	0.061 – 0.101
10	1	11	0.097 – 0.117
8	1	5	0.151 – 0.159

Table 3. Behavior of F-stat values as a function of *s* for urban HYDICE scenery.

Training Set SNR (dB)	Maximum F-stat Value	Number of Maximums	Range of *s*
15	1	222	0.219 – 0.661
12	1	185	0.437 – 0.805
10	1	133	0.691 – 0.955
8	1	47	1.097 – 1.189

Table 4. Behavior of F-stat values as a function of *s* for vegetative HYDICE scenery.

Training Set SNR (dB)	Maximum F-stat Value	Number of Maximums	Range of *s*
15	1	130	0.259 – 0.517
12	1	85	0.515 – 0.683
10	1	30	0.817 – 0.875
8	0.9706	1	1.245

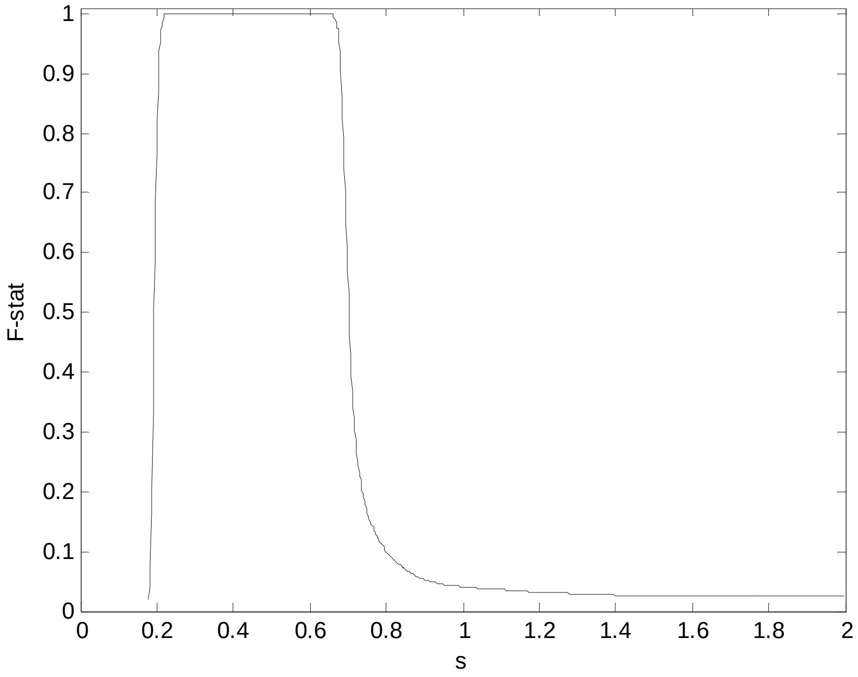


Figure 50. Linear search for *s* for urban HYDICE scenery with training set SNR = 15 dB.

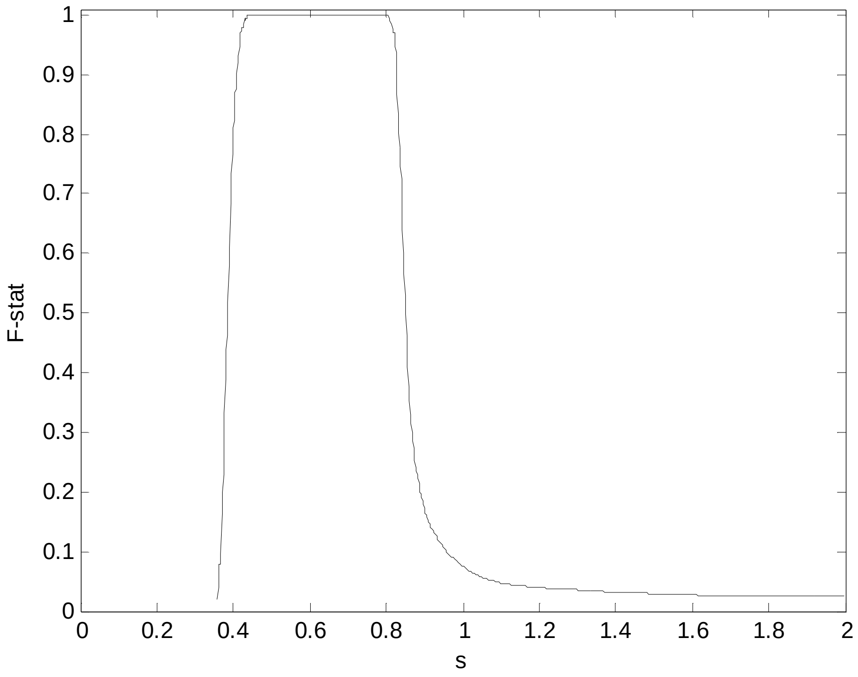


Figure 51. Linear search for *s* for urban HYDICE scenery with training set SNR = 12 dB.

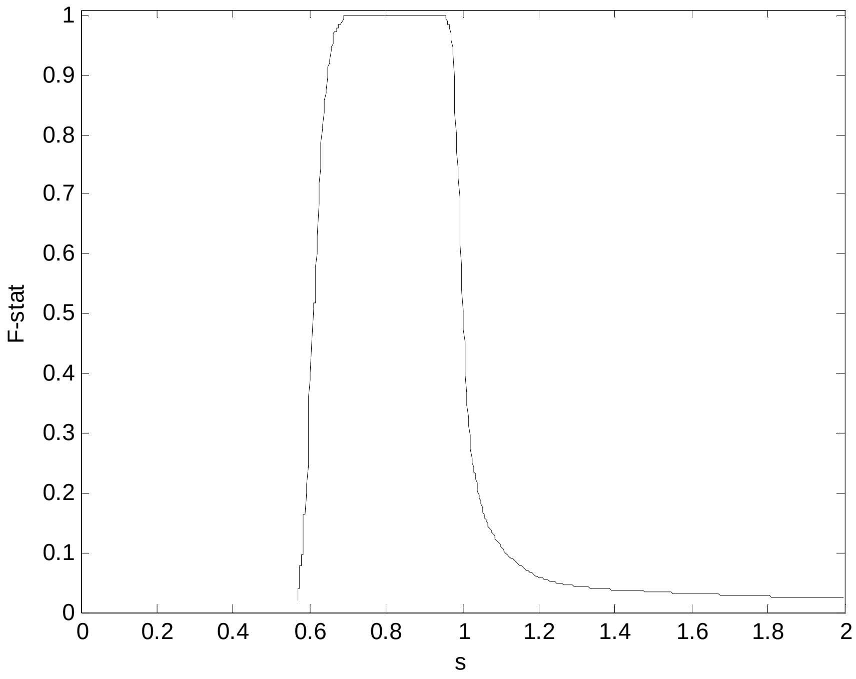


Figure 52. Linear search for *s* for urban HYDICE scenery with training set SNR = 10 dB.

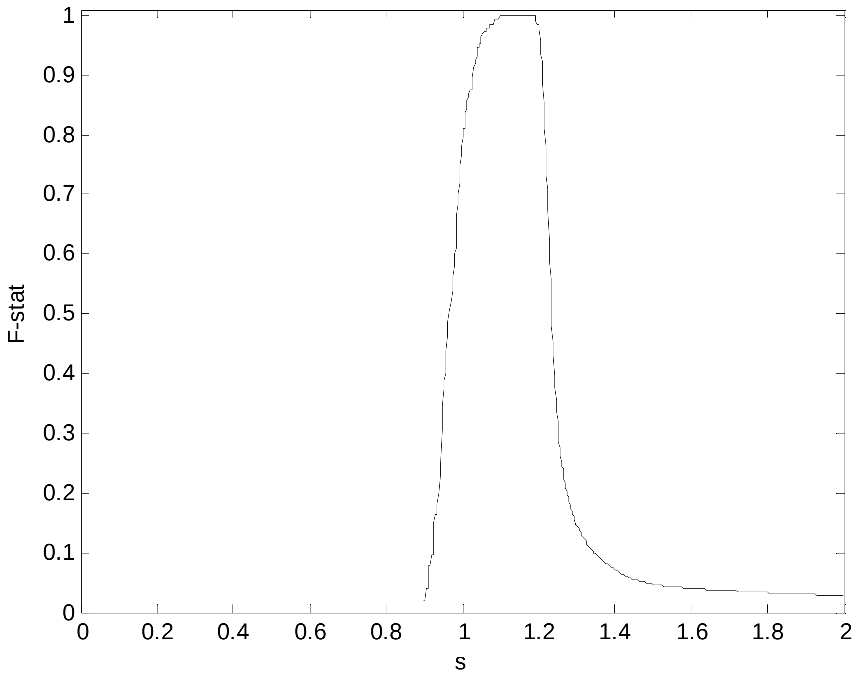


Figure 53. Linear search for *s* for urban HYDICE scenery with training set SNR = 8 dB.

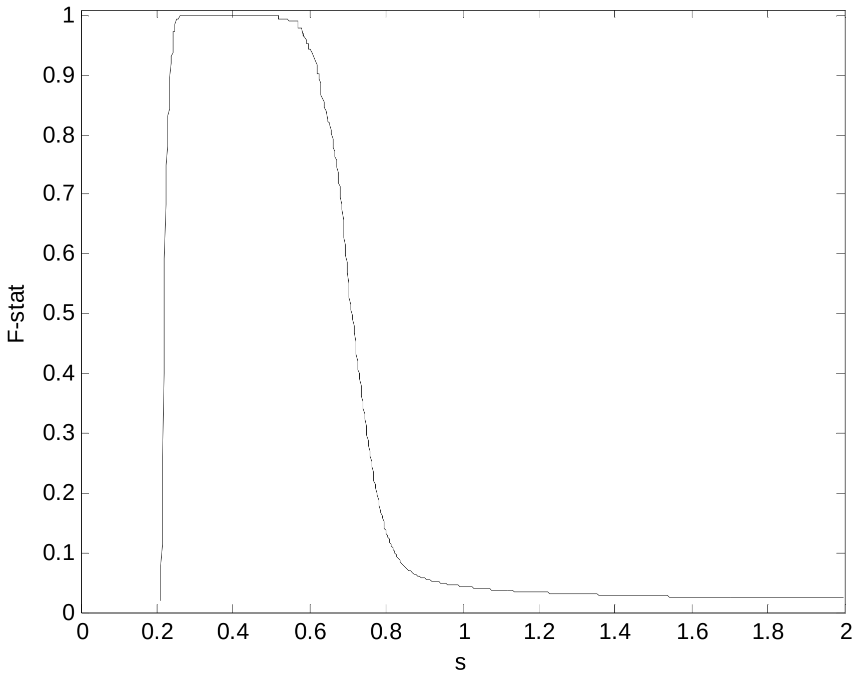


Figure 54. Linear search for *s* for vegetative HYDICE scenery with training set SNR = 15 dB.

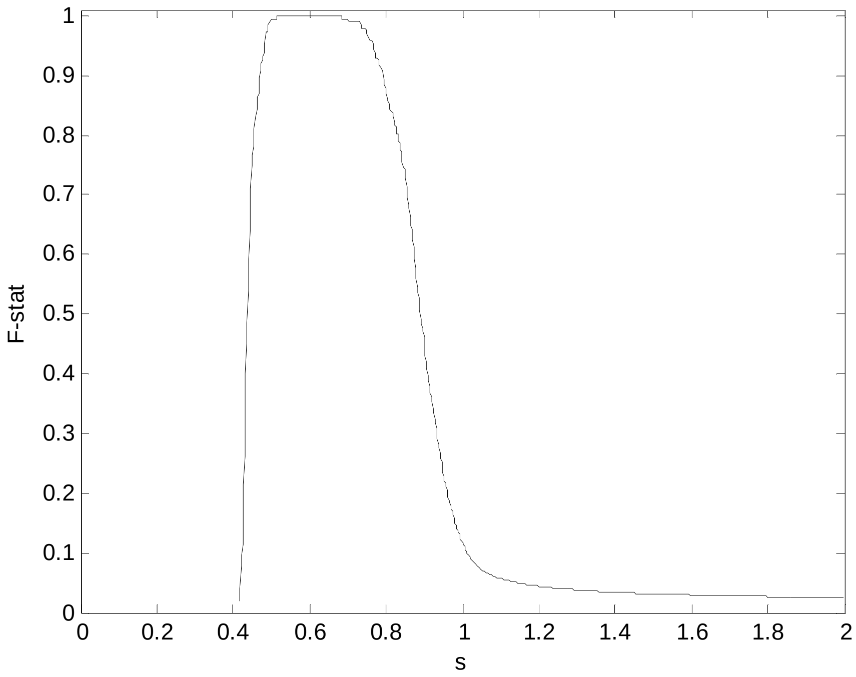


Figure 55. Linear search for *s* for vegetative HYDICE scenery with training set SNR = 12 dB.

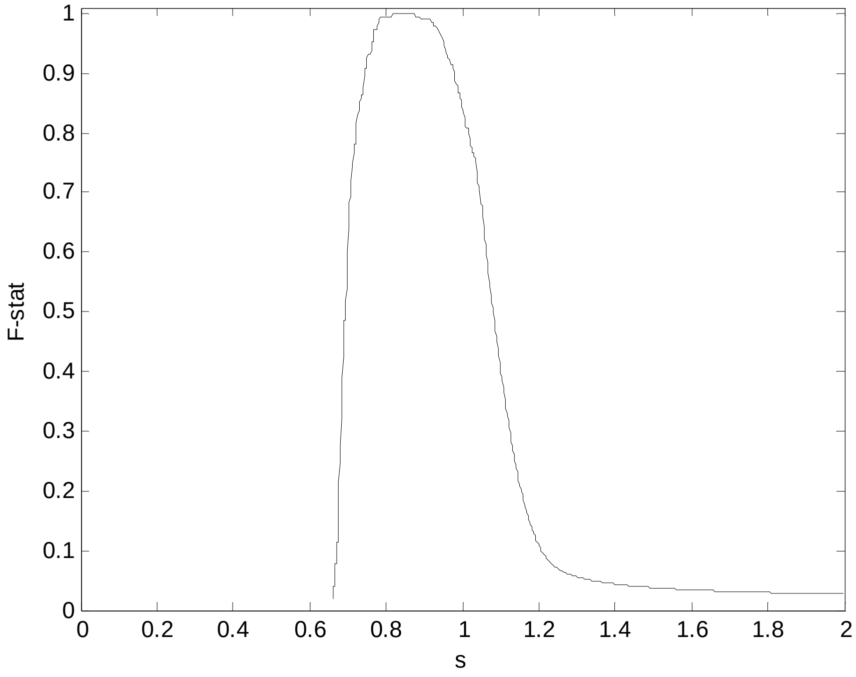


Figure 56. Linear search for *s* for vegetative HYDICE scenery with training set SNR = 10 dB.

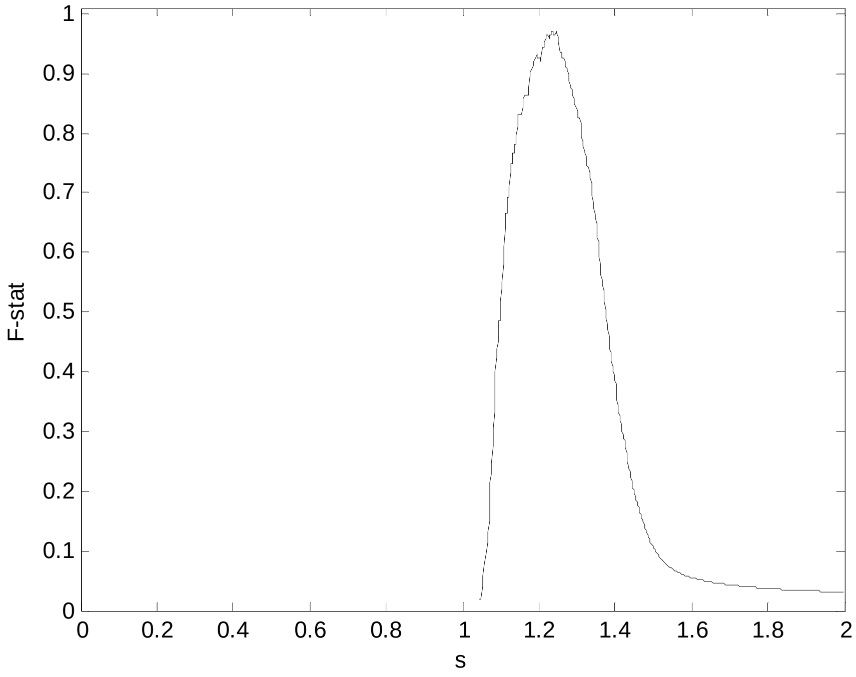


Figure 57. Linear search for *s* for vegetative HYDICE scenery with training set SNR = 8 dB.

2. Analysis of linear search results for s

As seen in Table 1, the search results yield a unique global maximum for SNRs of 10 dB and 8 dB. In all of the scenes, the number of maximums decreases as the variability in the training set increases. Also notice that the range of *s* values that yield the maximum F-stat value increases as the variability in the training set increases. For the CASI sensor, the vegetative scenery has a larger number of *s* values yielding the maximum F-stat value, as Table 2 shows. In fact, a unique global maximum is not reached for the vegetative CASI scenery. Concerning the HYDICE scenery, a considerably large number of *s* values produce the maximum F-stat value, as can be seen in Table 3 and Table 4. A unique global maximum is reached for the vegetative HYDICE scenery with a target training set SNR of 8 dB.

Let us take a closer examination of the behavior of the F-stat objective function with respect to the variation of *s*. If we substitute the definitions for *precision* and *recall* from (5.10) and (5.11) into the equation for the F-stat in (5.9), we obtain the following:

$$\begin{aligned} F &= 2\cdot\frac{\frac{TP}{(TP+FP)}\cdot\frac{TP}{(TP+FN)}}{\frac{TP}{(TP+FP)}+\frac{TP}{(TP+FN)}} \\ &= 2\cdot\frac{TP}{(TP+FN)+(TP+FP)} \qquad (5.12) \\ &= \frac{2TP}{2TP+FP+FN} \end{aligned}$$

After dividing the numerator and denominator of (5.12) by $2TP$, we obtain the following:

$$\begin{aligned} F &= \frac{1}{1+\left(\dfrac{FP+FN}{2TP}\right)} \\ &= \frac{1}{1+\left(\dfrac{T-TP+FP}{2TP}\right)} \end{aligned} \tag{5.13}$$

where $T = TP + FN$ represents the total number of targets. We can further simplify the expression for F in (5.13) by combining the variable terms in the denominator to obtain the following:

$$F = \frac{1}{1+G} \tag{5.14}$$

where

$$G = \frac{\left(T - TP + FP\right)}{2TP} \tag{5.15}$$

We stated earlier that smaller values of *s* will overfit the training data, while larger values of s will underfit the training data. Hence, as *s* ranges from small to large values, the TPs will range from very small values to a maximum value. Similarly, the FPs will range from very small values to a large value (as many as the number of background pixels in the scene). Accordingly, G will have a minimum at some point. Since the maximum number of TPs is given by T and the minimum number of FPs is zero, the smallest possible value of G is zero. These relationships are depicted in Figure 58.

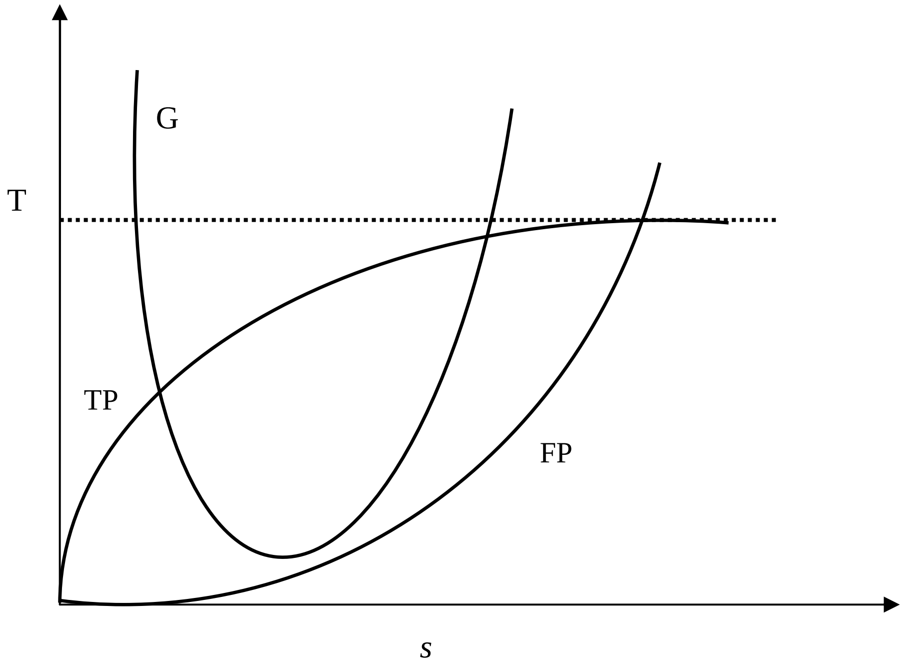


Figure 58. Relationships between TP, FP, and G.

Depending on the separability between the target and background signatures, the minimum (or minima) of G may occur either at one distinct *s* value or a range of *s* values. Figure 59 and Figure 60 show two extreme cases. In the first extreme case shown in Figure 59, the minimum value of G is nonzero since the target and background classes are not completely separable. In this case, a single *s* value yields the minimum value of G. In the second extreme case shown in Figure 60, G reaches its lowest possible value of zero indicating that the target and background classes are completely separable. In this case, a range of s values yields the minimum value of G. The width of this range is proportional to the amount of separability between the target and background classes.

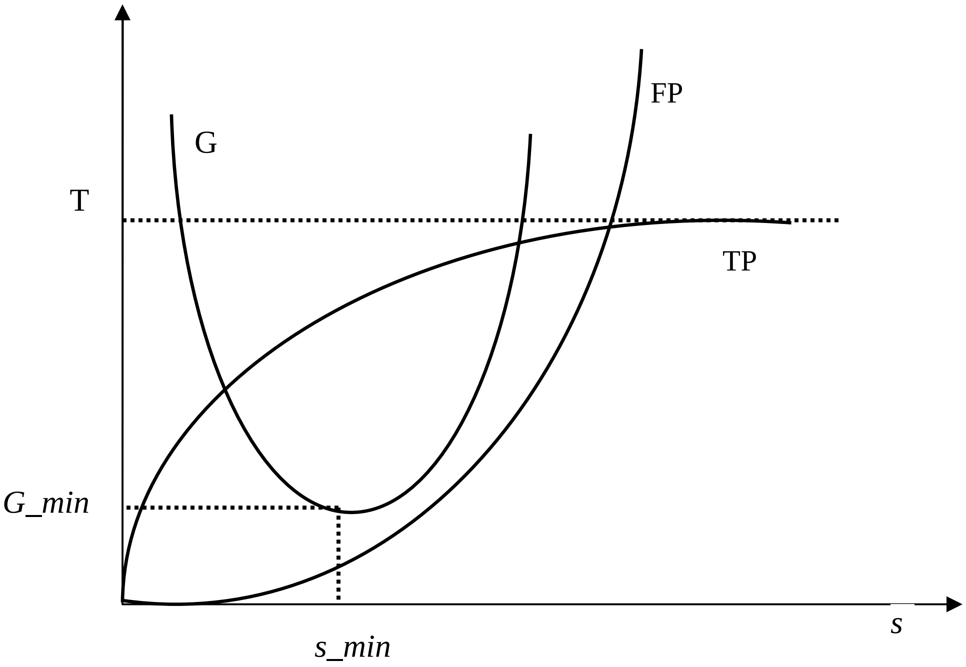


Figure 59. Behavior of TP, FP, and G when target and background classes overlap.

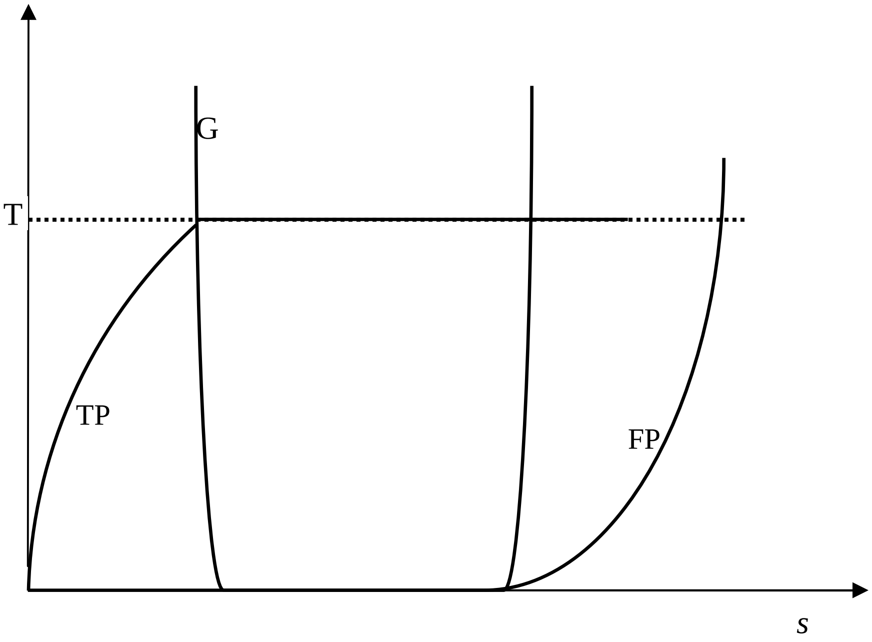


Figure 60. Behavior of TP, FP, and G when target and background classes are fully separated.

Note that when G reaches its absolute minimum value of zero, F reaches its maximum value of one. As the previous figures and tables show and in light of our previous discussion, the separation between target and background signatures in the urban CASI scenery appears to be less than in all the other cases. In both HYDICE scenes, the separation between target and background signatures appears to be large based on the number of maximums obtained and the range of *s* for each of the target training set SNRs. Notice how the width of the plateaus decreases as the target training set SNR decreases. As the variability of the target training set increases, the separation between target and background class signatures decreases, making the selection of *s* more crucial.

3. *Impact of range of s on detection results*

Performing an exhaustive linear search over a large range of candidate *s* values is a time-consuming process. Our ultimate goal is to employ an efficient search algorithm for finding the optimal *s*. As we saw in the previous section, an exhaustive search for s yields unimodal behavior for the F-stat values as a function of *s*. We also saw that, in most cases, a range of s values yielded the maximum objective function value. In this section, we first wish to gain insight into how the selection of *s* within a particular range of *s* values affects the detection results on our HSI data. For each of the data cubes and each of the target training set SNRs, we did the following:

1. Trained an SVDD using the smallest *s* value in the range.
2. Trained an SVDD using the median *s* value in the range.

3. Trained an SVDD using the largest *s* value in the range.
4. Recorded the confusion matrices for all three cases.

We have summarized the results of this process by providing the FPs and TPs of detection for all three scenarios for each data cube and target training set SNR. The results are provided in Table 5 - Table 8. In each cell, we have recorded the number of FPs (left) and TPs (right) for each detection scenario. Each cell will typically contain three entries—the top entry corresponds to the FPs and TPs using the smallest value of *s*, the middle entry corresponds to the FPs and TPs using the median value of *s*, and the bottom entry corresponds to the FPs and TPs using the largest value of *s*. A cell that only contains one entry indicates that a single value of *s* produced the global maximum for that target training set SNR (see Table 1 - Table 4).

As the results show, the number of TPs either remains the same or increases as *s* increases from its smallest to largest values. This agrees with our understanding that the SVDD boundary expands as *s* is increased. Along the diagonal entries, the target training set SNR matches the SNR of the targets in the scene—in these scenarios, any increase in TPs as a function of *s* does not appear substantial. Above the diagonal entries, the target training set variability is larger than the variability of the targets in the scene—in these scenarios, the increase in TPs as a function of *s* is even less dramatic. Below the diagonal entries, the target training set variability is less than the variability of the targets in the scene—in these scenarios, the increase in TPs as a function of *s* is the most dramatic.

Table 5. Impact of range of *s* on detection results for urban CASI scenery.

Training Set SNR (dB)	8 dB Scene	10 dB Scene	12 dB Scene	15 dB Scene
8	140/174	141/189	141/190	141/190
10	31/126	31/181	31/186	31/186
12	9/79 13/86 17/102	9/160 13/168 17/176	9/182 13/183 17/185	9/184 13/184 18/185
15	0/12 4/46 14/73	0/56 4/111 14/148	0/145 4/173 14/178	0/180 4/182 14/184

Table 6. Impact of range of *s* on detection results for vegetative CASI scenery.

Training Set SNR (dB)	8 dB Scene	10 dB Scene	12 dB Scene	15 dB Scene
8	0/183 2/184 12/186	0/186 2/186 12/187	0/186 2/186 12/187	0/186 2/186 12/187
10	0/152 0/177 4/181	0/183 0/186 4/186	0/184 0/186 4/186	0/184 0/186 4/186
12	0/68 0/166 21/181	0/147 0/186 19/187	0/181 0/186 19/187	0/181 0/186 19/187
15	0/3 0/120 9/169	0/45 0/174 9/186	0/119 0/184 9/186	0/178 0/184 9/186

Table 7. Impact of range of *s* on detection results for urban HYDICE scenery.

Training Set SNR (dB)	8 dB Scene	10 dB Scene	12 dB Scene	15 dB Scene
8	1/200 1/200 4/200	1/200 1/200 4/200	1/200 1/200 4/200	1/200 1/200 5/200
10	1/31 1/198 4/200	1/190 1/200 4/200	1/191 1/200 4/200	1/191 1/200 4/200
12	1/6 1/170 3/200	1/30 1/199 4/200	1/186 1/199 4/200	1/186 1/199 4/200
15	0/3 1/91 3/200	0/3 1/196 3/200	0/4 1/196 3/200	0/182 1/196 3/200

Table 8. Impact of range of *s* on detection results for vegetative HYDICE scenery.

Training Set SNR (dB)	8 dB Scene	10 dB Scene	12 dB Scene	15 dB Scene
8	32/189	32/198	32/198	32/198
10	0/119 0/142 1/164	0/190 0/192 1/194	0/194 0/194 1/195	0/194 0/194 1/195
12	0/17 0/85 1/148	0/122 0/184 1/192	0/187 0/193 1/194	0/191 0/193 1/194
15	0/4 0/34 1/131	0/4 0/147 1/192	0/49 0/189 1/194	0/182 0/191 1/194

The most significant finding in these results is that it is advantageous to use an *s* value closest to the largest *s* value in the range of *s* values for a particular target training set SNR. In doing so, we guarantee that we will maximize the number of TPs, despite the marginal increase in FPs as a result. Additionally, we can use an efficient search algorithm for finding the optimal *s* value, as will be discussed in the next section.

4. Exploration of an efficient search method for s

An optimization problem generally involves the minimization or maximization of an objective function $f(x)$ where x is either univariate or multivariate. If there are any inequality or equality constraints that must be satisfied in the optimization, the process is known as *constrained optimization*; otherwise, the process is *unconstrained optimization*. In *multimodal* functions, we must consider the possibility of both local and global optima, as Figure 61 shows.

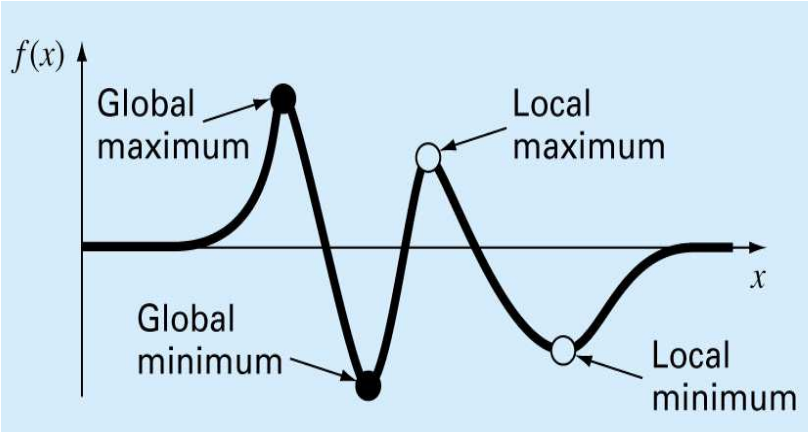


Figure 61. Example of a multimodal function with local and global optima.

