Matched Filter Stochastic Background Characterization for Hyperspectral Target Detection

by

Jason E. West

Submitted to the Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements for the Master of Science Degree

College of Science

Rochester Institute of Technology

2005

Signature of the Author ______________________________

Jason E. West

Accepted by ______________________________

Coordinator, Graduate Degree Program

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Captain Jason E. West, USAF
has been examined and approved by the thesis committee as satisfactory
for the thesis required for the M.S. degree in Imaging Science

____
Dr. John Schott, Thesis Advisor

____
Dr. John Kerekes

____
Dr. David Messinger

____
Date

THESIS RELEASE PERMISSION

ROCHESTER INSTITUTE OF TECHNOLOGY

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis

Matched Filter Stochastic Background Characterization
for Hyperspectral Target Detection

I, Captain Jason E. West, USAF, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature____

Date

Matched Filter Stochastic Background Characterization for Hyperspectral Target Detection

by
Jason E. West

ABSTRACT

Algorithms exploiting hyperspectral imagery for target detection have continually evolved to provide improved detection results. Adaptive matched filters, which may be derived in many different scientific fields, can be used to locate spectral targets by modeling scene background as either structured (geometric) with a set of endmembers (basis vectors) or as unstructured (stochastic) with a covariance matrix. In unstructured background research, various methods of calculating the background covariance matrix have been developed, each involving either the removal of target signatures from the background model or the segmenting of image data into spatial or spectral subsets. The objective of these methods is to derive a background which matches the source of mixture interference for the detection of sub pixel targets, or matches the source of false alarms in the scene for the detection of fully resolved targets. In addition, these techniques increase the multivariate normality of the data from which the background is characterized, thus increasing adherence to the normality assumption inherent in the matched filter and ultimately improving target detection results. Such techniques for improved background characterization are widely practiced but not well documented or compared. This thesis will establish a strong theoretical foundation, describing the necessary preprocessing of hyperspectral imagery, deriving the spectral matched filter, and capturing current methods of unstructured background characterization. The extensive experimentation will allow for a comparative evaluation of several current unstructured background characterization methods as well as some new methods which improve stochastic modeling of the background. The results will show that consistent improvements over the scene-wide statistics can be achieved through spatial or spectral subsetting, and analysis of the results provides insight into the trade-spaces of matching the interference, background multivariate normality and target exclusion for these techniques.

Contents

List of Figures

Section 2 Figures

Section 3 Figures

Section 4 Figures

Appendix A Figures

List of Tables

1.0 Introduction

Hyperspectral Imagery Overview

Hyperspectral imagery (HSI) may be defined as imagery taken over many (usually more than one hundred) spectrally contiguous and spatially co-registered bands. In contrast to panchromatic imagery, which integrates light over a wide range of wavelengths, or multispectral imagery, which selectively captures light of several different narrow bands, HSI data contains a nearly continuous electromagnetic (EM) spectrum behind each spatial pixel in a scene. The motivation for building costly HSI sensors lies in the added information content of the spectral dimension. While few materials can be distinguished in a pan image, most materials have somewhat unique characteristics across the EM spectrum. There are many ways to use HSI data to perform spectral analysis, including classification, anomaly detection, and target detection. Classification is the process of assigning a land cover type (or class) to each pixel within a scene. Anomaly detection refers to locating pixels in the scene that are different from all other pixels. Target detection attempts to locate pixels containing a target material of known spectral composition. While techniques from all three processes will be employed in this work, target detection will be the main focus.

The Target Detection Problem

Target detection has many applications, including military reconnaissance and environmental studies. Searching for the presence of a specific material over a large area poses many practical difficulties. The prospect of using remotely sensed HSI to perform this task in an accurate and timely manner has driven the research community to generate many different types of target detection algorithms. Spectral matched filter target detection is a well developed technique using a known target spectrum to search for the presence of that spectrum in a scene. Most forms of the spectral matched filter contain a model of scene background (often called clutter) which is used to suppress the appearance of background and increase target contrast. Many detectors may use structured (geometric) backgrounds, which are formed from collections of the most spectrally “pure” pixels in the scene, called endmembers. Other detectors model background with unstructured (stochastic) representations, which take the form of first and second order spectral statistics (mean and covariance) estimated from the scene data. Because background characteristics change from scene to scene, these filters must adapt based on specific image content. The adaptive matched filter (AMF) is a detector that models and suppresses an unstructured background and then uses a known target spectrum to search for that target in an HSI data set. Much of the work to improve AMF performance has focused on improved scaling of detector results. By scaling the data to increase separation between target detects and false alarms, a threshold can be more easily set to automate the decision process.

Another and perhaps more fruitful approach to improved detection results is to improve the estimation of background made by the AMF, thus increasing the suppression of unwanted signal and creating greater separation between target detects and false alarms before scaling. Due to the assumption inherent in the thresholded AMF of a multivariate Gaussian (or normal) background, improvements to background models can be achieved by increasing the multivariate normality (MVN) of data used to represent image background. Based on this concept, several methods for calculating background statistics have been developed in the field which are reported to improve detection results for specific targets, detectors, and HSI data sets. Each of these methods takes cues from the imagery to establish a rationale for selecting which data is to be included in the estimation of background statistics. Establishing this rationale to model scene background and selecting a mathematical technique to calculate background statistics are together called the characterization of background.

Approach

The work performed for this thesis involved the implementation of several algorithms currently in the literature, along with the development of several new techniques that build upon or depart from previous work. Methods employing target exclusion, spatial subsetting (windowing), and spectral subsetting (classification) were tested for improved background modeling. First, a manual method of windowing was tested against the full scene statistics. Next, the sliding window background characterization method of the RX algorithm was implemented to test the performance of automated spatial subsetting. The K-Means and Stochastic Expectation Maximization (SEM) classification algorithms were applied to the spectral subsetting problem, and a modification of SEM called the Statistical Distance Classifier (SDC) was used to improve the predictability of results. Several methods combining the spatial and spectral subsetting of data were explored to learn about the phenomenology of interference and false alarms.

Spatial and spectral methods of target exclusion were applied to each technique in an attempt to characterize the vulnerability of different subsetting methods to the presence of target pixels in the background data pool (target contamination). To measure adherence to the MVN assumption, the chi-squared multivariate normality test with a goodness of fit metric was applied to each type of background. In order to evenly test these backgrounds as applied to detection, a single scaling method was selected from the family of unstructured matched filters. The generalized likelihood ratio test (GLRT) detector was applied to find several different targets in two different scenes. Finally, receiver operator characteristic (ROC) curves were used in combination with a summary average false alarm rate (AFAR) metric to quantify detection results.

The experiment investigated a correlation of the methods for estimating background statistics, the normality of the background, and the sensitivity of these techniques to target contamination, with the rate of detect versus false alarm performance. The results of the experiment lead to the identification of the best background characterization techniques to employ for a given type of detection problem.

2.0 Background and Literature Review

2.1 HSI Processing

The target detection problem begins with a raw HSI data set (in digital counts) which is passed through a number of processing steps before a matched filter is applied. Calibration, atmospheric compensation, dimensionality reduction, and background estimation are generally performed in order to seed a detector with useful data. Several techniques have been developed to perform these steps, and there are many parameters to set along the way. The interdependencies of parameters selected in the collection, processing, and exploitation of HSI data are an important consideration when working to advance any portion of the process. Each of the many techniques will influence the final result in some manner. In order to ensure constructive research for a single technique in a certain step, it is important to understand the entire process, from photon to decision, and to consider the effect of each method and parameter.

The hyperspectral imaging chain for visible and near infrared (VISNIR) imagery begins with a physics based model for photons propagating from the sun to the sensor aperture. The governing equation for the radiance reaching a sensor from solar photons may be expressed

$$L_{s\lambda}[wm^{-2}sr^{-1}\mu m^{-1}]=[E'_{s\lambda}\cos\sigma'\tau_1(\lambda)\frac{r(\lambda)}{\pi}+FE_{d\lambda}\frac{r_d(\lambda)}{\pi}+(1-F)L_{b\lambda avg}r_d(\lambda)]\tau_2(\lambda)+L_{u\lambda} \qquad 2.1$$

where E'$_s$ is the exoatmospheric solar irradiance, σ' is the illumination zenith angle, τ_1 is the atmospheric transmission from the sun to the target, F is the fraction of the hemisphere above the target filled by sky, E_d is downwelled irradiance, r_d is target reflectance assuming a diffuse target, L_{bavg} is the average background radiance, τ_2 is the atmospheric transmission from the target to the sensor, and L_u is the upwelled radiance [Schott, 1997]. This equation assumes an approximately uniform (Lambertian) target and approximately diffuse downwelled and background radiance terms, and includes angular dependencies not shown. From equation 2.1, it is apparent that photons leaving the sun, interacting with the target, and eventually entering the sensor are affected by atmospheric transmission and scattering as well as viewing angle. Once at the aperture, photons propagate through the sensor optics and are eventually measured by the detector array and converted to electrons. The resulting electrical signal passes through sensor electronics and is stored in memory in the form of digital counts.

In order to use the data for traditional matched filtering, the process can be reversed in calculation to obtain a measure of surface reflectance. Calibration is the process of converting sensor digital count (raw sensor data) to radiance at the aperture. Optics are tested and characterized during fabrication and each sensor component has a known effect on the output signal. An attempt to correct for imperfect diffraction of photons entering the front aperture or modulation of the signal occurring between the detector array and the memory device can be performed by sensor calibration. The next processing step is atmospheric compensation, which attempts to remove the effects of atmospheric absorption and scattering as well as illumination angle from the radiance image, converting radiance at the aperture to reflectance at the target surface. Some preprocessing methods combine the two steps, converting from digital count directly to reflectance. Otherwise, HSI data sets are obtained in radiance units or are accompanied by sensor models, and a preferred method of atmospheric compensation can be applied.

Due to the high dimensionality of HSI data, another common preprocessing step reduces the number of bands or spectral channels in the data cube. Bad bands lists, which often accompany data sets, are used to exclude spectral bands that are in atmospheric absorption features or are problematic due to sensor defects. Other methods use vector algebra to rotate the data into new axes where some of the data can be discarded without significant loss of information. The resulting HSI cube contains most of the image information, little noise or meaningless spectral content, and is ready to be processed by classification and anomaly or target detection algorithms.

Before any type of target detection can be performed, the HSI data is often preprocessed to correct for the atmosphere and reduce the dimensionality of the data in order to produce a computationally manageable data set. In order to best assign meaning to a detection result, it is important to understand both the underlying assumptions of preprocessing and the resulting data upon which the detectors are applied. The following sections will discuss these preprocessing steps in more detail.

2.1.1 Atmospheric compensation

Many methods for atmospheric compensation have been developed, including CIBR [Carrere and Conol, 1993], ATREM [Gao et al, 1993], and NLLSSF [Green et al, 1993]. Each of these uses different techniques to estimate atmospheric content and apply a radiative transfer model to the data. Another technique, which is simple and commonly used, is the empirical line method

(ELM). Given field measured ground truth spectra and the location of corresponding radiance pixels within the scene, ELM generates a linear fit to solve for a gain and offset according to the equation

$$L = g\,r + b \qquad 2.2$$

where r is reflectance, g is gain, L is radiance, and b is offset bias [Schott, 1997]. In order to achieve a linear fit that best represents the data, an attempt is usually made to maximize the number of truth spectra -- and thus the number of points for the fit -- and to populate the ELM data with extrema in reflectance (i.e. selecting both bright and dark objects, like concrete and water). The estimated linear transform is then applied to every pixel in the scene, employing an assumption of uniform atmospheric content and illumination conditions across the sensor field of view. The result is an HSI data set transformed into reflectance units, which can then be compared to ground truth reflectance spectra. It is also possible to use ELM to convert directly from raw digital counts to reflectance as long as the sensor model is assumed to be linear. The goal of either is to obtain a scene and target spectrum that are in the same units. It is also important to note, keeping in mind that a statistics based target detector is the final step of the process, that linear atmospheric compensation routines will not change the image statistics and will therefore not interfere with a comparison of statistics based detection results [Manolakis et al, 2001]. A representation of ELM is shown in Figure 2.1, which depicts a line of best fit through several ground truth points, providing a linear transformation between radiance and reflectance.

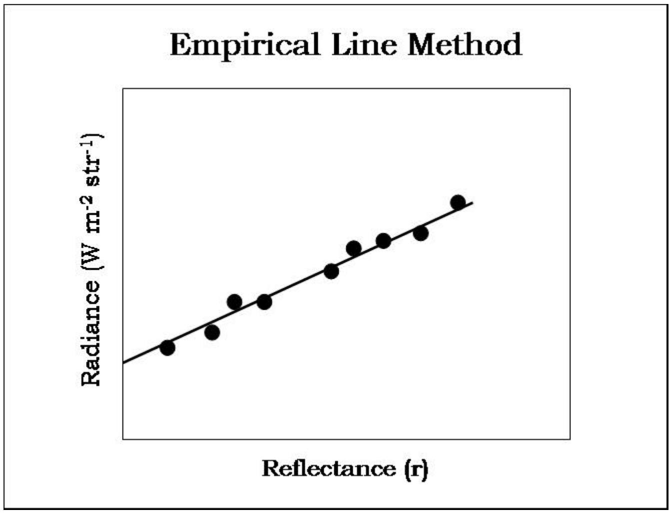


Figure 2.1 ELM Representation

2.1.2 Dimensionality Reduction

An HSI data cube typically contains hundreds of spectral bands and thousands of pixels (observations) which can lead to unmanageable computation times when performing matrix operations on each pixel. Additionally, much of the data contained in a typical HSI scene (e.g. bad data or noise) can be detrimental to detection algorithms. Many dimensionality reduction techniques have been developed to ameliorate these problems. The first and most obvious is the exclusion of bad bands from the data. Many imaging systems include bands in water absorption portions of the spectrum. Obviously, bands where the atmosphere is opaque will obscure the true spectral shape of the data and confound detection algorithms. The number of bad bands will depend on the sensor and the conditions at the time of image capture. Bad band exclusion is a helpful, simple, and commonly used technique, but the number of bands that may be eliminated by this process are limited.

Other techniques, like principle component analysis (PCA), have been developed to provide greater dimensionality reduction [Schott, 1997]. PCA is based on the assumption that statistical variation in the data is related to information content. By transforming the data into a space where the bands are uncorrelated and then removing bands that contain the least statistical variation (information), PCA can dramatically reduce the number of bands in an image. This transformation is devised through a solution to the generalized eigenvalue problem

$$\Sigma = U^T \Lambda\, U \tag{2.3}$$

where Σ is the covariance matrix, U is an orthogonal matrix ($U^{-1} = U^T$) with columns containing the eigenvectors, and Λ is a diagonal matrix containing the eigenvalues (variances) along the diagonal. The variances are typically ordered from largest to smallest (denoted σ_1, σ_2, σ_3,....,σ_N), and the eigenvectors in U reordered accordingly. The PCA transform is then performed by multiplying each pixel (x) in the original data by the matrix of ordered eigenvectors (E), thus resulting in data where the covariance matrix is uncorrelated, which can be expressed

$$X_{PC} = E^T x \tag{2.4}$$

where X_{PC} is the principle component transform result. The rotations involved in PCA alter, but do not corrupt, the scene statistics. However, discarding bands with little to no information is a nonlinear process that will alter the content and form of scene statistics.

2.1.3 Spectral Mixing Models

Due to the low spatial resolution of HSI data, most target detection algorithms apply to subpixel targets. If the ground instantaneous field of view (GIFOV) of a detector element in the sensor is larger than the target of concern, the image pixel containing the target will also contain other materials. It is therefore important to understand the physical construction of the data and to adopt a model of pixel formation. The most commonly used representation is the linear mixing model (LMM). This model assumes that if photons arriving at a detector element originate at different materials within the spatial coverage area of the pixel, the resulting spectrum will be a linear combination of the individual spectra of the constituent materials. Spectrally distinct materials which combine to form pixel spectra are often called endmembers. The LMM for a mixed pixel, represented by the vector X(x,y), in an ℓ-band image containing p endmembers may be expressed as

$$X(x,y) = M\alpha(x,y) + n(x,y) \qquad 2.5$$

where M is an ℓ x p matrix containing the endmember spectra, α is a p-dimensional vector containing the fractional abundances of each endmember in M, and n is an additive random ℓ-dimensional noise vector [Harsanyi and Chang, 1994]. Classification based on the LMM is often accomplished by least-squares inversion to reverse the process, unmix the pixel spectra, and solve for the abundance map α(x,y).

Another model for how pixels are formed in HSI is the stochastic mixing model (SMM). Instead of assuming that a mixed pixel is populated by linear combinations of endmembers, SMM assumes combinations of vectors randomly selected from multivariate Gaussian probability endmember distributions [Stocker and Schaum, 1997]. The SMM may be represented as

$$x = M_1\alpha + (1-\alpha)M_2 \tag{2.6}$$

where M_1 and M_2 are two endmembers selected independently from their respective distributions and α is the mixing fraction [ibid]. This can be adapted for target detection simply by changing the nomenclature, replacing M_1 with a target distribution M_T and M_2 with a background distribution M_B. The additive noise factor is assumed insignificant or included in the endmember distributions. [Manolakis et al, 2001]. The SMM may also be applied to combine image statistics, a technique discussed in a later section.

2.1.4 Classification

Image classification is a technique widely used to determine land cover in multispectral or hyperspectral data. Classification algorithms are either supervised, using in-scene training data specified by a user, or unsupervised, using mathematical techniques that do not require training data. K-means and stochastic expectation maximization (SEM) are two types of unsupervised classification that may be useful for improved target detection.

K-means is a simple and commonly used unsupervised image classification method which requires the user only to specify the number of classes. Random class means are generated and each image pixel is assigned to a class by calculating a minimum distance to the mean. New class means are calculated based on the pixel assignments and the process is repeated until the means stop changing by a thresholded amount [Schott, 1997]. Figure 2.2 is a depiction of the k-means classification process, where three distinct classes are present in a two band image. The red points are class means based first on random pixel assignment, then on assignment of pixels with smallest Euclidean distance. These points get progressively closer to the class means and stop migrating when they represent stable means of each class.

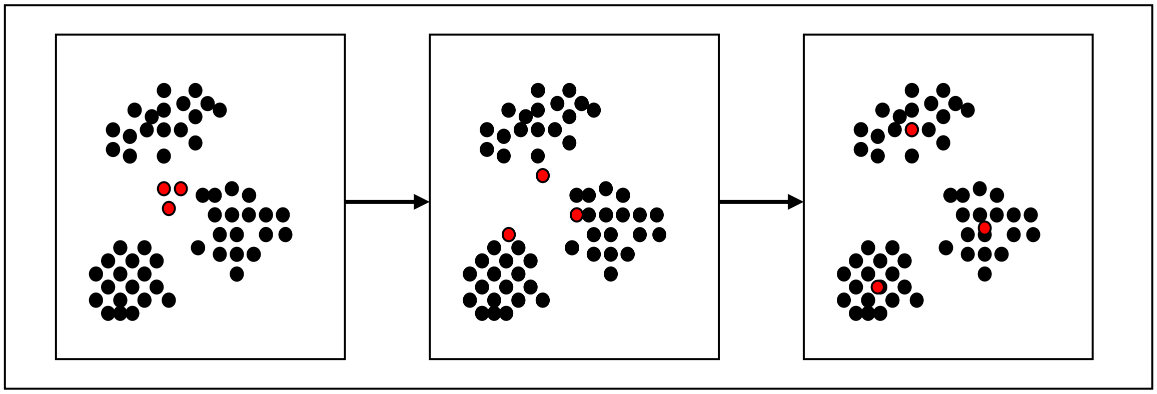


Figure 2.2 K-Means Classification

SEM is another approach to classification which considers not only first order, but also second order information when assigning pixels to a class. The SEM algorithm requires the user to specify the maximum possible number of classes and a minimum class size threshold, and the result is the number of classes and a mean and covariance for each class. Classes are modeled as normal distributions, combining additively to form the image. The conditional probability of a pixel belonging to a certain class can be modeled by a Gaussian distribution, and the probability expressed

$$p(x \mid i) = \frac{1}{(2\pi)^{-\ell/2}\left|\Sigma_i\right|^{1/2}} e^{-\frac{1}{2}\left[(x-\mu_i)\Sigma_i^{-1}(x-\mu)\right]} \qquad 2.7$$

where μi and Σi are mean and covariance for each class, ℓ is the number of spectral bands, and |Σi| is the determinant of the covariance matrix. Respectively, the combined probabilities for the image can be expressed

$$p(x) = \sum_{i=1}^{k} p(i)\left[\frac{1}{(2\pi)^{-\ell/2}\left|\Sigma_i\right|^{1/2}}\, e^{-\frac{1}{2}\left[(x-\mu_i)\Sigma_i^{-1}(x-\mu)\right]}\right] \qquad 2.8$$

where k is the number of classes and p(i) is the probability of each class existing. The distributions in p(x) exist in the data, allowing for estimation of the statistics on the right hand side of equation 3.11 [Schott, 2004]. The algorithm relies on *a posteriori* estimates of class statistics, and must therefore be initialized with some values for mean and covariance. This can be done by randomly assigning pixels to each class, or by using the results of a previous classification. Once pixels have been assigned to classes, the mean and covariance are calculated by

$$\mu_m^{n+1} = \frac{1}{N_m^n}\sum_{i=1}^{N_m^n} x_{i,m}^n \qquad 2.9$$

$$\Sigma_m^{n+1} = \frac{1}{N_m^n}\sum_{i=1}^{N_m^n} (x_{i,m}^n - \mu_m^n)(x_{i,m}^n - \mu_m^n)^T \qquad 2.10$$

where n is the current iteration and N_m^n is the number of pixels assigned to class m in iteration n. The probability of class m existing in the next iteration can also be calculated by

$$p_m^{n+1}(i) = \frac{N_m^n}{N} \qquad 2.11$$

where the probability must exceed the user defined threshold for minimum class size. The conditional probability p(x|i) in the next iteration can then be estimated from the results of equations 2.9, 2.10, and 2.11 by the normal distribution in equation 2.7. The result of the process is an estimated *a posteriori* probability of each class, which can be expressed

$$p_m^{n+1}(i \mid x) = \frac{p_m^{n+1}(i)\; p_m^{n+1}(x \mid i)}{\sum_{q=1}^{M} p_q^{n+1}(i)\; p_q^{n+1}(x \mid i)} \qquad 2.12$$

where M is the user defined maximum number of classes [Masson and Pieczynski, 1993]. In practice, to substitute for the product p(i)p(x|i), a discriminant function can be derived using the natural log to eliminate the exponential in the expression. This discriminant function takes the form

$$Di = \ln(p(i)) - \frac{\ell}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i) \qquad 2.13$$

which preserves the rank order of the probabilities [Schott, 1997]. Pixels are reclassified based on these probabilities and the new mean, covariance, and p(i) values are calculated for each class. These values then seed the next iteration and the process continues until a convergence criterion has been satisfied. For determining convergence, a simple method such as thresholding the change in class means is a viable option.

Incorporating second order statistics into the classification process allows the algorithm to consider spectral shape of a class, which leads to more accurate classification. The method will, in turn, provide better class statistics within the classified image.

2.2 Matched Filters

There are a growing number of spectral matched filters which perform the task of locating target spectra within a data set. The fundamental principle behind the matched filter remains constant with each new addition. The spectral matched filter operates in much the same way as the spatial matched filter, both using a target function to search for the location of a similar function. The ideal spatial matched filter returns a Dirac delta, or impulse response, at the location of the function, as expressed by

$$f\,[x-x_o, y-y_o] * m\,[x,y] = \partial\,[x-x_o, y-y_o] \qquad 2.14$$

where

$$m\,[x,y] = \Im^{-1}\left\{\frac{1}{F\,[\xi,\eta]}\right\} \qquad 2.15$$

where * denotes a convolution, $\Im^{-1}$ is the inverse Fourier transform operator, f[x] is the function, F[ξ] is the Fourier transform of the function, m[x] is the matched filter, and x_o and y_o are the offsets or translations [Gaskill, 1978]. For multispectral or hyperspectral data, the spectral matched filter follows the same concept, testing each pixel location for similarity to a spectral function. In practical application, this is performed as a matrix multiplication of target spectrum with each pixel, where the larger the magnitude of the result the more similar the pixel is to the target. For real world HSI applications, the results of the quantized ideal matched filter are not nearly distinct enough to easily threshold. This problem drives the need to design matched filters which increase the disparity between target and non-target returns.

2.2.1 Spectral Angle Mapper

The spectral angle mapper (SAM) is a quick and easy method often used to test pixel spectral similarities without any characterization of background. It is based on a simple vector projection, and the result is the cosine of the angle between a reference vector and a pixel vector. The operator can be expressed as a scalar product

$$\cos(\theta) = \left(\frac{d \bullet x}{|d||x|} \right) \qquad 2.16$$

where d is the reference spectrum, x is a pixel, and θ is the angle between them. This metric gives a measure of similarity between the vectors, although it considers only relative spectral angle and not the relative vector magnitudes. The SAM result can also be performed by matrix multiplication, which can be expressed

$$T_{SAM} = \frac{d^T x}{(d^T d)^{1/2} (x^T x)^{1/2}} \qquad 2.17$$

where we will adopt T as the standard form of a detection statistic or result. In this case, cos(θ) will always be positive with a range from zero to one because all spectra vectors have positive components [Manolakis et al, 2001]. In this form, SAM clearly resembles an ideal matched filter. SAM is commonly used in the early stage of processing algorithms because it is computationally inexpensive and does not require any statistical information about the scene.

2.2.2 Structured and Unstructured Backgrounds

In order to take the next step and include a model for background in the matched filter, it is important to recognize two distinct categories of background characterization. Structured backgrounds, like the kind used in the orthogonal subspace projection (OSP) algorithm [Harsanyi and Chang, 1994], are concatenations of background endmember spectra. The OSP algorithm projects pixels onto a subspace orthogonal to the selected structured background endmembers or basis vectors. This method involves defining the background endmembers that exist within a scene by use of an endmember selection routine like singular value decomposition (SVD) [Healey and Slater, 1999], MaxD [Lee, 2003], PPI [Boardman et al, 1995], or N-FINDR [Winter, 1999]. This category is often referred to as the geometric approach to background characterization.

In contrast, the stochastic approach derives an unstructured background from the image data. In the matched filter equation, this takes the form of a statistical mean and a covariance or correlation matrix, where the mean, covariance matrix, and correlation matrix may respectfully be expressed

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x(n) \tag{2.18}$$

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N} \left(x(n)-\mu\right)\left(x(n)-\mu\right)^{T} \tag{2.19}$$

$$R = \frac{1}{N}\sum_{n=1}^{N} \left(x(n)\right)\left(x(n)\right)^{T} \tag{2.20}$$

This concept for representation of background is the basis of several variations of clutter matched filters. The correlation matrix is often applied to mean subtracted data to avoid the unnecessary subtraction in calculation of the covariance matrix. Using the correlation matrix in place of the covariance matrix will change the raw results. A more thorough comparison of the two will be discussed in a later section.

When applying first and second order statistics to the target detection problem, it is important to note the underlying assumptions. First, the background is homogeneous and exhibits multivariate Gaussian behavior. Second, the background spectrum interfering with the target signature has the same covariance as the background training pixels. Finally, the spectra of the target and background must combine in an additive manner [Manolakis and Shaw, 2002]. These assumptions must be made because of the decision theory upon which adaptive matched filters are based, which will be covered in the next section.

2.2.3 Adaptive Subspace Matched Filters

The Neyman-Pearson theorem establishes a binary hypothesis for the presence or absence of target material in a pixel. Each hypothesis is assumed to represent a normal probability density function (PDF). This can be represented by

$$\begin{aligned} &H_0 : x = U\gamma + n, \text{ target is absent} \therefore PDF = P(\mu, \Sigma_b) \\ &H_1 : x = U\gamma + d\alpha + n, \text{ target is present } \therefore PDF = P(\mu, \Sigma_t) \end{aligned} \qquad 2.21$$

where Σ_b is the background covariance, and Σ_t is the covariance with target present. These hypotheses can be compared by forming a likelihood ratio

$$\ell\,(x) = \frac{H_0}{H_1} = \frac{P(\mu;\Sigma_b)}{P(\mu;\Sigma_t)} > \tau \qquad 2.22$$

where τ is a threshold which will determine the probability of a false alarm (P_{fa}). Any detector of this type, which has an adjustable threshold allowing the user to set P_{fa}, can be called a constant false alarm rate (CFAR) detector. According to the underlying assumptions, the PDF associated with each hypothesis has a normal distribution, and can therefore be represented by

$$P(\mu;\Sigma) = \frac{1}{(2\pi)^{L/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \qquad 2.23$$

where $|\Sigma|$ is the determinant of the covariance matrix. While in equation 2.23 the distribution follows the traditional Gaussian PDF, it is important to note that HSI data are not generally normal, especially for scenes with many material types. Therefore, Neymen-Pearson detectors should be expected to lose accuracy as the PDF of the HSI data strays from Gaussian distributions. A solution to the over reaching normality assumption is currently being pursued by several research groups in the field [Stein et al, 2002], [Manolakis et al, 2002]. In order to establish the likelihood ratio in equation 2.22, the data mean and covariance, which are not generally known, must be estimated from the data. This requires an adaptive detector, which is usually developed as a generalized likelihood ratio test (GLRT). Substituting equation 2.23 into equation 2.22 and computing the natural logarithm, we arrive at the quadratic detector

$$T = \frac{1}{2}(x-\mu_b)^T \Sigma_b^{-1}(x-\mu_b) - \frac{1}{2}(x-\mu_t)^T \Sigma_t^{-1}(x-\mu_t) \qquad 2.24$$

which is the comparison of the Mahalanobis distance between the pixel spectrum x and the mean of the background and target distributions. While the covariance matrix of the background can be estimated from scene data, the target covariance matrix generally cannot be obtained. If the covariance matrices of the background and target are assumed to be equal, an adaptive matched filter that suppresses clutter based on background statistics can be derived either by maximizing the cost function [Manolakis and Shaw, 2002], or by minimization of the total energy of the filter output by the constrained energy minimization (CEM) technique [Ferrand and Harsanyi, 1997]. The result of either is a matched filter based on the target to pixel Mahalanobis distance, using a background covariance, scaled by the matched filter return of the target with itself (maximum possible signal to background ratio (SBR)). This matched filter, which we will call CEM, can be expressed

$$T_{CEM} = \frac{d^T \Sigma^{-1} x}{d^T \Sigma^{-1} d} \qquad 2.25$$

where the detection statistic T_{CEM} can be related to abundance, d again is the target spectrum, and $\Sigma = \Sigma_b$ to simplify the nomenclature. CEM may be seen as the unstructured equivalent to the scaled structured-background OSP operator, replacing the projection operator formed through concatenation of endmember spectra with the covariance matrix estimated from the scene background. The equations for these algorithms, which characterize background, share the same basic form as the simple matched filter in equation 2.23, but include an operator which compensates for spectral variation in scene background, thereby adapting to differing scene content.

Another type of detector derived by Kelly [1989], which uses the GLRT but does not employ an optimality test, can be expressed

$$T_{GLRT} = \frac{(d^T \Sigma^{-1} x)^2}{(d^T \Sigma^{-1} d)(1 + x^T \Sigma^{-1} x)} \qquad 2.26$$

This detector can be developed further to form a family of detectors including the adaptive coherence estimator (ACE)

$$T_{ACE} = \frac{(d^T \Sigma^{-1} x)^2}{(d^T \Sigma^{-1} d)(x^T \Sigma^{-1} x)} \qquad 2.27$$

and the so-called adaptive matched filter (AMF)

$$T_{AMF} = \frac{(d^T \Sigma^{-1} x)^2}{(d^T \Sigma^{-1} d)} \qquad 2.28$$

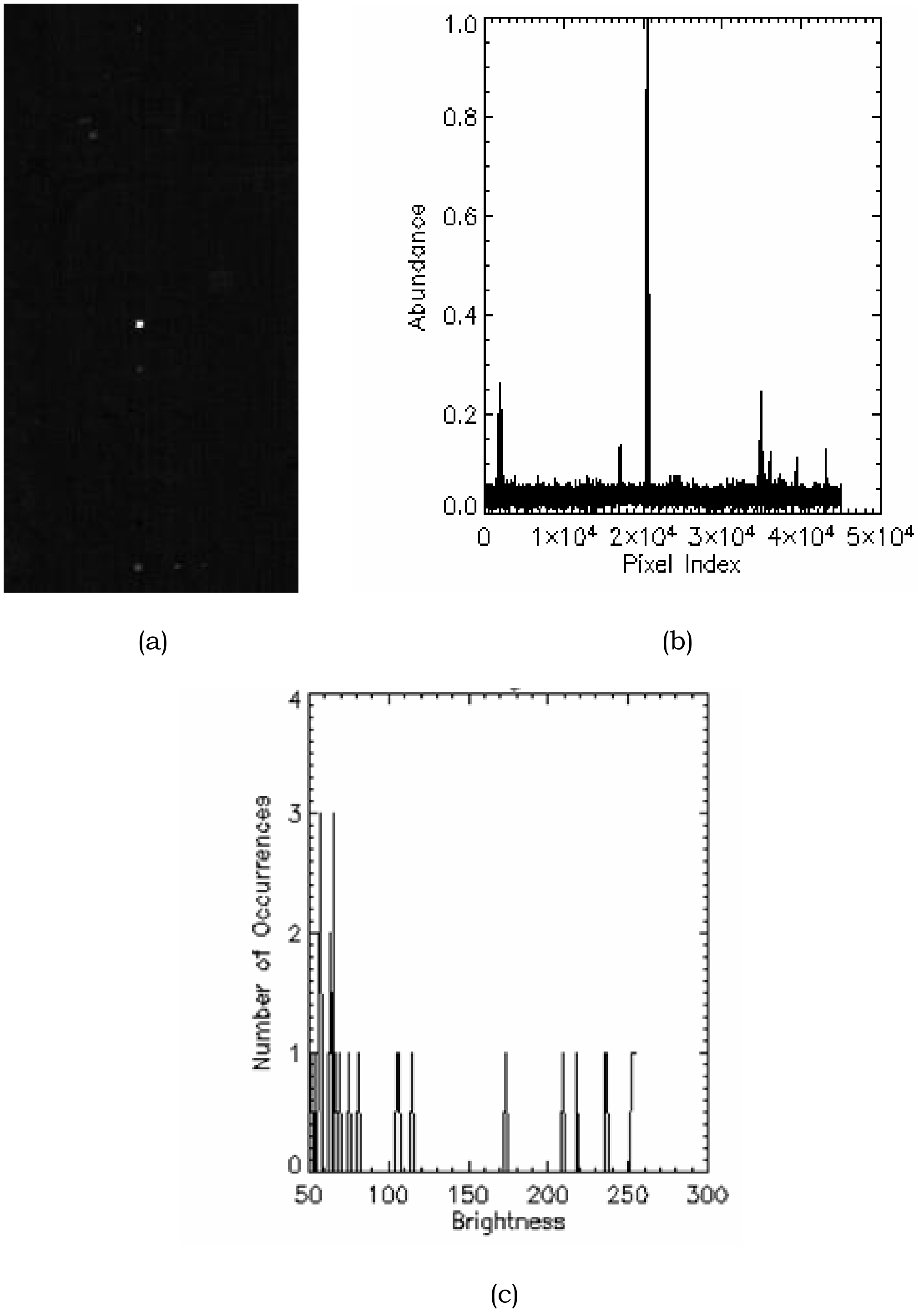
which are all different ways to stretch the detection statistic and achieve greater target to non-target separation [Manolakis et al, 2001]. While this family of improved detectors is valuable in practical application for more easily setting a CFAR threshold, the method used to calculate the background covariance may have a greater impact on overall detector performance. As previously noted, for all adaptive matched filters, information about the target and background statistics are not available and must be provided, or estimated from the given data set. The target spectrum is obtained from library data or measured ground truth, and background statistics are commonly derived using all available observations (pixels). Practical methods for estimating these statistics based on different characterizations of scene background will be covered in a later section.

2.3 Evaluation Metrics

There are many ways to visualize and quantify target detection results. The detection statistic may be examined in raw form, in an image, or plotted in order to help an analyst visualize the results. For the detection problem, the relationship between P_{fa} and P_d may also be tabulated to assess detector performance. To fully characterize this relationship, a receiver operator characteristic (ROC) curve can be plotted, giving a graphical representation of detector performance. When several ROC curves need to be compared, another metric may be developed to characterize ROC performance with a single numerical value. All of these techniques warrant consideration when evaluating the performance of an algorithm based upon detection results.

2.3.1 Detection Statistic Examination

The most obvious method for visualization is simply an image of the detection statistic. As the detector operates on each pixel, an image similar to an abundance map is populated with the returned values of T. The brighter the pixel appears on the map, the greater the probability of target presence (or the greater the target abundance) within the pixel. This image may also be plotted, pixel brightness versus pixel index, to allow for easier visual gray scale comparison. Another way to visualize raw detector output is the use of a histogram, which allows the user to observe the number of occurrences of each pixel value. An example detection statistic image, brightness plot, and the corresponding histogram (for target pixels only) are shown in Figure 2.3.



(a) (b)

(c)

Figure 2.3 Example Target Detection Results Examined by a Detection Statistic
(a) Image, (b) Plot, and (c) Histogram (Target Pixels Only).

2.3.2 Receiver Operator Characteristic Curves

When ground truth is available for a given scene -- that is, all target pixel locations are known -- the most common method for evaluation of detection algorithms is a comparison of the probability of detection (P_d) and the probability of false alarm (P_{fa}). Short of compiling the data necessary to form a ROC curve, detector performance at different thresholds (τ) can be characterized by varying τ to tabulate three discrete results for P_d and P_{fa}. First is the value of P_{fa} when P_d is unity, second is the value of P_d when P_{fa} is zero, and third is the value of P_d when P_{fa} is set at some small acceptable value. If this is repeated for a large number of thresholds, a full ROC curve can be formed. An example of a comparison of several ROC curves is shown in Figure 2.4.

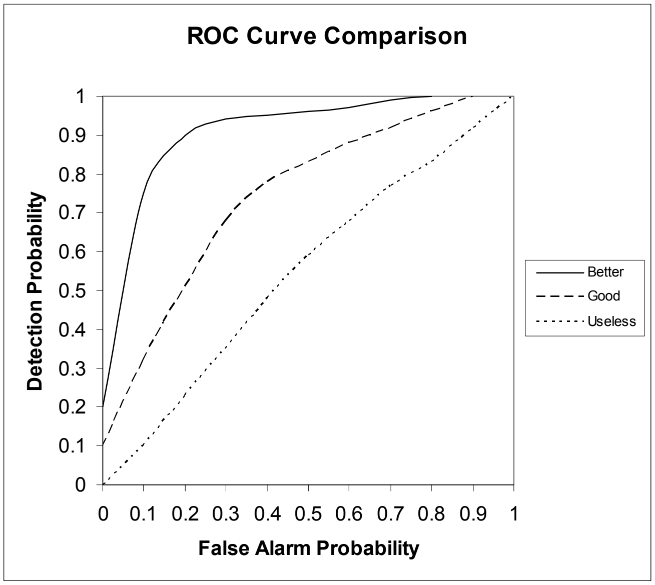


Figure 2.4 Example ROC Curves for Detector Performance Evaluation

The ideal ROC result would have a steep slope along the y-axis and reach a maximum detection rate at very low rates of false alarm. The three curves in

Figure 2.4 represent varying detector performances. It is important to note that each value for P_d in a ROC curve has a confidence interval associated with it, based on the relative number of targets in the scene. This should be considered to ensure that improvements in P_d are meaningful in images with few targets. For this reason, the terms detection rate and false alarm rate will generally substitute for the probability terms.

For target detection experiments involving several or even hundreds or thousands of ROC curves, a summary metric is needed to characterize ROC performance. One such summary technique uses an estimate of the area under the ROC curve to capture detector performance in a metric called the average false alarm rate (AFAR) [Bajorski et al, 2004]. AFAR is simply the average of all of the false alarm rates measured at the detection of each target. To capture detection performance at only high detection rates, a partial AFAR can be calculated by averaging the false alarm rates up to a given level of detection.

No single metric can perfectly capture the performance of a detector. Study of detection and false alarm rates can be helpful in performance comparisons, but raw results remain the most descriptive and likely the most involved method of analyzing detection results.

2.3.3 Multivariate Normality Tests

The assumption that a given data set has been drawn from a normal distribution is central to statistical processing techniques across many disciplines. As a result of the increased use of multivariate data over the past fifty years, many methods to test a data set for univariate normality (UVN) have been extended and new tests developed for the multivariate normality (MVN) case. The Chi-Squared, Kolmogorov-Smirnov, Anderson-Darling, Cramér-von Mises, Shapiro-Wilk, Mardia, and Henze-Zirkler tests are among those used in fields like communications, business, medicine, and imaging to test how well a univariate set of data can be modeled by a Gaussian distribution [Mecklin and Mendfrom, 2004]. MVN tests can generally be separated into four categories: graphical examination with correlation coefficients, tests for goodness-of-fit, skewness and kurtosis tests, and consistent procedures based on the empirical characteristic function [ibid]. The fact that more than fifty such tests are documented in the literature alone is an indication that the MVN problem is complex and not well refined.

To select the most appropriate test for HSI backgrounds, it is important to consider the dimensionality and number of observations in the data. For instance, tests such as Royston's H test expand the Shapiro-Wilk W test from a maximum of 50 to a maximum of 2000 bands [Royston, 1982]. Extensions of the Kolmogorov-Smirnov (KS) test are also meant for large amounts of data and therefore may be more appropriate for HSI MVN testing. Another popular test is Mardia's PK test, which employs the third and fourth statistical moments, skew and kurtosis, to respectively characterize asymmetry and the heaviness of the "tails" of the data. Data sets with negative kurtosis are considered "light tailed" and those with positive kurtosis are "heavy tailed." While third and fourth order statistics can provide an indication of normality, the single metric result of these tests has been shown not to provide the most consistent measure of MVN [Mecklin and Mendfrom, 2004].

A simple yet computationally exhaustive way to judge MVN is to examine the normality of each variable in linear combination with every other variable.

When dealing with HSI data, the great numbers of band combinations, $\sum_{i=2}^{n} n^i$, makes this approach practically infeasible. A true measure of MVN would require a characterization of every combination. In an attempt to simplify the problem, it is tempting to consider the normality of each band independently. The chi-squared MVN test can be easily employed to do this, but it is important to fully understand how the test is performed and what the results indicate.

The chi-squared method exploits the fact that the distribution of squared error (statistical distance from the mean) in a data set that is sampled from a normal distribution will behave like a chi-squared random variable [Johnson and Wichern, 2002]. The familiar squared statistical (Mahalanobis) distance is measured by

$$d^2 = (x-\mu)^T \Sigma^{-1} (x-\mu) \qquad 2.29$$

and can be compared to a chi-squared distribution given

$$P_{\chi^2}(x) = \frac{x^{p/2-1} e^{-x/2}}{\Gamma(p/2)\, 2^{p/2}} \qquad 2.30$$

where x is the independent variable of the distribution (not the pixel vector), p is the degrees of freedom (number of bands) and Γ is the gamma function. Similar to the quality measure given by a Q-Q plot (except in this case the two quantiles are the distance metric and the chi-squared probability), the two metrics will exhibit a linear relationship with unity slope and zero bias, where divergence from linearity indicates a deviation from normality. In order to construct this plot, the Mahalanobis distances are ordered smallest to largest and plotted versus the upper percentiles of the chi-squared distribution given by

$$\chi^2((n-j+1/2)/n) \qquad 2.31$$

where n is total number of bands and j is the number corresponding to the ordered statistical distance [ibid]. A notional example of a chi-squared plot is shown in Figure 2.5, where the data seem to follow the dashed line well, indicating normality.

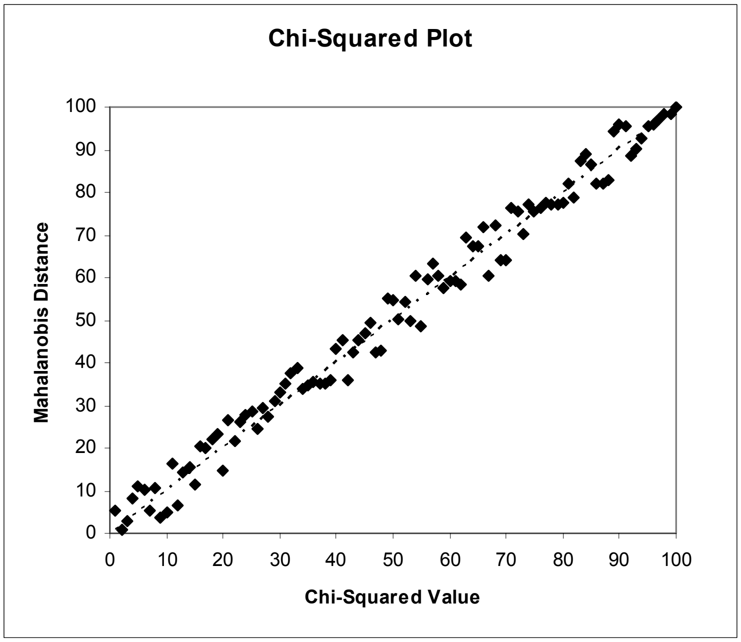


Figure 2.5 Example Chi-Squared Plot for MVN Test

The chi-squared plot is a valuable visual tool for identifying gross deviations from normality and outliers in the data. Without comparing any of the bands in the data, however, it is misleading to attempt to define a metric of MVN based on a plot of within-band UVN. This being said, applying a single metric characterizing the MVN of several data sets for the purpose of comparison may be valid.

Quantifying the "straightness" of the chi-squared line or the goodness-of-fit (GoF) of the data to a normal distribution is often accomplished using a correlation coefficient test. This test reduces the chi-squared curve to a single metric by relating the expected value (χ^2) to the value in the distribution (d^2) by the equation

$$\chi^2_{GoF} = \frac{1}{N}\sum_{i=1}^{N}\frac{(d_i^2 - \chi_i^2)^2}{\chi_i^2} \qquad 2.32$$

where a lower value indicates greater multivariate normality [Snedecor, 1989].

2.4 Background Characterization

Characterization of scene background involves a decision about how data will be collected that best represents the "true" background from which the target must be distinguished. This data is then used in the calculation of background statistics. When implementing an adaptive matched filter, image-wide covariance is the most common method for background characterization. Estimating the background based on all of the variability in the scene is a good starting point, but is overall a poor method of background characterization. Improvements on the scene-wide method have been shown to increase detection performance dramatically [Reed and Yu, 1990], [Ashton, 1998], [Manolakis et al, 2001], [Funk et al, 2002]. Two types of techniques for improved background statistics estimation are easy to identify in the literature: target exclusion and data subsetting. Target exclusion methods assume that background error is due to the presence of target pixels in the data from which background covariance is calculated. Data subsetting methods assume that error in scene-wide covariance is the result of a mismatch between the background and the source of interference as well as a failure of the Gaussian approximation for the scene-wide data. The scene-wide data can be divided into subsets by drawing either a spatial or a spectral distinction between pixels. By drawing these distinctions, pixels from the scene are selected to be included or excluded from the background statistics data pool. Subsets are generally more MVN than the full scene, and may be selected to match the interfering signal. The difference between the spatial and spectral subsetting techniques is the type of interfering signal they are attempting to match. Techniques which characterize background using a spatial subset of the data surrounding a test pixel assume that interference is caused by the subpixel mixing of a target with its surroundings. Techniques which characterize background by spectrally subsetting the data assume that interference is caused by target-like pixels in the scene. How these assumptions drive the application of detectors using these background characterization techniques is a central question of this study and will be discussed in a later section.

2.4.1 Spatial Subsetting

One simple method to improve over scene-wide statistics is to select a spatial subset of the image by hand to serve as the background. Selecting a part of the image that was collected before reaching the target area ensures that targets are not present in the background and that the species present in the background are representative of those around the target. Such a spatial subset of the image would likely be more multivariate normal than the full scene, providing better data for the background statistics of the matched filter. For test data sets, this method is employed by selecting a region of the image known not to contain target with the assumption that a target approach region would be available in real-world operational imagery. This target approach method has been shown to provide improved detection results [Manolakis et al, 2001] over full scene statistics, and serves as a good baseline for background characterization.

The RX algorithm, named for its developers Irving Reed and Xiaoli Yu, can be used for both anomaly detection and target detection [Reed and Yu, 1990]. It is a combination of the spectral and spatial matched filters, using an anomaly detection formulation of the GLRT and a convolution kernel set to match the assumed shape of a fully resolved multi-pixel target. The anomaly detection form of RX does not rely on a known target spectrum d, and commonly takes the form

$$RX = (x - \mu)^T \Sigma^{-1} (x - \mu) \qquad 2.33$$

During development of RX, violation of the normality assumption was a primary consideration in the formulation of the problem. To improve the detection process, a sliding window was used to spatially subset the data. The underlying assumption is that the background can be forced into Gaussian behavior by assuming a non-stationary mean [ibid]. This is somewhat intuitive considering that intra-class variation is more normal than inter-class variation and a spatial subset of the scene is likely to contain fewer classes. It has also been demonstrated that this method of local mean subtraction intrinsically increases the MVN of the data [Li et al, 2003]. According to the construction of the RX algorithm, the mean of the background can be calculated within a windowed subset of the data and the more slowly varying covariance can be calculated from a larger, target-free subset and approximated by a diagonal matrix. The third moment (approaching zero) was used to approximate the normality of the spatial subset. Minimizing skew in this way was assumed to "create a distribution which is as close to Gaussian as possible" [ibid]. The spatial matched filter portion of the algorithm is not applicable to either the sub pixel problem or the characterization of background, and therefore is not a consideration for this study.

2.4.2 Spectral Subsetting

Many new target detection algorithms have sought to exploit the methods of HSI classification to gain advantage in the matched filter detection problem [Funk et al, 2002], [Ashton, 1998], [Stein et al, 2001]. The premise for improved detection results through pre-clustering (spectrally subsetting) is achieving greater target contrast by comparing the target only to pixels in a single class. While at first this may seem as though it would confound the problem, it is easy to determine that while a low contrast, low abundance target may be lost in a cluttered scene and indistinguishable using scene-wide statistics, any spectral deviation from a single class will be easy to identify and accentuate using class statistics. The expectation is that subsetting techniques may provide a more normal background, populating the covariance matrix with data better fitting the Gaussian assumption, and thus normalizing the target and pixel spectra with spectral variances that better differentiate between target and background species. The AMF can be adapted to incorporate results from a k-means clustering by

$$T_{CEM}^{k} = \frac{d^T \Sigma_k x}{d^T \Sigma_k d} \tag{2.34}$$

where k is the class number, and values are mean subtracted using class means. This method was shown to increase the performance of several forms of the adaptive matched filter when applied to HSI for plume detection [Funk et al, 2001].

A full discussion of plume detection phenomenology is outside the scope of this work, but the significant differences in the type of mixtures involved in plume detection versus hard target detection reveals an important difference in the application of pre-clustering. The best background to use in plume detection can be derived from the ground class over which the plume passes. For large plumes, different regions of gas may pass over (and mix with) different ground classes. To detect the plume over any ground class, the Funk k-means pre-clustering method applied each cluster individually as a background, and then fused the results to form one detection statistic image. Unlike effluent species, the best background for hard targets is likely to be represented by a single class. However this class may be selected, it is used to model and suppress false alarms in the scene or sub pixel mixing from a single class. Given the differences in the two detection problems, the plume detection results using k-means pre-clustering do show proof of concept. Along with these results, it was noted that improvement could be made over the k-means method by use of a classification algorithm that exploits both mean and covariance [ibid].

The k-means pre-clustering technique has also been used to improve anomaly detection results in multispectral IR imagery [Ashton, 1998]. In the same study, an adaptive Bayesian clustering algorithm was also applied in an attempt to improve on the k-means results. The Bayesian clustering algorithm used a probabilistic model given by

$$P(k \mid x) = P(x \mid k)\, P(k) \qquad 2.35$$

where the statistical pixel distribution for P(x|k) was given by a multivariate normal distribution and the probabilistic region distribution, P(k), was generated with a Gibbs random field. The k-means results were used to seed the algorithm, and several other techniques were applied to reduce the computation time, which can be prohibitive for this type of stochastic relaxation classifier. This study used pre-clustering to search for hard targets, necessitating a method of background class selection. To arrive at a single set of statistics for the detector, the classification result for each test pixel was used along with a grid method to interpolate between statistics of grid points surrounding each test pixel. Anomaly detection was performed using the pre-clustered data by identifying the pixels least likely to fall into the background. The ROC results are given in Figure 2.7, showing an improvement in anomaly detection over a standard RX result through spectral clustering (k-means), and an even greater improvement through Bayesian clustering.

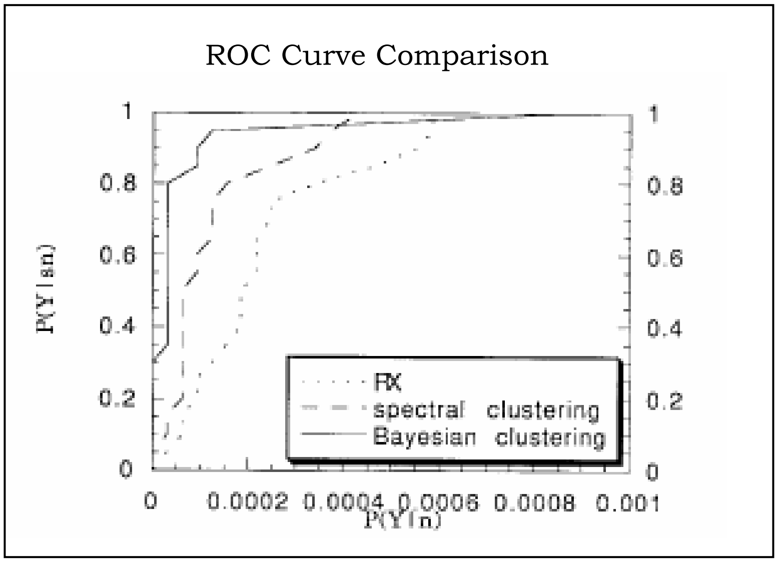


Figure 2.7 ROC Curve Comparison of RX, K-Means, and Bayesian Pre-Clustering Applied to Anomaly Detection [Ashton, 1998]

The SEM algorithm, which uses mean and covariance to classify HSI data, has also been used for pre-clustering prior to applying a matched filter. The RX anomaly detector was adapted to operate on data pre-clustered into k SEM classes, taking the form

$$RX_k = (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \qquad 2.36$$

where μ_k and Σ_k are the mean and covariance of each class [Stein et al, 2001]. The class statistics used for each test pixel were derived from the class to which the pixel was assigned. Therefore, just as the traditional RX algorithm identifies pixels which are dissimilar to their surroundings, this pre-clustering implementation of RX identifies pixels which are dissimilar to their SEM class. In the results of this study, improvement over scene-wide statistics was not explicitly stated because the technique was used as part of a fusion of RX and SEM detectors.

With the ability to achieve improved detection results through both spatial and spectral subsetting well established, the advantages of the two algorithms was simultaneously exploited through fusion. Fusion rules, including model selection, AND and OR, and joint-density, were employed to create decision boundaries encompassing thresholds for each algorithm and for a normality metric. At thresholds set for low levels of false alarms (10-100 per km^2), this technique reduced false alarms by .25 to 2 orders of magnitude [ibid]. Figure 2.8 is an example ROC curve from the fusion results, showing that fusion provided better detection performance, especially at lower false alarm rates.

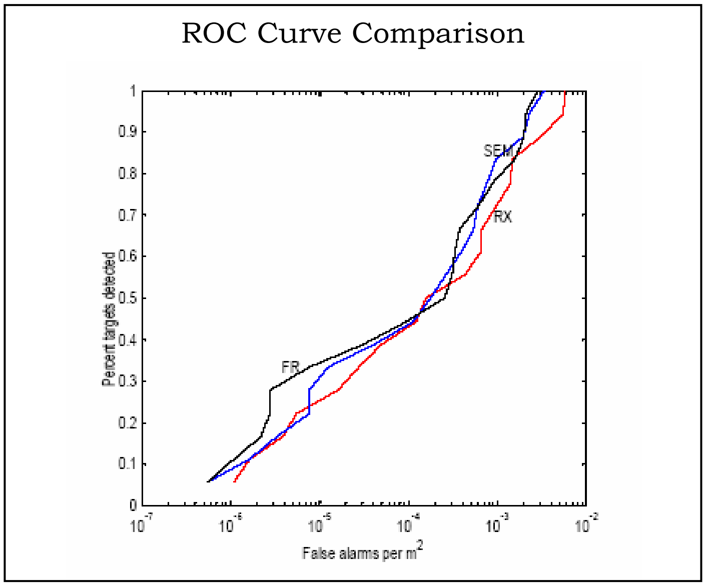


Figure 2.8 ROC Comparison of, RX, SEM, and Fusion (FR) Algorithms [Stein et al, 2001]

2.4.3 Target Exclusion

Another method for improving estimation of background covariance is the exclusion of target pixels from the background data. If target pixels are included in an estimation of background variability, the characteristic spectral features of the target may be suppressed along with the background and the matched filter results will be useless. One common target exclusion method involves the manual selection of a region of the data where target is known not to be present (i.e. the target approach region) [Manolakis et al, 2001]. While this method ensures that the target will not affect the background covariance estimation, it is largely dependent on the user's selection of a region that best characterizes background. An automated form of target exclusion involves the use of a non statistical detector to detect target-like pixels, which can then be excluded from the calculation [Manolakis et al, 2000]. Both of these methods will attempt to ensure target spectra are not included in the background and will therefore not adversely affect the covariance estimation.

As described in previous sections, techniques involving spatially or spectrally subsetting the scene data can also be employed to better estimate background covariance. These methods have been shown to provide improved detection results [Reed and Yu, 1990], [Manolakis et al, 2000]. However, target contamination problems arise when applying techniques that restrict the amount of data used to estimate the matched filter covariance. First, in a smaller data pool the target spectra will have greater statistical influence on the covariance, thus increasing the importance of employing a target exclusion method. Second, the number of data points used to calculate the covariance may not be sufficient to generate a statistically stable covariance. If the number of pixels in a subset is less than the number of bands, the covariance matrix will be singular and cannot be inverted, and if the number of pixels is only a few times the number of bands, the matrix is usually highly elliptical and errors resulting from its use can be significant [Kuo and Landgrebe, 2002].

2.4.4 Adaptive Covariance Estimation

The concept of adaptive covariance estimation was originally developed to generate class statistics from small groups of training samples for supervised classification applications. In a supervised classification, the user may not specify enough pixels to enable calculation of a stable, non-singular covariance. Consequentially, a variety of techniques were established to estimate the covariance of small training groups in order to seed quadratic classifiers with a more accurate covariance. The same concept may be applied to the calculation of covariance from a small data subset for the purpose of seeding a quadratic target detector with a more accurate characterization of background.

The performance of a given covariance estimator is largely dependent on the specific scene content. In some cases, where the covariance of all classes is very similar, the scene-wide covariance may be the most accurate. In other cases, mixtures of covariance from several classes may best represent truth. The best method or mixture to use for a given data set will depend on the true statistics of the classes, the number of features, and the number of samples in the subset [Hoffbeck and Landgrebe, 1996]. These parameters, and their influence on detector output, cannot be known for each case. This is what drives the need for an adaptive covariance estimator, which might be seen as a way to compromise between scene-wide and subset statistics. Regularization is the process of reducing the amount of estimation made during covariance calculation by assuming a mixture of statistics. A regularization (or mixing) parameter α_i can be used to scale the relative amounts of each covariance by

$$\hat{\Sigma}_i = (1-\alpha_i)\Sigma_i + \alpha_i \Sigma_p \qquad 2.37$$

where $\hat{\Sigma}_i$ is the improved estimate of class i covariance, Σ_i is the covariance of class i from the samples, and Σ_p is the average covariance of all classes called the pooled covariance [Tadjudin and Landgrebe, 1999]. One might note that this mixture of covariance shares the form of the SMM for endmember mixtures.

These types of covariance estimators may apply to the target detection problem in scenes where targets will contrast with pure endmembers but not with mixtures of those endmembers. In other words, if the target is spectrally distinct from dirt and from grass, but spectrally similar to dirty grass, a spectral subsetting of the data that captures several levels of dirty grass will be helpful in distinguishing the target. This type of analysis was applied to develop a new type of detector that includes target statistics, called the Finite Target Matched Filter (FTMF) [Stocker and Schaum, 1997]. In this study, the SMM was used to characterize mixtures of classes in order to better discern low contrast targets. Figure 2.9 is a scatter plot showing several mixed classes surrounded by ovals and a set of target pixels that can be detected due to the improved analysis of background statistics. This type of detector, which involves target statistics, is beyond the scope of this research, but the techniques used to characterize the background using detailed analysis of class mixtures is directly applicable.

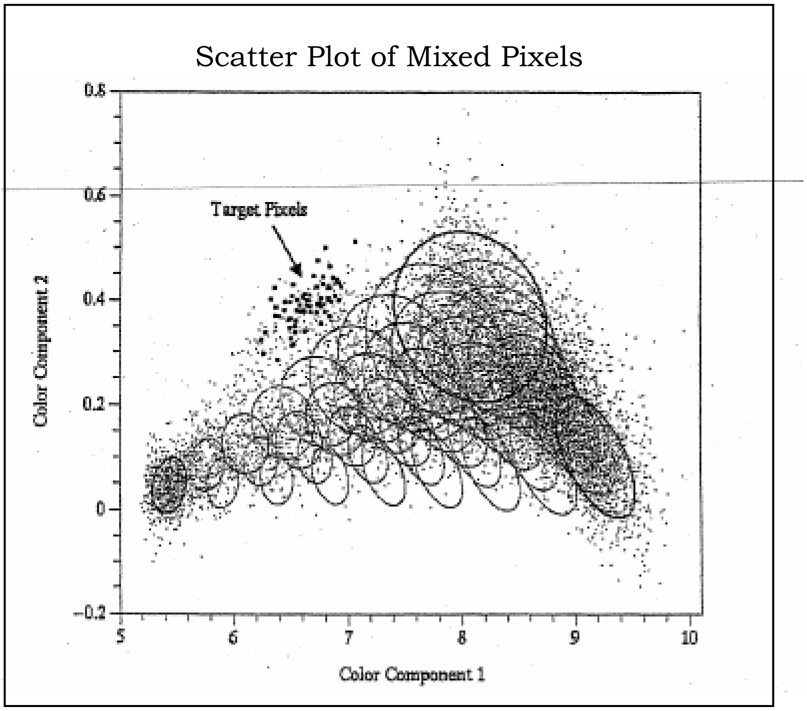


Figure 2.9 Low Contrast Target in Mixed Pixel Scatter Plot [Stocker and Schaum, 1997]

2.5 Background and Literature Review Summary

Literature in the field of hyperspectral image processing abounds with techniques that may be used to preprocess target detection data, as well as techniques that may be employed to improve target detection results. Atmospheric compensation and dimensionality reduction may be used to seed detectors with smaller, more manageable data sets in units that are convenient for matched filtering. Several types of detectors have been developed in the field, many of which use the stochastic approach for modeling scene background. These detectors can be compared using a number of evaluation metrics, the most popular of which is the ROC curve. The statistical structure of the background data and the detection results also can be evaluated to provide insight into the extent that the underlying assumptions of the process are violated. A collection of techniques to perform preprocessing, matched filter detection, and evaluation of results forms the foundation for a study of background characterization methods.

In an attempt to improve detection results beyond the formulation of new types of matched filters, the research community has developed several methods for better characterizing scene background. That is to say, an attempt has been made to more carefully select the data that is to be called background, and improve the method for estimating the statistics of the background from that data. Excluding targets and subsetting the data either spatially or spectrally are the types of methods that have been tested and reported in the literature. These background characterization methods are interleaved with techniques in image classification, background independent target detection, anomaly detection, and adaptive covariance estimation. Each of these related tools have more than one algorithm by which they can be employed, and may be used alone or in combination when applied to target detection.

The next section will address how these many techniques have been organized to form an experiment testing the current state of background characterization. Along with current methods, a few new approaches will be described which combine concepts from several existing algorithms. Beginning with the selection of targets and data upon which to operate, and continuing with a description of the implementation of methods described in the literature, an approach to the experiment will be framed in a tractable set of steps leading to conclusions about the practical application of background characterization for matched filter target detection.

3.0 Experimental Approach

The algorithms and processing steps required to perform a comparative experiment of background characterization methods fell into four progressive categories: selection and preprocessing of the test data, matched filter implementation, the implementation of each method of background characterization, and the implementation of evaluation tools to analyze background data and detection results. Existing tools were used exclusively for the preprocessing of data, while new tools were created for all of the techniques in the remaining categories. Some parts of the experiment required customization of existing algorithms, while other parts necessitated the development and implementation of new algorithms. The following sections will detail the rationale behind the decisions made in the selection and implementation of the algorithms used in the experiment. As much as possible, the trade-spaces for these decisions will be discussed to allow future work to adapt these implementations to further study the background characterization and target detection problems.

3.1 Experimental Data

Evaluation of target detection algorithms may be accomplished by two methods. First, a general test for detection power may be derived for a universal application of the detector. Second, an empirical test may be performed on a specific data set and generalizations assumed based on the outcome. Without using real data, the theoretical detection capability of a detector can be demonstrated and comparisons of techniques have been made by making some assumptions which simplify the detection problem [Manolakis and Shaw, 2002], [Bajorski, 2005]. However, given the lack of a realistic data-independent test for target detection capability, the empirical method becomes the only option. Without a fully general test, the selection of an appropriate HSI data set to test empirically is a critical step in performing algorithm evaluation. The types of targets and backgrounds existing within a specific data set will largely determine the results of target detection. Without a widely accepted metric for HSI quality, characterizing a data set is a difficult task. There are a couple of solutions to this problem which prevent the need for a detailed description of the experimental data. First is selection of a data set that is most commonly used in the field or considered a “gold standard” for testing a given problem. Second is the use of synthetic data which can be easily characterized with a few straightforward parameters. For this study, a pair of "gold standard" data sets were used. Specifically, data from the Forest Radiance I and Desert Radiance II experiments were used to test the background characterization techniques.

3.1.1 The Canonic Data Set

The Forest Radiance I and Desert Radiance II experiments were conducted as part of the Hyperspectral MASINT Support to Military Operations program, using the HYDICE sensor to obtain hyperspectral imagery for use by the Department of Defense. The utility of the imagery captured in the Forest Radiance I experiment for evaluating hyperspectral algorithms has been well established in the literature [Bergman, 1996], [Chang and Du, 1999], [Healey and Slater, 1999], [Manolakis and Shaw, 2002]. The Forest Radiance I data that was used is part of a Canonic data set, provided to RIT by MIT Lincoln

Laboratory. The 210 band HYDICE imagery was collected over the U.S. Army Aberdeen Proving Grounds, Maryland, in August 1995. Several data cubes were taken during the experiment, including runs at varying altitude. To allow for both fully resolved and sub pixel targets, the lowest altitude pass, labeled Run05, was used for this study. Spatially, the Run05 image covers 320 by 1280 pixels, which are approximately one square meter. Spectrally, it covers the VISNIR portion of the EM spectrum from 0.4 to 2.5 µm, with spectral bands about 0.010 µm wide [Manolakis and Shaw, 2002]. Figure 3.1 is a three band example image taken from Run05 of the Forest Radiance I data set.

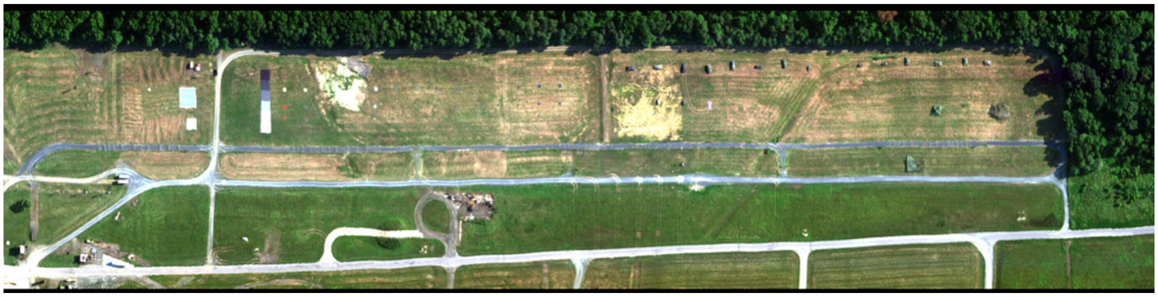


Figure 3.1 Example Image from Forest Radiance I Data Set

The scene is composed of grass, soil, trees, roads, and twenty-two distinct targets. The scene provides a relatively uncluttered background with only a handful of ground cover types; a consideration which will play an important role in generalization of algorithm results.

The Desert Radiance II data also contained passes at several altitudes and was included in the Canonic data set. Again, the lowest altitude pass, labeled Run03, was used. This cube has pixel dimensions of 320 by 960 and roughly the same spatial geometry as Forest Radiance I. The same sensor was used, so the data has the same spectral characteristics. A three band example of Run03 from the Desert Radiance II experiment is shown in Figure 3.2.

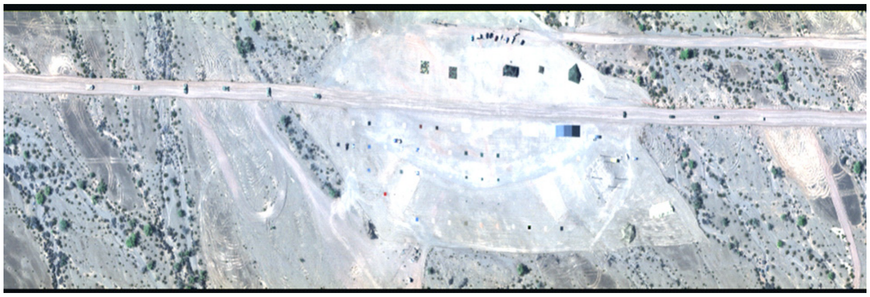


Figure 3.2 Example Image from Desert Radiance II Data Set

The desert scene is composed of sand, scrub brush, and road and contains fourty-eight distinct targets. As a benefit of being collected during the same program, both the Forest and Desert Radiance data sets share many of the same targets. The extensive ground truth available for the image includes measured spectra for the targets with corresponding target truth maps. The truth maps separate target pixels into five categories. From these, truth maps were established by the numbering convention in Table 3.1.

Pixel Category	Truth Map Value
Guard	-1
Background	0
Sub Pixel Targets	2
Targets in Shadow	4
Targets with Solar Glare	6
Fully Resolved Targets	8

Table 3.1 Truth Map Pixel Value Assignments

Fully resolved targets were pixels selected to contain 100% abundance of the target signal, glare targets were pixels containing some degree of specular solar reflection (usually from vehicle windshields), shadowed targets were pixels on the target that were in the shadow of another part of the target, sub pixel targets were pixels containing less than 100% abundance of the target, background pixels were those known not to contain target, and guard pixels were pixels, usually surrounding the target, with unknown content. For the purpose of the experiment, guard pixels were ignored when calculating results. Figure 3.3 is a false color example of a target truth map, where guard pixels are black, sub pixel targets are yellow, shadow target pixel are blue, and full pixel targets are red.

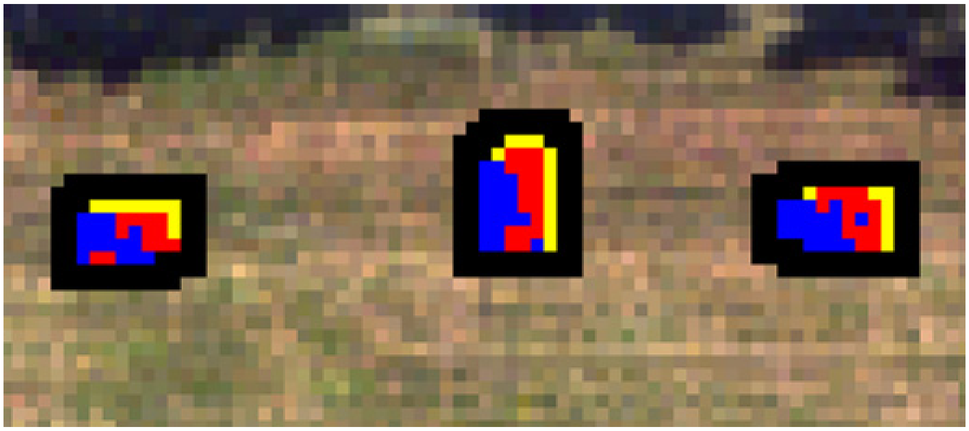


Figure 3.3 Example Target Truth Map

3.1.2 Canonic Data Set Preprocessing

The purpose of the Canonic data set was to provide an equal footing for evaluation of exploitation algorithms. In order to level the field for algorithms operating in reflectance space, several atmospheric compensation algorithms were run on the data and analyzed. Likely due to the calibration panels present in the scenes, the empirical line method was shown to produce the least error [MIT Lincoln Laboratory, 2004]. Based on this analysis, the ELM compensated data provided in the data set was used for the study.

As previously mentioned, the raw HYDICE data contained 210 spectral bands across the VISNIR portion of the spectrum. To eliminate atmospheric absorption features and bands degraded by sensor artifacts, the Canonic data set included bad bands lists which cut the Forest and Desert Radiance images to 145 and 144 bands respectively. In order to maintain the statistical integrity of these images, the full spectral dimensionality was used as much as possible. With the exception of the use of PCA to analyze the data and in a few cases to speed classification, no further dimensionality reduction was performed.

The next step to prepare the data for exploitation was the construction of target spectra from the collected ground truth. Ground truth data files in the data set contained spectra with 430 bands. While this provided thoroughly over-sampled data, the band centers of the truth measurements did not align with HYDICE band centers for either run. This necessitated an interpolation between bands in the measurements. While many types of interpolation and convolution techniques were available, the ENVI spectral resampling tool proved convenient and sufficiently accurate. Bad bands lists were also applied to keep the same 145 or 144 bands as the imagery. Figure 3.4 shows a raw ground truth spectrum and the results of resampling and bad band exclusion.

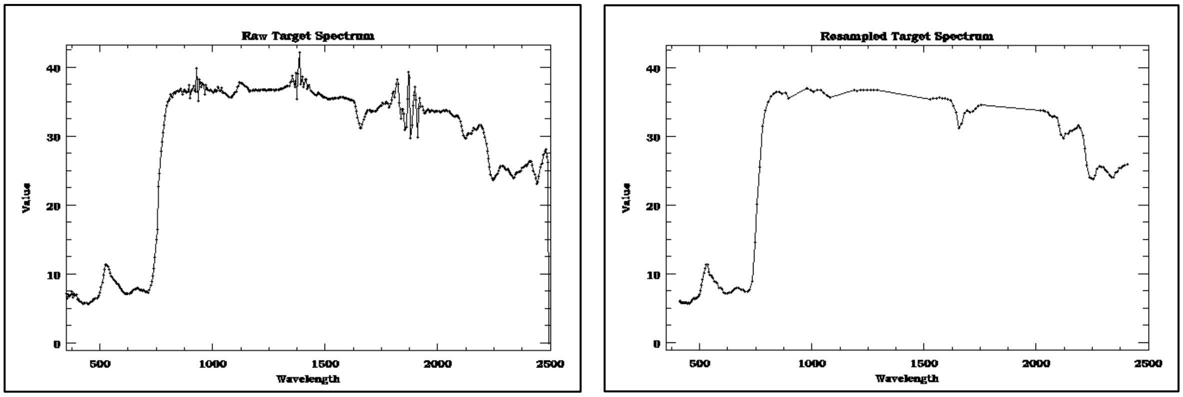


Figure 3.4 (a) Raw Ground Truth Spectrum and (b) Same Spectrum after Resampling and Bad Band Exclusion.

Several ground truth measurements were made on each target to provide more accurate calculation of a target truth spectrum. For homogeneous targets like panels or tarps, averaging these spectra served to reduce the measurement noise and provide a single reliable spectrum. This process was much more questionable when considering heterogeneous targets such as camouflage netting or vehicles. The camouflage measurements were undoubtedly affected by the measurement location on the camouflage and voids in the netting must have made it difficult to capture a measurement untainted by the subsurface. For the vehicle targets, locations of the measurements were documented and separated into categories. In order to construct a single target spectrum, only measurements taken on the main part of each vehicle target were averaged. For example, the few measurements taken on the turret of a vehicle were excluded and the many measurements taken on the body were averaged. This technique is certainly a source of error in the detection chain, especially for the low altitude runs, where there are many fully resolved pixels on each target. For the high altitude runs, the spectral signatures are actually averaged during collection when they are captured in a single detector element, a fact which has been used to explain better performance at higher altitudes when using averaged ground truth spectra [Cisz et al, 2005]. For the most accurate detection results, a separate target class could be established for each type of target spectrum on a heterogeneous target, but for the sake of practicality all pixels on a target were sought using the averaged spectra of the dominant target region.

Another target spectrum consideration to note is the use of a spectral median function vice the spectral mean to establish a single target spectrum. The spectral median is the spectrum from a set which contains the most occurring median values in a band-to-band comparison. Assuming that most of the variability between several measurements on a single target is due to illumination, the spectral median would capture the "middle" illuminated case. Averaging the set of spectra results in an artificially smooth target spectrum, while the spectral median preserves the shape (and unfortunately, the noise) of a real ground truth measurement. Figure 3.5 shows a collection of ground truth spectra measured for a representative target in the scene. The figure includes the spectral mean and median of the set for comparison.

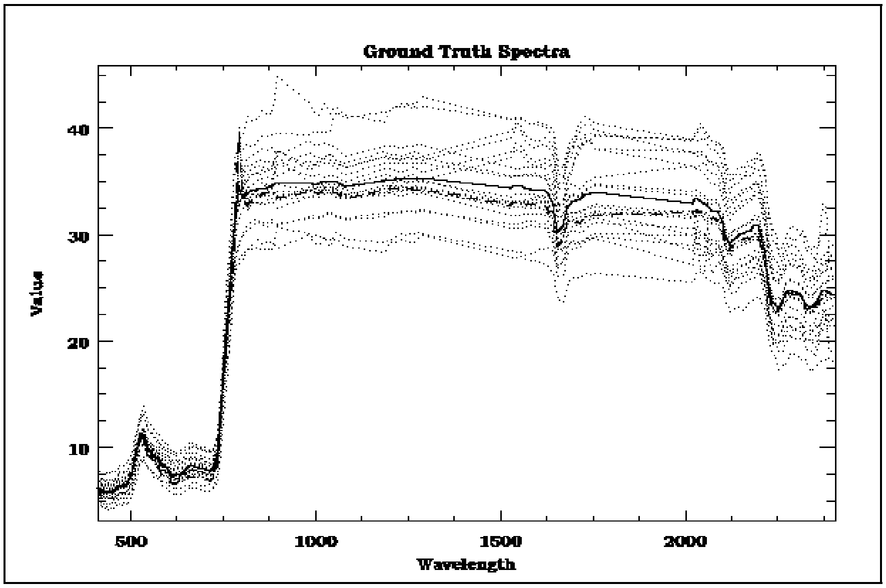


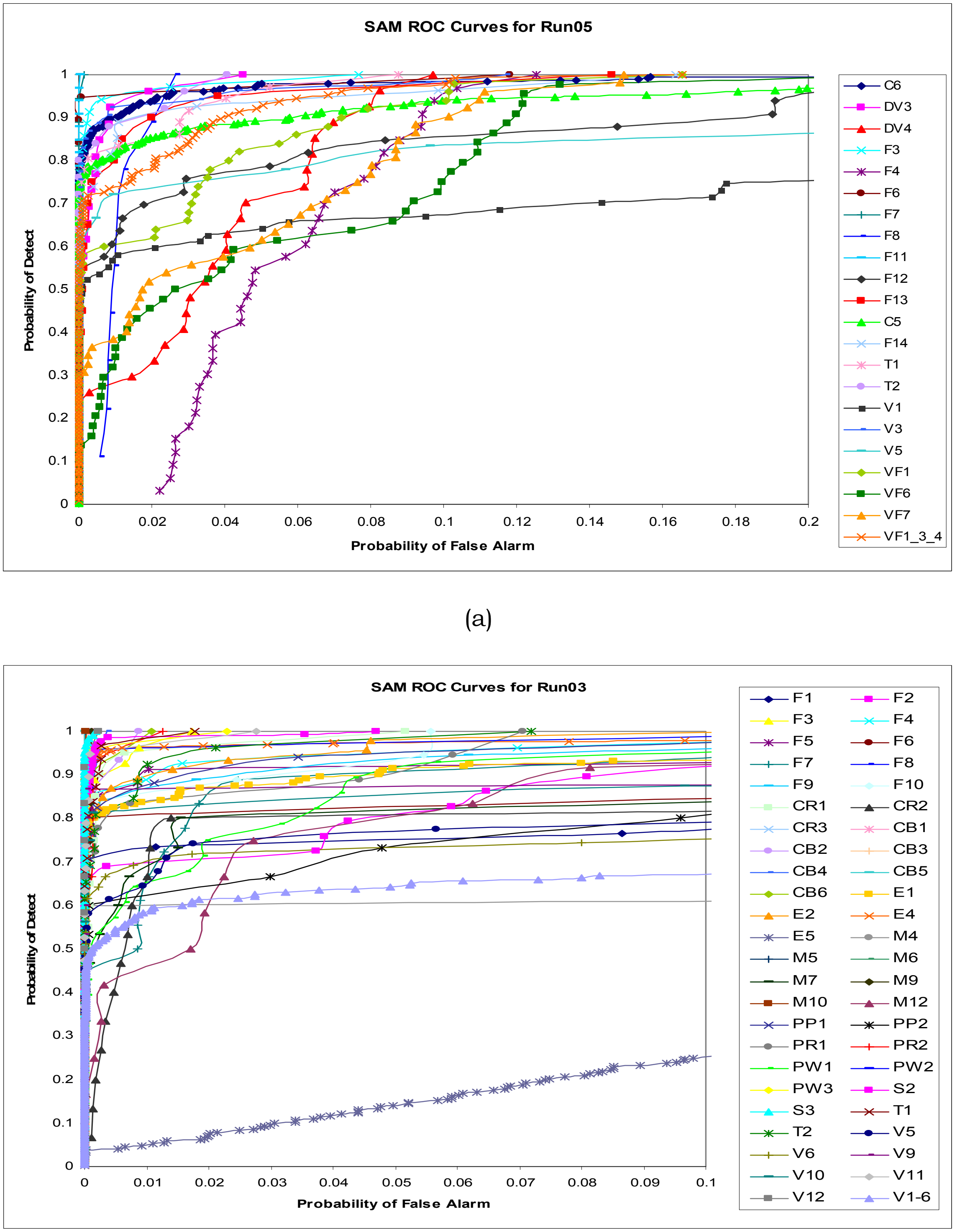
Figure 3.5 Collection of Ground Truth Spectra for a Single Target with the Spectral Mean (Solid) and Spectral Median (Dashed) Shown.

No comparison between these methods or other methods that may be used to combine several measurements into a single target spectrum was performed as part of this study. However, it is important to note that the spectral averaging method used is not the only option available.

The final preprocessing step involves the number and variety of targets within the scene. The collection experiment included a great number of targets which were designed to give many test cases to detection algorithms. One possible use of the data set is to test the ability of an algorithm to distinguish between several similar man made materials, i.e. two different types of green cloth. Another plausible use is to isolate each of the targets and test the ability of an algorithm to locate the target in a natural background. This latter technique was used in this study through the simple method of assigning all target pixels -- aside from those being sought -- as guard pixels. As previously noted, guard pixels were excluded from the computation of results, and thus not considered detects or false alarms. While this does artificially inflate the detection rate by using an entirely unrealistic level of ground truth (specifically, the location of all placed targets in the scene), in actuality it only changes the starting point of the experiment by changing the test data set. This eliminates the cases of false alarms on other similar targets in the scene equally for all algorithms and improves the overall shape of the ROC curve.

3.1.3 Target Selection

One of the strengths of the Forest and Desert Radiance data sets is the number and variety of targets they contain. Being limited by computation power and time, an effort was made to limit the number of targets investigated while preserving the benefit of target variety. To do this, each target in both data sets was detected using the SAM algorithm. This background-independent matched filter provided a simple and fast method of generating a detection statistic image from which a ROC curve could be generated. Figure 3.6 (a) and (b) show the ROC curves for all targets in the Forest and Desert Radiance scenes respectively.



(a)

(b)

Figure 3.6 Spectral Angle Mapper ROC Curves for All Targets in (a) Forest Radiance and (b) Desert Radiance Images.

Average false alarm rates were calculated from each of the curves and tabulated along with the number of target pixels and a team of analysts from the RIT target detection group selected a subset of the targets to serve as representative of the set. Table 3.2 shows the AFAR results and pixel counts for each target, with the selected targets highlighted in red.

Forest Radiance I – 22 Targets

Target	AFAR	N
V1	0.10133	67
V5	0.06551	18
F4	0.05644	33
VF6	0.05301	44
VF7	0.04056	52
F12	0.03908	33
DV4	0.03613	27
C5	0.02365	344
VF1	0.02340	50
F14	0.01242	27
VF134	0.01218	128
F8	0.01212	9
F13	0.01208	20
T1	0.00785	36
F6	0.00625	19
V3	0.00595	53
C6	0.00581	492
DV3	0.00430	26
T2	0.00427	25
F3	0.00331	35
F7	0.00009	18
F11	0.00002	33

(a)

Desert Radiance II – 48 Targets

Target	AFAR	N
M4	0.357627	15
E5	0.312323	292
V1-6	0.247871	235
V6	0.169615	39
V9	0.122027	15
V10	0.119890	16
CR2	0.111754	15
V5	0.101314	31
F1	0.070784	34
M12	0.055823	12
M7	0.049280	15
PP2	0.046607	17
F6	0.031118	15
F2	0.030571	29
F5	0.028649	34
E1	0.026687	221
PW1	0.019088	28
F9	0.018764	18
F7	0.018350	18
PP1	0.012425	17
E4	0.011887	385
PR1	0.010315	18
F4	0.009805	27
F10	0.007680	18
PW2	0.006541	23
E2	0.005622	46
T2	0.004851	26
CR1	0.003567	20
V11	0.001612	19
PR2	0.001356	15
F3	0.001260	27
CB2	0.001224	15
CB6	0.001193	9
PW3	0.001084	23
T1	0.001011	31
S2	0.000817	132
CB4	0.000481	15
M9	0.000383	12
M6	0.000319	12
CB1	0.000226	15
V12	0.000189	12
M5	0.000143	12
M10	0.000094	12
F8	0.000058	12
CB3	0.000046	15
S3	0.000040	120
CR3	0.000031	12
CB5	0.000028	15

(b)

Table 3.2 Target Name, AFAR and Number of Target Pixels for All Targets in the
(a) Forest Radiance and (b) Desert Radiance Images.

In two cases, several of the targets were the same class of vehicle and were assumed to be similar in composition and paint characteristics. These two sets, VF1, VF3, and VF4 (labeled VF134) as well as V1 through V6 (labeled V1-6) were combined into two single targets by averaging all available ground truth spectra and detecting all targets in the class. In general, an attempt was made to select targets of differing size, type, and difficulty of detection. While all of the targets were not used for all of the examples, only the targets selected were examined in any of the experiments.

3.2 Matched Filter Implementation

The computer programming involved in the implementation of a matched filter was a relatively straightforward process. Using a matched filter to generate a detection statistic map is as simple as an inversion followed by a series of matrix multiplication and simple algebraic operations repeated on each pixel in the scene. The process of deciding which statistics are appropriate and how they are to be calculated and inverted is not as simple as plugging them into the matched filter equation. Differences in the type of statistics employed, the method of calculation, and the method of inversion can lead to dramatically varying results. This necessitates a careful examination of these decisions and a study of the tradeoffs.

3.2.1 Mean Centering and Covariance Versus Correlation

Two discrepancies among techniques in the literature are the use of a covariance versus a correlation matrix and the omission or inclusion of the mean. For instance, the GLRT detector may be implemented in either of the following manners.

$$T_{GLRT} = \frac{((d-\mu)^T \Sigma^{-1} \, (x-\mu))^2}{((d-\mu)^T \Sigma^{-1} \, (d-\mu))(1+(x-\mu)^T \Sigma^{-1} \, (x-\mu))} \tag{3.1}$$

$$T_{GLRT} = \frac{(d^T R^{-1} \, x)^2}{(d^T R^{-1} \, d)(1+(x^T R^{-1} \, x)} \tag{3.2}$$

While these may seem to be separate decisions, it is simple to establish a link in the mathematics and physical logic behind this decision. First, consider the method of calculating both types of matrices.

$$\Sigma = E\{(x-\mu)^T (x-\mu)\} \tag{3.3}$$

$$R = E\{x^T x\} \tag{3.4}$$

From the expectation values, we see that covariance works in a mean subtracted space while correlation does not. When substituting these expectation values into the statistical distance equation, the expectation value of the distance is unity when the target is equal to the test pixel and the covariance is ideally formulated, i.e.

$$E\{T \mid d = x\} = (x-\mu)^T \Sigma^{-1} \, (x-\mu) = \frac{(x-\mu)^T (x-\mu)}{(x-\mu)^T (x-\mu)} = 1 \tag{3.5}$$

When plugged into the full GLRT expression, the expectation value is actually one-half, due to the denominator of the expression. The same substitution can be made for the correlation matrix and the non mean subtracted version of the GLRT.

A physical rationale behind the decision to mean subtract is not widely explored in the literature. One simple rationale is related to the projection of the data into reflectance space by atmospheric compensation. Errors in compensation lead to unreliable absolute magnitudes of reflectance. The correlation matrix has not removed the magnitude of the data and may therefore carry this error through. On the other hand, the covariance matrix puts both ground truth spectra and scene spectra in a mean centered space and thus may lessen the effects of compensation error. Another rationale for mean subtraction is to link this decision with the perceived contrast of the target. If the target is very dissimilar compared to the background, the (x-µ) term will be substantial when testing a target pixel (i.e. when x = d), but will be very small when the test pixel is a background pixel (i.e. when x = E{x} = µ). However, if the target spectrum is very similar to the background, the (d-µ) term will be a highly variable and small value (ε), which will be matched to the natural target variability (n) in the test pixel. In equation form,

$$\begin{aligned} &when\ x = d + n\ \ and\ d - \mu = \varepsilon \\ &(x-\mu) = (\varepsilon + \mu + n - \mu) \approx n\ \ for\ n >> \varepsilon \\ &so\ (d-\mu)^T (x-\mu) \approx \varepsilon^T\ n \end{aligned} \qquad 3.6$$

which is obviously not a good measure of target presence in a target test pixel. Additionally, the ε term may be a better match to the mean subtracted pixel spectrum of a non-target pixel. Searching for a target with a flat or weak signal like ε increases susceptibility to false alarms. Another proposed mean subtraction scheme is to mean subtract the test pixel, but not the target spectrum. This implementation of the matched filter would take the form

$$T_{GLRT} = \frac{(d^T \Sigma^{-1}\ (x-\mu))^2}{(d^T \Sigma^{-1}\ d)(1 + ((x-\mu)^T \Sigma^{-1}\ (x-\mu))}\,. \qquad 3.7$$

and will be called the low contrast mean subtraction technique. This filter is certainly not optimal as it attempts to match the original target spectrum with a mean subtracted pixel, normalized by the mean subtracted covariance. For low contrast targets this method may be preferable to the matching of ε with the mean subtracted pixel. The application of one of these methods will thus depend on a reliable measure of target contrast. The development of such a metric and the respective tailoring of the detection statistics will be introduced in this study, but a comprehensive examination of this relationship has been left for future work. For the main portion of the experiment, the covariance matrix with mean subtracted spectra was the statistical set chosen, accepting the possibility of poor performance for low contrast targets. With this decision made, the next step is to examine the method of covariance calculation.

3.2.2 Method of Covariance Calculation

Four methods of covariance calculation were examined to establish a comparison between two programming platforms; an estimation technique and the type of variable (number of carried significant digits). IDL and Matlab both contain covariance functions which perform a traditional covariance calculation, both normalizing by 1/(n-1) where n is the number of samples. Another common method of covariance estimation is to derive the covariance from the correlation matrix. The two can be directly related by expanding the inner product of the covariance matrix equation, discarding the cross terms, and then pulling out the n-independent means to produce

$$\tilde{\Sigma} = \frac{1}{N}\sum_{n=1}^{N}(x(n))(x(n))^T - \mu\mu^T \qquad 3.8$$

or

$$\tilde{\Sigma} = \hat{R} - \mu\mu^T \qquad 3.9$$

where the ~ denotes estimation due to loss of the cross terms and ^ denotes estimation from a sample data set. It should be noted that while in the case of the expectation value, the cross terms cancel; in the case of estimation from sample data, discarding the cross terms will cause error. Finally, all calculations can be done in floating point or double precision variables. A simple experiment was used to determine the amount of error in each of the methods in order to establish confidence in one method.

To test the error involved in each method, a synthetic data set was generated from a given covariance matrix and each routine was used to calculate a covariance to be compared to the original. To generate the target covariance matrix, the IDL covariance routine was used to calculate the covariance of a portion of the Forest Radiance I data. Eigenvalue decomposition was performed on the covariance by the equation

$$\Sigma = \Phi^T \ \Lambda \ \Phi \qquad 3.10$$

where Φ is the matrix of eigenvectors and Λ is a diagonal matrix containing the eigenvalues. In order to form a random pixel ŷ that comes from a data set with covariance Σ, a random vector x was formed with a zero mean and identity covariance was formed, and Λ was divided into $\Lambda^{1/2}\Lambda^{1/2}$, to give

$$\Sigma = E[\hat{y}^T \hat{y}] = \Phi^T \Lambda^{1/2} \, x^T \ x \ \Lambda^{1/2} \, \Phi \quad \therefore \quad \hat{y} = \Phi \ \Lambda^{1/2} \ x \qquad 3.11$$

where x was generated by the IDL “RandomN” random number generator. This process was repeated, with a changing seed for the number generator, to construct an image. The image in Figure 3.7 is a sample from the 170 band 120,000 pixel test data set generated by this process. For this experiment, the decision to use 170 bands versus 145 was driven by the selection of a different bad band exclusion set from the same Forest Radiance I data.

Figure 3.7 Three Bands from the Test Data Set

Using this synthetic data with a known covariance matrix, an root mean squared difference (RMSD) comparison was made between algorithms. The RMSD and later the root mean squared error (RMSE) were calculated by the same equation

$$RMSE = \sqrt{\frac{1}{k^2}\sum\left(\Sigma_i - \Sigma_j\right)^2} \quad 3.12$$

where the test covariance Σ_i was compared to another test covariance, or the target covariance Σ_j. The table below shows the RMS difference between routines run on the test data set.

IDL vs. Matlab	IDL vs. $\tilde{\Sigma}$	Matlab vs. $\tilde{\Sigma}$	IDL vs. Flt	Mat vs. Flt
2.976 e-2	2.421 e-14	2.976 e-2	2.054 e-6	2.976 e-2

Table 3.3 RMS Difference Comparison of Covariance Calculations

Interestingly the estimation $\widetilde{\Sigma} = R - \mu\mu^T$ seems to have resulted in a very low RMS difference compared to the IDL routine. One possible explanation for this is that the method of constructing the test data was conducive to low error by this method of estimation. For real-world data sets, this estimation is not expected to perform as well. Another observation is that the computationally equivalent Matlab and IDL routines seem to have diverged somewhat in their result. To get an idea of the accuracy of the routines, the results were also compared to the original covariance from which the data set was created. The table below shows RMSE for each routine.

Matlab	IDL	IDL $\widetilde{\Sigma}$	IDL (Flt)
2.998 e-2	2.734 e-3	2.734 e-3	2.735 e-3

Table 3.4 RMSE Comparison of Covariance Calculations

As expected, the IDL routines, including the estimation and the floating point calculation are all very close. The results also indicate that the IDL routines resulted in less RMSE than Matlab for this test data. This may be due to the IDL-generated target covariance. It should also be noted that while data and results were exchanged between Matlab and IDL through double precision ASCII formatted text files, the data exchange does represent a possible source of error which was introduced only to the Matlab result. The result is generally meant to show that comparable results were achieved using both Matlab and IDL. A full comparison with a more robust data set transferred under more controlled conditions would be required to declare a conclusive "winner" between the two platforms, but for the purpose of this study it was concluded that the IDL covariance routine was sufficiently accurate.

3.2.3 Method of Covariance Inversion

The next step in the process is inverting the covariance matrix in order to put it to use in the measure of statistical distance. The matrix inverse is a matrix which, when multiplied by the original matrix, will result in the identity matrix (i.e. $\Sigma \Sigma^{-1} = I$). This equation may be solved using Gaussian elimination, a technique for solving a series of simultaneous equations. As a simple example, consider the matrix equation

$$\begin{bmatrix} 6 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 14 \end{bmatrix}$$

and it's Gaussian elimination solution, found by performing the steps:

row 2 – (1/3 * row 1) results in $\begin{bmatrix} 6 & 3 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 15 \\ 9 \end{bmatrix}$

divide by 3 results in $\begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$

so $x_2 = 3$ and $x_1 = 1$

This can only be performed under two conditions: there must be more equations than unknowns and the equations must be independent (e.g. not related by a simple factor). When a covariance matrix has been built from underdetermined statistics using fewer pixel samples than bands, the result is a singular matrix which cannot be inverted by this process. A simple check for singularity is testing for a null result for the determinant of the covariance matrix. Another sure sign of a singular matrix is one or more zero valued eigenvalues. This second test leads to another method for inverting matrices.

The singular value decomposition (SVD) inversion method uses the solution to the equation

$$\Sigma = U \; SV \; V^T \tag{3.13}$$

where U and V are orthogonal matrices and SV is a diagonal matrix containing the singular values. The utility of this relation in performing matrix inversion is shown by

$$\Sigma^{-1} = U \; SV^{-1} \; V^T \tag{3.14}$$

where the inverse of the singular value matrix can be calculated by inverting each of the singular values along the diagonal. In the case of a singular, non-invertible matrix where at least one case of $\sigma_i = \lambda_i = 0$ occurs, the null singular value can be set to a minimum threshold and the inversion taken as an approximation. In the case of well formed (non-singular) covariance matrices, SVD inversion should theoretically produce a result very close to the Gaussian elimination method. In cases of singular matrices, where Gaussian elimination solutions must be forced, the SVD inversion technique may provide better results.

Another method, tailored for the singular matrix case, is the pseudo inverse defined by the relation

$$\Sigma^{-1} = (\Sigma \; \Sigma^T)^{-1} \; \Sigma^T \tag{3.15}$$

where $\Sigma \; \Sigma^T$ is a non-singular matrix that may be inverted by Gaussian elimination. This is the shortest length least squares solution to the simultaneous equation problem, and is often called Moore-Penrose inversion and denoted Σ# or Σ' [Johnson and Wichern, 2002]. In this study, it will simply be called the pseudo inverse, being sure not to confuse this nomenclature with other SVD pseudo inversion algorithms or the common pseudo inverse for complex matrices.