In the previous section, the F-stat was our objective function that we attempted to maximize as a function of our univariate *s* parameter. Because we have no constraints to satisfy in the process, our problem is one of unconstrained optimization. We saw that the F-stat exhibited unimodal behavior, but there typically was not a unique global maximum but rather a "plateau" of maxima. However, further experiments revealed that we could use an *s* value corresponding to the right side of the plateau as our optimal *s* value. In the next section, we provide background for the golden-section search algorithm that we have used for our univariate unconstrained optimization problem. We then describe our algorithm for modifying this method to compensate for the fact that a single maxima does not always exist in our application of the optimization of *s*.

a. The golden-section search algorithm

The golden-section search algorithm first requires two initial guesses x_l and x_u that contain the maximum. Then, two interior points x_1 and x_2 are chosen according to the *golden ratio* as follows:[37]

$$x_1 = x_l + d \tag{5.16}$$

$$x_2 = x_u - d \tag{5.17}$$

where

$$d = R \cdot (x_u - x_l) \tag{5.18}$$

In (5.18), R is the golden ratio. For details concerning the calculation of R, the reader is encouraged to consult ref 37. Once x_1 and x_2 are chosen, the objective function is evaluated at these two points. If $f(x_1) > f(x_2)$, the domain of x to the left of x_2 (from x_l to x_2) does not contain the maximum and can be eliminated. Then, x_2 becomes the new x_l. If $f(x_2) > f(x_1)$, the domain of x to the right of x_1 (from x_1 to x_u) can be eliminated. In this case, x_1 becomes the new x_u. These scenarios are shown in Figure 62, and Figure 63, respectively.

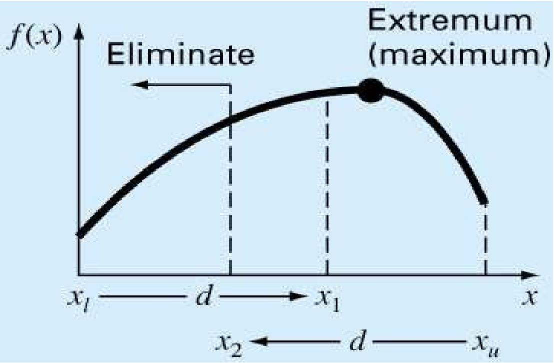


Figure 62. Golden-section search for $f(x_1) > f(x_2)$.[37]

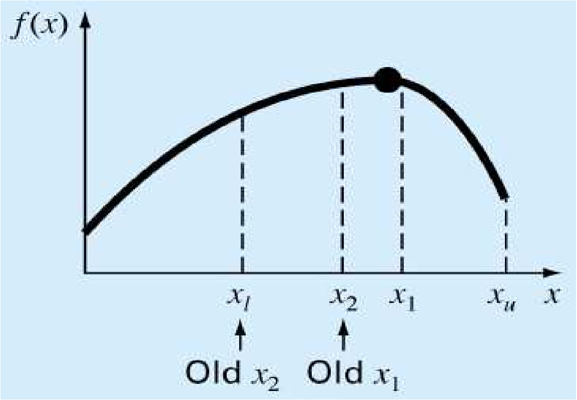


Figure 63. Golden-section search for $f(x_2) > f(x_1)$.[37]

The benefit of using the golden ratio is that we do not need to recalculate all the function values in the next iteration. If $f(x_1) > f(x_2)$, then x_1 gets assigned as the new x_2 value; otherwise, x_2 gets assigned as the new x_1 value. The golden-section search algorithm terminates when the following stopping criterion is satisfied:

$$\left| x_u - x_l \right| < \varepsilon \tag{5.19}$$

b. Novel utilization of the golden-search algorithm

Earlier, we concluded that choosing an *s* value closer towards the largest value in the range of *s* values will provide desired detection results. In this section, we present the steps that we have taken to use the golden-search algorithm in conjunction with our requirements. The steps are as follows:

1. Generate the validation set consisting of 100 target signatures and 8000 background signatures as previously discussed.
2. Generate 100 SVDD training set signatures according to the Adaptive Gaussian model. The variance σ^2 will be varied to achieve SNRs of 8 dB, 10 dB, 12 dB, and 15 dB.
3. Run the golden search algorithm with $x_l = 0$ and $x_u = 2$. Capture the value of the objective function $F(s_1) = F_{\max}$ and the corresponding value s_1.
4. Run the golden search algorithm again with $x_l = s_1$ and $x_u = 2$. Capture the value of the objective function $F(s_2)$ and the corresponding value s_2.

5. As long as $F(s_i) = F_{\max}$, step 4 is iteratively repeated with $x_l = s_{i-1}$ and $x_u = 2$. This means that we are "walking along" the acceptable range of *s* values, but we want to keep traversing this range until $F(s_i)$ starts to decrease.
6. If $F(s_i) < F_{\max}$, set the optimal *s* value $s_{opt} = s_{i-1}$. This indicates that we have reached an *s* value that has produced an objective function value that is less than $F_{\max}$. This corresponds to an *s* value that takes us beyond the desired range of *s* values. As a result, we have selected the most recent (and largest) s value that yields $F_{\max}$.

5. Conclusion

In this section, we have developed a method for optimization of the SVDD kernel parameter *s*. We first studied the behavior of the F-stat objective function over a continuous range of s values to find that it exhibited unimodal behavior. In many instances, a range of *s* values yielded the maximum F-stat value, and we found that this was affected by the separability between the target and background signature classes. We also studied how selection of *s* within the range of *s* values affected detection results and found that it was advantageous to pick the largest *s* value in the range. Accordingly, we used the golden-section search algorithm for 1-D unconstrained optimization as a means of efficiently finding the optimal *s* value. The algorithm was used in an iterative fashion to locate the largest *s* value in the range of acceptable *s* values.

D. Impact of the Value of *N*

1. Introduction

In this section, our goal is to determine a sufficient number of signatures to generate for the SVDD target training class. Earlier, we mentioned the *Hughes phenomenon*, which requires that the number of training samples should increase exponentially for the likelihood-ratio based algorithms. Recall that the SVDD is a boundary method for describing a class of data; the quadratic classifiers discussed earlier make assumptions about the density of the data and then proceed to estimate the density parameters (i.e., mean and covariance) from the data. Hence, an advantage to the use of a support vector-based algorithm is the potential for small sample-sizes for the training sets with respect to the dimensionality of the data. We will vary the value of N and observe its effect to determine a value of N that provides acceptable detection performance. Small values of N lead to faster training times. In contrast, large values of N are not desirable since the SVDD training time increases quadratically with N.[14]

In the previous section, we saw that, for our algorithm that finds the optimal s parameter, an SVDD has to be trained and applied to an independent validation set for each candidate s value. The length of time it takes to train each of these SVDDs is proportional to the number of target training signatures N that we generate. If we can train the SVDD with a relatively small number of signatures, we can substantially decrease the total training time.

2. Experiments

To assess the impact of *N* on SVDD detection results, we will do the following:

1. Load one of four HSI data cubes—*CASI_urban_10*, *CASI_veg_10*, *HYDICE_urban_10*, *HYDICE_veg_10*. We will use the data cubes containing targets generated by the adaptive Gaussian model.
2. Generate the independent validation set consisting of 100 generated target signatures and 8000 randomly selected background signatures. The 100 generated target signatures are also generated using the adaptive Gaussian model with $SNR = 10\,dB$ to match the SNR of the scene targets.
3. Generate *N* SVDD training set signatures according to the adaptive Gaussian model with $SNR = 10\,dB$ to match the SNR of the scene targets. We will vary the value of *N* from 10 samples to 200 samples in increments of 5. Thus, the 5 new samples that are generated at each iteration are appended to the previously *N* generated samples.
4. Train an SVDD using each value of *N* and apply it to the data cube. The optimal *s* parameter is chosen using our algorithm from the previous section.

For each data cube, we have plotted the total training time as a function of *N*. We have also plotted the F-stat as a function of *N* for the validation set and the scene.

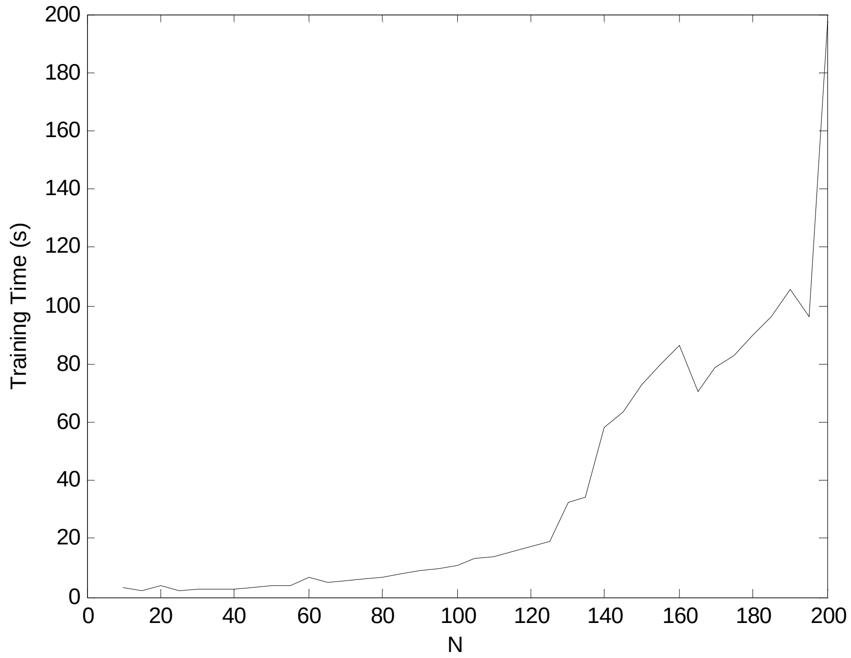


Figure 64. SVDD training time as a function of *N* for the *CASI_urban_10* scene.

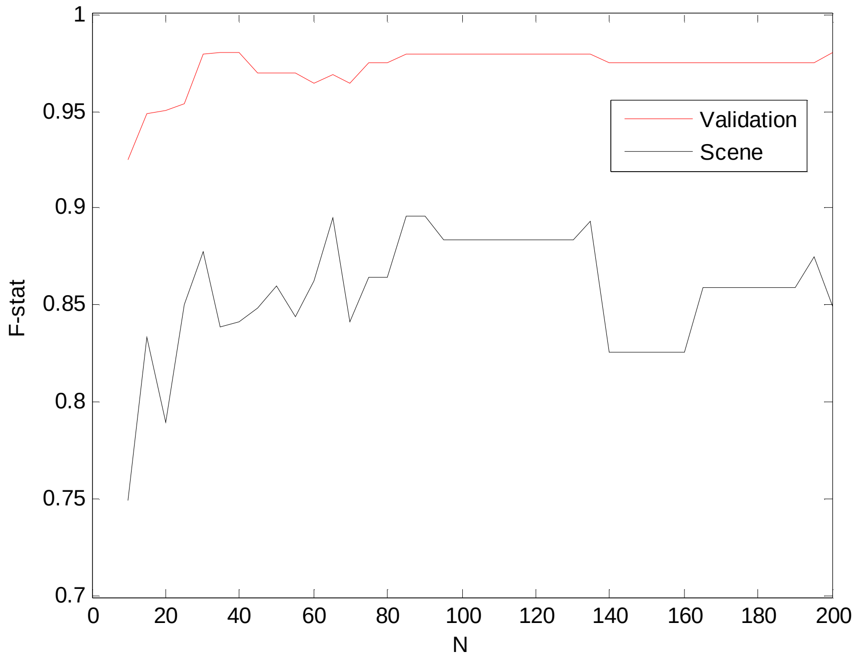


Figure 65. Behavior of F-stat as a function of *N* for the *CASI_urban_10* scene.

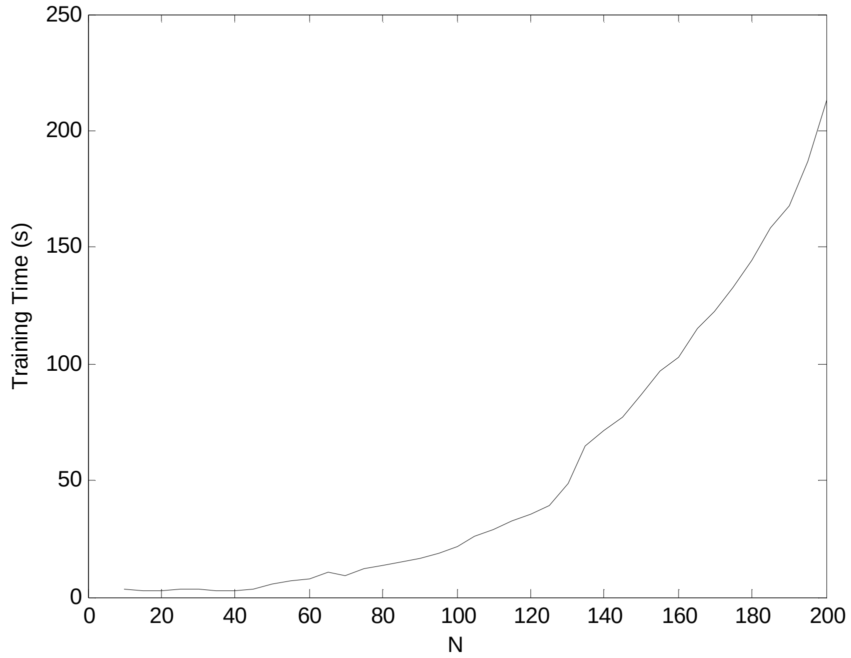


Figure 66. SVDD training time as a function of *N* for the *CASI_veg_10* scene.

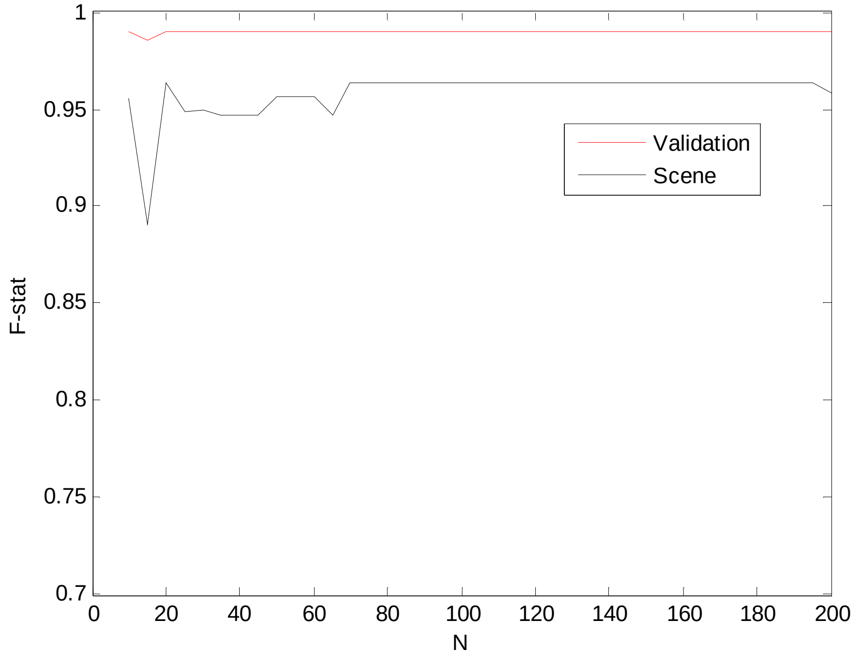


Figure 67. Behavior of F-stat as a function of *N* for the *CASI_veg_10* scene.

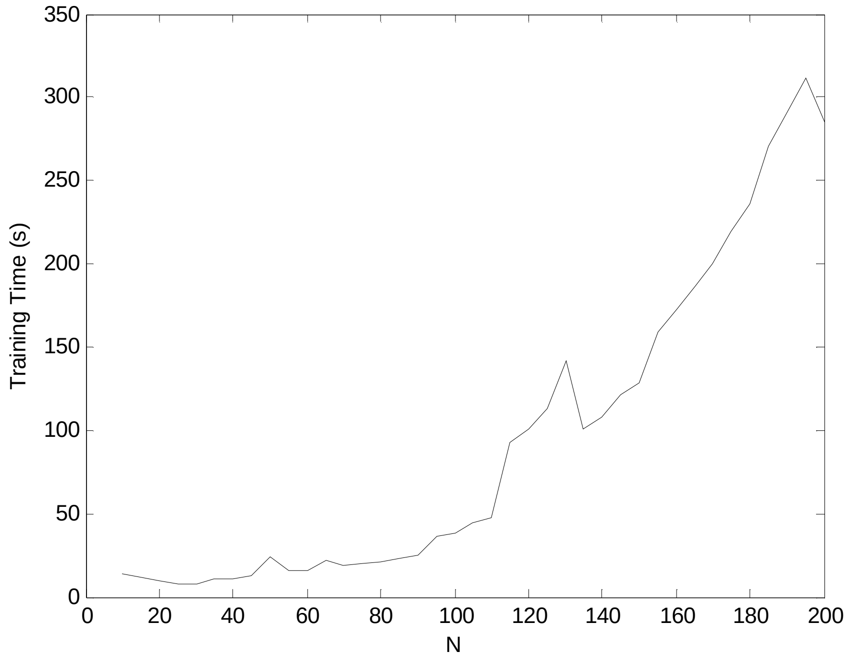


Figure 68. SVDD training time as a function of *N* for the *HYDICE_urban_10* scene.

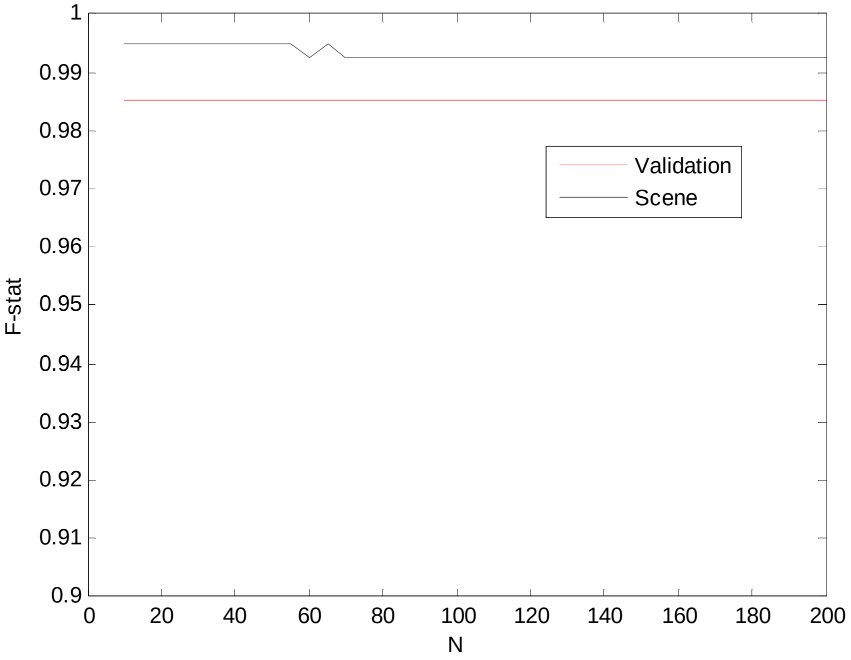


Figure 69. Behavior of F-stat as a function of *N* for the *HYDICE_urban_10* scene.

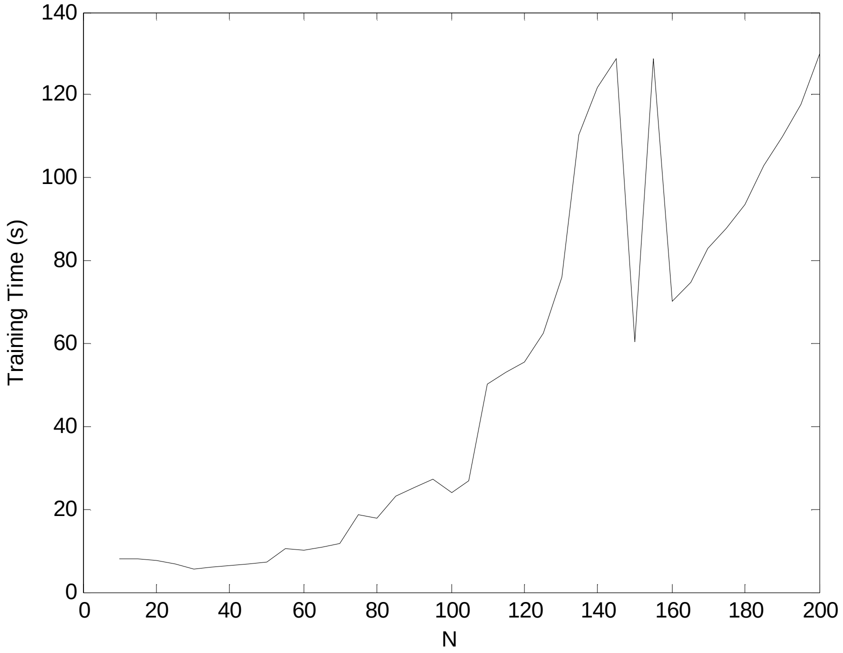


Figure 70. SVDD training time as a function of *N* for the *HYDICE_veg_10* scene.

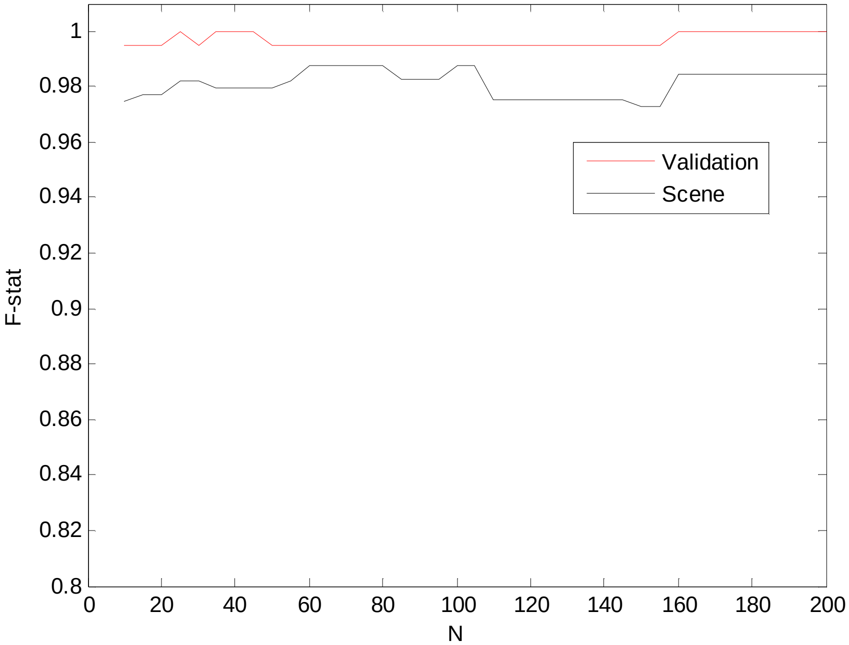


Figure 71. Behavior of F-stat as a function of *N* for the *HYDICE_veg_10* scene.

The results are shown in Figure 64 - Figure 71. As the plots for the training times show, we can see the quadratic behavior as N is varied. For example, in Figure 64, with $N = 90$, the total training time is ~8.78 s; for $N = 135$, the total training time is ~34.22 s. Hence, a 50% increase in the number of training samples led to almost a 400% increase in the training time. This trend is uniformly exhibited across all four scenes, except in a couple of instances where the optimization of s allows for some of the training times to deviate from the norm.

Let us examine the behavior of the F-stat objective function for the CASI data cubes. For the *CASI_urban_10* data cube in Figure 65, the F-stat exhibits similar behavior for the validation set and scene. In general, the F-stat increases from the initial training set size of 10 samples to a maximum value around 85 samples. Beyond 85 samples, the F-stat remains fairly constant for the validation set at a value of ~0.98. For the scene, the F-stat tapers around a maximum value of ~0.89 and does not increase beyond 85 samples. For the *CASI_veg_10* scene in Figure 67, the F-stat again exhibits similar behavior for the validation set and scene. In both cases, using the smallest training set size of 10 samples yields an F-stat value near the maximum. Also, at 15 samples, the F-stat drops for both cases. Beyond 70 samples, the F-stat reaches a constant maximum value for the validation set and scene.

Let us now look at the F-stat behavior for the HYDICE data cubes. As Figure 69 shows, the F-stat remains nearly constant for all values of N in both the validation set and scene for *HYDICE_urban_10*. For the validation set, the F-stat remains at a constant value of ~0.985. For the scene, beyond 70 samples, the F-stat remains constant at

~0.993—just shy of its maximum value of ~0.995. For the *HYDICE_veg_10* scene in Figure 71, the F-stat on the validation set remains nearly constant as the values only vary between ~0.995 and 1. For the scene, the F-stat values range between ~0.972 and ~0.987. Beyond 60 samples, the F-stat for the scene does not show any improvement.

3. Conclusion

In the CASI data cubes, the minimum number of samples required to yield acceptable detection performance is 85 and 70 for the urban and vegetative scenery, respectively. The urban scene showed the most fluctuation in the F-stat value as a function of *N*, indicating that this scene is more sensitive to the variation of *N*. This is consistent with what we saw earlier in the optimization of s in that the separability between the target and background signatures for the urban CASI scenery is less compared to the other scenes. For the HYDICE scenery, the results demonstrate that the choice of *N* is even less critical with the behavior of the F-stat remaining nearly constant across all values of *N*. This lines up with our intuition from earlier since we saw that the HYDICE scenery exhibits large separability between target and background signatures.

Our results regarding the selection of N have confirmed our notion that only a relatively small number of training signatures is required with respect to the dimensionality (number of bands), thus eliminating the *Hughes phenomenon* problem. Based on our results, we have seen that values as low as N = 20 samples and values as high as N = 90 samples are sufficient for our SVDD-based detection scheme, depending on the separability between target and background signatures.

E. Impact of the Value of σ^2

1. Introduction

In the generation of training signatures for the target class, we must choose a value of σ^2, regardless of the choice of target variability model. We mentioned that the constant σ^2 is the per-band variance that is introduced to control the level of variability in the generated target signatures. Small values of σ^2 may prove insufficient for capturing the true spectral variability of the target class; in contrast, large values of σ^2 could allow background signatures to be captured by the target SVDD, thus generating undesired false positives. In this section, we will use different values of σ^2 to explore the impact on the detection process for both target variability models.

2. Experiments

To explore the impact of σ^2 on the SVDD detection results, we will do the following:

1. Load one of four HSI data cubes—*CASI_urban_12*, *CASI_veg_12*, *HYDICE_urban_12*, *HYDICE_veg_12*, corresponding to both target variability models.
2. Generate 100 SVDD training set signatures according to both target variability models. We will vary the value of σ^2 to achieve SNRs of 8 dB, 10 dB, 12 dB, and 15 dB.

3. Generate the independent validation set consisting of 100 generated target signatures and 8000 randomly selected background signatures. The 100 target signatures are generated using the corresponding target variability model with values of σ^2 to achieve SNRs of 8 dB, 10 dB, 12 dB, and 15 dB. Recall that this validation set is used to optimize the *s* parameter using the algorithm developed earlier.
4. Train an SVDD using each value of σ^2 and apply it to the data cube.

For each data cube, we have provided the detection results in the form of confusion matrices according to the respective value of σ^2 used to train the SVDD. Table 9 - Table 24 show the impact of σ^2 for all four scenes of the simple Gaussian model. Table 9 - Table 12 show the impact of σ^2 on the *CASI_urban_12* data cube. As these tables show, the choice of σ^2 has a significant bearing on the detection results. As the value of σ^2 increases (i.e., the SNR decreases), the number of FPs and TPs increases. In the scenario shown in Table 12 where the target training set variability has been underestimated, the TPR is terribly low in comparison with the other scenarios. As Table 13 - Table 16 show, the choice of σ^2 does not dramatically affect the detection results for *CASI_veg_12*. The number of FPs is slightly increased when the target training set variability is significantly overestimated as shown in Table 13.

Table 9. Results on *CASI_urban_12* with SNR = 8 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	38458	8
Predicted Target	1342	192

Table 10. Results on *CASI_urban_12* with SNR = 10 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39533	14
Predicted Target	267	186

Table 11. Results on *CASI_urban_12* with SNR = 12 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39727	16
Predicted Target	73	184

Table 12. Results on *CASI_urban_12* with SNR = 15 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	176
Predicted Target	2	24

Table 13. Results on *CASI_veg_12* with SNR = 8 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39760	13
Predicted Target	40	187

Table 14. Results on *CASI_veg_12* with SNR = 10 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39791	13
Predicted Target	9	187

Table 15. Results on *CASI_veg_12* with SNR = 12 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39791	13
Predicted Target	9	187

Table 16. Results on *CASI_veg_12* with SNR = 15 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39792	14
Predicted Target	8	186

Table 17. Results on *HYDICE_urban_12* with SNR = 8 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	0
Predicted Target	2	200

Table 18. Results on *HYDICE_urban_12* with SNR = 10 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	0
Predicted Target	2	200

Table 19. Results on *HYDICE_urban_12* with SNR = 12 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	0
Predicted Target	2	200

Table 20. Results on *HYDICE_urban_12* with SNR = 15 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	0
Predicted Target	2	200

Table 21. Results on *HYDICE_veg_12* with SNR = 8 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39791	3
Predicted Target	9	197

Table 22. Results on *HYDICE_veg_12* with SNR = 10 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39800	6
Predicted Target	0	194

Table 23. Results on *HYDICE_veg_12* with SNR = 12 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39800	6
Predicted Target	0	194

Table 24. Results on *HYDICE_veg_12* with SNR = 15 dB for simple Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	5
Predicted Target	2	195

The impact of σ^2 on *HYDICE_urban_12* is shown in Table 17 - Table 20. As the tables show, the different values of σ^2 have no effect on the detection results. With respect to the *HYDICE_veg_12* data cube, Table 21 - Table 24 show that the choice of σ^2 has only a slight effect on the detection results.

Table 25 - Table 40 show the impact of σ^2 for all four scenes using the adaptive Gaussian model. Table 25 - Table 28 show the impact of σ^2 on *CASI_urban_12*. As these tables show, the choice of σ^2 has a significant bearing on the detection results. As the value of σ^2 increases (i.e., the SNR decreases), the number of both FPs and TPs increases. As Table 29 - Table 32 show, the choice of σ^2 does not dramatically affect the detection results for *CASI_veg_12* as it did with *CASI_urban_12*.

The impact of σ^2 on *HYDICE_urban_12* is shown in Table 33 - Table 36. As these tables show, the variation of σ^2 has a negligible effect on the detection results. With respect to the *HYDICE_veg_12* data cube, Table 37 - Table 40 show that the choice of σ^2 can increase the number of FPs if it is chosen too high.