To test the inversion of the covariance matrix, a simple test was devised to multiply the supposed inverted matrix by the original and test how close the result is to the identity matrix. For this test, we will define RMSE by

$$RMSE = \sqrt{\frac{1}{k^2}\sum\left(I - \Sigma\,\Sigma^{-1}\right)^2} \qquad 3.16$$

where I is the identity matrix and k is again the number of bands. In order to study the relationship between number of sample pixels used in the covariance calculation and singularity, a number of different sample sizes were taken from the Forest Radiance I data set, including 17,000, 1700, 170, and 17 (in order to capture 100, 10, 1, and 0.1 times samples as bands). In order to ensure there was no specific data dependency, the data set was divided into twelve sections labeled A through L in Figure 3.8.

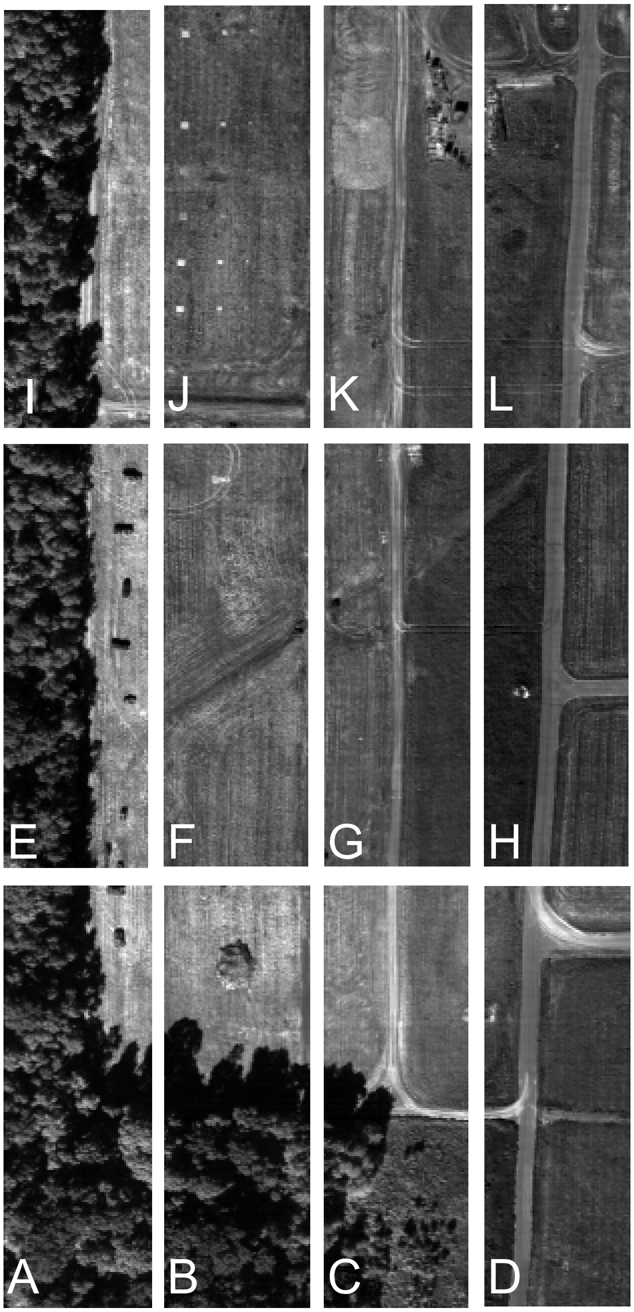


Figure 3.8 Forest Radiance I Inversion Sections

The sections were again subdivided into the previously mentioned subsections, covariance matrices were calculated using the IDL routine, the inverse of each was taken, and the corresponding RMSE was calculated and plotted versus number of pixels. It was noted that each of the sections provided comparable plots, indicating that the covariance inversion was independent of the specific data taken from the data set. The plot in Figure 3.9 uses the average RMSE for each algorithm over the twelve regions and is in a log-log scale to better show the behavior of the three algorithms.

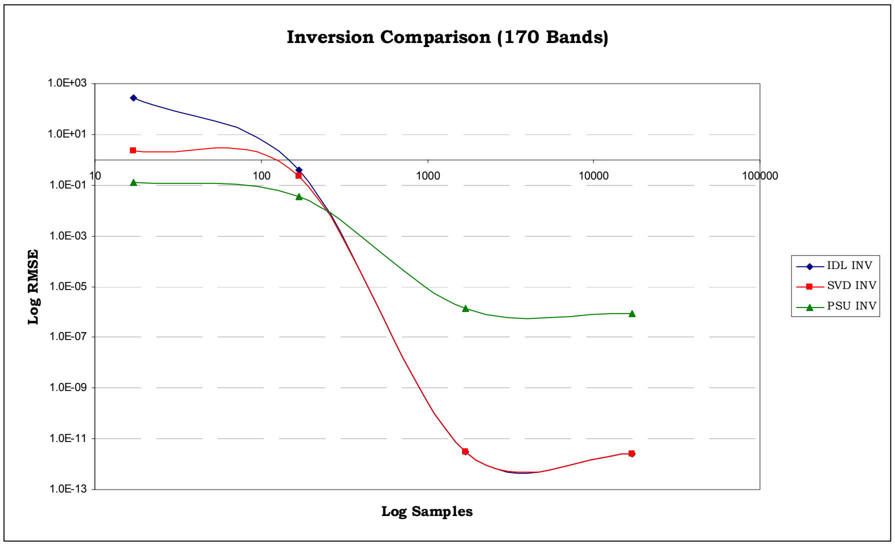


Figure 3.9 Comparison of Inversion Techniques

Note that the curved lines of fit do not represent the data (as depicted by the points) and are only intended to show the general trend. It is apparent from the plot that IDL (Gaussian elimination) inversion and SVD inversion behave nearly identically for well formed covariance matrices and SVD out performs the IDL invert function for poorly formed cases, as expected. Pseudo invert performs poorly in the well formed case, but is the algorithm with lowest error for the poorly formed case. For greater fidelity on the behavior of these algorithms, the plot in Figure 3.10 shows the region where number of samples approaches and exceeds the number of bands. This plot clearly shows the dramatically increasing error in the IDL and SVD algorithms. Again, while SVD outperformed the IDL routine in the poorly formed case, the pseudo inverse prevails with orders of magnitude less RMSE.

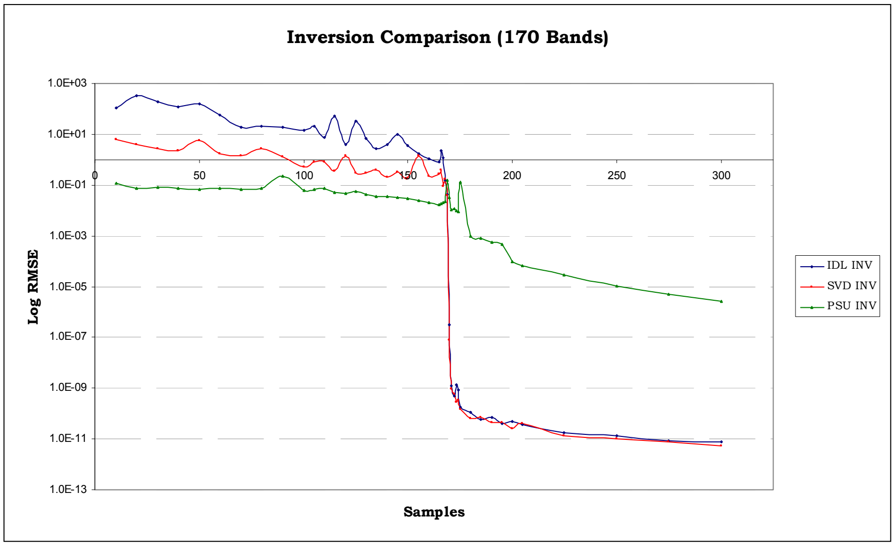


Figure 3.10 170 Band Comparison of Inversion Techniques

To test this behavior for a different number of bands, the same experiment was conducted for twenty-five and then ten bands, the results for which are given respectively in the Figures 3.11 and 3.12. These plots lead to the conclusion that the IDL Gaussian elimination invert function should be used for well formed statistics and the Moore-Penrose invert should be used in the ill-formed case. This conclusion was considered during implementation of all automated background characterization routines where the number of samples was a variable. The number of samples in most backgrounds in this study was sufficient to allow for the use of Gaussian elimination, but there were a few cases where data-starved backgrounds required the pseudo inversion. Detection results based on inversion of singular covariance matrices with the pseudo inverse are pointed out in the results section.

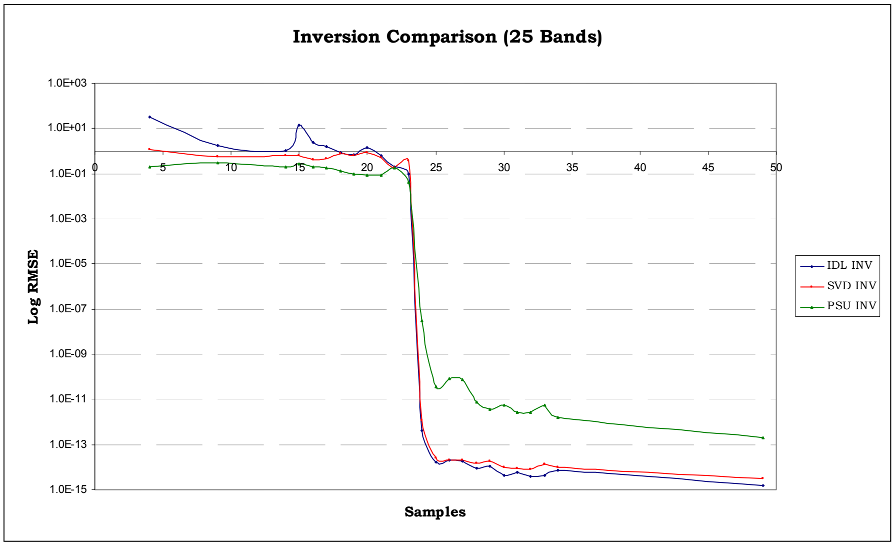


Figure 3.11 Twenty-Five Band Comparison of Inversion Techniques

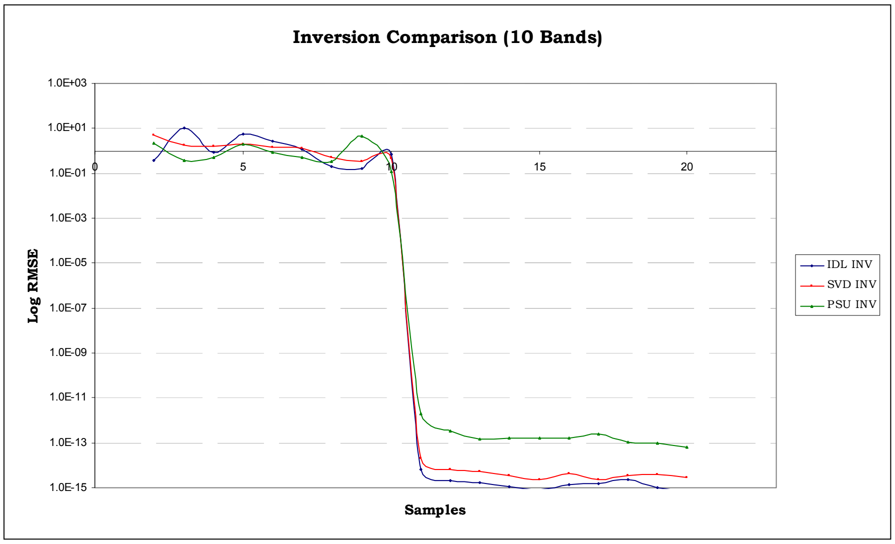


Figure 3.12 Ten Band Comparison of Inversion Techniques

3.2.4 Matched Filter Selection

Various forms of the matched filer are present in the literature, and each time a new variation is presented, a comparison is performed against other filters. Operating under the hypothesis that background characterization sets the statistics and the matched filter only stretches those statistics into more or less favorable directions leads to the assumption that improvements in background characterization outweigh improved formulation of the detector. This assumption, along with the concept of experimentally holding constant the error inherent in a given detector, drives the need to select a single detector and apply it to each background characterization technique. After preliminary results indicated that it may outperform some of the other unstructured detectors, the GLRT detector was selected as the standard. It is undoubtedly true that different pairings of detectors with background characterization methods would provide optimal results, but this type of analysis is left to future work.

3.3 Background Characterization Implementation

Following the current literature, background characterization methods are divided into two categories: spatial and spectral subsetting. Spatial subsetting can be performed either by manual selection of backgrounds in the target approach region of the image, or calculating a spatially variant background by sliding a window over the data. Spectral subsetting is performed by classification (also called pre-clustering or segmenting) of the image in order to derive statistics that better represent background for a given target problem. Spectral subsetting can be used in conjunction with spatial subsetting in order to aid in the decision of which cluster statistics to use as background. The MVN of these backgrounds will differ, and all will be an improvement over the scene-wide data. Several methods for excluding targets from the background have also been employed in both spatial and spectral subsetting techniques. The numerous techniques to be discussed in this section which have been implemented for comparative experimentation, each with their own set of parameters and phenomenology, represent the main thrust of this thesis.

3.3.1 Spatial Subsetting Implementation

The first method of spatial subsetting to be examined was the so-called target approach method. If the user has some knowledge of the general location of the target, part of the image can be captured while the sensor approaches the target area. This serves to ensure the non-existence of target in the background, while capturing species likely to be contained in the target surroundings. Given the Forest and Desert Radiance data sets, the target approach method was simulated by selecting regions of the image where targets are known not to exist. Figure 3.13 (a) and (b) show the target approach regions used in this experiment for the forest and desert scene respectively.

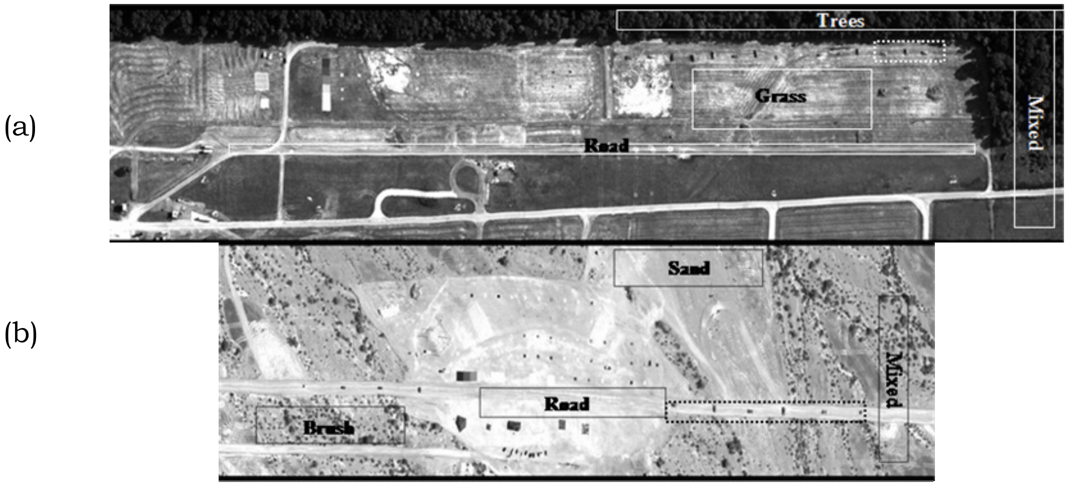


Figure 3.13 (a) Forest Radiance Target Approach Backgrounds,
(b) Desert Radiance Target Approach Backgrounds [West et al, 2005]

These backgrounds were selected in a manner similar to previous studies [Manolakis et al, 2001]. Once the spatial subsets were selected, statistics were calculated, evaluated and used in matched filter detection.

The next method of spatial subsetting is the RX sliding window technique. Since the introduction of the original algorithm in 1990, there have been numerous adaptations. For this reason, it is important to be explicit in the formulation of the algorithm. The sliding window constructed for this experiment is depicted in Figure 3.14, showing the covariance window, mean window, exclusion window, and detection window.

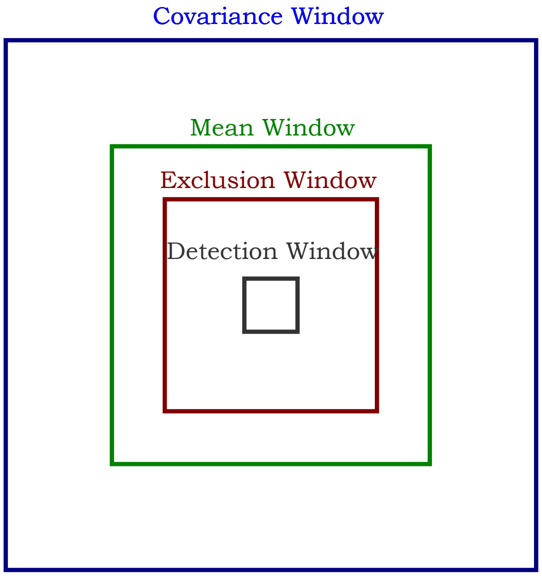


Figure 3.14 Four-Part Sliding Window Implementation

Quite simply, the four main variables considered in this experiment were the sizes of these windows. This section will describe the factors to consider when selecting the sizes of these windows in order from the outside in.

The RX algorithm can be implemented with the covariance window fixed and encompassing the full scene. The original implementation involved a rapidly varying mean and a more slowly varying covariance. The size of the window will impact the stability of the statistics, the MVN of the data seeding the covariance matrix, and the potential to include target species in background. Selection of the size of the covariance will therefore depend on the number of bands in the data cube, the composition of the scene, and the size, shape, and possibility of multiple occurrences of the target. For this experiment the targets were masked to eliminate dependency on the target size, shape and frequency. So, results will give insight into the utility of various sized windows in proportion to the number of bands for scenes with a composition similar to the test data.

Another consideration of the covariance window is the relationship between the mean used to calculate the covariance and the mean used to mean subtract the test pixel. The order of operations for these calculations has varied in the literature. For this implementation, the mean of the data in the covariance window is used to generate the covariance, and simultaneously the mean of the data in the mean window is subtracted from the test pixel. In other studies the covariance has been calculated *after* locally mean subtracting the data [Li et al, 2003]. As a separate step, local mean-subtraction (like convolution) increases the MVN of the data. A covariance matrix calculated from a local window of the local mean-subtracted data set would therefore better adhere to the MVN assumption of the matched filter and provide better separation of target and background. However, local mean subtraction will change the spectra of identical targets in different backgrounds, and therefore spread the distribution of target returns in the detection statistic result. Additionally, the use of a covariance matrix in this case involves a double mean subtraction with two different means. A direct comparison of these methods and their influence on the spread of detection result statistics is left for future work. This study will examine several covariance window sizes to include the extremes of the singular matrix and the full scene, each calculated before or simultaneously along with the mean.

The mean window is the essence of the RX algorithm. Subtracting the local mean from a pixel subtracts a realistic expected value of the pixel if target is not present. In other words, the assumption of the RX algorithm is that the source of interference is the mixing of non target species from within the target pixel, and therefore the removal of that interference will unveil the target. In the matched filter, removing the background expectation value from a test pixel containing target causes the (x-μ) term to strongly resemble the mean subtracted target spectrum (d-μ), thus resulting in a high matched filter score. So, it is easy to see that target presence in this calculation of mean would be extremely detrimental to the process. While target contamination is less of an issue for sub pixel or point targets, it represents a significant weakness in this approach for fully resolved targets. This is another factor contributing to the conclusion that RX is more suited to locate sub pixel rather than full pixel targets. The heavy influence of target contamination necessitates the exclusion window, inside of which no pixels are used for statistical calculations.

Given the approximate physical size of the target being sought and the collection parameters (specifically GSD) of the scene, it is reasonable to assume that a user could estimate the size of the target in pixels. Making the exclusion window twice the size of the target ensures that when pixels on one edge of the target were being detected, pixels on the other edge will still be excluded. For point targets, the exclusion window could simply be a single pixel. Knowing the proper size of the exclusion window must be considered an assumption of this implementation because for oddly shaped or variable sized targets this level of *a priori* knowledge may not be realistic. Another failure point of this method is the case of more than one target placed spatially proximal to each other. In this case, instead of the statistics being self-contaminated by the target being detected, the statistics are cross-contaminated by neighboring targets.

To address both of the self- and cross-contaminated cases, a spectral target exclusion method was employed in conjunction with the sliding exclusion window. This technique involved loosely thresholding the detection statistic map from the SAM algorithm to develop a binary map of possible target locations. These pixels may then be excluded from the mean and covariance windows. A full analysis of the effectiveness of spectral target exclusion by SAM pre-filtering was not part of the spatial subsetting study, but realistic runs were performed to verify the viability of the technique. A more comprehensive study of target exclusion was performed for the spectral subsetting techniques. Description of the study and analysis of the concepts involved in target exclusion will be included in the spectral subsetting implementation section.

The final window size variable is that of the detection window. The presence of this window served only to speed detection by operating the detector on more than the single central test pixel. Given RX run times of several hours for the test images, the use of a nine pixel detection window made run times manageable. Setting a detection window size greater than one assumes a similar local background for each pixel within the window. Because testing this assumption would be very difficult, the nine pixel detection window was generally used for screening and the single pixel detection run for final results. This allowed for many more iterations on other window sizes without changing the final reported outcome.

Aside from the susceptibility of the RX algorithm to target influence, another weakness of the technique springs from the central assumption that the best background for a test pixel may be derived from its immediate surroundings. This assumption certainly fails for fully resolved targets. Consider the case of a scene containing several identical fully resolved target pixels in different surroundings. Subtracting a different local mean from each test pixel leads to the undesirable consequence of a different matched filter score for the same exact target. In addition, there are cases where the algorithm may fail even for subpixel targets. While this assumption is valid for test pixels surrounded by relatively homogeneous regions, it fails along the edges, or transitions, between land cover types. The phenomenology involved with targets sitting at class transitions is difficult to understand and characterize. For example, it would be difficult to characterize the signal interfering with a sub pixel sized target resting on grass partly in the shadow of a prominent tree line. This may involve estimating the mixture behavior of direct, downwelled, and multiple bounce photons, or some combination of the three depending on the collection geometry, atmospheric conditions and adjacency effects. These parameters were not independently considered, and no modifications of the algorithm have been made to combat this problem, but this deficiency in the algorithm is worthy of consideration when judging detection performance.

3.3.2 Spectral Subsetting Implementation

The spectral subsetting of data for improved background characterization involves a number of decisions, assumptions, variables, and parameters which need to be understood for the successful implementation of the techniques. The method of classification, the method of calculating and applying the statistics of those classes, methods of improving class statistics, and the exclusion of targets are the main considerations addressed in this study. The number of classes is another realistic consideration; however, the test data set had established classifications which guided this decision. The methods of classification, improvement of statistics and target exclusion, which will be discussed in detail, are less important than the method of calculating and applying the statistics which will serve as background for the matched filter. Figure 3.15 depicts the generation of the individual class statistics, which are calculated using the data from the image cube that corresponds to the class assignment from the classmap. As with the target approach method, the question for the analyst at this point is which class will best serve as a matched filter background. The methods explored in this study automate this decision process. Pre-clustering can also be combined with the RX sliding window for calculation of the local mean. This leads to eight distinctive techniques which were implemented, each using the classmap in a different manner to characterize the background. First, the means applied to the matched filter were calculated either by a single class or locally from surrounding pixels.

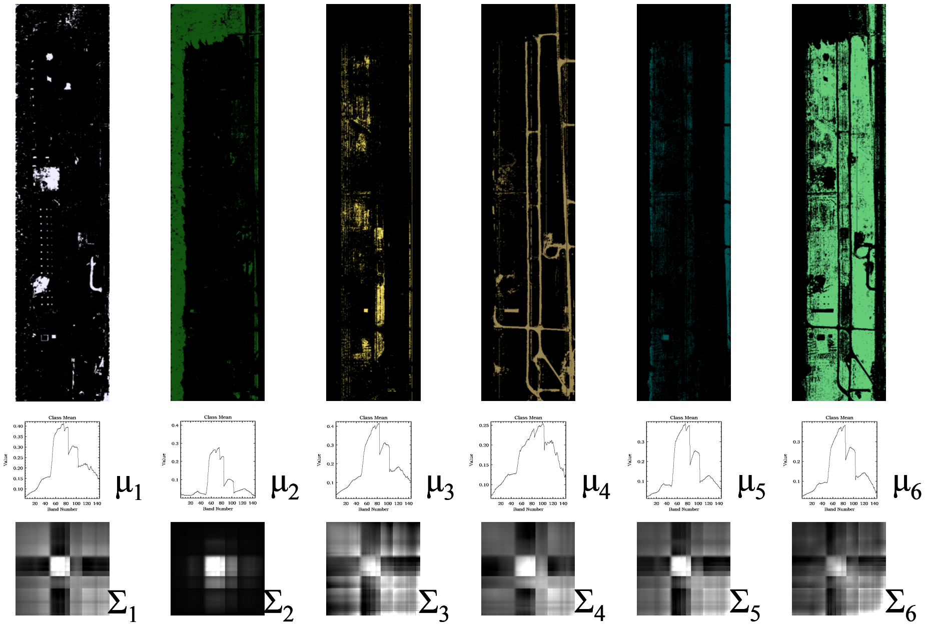


Figure 3.15 Pre-Clustering Class Statistics Generation

This concept is illustrated by Figure 3.16, depicting a mean computed from pixels neighboring the test pixel (just as in the RX implementation), as well as a mean computed from the pixels corresponding to a given class from the classmap.

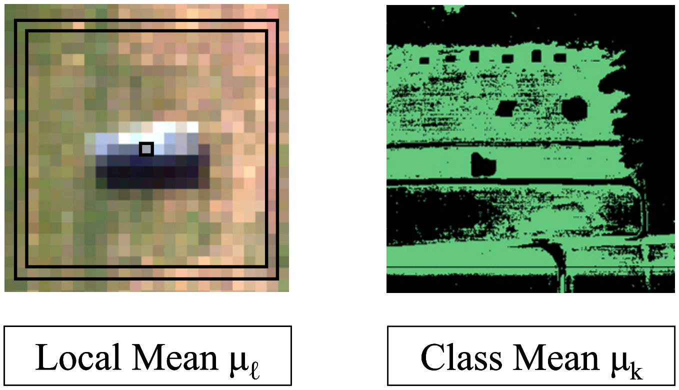
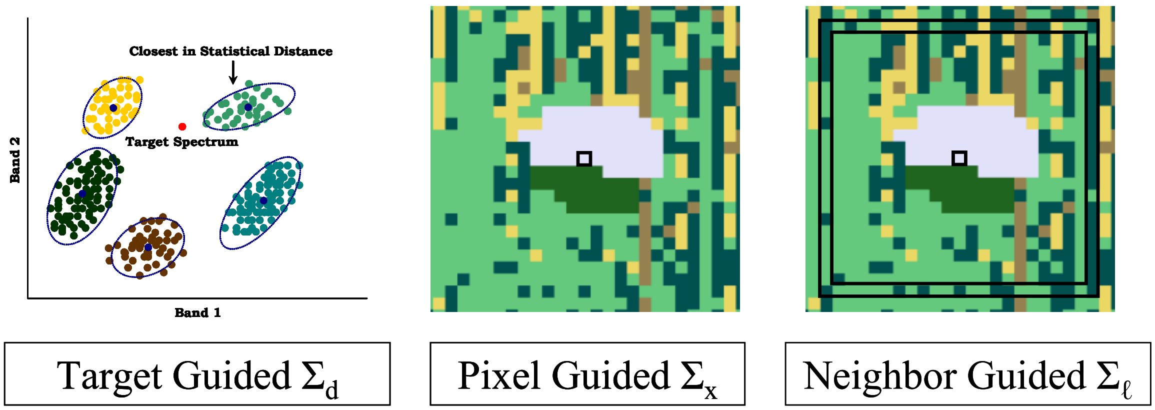


Figure 3.16 Pre-Clustering Mean Selection Methods

Next, an attempt was made to automate the decision of which class or classes are to be used as background based on one of three indicators. First is the relationship between the target pixel and the class (target guided). Second is the relationship between the test pixel and the class (pixel guided). Third is the relationship between the neighbors of the test pixel and the class (neighbor guided). Figure 3.17 is a depiction of the data which guides the decision of which class statistics to use. The class is selected by statistical distance to the target, or using the classmap to identify class assignment of either the test pixel or the neighbors of the test pixel.



Target Guided Σ_d Pixel Guided Σ_x Neighbor Guided Σ_ℓ

Figure 3.17 Pre-Clustering Class Selection Methods

The information of neighboring classes can then be used to select the most common local class, or to create a mixture of class statistics. Two techniques, taking either a class or local mean, fall into each of these four categories. Table 3.5 gives the abbreviated name of the eight techniques based on the two methods of mean calculation and the four methods of class selection.

	Target Guided	Pixel Guided	Neighbor Guided - Mode	Neighbor Guided - Mixed
Class Mean	CMTG	CMPG	CMNG-M	CMNG-X
Local Mean	LMTG	LMPG	LMNG-M	LMNG-X

Table 3.5 Pre-Clustering Technique Names Based on Mean and Cluster Selection

To denote which statistics each of these methods uses, Table 3.5 can also be populated with subscripted mean and covariance variables. Table 3.6 gives this representation of the techniques showing the covariance calculation guided by the target (d), the test pixel (x), the mode of the neighbors (mode), or a mixture of the neighbors (mix), and the mean either calculated for the class along with the covariance or calculated locally (ℓ).

	Target Guided	Pixel Guided	Neighbor Guided - Mode	Neighbor Guided - Mixed
Class Mean	(μ_d , Σ_d)	(μ_x , Σ_x)	$(\mu_{mode} , \Sigma_{mode})$	$(\mu_{mix} , \Sigma_{mix})$
Local Mean	(μ_ℓ , Σ_d)	(μ_ℓ , Σ_x)	$(\mu_\ell , \Sigma_{mode})$	$(\mu_\ell , \Sigma_{mix})$

Table 3.6 Pre-Clustering Technique Variables Based on Mean and Cluster Selection

The sections which describe these techniques and discuss the parameters required to run each algorithm are preceded by a discussion of the implementation of the three classification algorithms.

3.3.2.1 Classification Algorithm Implementation

Three different types of classification algorithms were implemented to test pre-clustering target detection techniques. The K-Means and SEM classification routines were used to generate classmaps for the test data to allow for a direct comparison between algorithms. Later, a modification to SEM dubbed the statistical distance classifier (SDC) was implemented as part of the pre-clustering experiment, which will be discussed in the next section. Once the classmap for an image was generated, a simple program divided the full rank data cube into pools from which means and covariance matrices were calculated. As discussed in the data section, part of the overall experimental design involved isolating each target in the scene in order to remove the unrealistic influence of the many spatially proximal man-made objects present. For consistency, this was carried through in the classification process by removing all targets except the one being sought from the image prior to classification. This ensured that other target species in the scene would not influence the performance of the classification algorithm, and automatically removed all other targets from the class statistics. The parameters required to run the algorithms were held constant for the classification of data containing each target. The selection of these parameters was done with a balance of experimental goals and real-world application in mind.

The only parameter required in the K-Means algorithm was the number of classes (K). The Canonic data set included classmaps that contained five and six classes for the Desert Radiance and Forest Radiance scenes respectively, so this convention was adopted. To speed processing time for SEM, the K-Means classmaps were used as the algorithm starting point, and the data set was reduced in dimension using PCA. The SEM convergence parameter, measuring the change in class means, was tightly set to avoid early convergence, and the class size parameter was loosely set to allow SEM to eliminate classes if they became too small. Once the algorithm had run to the maximum iterations allowed, a local minimum was observed in the convergence parameter and the associated classmap was selected as the result. This method of running the algorithm prevented early convergence, and ensured the best possible classmap for a given number of classes. Reinforcing the K values given in the Canonic data set, there was no difficulty is selecting the SEM convergence parameter which would result in the correct number of classes. So, while agreement in number of classes was forced for the experiment, the same number of classes could have been independently the same. This does not, however, mean that the number of classes used was ideal. In fact, the variety of pixel types (even in this relatively low clutter scene) ensures that mistakes were made during classification. To address this, a method which is not uncommon in statistics was implemented to alleviate some of this error.

As part of the classification process, every pixel in the image was assigned to a single class. For the SEM algorithm, these classes were formed by the Gaussian maximum likelihood of the pixels belonging to a given class. As implemented, the SEM algorithm does not allow for an "other" class of anomalous pixels that have a very low GML probability of belonging to any class. To compensate for this, the method of calculating the final class statistics was altered to exclude the anomalous pixels in each class. Aside from introducing error into the measure of background signal used in the matched filer, including anomalous pixels decreased the MVN of class statistics, seeding the matched filter with data in greater violation of the Gaussian assumption. To eliminate these pixels, a technique called statistical distance exclusion (SDE) was implemented. SDE calculates the class statistics, measures the statistical distance of each pixel in the class pool, eliminates the pixels which lay outside of a given threshold, and then recalculates class statistics without those pixels. This can be combined with spectral target exclusion to ensure class statistics are not contaminated. To illustrate these techniques, Figure 3.18 shows the chi-squared plot for an example class distribution taken from the Forest Radiance Run05 image.

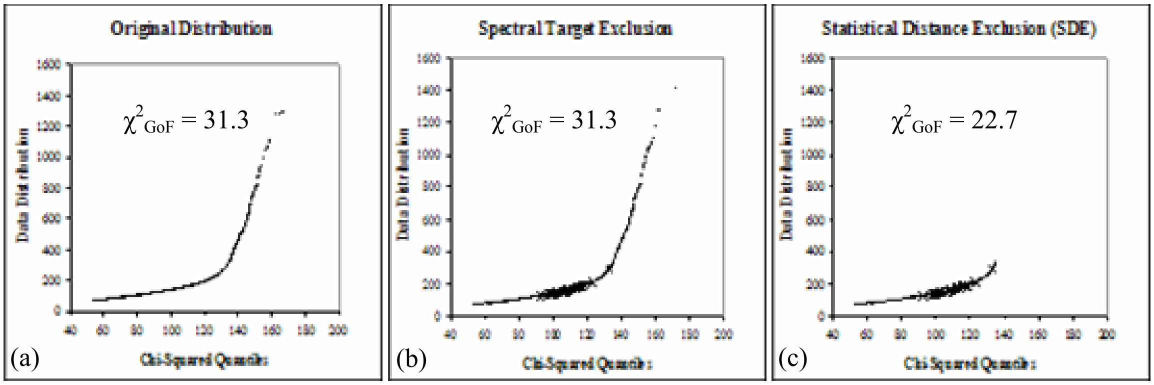


Figure 3.18 Chi-Squared MVN plots showing (a) the original distribution, (b) pixels excluded by SAM prescreening (marked x), and (c) the tail of the distribution reduced by statistical distance exclusion (points omitted). [West et al, 2005]

Now that the implementations of classification and class statistics calculation have been described, the eight methods of applying these statistics to the target detection problem will be explained in the following sections.

3.3.2.2 Class Mean Target Guided

This method simply used the statistics derived from a single cluster as a background for the matched filter. Figure 3.19 depicts the CMTG selection of mean and covariance. The central assumption of the CMTG method is that a single class can be selected to best represent background. The only data available to make this decision (without relying on spatial information in the image) is the target vector. Using this vector to drive the selection of a single class has a physical implication worth describing.

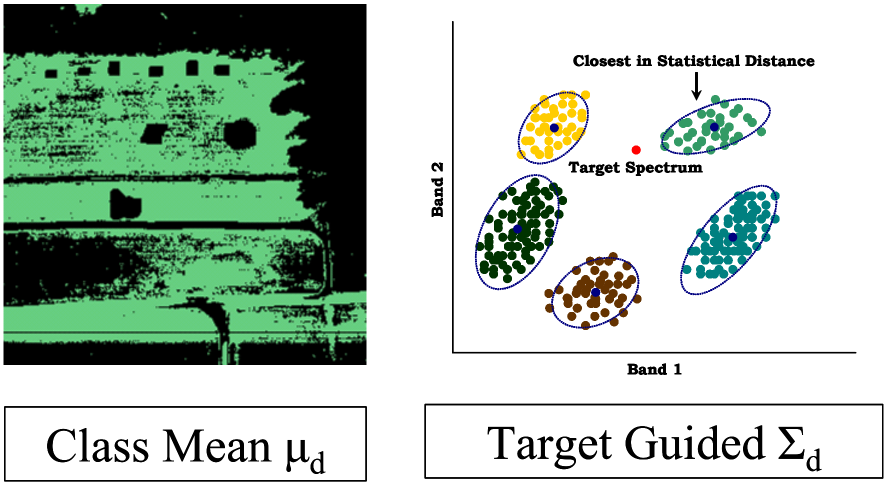


Figure 3.19 Class Mean Target Guided (CMTG) Pre-Clustering

The underlying hypothesis is that the cluster closest to the target will serve as the best background. This hypothesis logically only holds true when the source of interference is not sub pixel mixing. If spatial information is not considered in the algorithm, the results are assumed to be location invariant (so it doesn't matter where in the scene the target rests). This assumption can only be realistic for fully resolved targets in the open (with no adjacency effects). If this is the case, the interference most likely to cause false alarms will come from other target-like species in the scene (impersonators). If these pixels are present in the most target-like class, they will be included in the background and suppressed. In practice, there are two major difficulties when implementing a technique based on this logic. First, the most target-like class will also contain the target pixels, so they must be excluded from the class. Second, the K-Means and SEM discrimination metrics differ from those of the

GLRT detector. Therefore, the classification algorithm and detection algorithm do not agree on which pixels in the scene are similar to the target.

Spectral target exclusion can be applied to handle the cluster target contamination problem by eliminating possible targets from the background data pool. As with the spatial subsetting experiment, the viability of the spectral exclusion technique was confirmed, but the Canonic truth maps were used for the final results to remove the influence of the differing levels of target exclusion among the various targets. In an attempt to abate the problems with mismatched modalities between classifier and detector, a third clustering algorithm was used. A single line code change from the SEM algorithm, the statistical distance classifier (SDC) reduced the SEM discriminant function to use only the statistical distance measure. Recalling that the SEM discriminant was

$$Di = \ln(p(i)) - \frac{\ell}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma_i| - \frac{1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i) \qquad 2.13$$

the SDC discriminant was simply

$$Di = (x-\mu_i)^T \Sigma^{-1}(x-\mu_i) \qquad 3.17$$

The SEM classmaps were used as the algorithm starting point and SDC was restricted in the number of iterations so as to not change the number of classes in the scene. By changing the discrimination metric, SDC used a measure to compare pixels in much the same way as the GLRT. Ideally, the clustering and detection algorithms would have matching spectral discrimination metrics, but that was left to future work. A real-world algorithm based on these principles would involve clustering the data, selecting the cluster with the smallest statistical distance from the target, running a spectral target exclusion routine, calculating the statistics of the resulting data pool, and then running a matched filter over the original image. As previously discussed, this algorithm would be best theoretically suited for fully resolved targets. The next algorithm attempts to extend this concept to the sub pixel detection problem.

3.3.2.3 Local Mean Target Guided

The LMTG technique uses the same method of cluster selection as the previous technique, but instead of a class mean this method uses the RX sliding window to calculate a local mean around each test pixel. Figure 3.20 depicts the LMTG selection of mean and covariance.

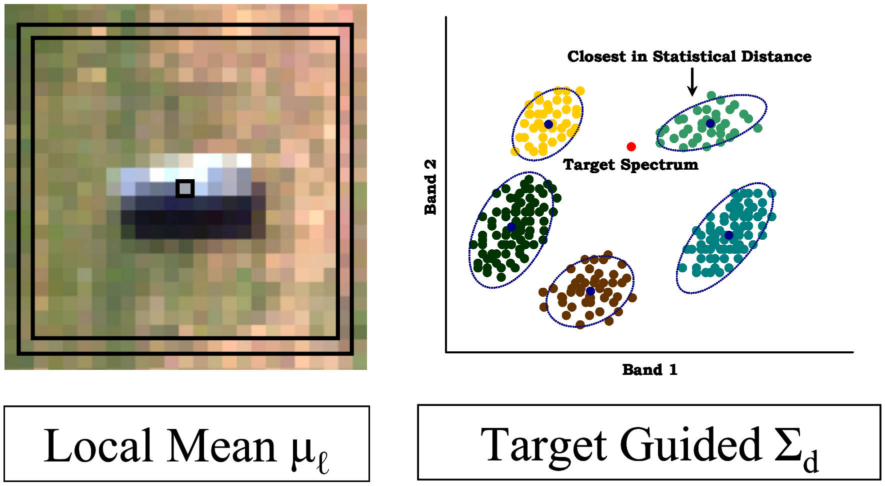


Figure 3.20 Local Mean Target Guided (LMTG) Pre-Clustering

This can be accomplished using the traditional RX implementation, but with a stationary cluster covariance instead of a global covariance. By suppressing target-like species with the covariance and sub pixel mixing with the local mean, this method provides a theoretical measure of versatility. Like the RX algorithm, however, this technique requires an accurate estimate of target size in order to employ spatial target exclusion and avoid self-contamination and assumes that other targets situated proximal to the test pixel have been suppressed by spectral target exclusion to avoid cross-contamination. These exclusion techniques again are critical due to the small sample size of the mean window around the test pixel.

3.3.2.4 Pixel Guided Selection Techniques

Two of the techniques examined use the test pixel to decide which cluster statistics to use in the matched filter. The class mean pixel guided (CMPG) technique operates on the test pixel using the mean and covariance of the class to which the test pixel is assigned. The local mean pixel guided (LMPG) technique uses a covariance based on test pixel class assignment, but a mean from a local window of pixels. Figures 3.21 and 3.22 depict the CMPG and LMPG techniques respectively. The same assumption about suppressing the interference from mixing versus suppressing target impersonators applies to these methods. This technique assumes that the class that best represents the background is the class to which the test pixel belongs. This assumption holds for the anomaly detection problems explored in [Ashton, 1998]. For the target detection problem, this may be the case for test pixels containing target if the classification and detection algorithms have a similar discriminant function. However, using the class covariance for a non-target test pixel may not be the optimal method for false alarm suppression.

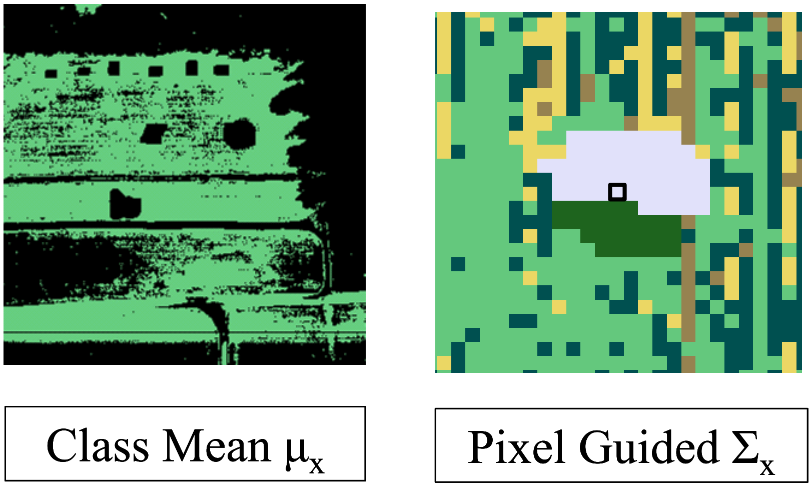


Figure 3.21 Class Mean Pixel Guided (CMPG) Pre-Clustering

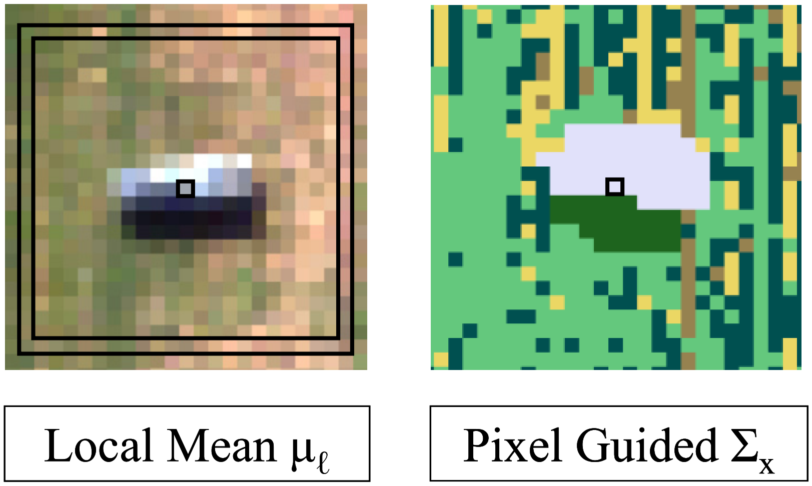


Figure 3.22 Local Mean Pixel Guided (LMPG) Pre-Clustering

3.3.2.5 Neighbor Guided – Mode Selection Techniques

The neighbor guided – mode selection techniques poll neighboring pixels to identify which classes are proximal to the test pixel. These techniques use the statistical mode class to derive the matched filter background. In the class mean neighbor guided – mode (CMNG-M) method, the most common class assignment among the pixels in the local window is used to select which class mean and covariance to use in the GLRT detector. Similarly, the local mean neighbor guided – mode (LMNG-M) method, uses the mode to select a class covariance, but uses the mean of the local window to operate on the test pixel. Figures 3.23 and 3.24 respectively depict the CMNG-M and LMNG-M selection of mean and covariance.

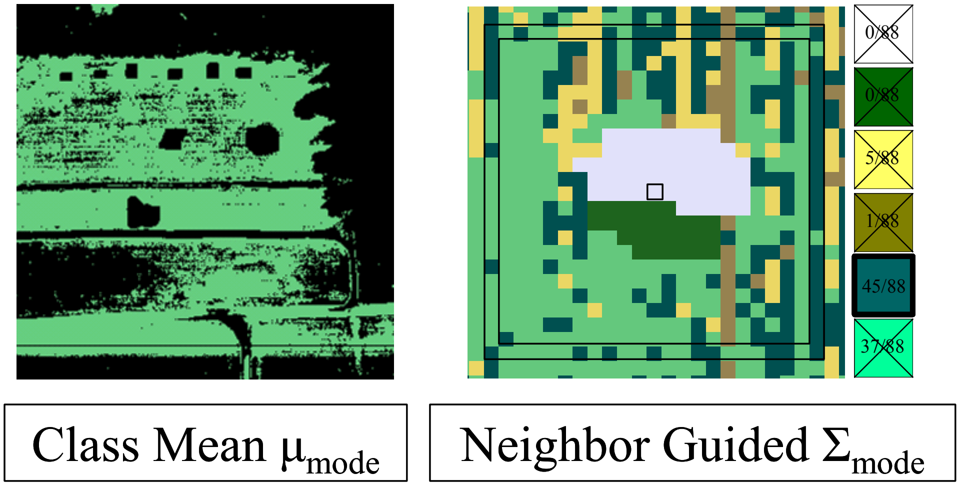


Figure 3.23 Class Mean Neighbor Guided - Mode (CMNG-M) Pre-Clustering

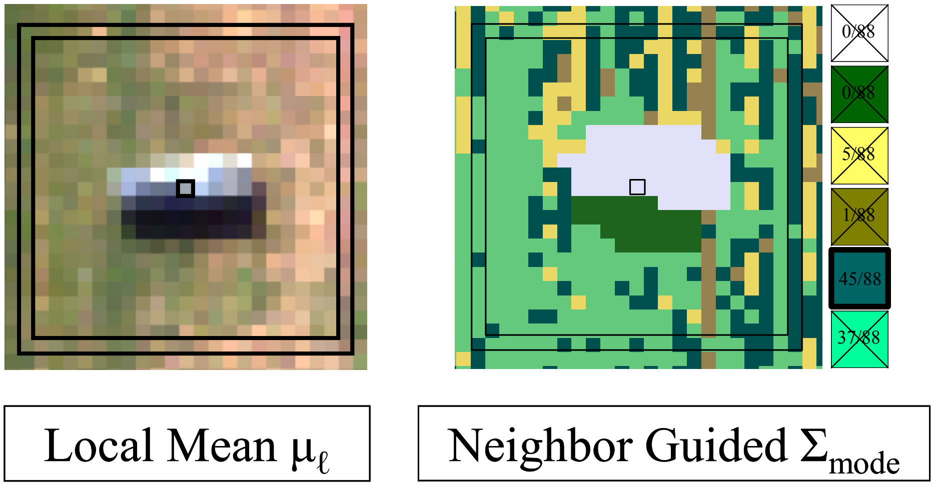


Figure 3.24 Local Mean Neighbor Guided - Mode (LMNG-M) Pre-Clustering

Assuming that the pixels most abundant in the surroundings of the test pixel would dominate the statistics, these methods substitute the well-formed statistics from pre-clustering for the more variable statistics of a sliding covariance window. So, similar to the traditional implementation of RX, this method assumes that the best background statistics can be derived from the immediate surroundings of the test pixel. However, using the mode class of the local window inherently discards information gained by polling the test pixel surroundings. The presence of other classes in the local window may have an impact on detection, especially along class transitions. The next methods seek to remedy this theoretical downfall by incorporating all of the information gained in polling the neighboring pixels.

3.3.2.6 Neighbor Guided – Mixed Statistics Techniques

The next two methods diverge from the paradigm of selecting a single class to provide statistics for the matched filter background. The class mean neighbor guided – mixed (CMNG-X) technique, first called the Adaptive RX algorithm, was developed early in the research and uses the mean and covariance of each class in mixtures based on the surroundings of the test pixel. Like the traditional RX, the algorithm assumes that the best background can be described by the neighbors of a test pixel. In an attempt to capture only the immediate surroundings of the test pixel while avoid the issues of underdetermined statistics, CMNG-X polls the classmap to figure out which classes are present around the test pixel and uses the well formed statistics of those classes as a matched filter background. This was inspired by techniques in adaptive covariance estimation, where covariance mixtures were used to overcome small sample sizes. Figure 3.25 depicts the CMNG-X selection of mean and covariance.

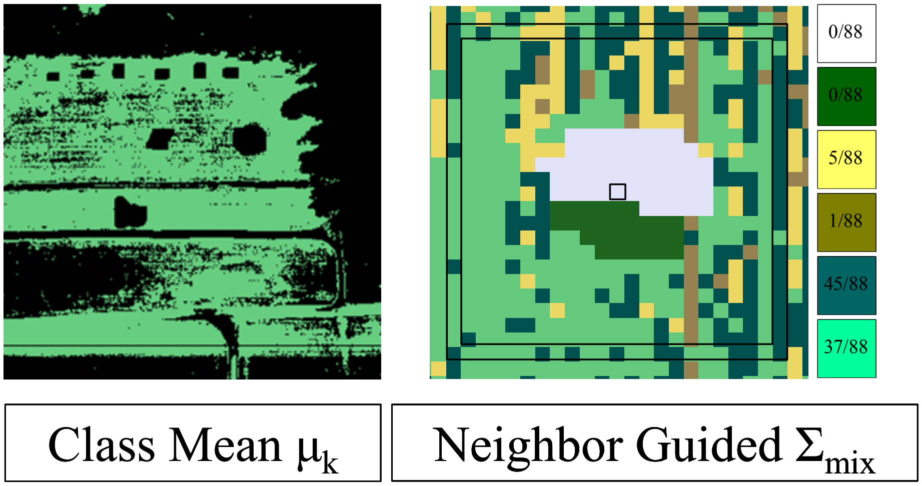


Figure 3.25 Class Mean Neighbor Guided - Mixed (CMNG-X) Pre-Clustering

Instead of selecting a single class from the results of the polling, the matched filter statistics are derived from the class statistics through a simple linear mixture by the equations

$$\Sigma = \sum_{i=1}^{k} \alpha_i \Sigma_i \qquad \mu = \sum_{i=1}^{k} \alpha_i \mu_i \qquad 3.18$$

where i is the class number and α_i is the fraction of the class present in the torus. Establishing the mathematical validity for linear mixtures of covariance is important because the mixture fraction is often squared in the literature. Both methods are appropriate under certain conditions, but the distinction is between mixing distributions within a pixel and mixing pixels from separate distributions. The linear mixing model provides the basis for a case of several endmembers present in a single pixel. For a two class example, a mixed pixel containing target and background can be expressed

$$x = \alpha d + (1-\alpha)B + n \qquad 3.19$$

where α is a the abundance of the target, d, the background is B, and n is the noise term. Assuming the noise is negligible and the data are mean centered, the covariance matrix for a set of mixed pixels would be

$$\Sigma = E\{(\alpha d + (1-\alpha)B)^T(\alpha d + (1-\alpha)B)\} \qquad 3.20$$

resulting in a squared α term. If, however, we have the covariance of several classes in the image, calculated from several pixel pools, combining these can be done by linear mixtures. For a two class example, given the covariance matrix of each class,

$$\Sigma_1 = E\{x_1^T x_1\} \text{ and } \Sigma_2 = E\{x_2^T x_2\} \qquad 3.21$$

and given the ratio of the number of pixels from each class included in the data pool, α, the combined covariance of the two pools can be approximate by

$$\hat{\Sigma}_{12} = \alpha\, E\{x_1^T x_1\} + (1-\alpha)\, E\{x_2^T x_2\} \qquad 3.22$$

which is essentially the same expression as equation 2.42 introduced in the adaptive covariance estimation section. Class means are pooled using the same expressions and mixture coefficients and applied in the GLRT expression.

Covariance estimation is necessary because of the instability of second order statistics from small data pools. The fact that class means do not suffer from the same difficulties with limited sample sizes leads to the local mean neighbor guided – mixed (LMNG-X) method uses the mean of pixels in a local window along with a mixture of covariance matrices. Using information from the neighboring pixels to mix statistics is unique from the literature. Figure 3.26 depicts the LMNG-X selection of mean and covariance.

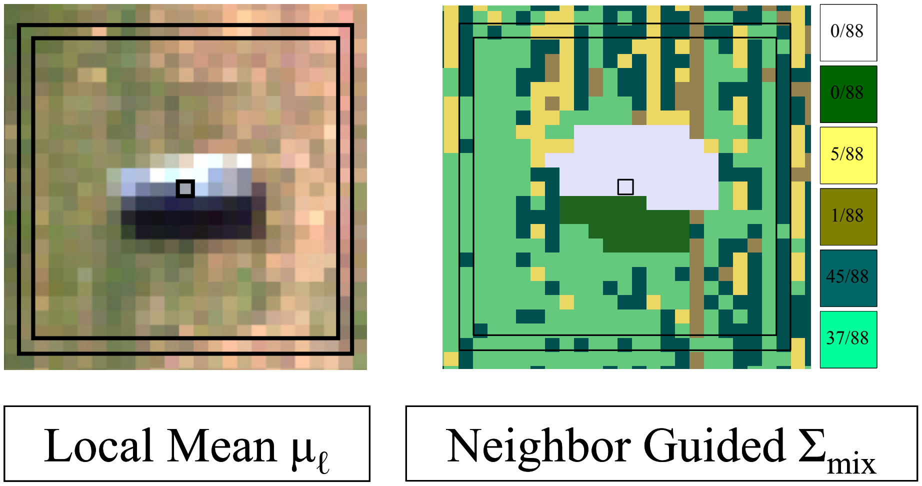


Figure 3.26 Local Mean Neighbor Guided Mixed (LMNG-X) Pre-Clustering

Both of the neighbor guided – mixed techniques incorporate all of the information obtained by identifying the classes present in the test pixel surroundings. This gives the mixed techniques a theoretical advantage in detecting targets that lay along land cover transitions, or are impacted by adjacency effects.

3.3.3 Spectral Subsetting Target Exclusion

Each of the eight methods of selecting mean and covariance for the matched filter make different assumptions about the target and background space. While it is true that each is able to employ spatial and spectral target exclusion methods and may be vulnerable to target contamination to a different extent, by perfectly excluding target and holding the contamination vulnerability variable constant the underlying assumptions of the algorithms may be tested. Perfect exclusion was employed for the main comparison experiment, but examination of the viability of target exclusion methods was also examined as a guide for the practical application of these methods.

To visualize the pre-filtering exclusion problem, Figure 3.27 shows an illustration of overlapping background and target distributions. Narrowing and separating these distributions is the ultimate goal of the detection problem. If the distributions were separable by pre-filtering alone, the problem would be solved and the target would be considered extremely high contrast. As shown in the target selection section, none of the test targets were found without false alarms using SAM alone. Spectral target exclusion involves choosing a threshold along the x-axis in Figure 3.27, and excluding all pixels above that threshold.

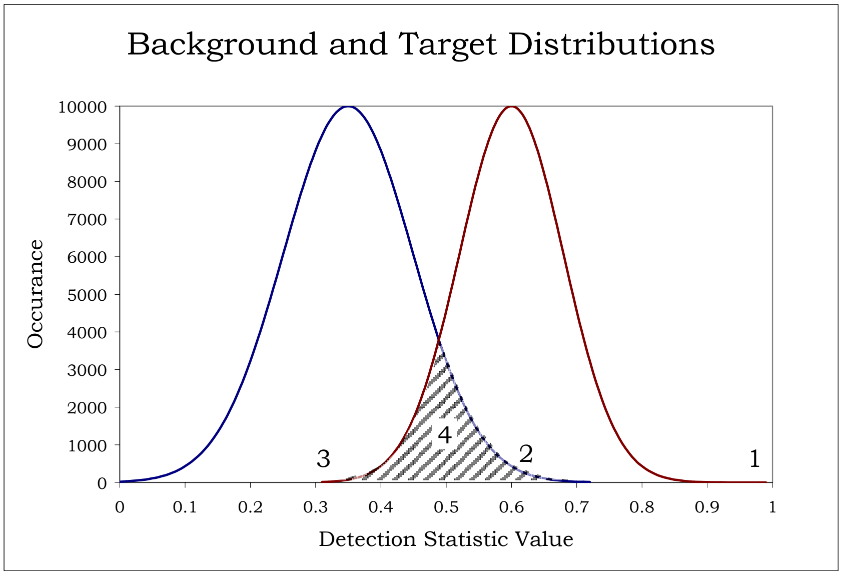


Figure 3.27 Example of Overlapping Background and Target Distributions

In the figure, the numeric labels denote the thresholds used in the experiment. The four cases include: (1) excluding no targets from the scene, (2) excluding half of the targets (detection rate of 0.5), (3) excluding all possible targets and as much background as needed (detection rate of 1.0), and (4) perfect exclusion using the truth map to exclude only target and no background in the shaded region. Obviously, in a real-world application, detection rates cannot be used to set a pre-filtering threshold. This experiment can, however, inform a user performing real-world detection as to the appropriate level of exclusion for a given problem. In practice, the level of exclusion depends on the purpose of detection (i.e. image screening, which accepts some false alarms for an overall higher detection rate, or automated target detection, which requires a low false alarm rate for some of the pixels on target). The overlap of the distributions is the critical factor in performing this type of exclusion, so the appropriate level of exclusion will vary for different targets and pre-screening techniques. Similar to the concept that was applied in matching the classification and detection discriminant functions for the target guided methods, another technique explored in this study was the use of target contaminated results to pre-filter detection. Auto exclusion is the process of iterating with a given technique by running detection without excluding targets, then thresholding those results to develop an exclusion map, and then running detection using that map. Comparing auto exclusion to SAM exclusion for various levels of contamination will demonstrate how different levels of distribution overlap (and different pixels overlapping) will impact pre-filtered detection.

Again, for the central part of this experiment, in order to remove the dependency on the ability of pre-filtering to detect a given target, the fourth method was employed. This detracts from the realism of the results, but adds to the consistency of measurement for comparison. The implementation of the methods for evaluating results in order to expose the strengths and weaknesses of all of the algorithms studied will be discussed in the next section.

3.4 Evaluation Metric Implementation

Two evaluation techniques were implemented to examine the various methods of background characterization described above. To scrutinize the data pool from which a given background was calculated, the Chi-Squared test for MVN was implemented without deviation from the method outlined in the literature. The rank ordered statistical (or Mahalanobis) distance of each pixel in the distribution was plotted against the expected value from the chi-squared distribution, and the goodness of fit was measured via equation 2.32.

To implement the formation of a ROC curve, a few simple modifications were made to tailor detection results for the experiment. As mentioned in the background section, the axes of the ROC curve plot were changed to detection rate versus false alarm rate. Points on the ROC curve were generated by counting the number of false alarms at each occurrence of a target pixel detect and dividing by the total number of background pixels. Due to the limited number of target pixels for certain targets, reporting a probability of detect and false alarm would be an over-generalization of the results.

In keeping with the rest of the experiment, all targets except the one being sought were discarded for ROC curve formation. This was a simple implementation given the numbering scheme of the truth maps (given in Table 3.1). Pixels on other targets were labeled as guard pixels in the truth maps, and thus automatically excluded from the ROC curve calculations. Again, this allowed for an isolation of each target in a natural background, and eliminated spatial dependencies in some of the results, but also eliminated the possibility of studying the target discrimination power of background characterization techniques.

Like the goodness of fit metric in the Chi-Squared MVN test, the average false alarm rate (AFAR) metric was used to reduce a curve to a single number. AFAR is an approximation of the area above the ROC curve and was calculated by averaging all of the values along the x-axis of the curve. The ability of AFAR to provide a comparison between algorithms is linked to the type of false alarms occurring in the ROC curve. To illustrate this, Figure 3.28 shows ROC curves for two different algorithms detecting the same target.

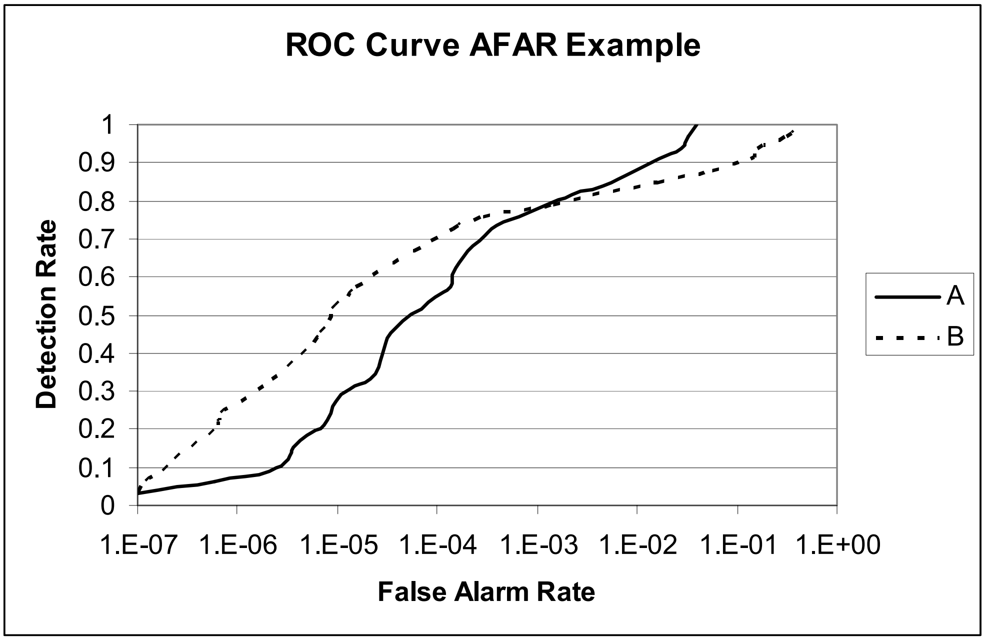


Figure 3.28 Example ROC Curve Illustrating the Need for a Partial AFAR

The AFAR for algorithm A is an order of magnitude less than for algorithm B, yet the ROC curve demonstrates that algorithm B is preferable for operation at false alarm rates lower than 1.E-03. The last few target pixels were more difficult for algorithm B, driving up the final AFAR value in spite of good performance at low false alarm rates. A partial AFAR, averaging false alarms only to a certain rate of detection, could be calculated to show better performance for algorithm B. In this case, selection between AFAR and partial AFAR would reverse the decision of which detector is best. Not unlike the decisions surrounding the level of target exclusion, evaluation of a detection result with partial AFAR requires a fundamental decision about the type of application in which the detector will be employed. Algorithm A would be preferable for applications where some target pixels needed to be identified with the fewest possible false alarms. As an example, an automated target detection system might require low false alarm rate operation with a tolerance for not locating every pixel on a given target. Likewise, algorithm B would be preferable for applications where some false alarms are acceptable but false negatives (i.e. missed targets) are not. An application allowing for a high false alarm rate might be searching through large data sets to flag potential targets for further scrutiny by an analyst. Selecting the latter application, and for consistency in results, the full AFAR was calculated for each detection result in this experiment. Another practical reason for using the full AFAR metric was that the majority of high contrast target pixels were detected without false alarms using any of the methods. In order to compare the methods, the most difficult pixels on these relatively easy targets needed to be included. The use of a partial AFAR would have changed the relative ranking of algorithms in detecting some of the low contrast targets, but would have had minimal impact on the observed trends overall.

The full AFAR included the false alarms detected in locating every pixel on a given target in the image. To compare the performance of several background characterization techniques in finding sub pixel and fully resolved targets, a separate ROC curve and AFAR metric were calculated for the detection of only those pixels identified in the truth maps as sub or full pixel targets. Full pixels targets were generally easier to find than other types, and sub pixels were generally more difficult. The AFAR result for the all pixels category incorporated false alarms from the detection of all types of target pixels and therefore was always the greatest value. In cases where the target contained glare or shadow pixels, the full and sub AFAR results did not combine to form the all pixels result. Included in Appendix A are ROC curves for the all pixels case along with AFAR results for the all, full, and sub pixel targets on one plot. While inspection of the ROC curve remains the most informative evaluation method, a complete picture of detection performance can be summarized with these three AFAR results. For this reason, only the summary AFAR charts are given in the results section.

3.5 Experimental Approach Summary

The goal of this experiment was to quantify the relative performance of several methods of stochastic background characterization. Given the wide variety of methods collected from the literature, the methods have been organized into two groups: those that rely on spatial information within the scene to select a background, and those that rely on spectral information in the cube to select a background. Two methods of spatial subsetting were implemented, allowing for the selection of a target approach region to serve as the background data, or the selection of RX sliding window sizes to surround the test pixel and calculate a background from the local window. The eight methods of spectral subsetting implemented allow for the use of a class or local mean, and selection of a method for guiding the decision of which class statistics to use. The target guided method uses the target spectrum to select the class closest in statistical distance from the target. This method requires a better match between classification and detection discriminant functions, necessitating the use of the statistical distance classifier. The pixel guided method uses the class identity of the test pixel to select which statistics to use. The neighbor guided method polls pixels around the test pixel and uses those class identities either to select the statistics of the most common class, or to mix statistics based on the ratios of classes present.

The ways in which each of these methods handles multivariate normality, target exclusion, and detection of low contrast targets are other factors considered in this study. The chi-squared test for MVN was implemented, and techniques in spatial and spectral target exclusion were explored to provide insight into the influence of target contamination on backgrounds from different characterization techniques. Finally, a low contrast method of mean subtraction was implemented to test detection of low contrast targets. To summarize the techniques used to this point, the thesis block diagram given in Figure 3.29 depicts all of the steps in the target detection chain. Figure 3.30 shows the categories of background characterization techniques considered in the study. Each box below the dotted line in Figure 3.29 represents a set of techniques which have been tested and compared during the course of this research.

With all of the implementation of the experiment covered, the next section will present and discuss the results of the experiment for each technique. Examples of multivariate normality, target influence, and low contrast detection will be presented along with the results to give the context for each of these considerations. Finally, all of the results will be compared to provide insight into the relative performance of the background characterization techniques.

3.6 Thesis Block Diagram

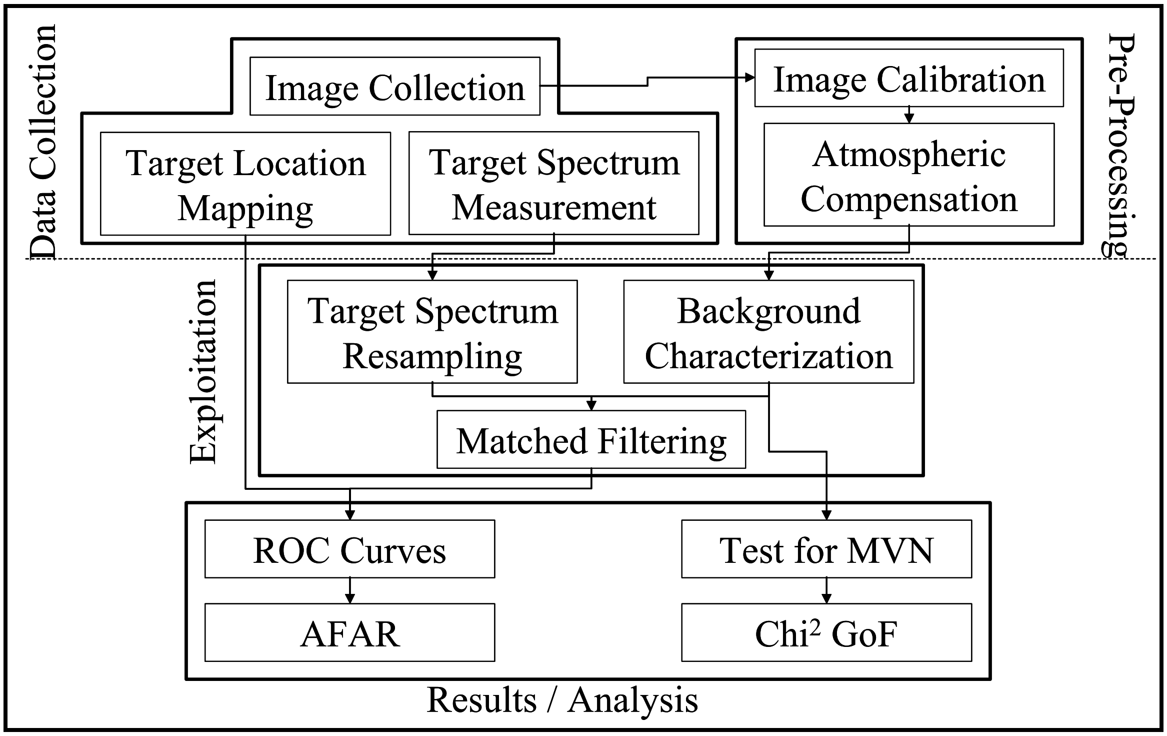


Figure 3.29 Target Detection Data Flow Chart

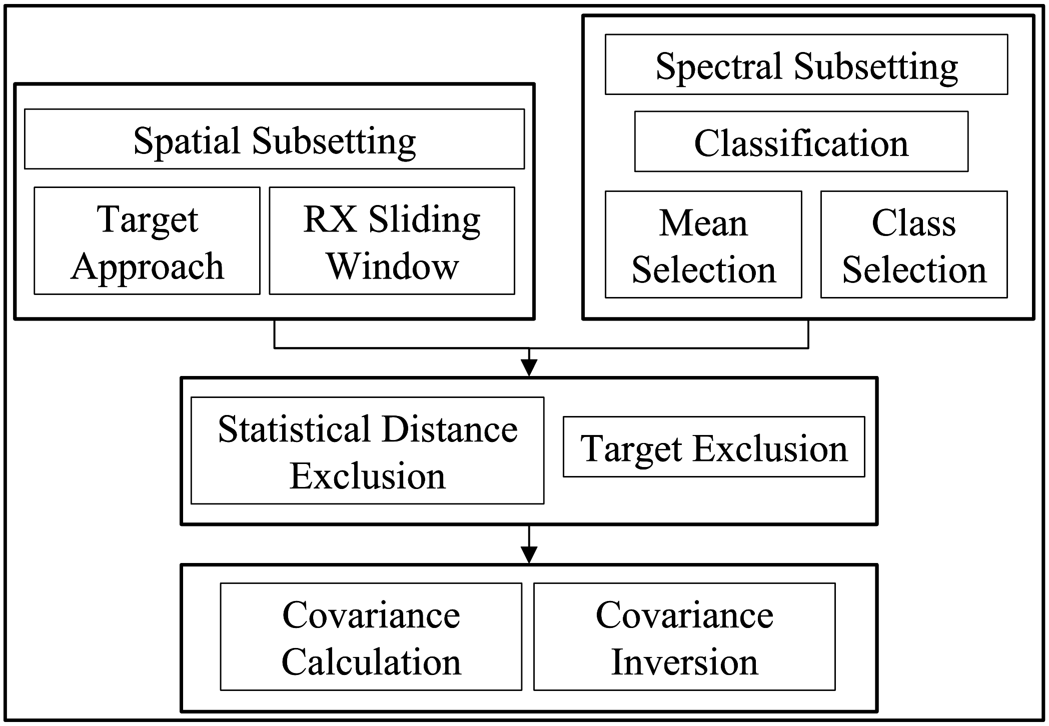


Figure 3.30 Background Characterization Data Flow Chart

4.0 Results and Discussion

Having an understanding of background characterization techniques in the current literature and given the issues, trade spaces, and considerations involved with implementing current and new algorithms, a series of experiments were conducted to test different background characterization techniques. Several techniques within the categories of spatial and spectral subsetting were tested for performance under ideal target exclusion conditions. The multivariate normality of the background data in each of these techniques was measured. The influence of target contamination was tested for each type of technique in order to inform the process of real-world detection. Finally, a few brief experiments demonstrated how statistical distance exclusion and low contrast detection (mean subtraction of the test pixel and not the target spectrum) are good subjects for future research.

The background characterization technique comparison portion of the experiment considered eighteen different backgrounds for eight different targets in each of two scenes. In an attempt to make the results concise, the eight targets were divided into two groups based on their relative difficulty of detection. Low contrast targets were those with an AFAR of greater than 1E-3 and high contrast targets were those with an AFAR of less than 1E-3. For both of the scenes, one target proved to be exceedingly low contrast and one target was very high contrast. To eliminate the influence of the inconsistent results from these targets, they were excluded from the low and high contrast averages.

To allow for comparison of performance in detecting full and sub pixel targets, the AFAR graphs provide the AFAR of all, full, and sub pixels on the target. These did not combine to form the all pixels results in every case because, from the construction of the truth maps, the "all pixels" category may have included shadow or glare target pixels depending on the target. In order to provide greater separation of the results, logarithmic scaling was used on the axis measuring false alarms. This stretched the x-axis and flattened ROC curves to aid in examination. The logarithmic scale also helped to separate AFAR bars, but it necessitated inversion of the graph, so it should be noted that higher is better for bars on the AFAR graphs. The results for all of the targets, including ROC curves and AFAR graphs are given on a target-by-target basis in Appendix A, grouped by characterization technique. The individual results of the extensive testing of pre-clustering target influence and MVN are also given in Appendix A. Many of the IDL coded implementations of the techniques used in this study are included in Appendix B for reference.

After considering detection results, MVN, and target influence for each of the background characterization techniques separately, the last section provides an overall comparison across methods in spatial and spectral subsetting. With all of these results, the discussion will point out adherence and deviation from the expected behavior and provide insight into the phenomenology driving some of the results. This comparative evaluation will lead to some conclusions about the practical application of these background characterization techniques.

4.1 Spatial Subsetting Results

Spatial subsetting was divided into two categories: the target approach method and the RX sliding window. Results of detection using scene-wide statistics are included with the former. ROC curves and AFAR graphs for each target individually are included in Appendix A.

4.1.1 Target Approach Method Detection Results

Results of detection with the target approach regions and the scene-wide statistics are shown in Figures 4.1 and 4.2 for the Forest and Desert Radiance images respectively. The first observation to note was that none of the target approach regions consistently outperformed the scene-wide statistics in the forest scene. This was not surprising considering the forest scene was relatively low in clutter content, especially after exclusion of the other targets in the scene to isolate the target being sought. In earlier stages of this research, it was noted that scene-wide statistics including all other targets did not provide this level of detection. The next observation was that the mixed background performed well for the high contrast forest scene targets and exceptionally well for all of the targets in the desert scene. This indicated that the species in the broader image were well suited to serve as background, but there was a detrimental impact of including all pixels in the background statistics. This was much more the case for the desert image than for the forest image. The underlying basis of this disparity, which will become clearer after examining the MVN of the data sets, caused this type of result to recur throughout the techniques studied.

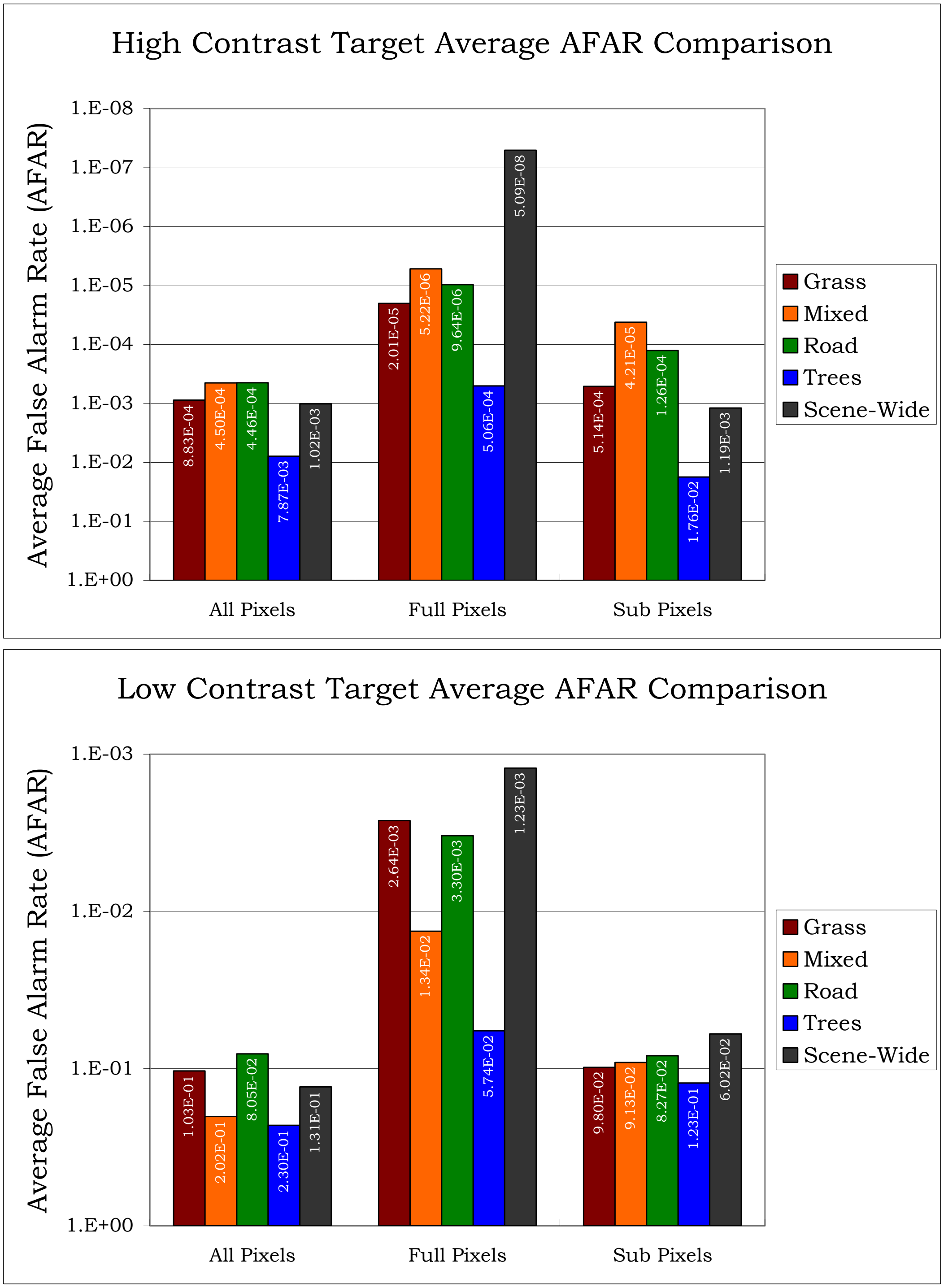


Figure 4.1 Run05 Target Approach Summary AFAR Plots

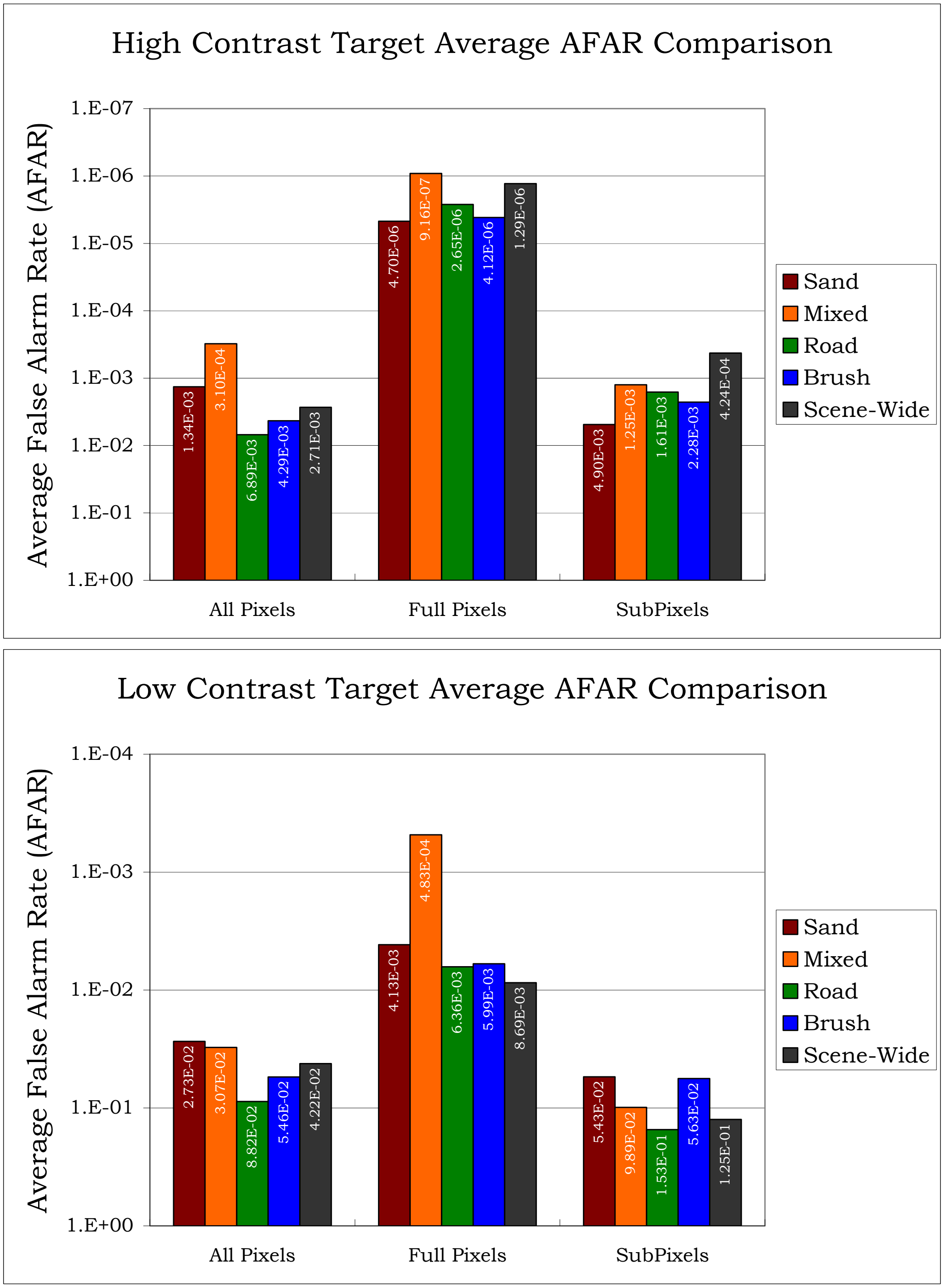


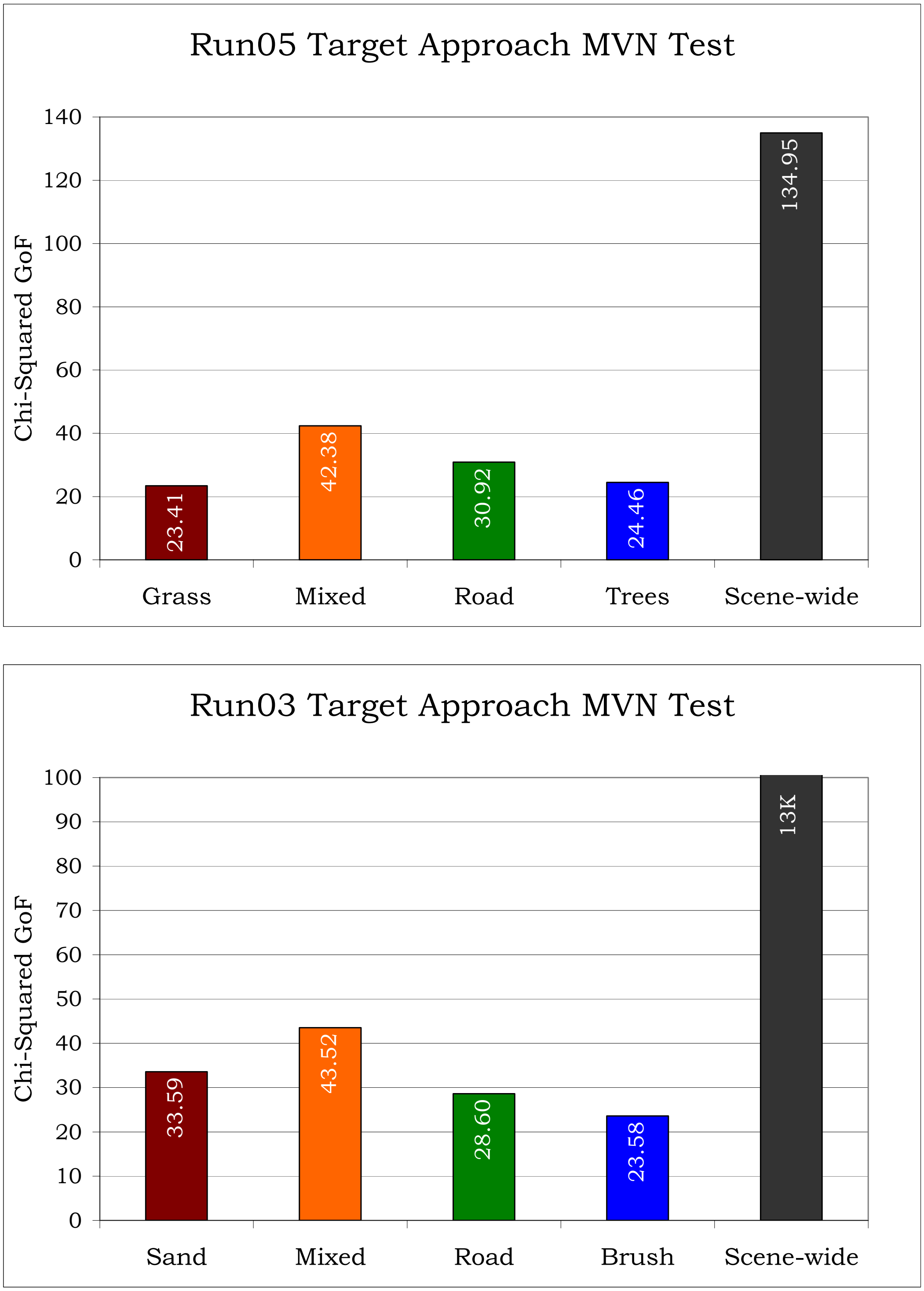
Figure 4.2 Run03 Target Approach Summary AFAR Plots

Given these results, the *a priori* selection of an appropriate target approach region to serve as background for a target not in the study set would be difficult. The measure of statistical distance between the target and each of the backgrounds cannot be used to guide selection of the best background. In line with statistical intuition, the full scene and mixed region backgrounds were respectively closest and second closest to each target. This was the case even when other backgrounds such as road or grass provided the best detection results. While the mixed region and scene-wide statistics served as the most consistent backgrounds overall (with the lowest averaged AFAR for full and sub pixel targets) the individual results in Appendix A reveal that each background in this section of the experiment -- with the exception of the forest scene trees region -- served as the best background for all of the pixels on at least one target. That being said, these results may have been influenced by the presence of shadow and glare pixels in the all pixels category. For full and sub pixel targets, either the mixed region or the scene-wide background provided the lowest AFAR for every target in the study. For reasons that will be explained later in this section and in the interest of avoiding simply ruling out the trees region in the forest scene as a viable background, the scene-wide statistics in the forest scene and the mixed region in the desert scene were identified as the best backgrounds for this technique. This decision could have been guided by *a priori* information about the image, but not by a measure of statistical distance (or any other target to background measurement). The next sections will discuss the rationale for this decision, as well as some of the factors to consider when applying the target approach method.

4.1.2 Target Approach Method MVN Test

The chi-squared test for multivariate normality was applied to each of the backgrounds, along with a goodness of fit test to measure the total normalized deviation from the expected line. Figure 4.3 shows the GoF for the target approach regions and scene-wide data for the forest and desert images. In both cases, regions of vegetation were the most multivariate normal, the mixed regions were relatively less multivariate normal, and the scene-wide data were decidedly non-multivariate normal. One significant observation was that the desert scene had a scene-wide GoF two orders of magnitude greater (worse) than the forest scene, while the target approach subsets had comparable measures of MVN. This was the most likely cause for the failure of the desert scene-wide statistics as a matched filter background. The cause of this poor MVN measure was the presence of sensor noise in the data. This level of noise, which was not present in the forest scene, exceeded the MVN sensitivity of the matched filter.

Referencing the detection results, the mixed background was a very strong performer in the desert scene, and yet it was not the most normal of the target approach regions. Combining this with the fact that the mixed and full scene data contained roughly the same species, it is clear that any background must be within a certain MVN threshold in order to avoid confounding the matched filter results. This observation relates to the selection of the best region for the realistic application of this method. Using an MVN test, it would have been possible to determine the normality of the full scene and then select a mixed subset, if the MVN metric exceeded a certain threshold. A third option


Run05 Target Approach MVN Test

Run03 Target Approach MVN Test

Figure 4.3 Run05 and Run03 Target Approach MVN Test

might be to employ statistical distance exclusion to improve the MVN of the data to meet the threshold.

The target approach method inherently assumed the absence of target species due to user knowledge of the scene. If this level of information was not available, or if the full image was used as a background, the presence of target species in the background would influence the result. The next section looks at how targets might influence the statistics of target approach regions.

4.1.3 Target Approach Method Target Influence

With truth maps available for the test data sets, the target approach regions were selected with absolute confidence that no target species existed in the background. This level of information may or may not be available in a real-world application of the method. To study the impact of target presence in a hand-picked region, target pixels from a target elsewhere in the scene were added one at a time to the grass target approach region. The statistics of the 18,000 pixel region were calculated with each addition and detection was performed on the image. So few samples in such a large data pool may not seem statistically significant and there may in fact be little change to the mean of the background due to target contamination. However, the matched filter's use of the covariance to suppress the band-to-band variation of the background makes the changes in the shape of the covariance matrix a more important factor. To get a sense for the shape of the covariance with the addition of each target pixel, the covariance matrices were decomposed with eigenvalue decomposition. After rank ordering the eigenvectors by their respective eigenvalues and therefore ordering them by statistical significance, the spectral angle between corresponding eigenvectors for each covariance was measured. As the shape of the covariance matrix changed, the spectral angle for corresponding eigenvectors increased. Figure 4.4 shows the results for four targets in the forest scene. Note that for the addition of each successive target pixel, the shape of the covariance changed in the same direction to a greater extent.

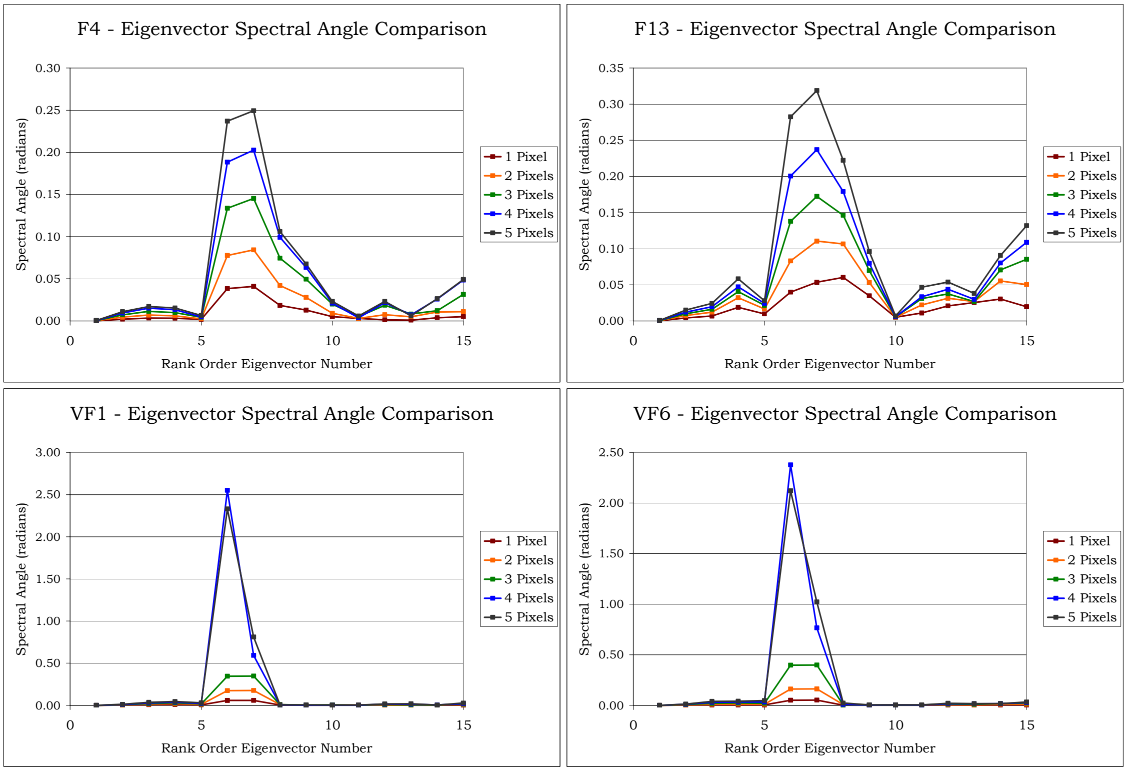


Figure 4.4 Target Influence Eigenvector Spectral Angle Comparison

Knowing that the covariance changed, the extent to which these changes impacted detection were observed by looking at the matched filter returns on the target and background. Figure 4.5 shows the maximum and minimum return from all pixels on the target for each of four targets along with the maximum and minimum background return. The maximum target return was the first detect and the maximum background return was the first false alarm. These plots represent the extrema of the target and background distributions in the detection statistic map. The maximum return on the target was always closer to (and for F4 even crossed) the maximum background return; the figure demonstrates that the addition of target pixels to the background affects greater overlap of the distributions.

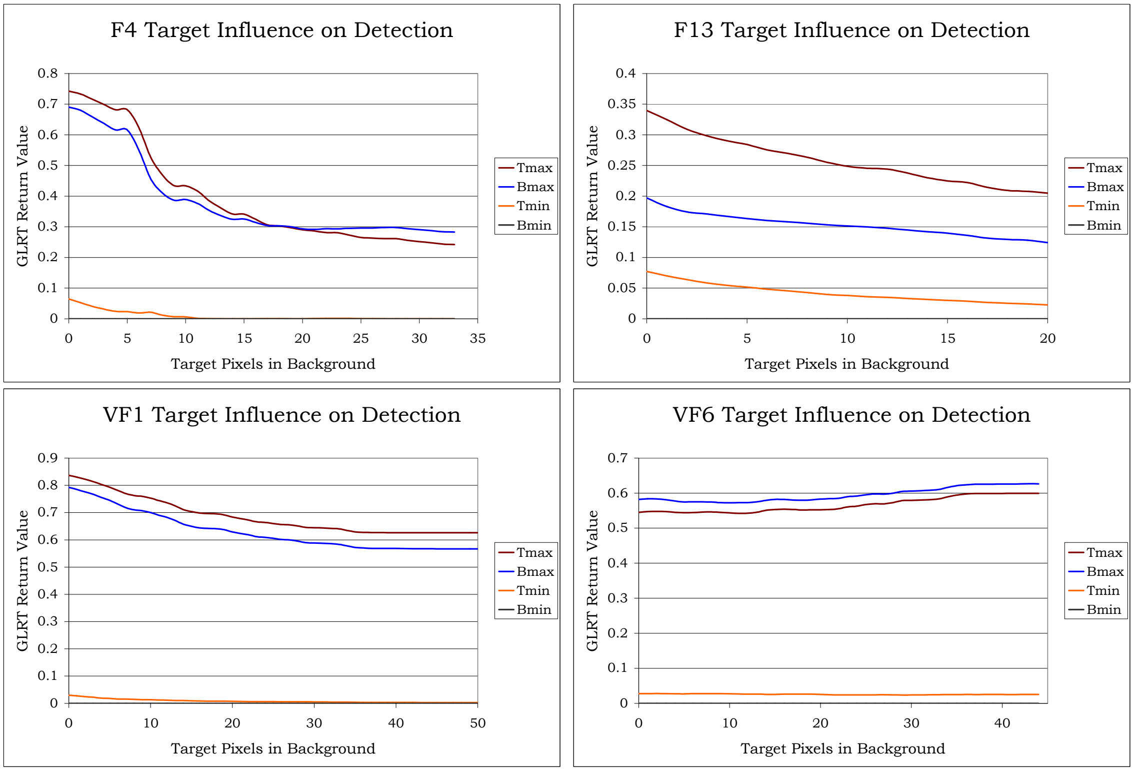


Figure 4.5 Target Influence Eigenvector Spectral Angle Comparison

In the target approach region method, the possibility of target contamination can be negated by some knowledge of general target location. However, matching the source of the interference causing false alarms (namely sub pixel mixing or target-like species) is left to the judgment of the analyst. The next section will provide the results of the RX sliding window, a technique which automates this process by making a few assumptions about the data and attempts to match the interference while providing greater multivariate normality and excluding target pixels from the background.

4.1.4 RX Sliding Window Detection Results

The variables in the RX sliding window experiment were the sizes of the four windows in Figure 3.13. The detection window was held at a single pixel in order to generate a unique background for each pixel in the scene. The exclusion window was set to twice the size of the target, assuming that a similar approximation could be made about the target in real-world applications. The size of the mean window was fixed at two pixels larger than the exclusion window to allow for a rapidly varying mean. These decisions allowed for a focus on various sizes of the covariance window.