Table 25. Results on *CASI_urban_12* with SNR = 8 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39623	10
Predicted Target	177	190

Table 26. Results on *CASI_urban_12* with SNR = 10 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39736	14
Predicted Target	64	186

Table 27. Results on *CASI_urban_12* with SNR = 12 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39785	15
Predicted Target	15	185

Table 28. Results on *CASI_urban_12* with SNR = 15 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39798	30
Predicted Target	2	170

Table 29. Results on *CASI_veg_12* with SNR = 8 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39783	13
Predicted Target	17	187

Table 30. Results on *CASI_veg_12* with SNR = 10 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39799	14
Predicted Target	1	186

Table 31. Results on *CASI_veg_12* with SNR = 12 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39781	13
Predicted Target	19	187

Table 32. Results on *CASI_veg_12* with SNR = 15 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39789	13
Predicted Target	11	187

Table 33. Results on *HYDICE_urban_12* with SNR = 8 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39796	0
Predicted Target	4	200

Table 34. Results on *HYDICE_urban_12* with SNR = 10 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39796	0
Predicted Target	4	200

Table 35. Results on *HYDICE_urban_12* with SNR = 12 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39797	0
Predicted Target	3	200

Table 36. Results on *HYDICE_urban_12* with SNR = 15 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39797	0
Predicted Target	4	200

Table 37. Results on *HYDICE_veg_12* with SNR = 8 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39766	1
Predicted Target	34	199

Table 38. Results on *HYDICE_veg_12* with SNR = 10 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39792	2
Predicted Target	8	198

Table 39. Results on *HYDICE_veg_12* with SNR = 12 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39793	2
Predicted Target	7	198

Table 40. Results on *HYDICE_veg_12* with SNR = 15 dB for adaptive Gaussian model.

	Actual Background	*Actual Target*
Predicted Background	39792	3
Predicted Target	8	197

3. Conclusion

As we have seen, in some cases, the choice of σ^2 is critical for robust SVDD detection performance. In other cases, it has a marginal impact. As we saw in section C regarding the optimization of the *s* parameter, the separation between the target class signatures and the signatures of the background classes is crucial. As we saw earlier, the separation between target and background signatures in the urban CASI scenery was less than the other scenes. The results shown in Table 9 - Table 12 and Table 25 - Table 28 confirm this notion. As σ^2 is increased for the training set, the SVDD expands, allowing more objects (both target and background signatures) to enter the sphere.

In the other scenes, the larger separation between target and background signatures allows for the results to be less sensitive to the choice of σ^2. As shown in Table 3 in section C, we saw that the urban HYDICE scenery had the widest range of *s* values that yielded a global maximum. This was an indication that the separation between target and background signatures in this scene was relatively large. The results shown in Table 17 - Table 20 and Table 33 - Table 36 confirm this notion as the choice of σ^2 has virtually no effect on the detection results.

F. Decision-level Fusion Via the SVDD and Parameter Variation

1. Introduction

As we saw in the last section, the choice of σ^2 for the target training set has an impact on the detection results, depending on the separation between target and background signatures. Because separability varies across all types of scenes and targets, the selection of an appropriate σ^2 is difficult. Furthermore, we do not know *a priori* the severity of the variability that is to be expected of the targets that may be present in the scenes.

In this section, we will investigate the use of decision-level fusion for alleviating the problem of choosing a proper value of σ^2. The next section provides the theory behind the decision-level fusion techniques we will use. Section 3 provides some experiments using decision-level fusion on some of our HSI data, and section 4 provides our conclusions regarding the results.

2. Theoretical Background

A recent trend in the design of classification systems has been the use of information fusion, or the combination of classifiers. For data sets with complex and highly nonlinear decision boundaries, a single classifier may not yield a desirable classification rate. In the PR literature, it has been observed that the sets of misclassified patterns do not overlap. This implies that different classifiers potentially offer complementary information about the patterns to be classified, and their combination could prove useful.[38] The combination of classifiers may occur at either the data level, feature level, or decision level stages of the classifier.[27] Each classifier may use the same representation (features) of the input pattern or its own representation of the input pattern. An example of combination in the same feature space would be neural network designs having the same architecture but different sets of weights obtained through a variety of training strategies.[38]

A handful of strategies exist for combining the outputs of each individual classifier to form a single decision.[38,39] In *decision-level fusion*, the outputs (labels) of each classifier are combined using a specific rule, yielding a single label for each input pattern. In the context of target detection, two possible labels exist—a 0 indicates that no target is present (class ω_0), while a 1 indicates that a target is present (class ω_1). The traditional rules for decision-level fusion include the AND, OR, and majority vote (MV) schemes.[27] Using the AND rule, class ω_1 (target) is declared if all classifiers predict class ω_1; otherwise, class ω_0 (background) is declared. With the OR rule, class ω_1 (target) is declared if at least one of the classifiers predicts class ω_1; otherwise, class ω_0 (background) is declared. Using the MV scheme, class ω_1 (target) is declared if a majority of the classifiers predicts class ω_1; otherwise, class ω_0 (background) is declared. In the MV scheme, each classifier has equal bearing on the final decision.

We intend to extend the concept of decision-level fusion of classifiers by investigating the combination of detectors to facilitate SVDD-based target detection. It is reasonable to expect that detector fusion will provide robust detection, given that the fusion of classifiers has had successful application in hyperspectral and remotely sensed imagery.[40-42]

3. Experiments

In this section, we will use decision-level fusion concepts by training a collection of SVDDs with different variance parameters σ^2 for each of the target training sets. The following enumerate the steps we have taken:

1. Load one of eight HSI data cubes—*CASI_urban_10, CASI_urban_12, CASI_veg_10, CASI_veg_12, HYDICE_urban_10, HYDICE_urban_12, HYDICE_veg_10, HYDICE_veg_12*. We will use the data cubes corresponding to both target variability models.
2. Generate three SVDD training sets, each consisting of 100 signatures according to both target variability models. Each SVDD will have a value of σ^2 to achieve SNRs of 7 dB, 9 dB, and 11 dB. These SVDDs will be respectively referred to as *SVDD_7*, *SVDD_9*, and *SVDD_11*.
3. Generate three independent validation sets consisting of 100 generated target signatures and 8000 randomly selected background signatures. The 100 generated target signatures are generated using the corresponding target variability model with values of σ^2 to achieve SNRs of 7 dB, 9 dB, and 11 dB. Each validation set is used to optimize the *s* parameter of the respective SVDD training set. The values of 7, 9, and 11 were purposely selected to explore the

impact of training sets with different levels of variability that do not identically coincide with the SNRs of the inserted scene targets.

4. Train all three SVDDs and apply them to each data cube.
5. Capture the detection results of each individual SVDD as well as the decision-level fusion detection results using the AND, OR, and MV schemes.

For sake of brevity, we have presented the results by listing only the FPs and TPs for each detection scenario since the other two quantities (TN and FN) are easily calculated from the FPs and TPs. Table 41 - Table 48 provide the results for the eight data cubes corresponding to the simple Gaussian model, while Table 49 - Table 56 show the results for the adaptive Gaussian model.

In these experiments, we are using data cubes containing inserted targets with SNRs of 10 dB and 12 dB. The 10 dB scene is used so that we may assess the decision fusion results when the target variability in the scene (i.e., 10 dB) lies within the range of the target variability of the individual training sets (7 dB, 9 dB, and 11 dB). The 12 dB scene is used so that we may assess the decision fusion results when the target variability in the scene (i.e., 12 dB) lies outside the range of the target variability of the individual training sets.

Table 41. Decision-level fusion results for *CASI_urban_10* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	2959	682	94	94	2959	682
TPs	194	191	92	92	194	191

Table 42. Decision-level fusion results for *CASI_urban_12* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	2962	724	124	124	2962	724
TPs	194	191	185	185	194	191

Table 43. Decision-level fusion results for *CASI_veg_10* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	94	9	9	9	94	9
TPs	187	187	187	187	187	187

Table 44. Decision-level fusion results for *CASI_veg_12* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	119	9	33	9	119	33
TPs	187	187	187	187	187	187

Table 45. Decision-level fusion results for *HYDICE_urban_10* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	8	8	7	7	8	8
TPs	200	200	200	200	200	200

Table 46. Decision-level fusion results for *HYDICE_urban_12* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	2	2	2	2	2	2
TPs	200	200	200	200	200	200

Table 47. Decision-level fusion results for *HYDICE_veg_10* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	161	0	0	0	161	0
TPs	198	194	194	194	198	194

Table 48. Decision-level fusion results for *HYDICE_veg_12* scene for simple Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	163	2	0	0	163	2
TPs	198	195	194	194	198	195

As Table 41 and Table 42 show, the level of variability in the target training set has a significant impact on the detection results for *CASI_urban_10* and *CASI_urban_12*. Notice in Table 41 how using an underestimated training set with *SVDD_11* causes a sharp decline in the number of TPs. Observe how the AND decision rule leads to the lowest number of FPs and TPs. In fact, it provides identical results to the *SVDD_11* training set. In contrast, the OR decision rule yields both the highest number of FPs and TPs—identical to the *SVDD_7* training set. The MV scheme provides a compromise between the two, yielding the same results as *SVDD_9*. For the *CASI_urban_12* data cube in Table 42, the AND rule yields the best decision-level fusion performance, mimicking the results of *SVDD_11*.

As we saw earlier, the vegetative CASI scenery is not as difficult and, hence, the detection results do not significantly vary between the individual training sets. Once again, for the vegetative CASI scenery results in Table 43 and Table 44, the AND decision rule yields the lowest number of FPs and TPs, the OR decision rule yields the highest number of FPs and TPs, and the MV scheme provides a compromise between the FPs of the two.

With respect to the urban HYDICE scenery, Table 45 and Table 46 clearly confirm the results from the previous section that the variation of σ^2 has a negligible impact on the detection results because of the relatively large separation between target and background class signatures. Accordingly, the decision-level fusion results using all three schemes are identical. The results shown in Table 47 and Table 48 for the vegetative HYDICE scenery echo that of the other scenes in that the MV scheme provides a robust balance on detection performance when compared to the AND and OR rules for fusion.

With respect to the results in Table 49 - Table 56 using the adaptive Gaussian model, similar trends are observed. The urban CASI scenery shows the largest fluctuation in detection results as a function of the training set used, while the urban HYDICE scenes are virtually unaffected by the level of variability in the training set. Regarding the decision fusion results for the adaptive Gaussian model, the AND decision rule yields the lowest number of FPs and TPs, the OR decision rule yields the highest number of FPs and TPs, and the MV scheme provides a balance between the two.

Table 49. Decision-level fusion results for *CASI_urban_10* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	297	57	26	26	297	57
TPs	192	182	179	179	192	182

Table 50. Decision-level fusion results for *CASI_urban_12* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	314	121	29	29	314	121
TPs	192	189	186	186	192	189

Table 51. Decision-level fusion results for *CASI_veg_10* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	25	1	0	0	25	1
TPs	187	186	186	186	187	186

Table 52. Decision-level fusion results for *CASI_veg_12* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	26	12	9	8	26	13
TPs	187	186	186	186	187	186

Table 53. Decision-level fusion results for *HYDICE_urban_10* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	2	2	2	2	2	2
TPs	200	200	200	200	200	200

Table 54. Decision-level fusion results for *HYDICE_urban_12* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	12	4	4	4	12	4
TPs	200	200	200	200	200	200

Table 55. Decision-level fusion results for *HYDICE_veg_10* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	61	5	0	0	61	5
TPs	199	198	192	192	199	198

Table 56. Decision-level fusion results for *HYDICE_veg_12* scene for adaptive Gaussian model.

	SVDD_7	*SVDD_9*	*SVDD_11*	*AND*	*OR*	*MV*
FPs	72	10	9	9	72	10
TPs	199	198	198	198	199	198

4. Conclusion

The experiments in this section have demonstrated the utility of decision-level fusion for our SVDD-based target detection scheme. As we mentioned in the introduction, the purpose of our decision-level fusion application was the potential for achieving robust detection performance without having to determine beforehand the level of variability to be expected in the target class. In all the data cubes, the AND rule yielded the lowest numbers of FPs and TPs. This lines up with our intuition since the AND rule is conservative in the sense that it only labels a signature as target if *all* of the detectors label it as such. In contrast, the OR rule yielded the highest numbers of both FPs and TPs for all data cubes. Once again, this is intuitively expected since the OR rule is liberal in the sense that it labels a signature as target if at least *one* of the detectors labels it as such. The MV scheme was able to maintain a relatively low number of FPs with a high number of TPs for all data cubes, thus providing the desired balance between the AND and OR rules. Accordingly, we will implement the MV fusion rule in all future experiments related to SVDD-based target detection. Chapter VII will provide the detection results of our SVDD-based detection scheme on all data cubes corresponding to both target variability models.

CHAPTER VI

SPECTRAL FRINGE-ADJUSTED JOINT TRANSFORM CORRELATION

A. Introduction

In this chapter, we present the SFJTC technique and explore the use of the DWT as a pre-processing tool for SFJTC-based target detection. Unlike the kernel-based SVDD approach to target detection that we formulated in Chapter V, the SFJTC technique is a deterministic approach to target detection. Each pixel signature in the hyperspectral data cube is compared to the target signature to determine whether a match exists. The basic premise of our work is that select coefficients generated from a desired level of the DWT decomposition of the signatures should be used in place of the original signatures for improved SFJTC-based detection.

Section B provides the mathematical framework of the SFJTC technique. In section C, we describe the detection statistic formulation used to quantify the detection process. Section D reviews the wavelet transform and discusses current applications of it in the hyperspectral literature. In section E, we will explore how use of the DWT coefficients impacts the detection results on our HSI scenery. Section F presents our supervised training algorithm for selecting an optimal set of DWT coefficients for a particular target and scene prior to detection.

B. Theoretical Background

In a JTC technique, when the input pixel contains only one spectral signature, the correlation output produced includes three terms: a strong DC or zero-order term at the center, flanked by a pair of cross-correlation terms in the correlation array. The FJTC technique provides enhanced correlation performance, where the joint power spectrum is first modified by eliminating the zero-order terms and then multiplied by a real-valued filter known as the *fringe-adjusted filter* (FAF). Accordingly, this technique overcomes the problem of zero-order correlation terms and produces sharp delta-function-like peaks.[43]

In the SFJTC technique, the reference, $\boldsymbol{r}(x)$, and input, $\boldsymbol{t}(x)$, spectral signatures are situated on the x-axis with a distance of separation of $2x_0$. The input joint signature $\boldsymbol{f}(x)$ can be expressed as the follows:

$$\boldsymbol{f}(x) = \boldsymbol{r}(x + x_0) + \boldsymbol{t}(x - x_0) \tag{6.1}$$

After applying the Fourier transform to the expression in (6.1), we obtain the following expression:

$$F(u) = |R(u)| \exp[j\phi_r(u)] \exp(jux_0) + |T(u)| \exp[j\phi_t(u)] \exp(-jux_0) \tag{6.2}$$

where $|R(u)|$ and $|T(u)|$ are the amplitudes, and $\phi_r(u)$ and $\phi_t(u)$ are the phases of the Fourier transforms of $\boldsymbol{r}$ and $\boldsymbol{t}$, respectively, and u is the frequency-domain variable. If we take the magnitude of $F(u)$, we obtain the *joint power spectrum* (JPS) as given by the following:

$$\begin{aligned}\left|F(u)\right|^2 = \left|R(u)\right|^2 + \left|T(u)\right|^2 + \left|R(u)\right|\left|T(u)\right| \ \exp[\,j\{\phi_r(u) - \phi_t(u) + 2ux_0\}] \\ + \left|R(u)\right| \ \left|T(u)\right| \exp[\,j\{\phi_t(u) - \phi_r(u) - 2ux_0\}]\end{aligned} \tag{6.3}$$

In (6.3), the first two terms correspond to the zero-order terms and the last two terms are the desired cross-correlation components between the reference and input signatures. To eliminate the zero-order terms, we include a Fourier plane image subtraction technique[44], where the input-signature-only power spectrum and the reference-signature-only power spectrum are subtracted from the JPS found above. Afterwards, the modified JPS can be expressed as follows:

$$\begin{aligned}\left|I(u)\right|^2 &= \left|F(u)\right|^2 - \left|R(u)\right|^2 - \left|T(u)\right|^2 \\ &= \left|R(u)\right|\left|T(u)\right| \ \exp[\,j\{\phi_r(u) - \phi_t(u) + 2ux_0\}] + \\ &\quad \left|R(u)\right| \ \left|T(u)\right| \exp[\,j\{\phi_t(u) - \phi_r(u) - 2ux_0\}]\end{aligned} \tag{6.4}$$

Classical JTC techniques yield large correlation side lobes and large correlation peak widths, leading to degraded detection performance.[45,46] To provide sharp correlation peaks and small correlation side lobes, the modified JPS in (6.4) is multiplied by the FAF before performing the final inverse Fourier transform. The FAF is characterized by the transfer function defined as follows:

$$H(u) = \frac{A(u)}{B(u) + \left|R(u)\right|^2} \tag{6.5}$$

where $A(u)$ and $B(u)$ are either constants or functions of u. When $A(u) = 1$ and $\left|R(u)\right|^2 \gg B(u)$, the FAF can be approximated as follows:

$$H(u) \approx \frac{1}{\left|R(u)\right|^2} \tag{6.6}$$

The last step of the SFJTC algorithm is an inverse Fourier transform of the filtered, modified JPS, yielding the correlation output:

$$\boldsymbol{C}(x) = F^{-1}\{H(u) \times \left|I(u)\right|^{2}\} \tag{6.7}$$

A block diagram of the SFJTC algorithm is shown in Figure 72.

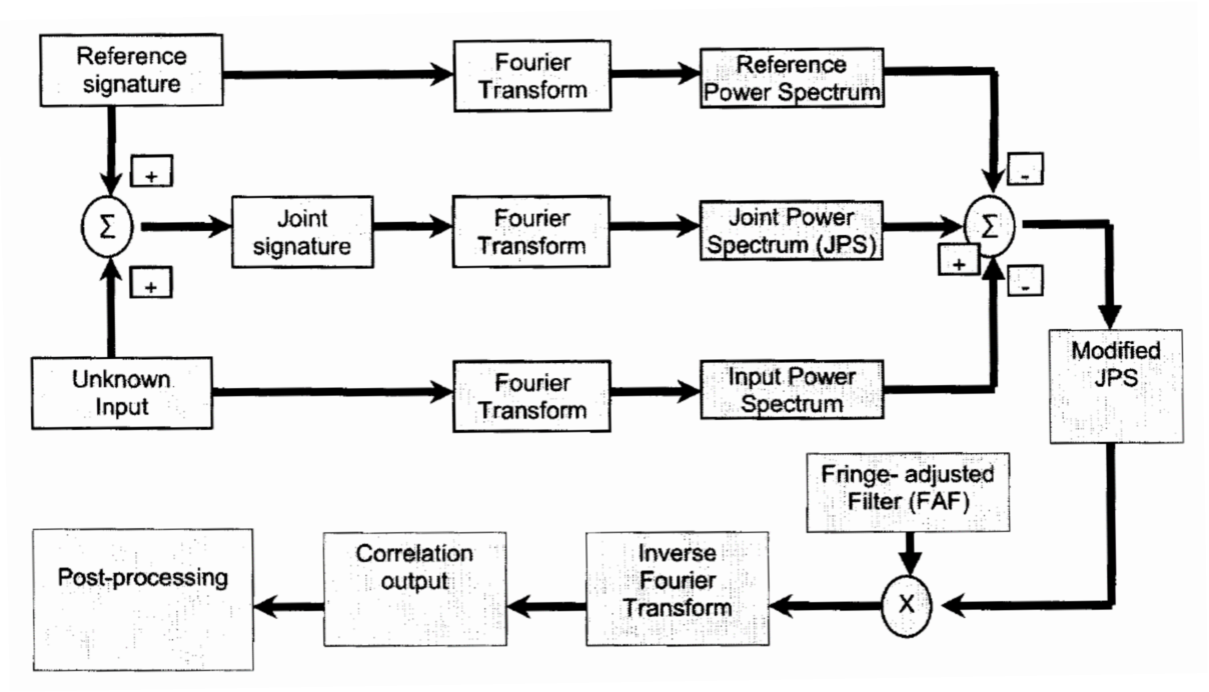


Figure 72. 1-D SFJTC algorithm.

C. Detection Statistic Formulation

As mentioned earlier, a hyperspectral scene is represented as a 3-D data cube, where the first two dimensions are spatial with N_R rows and N_C columns, and the third dimension is spectral with N_B bands. Thus, the scene contains a total of $N_R \times N_C$ pixels.

In our work, we will use a pixel-based processing approach where pixel (i, j) has a corresponding hyperspectral signature vector $\mathbf{s}_{i,j}$ given as follows:

$$\begin{gathered}\mathbf{s}_{i,j} = [s_{i,j}(1), s_{i,j}(2), \ldots s_{i,j}(N_B)]^T \\ (i = 1, 2, \ldots, N_R, j = 1, 2, \ldots, N_C)\end{gathered} \tag{6.8}$$

All of the signature vectors in the hyperspectral image are subject to the SFJTC algorithm, and their correlation outputs $C_{i,j}(x)$ with the target spectral signature are computed. Previous work has indicated that sole use of the largest value of the correlation output, known as the *peak*, does not yield distinguishable correlation performance.[18] As a result, we have computed a decision metric D for each pixel (i, j) defined by the following:

$$D = \left(\frac{peak}{\mu_{clutter}} \right)^{\alpha} \tag{6.9}$$

where $peak = \max(C(x))$ and occurs at location $x = x_{peak}$ in $C(x)$. The *clutter* in the correlation output consists of all values of $C(x)$ *excluding* the value at x_{peak}; it can be interpreted as a measure of noise in the correlation output.[44] In (6.9), $\mu_{clutter}$ represents the average value of the clutter and is given by the following:

$$\mu_{clutter} = \frac{1}{L_C - 1} \sum_{j, j \neq x_{peak}} C(x) \tag{6.10}$$

where L_C is the length of the correlation output vector $C(x)$.The exponent α in (6.9) can be selected to adjust the range of values that D can take. In our work, we have empirically set $\alpha = 0.25$. The detection statistic given by (6.9) is more reliable than the *peak* since it uses more information from the correlation output.

D. The Wavelet Transform

1. Introduction

The wavelet transform is a mathematical tool for performing a multiresolution analysis (MRA) of a signal.[47-49] Wavelets have had many successful applications in data compression[50], noise removal[51], and texture classification[52]. In the signal processing arena, they are commonly used to represent a signal in terms of its global features, yielding the general shape of the signal, and its local features, yielding the details that make the signal unique. The wavelet transform has been developed in both the continuous and discrete domains—the continuous wavelet transform (CWT) and discrete wavelet transform (DWT), respectively.

Wavelets have also been used in the exploitation of HSI data. Bruce and Li have investigated the feasibility of derivative analysis of hyperspectral signatures for computing space-scale images and spectral fingerprints[53]. The application of wavelet-based feature extraction for the classification of agricultural hyperspectral imagery has been studied by Bruce, Koger, and Li.[54] They show that the wavelet transform approach is superior to conventional feature extraction methods in terms of overall classification accuracy. The use of the DWT for dimensionality reduction of HSI data has been investigated by Kaewpijit, Le Moigne, and El-Ghazawi.[55] They show that the DWT is superior to principal components analysis (PCA) for dimensionality reduction and yields better or comparable classification accuracy on HSI data, in addition to being more computationally efficient than PCA. Bruce, Morgan, and Larsen investigate the use of the wavelet coefficients' scalar energies as features for use in a statistical classification system of hyperspectral signals, particularly focusing on the ability to classify subpixel targets.[56]

2. The continuous wavelet transform

The continuous wavelet transform (CWT) of a signal $f(t)$ is an inner product between $f(t)$ and the wavelet function, and is defined as follows:

$$W_{\psi} f(a,b) = \left\langle f, \psi_{a,b} \right\rangle = \int_{-\infty}^{+\infty} f(t) \psi_{a,b}^{*} dt, \qquad (6.11)$$

where $\langle .,. \rangle$ denotes the standard inner product on $L^2(\Re)$. $\psi_{a,b}^{*}$ is the complex conjugate of $\psi_{a,b}$, which is the family of wavelet functions that satisfy the following:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi\left(\frac{t-b}{a}\right), a, b \in \Re, a \neq 0 \qquad (6.12)$$

The function $\psi(t)$ must satisfy the following *admissibility condition* in order to qualify as a wavelet:

$$\int_{-\infty}^{+\infty} \frac{|F(\psi(t))|^2}{|\omega|} d\omega < \infty \qquad (6.13)$$

where $F(\cdot)$ denotes the Fourier transform, and ω is the frequency-domain variable. The admissibility condition implies two things. First, the mother wavelet is oscillatory in nature and has an average value of zero. Second, it exhibits "compact support" and decays exponentially. As a result of the continuous scale and translation parameters a and b, the CWT is highly redundant. To alleviate the redundancy, the wavelet series was introduced, which is simply a discretization of the CWT.[48]

3. The discrete wavelet transform

The DWT is a highly efficient alternative to the CWT that works by discretizing the scale and translation parameters a and b. A computationally fast implementation of the wavelet transform is known as the *Mallat* algorithm, which represents the wavelet basis functions with a filter bank of low-pass and high-pass filters that meet certain constraints.[48] The general form of the 1-D DWT is shown in Figure 73.

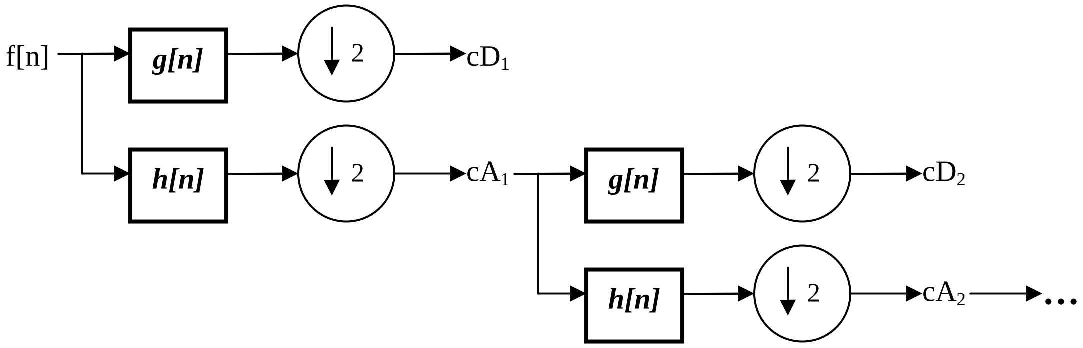


Figure 73. Recursive filtering diagram of the Mallat algorithm for 1-D DWT.

The original signal $\boldsymbol{f}[n]$ is passed through low-pass and high-pass filters with FIRs $\boldsymbol{h}[n]$ and $\boldsymbol{g}[n]$, respectively. The low-pass and high-pass filter coefficients are related by the following:

$$\boldsymbol{g}[L-1-n]=(-1)^n \boldsymbol{h}[n] \tag{6.14}$$

where L is the length of the filters. The filtered signals are then decimated by a factor of two, yielding the coefficients at the first level of decomposition. At each decomposition level, the outputs along the low-pass branch are known as the *approximation* coefficients, while the outputs along the high-pass branch are known as the *detail* coefficients. Multiple levels of decomposition are executed by iteratively repeating the filtering and dyadic decimation procedure on the approximation coefficients. The process is repeated K levels, and the approximation coefficients cA_K and detail coefficients $cD_j, j \in \{1,\ldots,K\}$ are known as the wavelet coefficients. The coefficients are often concatenated into a single vector and denoted by $\boldsymbol{w}$:

$$\boldsymbol{w}=[cA_K\ cD_K\ cD_{K\text{-}1}\ldots cD_1] \tag{6.15}$$

Many different families of wavelets with varying properties are defined.[49] In this work, we use the well-known Daubechies wavelets of order 4 (*db4*).

E. Use of the DWT Coefficients for SFJTC-based Detection

In this section, our goal is to explore any performance improvement by running the SFJTC detection algorithm in the wavelet domain. To do this, we will apply the DWT to the input scene signatures and target signatures and run the SFJTC algorithm using the DWT approximation and detail coefficients of the signatures. By doing this, we can assess the class separability of the DWT coefficients of the signatures relative to the original signatures. We will also explore the effect that the number of DWT decomposition levels has on the SFJTC detection performance. At each decomposition level, we will assess the impact of using particular combinations of the DWT approximation and detail coefficients at that level as well as all of the DWT coefficients. Table 57 shows the various combinations of DWT coefficients for decomposition levels 1 – 3. It should be noted that a large number of DWT combinations exist with respect to the combinations of *individual* coefficients. The combinations in Table 57 are simply the various possibilities of grouping the contiguous subsets of *approximation* and *detail* coefficients at a particular decomposition level.

Table 57. DWT coefficient combinations for decomposition levels 1 – 3.

Level	DWT Coefficient Combinations
1	$cA_{1,}$ cA_1cD_1
2	$cA_{2,}$ $cA_2cD_{1,}$ $cA_2cD_{2,}$ $cA_2cD_2cD_1$
3	$cA_{3,}$ $cA_3cD_{1,}$ $cA_3cD_1cD_{2,}$ $cA_3cD_1cD_3$ $cA_3cD_{2,}$ $cA_3cD_{3,}$ $cA_3cD_3cD_{2,}$ $cA_3cD_3cD_2cD_1$

For a given scene and the associated target signature, the DWT coefficients are generated using the *db4* wavelet at the desired decomposition level *j*. The SFJTC detection algorithm is run using each DWT coefficient combination in Table 57 of the scene and target signatures. To quantify improvement and compare performance in the detection process, ROC curves are generated, and the areas under the ROC curves (AUROCs) are computed. For sake of brevity and without loss of generality, only the

ROC curves corresponding to *CASI_urban_10* and *HYDICE_urban_10* for both target variability models are provided.

We first applied the procedure to the scenes based on the simple Gaussian model. Table 58 and Table 59 provide the AUROCs corresponding to the ROC curves for the detection results on the urban and vegetative CASI scenery, respectively. Similarly, Table 60 and Table 61 provide the same data for the urban and vegetative HYDICE scenery. Each table provides the AUROCs of SFJTC-based detection using the original signatures and the fourteen DWT coefficient combinations listed in Table 57. Table 62 - Table 65 correspond to the scenes based on the adaptive Gaussian model.

The results for the urban CASI scenery are shown in Table 58. Notice that a total of 56 DWT combinations are presented—14 combinations for each of the 4 data cubes. For this scene, 50 out of 56 DWT combinations, or ~89.28% of the combinations, provided either identical or improved AUROCs when compared to the AUROCs using the original signatures. Notice how DWT combination 6 (cA_2cD_2) does consistently poorly across all four data cubes. The ROC curves corresponding to *CASI_urban_10* are shown in Figure 74 and Figure 75.

Table 58. AUROCs of DWT combinations on urban CASI scenery using simple Gaussian model.