Figures 4.6 and 4.7 show the averaged AFAR results for high and low contrast targets in the forest and desert scene respectively. The covariance window sizes were selected to be roughly one, five, and ten times the number of bands, as well as a stationary scene-wide window. Given the different sizes of the targets and the geometry of concentric square windows, the actual number of pixels in the background varied slightly among the targets. After running the 150 pixel case (one times the number of bands), it was noted that several of the targets experienced singularity due to the spectral exclusion of samples in the already data-starved window. Results for each of the individual targets, which are included in Appendix A, as well as the averaged results in the figures, are marked with an asterisk to denote the use of singular matrices where applicable. Ideal exclusion was used to remove the influence of targets on the

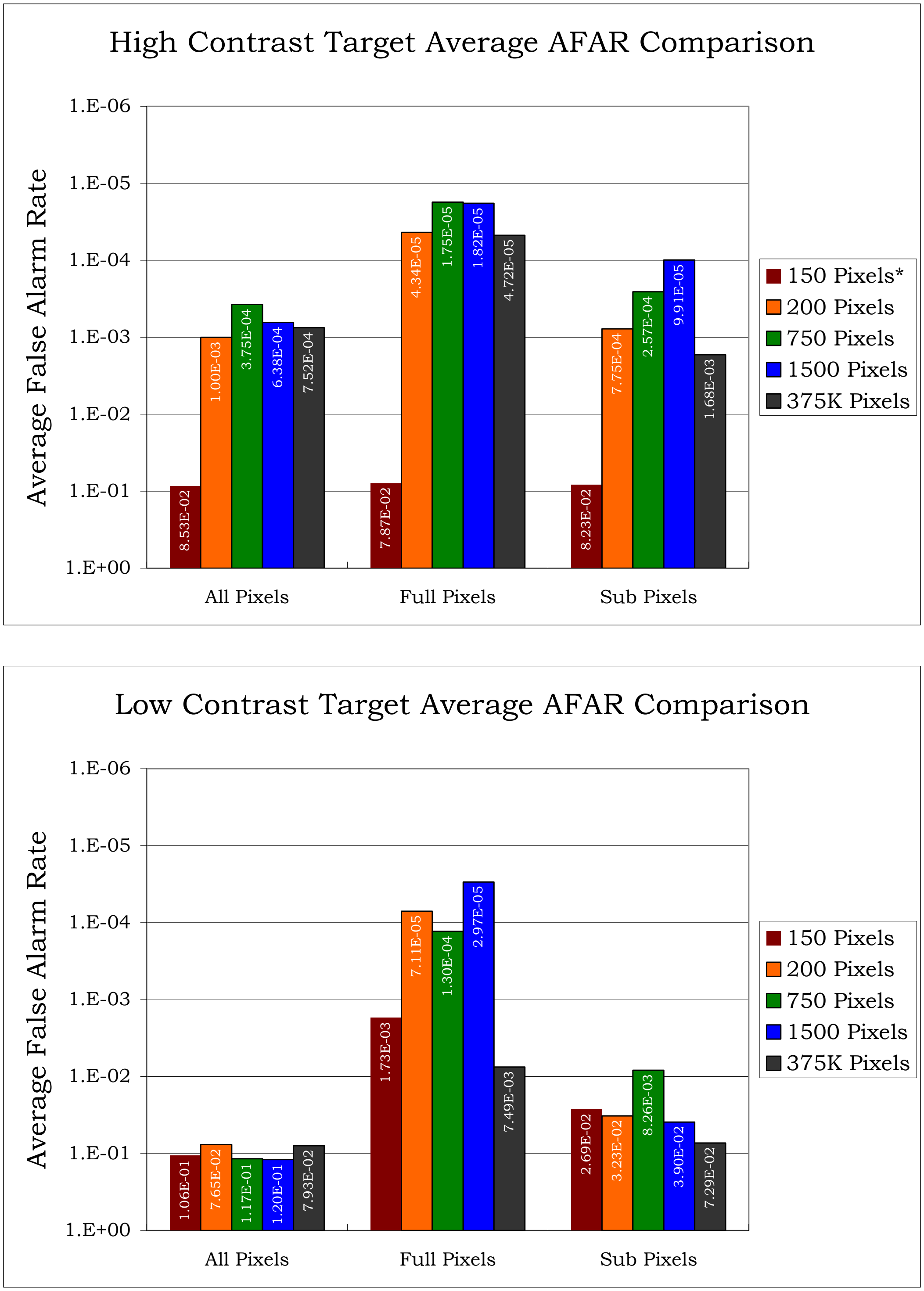


Figure 4.6 Run05 RX Sliding Window Summary AFAR Plots

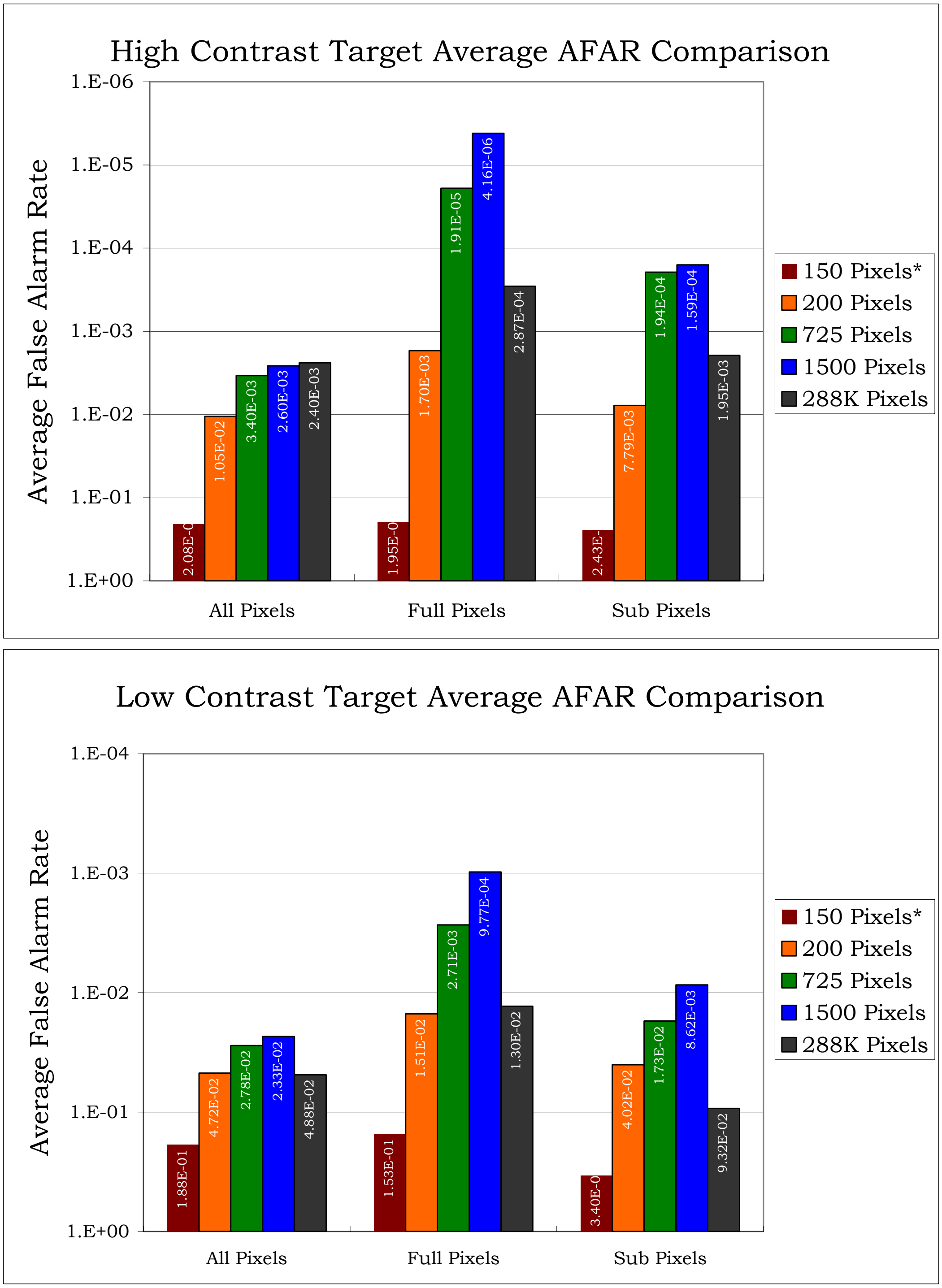


Figure 4.7 Run03 RX Sliding Window Summary AFAR Plots

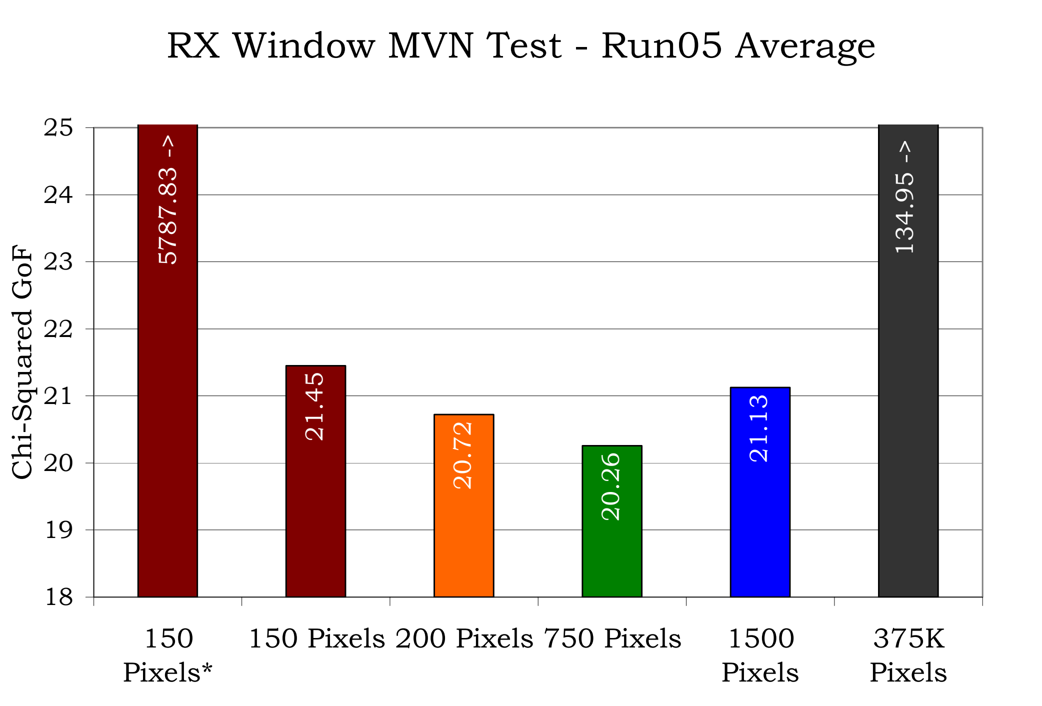
technique, but in these cases the exclusion caused increased instability in the covariance. As an additional source of disparity, this singularity triggered inversion by the Moore-Penrose pseudo inverse in the algorithm. While the pseudo-inversion technique was shown to provide the least error for singular matrices, the different inversion techniques resulted in very different returns for pixels within the same image. The 200 pixel case was run to replace the 150 pixel case and put this window size on equal footing with the other window sizes. From the averaged results, one observation is the dramatic impact of singularity on the 150 pixel backgrounds. The comparable performance of the low contrast forest scene targets, which did not experience singular backgrounds, indicated that it was singularity and not window size which caused the poor results. This can be seen on a target-by-target basis in the results in Appendix A.

Across all of the results, it was clear that the 1500 pixel backgrounds perform well, especially in the desert scene. For this technique, the 1500 pixel background was chosen as the representative top performer. The results of detection with 1500 pixel backgrounds reinforce the rule of thumb saying ten times the number of bands is required to generate stable statistics. There does, however, seem to be a weak trend toward improved performance using smaller window sizes for the low contrast targets and the "all pixels" category, especially in the forest scene. There is likely a relationship between the noise quality of the data and the extent to which the lower limits of sample size can be approached. For smaller window sizes, false alarms may occur where test pixels are proximal to noise pixels. Just as the background data may be starved by spectral exclusion, smaller windows sizes make the statistics more vulnerable to outliers like noise pixels.

Image noise is just another factor which adds to the trade-space for the window size decision. Along with the noise, and the desire to capture the local variability surrounding the test pixel, the multivariate normality of different sized backgrounds and the influence of target contamination are other factors to consider. The next sections will give examples of these and discuss how they might influence the window size decision.

4.1.5 RX Sliding Window MVN Test

To test the multivariate normality of the backgrounds for each target, data from the various sized covariance windows for the central target pixel on each target were measured with the chi-squared MVN test and goodness-of-fit metric. The results for each target individually are given in Appendix A, and the average GoF value from these is given in Figure 4.8, where the asterisk denotes a GoF with singular backgrounds Comparing these results to the detection results from Figures 4.6 and 4.7, the effect that extreme non multivariate normal backgrounds have on detection is clear. There is, however, no correlation between MVN and detection results for the relatively normal background sizes. This reinforced the notion that once a certain MVN threshold has been reached, small improvements between relatively normal backgrounds will be overshadowed by other phenomenology. For instance, note that the smaller backgrounds are more normal in the desert scene than in the forest scene. Disparate class means and noise in the desert image has driven



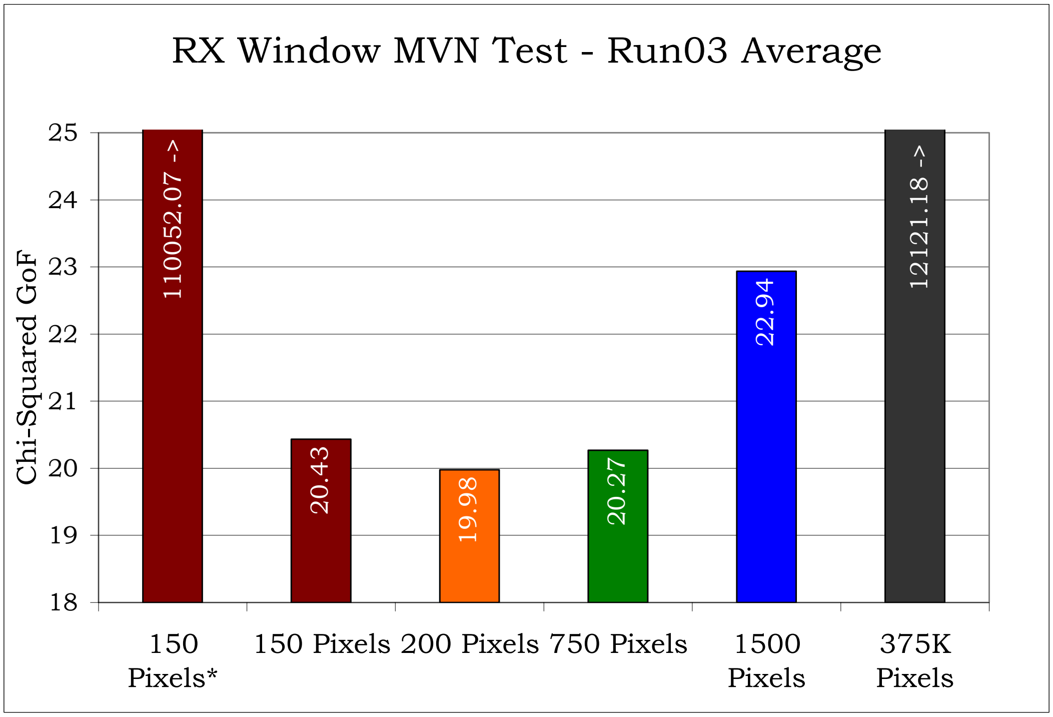


Figure 4.8 RX Sliding Window Chi-Squared GoF

up the scene-wide GoF, even though local variability is more normally distributed.

An additional observation was that for both scenes, the most normal backgrounds contained fewer samples than ten times the number of bands. This indicated that the minimization of skew mentioned in implementations of RX in the literature would not provide optimal detection. Once the background window was normal enough, other contributing factors played a larger role in determining detection success. Image noise was obviously one such factor. Target contamination is another important factor which is discussed in the next section.

4.1.6 RX Sliding Window Target Influence

Testing the influence of target contamination on detection using a sliding window involved controlling a number of variables. One controlled method of measuring target contamination is the implanting of target pixels into the background, but this was already accomplished in the target approach region contamination demonstration. In order to demonstrate a more realistic case, a target was selected from the scene which consisted of three spatially separated regions of target pixels located in close proximity.

Five different cases were run to demonstrate the utility in the spatial and spectral exclusion techniques used in this implementation of RX. First, correctly selected window sizes were used to produce a well formed background. Next, a self-contaminated case was run with the exclusion window set too small. Then, a cross-contaminated case was run with the covariance window set too large. Finally, the self- and cross-contaminated cases were run with spectral exclusion. A visual representation of these cases is given in Figure 4.9.

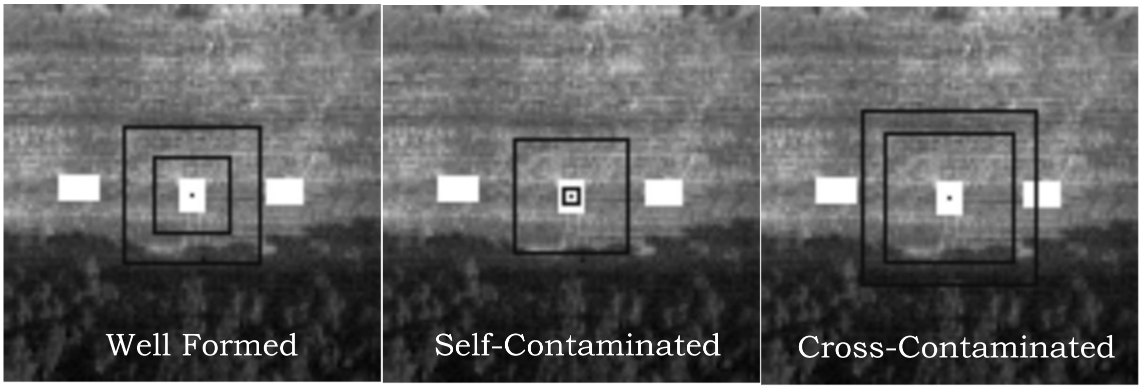


Figure 4.9 RX Well Formed, Self- and Cross-Contaminated Window Sizes

The spectral exclusion used SAM, thresholded to eliminate all of the target pixels in the scene and marked SAMX3 per the convention established in section 3.3.3. As previously shown, the number of samples in the background can have a large impact on detection; so, an attempt was made to hold sample size fixed. Due to the geometry of the scene, the well formed background contained 728 pixels, the self- and cross-contaminated cases contained 936 and 984 pixels respectively, and the spectral exclusion cases had backgrounds of fewer pixels to a varying degree depending on window location. The results of these cases, given in Figure 4.10, show the detrimental impact of contamination and the effectiveness of spectral exclusion in a realistic detection scenario. There were no exclusion regions in the image that caused singularity in this case, but that is another factor to consider when performing spectral exclusion.

There are many factors influencing the window size decision for this technique of background characterization. The window needs to be small

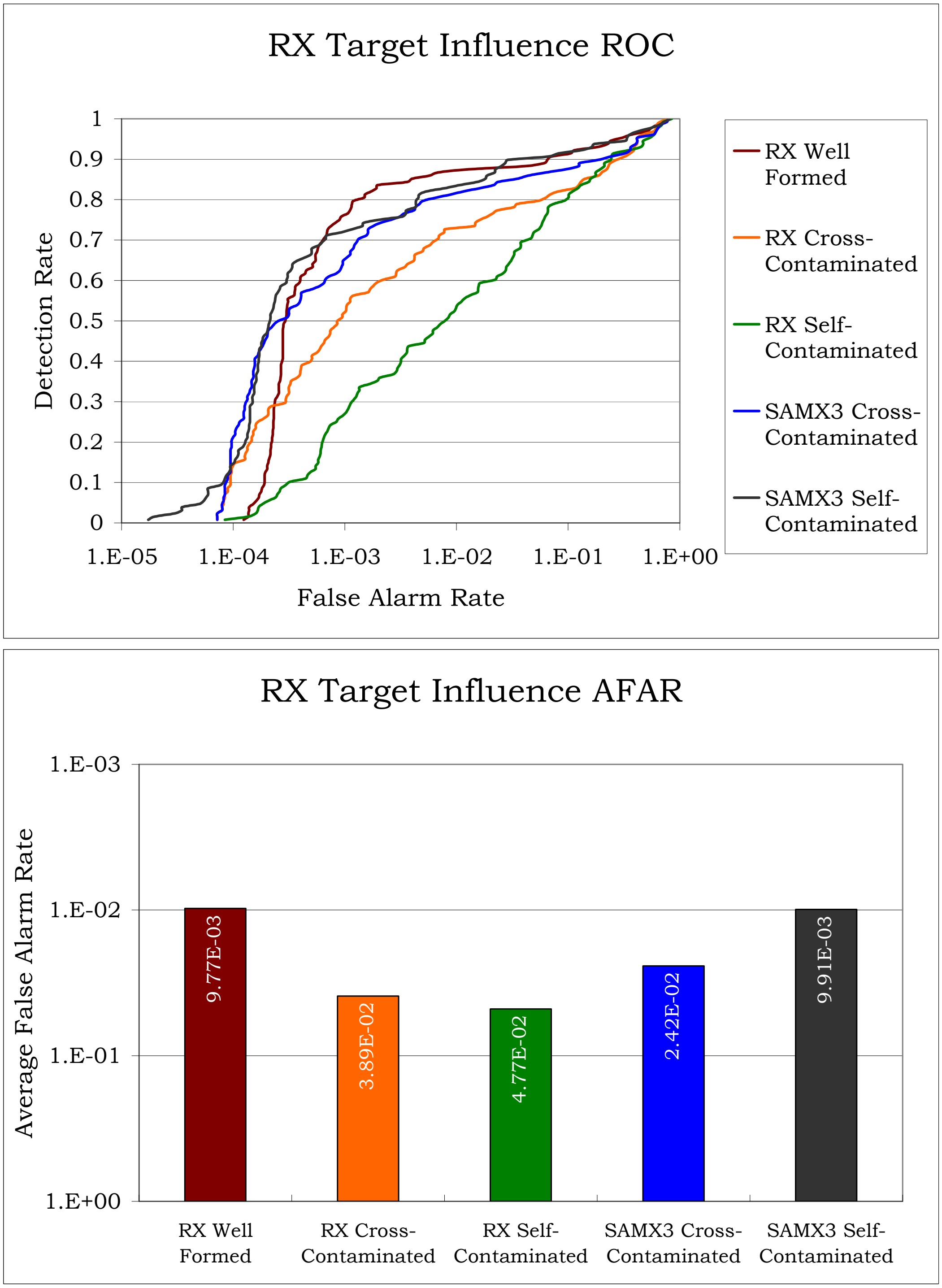


Figure 4.10 RX Sliding Window Target Influence Example

enough to capture the immediate surroundings of the test pixel, but large enough to provide stable statistics. Given certain information, like GSD and image noise characteristics, the window sizes can be adjusted to avoid self-contamination or to lessen the influence of noise pixels. Information about the number and proximity of targets in the scene would guide the use of spectral exclusion. For instance, if the analyst was certain there was only one target in the scene, spectral exclusion would not be used in order to avoid possible problems with singularity. If more than one target may be present, or if the exact size of the target is not known, the selection of window sizes is further complicated. Smaller exclusion windows risk self-contamination, but larger exclusion windows discard the valuable data spatially closest to the target. The probability of including neighboring targets increases with larger covariance window sizes, but the influence those target pixels may have on the background statistics decreases as window size increases. However, it is clear from this experiment that spectral exclusion can help ease these restrictions on the size of the exclusion and covariance windows.

To further explore the potential benefits of drawing spectral distinctions between pixels in the background, the next section will discuss the results of the collection of methods using spectral subsetting to improve background characterization for the matched filter.

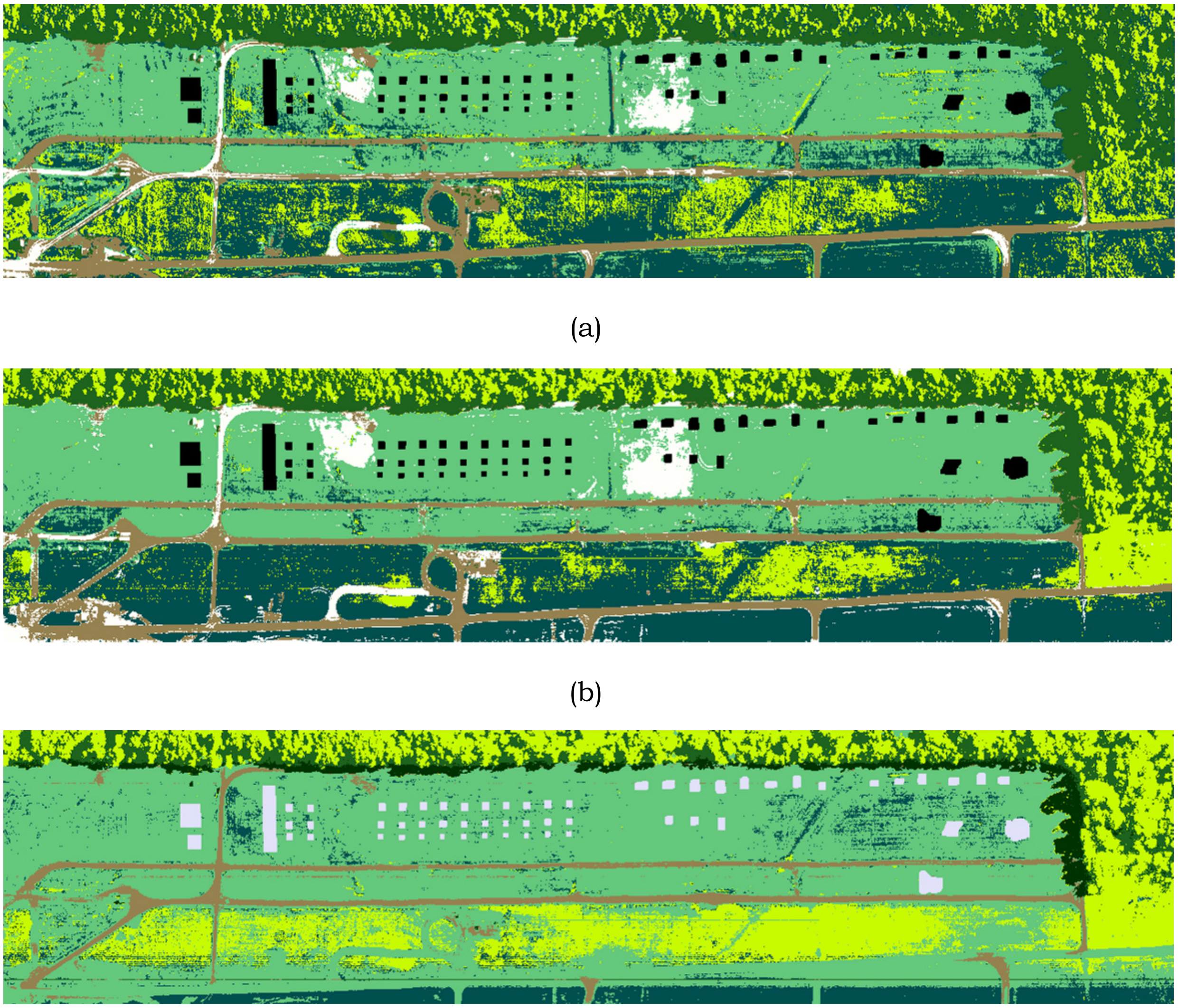
4.2 Spectral Subsetting Results

The spectral subsetting of data to improve detection involves the selection of a classification technique, the number of classes, methods to improve class statistics and exclude targets, and most importantly, the manner in which the class statistics will be used. In this section, the results of classification are discussed along with detection results using individual classes from the K-Means and SEM algorithms as backgrounds. The multivariate normality of the classes determined by each classification method are provided for comparison. The main results of the pre-clustering study are the results of detection using the eight different methods outlined in section 3.3. The results of an experiment testing several methods of spectral target exclusion are shown and results of statistical distance exclusion and low contrast target detection are provided and discussed.

4.2.1 Classification Results

The results of K-Means, SEM, and the statistical distance classification (SDC) modification of SEM are depicted in Figures 4.11 and 4.12 for the forest and desert scene respectively. As previously mentioned, the number of classes in each scene was driven by the Canonic data set. The structure within the K-Means and SEM results are very similar for both scenes. One difference between the two is that K-Means is more spatially cluttered, while SEM provides more homogeneous regions. SEM was more susceptible to sensor artifacts, as seen by the strait lines of pixels in the light green and white classes in Figures 4.11 (b) and 4.12 (b).

Class assignments were labeled according to the apparent relation to ground cover. For the forest scene, the bright and dark green classes are labeled light and dark trees, the light and dark blue-green classes are light and dark grass, and the brown class is road. The K-Means and SEM results contain a bright soil class in white and the SDC contains a shadow class in black. For the desert scene, the red class is labeled road, the green represents scrub brush, and the white, yellow, and brown represent light, medium, and dark sand. For these example images, all of the targets are masked out in black (or white for the forest scene SDC result). For the images in the experiment, classification was run with a given set of target pixels included in order to simulate a natural scene containing only one target.

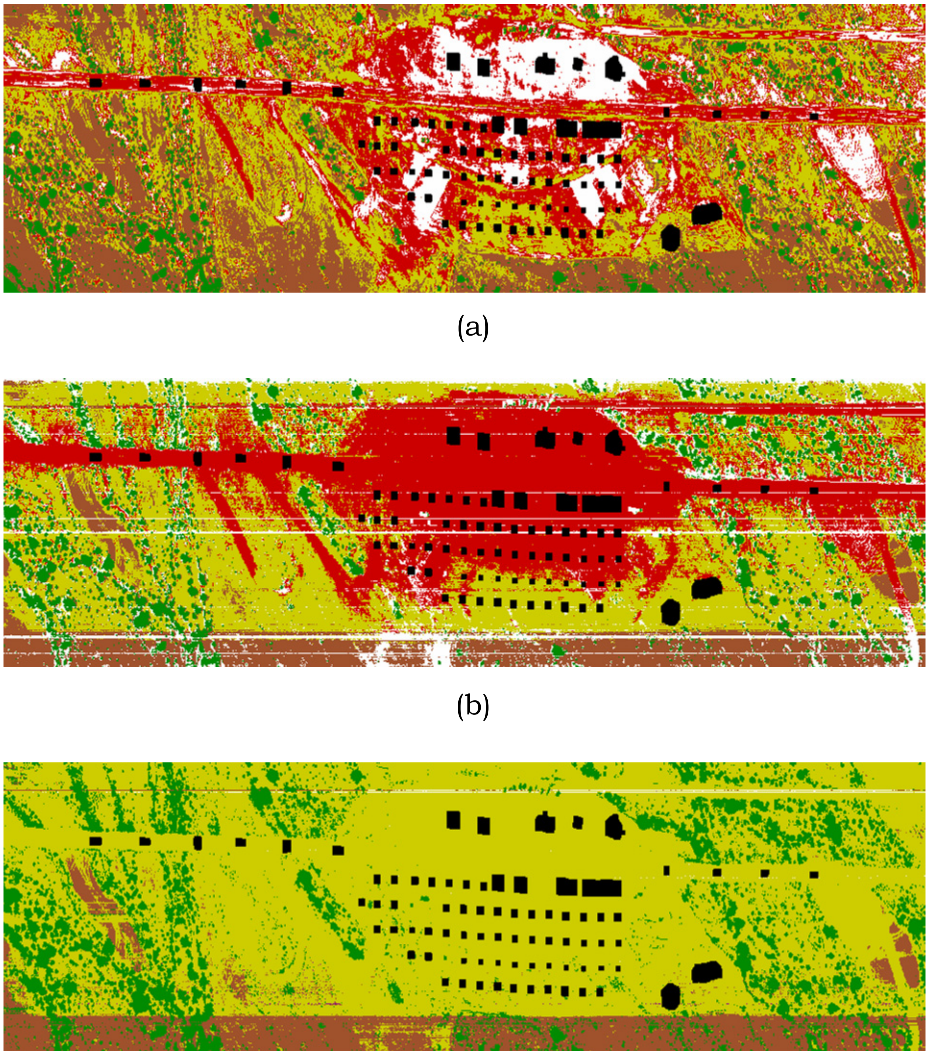


(a)

(b)

(c)

Figure 4.11 Run05 Classmaps Generated by (a) K-Means, (b) SEM and (c) SDC



(a)

(b)

(c)

Figure 4.12 Run03 Classmaps Generated by (a) K-Means, (b) SEM and (c) SDC

The results of SDC differ significantly from the other algorithms, especially for the desert scene. SDC showed a preference for some classes over others in both scenes, and combined several classes in the desert scene. In the forest scene, SDC created a shadow class and combined the bright soil class with the light grass class. For the desert scene, SEM combined the road and light sand classes and created a bright artifact class, and then SDC combined the road, light sand, and medium sand classes and created a bright and dark artifact class. The purpose of the SDC technique was to enable prediction of the best background class by using a similar classification discriminant function. The drawback of losing a good overall classification result is negated by the fact that all classes except the closest were discarded. This established the limited utility of the SDC technique, which can obviously only be used in the target guided methods.

In order to narrow the scope of the examination by eliminating one classification method, the next section provides a comparison between the K-Means and SEM algorithms for pre-clustering, including detection results and a multivariate normality test performed using all of the classmaps from the study.

4.2.2 K-Means versus SEM and Pre-Clustering MVN Comparison

To avoid running the entire experiment with both the K-Means and SEM algorithms, a simple method of comparing the potential performance of these algorithms in generating good matched filter backgrounds was performed through examination of detection results using each class independently as a background. Targets VF1 and V1 were selected as the respective example targets for the forest and desert scenes. Figures 4.13 and 4.14 give the ROC and AFAR results for detection of these targets using backgrounds from K-Means and SEM classification. The results are only slightly in favor of SEM, with two wins, one loss, and three apparent ties in the forest scene, and three wins and two losses in the desert scene. Poor performance of a single class would be a detriment to the target guided method. However, the fact that the target guided method uses SDC -- not SEM or K-Means -- allays concern for these specific classes in the selection of the generally better technique. Another contributing factor in this decision was the multivariate normality of each of the classes.

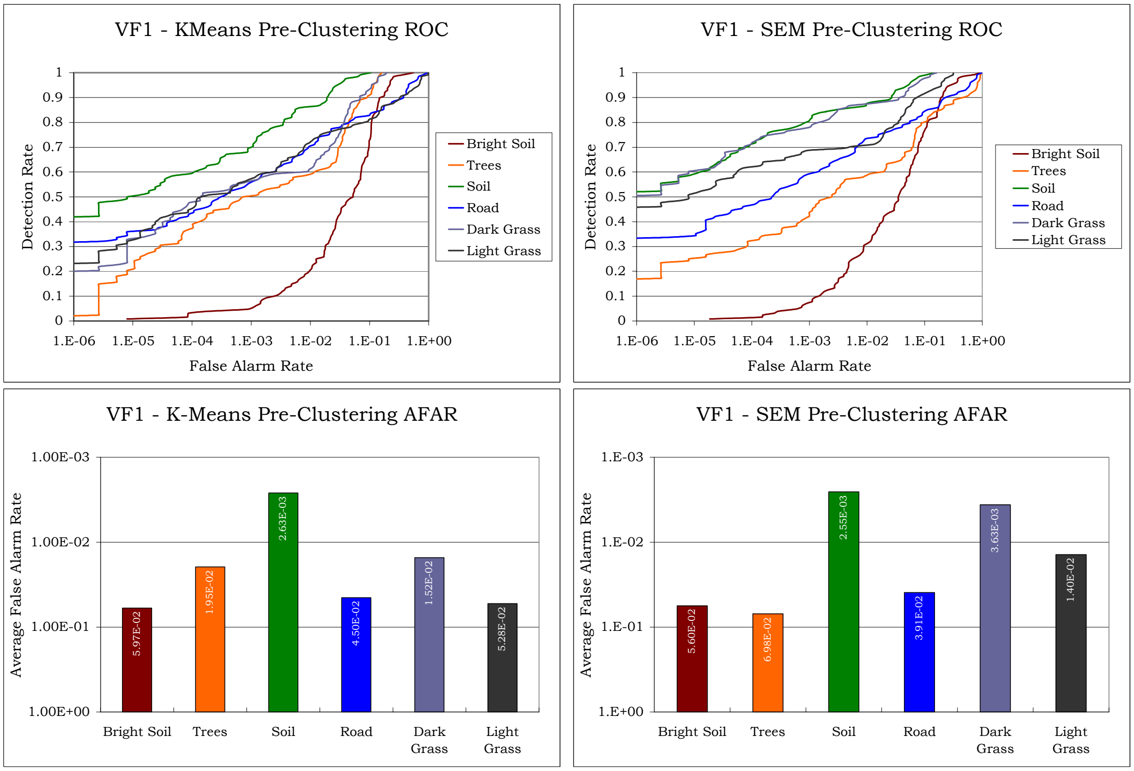


Figure 4.13 Run05 K-Means versus SEM Pre-Clustering Comparison

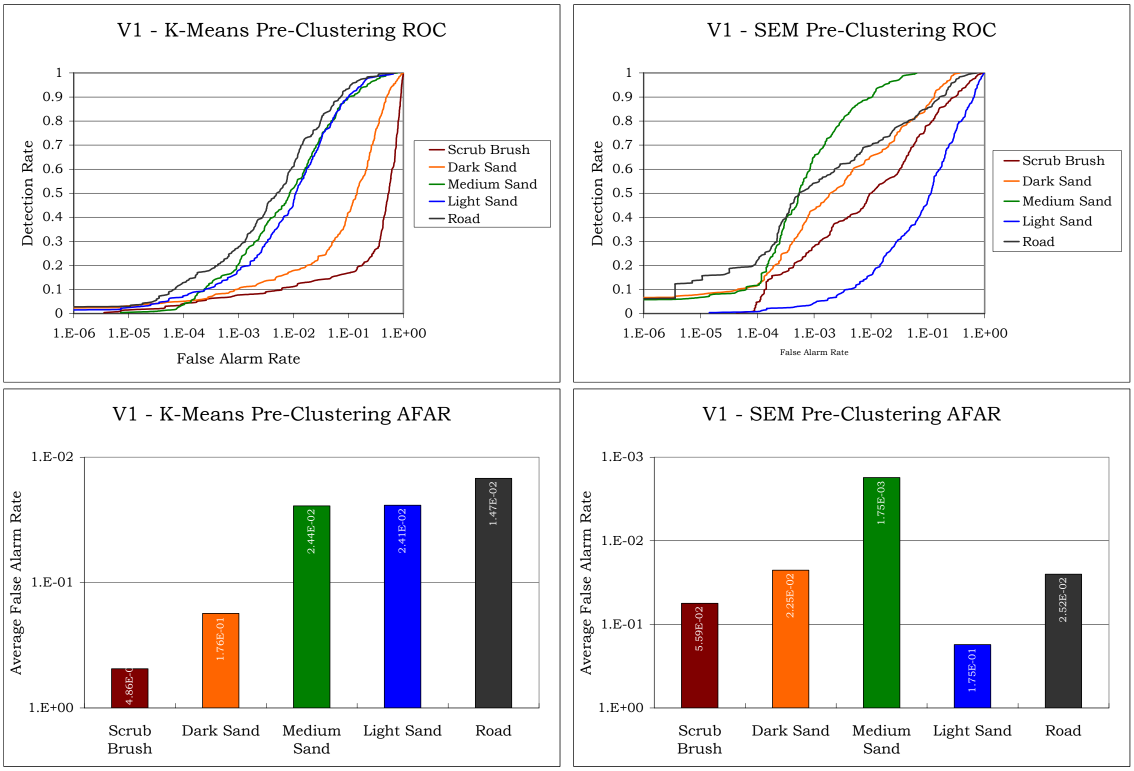
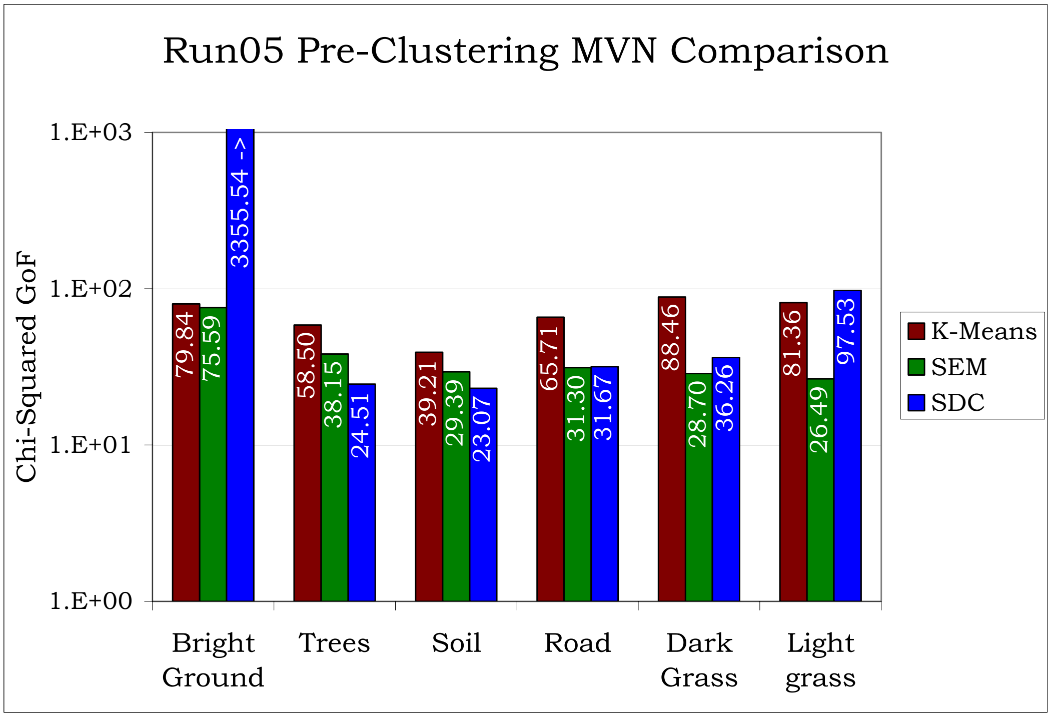


Figure 4.14 Run03 K-Means versus SEM Pre-Clustering Comparison

The MVN GoF metric for the data in each of the classes generated by K-Means, SEM, and SDC are included in Figure 4.15. Class assignments do not necessarily compare for the SDC technique, but the figures give a sense for the distribution of species within the classmaps. The MVN improvement of SEM over K-Means is clear, with only one K-Means class (the light sand) measuring more normal that its SEM counterpart. Incidentally, that light sand class is the K-Means background that most dramatically outperformed SEM in the detection results. The SDC MVN shows some improvement over SEM for many of the classes, but in both cases SDC resulted in a class that was extremely non multivariate normal. The only two classes actually used in the target guided method were the SDC light grass and scrub brush classes. Both of these



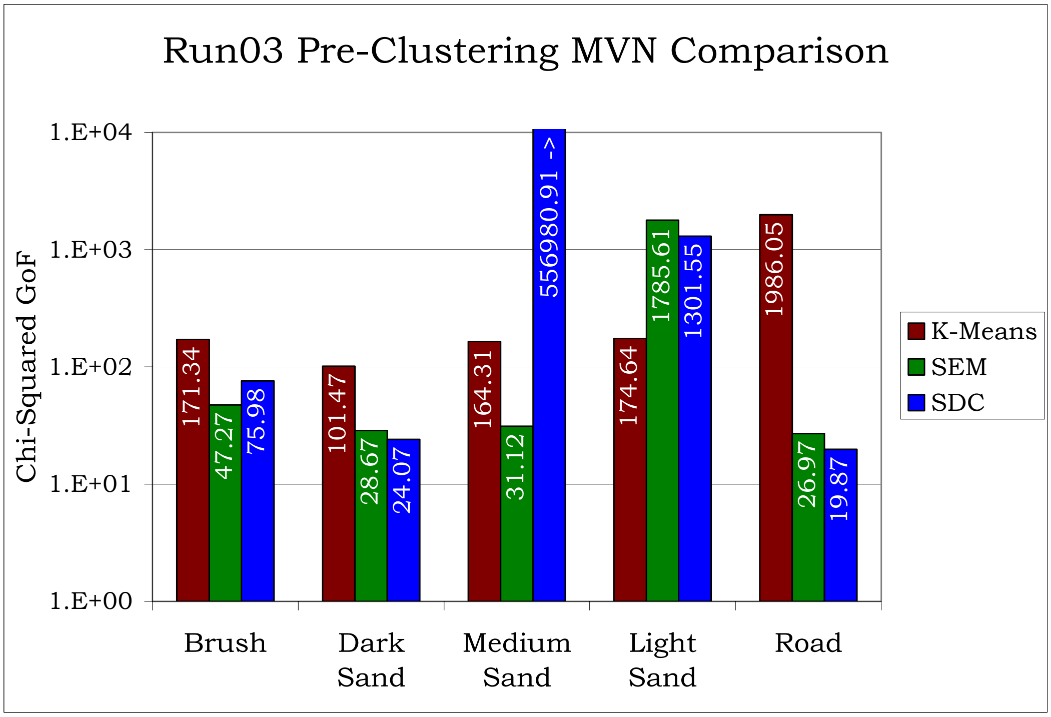


Figure 4.15 Run05 and Run03 Pre-Clustering MVN

classes were less multivariate normal than their SEM counterparts, but the class providing the best detection performance was predictable by statistical distance from the target spectrum.

Based on the marginal improvement of detection and the overall improvement in MVN, the SEM technique was used for the remainder of the experiment. The extent to which SEM provides improvement when used for pre-clustering is dependant on the data. For this data, the improvement in MVN did not translate directly into a lower false alarm rate; a more multivariate normal background provided a theoretical advantage in this application.

4.2.3 Pre-Clustering Detection Results

The eight techniques for pre-clustering detection were applied to the detection of the targets in the test data. As with other results, the individual ROC curves and AFAR graphs are in Appendix A. Figures 4.16 and 4.17 are the summary of the averaged AFAR for the three high contrast and the three low contrast targets for each scene. A number of observations were made from these results which gave insight into the performance of these background characterization techniques for different targets and backgrounds.