Index	Features	*CASI_urban_8*	*CASI_urban_10*	*CASI_urban_12*	*CASI_urban_15*
1	*Original*	0.2653	0.5393	0.9079	0.9947
2	cA_1	0.7059	0.9187	0.9857	0.9963
3	cA_1cD_1	0.4331	0.7478	0.9494	0.9955
4	cA_2	0.7091	0.8465	0.9192	0.9652
5	cA_2cD_1	0.6577	0.8974	0.9789	0.9964
6	cA_2cD_2	0.2246	0.2361	0.2113	0.2416
7	$cA_2cD_2cD_1$	0.6188	0.9032	0.9877	0.9982
8	cA_3	0.9686	0.9910	0.9947	0.9956
9	cA_3cD_1	0.9090	0.9835	0.9956	0.9974
10	$cA_3cD_1cD_2$	0.8752	0.9808	0.9959	0.9976
11	$cA_3cD_1cD_3$	0.9367	0.9874	0.9959	0.9973
12	cA_3cD_2	0.9615	0.9878	0.9956	0.9965
13	cA_3cD_3	0.9740	0.9915	0.9956	0.9963
14	$cA_3cD_3cD_2$	0.9402	0.9898	0.9975	0.9981
15	$cA_3cD_3cD_2cD_1$	0.7472	0.9599	0.9958	0.9987

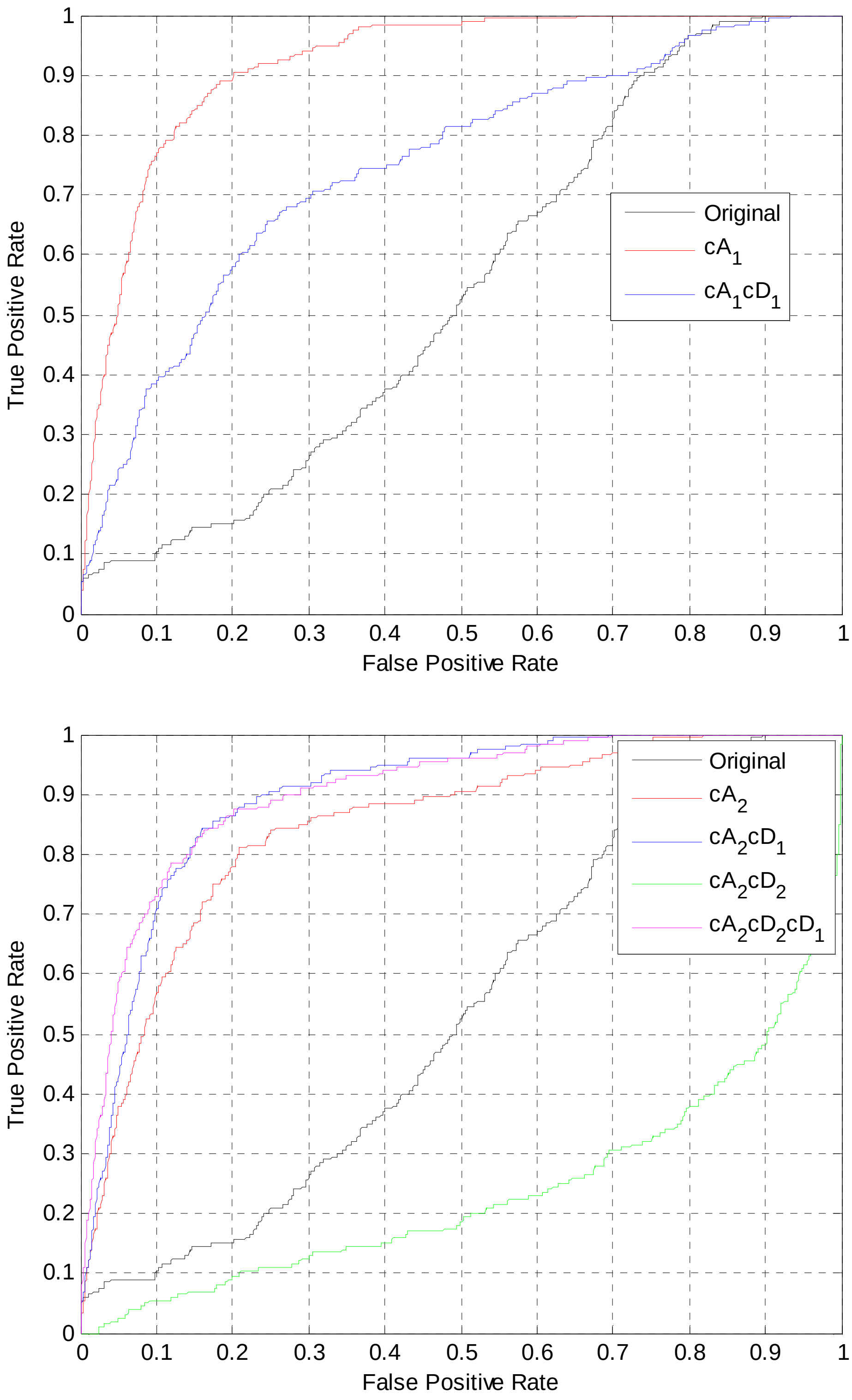
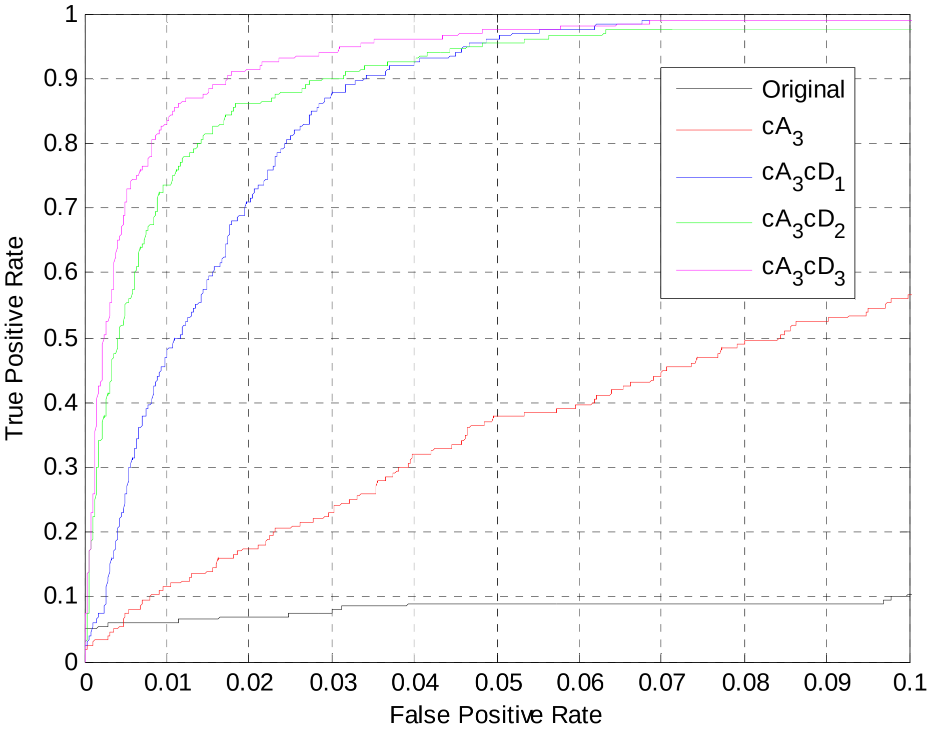


Figure 74. ROC curves of DWT level 1 and 2 coefficients on *CASI_urban_10* with simple Gaussian model.



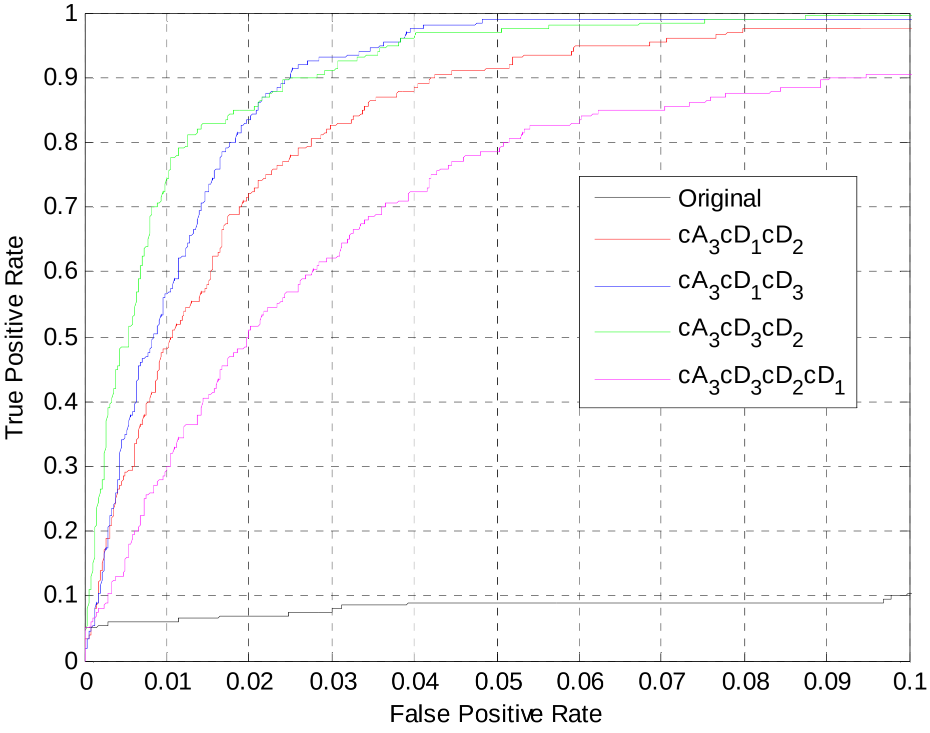


Figure 75. ROC curves of DWT level 3 coefficients on *CASI_urban_10* with simple Gaussian model.

Table 59. AUROCs of DWT combinations on vegetative CASI scenery using simple Gaussian model.

Index	Features	*CASI_veg_8*	*CASI_veg_10*	*CASI_veg_12*	*CASI_veg_15*
1	*Original*	0.3756	0.7779	0.9614	0.9916
2	cA_1	0.8273	0.9537	0.9888	0.9901
3	cA_1cD_1	0.5806	0.8669	0.9714	0.9921
4	cA_2	0.8740	0.9562	0.9866	0.9903
5	cA_2cD_1	0.8610	0.9656	0.9899	0.9909
6	cA_2cD_2	0.2302	0.2678	0.2864	0.2417
7	$cA_2cD_2cD_1$	0.8306	0.9584	0.9915	0.9916
8	cA_3	0.9752	0.9848	0.9852	0.9852
9	cA_3cD_1	0.9630	0.9894	0.9898	0.9898
10	$cA_3cD_1cD_2$	0.9521	0.9890	0.9899	0.9899
11	$cA_3cD_1cD_3$	0.9763	0.9893	0.9894	0.9894
12	cA_3cD_2	0.9838	0.9881	0.9882	0.9882
13	cA_3cD_3	0.9862	0.9881	0.9881	0.9881
14	$cA_3cD_3cD_2$	0.9779	0.9901	0.9901	0.9901
15	$cA_3cD_3cD_2cD_1$	0.9024	0.9876	0.9917	0.9917

The results for the vegetative CASI scenery are shown in Table 59. For this scene, 42 out of 56 DWT combinations, or 75% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. Once again, notice how DWT combination 6 (cA_2cD_2) consistently performs poorly across all four data cubes.

Table 60. AUROCs of DWT combinations on urban HYDICE scenery using simple Gaussian model.

Index	Features	*HYDICE_urban_8*	*HYDICE_urban_10*	*HYDICE_urban_12*	*HYDICE_urban_15*
1	*Original*	0.3459	0.6991	0.9695	0.9999
2	cA_1	0.3881	0.7126	0.9806	1.0000
3	cA_1cD_1	0.4268	0.7218	0.9669	1.0000
4	cA_2	0.7513	0.9308	0.9965	0.9999
5	cA_2cD_1	0.8226	0.9898	0.9997	1.0000
6	cA_2cD_2	0.9463	0.9988	0.9999	1.0000
7	$cA_2cD_2cD_1$	0.7201	0.9505	0.9998	1.0000
8	cA_3	0.9691	0.9948	0.9993	0.9993
9	cA_3cD_1	0.7962	0.9249	0.9734	0.9911
10	$cA_3cD_1cD_2$	0.8514	0.9775	0.9985	0.9988
11	$cA_3cD_1cD_3$	0.9719	0.9999	1.0000	1.0000
12	cA_3cD_2	0.9502	0.9894	0.9969	0.9971
13	cA_3cD_3	0.9450	0.9945	0.9991	0.9994
14	$cA_3cD_3cD_2$	0.9441	0.9930	0.9996	0.9997
15	$cA_3cD_3cD_2cD_1$	0.9303	0.9963	0.9998	0.9998

For the urban HYDICE scenery, Table 60 shows that 49 out of 56 combinations, or 87.5% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. The ROC curves corresponding to *HYDICE_urban_10* are shown in Figure 76 and Figure 77.

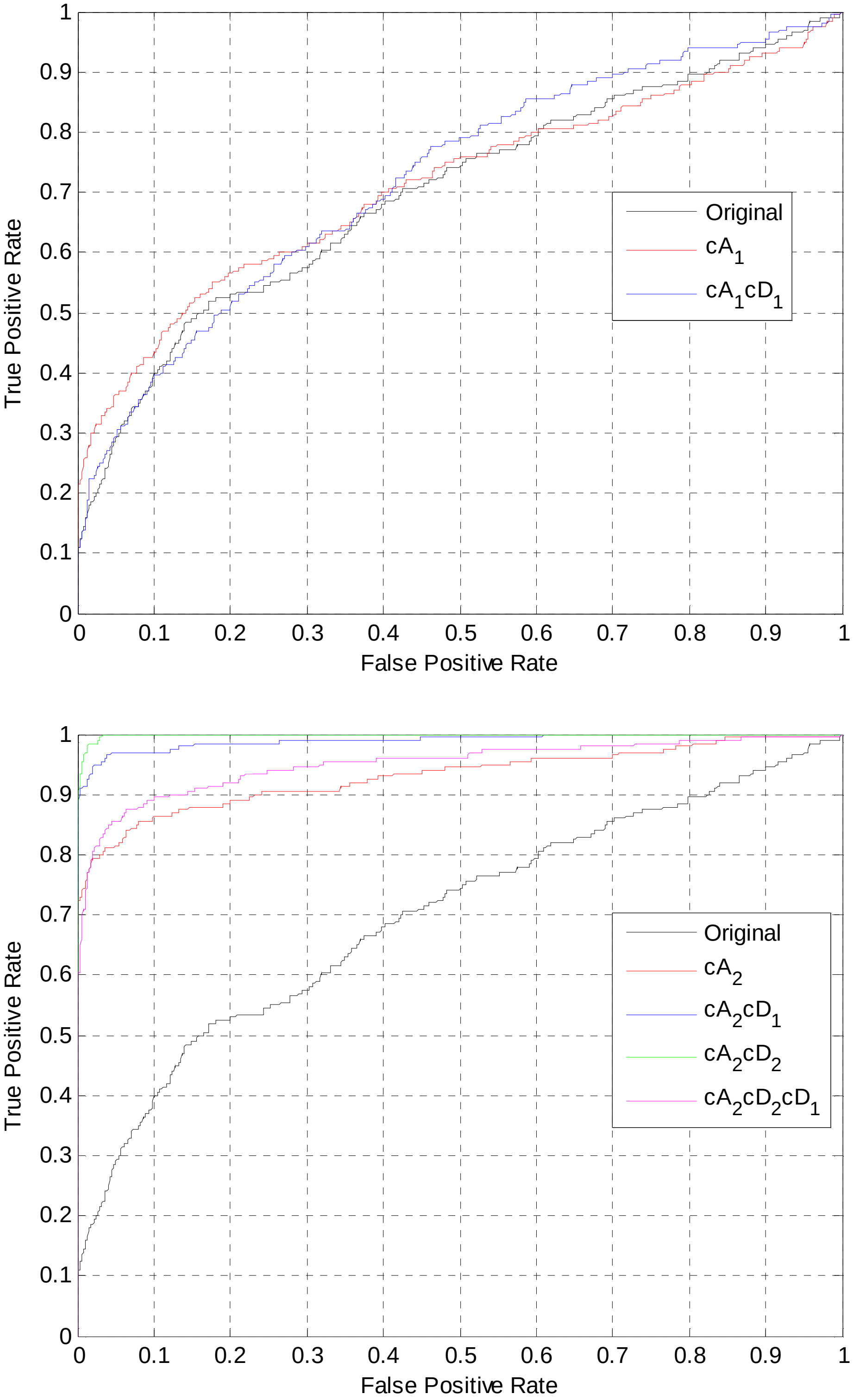
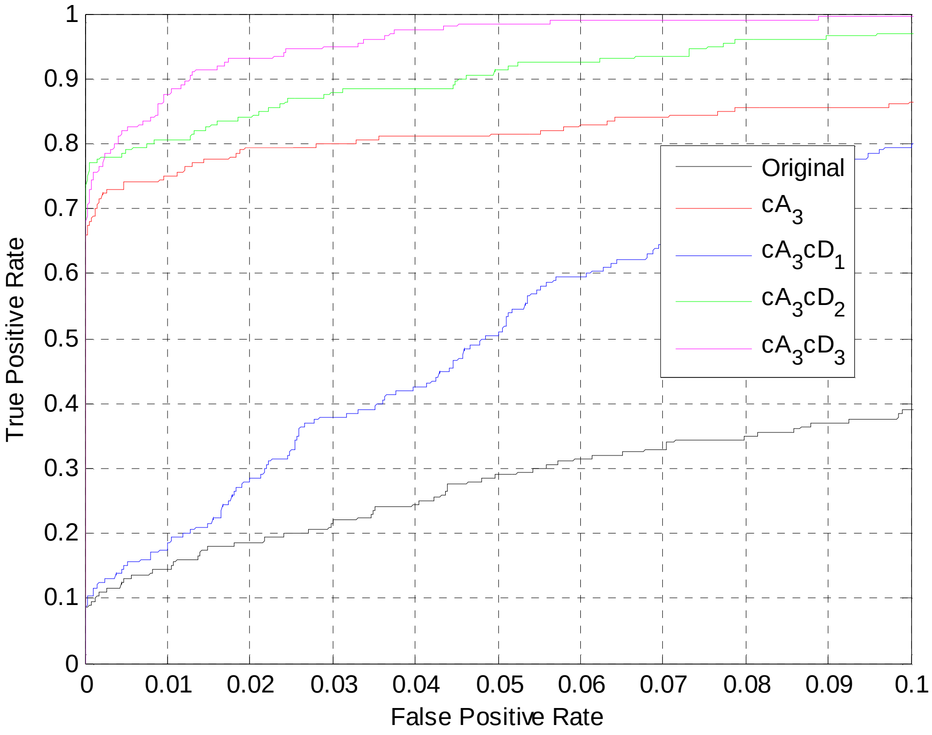


Figure 76. ROC curves of DWT level 1 and 2 coefficients on *HYDICE_urban_10* with adaptive Gaussian model.



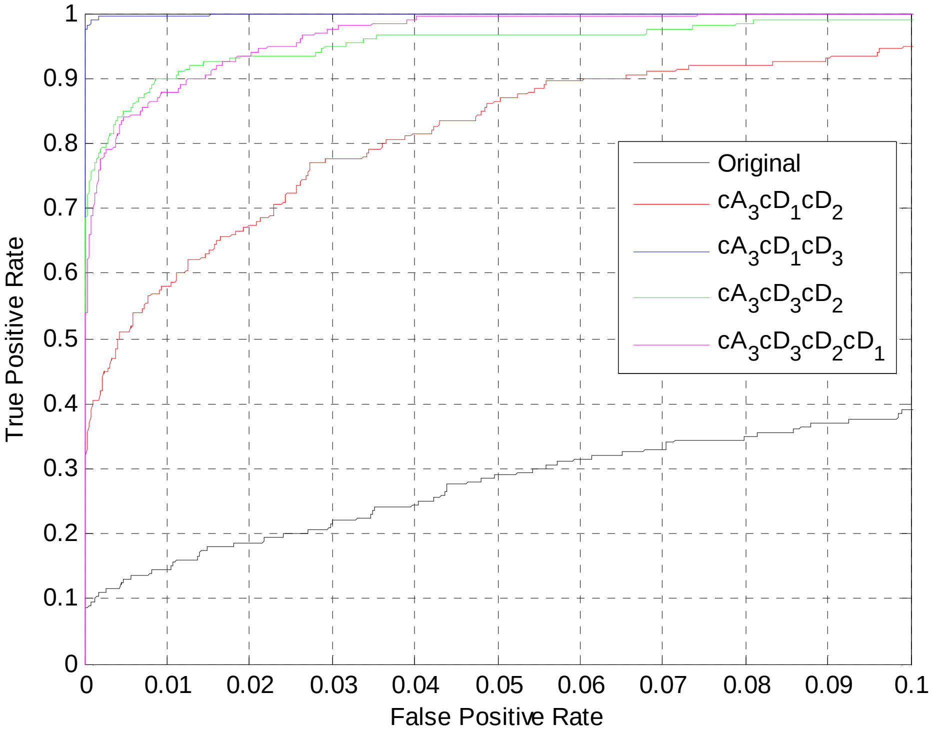


Figure 77. ROC curves of DWT level 3 coefficients on *HYDICE_urban_10* with adaptive Gaussian model.

Table 61. AUROCs of DWT combinations on vegetative HYDICE scenery using simple Gaussian model.

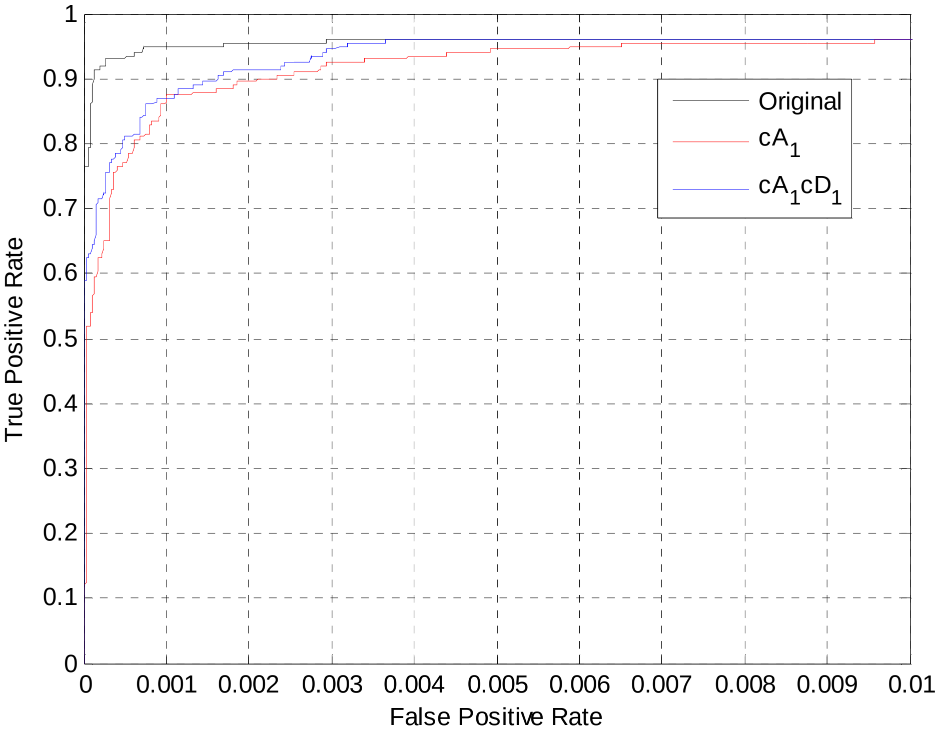
Index	Features	*HYDICE_veg_8*	*HYDICE_veg_10*	*HYDICE_veg_12*	*HYDICE_veg_15*
1	*Original*	0.3179	0.5223	0.9279	0.9986
2	cA_1	0.5796	0.9296	0.9948	1.0000
3	cA_1cD_1	0.4791	0.7887	0.9795	0.9995
4	cA_2	0.7515	0.9049	0.9712	0.9969
5	cA_2cD_1	0.6120	0.9537	0.9999	1.0000
6	cA_2cD_2	0.9189	0.9874	0.9987	0.9988
7	$cA_2cD_2cD_1$	0.8351	0.9705	0.9972	0.9984
8	cA_3	0.9525	0.9926	0.9927	0.9927
9	cA_3cD_1	0.8631	0.9899	0.9994	0.9995
10	$cA_3cD_1cD_2$	0.7964	0.9765	0.9992	0.9995
11	$cA_3cD_1cD_3$	0.9441	0.9986	1.0000	1.0000
12	cA_3cD_2	0.9137	0.9910	0.9927	0.9927
13	cA_3cD_3	0.2398	0.2510	0.2514	0.2467
14	$cA_3cD_3cD_2$	0.9968	1.0000	1.0000	1.0000
15	$cA_3cD_3cD_2cD_1$	0.9761	0.9998	1.0000	1.0000

For the vegetative HYDICE scenery, Table 61 shows that 48 out of 56 combinations, or ~85.71% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. Notice how DWT combination 13 (cA_3cD_3) does consistently poorly across all four data cubes.

Table 62. AUROCs of DWT combinations on urban CASI scenery using adaptive Gaussian model.

Index	Features	*CASI_urban_8*	*CASI_urban_10*	*CASI_urban_12*	*CASI_urban_15*
1	*Original*	0.9967	0.9980	0.9980	0.9980
2	cA_1	0.9935	0.9965	0.9968	0.9968
3	cA_1cD_1	0.9937	0.9969	0.9972	0.9972
4	cA_2	0.9424	0.9615	0.9668	0.9715
5	cA_2cD_1	0.9945	0.9970	0.9972	0.9972
6	cA_2cD_2	0.3185	0.2936	0.2588	0.2148
7	$cA_2cD_2cD_1$	0.9975	0.9984	0.9984	0.9984
8	cA_3	0.9890	0.9945	0.9954	0.9957
9	cA_3cD_1	0.9946	0.9972	0.9975	0.9975
10	$cA_3cD_1cD_2$	0.9950	0.9974	0.9976	0.9976
11	$cA_3cD_1cD_3$	0.9932	0.9970	0.9973	0.9974
12	cA_3cD_2	0.9904	0.9958	0.9965	0.9966
13	cA_3cD_3	0.9916	0.9957	0.9962	0.9963
14	$cA_3cD_3cD_2$	0.9958	0.9979	0.9981	0.9981
15	$cA_3cD_3cD_2cD_1$	0.9974	0.9987	0.9987	0.9987

For the urban CASI scenery based on the adaptive Gaussian model, Table 62 shows that 10 out of 56 combinations, or 17.86% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. As was the case with the urban and vegetative CASI scenery based on the simple Gaussian model, DWT combination 6 (cA_2cD_2) consistently performs poorly across all four data cubes. The ROC curves corresponding to *CASI_urban_10* are shown in Figure 78 and Figure 79.



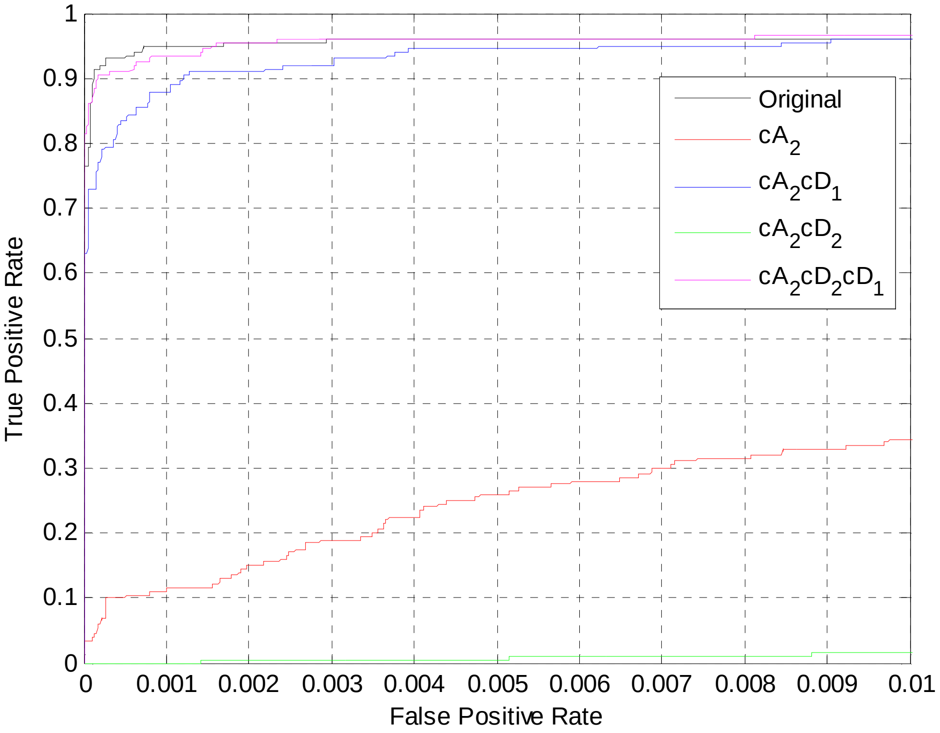


Figure 78. ROC curves of DWT level 1 and 2 coefficients on *CASI_urban_10* with adaptive Gaussian model.

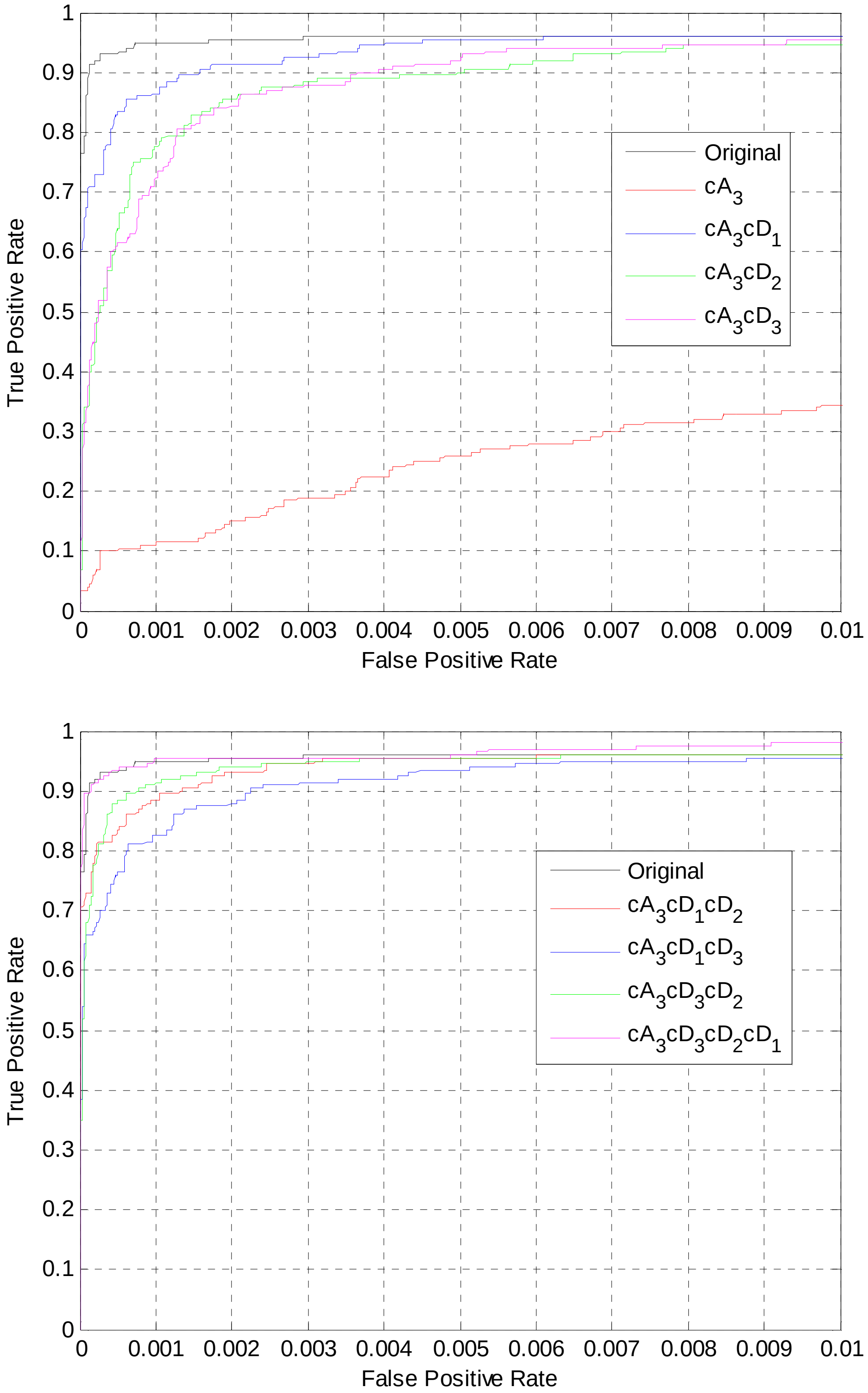


Figure 79. ROC curves of DWT level 3 coefficients on *CASI_urban_10* with adaptive Gaussian model.

Table 63. AUROCs of DWT combinations on vegetative CASI scenery using adaptive Gaussian model.