The first and most significant observation was the good performance of the class mean neighbor guided – mode (CMNG-M) technique. This method clearly dominated detection in the desert scene and performed well in detection of low contrast targets in the forest scene. This was an important finding considering that the detection of low contrast targets is of more concern to the research community. For the high contrast forest scene targets, the CMNG-M

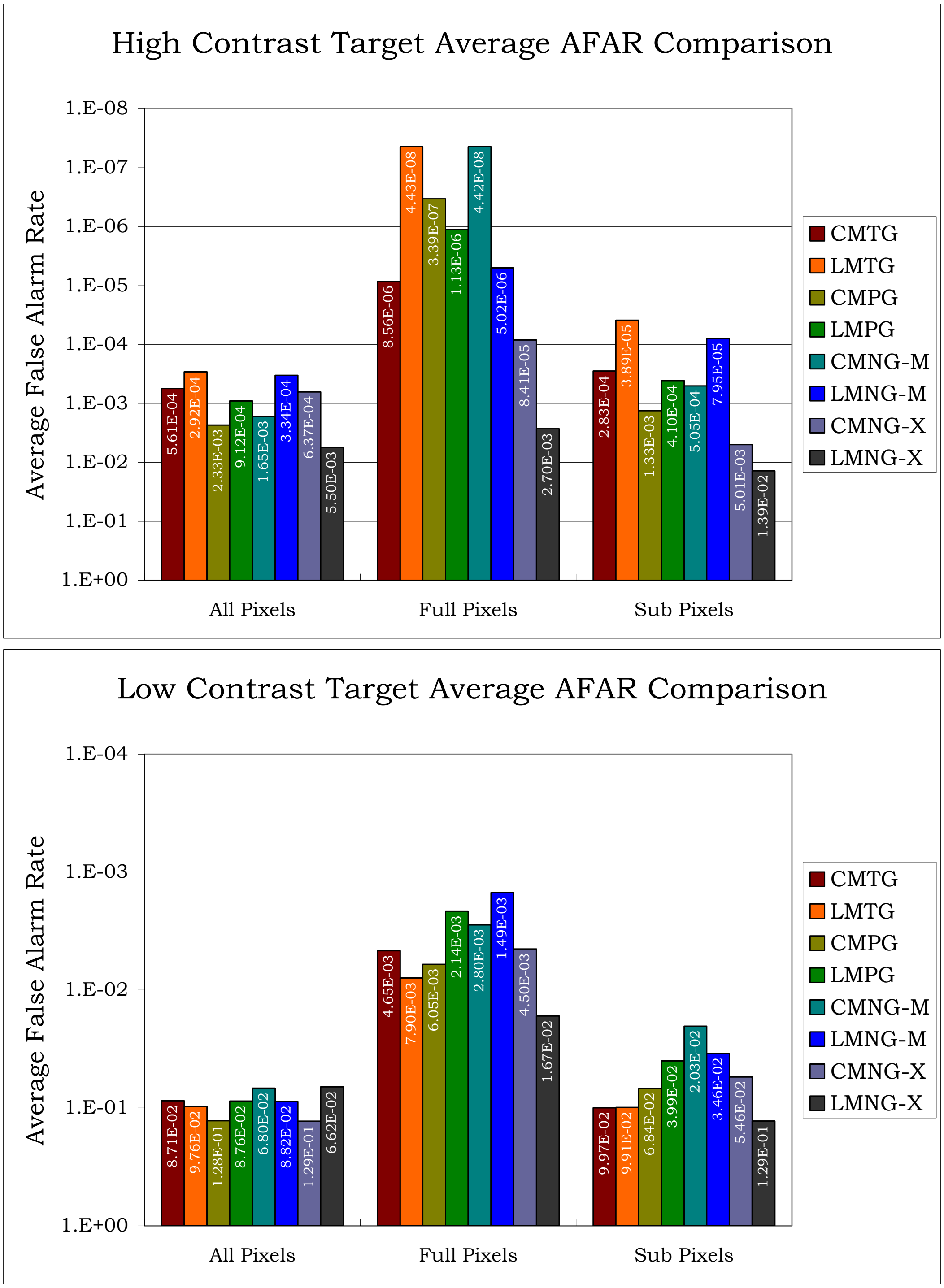


Figure 4.16 Run05 Pre-Clustering Summary AFAR Plots

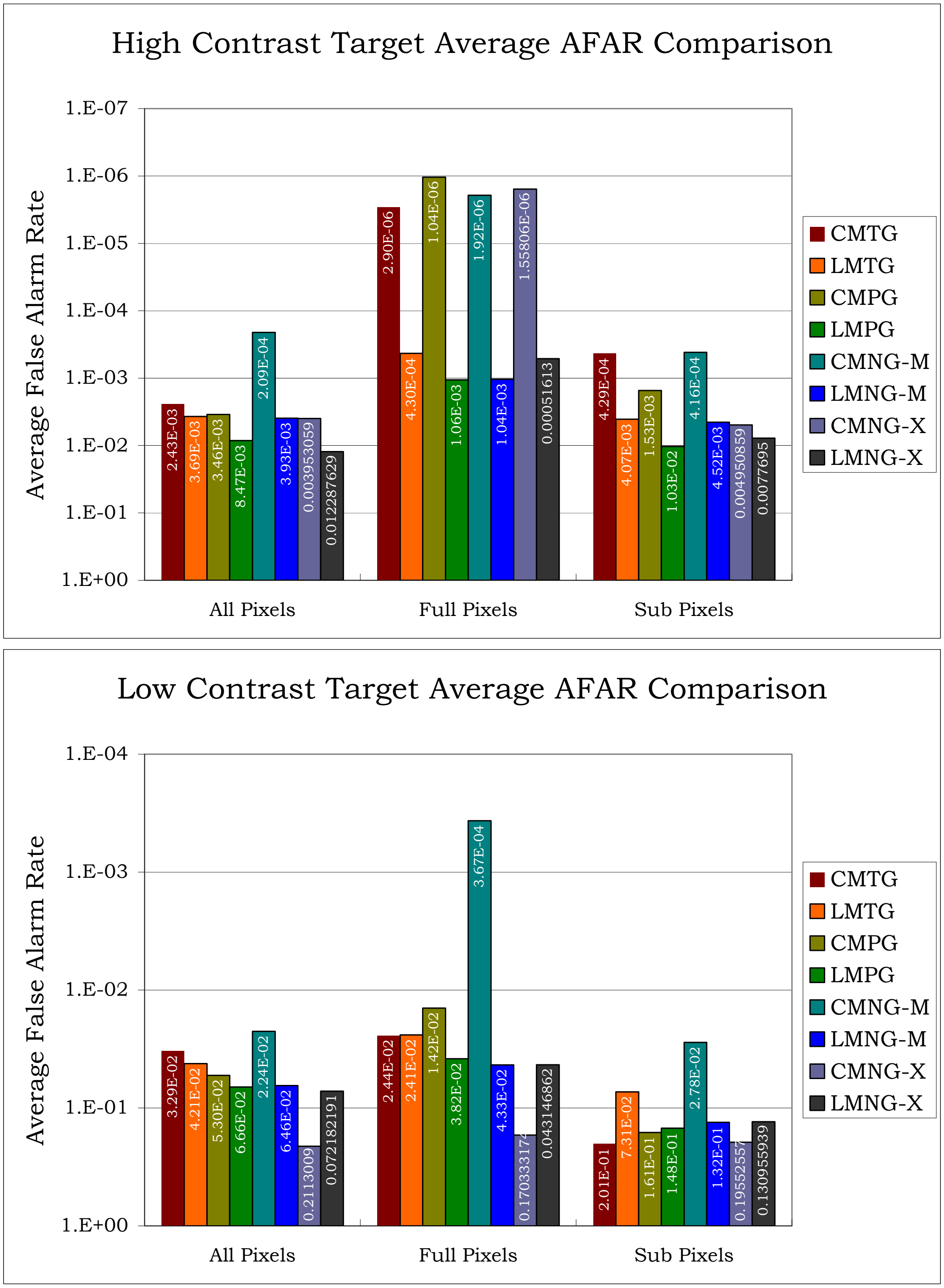


Figure 4.17 Run03 Pre-Clustering Summary AFAR Plots

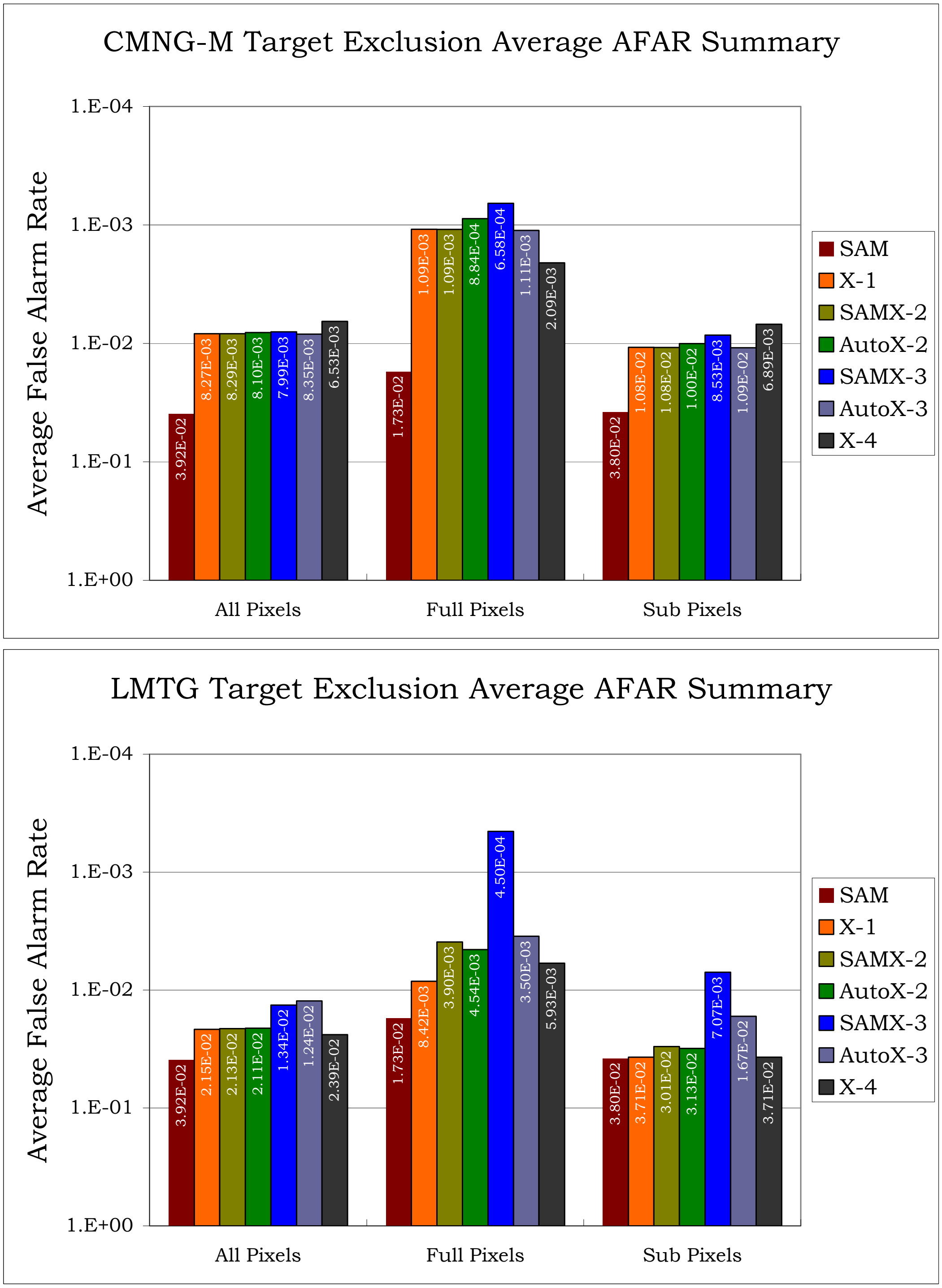
method performed well for full target pixels, but resulted in an AFAR an order of magnitude higher than other methods for the all and sub pixel targets. Interestingly, for those cases, the local mean neighbor guided-mode technique performed very well. This followed the expectation that the local mean was better suited to match sub pixel mixtures and the class mean was better suited to match false alarms elsewhere in the image. This expectation was generally reinforced by the high contrast forest scene results, where three out of the four class mean techniques outperformed their local mean counterparts for the full pixels, and vice versa for the sub pixels. This was not reinforced by the desert scene, but the fact that the desert scene high contrast targets were found with fewer false alarms by the class means for every pixel type led to the conclusion that the local mean techniques failed in the desert scene. This may have been due to image noise, higher spatial frequency of the class distribution, the proximity of targets in the scene, or the lesser extent of the adjacency effect due to the flat terrain (little shine from nearby objects) and the dry atmosphere (little scattering of ground leaving photons). Any of these may have given the class mean techniques in general (and the CMNG-M method specifically) an advantage in this scene.

Of the other methods, the target guided techniques worked well for both scenes. The local mean target guided (LMTG) method worked well on the high contrast forest scene targets and the class mean target guided (CMTG) technique worked well for high contrast desert scene targets. The LMTG and CMNG methods are somewhat complementary, using a local or class mean and a class or local covariance. These methods seem to have combined the pre-clustered statistics in the best ways for these scenes. The two methods which clearly underperformed were the pixel guided and the neighbor guided – mixed techniques. As suspected, the class identity of the test pixel was not the optimal method of class selection. Interestingly, the linear mixture of class statistics proved to be suboptimal as well. The mode techniques outperform the mixture techniques without incorporating information about all of the surrounding pixels. This indicated that in these images, the sub pixel mixture interference was dominated by a single source and did not combine evenly according to population fraction.

Given the overall level of performance, the CMNG-M technique was considered the best method for detection using pre-clustering. In order to explore the influence of target contamination in this method, an extensive experiment was performed using several realistic exclusion cases. For comparison, the LMTG method was included in the study. The next section will discuss how these results might guide the realistic application of these methods.

4.2.4 Pre-Clustering Target Influence

The one practical difficulty to implementing a pre-filtering or iterative target exclusion scheme is selecting a threshold for generating the exclusion map. In order to guide this decision, an experiment was conducted setting different levels of thresholds for two separate pre-filtering methods: SAM exclusion (SamX) and auto exclusion (AutoX). The selected thresholds, illustrated in Figure 3.17, were based on detection rate. Again, this does not represent a realistic implementation of a target exclusion scheme, but may help guide the selection of an appropriate threshold for real-world applications. ROC curves and AFAR graphs for each of the four targets used in this experiment are provided in Appendix A. Figure 4.18 shows the average AFAR for detection using different levels of exclusion. From the figure, it is clear that detection using the matched filter with no exclusion (X-1) is an improvement on the baseline SAM algorithm (SAM) provided in the figure for reference. However, when applying these detection results to create an exclusion map, the SAM results thresholded to exclude all of the target pixels seemed to be the best method. The fact that higher levels of exclusion provide improved results supports the use of loose thresholds for real-world target detection problems. Additionally, SAM exclusion outperformed auto exclusion due to a simple conceptual difference between the two. As stated earlier, auto exclusion was included in this experiment in an attempt to match the exclusion technique to the detection algorithm. Because higher levels of exclusion are preferable, matching the pre-filtering and detection algorithms may not be desirable. Non-target pixels in the overlapping distribution of the SAM results may not be pixels that a matched filter needs to suppress as false alarms. This is certainly not the case for auto exclusion, as the species in the overlapping region would expectedly be nearly identical. Additionally, noting that the perfect exclusion of targets did not always provide the best false alarm rates, it is possible that SAM exclusion was eliminating non-target species that were detrimental to the background in some other way. The exclusion of anomalous, noise, or otherwise undesirable pixels may also be accomplished by statistical distance exclusion, the results of which will be discussed in the next section.


CMNG-M Target Exclusion Average AFAR Summary

LMTG Target Exclusion Average AFAR Summary

Figure 4.18 Pre-Clustering Target Influence Summary AFAR Plots

The spectral exclusion problem also relates to the question of how the detection result is to be used. The desired level of exclusion should relate to the level of acceptable false alarms. In this study, the AFAR metric was calculated over all detection rates, tailoring the results for an application like image screening where higher levels of false alarms are acceptable in exchange for higher average detection rates for all pixels on a target. In this case, the results showed that a loosely thresholded SAM was most appropriate. The other case is an automated detection program that requires a very low false alarm rate while finding some of the targets, but can accept a higher overall number of false alarms. While the results of this experiment do not allow for conclusions about this type of application, it is likely that the auto exclusion method would be preferable. Given a well formed yet fully contaminated background, the matched filter generally outperformed SAM. In several cases where SAM found several false alarms before finding the first target pixel, the contaminated matched filter (X-1) was able to find a small number of target pixels without false alarms. By excluding the targets from the exposed end of the X-1 detection statistic distribution without excluding background pixels, the false alarm rate of the pixels in the overlapping region would decrease. The results in this section may serve to guide exclusion for image screening, but more in-depth experimentation is required to examine the role of auto exclusion for automated pre-filtering.

4.2.5 Statistical Distance Exclusion and Low Contrast Detection

Statistical distance exclusion (SDE) was tested for the potential of improving the MVN of class statistics. During the course of this research, SDE was implemented in many forms. The only parameter to select in the process is the number of pixels to discard for each class. In an attempt to automate the process, development of a program was begun to mathematically identify the knee in the distribution from a chi-squared MVN test curve. The completion of this tool was left for future work, but results from the manual method are provided in Appendix A. The goodness-of-fit was generally improved by the SDE method, but given the loose relationship between MVN and detection results, the matter of achieving improved background characterization remained. To narrow the scope of this examination, one method was selected to test the potential of SDE. Figure 4.15 shows that the SDC backgrounds used for the target guided methods were slightly less multivariate normal than their corresponding SEM classes. Therefore, the LMTG method was selected, and SDE was applied to improve the MVN of the closest class. One percent of the class was eliminated, and the MVN GoF improved from 97.53 to 35.42. The results of detection using these statistics are given in Figure 4.19, where the results with full statistics are labeled LMTG and the SDE results are labeled LMTG – SDE. The SDE backgrounds improved results for some but not all cases. The outliers in this class seemed to represent important sources of interference for certain targets, but not for others. This led to the trade-off between improving the MVN and suppressing false alarm pixels in the background.

The low contrast detection concept follows the mean subtraction scheme in equation 3.7. With the tools available and the modification simple to accomplish, cases were run using this type of mean subtraction during pre-clustering for the low contrast targets in the forest scene. Because this type of mean subtraction infers a local mean, the LMTG method was also appropriate for this study. The results of detection performed while mean subtracting both test pixel and target spectrum (the original LMTG) and mean subtracting only the test pixel (LMTG – LC) are shown in Figure 4.19. The results for these four targets showed that low contrast detection can both improve and hinder detection. An interesting observation was that the targets F8 and V1, which had an all pixel AFAR greater than 1.0E-1, experienced improved results with low contrast detection, while targets VF1 and VF6, where the all pixel AFAR was below 1.0E-1, did not. This indicated that the technique may only work for *extremely* low contrast cases. It was interesting that results for VF6 were hindered by low contrast detection but improved by SDE, while VF1 was not improved by either method. Each method was applied independently, but some level of a combination of methods (to include pre-filtering target exclusion) may provide the best background for a given target.

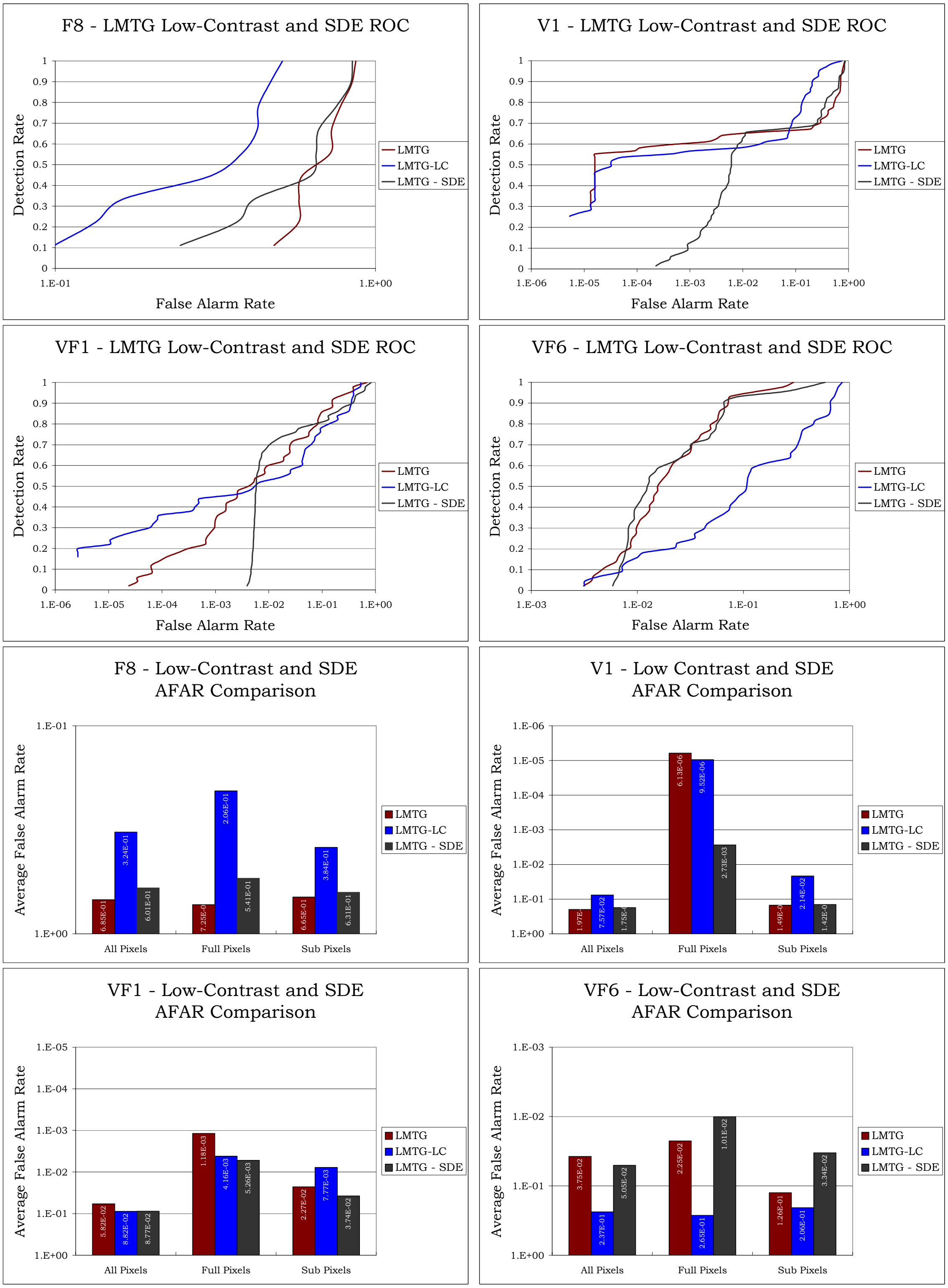


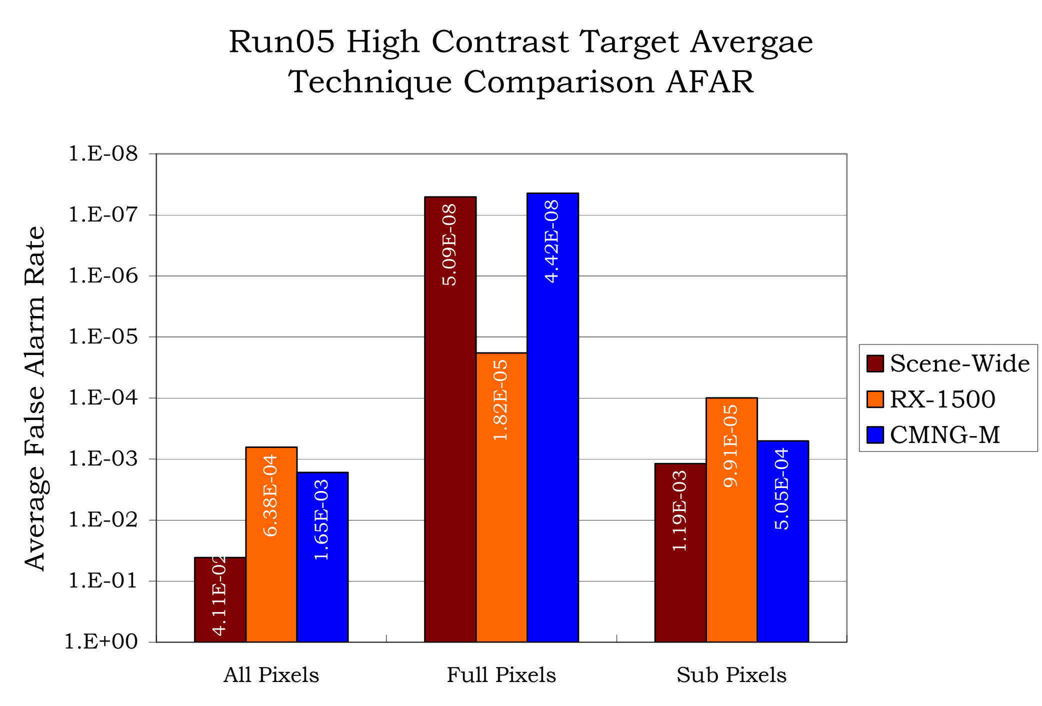
Figure 4.19 SDE and Low Contrast ROC and AFAR Results

4.3 Comparison of All Results

Considering the results of each background characterization method separately, observations have been made about the subtle and significant differences in the selection of parameters for each method. Target approach regions, RX window sizes, and methods for the selection of pre-clustering statistics have provided backgrounds leading to varying levels of detection performance. This study has explored the way in which each of these techniques provides greater multivariate normality and several methods for excluding target species from these backgrounds have been examined. As a final comparison, Figures 4.20 and 4.21 are the average AFAR for high and low contrast targets in both scenes for the representative techniques from each category. ROC curves and AFAR graphs for these three techniques for each individual target are included in Appendix A. By direct comparison using all of the target pixels, the spectral subsetting CMNG-M technique provides the best backgrounds for all but the forest scene high contrast targets. The hypothesis that RX is better suited for sub pixel detection and pre-clustering is better suited for full pixel detection was supported by all except the forest scene low contrast results. In general, the spatial or spectral subsetting methods provided backgrounds resulting in one-tenth to two orders of magnitude improvement in AFAR for all pixels compared to the scene-wide and target approach backgrounds.

These improvements were achieved by matching the source of mixture interference or false alarms elsewhere in the scene and by improving the multivariate normality of the data seeding the covariance in the matched filter. While the RX sliding window was able to improve the MVN to a greater extent, both spatial and spectral subsetting techniques provided backgrounds that were multivariate normal to an acceptable level. Given these relatively well formed statistics, the ability of each technique to match the interference dominated the result.

Targets were perfectly excluded from each of these backgrounds, so one final consideration is the susceptibility of these techniques to target contamination. The potential for each of these techniques to become contaminated in a real-world application is related to the level of available *a priori* information and is therefore not quantifiable. Target influence is related to the nature and number of target and background pixels in the background distribution and not the method used for selecting the background pixels. Therefore, all of the techniques in the study are exposed to the contamination problem and all need to employ spatial or spectral target exclusion when appropriate.



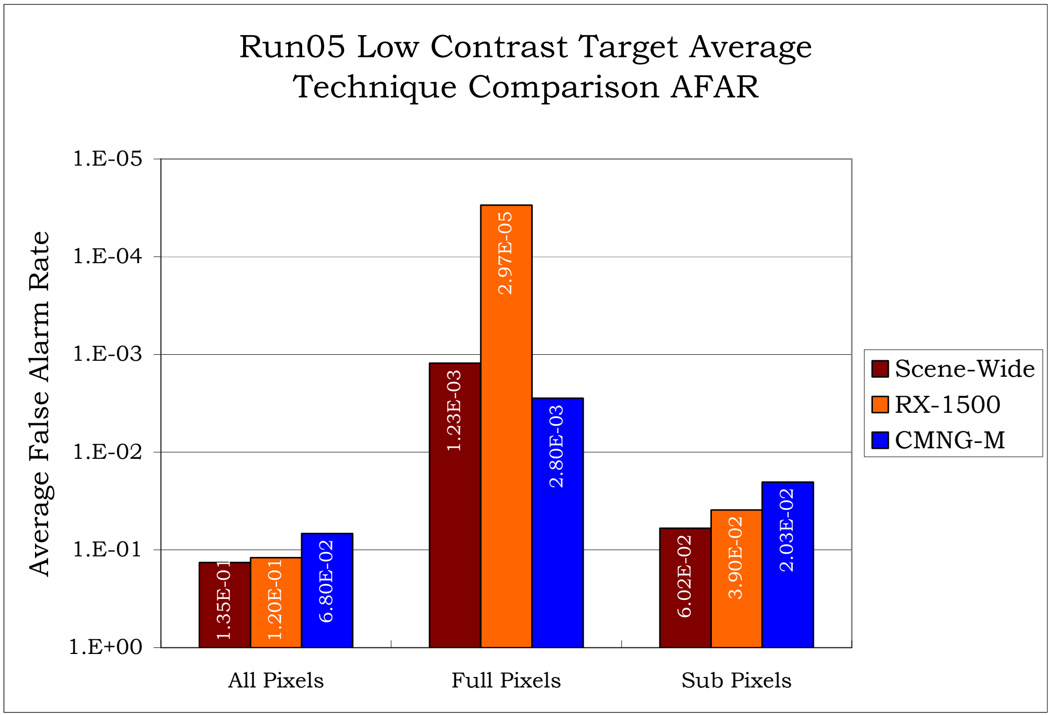


Figure 4.20 Run05 High and Low Contrast AFAR Final Comparison

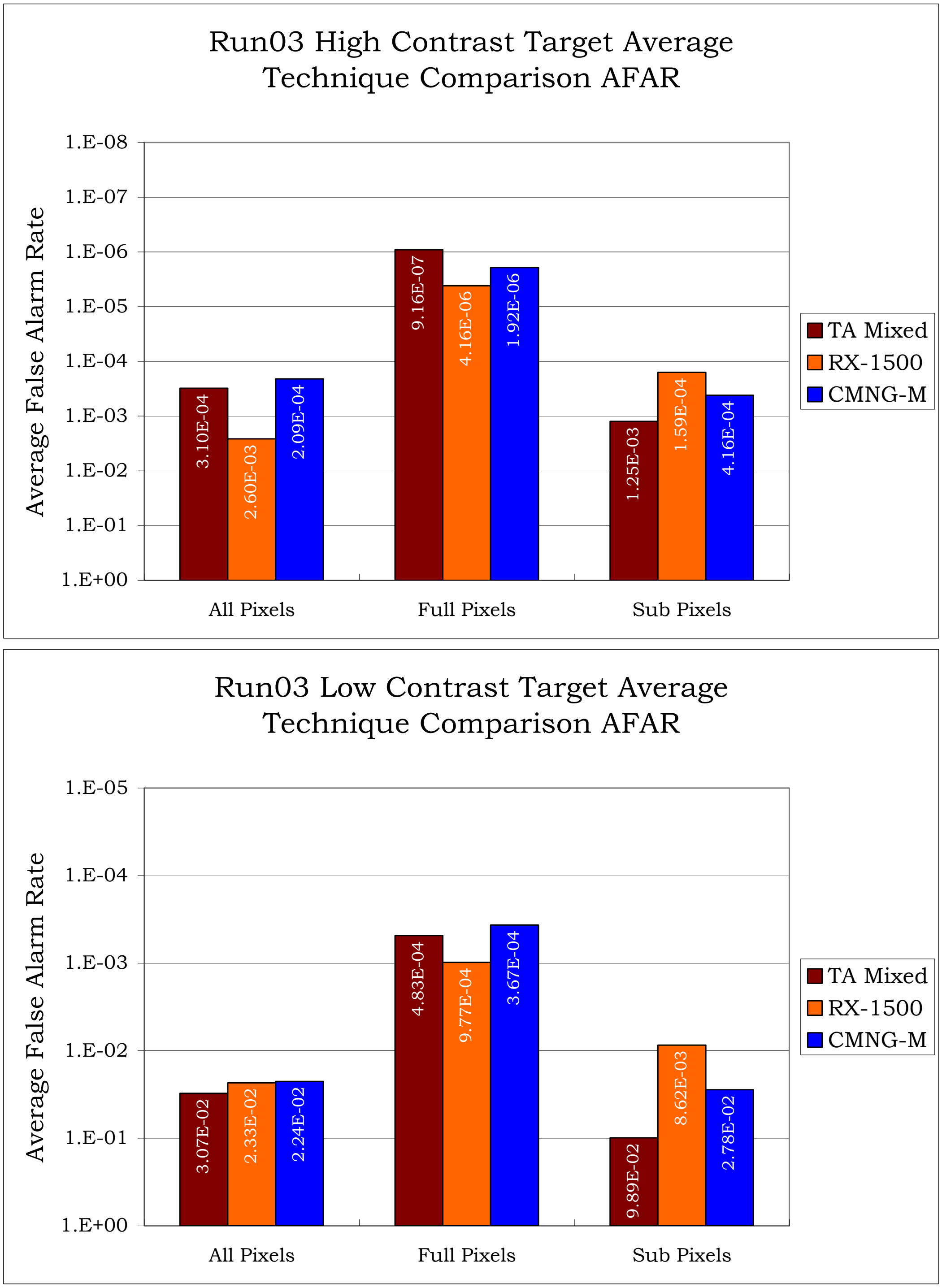


Figure 4.21 Run03 High and Low Contrast AFAR Final Comparison

5.0 Conclusions

The results of a comparative analysis of stochastic background characterization techniques have led to many observations about the way each method attempts to match the source of detection interference, improve the multivariate normality of the data, and handle the influence of target contamination. The main experiment removed the influence of target contamination through perfect exclusion and each technique improved the multivariate normality of the background to a level where small variations did not equate to improved detection performance. The observed differences in average false alarm rate (AFAR) are therefore more closely related to the ability of each technique to match the interference-causing false alarms. In the direct comparison of detection results, the RX sliding window and pre-clustering using the class mean neighbor guided – mode method consistently outperformed the scene-wide and target approach backgrounds. Improvements in the AFAR of one-tenth to two orders of magnitude were observed for all pixels on low and high contrast targets in the Forest Radiance I Run05 and Desert Radiance II Run03 images. Throughout the experiment, observations were made which may assist in the practical implementation of these methods. Possibilities for additional experimentation to further inform real world applications of background characterization were also noted.

For the target approach method, the scene-wide statistics provided a good matched filter background for an image with a relatively good measure of multivariate normality. The forest scene, with the targets removed, did not stray far enough from normality to confound detection results. The desert scene, however, provided an example of scene-wide statistics that failed as a background. In this case, it was suggested that the scene-wide measure of MVN or a measure of sensor noise levels would prompt the selection of a mixed target approach region to serve as background. The influence of target contamination was proven using the target approach method as target pixels were implanted in a target-free data pool and a steady change in the shape of the covariance matrix was observed. This change was correlated to a decrease in the performance of detection with only a few pixels added to a pool of 18,000 samples.

The RX sliding window algorithm with four concentric windows was tested to identify the proper sizing for the covariance window. The commonly accepted rule of selecting the number of samples equal to ten times the number of bands held for these targets and backgrounds. This result conflicted with the application of RX using selection of the most multivariate normal background. The number of samples used to estimate the covariance created a statistical stability that was more important than the statistical MVN in the application of the matched filter. In practical application, the type of data, noise levels, and scene clutter content will change this relationship. Additionally, the possibility of target contamination, which is driven by the level of *a priori* knowledge about the target and scene, must influence window size selection. The trade space includes the desire to capture the immediate surroundings of the target, the MVN of the data in the covariance window, the possibility of including targets, and the influence those targets will have on the background. The use of spatial and spectral target exclusion is extremely valuable when needed, but can lead to suboptimal performance if improperly implemented.

Pre-clustering was performed by classifying the imagery, developing statistics using the classmaps, and then deciding how to apply those statistics to the detection problem. K-Means and stochastic expectation maximization (SEM) were compared, and SEM was shown to generate backgrounds that provided modest improvements in detection but considerable improvements in background MVN. Eight methods were developed to apply pre-clustered statistics to detection by the selection of a local or class mean, and by guiding of class selection by the target spectrum, the pixel class assignment, or the class assignment of the neighbors of the test pixel (for use in either the mode or linear mixture of class statistics). The target guided methods required the use of a statistical distance classifier (SDC) in order to allow for predictability in the selection of the best background class given the target spectrum. The other methods, which used statistics from more than one class, employed the results of SEM classification. From the comparison of detection results, the class mean neighbor guided – mode (CMNG-M) technique demonstrated an advantage over other pre-clustering background characterization methods. CMNG-M was the best method for the desert scene and performed well in the forest scene for the detection of low contrast targets. The local mean target guided method performed well for the high contrast targets in the forest scene and represents a conceptual counterpart to the CMNG-M method. The pixel guided and neighbor guided – mixture methods consistently underperformed the other techniques.

The influence of target on the pre-clustered backgrounds was studied using the SAM algorithm as well as the matched filter results of a given method with a contaminated background. Several levels of exclusion were tested, and the best detection performance was achieved when spectral angle mapper (SAM) was used to eliminate all of the target pixels. While a large number of background pixels were excluded during pre-filtering to this level, it was apparent that the species in the overlapping portion of the SAM detection statistic distribution did not represent the highest likelihood matched filter false alarm. In this case, mixing modalities between the pre-filtering and detection algorithm seemed to provide an advantage.

The removal of pixels from the background using statistical distance exclusion (SDE) was another method studied. Discarding the outliers of the background distribution served to remove pixels that did not truly belong to their assigned class and to increase the multivariate normality of the background. MVN improvements were demonstrated, but SDE did not translate into improved detection in every case. In these instances, the benefit of including outliers in the distribution in order to suppress the unwanted signal outweighed the benefit from improved MVN.

An alternative mean subtraction method, which was aimed at improved detection for low contrast targets, was tested using all of the low contrast targets from the forest scene test set. The method provided improvement for two targets with an original AFAR of greater than one-tenth, but was detrimental to two targets with an AFAR less than one-tenth. This technique may be valuable only for extremely low contrast targets.

6.0 Future Work

Several aspects of the stochastic background characterization problem remain underdeveloped or unexplored. First and most importantly is an accurate measure of target contrast for a given detector. Knowing and understanding the difficulty of finding the target being sought is extremely valuable for these techniques. In this study, target contrast was determined after the fact through observation of AFAR. In practical application, a metric needs to be developed to provide this measure beforehand. This might go hand in hand with the development of an image quality metric for hyperspectral imagery (HSI) data. Both of these tools would inform the process and guide decision making for characterizing the background.

Another important undertaking is the application of the methods developed in this study to more and different data sets. One specific study would be measuring the relative performance of RX and pre-clustering on scenes with a high level of spatial clutter. The expectation is that pre-clustering would be less vulnerable to clutter and provide relatively normal backgrounds in scenes where the RX sliding window would fail. The use of pre-clustering statistics may also be changed in high clutter applications, or for different types of target detection. Applying SEM to pre-cluster high clutter imagery, or to search for different target such as plumes, may involve running each class independently and then fusing the results. For the RX window, high clutter imagery may increase the influence of the inversion method, necessitating a closer look at the automatic switch from Gaussian elimination to the Moore-Penrose pseudo inverse included in this implementation. Given the inequality of these methods, another way to handle singular RX covariance matrices may need to be developed.

Considering that a relatively stationary class mean with a covariance guided by species local to the test pixel provided good results for pre-clustering, the same type of parameters may need to be explored for RX. In this experiment, the size of the covariance window was varied and the mean window was held constant. Another experiment might vary the size of the mean window to determine if a more stable mean would be beneficial.

This study looked at eight different pre-clustering techniques but there are undoubtedly other ways to use classification results to improve detection. Continued work should look further into variations on some of the successful algorithms from this study. As a variation on the LMNG-M method, a future experiment might test the calculation of a local mean with only the pixels in the mode class. This would make the technique more similar to the CMNG-M, but would capture the local variability within the class during the mean subtraction.

More examination is also needed to determine the utility of SDE and low contrast detection. The theoretical advantages provided by these techniques need to be paired with more concrete guidelines for their application. One suggested improvement for SDE is excluding outliers on both ends of the distribution. Some of the classes exhibited tails at the high and low statistical distances and excluding only the high valued outliers worsened the multivariate normality and the performance. Automating the process of identifying the knee (or knees) in the distribution would be helpful in implementing this technique.

During analysis of the detection results, several algorithms perform similarly or with only modest improvement. To put small improvements in perspective, the development of a ROC or AFAR confidence interval would be very useful. With some degree of certainty attached, more aggressive conclusions may be reached about the relative capability of detection algorithms.

Another technique to consider is the automated implantation of sub pixel targets into every pixel in the scene to generate target and background distributions of equal size. This method would especially be applicable to any background characterization technique that has a spatial component and would remove the influence of spatial target location from the results.

Finally, the suite of pre-clustering techniques should be tested on more data and for more targets, including concealed or contaminated targets. While the CMNG-M was considered the top performing algorithm in general, other techniques were significantly better for specific targets.

7.0 Acknowledgements

This work was funded under the Office of Naval Research Multi-disciplinary University Research Initiative “Model-Based Hyperspectral Exploitation Algorithm Development” #N00014-01-1-0867. The author would also like to thank his wife Tamara and son Liam for their support, without which this work would not be possible.

8.0 References

E.A. Ashton, "Detection of Subpixel Anomalies in Multispectral Infrared Imagery Using an Adaptive Bayesian Classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, no. 2, pp. 506-517, March 1998.

P. Bajorski, "Analytical comparison of subpixel target detectors in structured models for hyperspectral images," *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, SPIE vol. 5806, no. 2, pp. 850-860, April, 2005.

P. Bajorski, E.J. Ientilucci, and J.R. Schott, "Comparison of Basis-Vector Selection Methods for Target and Background Subspaces as Applied to Subpixel Target Detection," Submitted for publication, 2004.

S.M. Bergman, "The Utility of Hyperspectral Data to Detect and Discriminate Actual and Decoy Target Vehicles," Master's Thesis, Naval Postgraduate School, December, 1996.

J.W. Boardman, F.A. Kruse and R.O. Green, "Mapping target signatures via partial unmixing of AVIRIS imagery," *Fifth JPL Airborne Earth Science Workshop*, vol. 1, pp. 23-26, 1995.

V. Carrere and J.E. Conel, "Recovery" of Atmospheric Water Vapor Total Column Abundance from Imaging Spectrometer Data Around 940nm – Sensitivity Analysis and Application to AVIRIS Data," *Remote Sensing of Environment*, vol. 44, pp. 179-204, 1993.

A.P. Cisz and J.R. Schott, "Performance comparison of hyperspectral target detection algorithms in altitude varying scenes," *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, SPIE vol. 5806, no. 2, pp. 839-849, April, 2005.

W.H. Ferrand and J.C. Harsanyi, "Mapping the distribution of mine tailings in the Coeur d' Alene River Valley, Idaho through the use of a constrained energy minimization technique," *Remote Sensing of Environment*, vol. 59, pp. 64-76, 1997.

C.C. Funk, J. Theiler, D.A. Roberts and C.C. Borel, “Clustering to Improve Matched Filter Detection of Weak Gas Plumes in Hyperspectral Thermal Imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, pp. 1410-1420, July 2001.

B. Gao, K.B. Heidebrecht, and F.H. Goetz, “Derivation of scaled surface reflectances from AVIRIS data,” *Remote Sensing of Environment*, vol. 44, pp. 165-178, 1993.

J.D. Gaskill, Linear System, Fourier Transforms, and Optics, John Wiley & Sons, New York, NY, 1976.

R.O. Green, J.E. Conel, and D.A. Roberts, “Estimation of aerosol optical depth, pressure elevation, water vapor and calculation of apparent surface reflectance from radiance measured by the airborne visible/infrared imaging spectrometer (AVIRIS) using a radiate transfer code,” *Proceeding of SPIE*, vol. 1937, pp. 2-11, 1993.

R.O. Green, B.E. Pavri and T.G. Chrien, “On-Orbit Radiometric and Spectral Calibration Characteristics of EO-1 Hyperion Derived With an Underflight of AVIRIS and *In Situ* Measurements at Salar de Arizaro, Argentina,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1194-1203, June 2003.

J.C. Harsanyi and C-I Chang, “Hyperspectral Image Classification and Dimensionality Reduction: An Orthogonal Subspace Approach”, *IEEE Transactions of Geoscience and Remote Sensing*, vol 32, no. 4, pp. 779-785, July 1994.

G. Healey and D. Slater, “Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions,” *IEEE Transactions of Geoscience and Remote Sensing*, vol. 37, pp. 2706-2717, November 1999.

J.P. Hoffbeck and D.A. Landgrebe, “Covariance Matrix Estimation and Classification with Limited Training Data”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763-767, July 1996.

T-J Hwu, C-P Han, and K. Rogers, “The Combination Test for Multivariate Normality”, *Journal of Statistical Computation and Simulation*, vol. 72, no. 5, pp. 379-390, January 2002.

E.J. Ientilucci and S.D. Brown, “Advances in wide-area hyperspectral image simulation,” *Targets and Backgrounds IX: Characterization and Representation,* Proceedings of SPIE, vol. 5075, pp. 110-121, September, 2003.

R.A. Johnson and D. W. Wichern, Applied Multivariate Statistical Analysis, Prentice Hall, Upper Saddle River, NJ, 2002.

E.J. Kelly, “Performance of an Adaptive Detection Algorithm; Rejection of Unwanted Signals,*” IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-25, no. 2, pp. 122-133, March 1989.

B-C Kuo and D.A. Landgrebe, “A Covariance Estimator for Small Sample Size Classification Problems and Its Application to Feature Extraction”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 4, pp. 814-819, April 2002.

K. Lee, “A Subpixel Target Detection Algorithm for Hyperspectral Imagery,” Ph.D. Dissertation, Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, May 2003.

Y. Li, A. Vodecek, R. Kremens, and A. Ononye, "A New Algorithm for Global Forest Fire Detection Using Multispectral Images," Targets and Backgrounds IX: Characterization and Representation, Proceedings of SPIE, vol. 5075, pp. 367-377, 2003.

D. Manolakis, G. Shaw and N. Keshava, “Comparative Analysis of Hyperspectral Adaptive Matched Filter Detectors,” *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI*, Proceedings of SPIE, vol. 4049, pp. 2-17, 2000.

D. Manolakis, D. Marden, J. Kerekes and G. Shaw, “On the Statistics of Hyperspectral Imaging Data,” *Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VII*, Proceedings of SPIE, vol. 4381, pp. 308-316, 2001.

D. Manolakis and G. Shaw, “Detection Algorithms for Hyperspectral Imaging Applications,” *IEEE Signal Processing Magazine*, vol. 19, pp. 29-43, January 2002.

P. Masson and W. Pieczynski, “SEM Algorithm and Unsupervised Statistical Segmentation of Satellite Images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 3, pp. 618-633, May 1993.

C.J. Mecklin and D.J. Mendfrom, “An Appraisal and Bibliography of Tests for Multivariate Normality”, *International Statistical Review / Revue Internationale de Statistique*, vol. 72, no. 1, pp. 123-138, April 2004.

E.C. Mercurio (ed.) “IAGT Involvement in RIT’s Wildfire Airborne Sensor Program,” Institute for the Application of Geospatial Technology – Technical Newsletter, vol. 1, no. 4, pp. 1-2, December 2003.

MIT Lincoln Laboratory, Canonic Data Set Documentation, Canonic/FR1/run05m51 /atcomp/atm_metrics.htm, 2004

A.E. Ononye, A. Vodacek, R.L. Kremens, Y. Li, and D. Merritt, “Empirical testing of subpixel detection of fire,” *Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery IX*, Proceedings of SPIE, vol. 5093, pp. 343-352, September 2003.

I.S. Reed and X. Yu, “Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 10, pp. 1760-1770, October 1990.

J.P. Royston, “An Extension of Shapiro and Wilk’s W Test for Normality to Large Samples*”, Applied Statistics*, vol. 32, no. 2, pp. 115-124, 1982.

W.E. Schaff, A. Copeland, M. Steffen, R. O’Connor, C. Simi, J. Zadnik, E.M. Winter, G. Healey, “Real-time data processor for the COMPASS hyperspectral sensor system,” *Imaging Spectrometry IX*, Proceedings of SPIE, vol. 5159, pp. 1-13, January 2004.

J. Schott, Remote Sensing: The Image Chain Approach, Oxford University Press, New York, NY, 1997.

J. Schott, Notes for Advanced Hyperspectral Algorithms Course, Rochester Institute of Technology, Spring 2004.

G.W. Snedecor and W.G. Cochran, Statistical Methods, Eighth Edition, Iowa State University Press, 1989.

D.W.J. Stein, A. Stocker, S. Beaven, “The Fusion of Quadratic Detection Statistics Applied to Hyperspectral Imagery,” SPAWAR Systems Center San Diego, Sponsored by CECOM RDEC Night Vision and Electronic Sensors Directorate, January 2001.

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*, vol. 19, pp. 58-69, January 2002.

A.D. Stocker and A.P. Schaum, “Application of stochastic mixing models to hyperspectral detection problems,” *Algorithms for Multispectral and Hyperspectral Imagery III*, SPIE vol. 3071, pp. 47-60, August, 1997.

S. Tadjudin and D.A. Landgrebe, “Covariance Estimation With Limited Training Samples”, *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 4, pp. 2113-2118, July 1999.

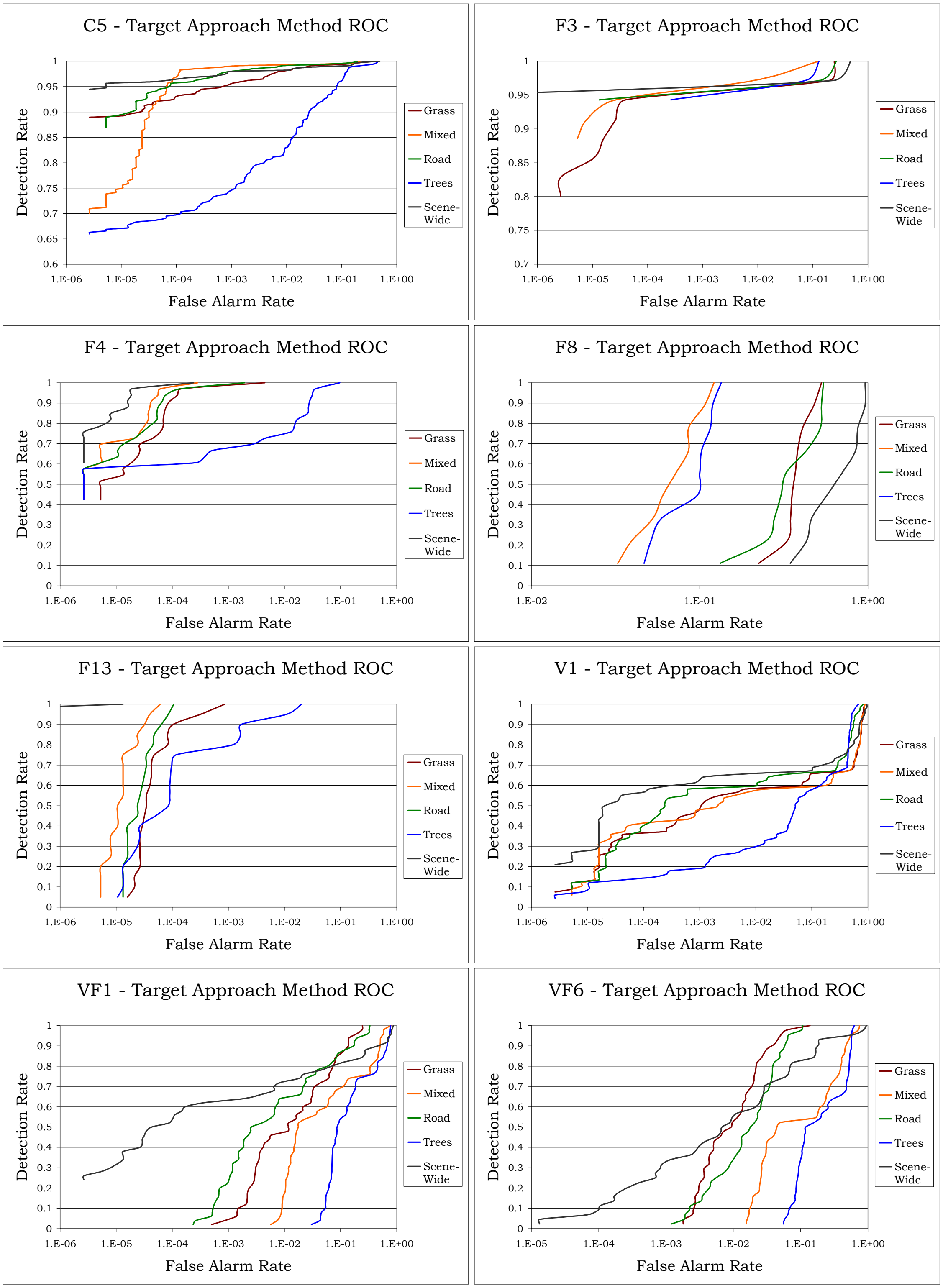
J.E. West, D.W. Messinger, E.J. Ientilucci, J.P. Kerekes, and J.R. Schott, "Matched filter stochastic background characterization for hyperspectral target detection," *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI*, SPIE vol. 5806, no. 1, pp. 1-12, April, 2005.

D.L. Williams, J.R. Irons, J.L. Barker, B.L. Markham, and J.A. Pedelty, “Landsat 7: maintaining Landsat data continuity into the 21st Century,” *Geoscience and Remote Sensing Symposium Proceedings, IGARSS ‘98*, vol. 1, pp. 256-258, July 1998.