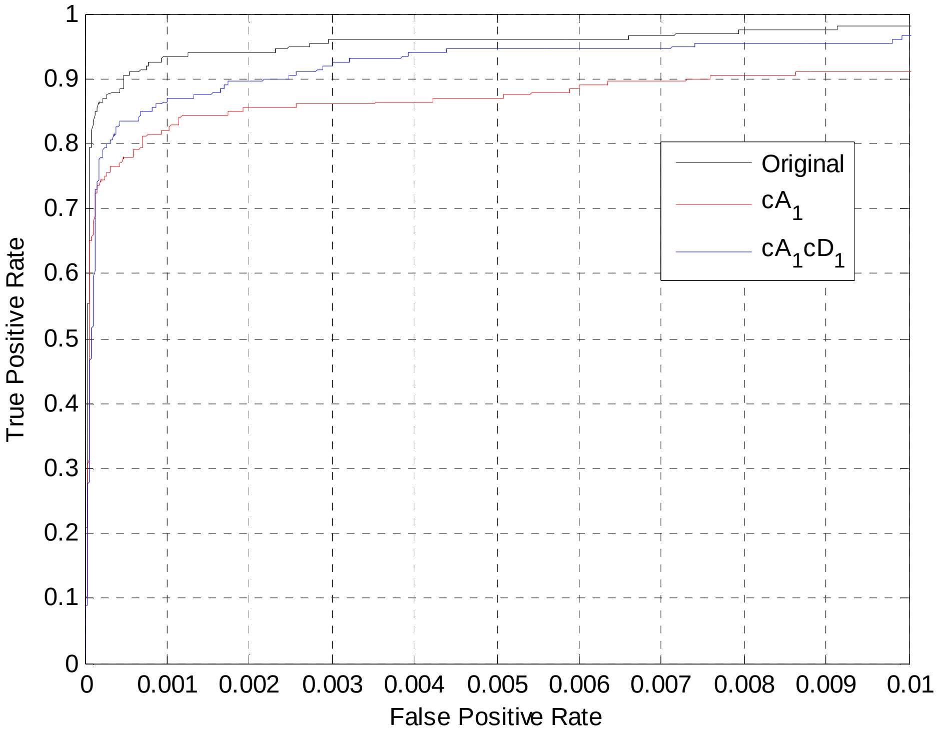
Index	Features	*CASI_veg_8*	*CASI_veg_10*	*CASI_veg_12*	*CASI_veg_15*
1	*Original*	0.9916	0.9916	0.9916	0.9916
2	cA_1	0.9891	0.9901	0.9901	0.9901
3	cA_1cD_1	0.9919	0.9929	0.9929	0.9929
4	cA_2	0.9866	0.9902	0.9903	0.9903
5	cA_2cD_1	0.9909	0.9909	0.9909	0.9909
6	cA_2cD_2	0.3455	0.3626	0.2722	0.2101
7	$cA_2cD_2cD_1$	0.9916	0.9916	0.9916	0.9916
8	cA_3	0.9849	0.9852	0.9852	0.9852
9	cA_3cD_1	0.9897	0.9898	0.9898	0.9898
10	$cA_3cD_1cD_2$	0.9898	0.9899	0.9899	0.9899
11	$cA_3cD_1cD_3$	0.9894	0.9894	0.9894	0.9894
12	cA_3cD_2	0.9880	0.9882	0.9882	0.9882
13	cA_3cD_3	0.9881	0.9881	0.9881	0.9881
14	$cA_3cD_3cD_2$	0.9901	0.9901	0.9901	0.9901
15	$cA_3cD_3cD_2cD_1$	0.9917	0.9917	0.9917	0.9917

Table 63 provides the results for the vegetative CASI scenery based on the adaptive Gaussian model. Twelve out of the 56 combinations, or 21.43% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. Yet again, DWT combination 6 (cA_2cD_2) consistently performs poorly across all four data cubes.

Table 64. AUROCs of DWT combinations on urban HYDICE scenery using adaptive Gaussian model.

Index	Features	*HYDICE_urban_8*	*HYDICE_urban_10*	*HYDICE_urban_12*	*HYDICE_urban_15*
1	*Original*	0.9715	0.9992	0.9999	1.0000
2	cA_1	0.7942	0.9910	1.0000	1.0000
3	cA_1cD_1	0.8662	0.9985	1.0000	1.0000
4	cA_2	0.7458	0.9644	0.9972	0.9999
5	cA_2cD_1	0.9752	0.9999	1.0000	1.0000
6	cA_2cD_2	0.9772	0.9999	1.0000	1.0000
7	$cA_2cD_2cD_1$	0.9516	0.9988	1.0000	1.0000
8	cA_3	0.8430	0.9816	0.9969	0.9993
9	cA_3cD_1	0.8613	0.9544	0.9829	0.9916
10	$cA_3cD_1cD_2$	0.9220	0.9928	0.9988	0.9988
11	$cA_3cD_1cD_3$	0.9835	0.9999	1.0000	1.0000
12	cA_3cD_2	0.9193	0.9841	0.9968	0.9971
13	cA_3cD_3	0.7589	0.9056	0.9776	0.9992
14	$cA_3cD_3cD_2$	0.9527	0.9946	0.9997	0.9997
15	$cA_3cD_3cD_2cD_1$	0.9675	0.9983	0.9998	0.9998

For the urban HYDICE scenery, Table 64 shows that 18 out of 56 combinations, or ~32.14% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. The urban HYDICE scenery does not have a particular DWT coefficient combination that performs poorly on any of the data cubes. The ROC curves corresponding to *HYDICE_urban_10* are shown in Figure 80 and Figure 81.



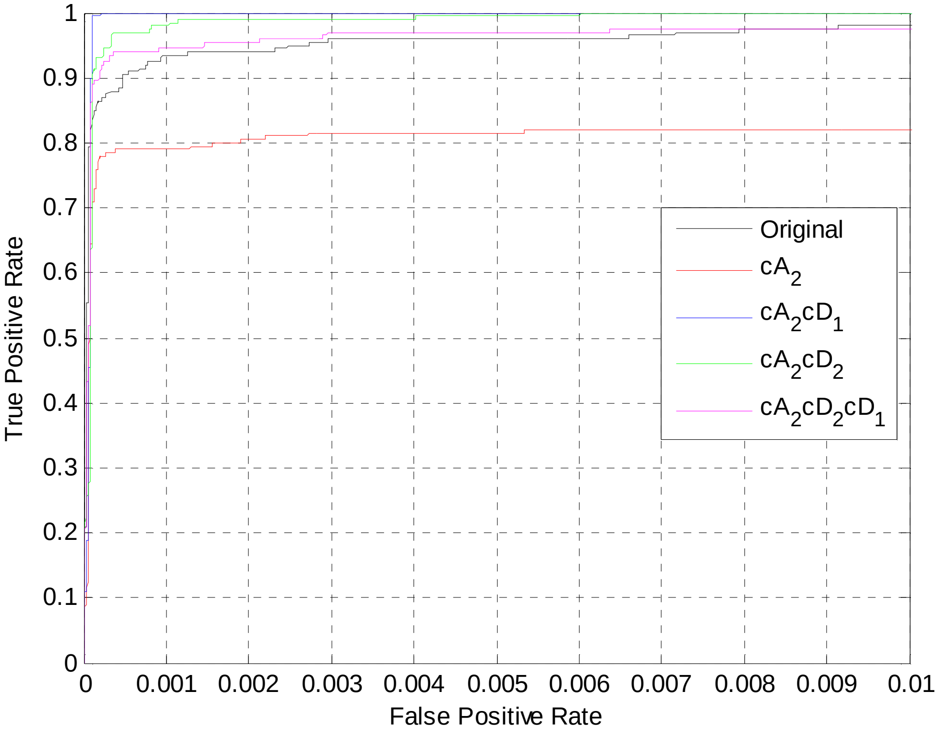


Figure 80. ROC curves of DWT level 1 and 2 coefficients on *HYDICE_urban_10* with adaptive Gaussian model.

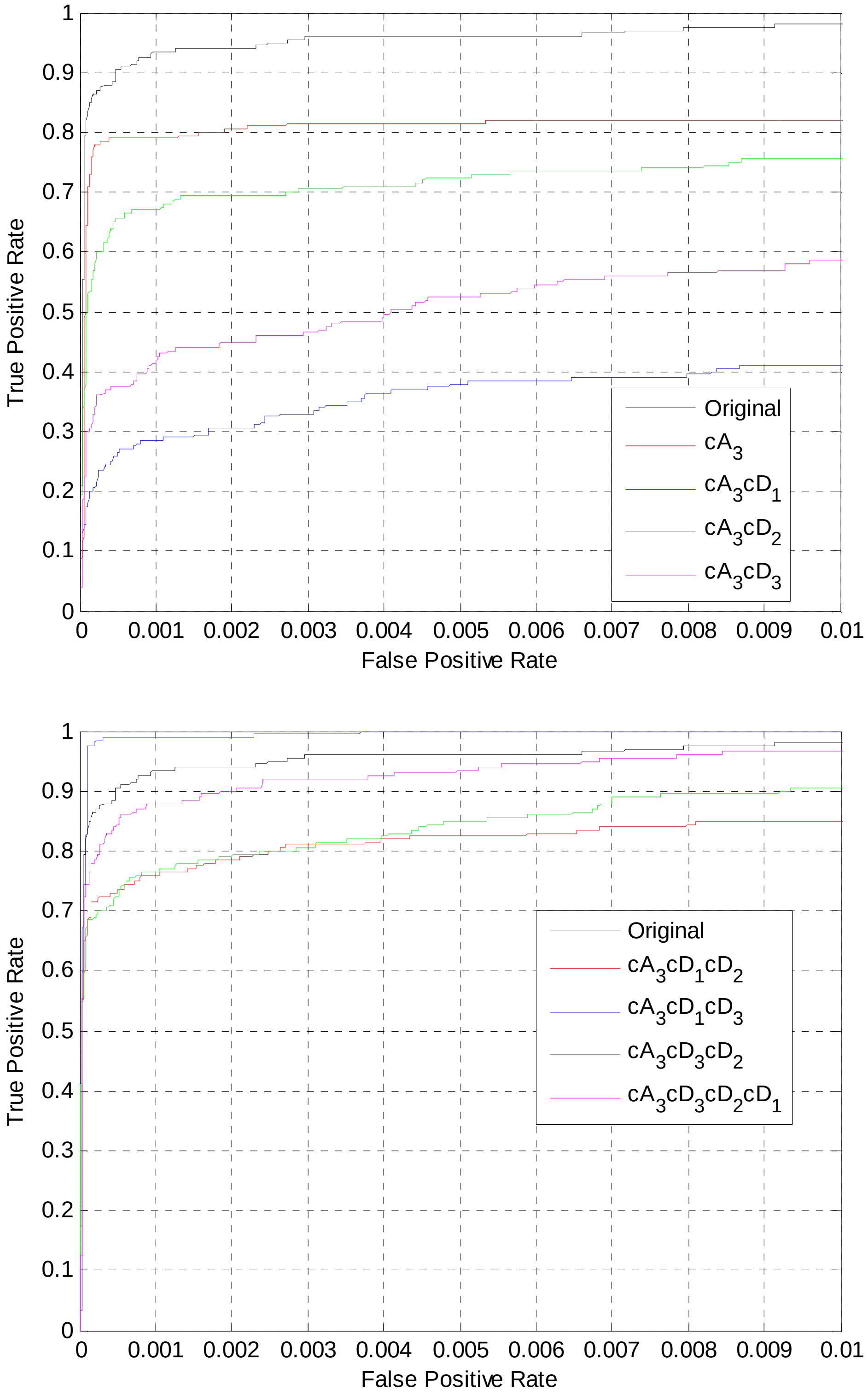


Figure 81. ROC curves of DWT level 3 coefficients on *HYDICE_urban_10* with adaptive Gaussian model.

Table 65. AUROCs of DWT combinations on vegetative HYDICE scenery using adaptive Gaussian model.

Index	Features	*HYDICE_veg_8*	*HYDICE_veg_10*	*HYDICE_veg_12*	*HYDICE_veg_15*
1	*Original*	0.9986	0.9986	0.9986	0.9986
2	cA_1	0.9962	1.0000	1.0000	1.0000
3	cA_1cD_1	0.9995	0.9995	0.9995	0.9995
4	cA_2	0.9914	0.9965	0.9969	0.9969
5	cA_2cD_1	1.0000	1.0000	1.0000	1.0000
6	cA_2cD_2	0.9988	0.9988	0.9988	0.9988
7	$cA_2cD_2cD_1$	0.9984	0.9984	0.9984	0.9984
8	cA_3	0.9893	0.9927	0.9927	0.9927
9	cA_3cD_1	0.9992	0.9995	0.9995	0.9995
10	$cA_3cD_1cD_2$	0.9993	0.9995	0.9995	0.9995
11	$cA_3cD_1cD_3$	0.9999	1.0000	1.0000	1.0000
12	cA_3cD_2	0.9927	0.9927	0.9927	0.9927
13	cA_3cD_3	0.3074	0.3330	0.3543	0.3324
14	$cA_3cD_3cD_2$	1.0000	1.0000	1.0000	1.0000
15	$cA_3cD_3cD_2cD_1$	1.0000	1.0000	1.0000	1.0000

For the vegetative HYDICE scenery shown in Table 65, 35 out of 56 combinations, or 62.50% of the combinations, yielded either identical or improved AUROCs when compared to the AUROCs using the original signatures. As was the case with the vegetative HYDICE scenery based on the simple Gaussian model, DWT combination 13 (cA_3cD_3) does consistently poorly across all four data cubes.

As the previous tables and figures have shown, use of the DWT coefficients as features can provide improved detection. Clearly, the improvement is affected by the model of target variability and the severity of variability of the targets in the scenes. In general, SFJTC detection using the original signatures deteriorates as the variability of the targets increases. However, particular combinations of DWT coefficients provide consistently reliable detection performance, regardless of the severity of variability of the scene targets. For the simple Gaussian model, examples of these DWT combinations include combination 13 (cA_3cD_3) in Table 58 and Table 59, combination 11 ($cA_3cD_1cD_3$) in Table 60, and combination 14 ($cA_3cD_3cD_2$) in Table 61. For the adaptive Gaussian model, examples of these DWT combinations include combination 15 ($cA_3cD_3cD_2cD_1$) in Table 62 and Table 63, combination 11 ($cA_3cD_1cD_3$) in Table 64, and combination 14 ($cA_3cD_3cD_2$) in Table 65.

By the same token, in some of the scenery, there appears to be a particular DWT coefficient combination that provides consistently poor results on all the data cubes. Examples of this include combination 6 (cA_2cD_2) in Table 58, Table 59, Table 62, and Table 63; these tables correspond to the urban and vegetative CASI scenery for both models, respectively. Similarly, another example includes combination 13 (cA_3cD_3) in Table 61 and Table 65 corresponding to the vegetative HYDICE scenery for both models, respectively.

As we have seen, results indicate that particular combinations of DWT coefficients perform better than others. In the next section, we will formulate an algorithm for choosing an optimal combination prior to detection.

F. Selection of Optimal Combination of DWT Coefficients

1. Procedure

The purpose of our training algorithm is to determine the best set of DWT coefficients from Table 57 to use for SFJTC-based detection on a scene. Given that the training data should contain samples from both the target and background classes, how do we wisely select samples from both classes?

Concerning the target class, we are only given a pure target signature from a library; we do not have samples that characterize its spectral variability. As we did earlier with the SVDD work, we will introduce spectral variability into the target class by generating *100* signatures according to both target variability models discussed in Chapter IV. In both models, we will fix the value of σ^2 to achieve a SNR of 10 dB to ensure sufficient spectral variability. For the background class, we can safely use random samples from the scene since targets occur with such low probability. Assuming no *a priori* knowledge of the scene, we randomly select *8000* pixels, 20% of the total pixels, for use as our background training samples. These 8100 signatures form the training set for the supervised coefficient selection process. The ratio of 100 target signatures to 8000 background signatures was chosen to mimic a low probability scenario in the training data. The SFJTC algorithm is run between the training set signatures and the pure target signature using each of the DWT combinations in Table 57. The DWT coefficient combination yielding the largest AUROC is selected as the optimal combination for a given scene.

2. *Results*

Let us take a detailed look at the comprehensive rankings of the supervised coefficient selection algorithm on the urban CASI scenery. In Table 66, the results are shown for all four data cubes, where each coefficient combination is ranked in descending order along with its corresponding AUROC metric on the training data.

Table 66. Supervised selection rankings for urban CASI scenery with simple Gaussian model.

Index	Features	*CASI_urban_8*	*CASI_urban_10*	*CASI_urban_12*	*CASI_urban_15*
1	*Original*	14 *(0.4961)*	14 *(0.4890)*	14 *(0.4918)*	14 *(0.4948)*
2	cA_1	9 *(0.9398)*	9 *(0.9363)*	9 *(0.9329)*	9 *(0.9336)*
3	cA_1cD_1	13 *(0.7490)*	13 *(0.7434)*	13 *(0.7453)*	13 *(0.7435)*
4	cA_2	12 *(0.8636)*	12 *(0.8599)*	12 *(0.8589)*	12 *(0.8601)*
5	cA_2cD_1	11 *(0.8934)*	11 *(0.8889)*	11 *(0.8868)*	11 *(0.8883)*
6	cA_2cD_2	15 *(0.1703)*	15 *(0.1734)*	15 *(0.1757)*	15 *(0.1749)*
7	$cA_2cD_2cD_1$	10 *(0.8966)*	10 *(0.8936)*	10 *(0.8921)*	10 *(0.8934)*
8	cA_3	1 *(0.9976)*	1 *(0.9971)*	1 *(0.9959)*	1 *(0.9961)*
9	cA_3cD_1	6 *(0.9924)*	6 *(0.9910)*	6 *(0.9890)*	6 *(0.9898)*
10	$cA_3cD_1cD_2$	7 *(0.9894)*	7 *(0.9876)*	7 *(0.9855)*	7 *(0.9859)*
11	$cA_3cD_1cD_3$	5 *(0.9948)*	5 *(0.9937)*	5 *(0.9915)*	5 *(0.9920)*
12	cA_3cD_2	4 *(0.9958)*	4 *(0.9952)*	4 *(0.9937)*	3 *(0.9944)*
13	cA_3cD_3	2 *(0.9975)*	2 *(0.9969)*	2 *(0.9958)*	2 *(0.9957)*
14	$cA_3cD_3cD_2$	3 *(0.9964)*	3 *(0.9954)*	3 *(0.9939)*	4 *(0.9942)*
15	$cA_3cD_3cD_2cD_1$	8 *(0.9711)*	8 *(0.9685)*	8 *(0.9654)*	8 *(0.9658)*

We can make several observations from Table 66. First, the optimal combination on all four data cubes is DWT combination 8 (cA_3), and the worst performing DWT combination is combination 6 (cA_2cD_2). In the previous section, we commented on how DWT combination 6 consistently performed poorly on several scenes, and our algorithm appropriately ranks this combination as such. Also, with the exception of a couple of instances, the rankings are identical on all four data cubes. Notice that the eight DWT combinations pertaining to the 3rd level decomposition, combinations 8 – 15, are ranked as the top 8 performing combinations. Figure 82 shows the 8000 randomly selected signatures from *casi_urban_10* that form the background class training signatures. Figure 83 shows the spectral variability of the 100 generated target signatures with the pure target signature in bold.

Table 67 shows the comprehensive rankings of the supervised coefficient selection algorithm on the urban HYDICE scenery using the simple Gaussian model. Once again, the results are identical on all the data cubes with the exception of a couple of instances. Again, the optimal combination for all four data cubes is DWT combination 8 (cA_3). Figure 84 and Figure 85 show the randomly selected background training signatures and the generated target training signatures, respectively.

For sake of brevity, the selected optimal DWT coefficient combinations for all of the data cubes for both target variability models are summarized in Table 68 – Table 71. For each data cube, the optimal DWT coefficient combination is listed along with the AUROC detection results using both the original signatures and the optimal DWT coefficient combination.

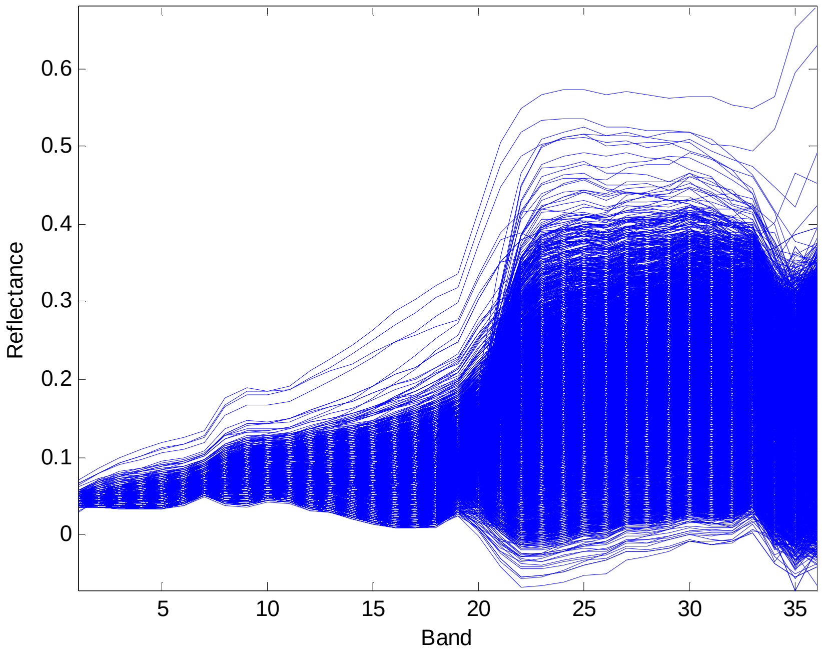


Figure 82. Randomly selected scene signatures used as background training data for *CASI_urban_10*.

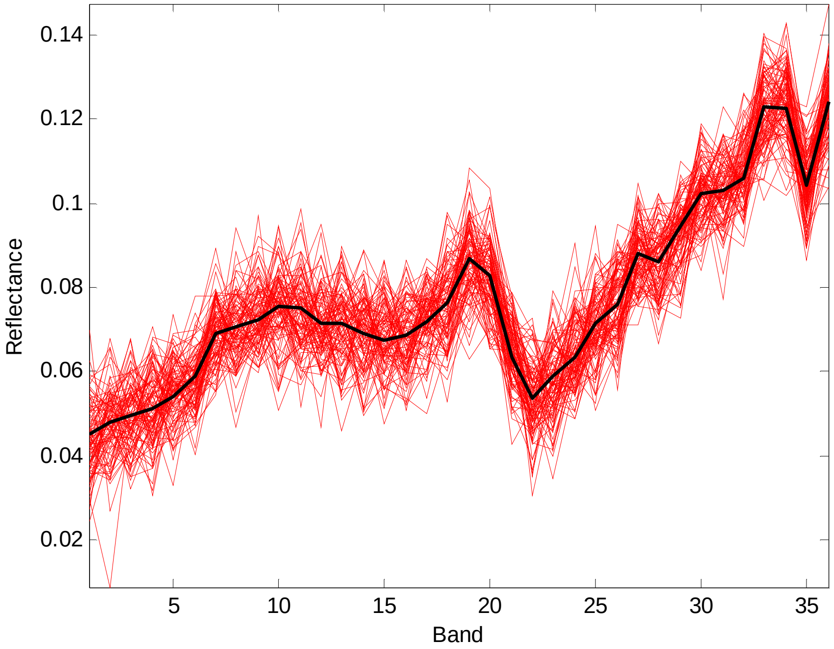


Figure 83. Generated target training signatures for *CASI_urban_10*.

Table 67. Supervised selection rankings for urban HYDICE scenery with simple Gaussian model.

Index	Features	*HYDICE_urban_8*	*HYDICE_urban_10*	*HYDICE_urban_12*	*HYDICE_urban_15*
1	*Original*	14 *(0.6395)*	15 *(0.6400)*	15 *(0.6366)*	15 *(0.6393)*
2	cA_1	13 *(0.6613)*	13 *(0.6616)*	13 *(0.6631)*	13 *(0.6633)*
3	cA_1cD_1	15 *(0.6389)*	14 *(0.6406)*	14 *(0.6369)*	14 *(0.6399)*
4	cA_2	12 *(0.8985)*	12 *(0.8994)*	12 *(0.8990)*	12 *(0.8995)*
5	cA_2cD_1	7 *(0.9895)*	7 *(0.9893)*	7 *(0.9889)*	7 *(0.9900)*
6	cA_2cD_2	3 *(0.9982)*	3 *(0.9976)*	3 *(0.9978)*	3 *(0.9982)*
7	$cA_2cD_2cD_1$	10 *(0.9592)*	10 *(0.9589)*	10 *(0.9589)*	10 *(0.9595)*
8	cA_3	1 *(0.9990)*	1 *(0.9992)*	1 *(0.9992)*	1 *(0.9993)*
9	cA_3cD_1	11 *(0.9090)*	11 *(0.9093)*	11 *(0.9091)*	11 *(0.9132)*
10	$cA_3cD_1cD_2$	9 *(0.9768)*	9 *(0.9762)*	9 *(0.9774)*	9 *(0.9782)*
11	$cA_3cD_1cD_3$	2 *(0.9984)*	2 *(0.9982)*	2 *(0.9983)*	2 *(0.9985)*
12	cA_3cD_2	6 *(0.9946)*	6 *(0.9942)*	6 *(0.9944)*	6 *(0.9952)*
13	cA_3cD_3	8 *(0.9881)*	8 *(0.9886)*	8 *(0.9884)*	8 *(0.9889)*
14	$cA_3cD_3cD_2$	4 *(0.9971)*	4 *(0.9969)*	4 *(0.9968)*	4 *(0.9975)*
15	$cA_3cD_3cD_2cD_1$	5 *(0.9963)*	5 *(0.9961)*	5 *(0.9961)*	5 *(0.9968)*

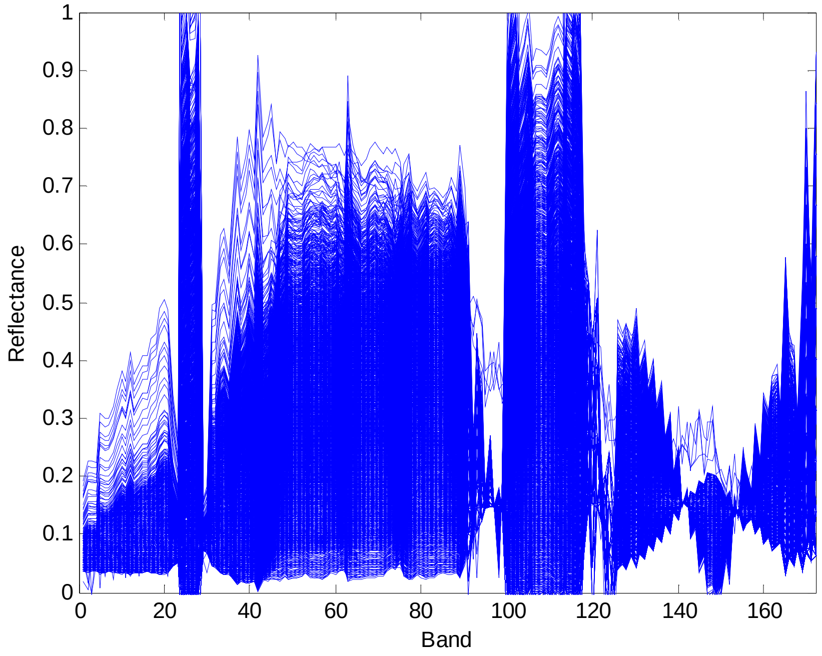


Figure 84. Randomly selected scene signatures used as background training data for *HYDICE_urban_10*.

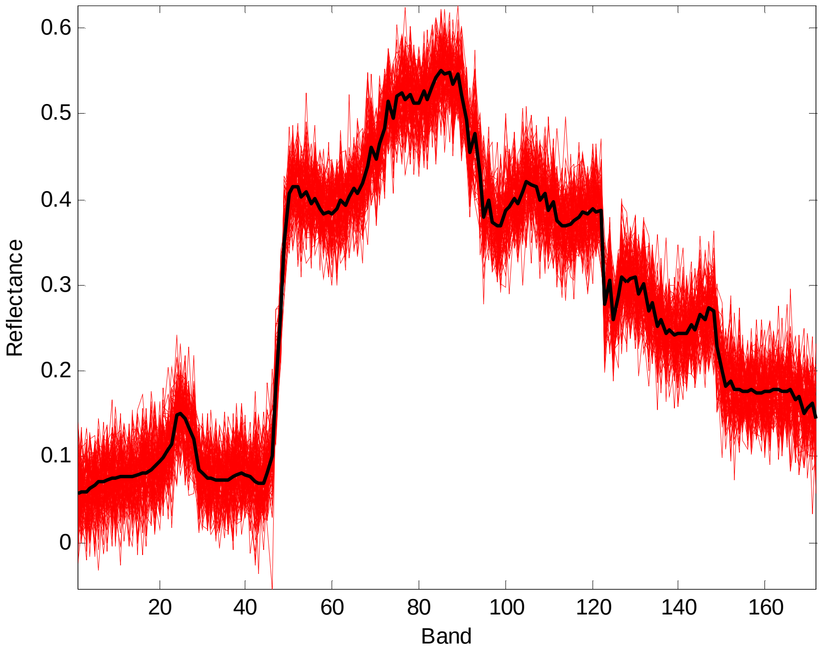


Figure 85. Generated target training signatures for *HYDICE_urban_10*.

Table 68. Optimal DWT coefficient results on CASI scenery with simple Gaussian model.

Data Cube	Optimal DWT Combination	AUROC using DWT Combination	AUROC using Original Data
CASI_urban_15	8 *(cA_3)*	0.9956	0.9947
CASI_urban_12	8 *(cA_3)*	0.9947	0.9079
CASI_urban_10	8 *(cA_3)*	0.9910	0.5393
CASI_urban_8	8 *(cA_3)*	0.9686	0.2653
CASI_veg_15	8 *(cA_3)*	0.9852	0.9916
CASI_veg_12	13 *(cA_3cD_3)*	0.9881	0.9614
CASI_veg_10	8 *(cA_3)*	0.9848	0.7779
CASI_veg_8	11 *($cA_3cD_1cD_3$)*	0.9763	0.3756

Table 69. Optimal DWT coefficient results on HYDICE scenery with simple Gaussian model.

Data Cube	Optimal DWT Combination	AUROC using DWT Combination	AUROC using Original Data
HYDICE_urban_15	8 *(cA_3)*	0.9993	0.9999
HYDICE_urban_12	8 *(cA_3)*	0.9993	0.9695
HYDICE_urban_10	8 *(cA_3)*	0.9948	0.6991
HYDICE_urban_8	8 *(cA_3)*	0.9691	0.3459
HYDICE_veg_15	15 *($cA_3cD_3cD_2cD_1$)*	1.0000	0.9986
HYDICE_veg_12	14 *($cA_3cD_3cD_2$)*	1.0000	0.9279
HYDICE_veg_10	15 *($cA_3cD_3cD_2cD_1$)*	0.9998	0.5223
HYDICE_veg_8	15 *($cA_3cD_3cD_2cD_1$)*	0.9761	0.3179

Table 70. Optimal DWT coefficient results on CASI scenery with adaptive Gaussian model.

Data Cube	Optimal DWT Combination	AUROC using DWT Combination	AUROC using Original Data
CASI_urban_15	15 ($cA_3cD_3cD_2cD_1$)	0.9987	0.9980
CASI_urban_12	15 ($cA_3cD_3cD_2cD_1$)	0.9987	0.9980
CASI_urban_10	15 ($cA_3cD_3cD_2cD_1$)	0.9987	0.9980
CASI_urban_8	15 ($cA_3cD_3cD_2cD_1$)	0.9974	0.9967
CASI_veg_15	15 ($cA_3cD_3cD_2cD_1$)	0.9917	0.9916
CASI_veg_12	12 (cA_3cD_2)	0.9882	0.9916
CASI_veg_10	15 ($cA_3cD_3cD_2cD_1$)	0.9917	0.9916
CASI_veg_8	15 ($cA_3cD_3cD_2cD_1$)	0.9917	0.9916

Table 71. Optimal DWT coefficient results on HYDICE scenery with adaptive Gaussian model.