M.E. Winter, “Fast Autonomous Spectral Endmember Determination in Hyperspectral Data”, *Thirteenth International Conference on Applied Geologic Remote Sensing*, Vancouver, British Columbia, Canada, March 1999.

APPENDIX A – Individual Detection Results

Full Scene and Target Approach Method ROC Curves



C5 - Target Approach Method ROC

F3 - Target Approach Method ROC

F4 - Target Approach Method ROC

F8 - Target Approach Method ROC

F13 - Target Approach Method ROC

V1 - Target Approach Method ROC

VF1 - Target Approach Method ROC

VF6 - Target Approach Method ROC

Figure A.1 Run05 Target Approach ROC Curves

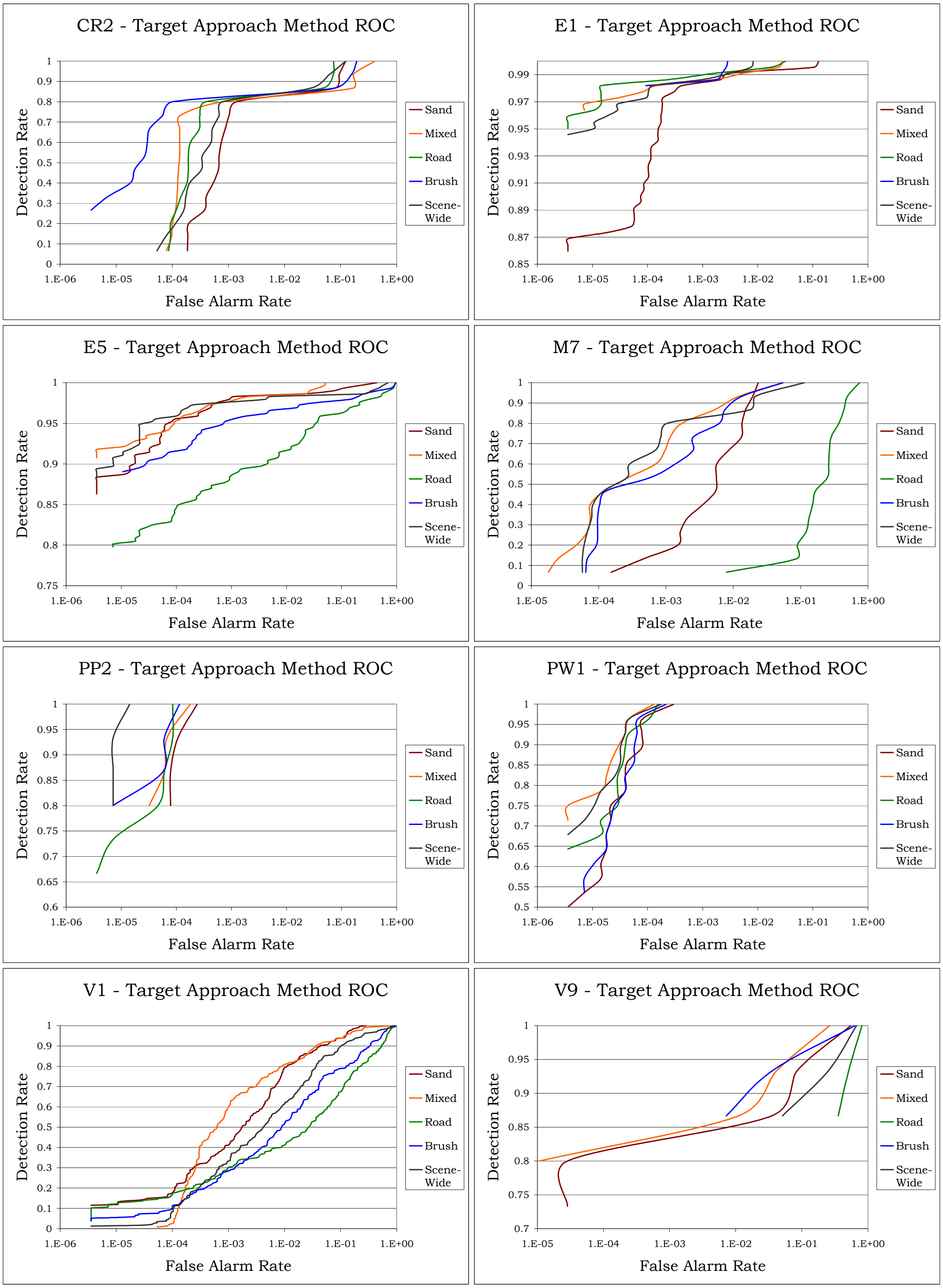
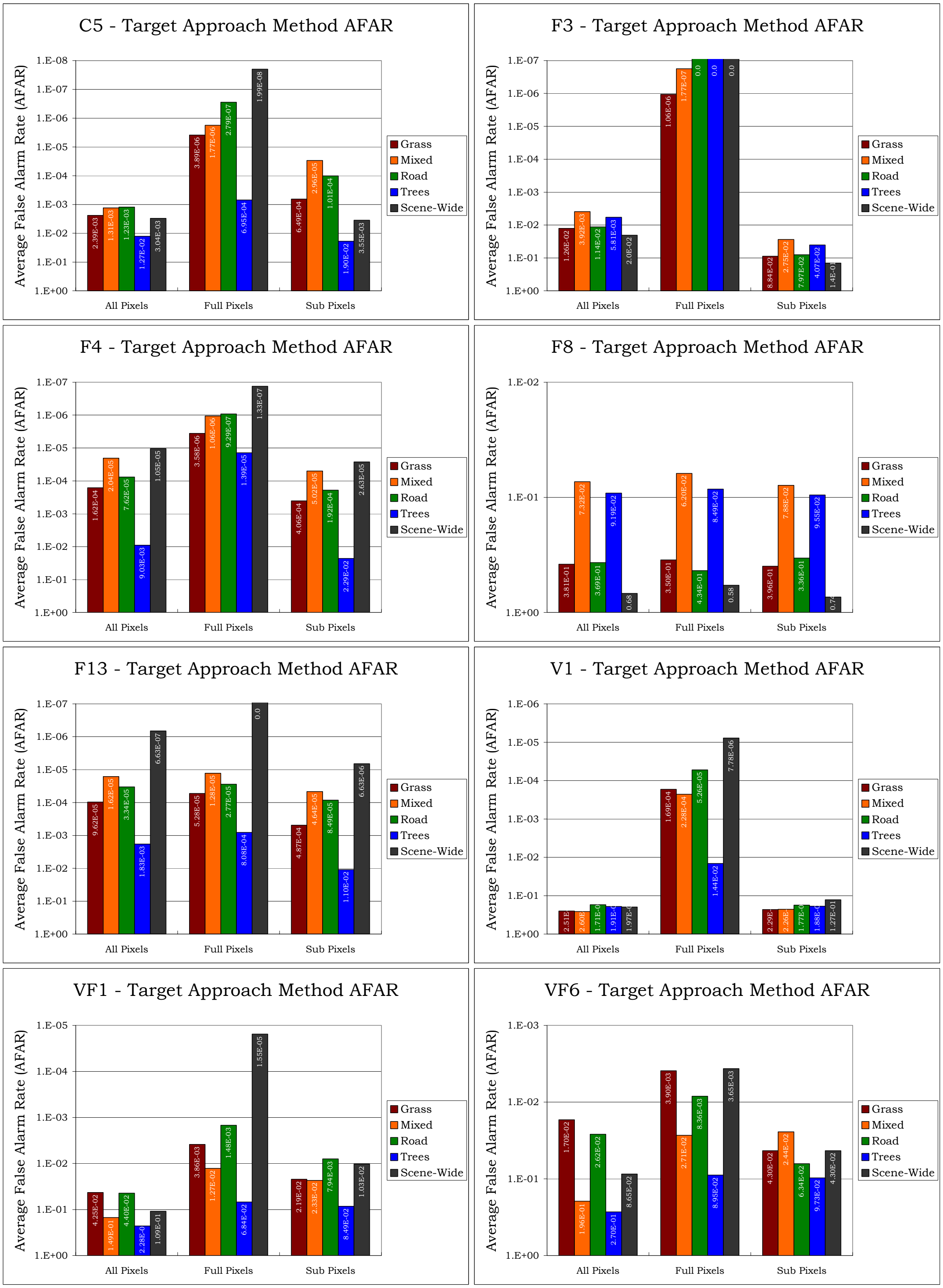


Figure A.2 Run03 Target Approach ROC Curves

Full Scene and Target Approach Method AFAR



C5 - Target Approach Method AFAR

F3 - Target Approach Method AFAR

F4 - Target Approach Method AFAR

F8 - Target Approach Method AFAR

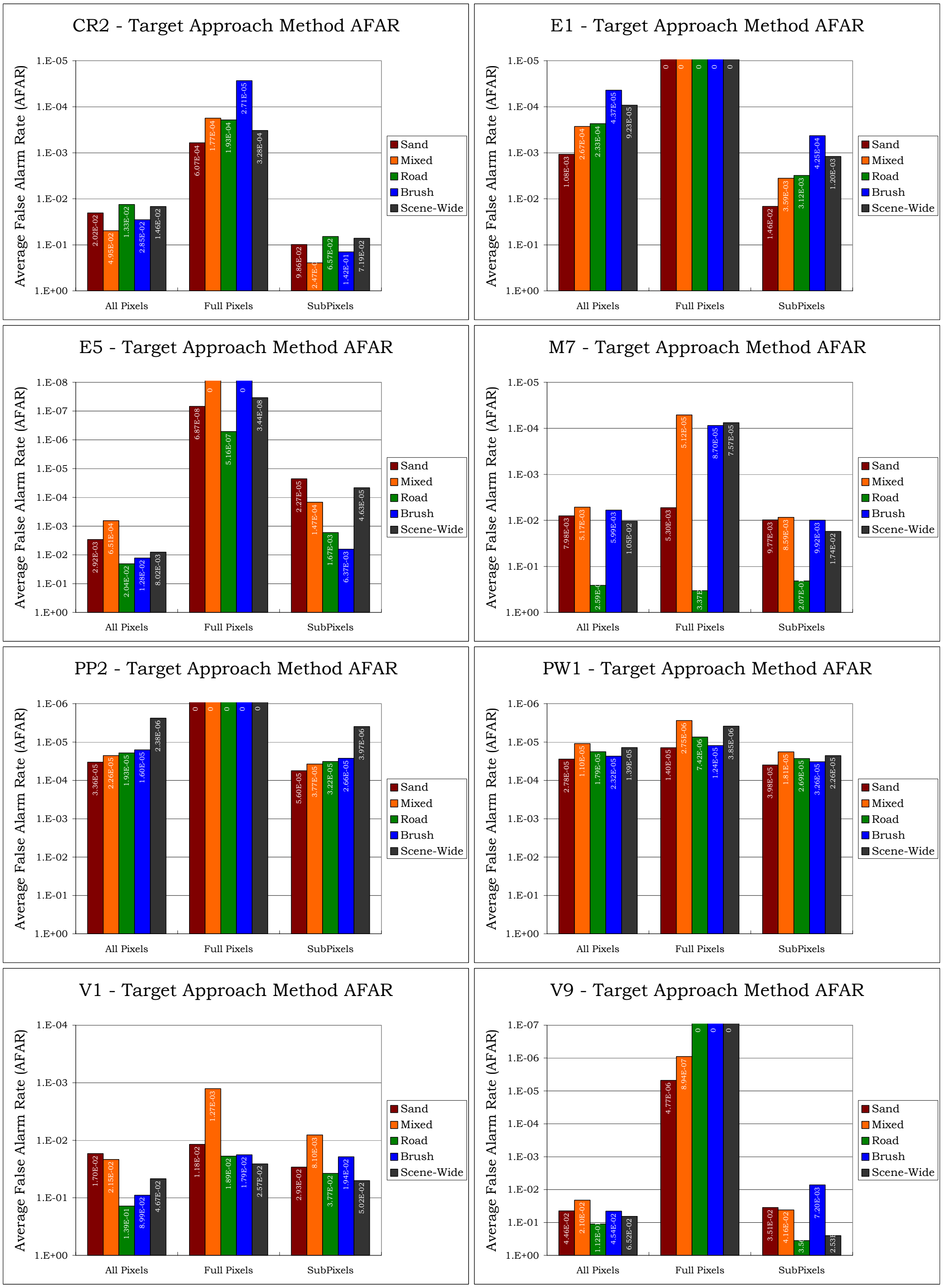
F13 - Target Approach Method AFAR

V1 - Target Approach Method AFAR

VF1 - Target Approach Method AFAR

VF6 - Target Approach Method AFAR

Figure A.3 Run05 Target Approach AFAR Plots



CR2 - Target Approach Method AFAR

E1 - Target Approach Method AFAR

E5 - Target Approach Method AFAR

M7 - Target Approach Method AFAR

PP2 - Target Approach Method AFAR

PW1 - Target Approach Method AFAR

V1 - Target Approach Method AFAR

V9 - Target Approach Method AFAR

Figure A.4 Run03 Target Approach AFAR Plots

RX Sliding Window ROC Curves

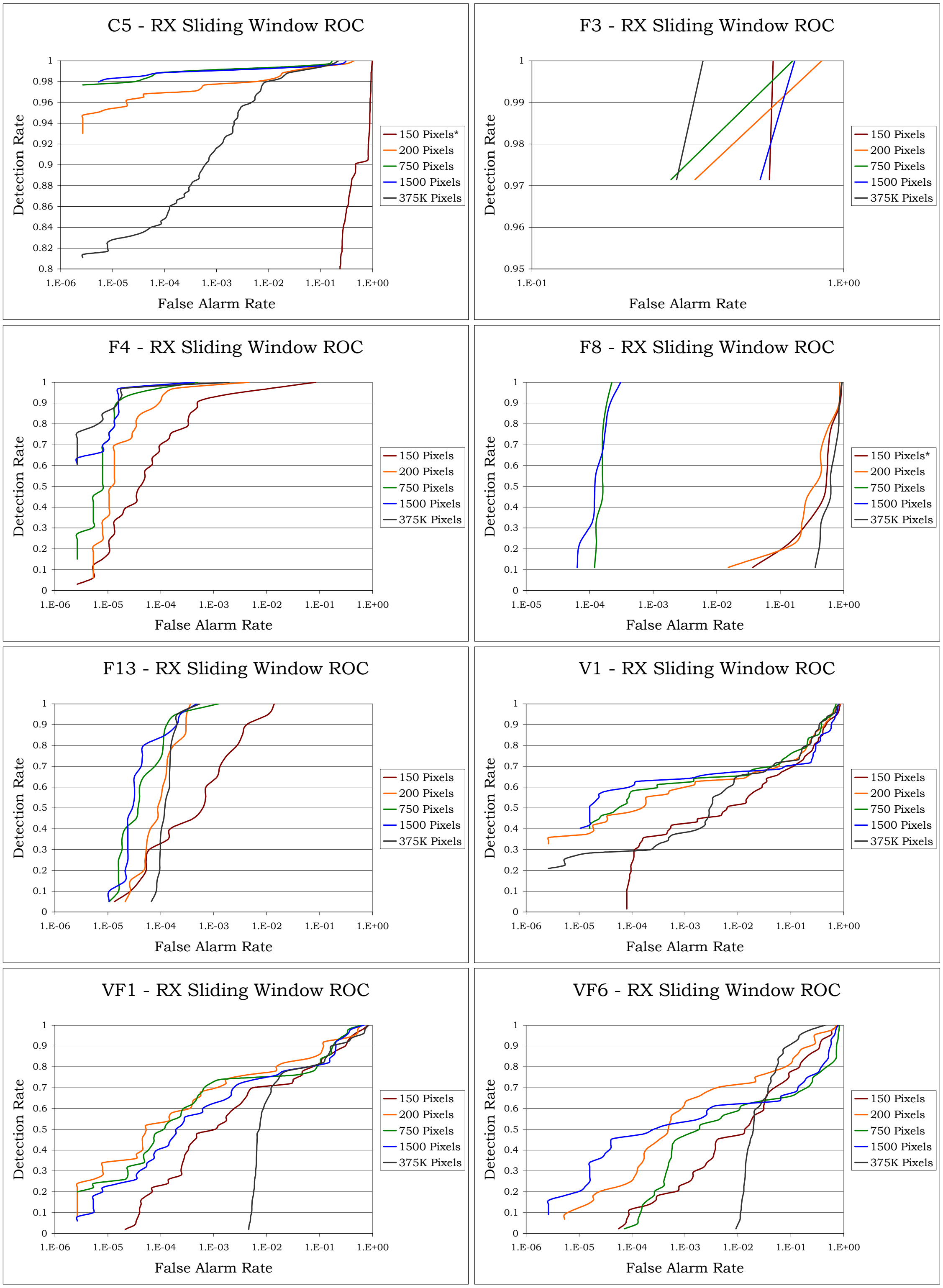


Figure A.5 Run05 RX Sliding Window ROC Results

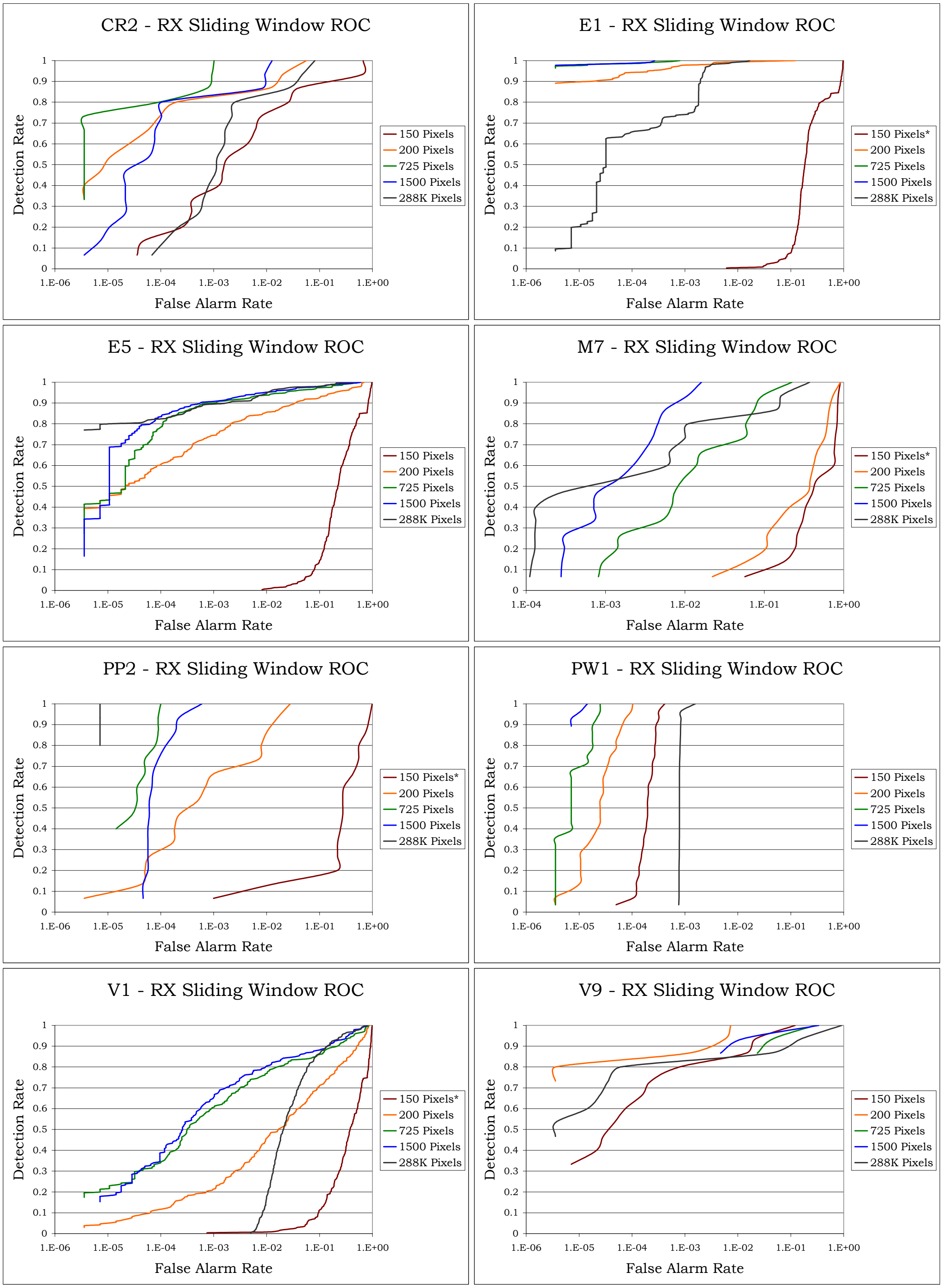


Figure A.6 Run03 RX Sliding Window ROC Results

RX Sliding Window AFAR

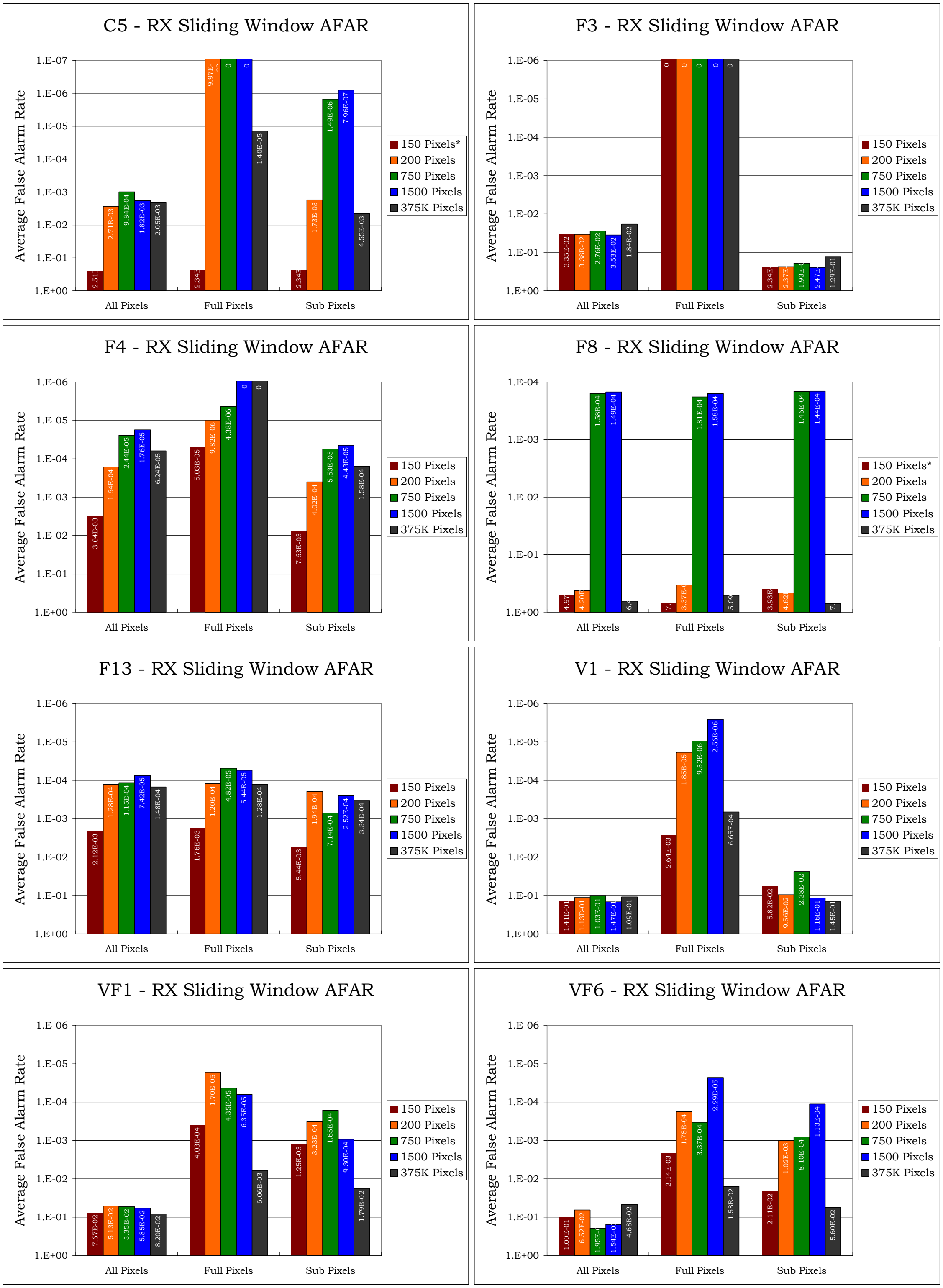
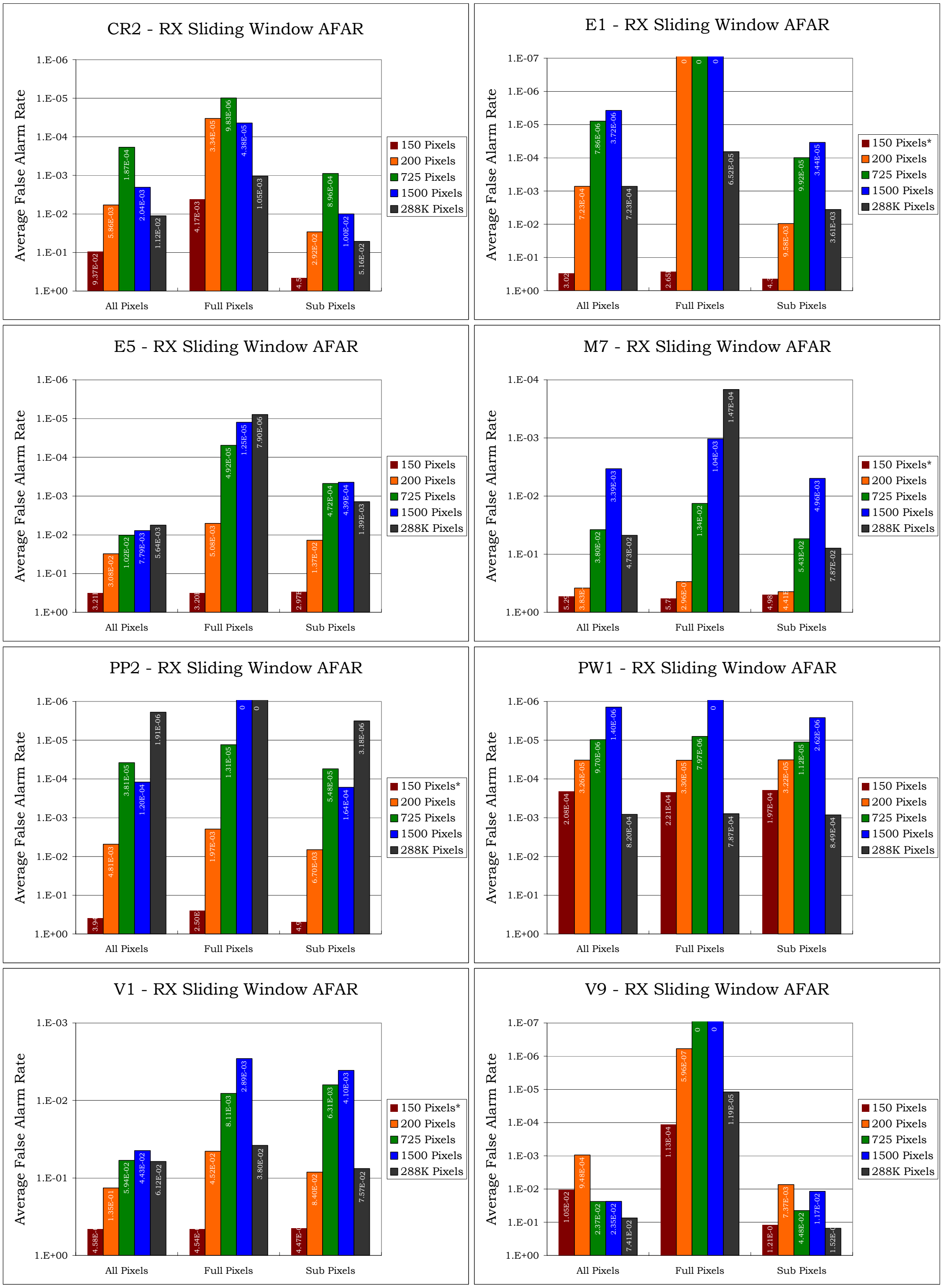