Data Cube	Optimal DWT Combination	AUROC using DWT Combination	AUROC using Original Data
HYDICE_urban_15	11 ($cA_3cD_1cD_3$)	1.0000	1.0000
HYDICE_urban_12	11 ($cA_3cD_1cD_3$)	1.0000	0.9999
HYDICE_urban_10	11 ($cA_3cD_1cD_3$)	0.9999	0.9992
HYDICE_urban_8	5 (cA_2cD_1)	0.9752	0.9715
HYDICE_veg_15	2 (cA_1)	1.0000	0.9986
HYDICE_veg_12	2 (cA_1)	1.0000	0.9986
HYDICE_veg_10	2 (cA_1)	1.0000	0.9986
HYDICE_veg_8	3 (cA_1cD_1)	0.9995	0.9986

3. Conclusion

The results on the CASI scenery for the simple Gaussian model are shown in Table 68. As the table shows, the improvement in detection performance is proportional to the severity of the variability. For the CASI scenery, the AUROC improvement is as much as 70% and 60% for *CASI_urban_8* and *CASI_veg_8*, respectively. As Table 69 shows, the same trends hold for the HYDICE scenery as far as the improvement in detection performance. In fact, the AUROC improvement is as much as 62% and 65% for *HYDICE_urban_8* and *HYDICE_veg_8*, respectively.

The results using the adaptive Gaussian model are shown in Table 70 and Table 71. In general, the SFJTC results using the original signatures are already high and do not degrade with increasing variability as is the case with the simple Gaussian model. Regardless, the use of the SFJTC technique with the optimal DWT combination provides slightly increased or identical performance on all the data cubes. The next chapter will highlight these performance improvements by depicting the ROC curves of SFJTC-based detection using the original signatures and the optimal DWT coefficient combinations.

CHAPTER VII

FINAL DETECTION RESULTS AND COMPARISONS

In this chapter, we will present the results of our SVDD-based target detection scheme on all data cubes corresponding to both target variability models. We will also provide the results of our SFJTC algorithm on all data cubes corresponding to both target variability models. For the SVDD technique, the number of FPs and TPs will be listed for each data cube. For the SFJTC algorithm, we will provide the results in the form of ROC curves and AUROCs. For sake of comparison, we will run the traditional algorithms discussed in Chapter III on all data cubes corresponding to both target variability models. The results for the traditional detection algorithms will also be provided in the form of ROC curves and AUROCs. Section A provides the detection results for the simple Gaussian model, while section B provides the detection results for the adaptive Gaussian model.

A. Results with Simple Gaussian Model

1. SVDD performance comparisons

Here, we present the results of our SVDD-based detection scheme on all data cubes corresponding to the simple Gaussian model. The FPRs and TPRs will be provided for each data cube. For sake of comparison, we provide the TPRs of the AMF and CEM stochastic detection algorithms corresponding to the FPRs exhibited by the

SVDD. This provides a means of directly comparing the detection algorithms since the SVDD inherently performs binary classification, while the AMF and CEM outputs are continuous values that must be thresholded to yield the final output labels. Table 72 – Table 75 provide the detection results for all data cubes.

Table 72. SVDD performance comparisons for urban CASI scenery using simple Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
CASI_urban_15	1.7085e-02 (680)	0.955 (191)	0.705 (141)	0
CASI_urban_12	1.8191e-02 (724)	0.955 (191)	0.730 (146)	0.045 (9)
CASI_urban_10	1.7136e-02 (682)	0.955 (191)	0.660 (132)	0.010 (2)
CASI_urban_8	1.4648e-02 (583)	0.295 (59)	0.680 (136)	0.215 (43)

Table 73. SVDD performance comparisons for vegetative CASI scenery using simple Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
CASI_veg_15	1.0050e-04 (4)	0.930 (186)	0.825 (165)	0
CASI_veg_12	8.2915e-04 (33)	0.935 (187)	0.810 (162)	0.020 (4)
CASI_veg_10	2.2613e-04 (9)	0.935 (187)	0.715 (143)	0.125 (25)
CASI_veg_8	5.7789e-04 (23)	0.935 (187)	0.690 (138)	0.175 (35)

As Table 72 and Table 73 show, the CEM detection algorithm does poorly on the urban and vegetative CASI scenery. With the exception of *CASI_urban_8*, the SVDD provides a significant increase in the TPR compared to the AMF. The improvement is as much as 29.5% and 24.5% in the urban and vegetative scenery, respectively.

Table 74. SVDD performance comparisons for urban HYDICE scenery using simple Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
HYDICE_urban_15	7.5377e-05 (3)	1.000 (200)	0.900 (180)	0
HYDICE _urban_12	5.0251e-05 (2)	1.000 (200)	0.660 (132)	0
HYDICE_urban_10	2.0101e-04 (8)	1.000 (200)	0.860 (172)	0
HYDICE_urban_8	5.0251e-05 (2)	1.000 (200)	0.465 (93)	0

Table 75. SVDD performance comparisons for vegetative HYDICE scenery using simple Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
HYDICE_veg_15	5.2764e-04 (21)	0.990 (198)	1.000 (200)	0.98 (196)
HYDICE_veg_12	5.0251e-05 (2)	0.975 (195)	0.985 (197)	0.940 (188)
HYDICE_veg_10	0	0.970 (194)	0.750 (175)	0.585 (117)
HYDICE_veg_8	0	0.300 (60)	0.805 (161)	0.280 (56)

For the urban HYDICE scenery, the CEM algorithm does miserably at such low FPRs. As Table 74 shows, at extremely low FPRs, the SVDD detects all targets and provides as little as a 10% increase in the TPR and as much as a 54.5% increase in the TPR when compared to the AMF. The CEM performs better on the vegetative HYDICE scenery, and both stochastic algorithms rival the performance of the SVDD for the two data cubes with the least variability. For *HYDICE_veg_10*, the SVDD outperforms the AMF and CEM by providing a 22% and 38.5% increase in the TPRs, respectively. However, for *HYDICE_veg_8*, the AMF outperforms the SVDD since the SVDD does not seem to capture the large spectral variability present in this data cube.

2. SFJTC performance comparisons

In this section, we present the results of SFJTC-based detection using both the original data and the optimal DWT coefficient combinations for each data cube. For sake of comparison, we have run the traditional stochastic AMF and CEM algorithms as well as the deterministic SAM and SID algorithms on the data cubes and have captured the AUROCs. For the ROC curves, the results of four detection algorithms will be shown: the AMF detector, the SID detector, the SFJTC technique using the original data, and the SFJTC technique using the optimal DWT coefficient combination (SFJTC_DWT). By doing this, our goal is to depict the performance differences between the following:

1. The SFJTC techniques using the original data and the optimal DWT coefficient combination.
2. The SFJTC technique using the optimal DWT coefficient combination and a traditional stochastic algorithm (AMF) and deterministic algorithm (SID).

Figure 86 - Figure 93 show the ROC curves corresponding to the urban and vegetative CASI scenery. Note that the range of values for the horizontal axis depicting the FPR may vary to highlight differences between the algorithms where applicable. Table 76 and Table 77 provide the AUROCs of detection for the traditional and SFJTC algorithms on the urban and vegetative CASI scenery. Figure 94 - Figure 101 are the ROC curves corresponding to the urban and vegetative HYDICE scenery. Table 78 and Table 79 provide the AUROCs of detection for the traditional and SFJTC algorithms on the urban and vegetative HYDICE scenery.

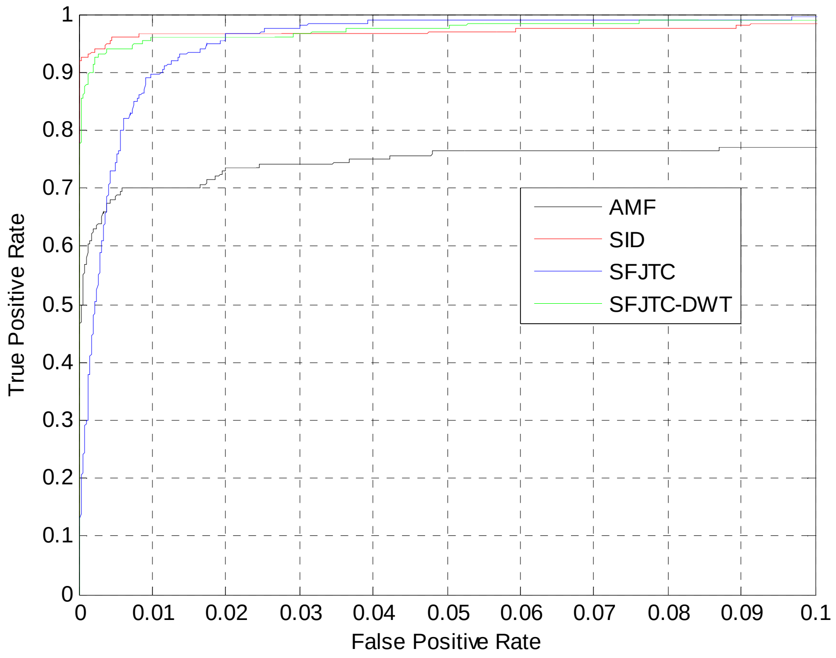


Figure 86. SFJTC ROC curve comparisons for *CASI_urban_15* using simple Gaussian model.

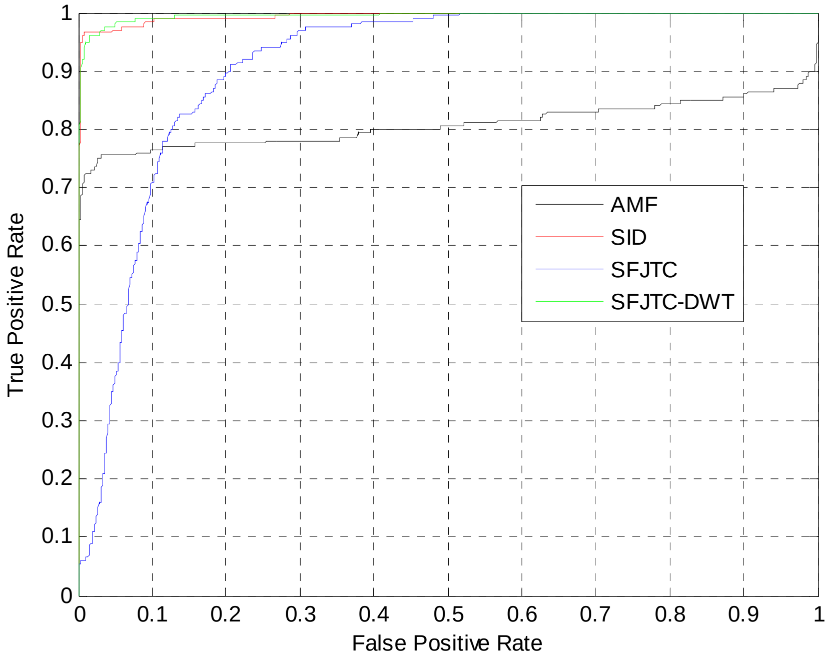


Figure 87. SFJTC ROC curve comparisons for *CASI_urban_12* using simple Gaussian model.

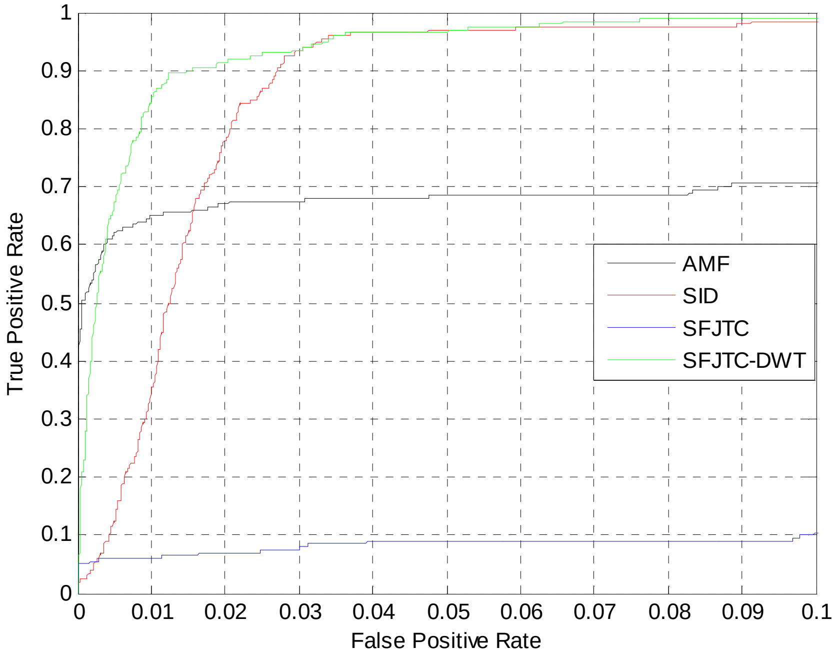


Figure 88. SFJTC ROC curve comparisons for *CASI_urban_10* using simple Gaussian model.

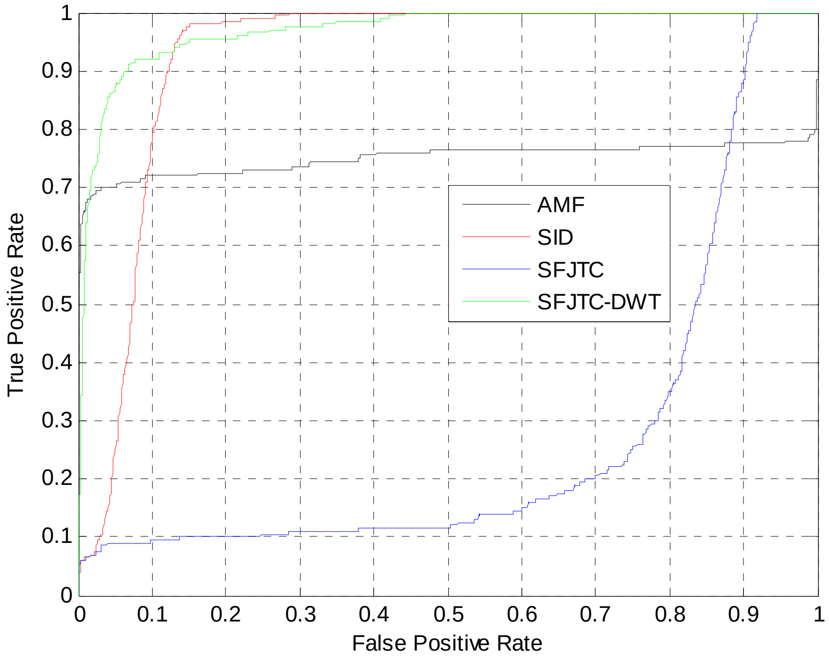


Figure 89. SFJTC ROC curve comparisons for *CASI_urban_8* using simple Gaussian model.

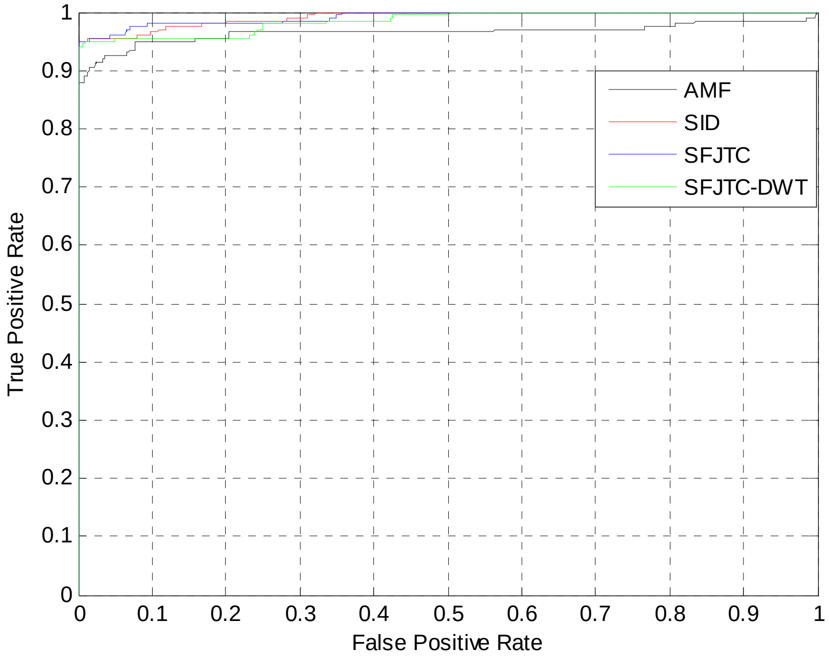


Figure 90. SFJTC ROC curve comparisons for *CASI_veg_15* using simple Gaussian model.

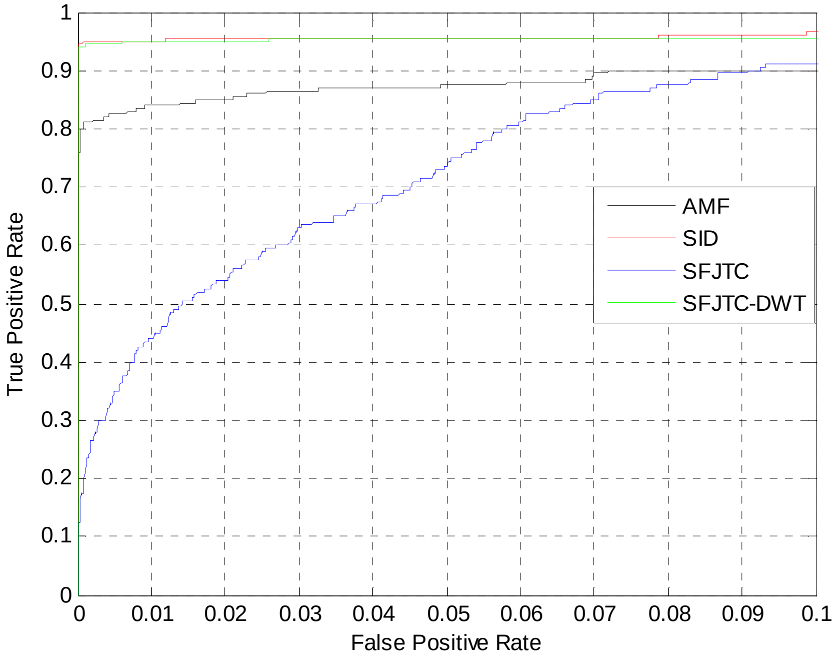


Figure 91. SFJTC ROC curve comparisons for *CASI_veg_12* using simple Gaussian model.

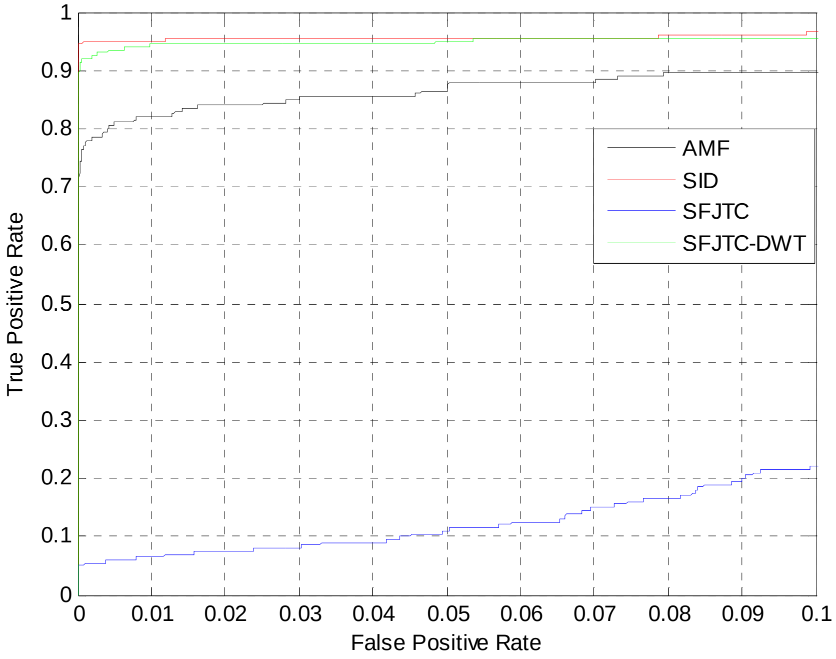


Figure 92. SFJTC ROC curve comparisons for *CASI_veg_10* using simple Gaussian model.

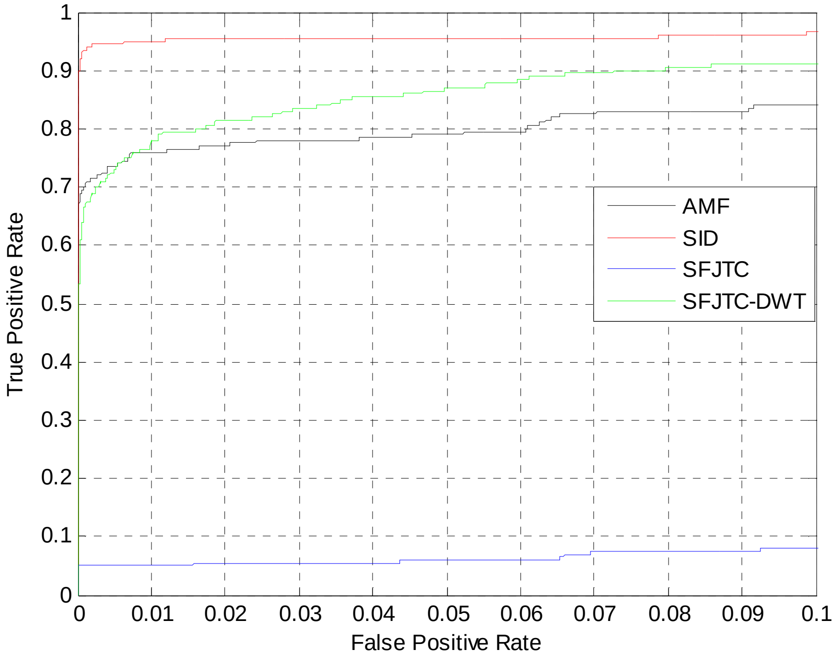


Figure 93. SFJTC ROC curve comparisons for *CASI_veg_8* using simple Gaussian model.

Table 76. SFJTC AUROC comparisons for urban CASI scenery using simple Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
CASI_urban_15	0.8386	0.4286	0.2445	0.9951	0.9947	0.9956
CASI_urban_12	0.8073	0.4883	0.2461	0.9938	0.9079	0.9947
CASI_urban_10	0.7603	0.4334	0.2643	0.9824	0.5393	0.9910
CASI_urban_8	0.7498	0.4716	0.2575	0.9239	0.2653	0.9686

Table 77. SFJTC AUROC comparisons for vegetative CASI scenery using simple Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
CASI_veg_15	0.9646	0.1646	0.3450	0.9915	0.9916	0.9852
CASI_veg_12	0.9208	0.2893	0.3509	0.9915	0.9614	0.9881
CASI_veg_10	0.9148	0.4930	0.3466	0.9915	0.7779	0.9848
CASI_veg_8	0.8539	0.4184	0.3516	0.9914	0.3756	0.9763

As Table 76 shows, the SFJTC_DWT technique yields the highest AUROC values on all data cubes for the urban CASI scenery. The ROC curves illustrate the performance improvement in running SFJTC using the DWT coefficients versus the original data. The ROC curves also show that the SFJTC_DWT technique outperforms the SID and AMF algorithms in progressively heavier scenarios of spectral variability. For the vegetative CASI scenery, the SID and SFJTC_DWT techniques provide very similar AUROC values. As shown by Table 77 and the ROC curves, the SID and SFJTC_DWT results for the vegetative CASI scenery are fairly uniform on all data cubes compared to the other algorithms. Again, notice the improvement in the SFJTC_DWT versus SFJTC using the original signatures as variability increases.

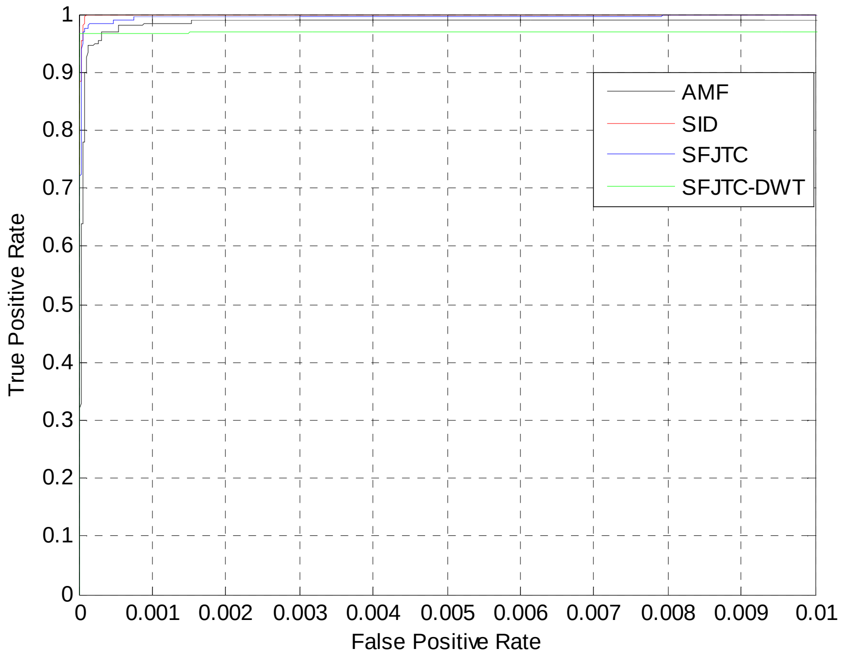


Figure 94. SFJTC ROC curve comparisons for *HYDICE_urban_15* using simple Gaussian model.

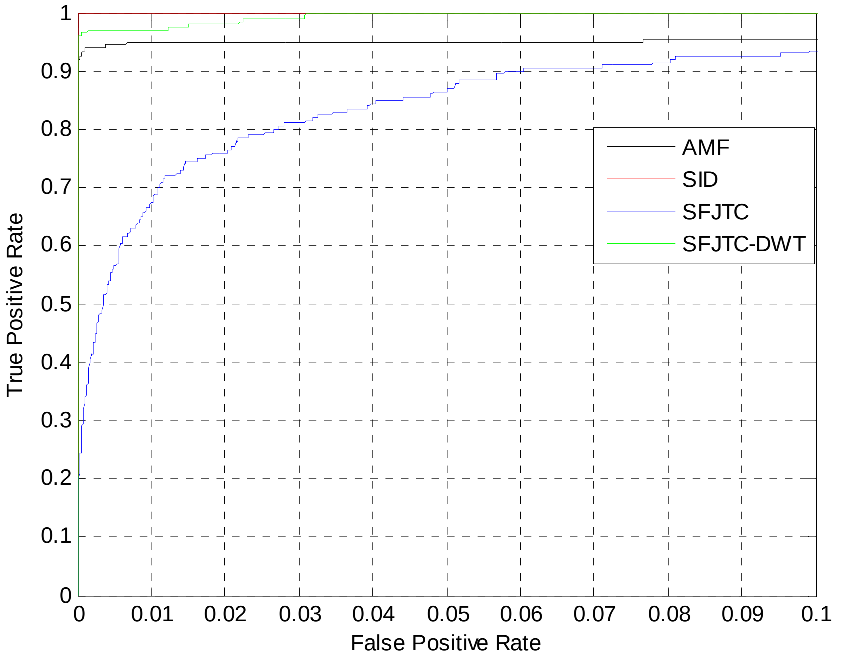


Figure 95. SFJTC ROC curve comparisons for *HYDICE_urban_12* using simple Gaussian model.

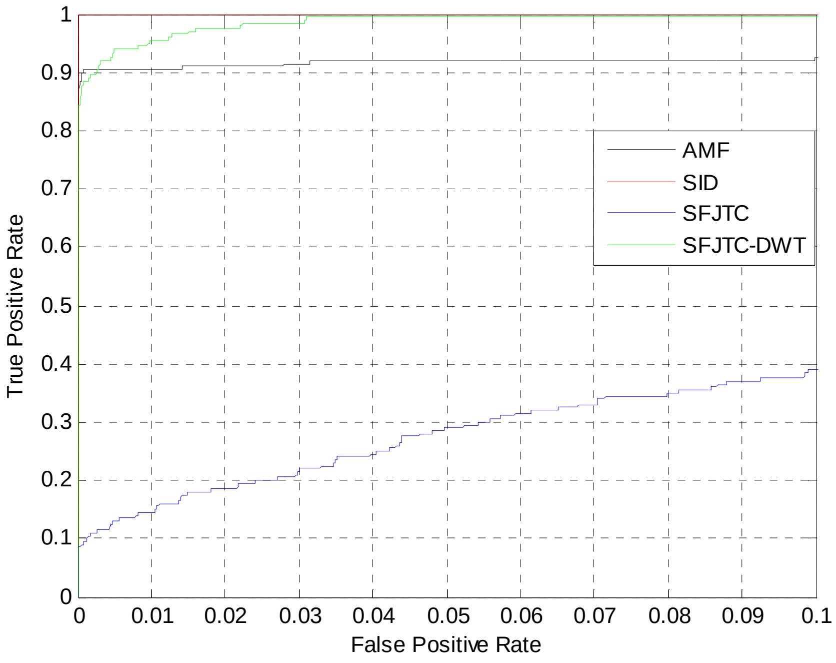


Figure 96. SFJTC ROC curve comparisons for *HYDICE_urban_10* using simple Gaussian model.

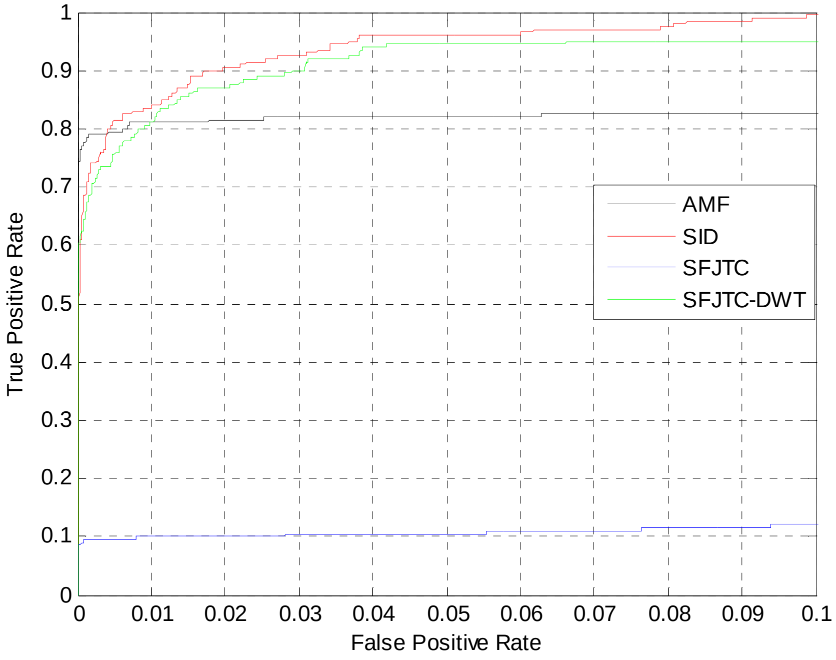


Figure 97. SFJTC ROC curve comparisons for *HYDICE_urban_8* using simple Gaussian model.

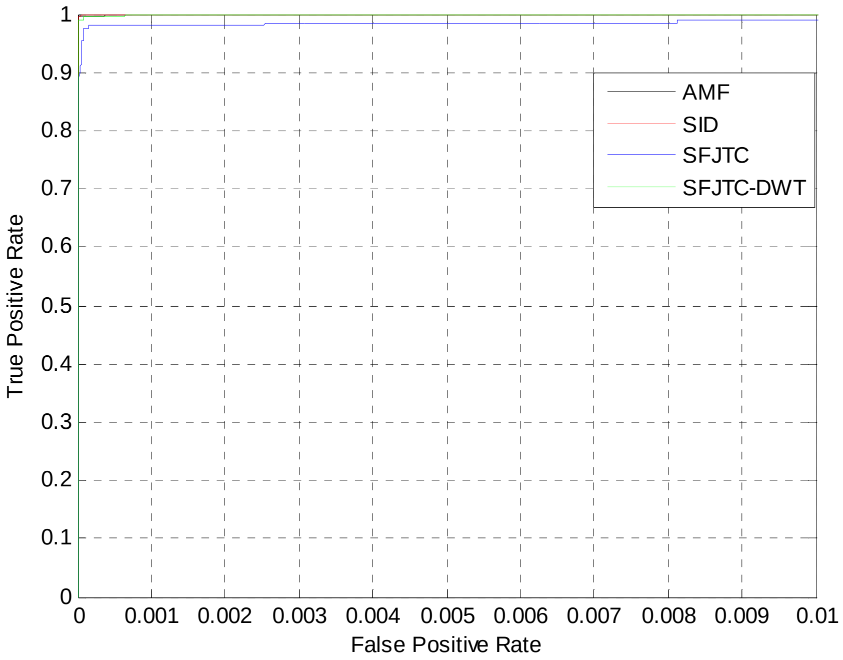


Figure 98. SFJTC ROC curve comparisons for *HYDICE_veg_15* using simple Gaussian model.

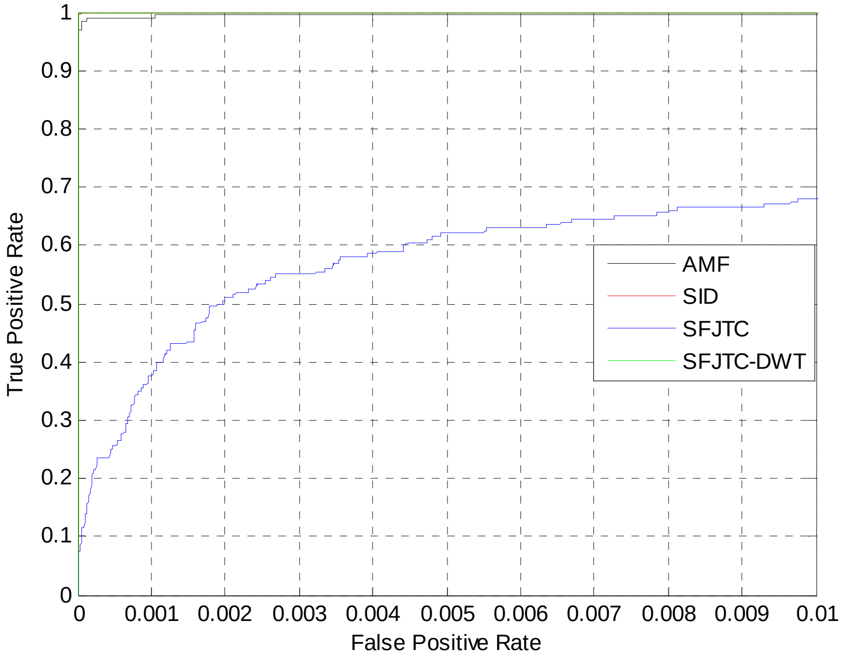


Figure 99. SFJTC ROC curve comparisons for *HYDICE_veg_12* using simple Gaussian model.

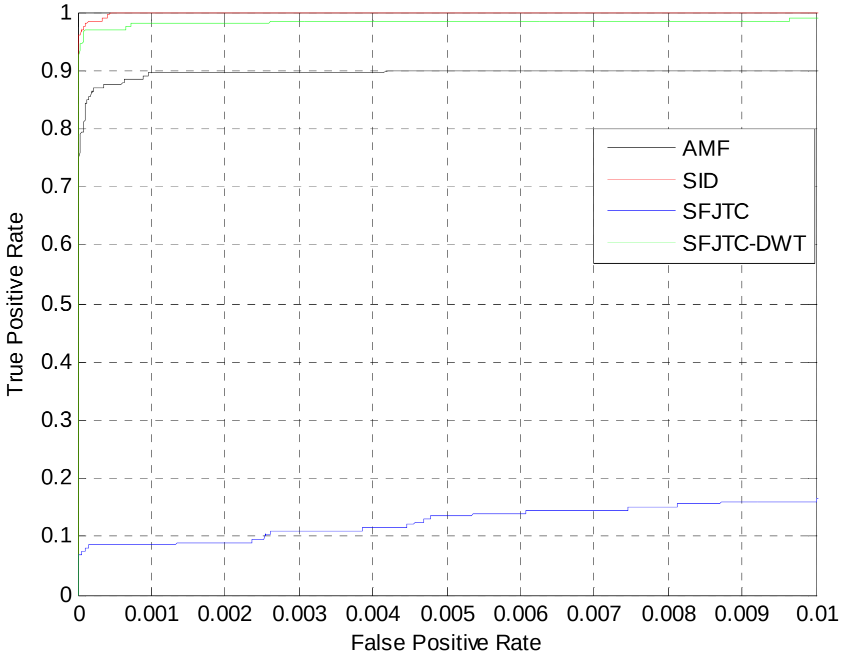


Figure 100. SFJTC ROC curve comparisons for *HYDICE_veg_10* using simple Gaussian model.

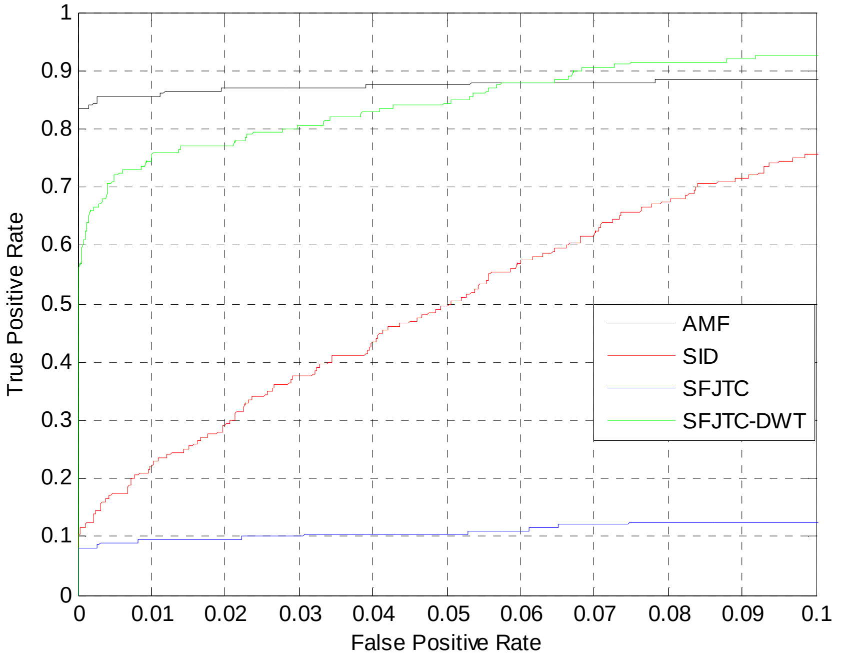


Figure 101. SFJTC ROC curve comparisons for *HYDICE_veg_8* using simple Gaussian model.

Table 78. SFJTC AUROC comparisons for urban HYDICE scenery using simple Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
HYDICE_urban_15	0.9947	0.9419	0.9464	1.0000	0.9999	0.9993
HYDICE_urban_12	0.9625	0.9326	0.9464	1.0000	0.9695	0.9993
HYDICE_urban_10	0.9309	0.9317	0.9463	0.9999	0.6991	0.9948
HYDICE_urban_8	0.8485	0.9350	0.9463	0.9928	0.3459	0.9691

Table 79. SFJTC AUROC comparisons for vegetative HYDICE scenery using simple Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
HYDICE_veg_15	1.0000	1.0000	0.6957	1.0000	0.9986	1.0000
HYDICE_veg_12	0.9950	0.9998	0.6966	1.0000	0.9279	1.0000
HYDICE_veg_10	0.9309	0.9939	0.6955	1.0000	0.5223	0.9998
HYDICE_veg_8	0.9002	0.9822	0.6991	0.9382	0.3179	0.9761

For the urban HYDICE scenery, all of the detection algorithms perform better than they did for the CASI scenery. As Table 78 shows, the SID and SFJTC_DWT techniques perform similarly on this scenery, with the SID yielding slightly higher AUROCs on all the data cubes. For the vegetative HYDICE scenery, similar patterns are observed, however the CEM also performs well and similarly to SID and SFJTC_DWT, even at heavier levels of spectral variability. Once again, the ROC curves illustrate the magnitude in performance gains achieved as a result of running SFJTC with the optimal DWT coefficients rather than the original signatures.

Table 80. Summary statistics of SFJTC AUROC comparisons for data based on simple Gaussian model.

Scenery	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
Urban CASI	0.7890 *(0.0415)*	0.4555 *(0.0292)*	0.2531 *(0.0094)*	0.9738 *(0.0337)*	0.6768 *(0.3380)*	0.9875 *(0.0127)*
Vegetative CASI	0.9136 *(0.0455)*	0.3414 *(0.1448)*	0.3485 *(0.0032)*	0.9915 *(0)*	0.7766 *(0.2835)*	0.9836 *(0.0051)*
Urban HYDICE	0.9342 *(0.0627)*	0.9353 *(0.0046)*	0.9463 *(0.0001)*	0.9982 *(0.0036)*	0.7536 *(0.3036)*	0.9906 *(0.0145)*
Vegetative HYDICE	0.9565 *(0.0490)*	0.9940 *(0.0083)*	0.6967 *(0.0017)*	0.9846 *(0.0309)*	0.6917 *(0.3258)*	0.9940 *(0.0119)*
All	0.8983 *(0.0807)*	0.6815 *(0.3034)*	0.5612 *(0.2861)*	0.9870 *(0.0225)*	0.7247 *(0.2836)*	0.9889 *(0.0111)*

Table 80 provides summary statistics of the AUROCs for the SFJTC techniques and traditional algorithms. For each type of scenery and sensor, we have computed the mean and standard deviation (shown in italics) of the AUROCs for the corresponding data cubes. We have also computed the mean and standard deviation of the AUROCs for all of the data cubes, as shown in the last row of Table 80.

For the urban CASI scenery, the most difficult scenery, SFJTC_DWT outperforms the other algorithms with the largest mean and smallest standard deviation AUROC values. For the vegetative CASI scenery and the urban HYDICE scenery, the SFJTC_DWT technique does second best, just shy of the SID algorithm. The SFJTC_DWT technique performs better than the SID and remaining algorithms on the vegetative HYDICE scenery. Over all of the scenes corresponding to the simple

Gaussian model, SFJTC_DWT provides the largest mean AUROC and smallest standard deviation of AUROC values.

B. Results with Adaptive Gaussian Model

1. SVDD performance comparisons

Here, we present the results of our SVDD-based detection scheme on all data cubes corresponding to the adaptive Gaussian model. As before, for sake of comparison, we provide the TPRs of the AMF and CEM stochastic detection algorithms corresponding to the FPRs exhibited by the SVDD. Table 81 – Table 84 provide the detection results for all data cubes.

Table 81. SVDD performance comparisons for urban CASI scenery using adaptive Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
CASI_urban_15	8.0402e-04 (32)	0.925 (185)	0.870 (174)	0
CASI_urban_12	3.0402e-03 (121)	0.945 (189)	0.885 (177)	0
CASI_urban_10	1.4322e-03 (57)	0.910 (182)	0.710 (142)	0
CASI_urban_8	2.8894e-03 (115)	0.825 (165)	0.690 (138)	0

Table 82. SVDD performance comparisons for vegetative CASI scenery using adaptive Gaussian model.

Data Cube	SVDD FPR	SVDD TPR	AMF TPR	CEM TPR
CASI_veg_15	2.5126e-05 (1)	0.930 (186)	0.910 (182)	0
CASI_veg_12	3.2663e-04 (13)	0.930 (186)	0.920 (184)	0
CASI_veg_10	2.5126e-05 (1)	0.930 (186)	0.785 (157)	0
CASI_veg_8	3.2663e-04 (13)	0.925 (185)	0.825 (165)	0

Table 83. SVDD performance comparisons for urban HYDICE scenery using adaptive Gaussian model.

Data Cube	FPRs	TPRs	AMF TPR	CEM TPR
HYDICE_urban_15	3.5176e-04 (14)	1.000 (200)	1.000 (200)	0
HYDICE _urban_12	1.0050e-04 (4)	1.000 (200)	0.995 (199)	0
HYDICE_urban_10	5.0251e-05 (2)	1.000 (200)	0.775 (155)	0
HYDICE_urban_8	2.0101e-04 (8)	1.000 (200)	0.860 (172)	0

Table 84. SVDD performance comparisons for vegetative HYDICE scenery using adaptive Gaussian model.

Data Cube	FPRs	TPRs	AMF TPR	CEM TPR
HYDICE_veg_15	4.0201e-04 (16)	0.990 (198)	1.000 (200)	1.000 (200)
HYDICE_veg_12	2.5126e-04 (10)	0.990 (198)	1.000 (200)	0.985 (197)
HYDICE_veg_10	1.2563e-04 (5)	0.990 (198)	1.000 (200)	0.990 (198)
HYDICE_veg_8	1.0050e-04 (4)	0.905 (181)	0.995 (199)	0.940 (188)

As Table 81 and Table 82 show, the SVDD is able to provide low FPRs on the order of 10e-03 and 10e-05 in the urban and vegetative CASI scenery, respectively. Once again, the CEM algorithm performs poorly on this scenery and cannot reliably operate at such low FPRs. The SVDD provides a TPR increase of as much as 20% and 10% in the urban and vegetative scenery, respectively.

For the urban HYDICE scenery, the CEM cannot operate at such low FPRs. As we have mentioned before, the urban HYDICE scenery is the least difficult of the hyperspectral imagery with respect to the separation between target and background signatures. This is evident by the extremely low FPRs and high TPRs. As Table 83 shows, the AMF performs just as well as the SVDD on the two data cubes with the least spectral variability. However, for the other two data cubes, the AMF cannot maintain high TPRs for the corresponding FPRs.

As Table 84 shows, the CEM algorithm is able to provide reliably high TPRs on the vegetative HYDICE scenery. For the first three data cubes, all three algorithms provide nearly identical TPRs. However, for *HYDICE_veg_8*, the AMF TPR is 10% larger than the SVDD and 5.5% larger than the CEM algorithm.

2. SFJTC performance comparisons

In this section, we present the results of SFJTC-based detection on the data cubes corresponding to the adaptive Gaussian model. Figure 102 – Figure 109 show the ROC curves corresponding to the urban and vegetative CASI scenery. Note that the range of values for the horizontal axis depicting the FPR may vary to highlight differences between the algorithms where applicable. Table 85 and Table 86 provide the AUROCs of detection for the traditional and SFJTC algorithms on the urban and vegetative CASI scenery. Figure 110 – Figure 117 are the ROC curves corresponding to the urban and vegetative HYDICE scenery. Table 87 and Table 88 provide the AUROCs of detection for the traditional and SFJTC algorithms on the urban and vegetative HYDICE scenery.

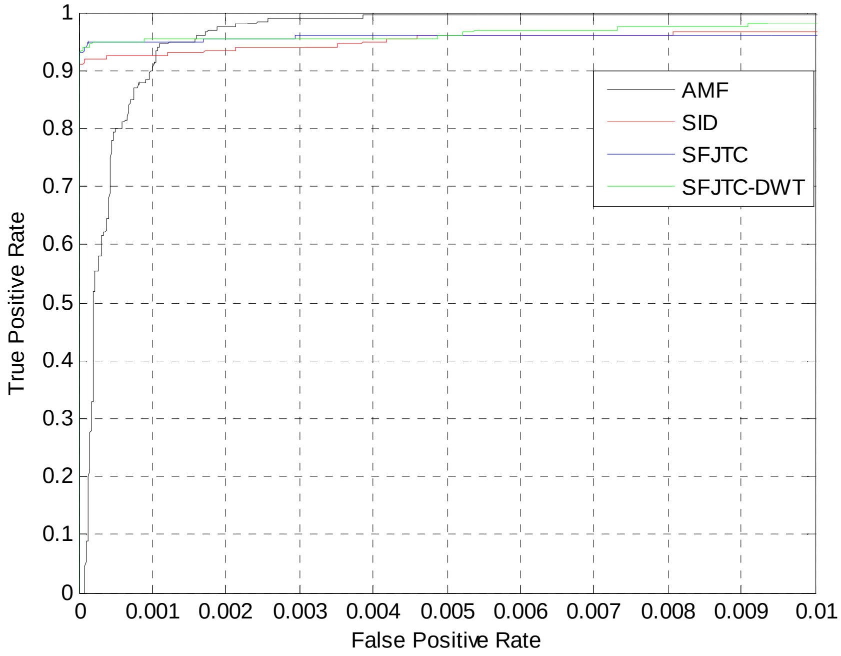


Figure 102. SFJTC ROC curve comparisons for *CASI_urban_15* using adaptive Gaussian model.

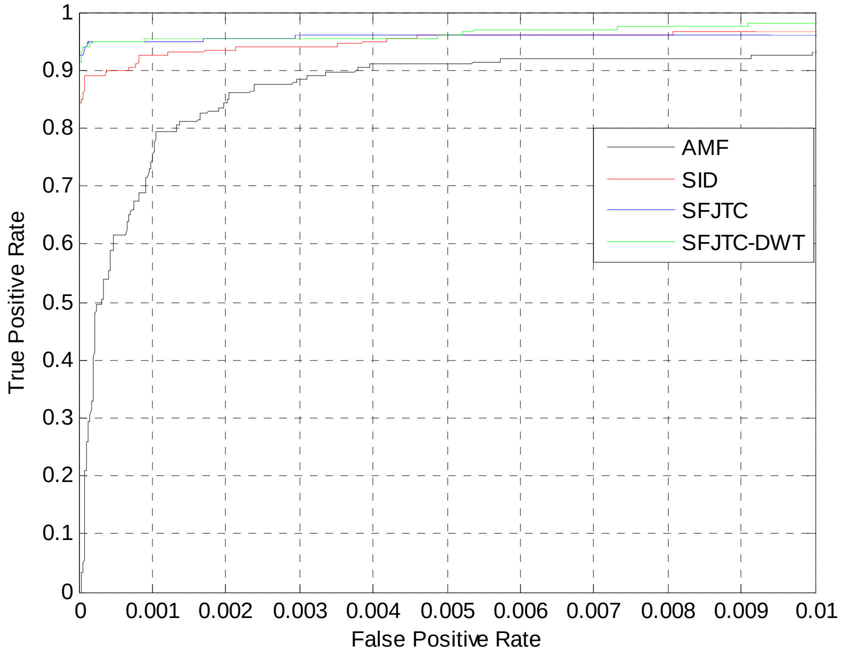


Figure 103. SFJTC ROC curve comparisons for *CASI_urban_12* using adaptive Gaussian model.

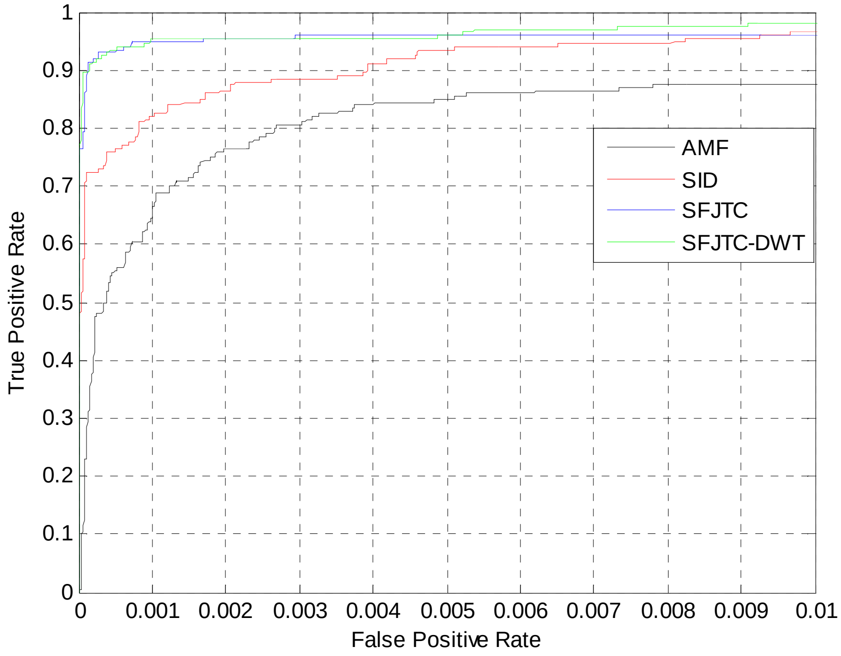


Figure 104. SFJTC ROC curve comparisons for *CASI_urban_10* using adaptive Gaussian model.

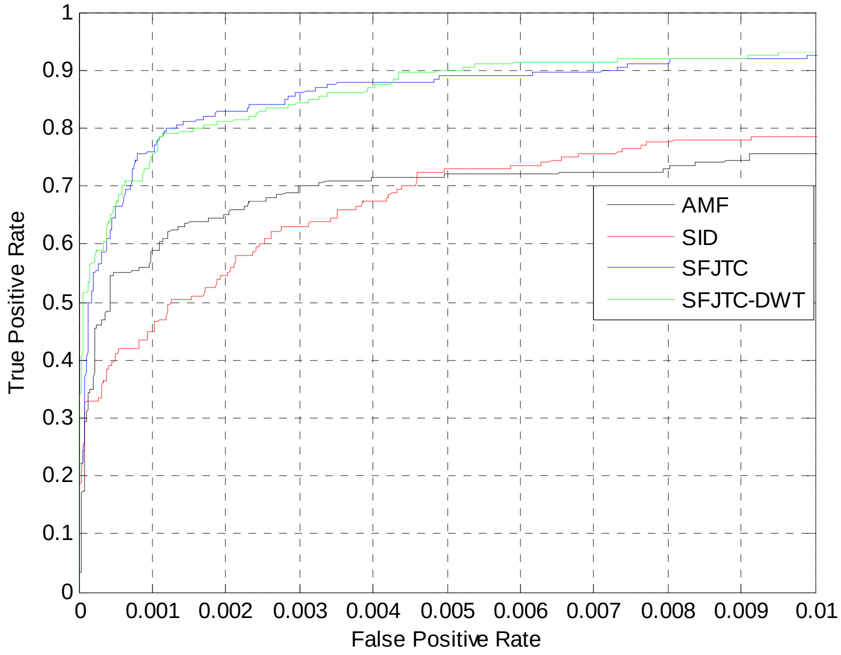


Figure 105. SFJTC ROC curve comparisons for *CASI_urban_8* using adaptive Gaussian model.

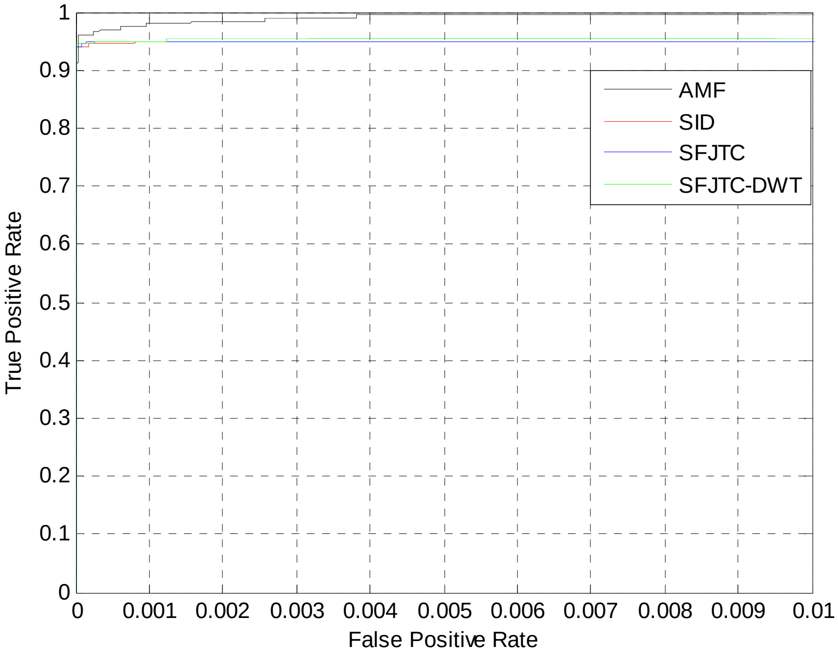


Figure 106. SFJTC ROC curve comparisons for *CASI_veg_15* using adaptive Gaussian model.

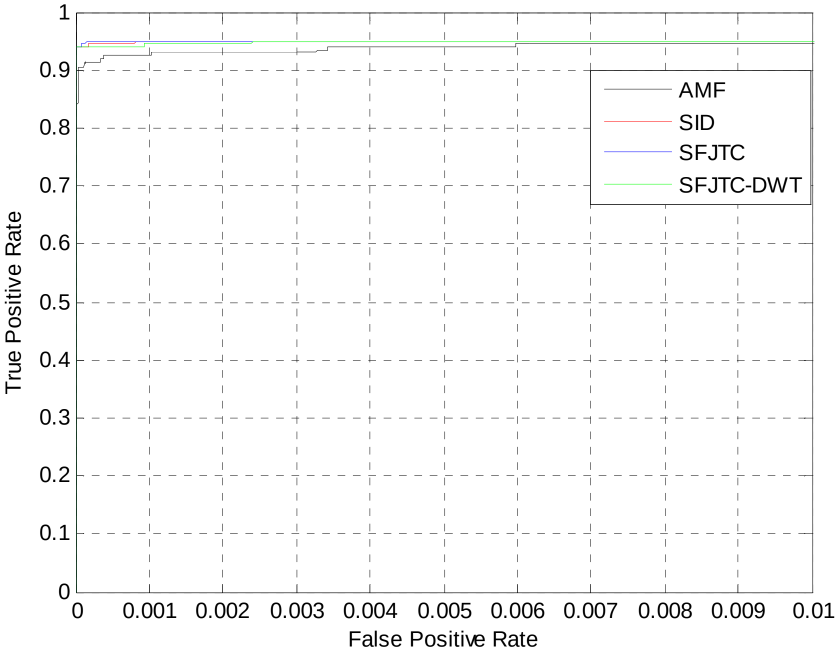


Figure 107. SFJTC ROC curve comparisons for *CASI_veg_12* using adaptive Gaussian model.

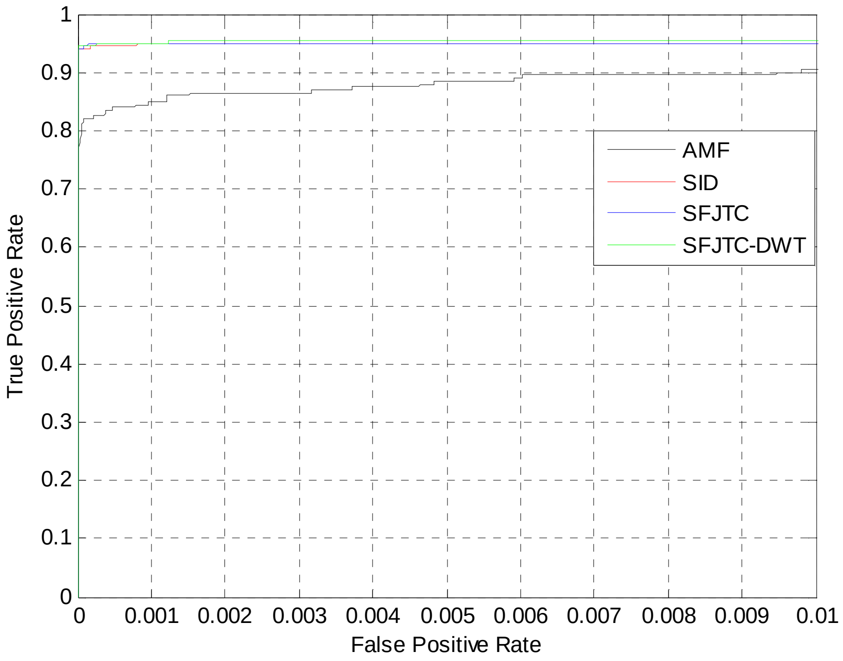


Figure 108. SFJTC ROC curve comparisons for *CASI_veg_10* using adaptive Gaussian model.

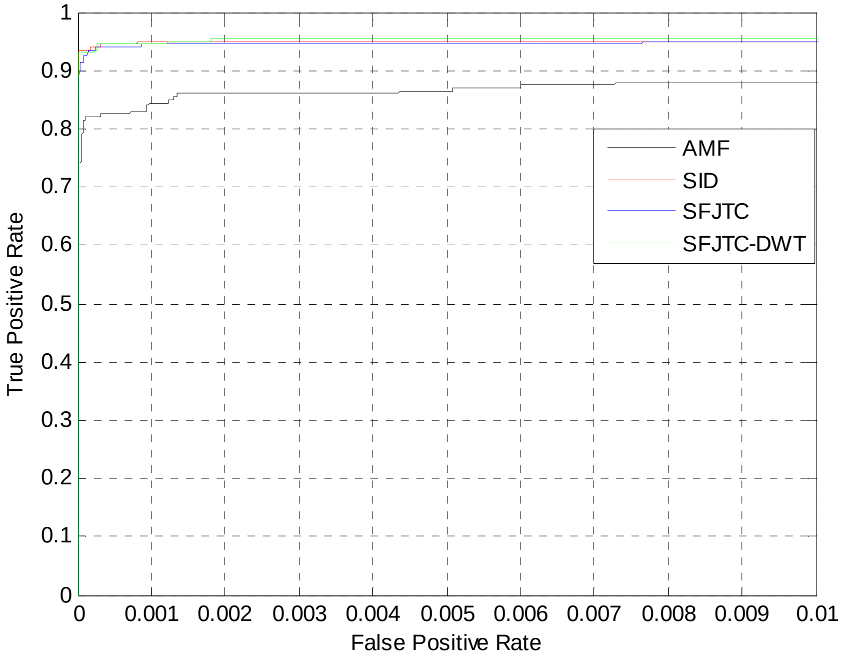


Figure 109. SFJTC ROC curve comparisons for *CASI_veg_8* using adaptive Gaussian model.

Table 85. SFJTC AUROC comparisons for urban CASI scenery using adaptive Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
CASI_urban_15	0.9995	0.3889	0.2464	0.9951	0.9980	0.9987
CASI_urban_12	0.9923	0.4052	0.2518	0.9951	0.9980	0.9987
CASI_urban_10	0.9746	0.4151	0.2629	0.9947	0.9980	0.9987
CASI_urban_8	0.9013	0.4538	0.2558	0.9857	0.9967	0.9974

Table 86. SFJTC AUROC comparisons for vegetative CASI scenery using adaptive Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
CASI_veg_15	0.9998	0.0957	0.3483	0.9915	0.9916	0.9917
CASI_veg_12	0.9892	0.0969	0.3517	0.9915	0.9916	0.9882
CASI_veg_10	0.9662	0.1377	0.3538	0.9915	0.9916	0.9917
CASI_veg_8	0.9568	0.2204	0.3765	0.9915	0.9916	0.9917

As Table 85 shows, SFJTC_DWT provides a marginal improvement over the SFJTC technique using the original signatures. With the exception of *CASI_urban_15*, SFJTC_DWT provides the largest AUROC values. The ROC curves shed more light onto the performance of these detectors. For example, Figure 104 clearly shows that both SFJTC techniques yield steeper ROC curves than the SID and AMF algorithms at extremely low FPRs. For the vegetative CASI scenery, the AMF and SID algorithms provide the largest AUROCs for the two data cubes with least variability, while both SFJTC techniques provide the largest AUROCS for the two data cubes with the most spectral variability. As Table 86 shows, the SID algorithm and SFJTC techniques provide the most consistent AUROCs on all the vegetative CASI scenery.

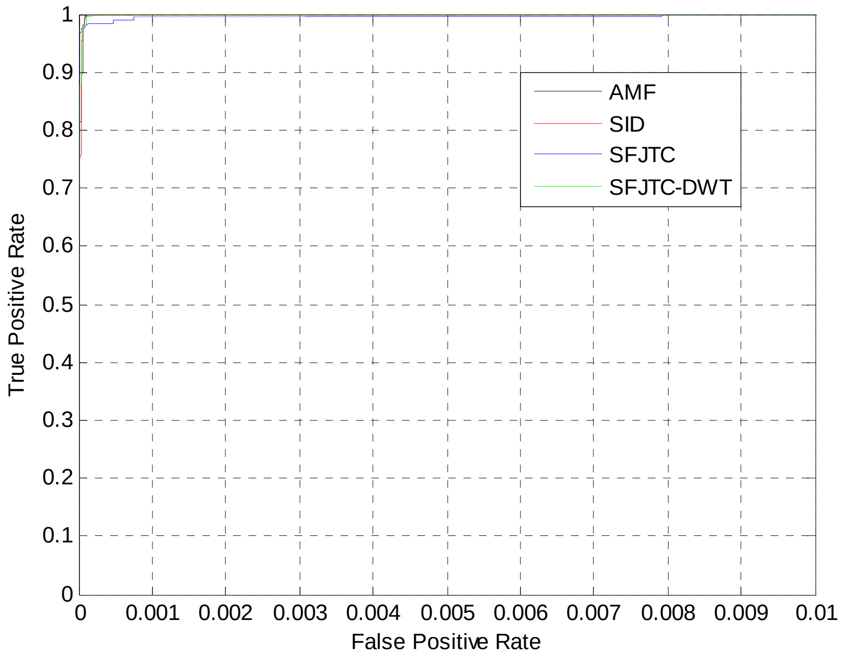


Figure 110. SFJTC ROC curve comparisons for *HYDICE_urban_15* using adaptive Gaussian model.

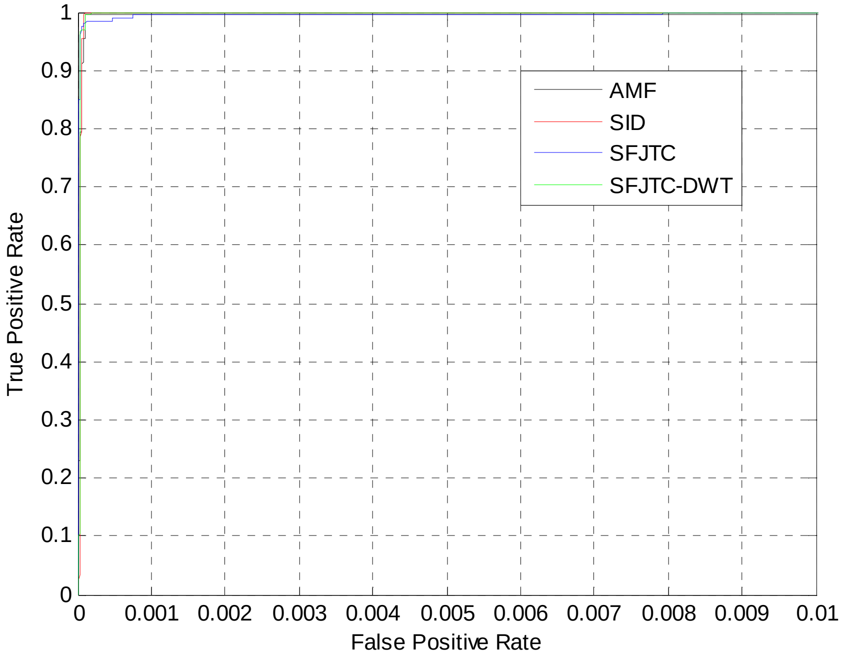


Figure 111. SFJTC ROC curve comparisons for *HYDICE_urban_12* using adaptive Gaussian model.

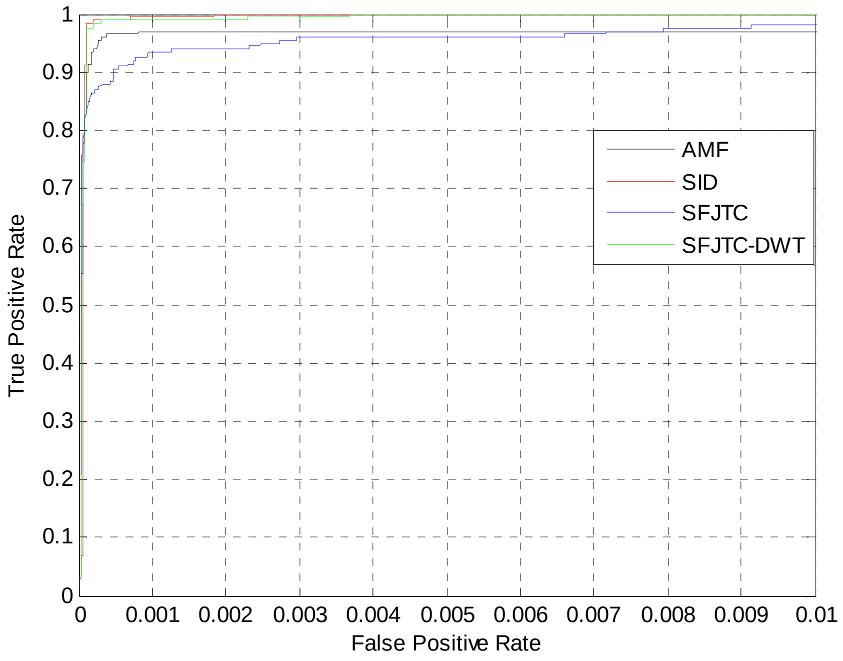


Figure 112. SFJTC ROC curve comparisons for *HYDICE_urban_10* using adaptive Gaussian model.

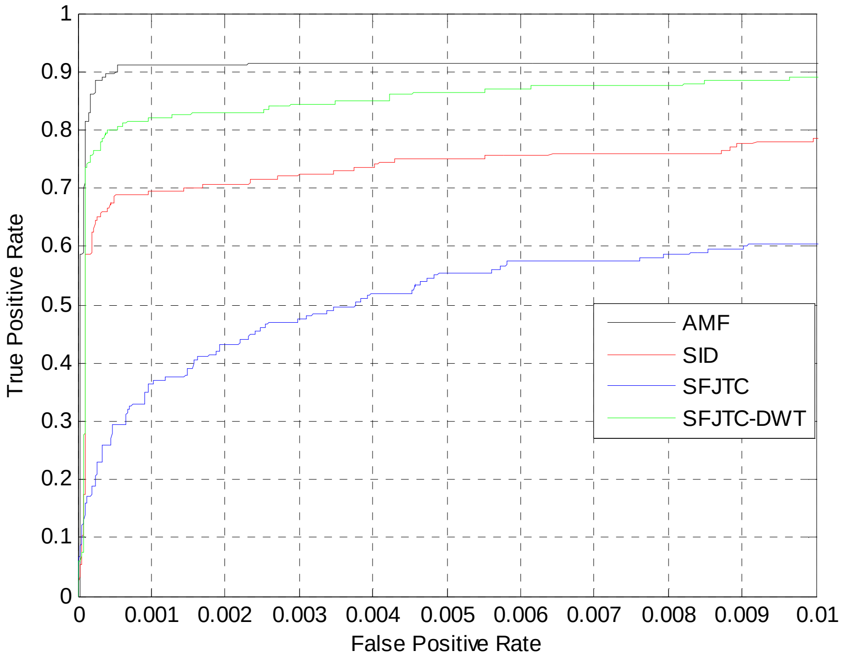


Figure 113. SFJTC ROC curve comparisons for *HYDICE_urban_8* using adaptive Gaussian model.

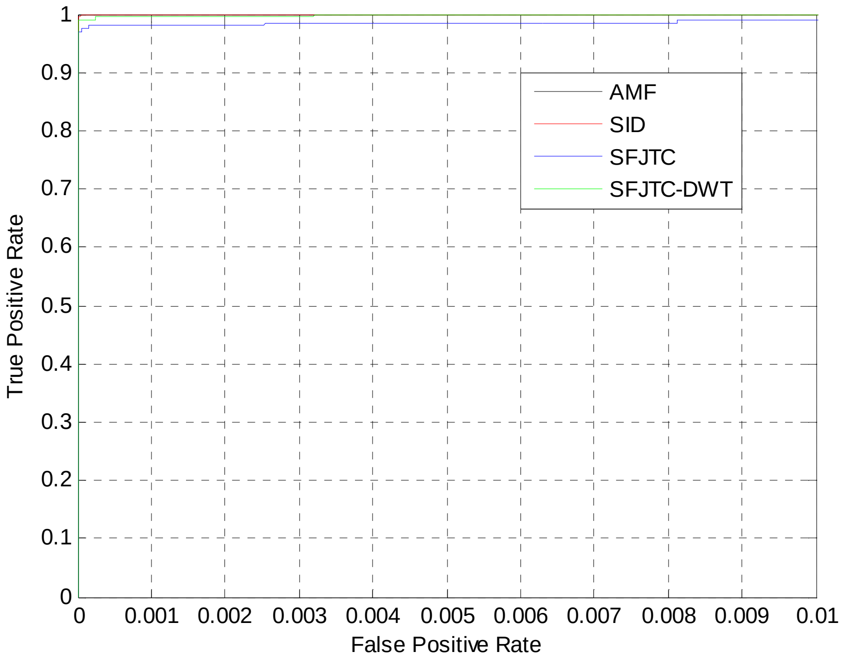


Figure 114. SFJTC ROC curve comparisons for *HYDICE_veg_15* using adaptive Gaussian model.

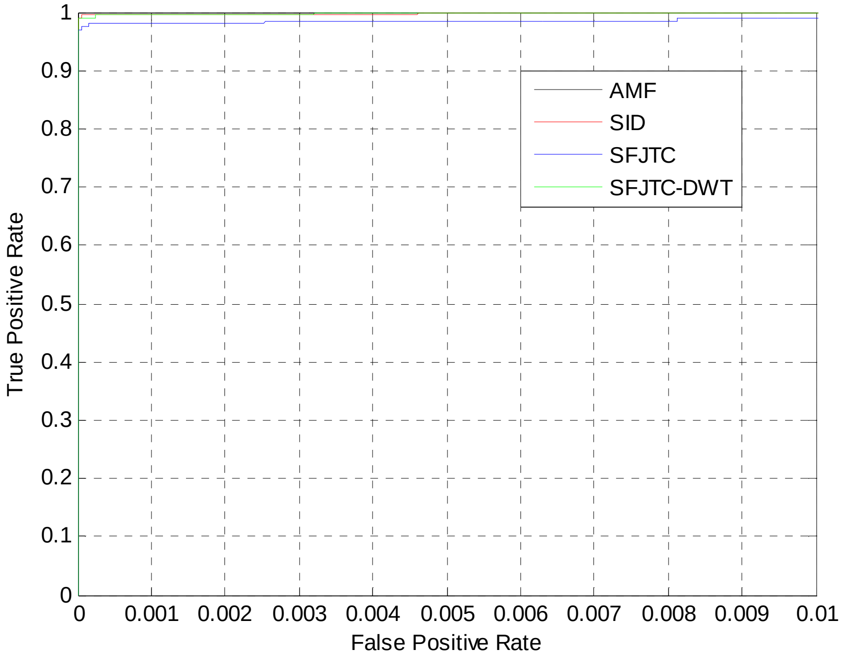


Figure 115. SFJTC ROC curve comparisons for *HYDICE_veg_12* using adaptive Gaussian model.

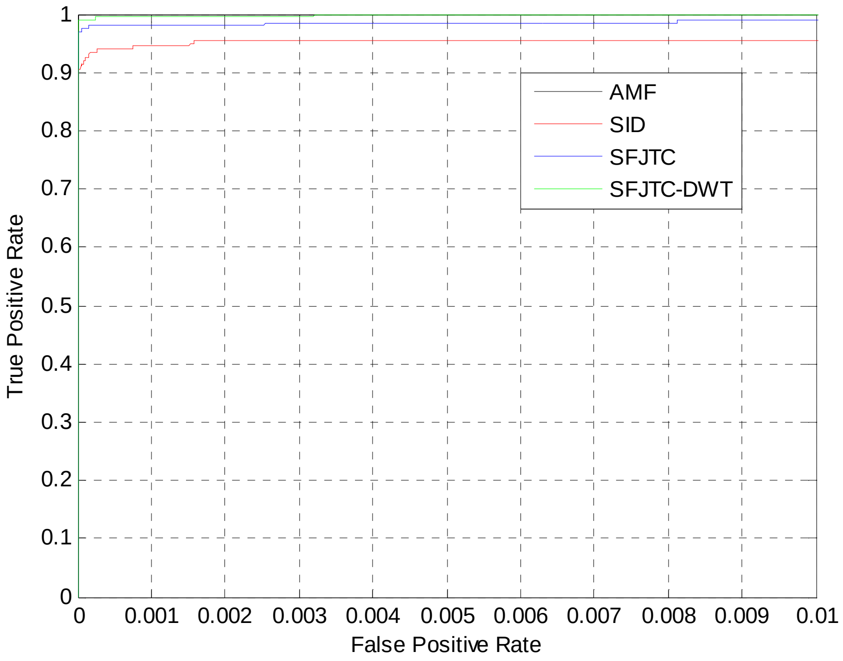


Figure 116. SFJTC ROC curve comparisons for *HYDICE_veg_10* using adaptive Gaussian model.

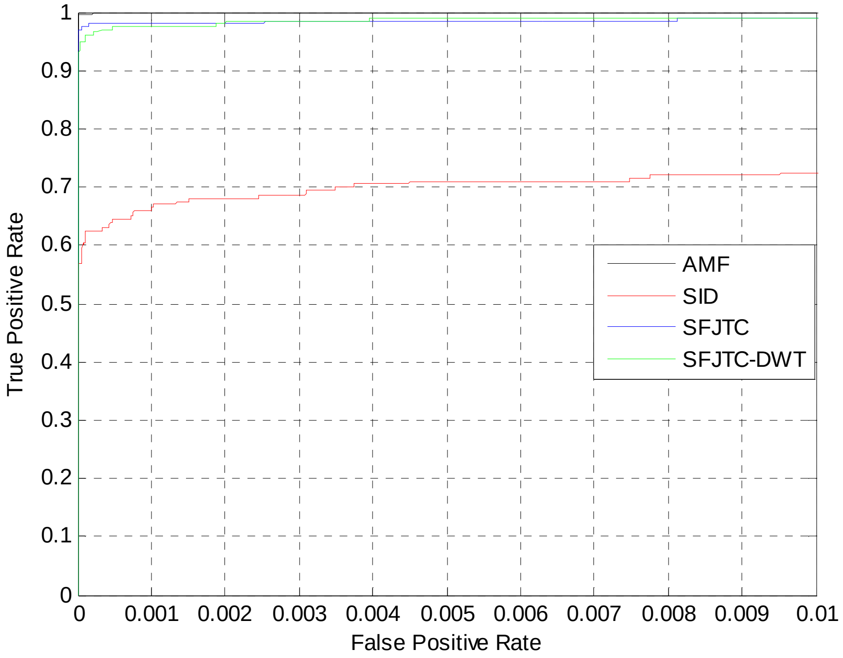


Figure 117. SFJTC ROC curve comparisons for *HYDICE_veg_8* using adaptive Gaussian model.

Table 87. SFJTC AUROC comparisons for urban HYDICE scenery using adaptive Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
HYDICE_urban_15	1.0000	0.9656	0.9463	1.0000	1.0000	1.0000
HYDICE_urban_12	0.9950	0.9501	0.9459	1.0000	0.9999	1.0000
HYDICE_urban_10	0.9780	0.9342	0.9460	0.9999	0.9992	0.9999
HYDICE_urban_8	0.9434	0.9066	0.9457	0.9800	0.9715	0.9752

Table 88. SFJTC AUROC comparisons for vegetative HYDICE scenery using adaptive Gaussian model.

Data Cube	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
HYDICE_veg_15	1.0000	1.0000	0.6948	1.0000	0.9986	1.0000
HYDICE_veg_12	1.0000	1.0000	0.6974	1.0000	0.9986	1.0000
HYDICE_veg_10	1.0000	1.0000	0.6913	0.9857	0.9986	1.0000
HYDICE_veg_8	1.0000	0.9992	0.6925	0.9370	0.9986	0.9995

As Table 87 shows, both SFJTC techniques and the SID algorithm yield stellar detection results for the first three data cubes of the urban HYDICE scenery. For the data cube with heaviest spectral variability, the SID provides a slightly larger AUROC than the SFJTC_DWT technique. Although the AMF and SFJTC_DWT algorithms have lower AUROCs than the SID algorithm for *HYDICE_urban_8*, Figure 113 shows that they yield steeper ROC curves at FPRs less than or equal to one percent.

Both SFJTC techniques yield similar results for the vegetative HYDICE scenery, as shown in Table 88. The AMF algorithm yields perfect detection rates for all levels of spectral variability, while the CEM and SID algorithms yield slightly lower AUROCs. With the exception of the SAM algorithm, all the algorithms yield superior results for the vegetative HYDICE scenery. For HYDICE_veg_8, the SID algorithm breaks down compared to the AMF and SFJTC algorithms as shown in Figure 117 and Table 88.

Table 89. Summary statistics of SFJTC AUROC comparisons for data based on adaptive Gaussian model.

Scenery	AMF	CEM	SAM	SID	SFJTC	SFJTC_DWT
Urban CASI	0.9669 *(0.0450)*	0.4158 *(0.0276)*	0.2542 *(0.0069)*	0.9926 *(0.0047)*	0.9977 *(0.0006)*	0.9984 *(0.0006)*
Vegetative CASI	0.9780 *(0.0199)*	0.1376 *(0.0585)*	0.3576 *(0.0128)*	0.9915 *(0)*	0.9916 *(0)*	0.9908 *(0.0018)*
Urban HYDICE	0.9791 *(0.0256)*	0.9391 *(0.0252)*	0.9460 *(0.0002)*	0.9950 *(0.0100)*	0.9926 *(0.0141)*	0.9938 *(0.0124)*
Vegetative HYDICE	1.0000 *(0)*	0.9998 *(0.0004)*	0.6940 *(0.0027)*	0.9807 *(0.0299)*	0.9986 *(0)*	0.9999 *(0.0002)*
All	0.9810 *(0.0277)*	0.6231 *(0.3738)*	0.5630 *(0.2836)*	0.9900 *(0.0153)*	0.9951 *(0.0071)*	0.9957 *(0.0067)*

Table 89 provides the first and second-order statistics (mean, standard deviation) of the AUROCs for the SFJTC techniques and traditional algorithms. For the urban CASI scenery, the most difficult scenery, SFJTC_DWT outperforms the other algorithms with the largest mean and smallest standard deviation AUROC values. For the vegetative CASI scenery, the SID and SFJTC algorithms provide the best results, with the SFJTC_DWT technique right behind them. For the urban HYDICE scenery, the SFJTC_DWT technique does second best, just shy of the SID algorithm. The AMF algorithm and SFJTC_DWT technique perform better than the SID and remaining algorithms on the vegetative HYDICE scenery. For all of the scenery corresponding to the adaptive Gaussian model, SFJTC_DWT is the best performer, providing the largest mean AUROC and smallest standard deviation of AUROC values.

CHAPTER VIII

CONCLUSIONS

A. Summary

In this work, we have addressed the fundamental problem of spectral variability in HSI target detection. We have obtained scenery from two different sensors in both urban and vegetative environments to test our proposed work.

In Chapter IV, we addressed the challenge of lack of training samples for the target class by creating two unique models to characterize the target class spectral variability. The first model makes no assumptions regarding inter-band correlation, while the second model uses a first-order Markov-based scheme to exploit spectral band-to-band correlation.

In Chapter V, we have developed a scheme that uses the kernel-based SVDD for use in full-pixel target detection scenarios in HSI. We have created an algorithm that addresses optimization of the SVDD kernel width parameter s using the golden-section search algorithm for unconstrained optimization. We investigated a proper number of signatures N to generate for the SVDD target class and found that only a small number of training samples is required relative to the dimensionality (number of bands). As a result, the SVDD-based detection scheme is not plagued by the *Hughes phenomenon* that can cause problems for the stochastic detection and classification algorithms. We have extended decision-level fusion techniques using the majority vote rule for the purpose of alleviating the problem of selecting a proper value of σ^2 for both of our target variability models.

In Chapter VI, we have shown that spectral variability may cause SFJTC-based detection performance to suffer and have addressed this by developing an algorithm that selects an optimal combination of the DWT coefficients of the signatures for use as features for detection for a particular scene and target. Overall, use of the SFJTC technique with the optimal DWT coefficient combination provides increased or identical performance on all the data cubes. In other words, use of the optimal DWT coefficients as features does not negatively affect the detection results in situations where the original signatures perform well.

In Chapter VII, we compared the performance of our SVDD-based and SFJTC-based detection schemes to the traditional stochastic and deterministic detection algorithms presented in Chapter III. In most cases, our SVDD-based detection scheme provides very low FPRs while maintaining higher TPRs than the stochastic AMF and CEM algorithms, especially in scenarios of heavier spectral variability. This holds true for both the simple and adaptive Gaussian models of target variability. For most scenarios, our results show that our SVDD-based detection scheme provides low FPRs while maintaining higher TPRs than the AMF and CEM stochastic detection algorithms. Our results also show that our SFJTC-based detection scheme using the DWT coefficients yields the largest mean AUROC values and smallest standard deviation of AUROC values compared to use of SFJTC using the original signatures and traditional stochastic and deterministic algorithms.

B. Recommendations for Future Work

We believe that future directions for the work in this dissertation should include the following:

- Investigation of various values of ρ between 0 and 1 that explore, in more detail, the impact of spectral band-to-band correlation using our adaptive Gaussian model of target variability.
- Investigation of Markov-based models that extend beyond the first order for more accurately capturing the spectral band-to-band correlation.
- In connection with the SVDD, the investigation of performing fusion of detectors at the *feature* level rather than the *decision* level.
- Investigation of modification of the SVDD technique that would weight the training samples according to a similarity criterion to the target signature.
- Investigation of use of the SVDD technique with negative examples[8] for potentially increasing detection performance for difficult scenery.
- Investigation of the use of the SVDD technique with sophisticated feature selection algorithms that retain useful bands and discard unwanted bands prior to detection.
- The investigation of alternative methods of combining the individual DWT approximation and detail coefficients for use as features for detection in SVDD-based and SFJTC-based detection schemes.
- Investigation into the use of other wavelet families for generating the DWT coefficients of the hyperspectral signatures.

REFERENCES

1. S. M. Yamany, A. A. Farag, and S.-Y. Hsu, “A fuzzy hyperspectral classifier for automatic target recognition (ATR) systems,” *Pattern Recognition Letters* **20,** 1431-1438 (1999).

2. D. Manolakis and G. Shaw, “Detection algorithms for hyperspectral imaging applications,” *IEEE Signal Processing Magazine* **19**, 29-43 (2002).

3. D. Manolakis, D. Marden, and G. Shaw, “Hyperspectral image processing for automatic target detection applications,” *Lincoln Laboratory Journal* **14**, 79-114 (2003).

4. N. Henz and T. Wagner, “A new approach to the BHEP tests for multivariate normality,” *Journal of Multivariate Analysis* **62**, 1-23 (1997).

5. J. E. West, *Matched filter stochastic background characterization for hyperspectral target detection*, Masters Thesis, Rochester Institute of Technology, Rochester, NY (August 2005).

6. B. Scholkopf and A. Smola, *Learning with Kernels-Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA (2002).

7. V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York (1998).

8. D. M. J. Tax and R. P. W. Duin, “Support vector data description,” *Machine Learning* **54**, 45-66 (2004).

9. Z. Zeng, Y. Fu, G. I. Roisman, Z. When, Y. Hu, and T. Huang, “One-class classification for spontaneous facial expression analysis,” *International Conference on Automatic Face and Gesture Recognition*, pp. 281-286 (2006).

10. R. Ji, D. Liu, M. Wu, and J. Liu, “The application of SVDD in gene expression data clustering,” *International Conference on Bioinformatics and Biomedical Engineering*, pp. 371-374 (2008).

11. C. Lai, D. M. J. Tax, R. P. W. Duin, E. Pekalska, and P. Paclik, “A study on combining image representations for image classification and retrieval,” *International Journal of Pattern Recognition and Artificial Intelligence* **18**, 867-890 (2004).

12. J. Munoz-Marf, L. Bruzzone, and G. Camps-Vails, “A support vector domain description approach to supervised classification of remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing* **45**, 2683-2692 (2007).

13. C. Sanchez-Hernandez, D. Boyd, and G. M. Foody, “One-class classification for mapping a specific land-cover class: SVDD classification of Fenland,” *IEEE Transactions on Geoscience and Remote Sensing* **45**, 1061-1073 (2007).

14. A. Banerjee, P. Burlina, and C. Diehl, “A support vector method for anomaly detection in hyperspectral imagery,” *IEEE Transactions on Geoscience and Remote Sensing* **44**, 2282-2291 (2006).

15. A. Banerjee, P. Burlina, and R. Meth, “Fast hyperspectral anomaly detection via SVDD,” *IEEE International Conference on Image Processing* **4**, 101-104 (2007).

16. M. S. Alam, M. Haque, J. F. Khan, and H. Kettani, "Fringe-adjusted joint transform correlator based target detection and tracking in forward looking infrared image sequence," *Optical Engineering* **43**, 1407-1413 (2004).

17. M. S. Alam and M. A. Karim, "Fringe-adjusted joint transform correlation," *Applied Optics* **32**, 4344-4350 (1993).

18. M. S. Alam and S. Ochilov, “Target detection in hyperspectral imagery using one-dimensional fringe-adjusted joint transform correlation (Invited Paper),” in *Proceedings of the SPIE Conference on Optical Pattern Recognition* **6245**, pp. 1-11 (2006).

19. W. Sakla, A. Sakla, and M. S. Alam, “Deterministic hyperspectral target detection using the DWT and spectral fringe-adjusted joint transform correlation (Invited Paper),” in *Proceedings of the SPIE Conference on Automatic Target Recognition* **6967**, pp. 1-11 (2008).

20. G. Shaw and H. Burke, “Spectral imaging for remote sensing,” *Lincoln Laboratory Journal* **14**(1), 3-28 (2003).

21. MicroImages, “Introduction to hyperspectral imaging,” http://www.microimages.com, accessed in 2006.

22. D. Landgrebe, “Hyperspectral image data analysis,” *IEEE Signal Processing Magazine* **19**, 17-28 (2002).

23. D. Manolakis, “Taxonomy of detection algorithms for hyperspectral imaging applications,” *Optical Engineering* **44**(6), 1-11 (2005).

24. C.-I. Chang, *Hyperspectral Data Exploitation*: *Theory and Applications*, Wiley, Hoboken, NJ (2007).

25. C. Stellman, G. Hazel, F. Bucholtz, J. Michalowicz, A. Stocker, and W. Schaaf, "Real-time hyperspectral detection and cuing," *Optical Engineering* **39**, 1928-1935 (2000).

26. D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Processing Magazine* **19**, 58-69 (2002).

27. A. R. Webb, *Statistical Pattern Recognition*, 2nd ed., Wiley, Hoboken, NJ (2002).

28. C.-I. Chang, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenium, New York, NY (2003).

29. C.-I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Transactions on Information Theory* **46**, 1927-1932 (2000).

30. ITRES Research, http://www.itres.com, accessed in 2007.

31. R. A. Schowengerdt, *Remote Sensing*, 2nd ed., Academic Press, San Diego (1997).

32. I. C. Chein, and D. C. Heinz, "Constrained subpixel target detection for remotely sensed imagery," *IEEE Transactions on Geoscience and Remote Sensing* **38**(3), 1144-1159 (2000).

33. A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, Englewood Cliffs, NJ (1989).

34. D. W. Scott, *Multivariate Density Estimation*, Wiley, New York (1992).

35. D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognition Letters* **20**, 1191-1199 (1999).

36. D. Jesse and G. Mark, "The relationship between precision-recall and ROC curves," *ACM International Conference on Machine Learning*, pp. 233-240 (2006).

37. S. C. Chapra and R. C. Canale, *Numerical Methods for Engineers*, fifth edition, McGraw-Hill, New York, NY (2008).

38. J. Kitler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**, 226-239 (1998).

39. L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 281-286 (2002).

40. A. Cheriyadat, L. M. Bruce, and A. Mathur, "Decision level fusion with best-bases for hyperspectral classification," *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data*, pp. 399-406 (2003).

41. M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Decision fusion for the classification of urban remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing* **44**, 2828-2838 (2006).

42. T. West, S. Prasad, and L. M. Bruce, "Multiclassifiers and decision fusion in the wavelet domain for exploitation of hyperspectral data," *IEEE International Geoscience and Remote Sensing Symposium*, pp. 4850-4853 (2007).

43. E. H. Horache and M. S. Alam, "Nonzero-order fringe-adjusted joint transform correlation using binary random phase mask," *Proceedings of the SPIE* **5807**, pp. 341-348 (2005).

44. M. S. Alam, A. Bal, E. H. Horache, S. F. Goh, C. H. Loo, S. P. Regula, and A. Sharma, "Metrics for evaluating the performance of joint-transform-correlation-based target recognition and tracking algorithms," *Optical Engineering* **44**, paper no. 067005 (2005).

45. B. Javidi and Q. Tang, "Chirp-encoded joint transform correlators with a single input plane," *Applied Optics* **33**, 227-230 (1994).

46. Q. Tang and B. Javidi, "Multiple-object detection with a chirp-encoded joint transform correlator," *Applied Optics* **32**, 5079 (1993).

47. S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, 674-693 (1989).

48. M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, Upper Saddle River, NJ (1995).

49. S. Mallat, *A Wavelet Tour of Signal Processing*, 2nd. ed., Academic Press, New York, NY (1999).

50. R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Transactions on Information Theory* **38**, 719-746 (1992).

51. S. G. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing* **9**, 1532-1546 (2000).

52. T. Chang and C. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing* **2**, 429-441 (1993).

53. L. M. Bruce and J. Li, "Wavelets for computationally efficient hyperspectral derivative analysis," *IEEE Transactions on Geoscience and Remote Sensing* **39**, 1540-1546 (2001).

54. L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Transactions on Geoscience and Remote Sensing* **40**, 2331-2338 (2002).

55. S. Kaewpijit, J. Le Moigne, and T. El-Ghazawi, "Automatic reduction of hyperspectral imagery using wavelet spectral analysis," *IEEE Transactions on Geoscience and Remote Sensing* **41**, 863-871 (2003).

56. L. M. Bruce, C. Morgan, and S. Larsen, "Automated detection of subpixel hyperspectral targets with continuous and discrete wavelet transforms," *IEEE Transactions on Geoscience and Remote Sensing* **39**, 2217-2226 (2001).

VITA

Wesam Adel Sakla received his Bachelor of Science degree in computer engineering and his Master of Science degree in electrical engineering from The University of South Alabama, Mobile, Alabama, in 2002 and 2005, respectively. Wesam joined the signal processing group in the electrical and computer engineering department at Texas A&M in 2005. He received his Doctor of Philosophy degree in electrical engineering in December 2009. His research interests include image processing, time-frequency analysis, automatic target recognition, kernel-based methods for pattern classification, support vector machines, neural networks, and biometric recognition.

Wesam may be reached through the electrical and computer engineering department at 214 Zachry Engineering Center, College Station, TX 77843-3128. His email is sakla1@gmail.com.