Figure A.7 Run05 RX Sliding Window AFAR Plots



CR2 - RX Sliding Window AFAR

E1 - RX Sliding Window AFAR

E5 - RX Sliding Window AFAR

M7 - RX Sliding Window AFAR

PP2 - RX Sliding Window AFAR

PW1 - RX Sliding Window AFAR

V1 - RX Sliding Window AFAR

V9 - RX Sliding Window AFAR

Figure A.8 Run03 RX Sliding Window AFAR Plots

4.1.2.3 RX Sliding Window MVN Gof

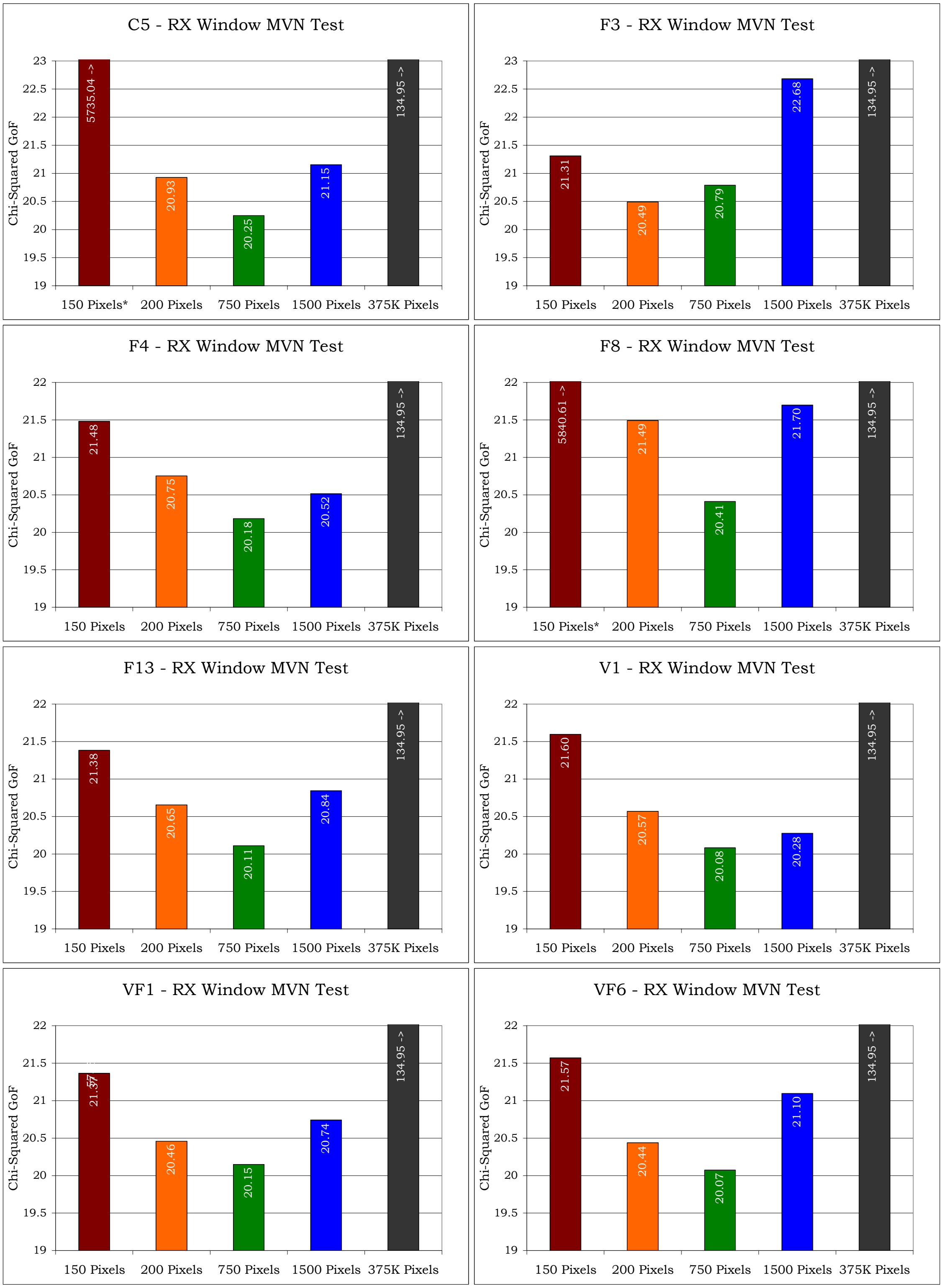


Figure A.9 Run05 RX Sliding Window MVN

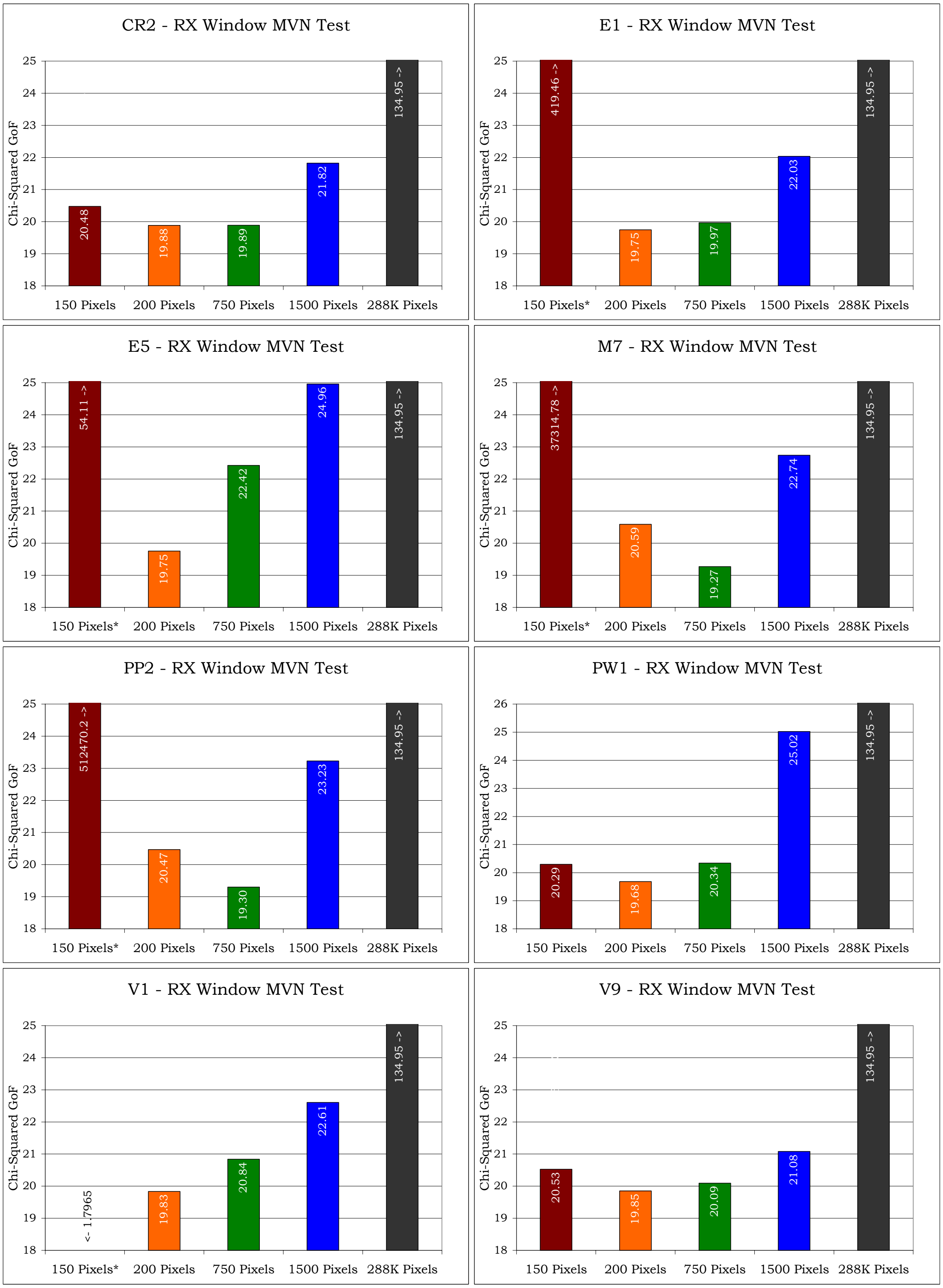


Figure A.10 Run03 RX Sliding Window MVN

Pre-Clustering ROC Curves

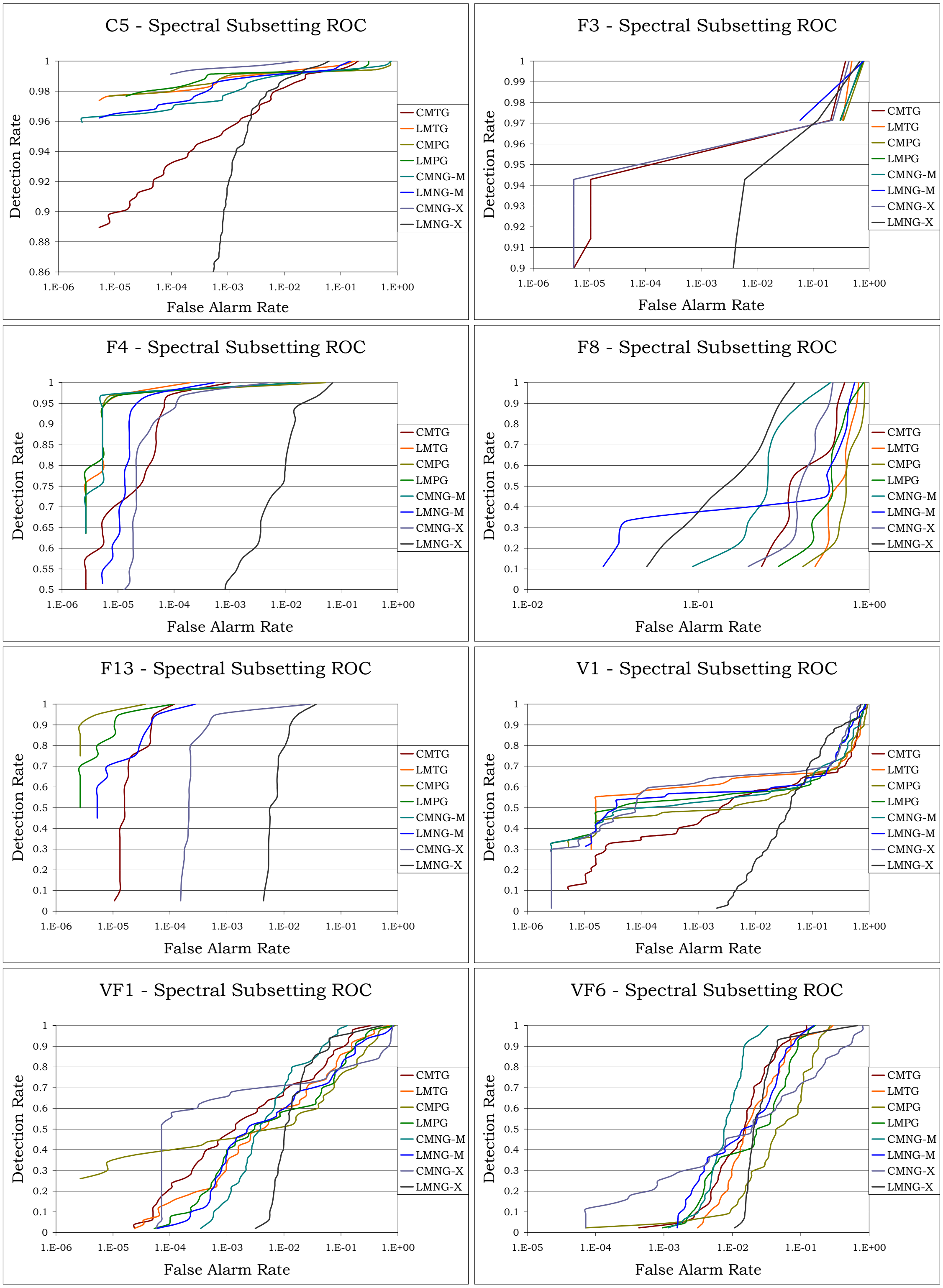


Figure A.11 Run05 Pre-Clustering ROC Results

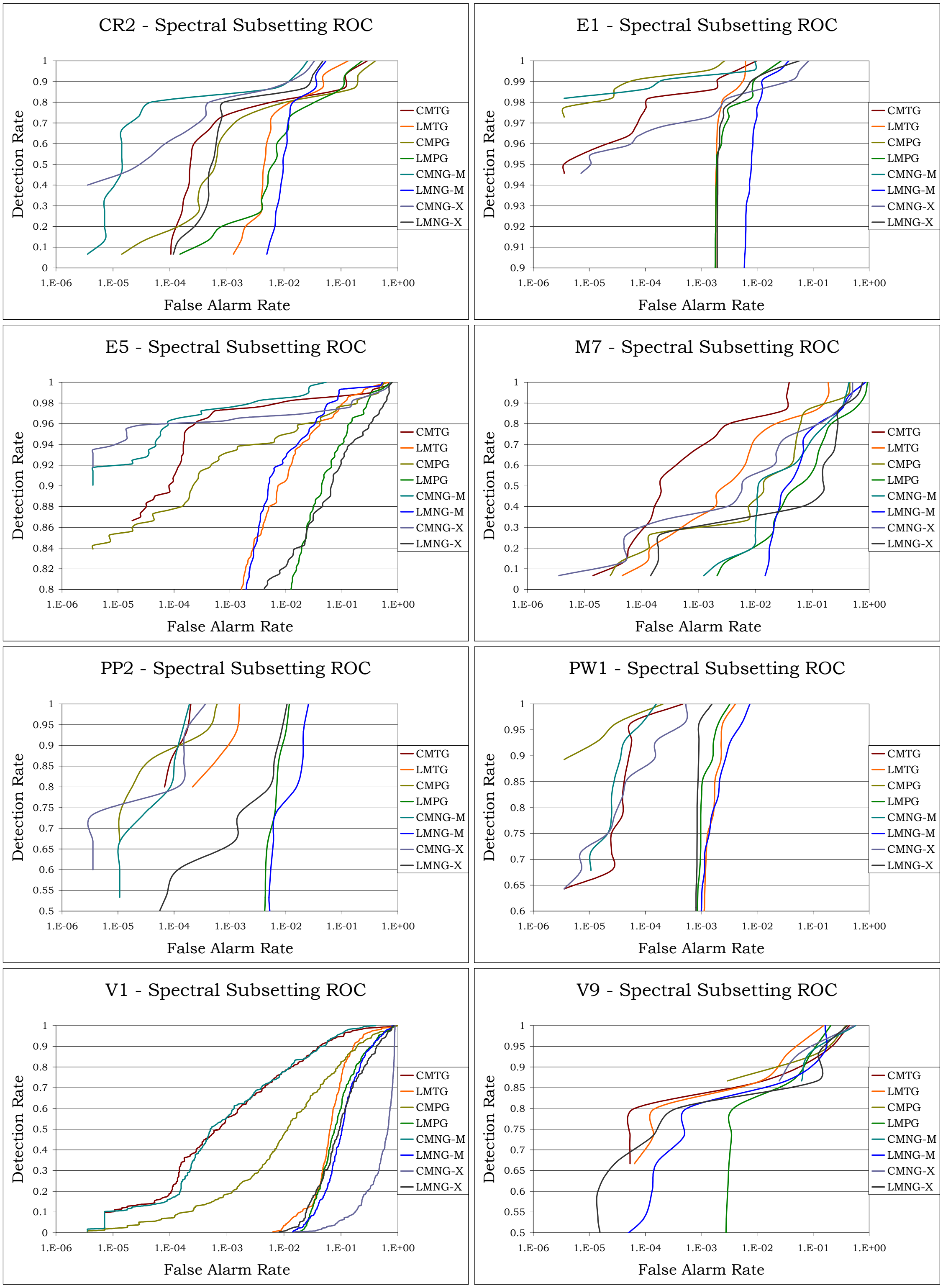


Figure A.12 Run03 Pre-Clustering ROC Results

Pre-Clustering AFAR

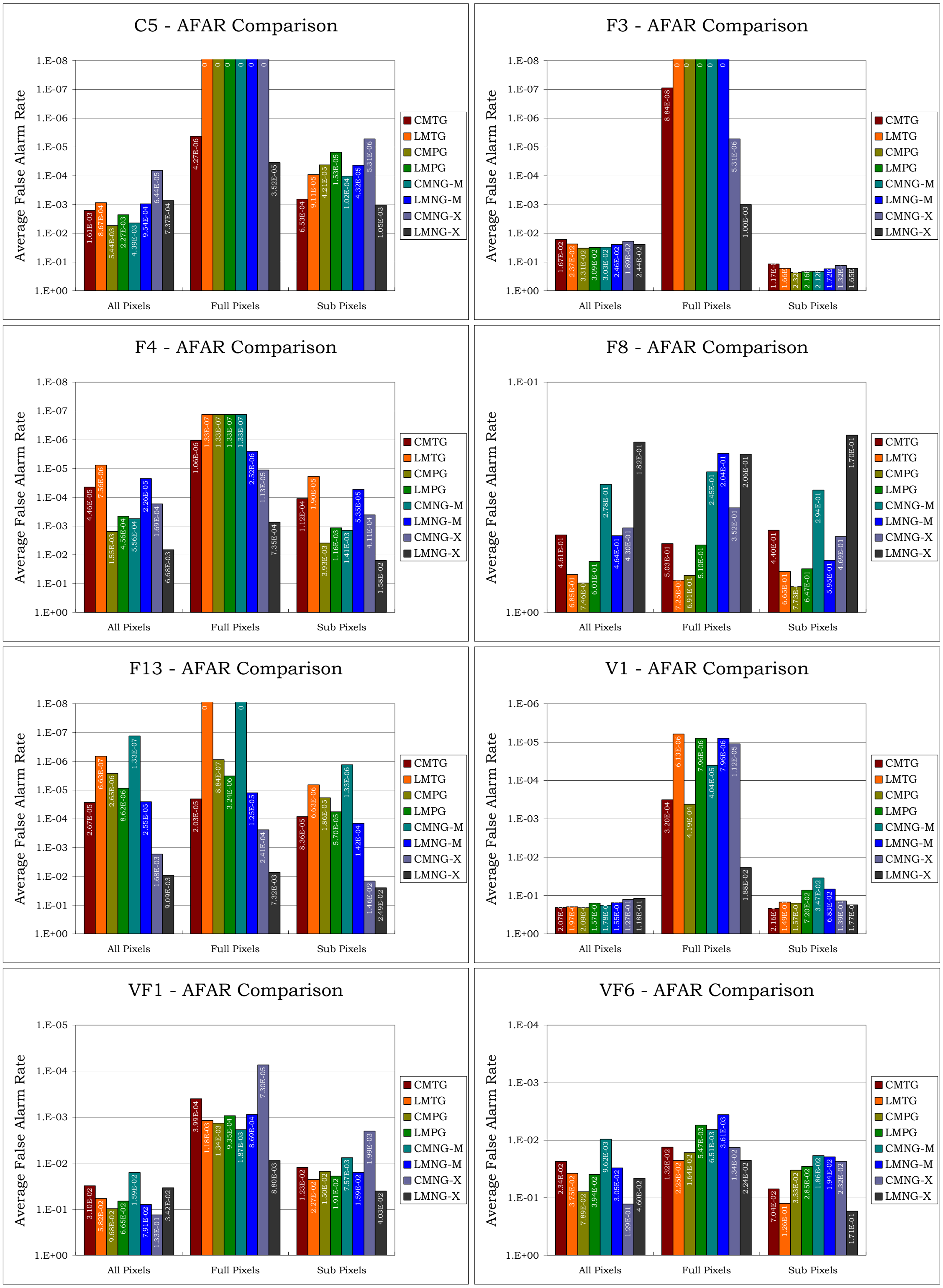


Figure A.13 Run05 Pre-Clustering AFAR Plots

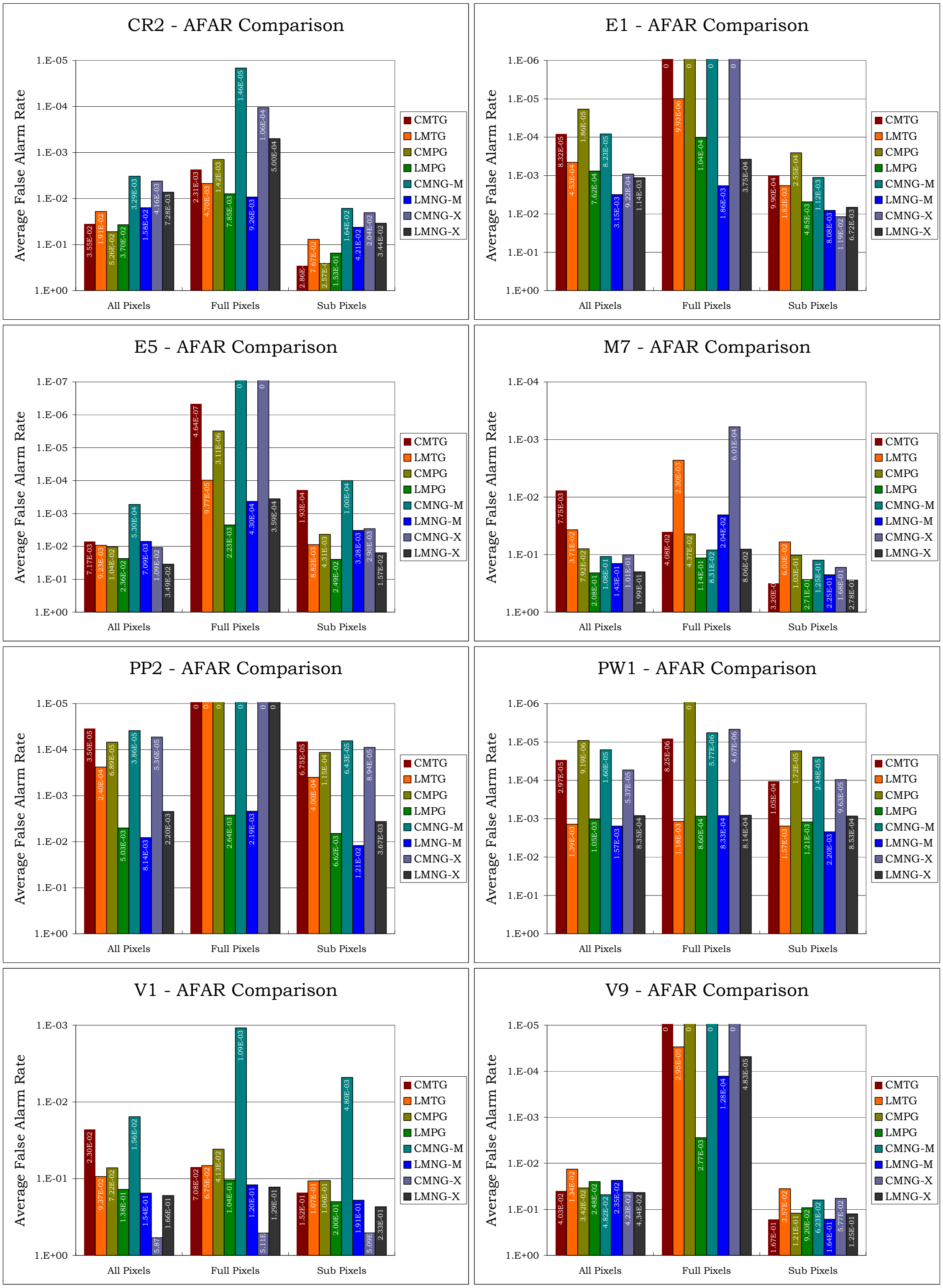


Figure A.14 Run03 Pre-Clustering AFAR Plots

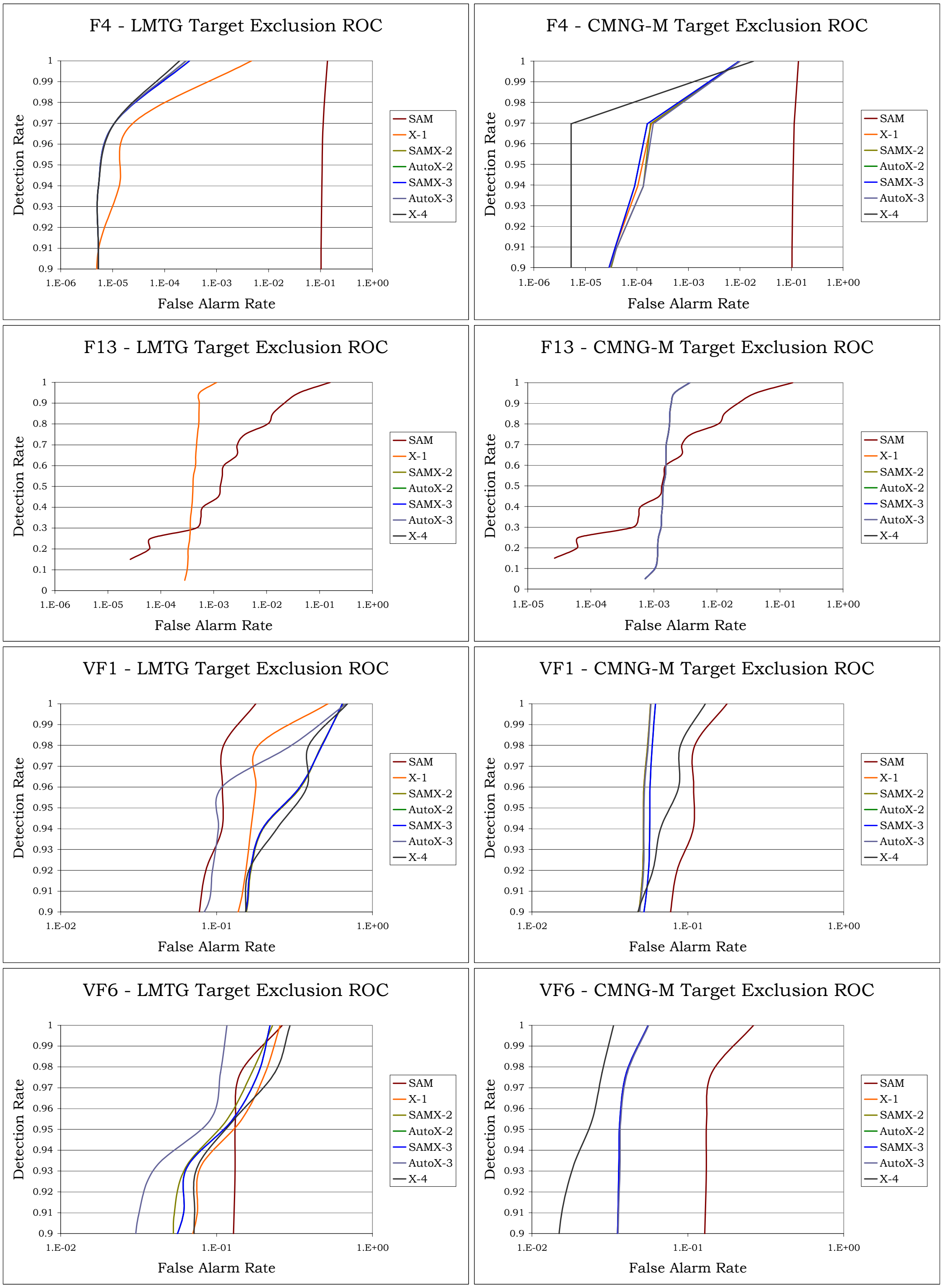


Figure A.15 Pre-Clustering Target Exclusion ROC Curves

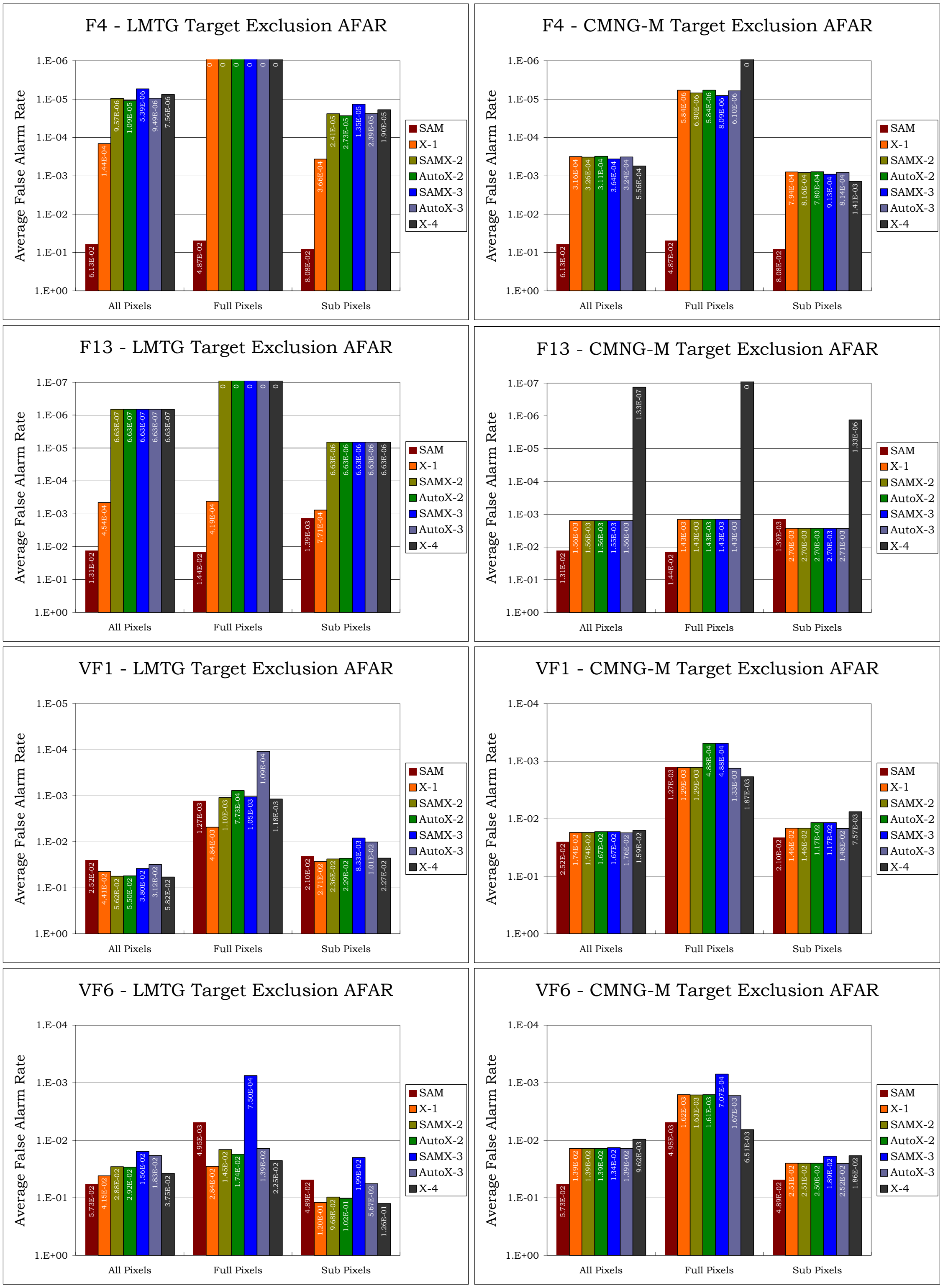


Figure A.16 Pre-Clustering Target Exclusion AFAR

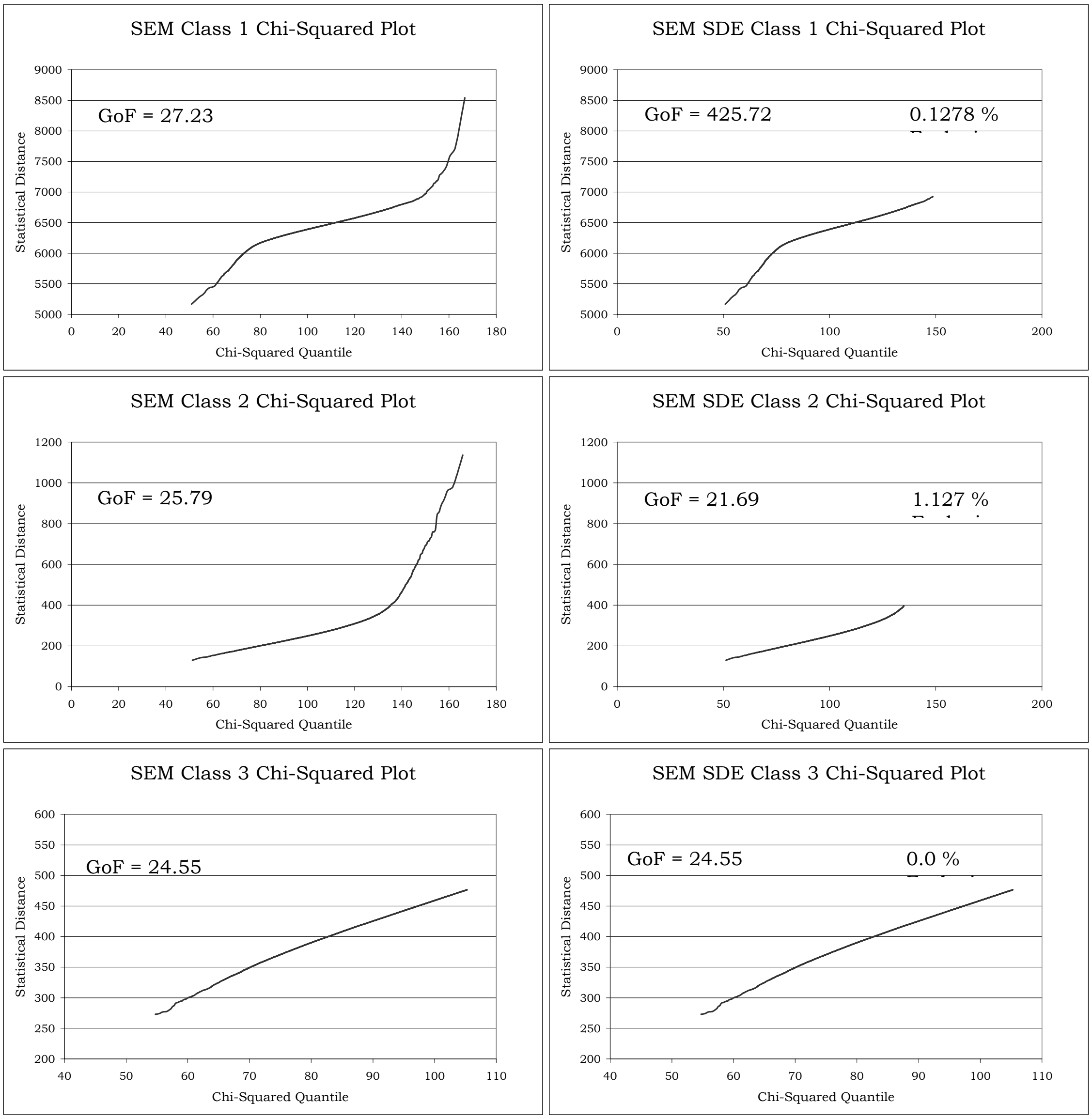


Figure A.17 SDE Chi-Squared Example for SEM Classes 1 – 3

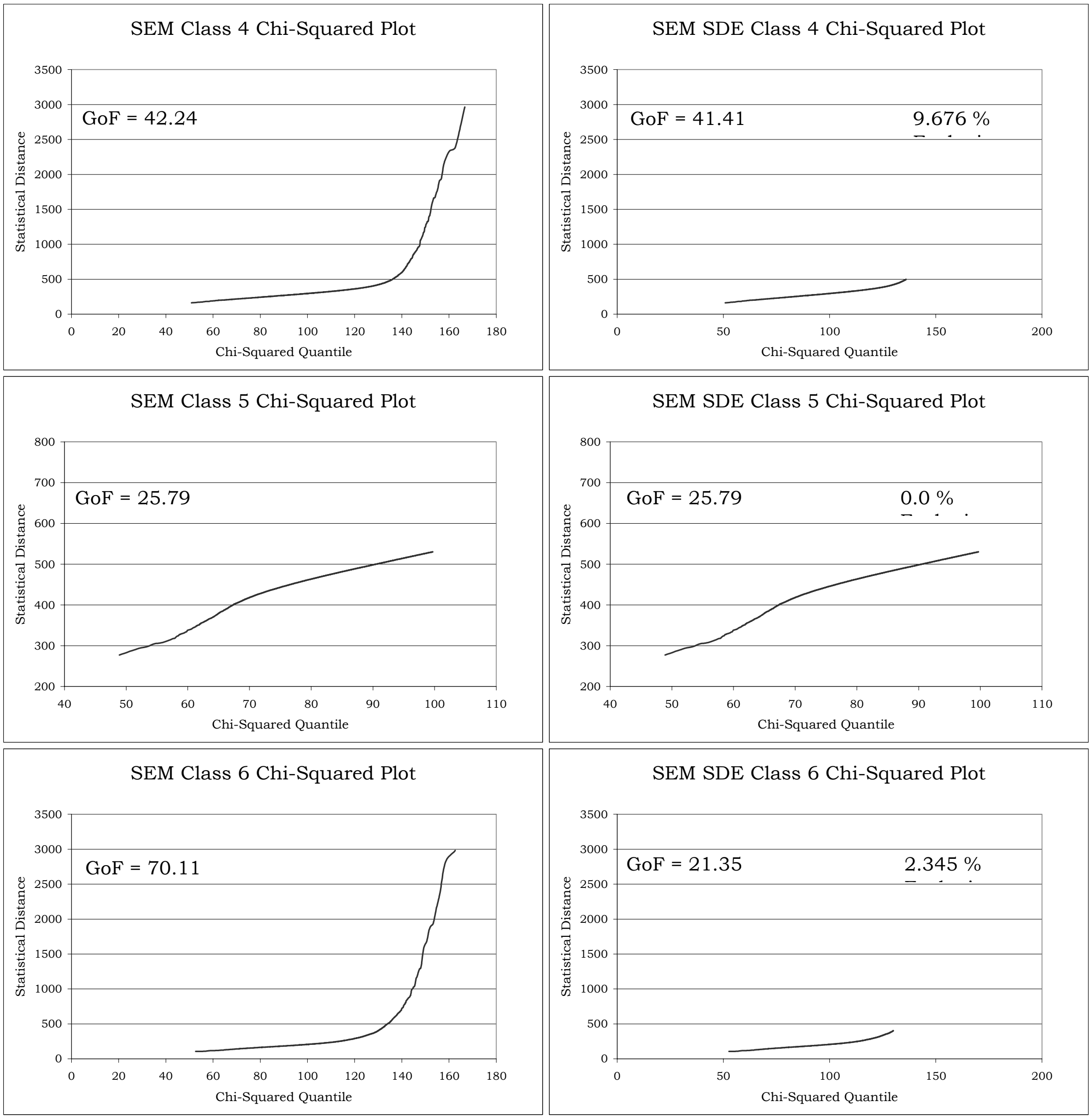


Figure A.18 SDE Chi-Squared Example for SEM Classes 4 – 6

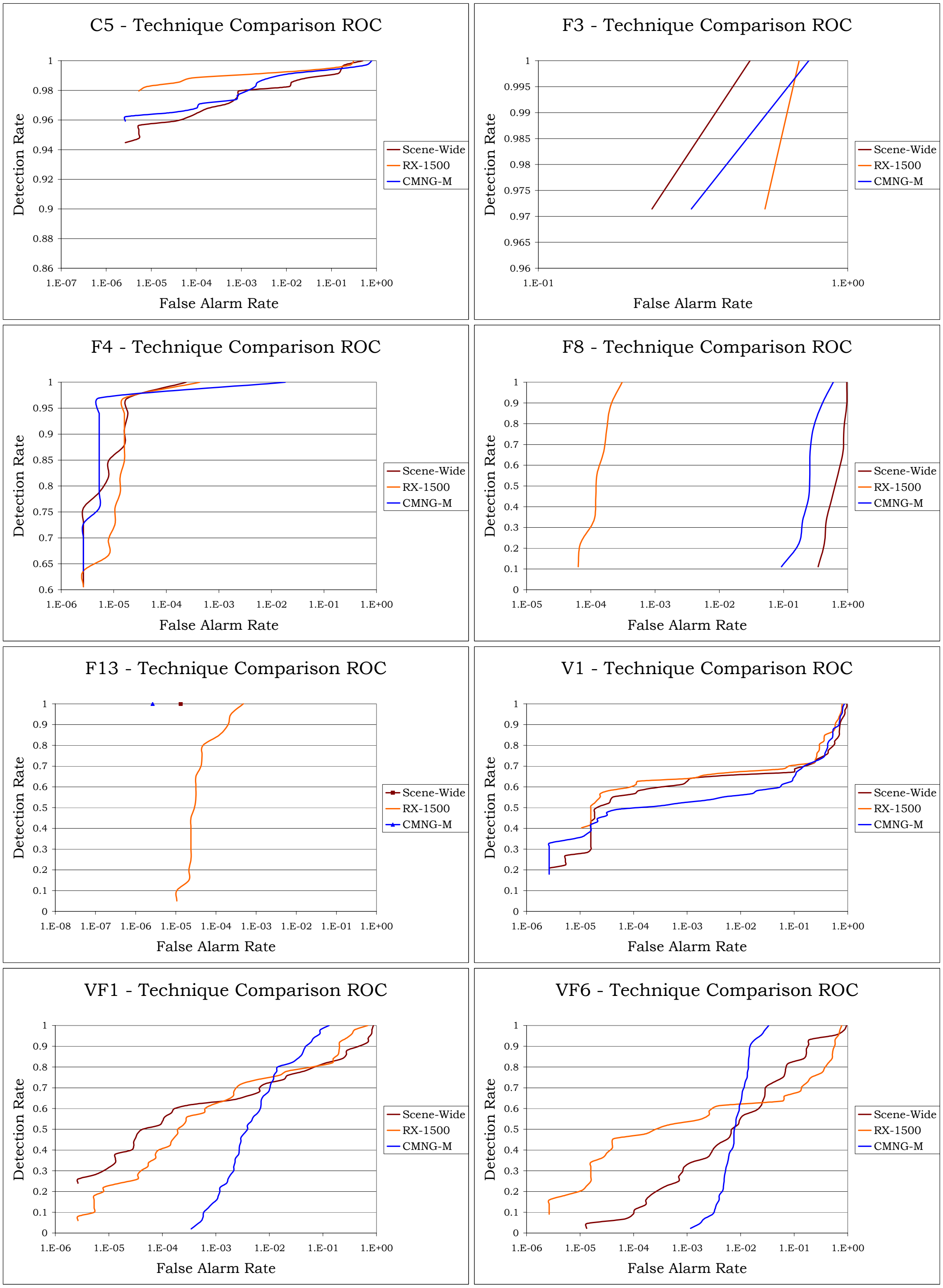


Figure A.19 Run05 Technique Comparison ROC Curves

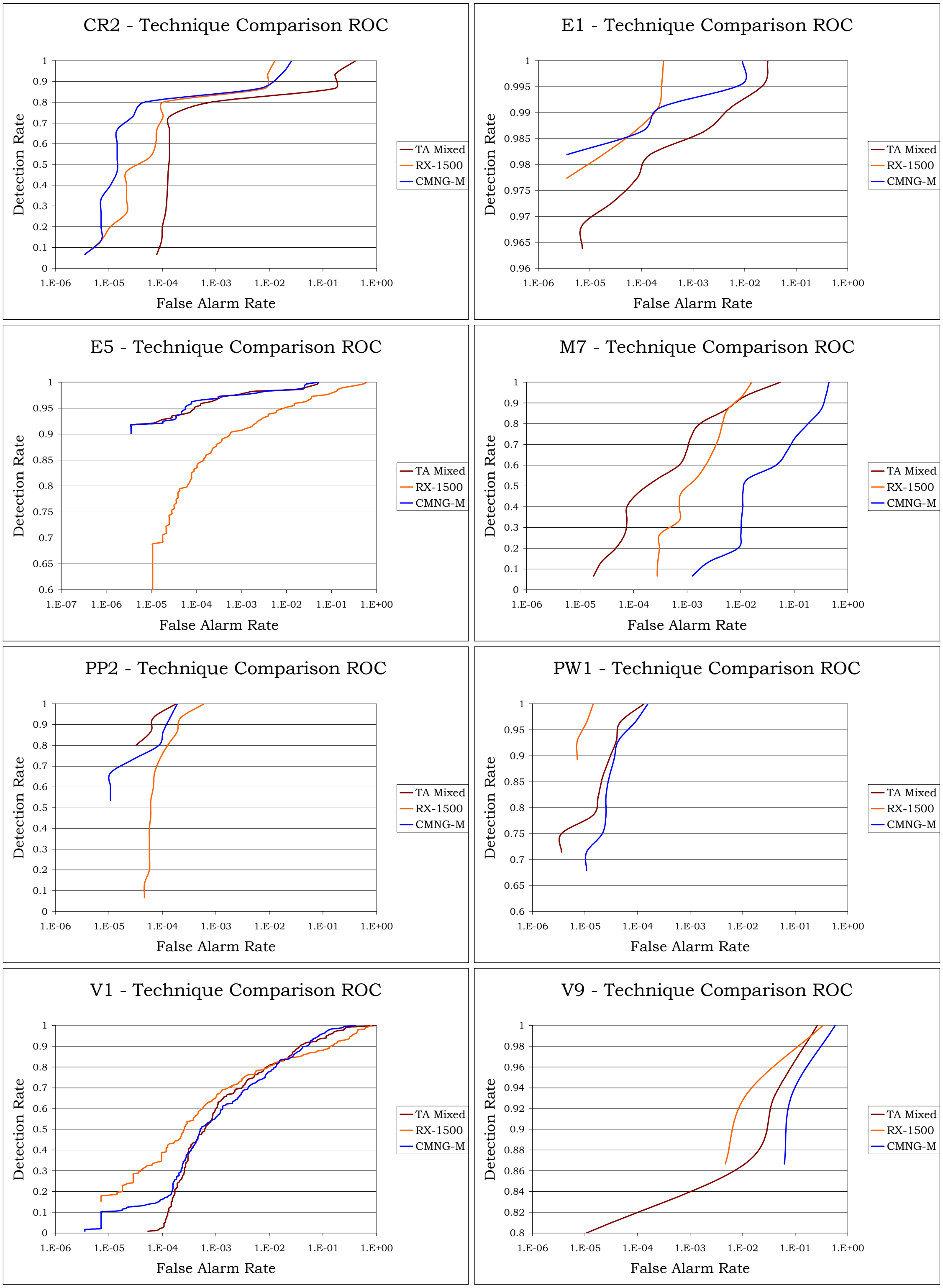


Figure A.20 Run03 Technique Comparison ROC Curves

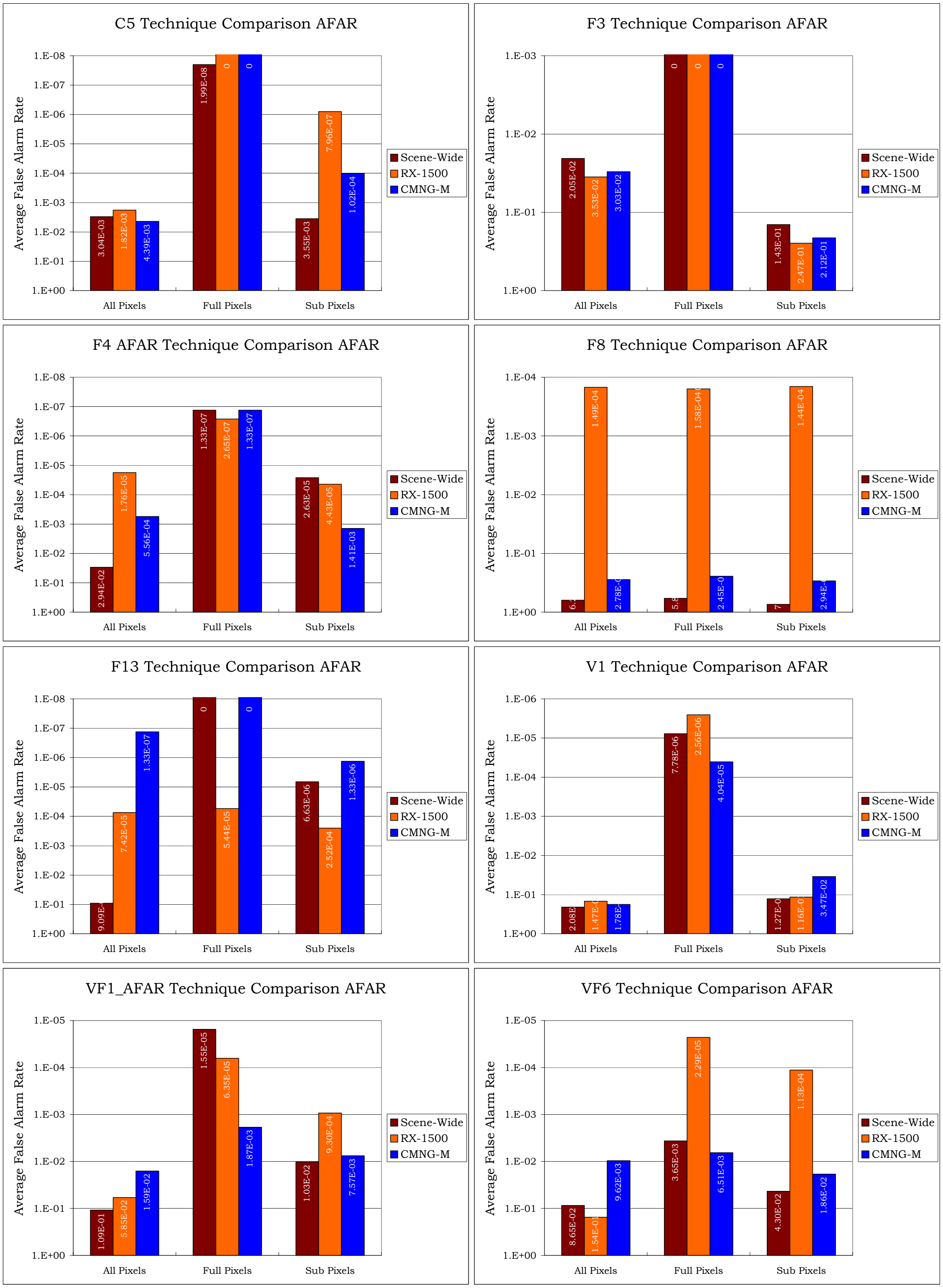


Figure A.21 Run05 Technique Comparison AFAR

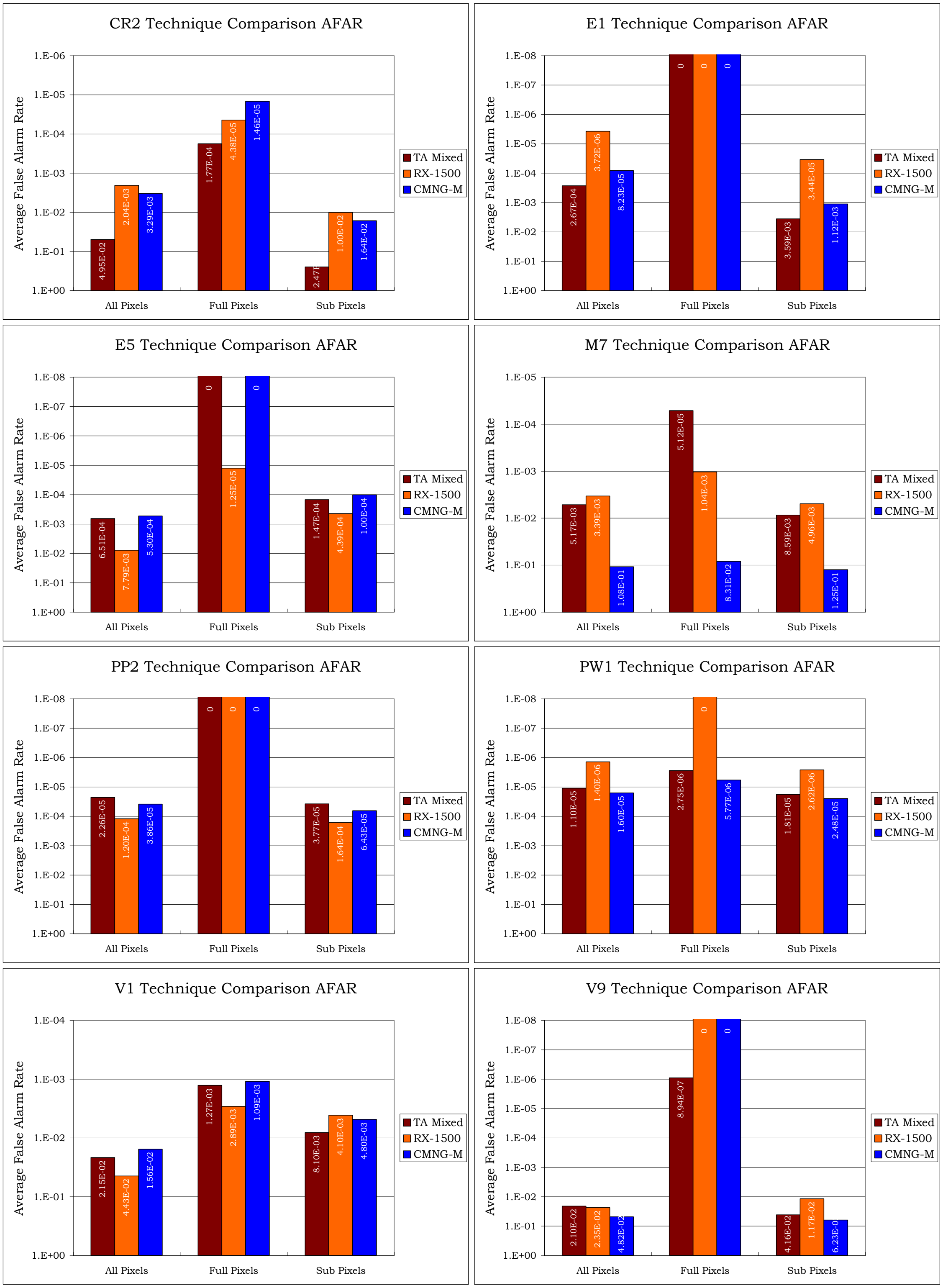


Figure A.22 Run03 Technique Comparison AFAR

APPENDIX B – IDL CODE

```
;+
;Stochastic Expectation Maximization (SEM) Classification Algorithm
;Written by:  Jason E. West
;6 Oct 04
;14Feb05 -- Changed to keep all classmaps
;7Mar05 -- Changed to allow input classmap
;13May05 -- Added Statistical Distance Classification

;***********************************************************************************************
;                              INPUTS
;
;   Image   - A multi/hyperspectral data set as [x,y,bands]
;   M        - The upper bound number of classes [choose high]
;   delP     - The threshold for minimum class probability, [0 to 1]
;               i.e. 0.1 eliminates classes with less than 10% of the image pixels
;   delu     - The convergence threshold for change in class means, [0 to 1]
;               i.e. 0.1 stops the algorithm if means change less than 10%
;   MaxIt   - The maximum number of iterations [choose high], if you don't see "SEM
;               has converged after X iterations" is means you never reached the
;               convergence criteria and you should raise MaxIt
;   CMapIn- Can include a starting point class map to speed things along
;
;***********************************************************************************************
;                              OUTPUT
;
;   ClassMap- An integer array [MaxIt,x,y] with each class assigned an integer
;
;***********************************************************************************************
;                              KEYWORDS
;
;   Radiance - Set if the image is in radiance units (reflectance is default)
;                  This is needed for carrying units (large or small) in the determinant.
;   SDC        - Set to perform classification based on the statistical distance only
;
;***********************************************************************************************
;                              SECTIONS
;
;   INITIALIZATION        - Assigns random statistics if not previously accomplished
;   MAXIMIZATION         - Classifies pixels into class with maximum probability
;   STOCHASTIC           - Calculates class statistics for each class
;   delP THRESHOLD TEST  - Tests classes for minimum size requirement
;   ESTIMATION                  - Generates new probabilities from class statistics
;   delu COVERGENCE TEST - Tests classes for min change in mean to exit
;
;***********************************************************************************************
;-

Function SEM,ImageIn,M,delP,delu,MaxIt,CMapIn,Radiance=Radiance,SDC=SDC

;Size inputs and set some variables
print,'SEM started' + systime() & wait,1.0
StartTime=double(systime(1))

isize=size(ImageIn)
pixels=isize[1]*isize[2]
bands=fix(isize[3])
Image = ptr_new(FltArr(pixels,bands))
*Image = reform(ImageIn,pixels,bands)            ;re-index the image into a pointer
ClassMap = ptr_new(IntArr(pixels))               ;pointer to the class maps
Result = intarr(isize[1],isize[2],MaxIt)
EstFlag = 0                                      ;Flags entrance into estimation section
FailPFlag = 0                                    ;Flags failure of the delP threshold
InitFlag=0                                       ;Counts how many times SEM reinitialized
ConvergeInitFlag = 0                             ;Flags Initialization to prevent convergence
CMapInFlag = 0                                   ;Flags an input classmap
```

```
If n_elements(CMapIn) GT 0 Then Begin
(*ClassMap)[*,*] = CMapIn[*,*]
CMapInFlag = 1
EstFlag = 1
EndIf

device,decomposed=0 & loadct,12   ;Set up the display

;************************************************************************************************

For Iteration = 0,MaxIt-1 Do Begin

        If M EQ 1 Then Begin
           Iteration = MaxIt
           print,format='(5X,"Thresholds set too tightly. Image classified as a single class.")'
        EndIf

;************************************************************************************************
;INITIALIZATION

   ;Random Init -- generate uniformly distributed PDF for p(i|x)
   If EstFlag EQ 0 Then Begin      ;Only do this if estimation is not P/A
      pix=ptr_new(dblarr(pixels,M))  ;Points to p(i|x)
      InitFlag=InitFlag+1     ;Count number of Initializations
      ConvergeInitFlag = 1    ;Marks random initilization for (to skip) convergence test
      icsize=floor(pixels/M)     ;Take every Mth pixel and assign it to a class
      stragglers=Pixels MOD M  ;In case there aren't exactly M factorable pixels

      For i=0l,M-1 Do Begin
      For j=0l,icsize-1 Do Begin
           (*pix)[j*M+i,i]=1
      EndFor
      EndFor

       If Stragglers GT 0 Then Begin   ;This will pick up the straggler pixels and
            For k=0l,Stragglers-1 Do (*pix)[icsize*M+k,0]=1 ;assign them all to the first class
       EndIf
    EndIf

;************************************************************************************************
;MAXIMIZATION

   ;Classify pixels into class of max p(i|x)
   If CMapInFlag EQ 0 Then Begin
      For i=0l,pixels-1 Do Begin
        MaxPix=max((*pix)[i,*])
        NewClass=Max((where((*pix)[i,*] EQ MaxPix)+1))
        (*ClassMap)[i]=NewClass
      Endfor
      ptr_free,pix   ;Once classification has been done, pix isn't needed
   EndIf Else CMapInFlag = 0

   Result[*,*,Iteration] = reform(*Classmap,isize[1],isize[2])

   If ConvergeInitFlag EQ 0 Then Begin ;Don't display randomly initialized classmaps
      window,iteration,xsize=isize[1],ysize=isize[2],title='Classmap'+string(iteration),xpos=0,ypos=0
      tvscl,Reform(*ClassMap,isize[1],isize[2]) & wait,1.0
   EndIf

;************************************************************************************************
;STOCHASTIC

   PrevCMean = ptr_new(dblarr(M,bands))

   If Iteration GT 0 Then *PrevCMean = *CMean    ;keep previous CMean for convergence test
   If Iteration GT 0 Then ptr_free,CMean,CCov,pi
   ;Calculate the class statistics
   CMean=ptr_new(dblarr(M,bands))              ;Pointer to M class mean vectors
   CCov=ptr_new(dblarr(bands,bands,M))         ;Pointer to M class covariance matricies
   pi=ptr_new(dblarr(M))                       ;Pointer to M class probabilities p(i)
   ClassSize = ptr_new(dblarr(M))

   For i=0,M-1 Do Begin
      ind=where(*ClassMap EQ i+1,ClassCounter)
      (*ClassSize)[i] = ClassCounter

      If (*ClassSize)[i] GT 1 Then Begin   ;Don't run calcs for empty classes
           For j=0,bands-1 Do (*CMean)[i,j]=mean((*image)[ind,j],/double)               ;Mean
           (*CCov)[*,*,i]=correlate(transpose((*image)[ind,*]),/covariance,/double)      ;Covariance
           (*pi)[i]=(*ClassSize)[i]/double(pixels)                                       ;p(i)
      EndIf Else (*pi)[i]=-1             ;Set p(i) = -1 for empty classes (error flag)

   Endfor

print,Format='(5X,"The p(i) for iteration",1X,I0,1X,"is")',iteration
print,*pi & wait,0.01

;****************************************************************************************************
;delP THRESHOLD TEST

If Min(*pi) GT delP Then Begin   ;only run estimation if class sizes meet threshold

;****************************************************************************************************
;ESTIMATION

   ;GML classifier from Schott p 274,
   ;p(i|x) ~ Di = ln[p(i)]- L/2*ln(2!pi) - 1/2*ln|S| - 1/2*(X-M)'S^-1(X-M)
   ;implemented as:  D1        D2            D3                D4
   ;only D4 is used if SDC keyword is set.

   EstFlag=1            ;Sets Flag so that random initialization is not performed

   pix=ptr_new(dblarr(pixels,M))           ;Points to p(i|x) (previous pix has been freed)
   LUDCCov=Ptr_new(dblarr(bands,bands,M))  ;Points to LU Decomposition Class Covariance
   ICCov=ptr_new(dblarr(bands,bands,M))    ;Points to the inverse of LUDCCov

   For i=0,M-1 Do Begin
      D1=alog((*pi)[i])
      D2=(bands*0.5)*alog(2*!DPi)

     If keyword_set(Radiance) Then Begin
         ;This calculates the determinant by LU decomposition carrying large numbers
          *LUDCCov=(*CCov)[*,*,i] ;Copy the class covariance into a temp matrix
          LA_LUDC,*LUDCCov,LUTemp  ;Perform LU decomposition on that matrix
          CarryFactor = Round(ALog10(Max(Diag_Matrix(*LUDCCov))))-1;Find N in *10^N for the diagonal
          ;Divide the diagonal by that factor, take the product (i.e LA_Determ), compensate for the carry factor
          ;remember ln(a) + ln(b) = ln(ab)
               D3=double(0.5)*((alog(abs(Product(Diag_Matrix(*LUDCCov)/$
                    (double(10)^carryFactor)))))+alog(double(10)^CarryFactor))
      EndIf Else Begin
          ;This calculates the determinant by LU decomposition carrying small numbers
          *LUDCCov=(*CCov)[*,*,i] ;Copy the class covariance into a temp matrix
          LA_LUDC,*LUDCCov,LUTemp  ;Perform LU decomposition on that matrix
          CarryFactor = Round(ALog10(Min(abs(Diag_Matrix(*LUDCCov)))))-1;Find N in *10^-N for the diagonal
          ;Divide the diagonal by that factor, take the product (i.e LA_Determ), compensate for the carry factor
          ;again ln(a) + ln(b) = ln(ab)
               D3=double(0.5)*((alog(abs(Product(Diag_Matrix(*LUDCCov)/$
                    (double(10)^carryFactor)))))+alog(double(10)^CarryFactor))
      EndElse

     ;Invert the covariance using IDL Invert for N > bands or PseudoInvert for N < bands
      If (*ClassSize)[i] GE Bands $
      Then *ICCov=Invert((*CCov)[*,*,i])$
      Else *ICCov=(Invert(transpose((*CCov)[*,*,i])##(*CCov)[*,*,i])) ## transpose((*CCov)[*,*,i])
      For j=0l,pixels-1 Do Begin

      D4 = double(0.5)*(transpose((*Image)[j,*]-(*CMean)[i,*]) ## $
                    reform(*ICCov,bands,bands) ## ((*Image)[j,*]-(*CMean)[i,*]))

      ;Combine the four factors (or just take D4 for SDC)
      IF Keyword_Set(SDC) Then (*pix)[j,i] = D4  Else (*pix)[j,i] = D1 - D2 - D3 - D4

   EndFor
   EndFor

ptr_free,LUDCCov,ICCov,ClassSize

;****************************************************************************************************

EndIf Else Begin  ;End TEST for delP threshold - if max(pi) LT delP, estimation has been skipped
          M = M - 1       ;Decrement number of classes
          EstFlag=0       ;Reset Estimation Flag to ensure random re-initialization
          FailPFlag = 1   ;Mark Failure of delP threshold test and skips Convergence Test
EndElse

;****************************************************************************************************
;delu CONVERGENCE TEST

If iteration EQ 0 Then FailPFlag=0 Else Begin ;PrevCMean is not defined for 0th iteration

  If FailPFlag GT 0 Then FailPFlag = 0 Else Begin ;Don't converge/reset flag if you've just failed the delP test

      If ConvergeInitFlag GT 0 Then ConvergeInitFlag=0 Else Begin ;Don't converge/reset flag on randomly
                                                                   initialized data
          MeanCMean=ptr_new(dblarr(M))
          MeanPMean=ptr_new(dblarr(M))
          For i=0,M-1 Do Begin
             (*MeanCMean)[i]=Mean((*CMean)[i,*])
             (*MeanPMean)[i]=Mean((*PrevCMean)[i,*])
          EndFor
          ;Absolute Change Fraction for the Mean Mean
          ACFM = (abs(*MeanCMean - *MeanPMean)) / *MeanPMean
          print,format='(5X,"The max ACFM for iteration",1X,I0,1X,"is")',iteration
          Print,Max(ACFM) & wait,1.0
          If max(ACFM) LT delu Then Begin
             print,format='(5X,"SEM has converged after",1X,I0,1X,"iterations.")',Iteration
             wait,1
             Iteration=MaxIt
          EndIf
          ptr_free,PrevCMean,MeanCMean,MeanPMean

    EndElse
  EndElse
EndElse   ;End TEST for convercence

;****************************************************************************************************

ENDFor ;End iterations - increments iteration #
;****************************************************************************************************
;RESULTS

print,format='(5X,"SEM Initialized",1X,I0,1X,"times for this run.")',InitFlag
print,format='(5X,"SEM found",1X,I0,1X,"classes in the image.")',M
EndTime = double(systime(1))
RunTime = EndTime - StartTime
print,format='("SEM completed",2X,A24,2X,"and lasted",1X,I5,1X,"seconds.")',systime(),RunTime

Return,Result

ptr_free,Image,ClassMap,CMean

END ;SEM
```

```
;+
;RX Sliding Window Detection
;Written by:  Jason West
;18 June 05

;*******************************************************************************************
;                          FUNCTION
;
;   This program moves a sliding detection window over an HSI image, and
;   runs a spectral matched filter on the center pixel of the window using a
;   mean calculated with pixels outside of the detection window, but inside
;   of a specified mean window, and a covariance calculated with pixels
;   outside of the detection window, but inside of a specified covariance window.
;
;   The algorithm contains no spatial matched filter, but the size of the detection
;   window should be approximately twice the size of the target to ensure no target
;   pixels are included in the covariance calculation.
;
;          ______________________________
;         |                1                 |Covariance Window
;         |  1  ___________________          |
;         |    |           1           |Mean Window
;         |    |       __________      |     |
;         |    |      |           |Exclusion Window
;         |    |      |           |     |     |
;         | 1 |      |      X  Detection Pixel
;         |    |      |           |     |     |
;         |    |  1  |_________|  1 |  1  |
;         |    |                       |     |
;         |    |___________________|     |
;         |          1             1         |
;         |____________________________|
;
;
;
;*******************************************************************************************
;                          INPUTS
;
;   Image         - A multi/hyperspectral data set as [x,y,bands].
;   Target         - A target spectra as [1,bands].
;   Cov_Win      - The odd integer side-length of the covariance window to be used (in
;                      pixels).  Should be equal to the sqrt of the number of bands in the
;                      image plus the size of Det_Win AT A MINIMUM.
;   Mean_Win  - The odd integer side-length of the mean window to be used (in pixels).
;   Exl_Win      - The odd integet side-length of the exclusion window to be used (in
;                      pixels).  Should be at least twice the size of the target.
;   XMap          - Location of possible target pixels to be excluded from the background.
;                      where pixels of value NE 0 are excluded and EQ 0 of kept.
;   CCov           - A common covariance to be used for all windows.  i.e. using only a
;                      sliding mean.
;
;*******************************************************************************************
;                          KEYWORDS
;
;   Low_Contrast - Set to perform mean subtraction on only the test pixel and not
;            the target spectrum.  May be used for extrememly low contrast targets.
;
;*******************************************************************************************
;                          OUTPUT
;
;   DetRes - An array [x,y] giving the target detection statistic.
;
;*******************************************************************************************
;                          SECTIONS
;
;   SetUp       - Reformat/size/test inputs
;   Detection  - Calculate covariance and run detection for all windows
;   Results     - Stop the clock, print the time, return the results
;-
```

```
Function RX,Image,Target,Cov_Win,Mean_Win,Exl_Win,XMap,CCov,Low_Contrast=Low_Contrast

;************************************************************************************************
;                                SETUP

;Check inputs and set some variables

StartTime = double(systime(1))  ;start the clock
isize=size(image)     ;size the image
ix=isize[1]
iy=isize[2]
bands=isize[3]
pixels=ix * iy
Cov_Win = (2.0 * Floor(Cov_Win/2.0)) + 1.0   ;ensure window sizes are odd
Mean_Win = (2.0 * Floor(Mean_Win/2.0)) + 1.0
Exl_Win = (2.0 * Floor(Exl_Win/2.0)) + 1.0

print,format='("Targets in the outer",1X,I0,1X,"pixels will not be detected!")',Floor(Cov_Win / 2.0) & wait,0.1

BackWindow = ptr_new(fltarr(bands,Cov_Win^2.0 - Exl_Win^2.0))   ;The sliding covariance window
MeanWindow = ptr_new(fltarr(bands,Mean_Win^2.0 - Exl_Win^2.0))  ;The sliding mean window
XWindow = ptr_new(fltarr(Cov_Win^2.0 - Exl_Win^2.0))            ;The exclusion map window for covariance
XMWindow = ptr_new(fltarr(Mean_Win^2.0 - Exl_Win^2.0))          ;The exclusion map window for mean
BackCov = ptr_new(dblarr(bands,bands))                          ;The covariance of BackWindow
BackCovI = ptr_new(dblarr(bands,bands))                         ;Inverse of the covariance
BackMean = ptr_new(dblarr(1,bands))                             ;Mean of Backwindow
DetRes = ptr_new(dblarr(ix,iy))                                 ;The detection results
NoXSlides = (ix - Cov_Win - 1)                   ;how many slides in the x direction
NoYSlides = (iy - Cov_Win - 1)                   ;how many in y
NoWindows = (NoXSlides+1) * (NoYSlides+1)   ;how many total subsets
ComAdj = ((Cov_Win - Exl_Win)/2.0)               ;common index adjustment for the cov window
CenPix = ceil(Cov_win/2)                         ;the detection pixel
Size13 = Cov_Win * ComAdj                        ;window sizes for the 4 cov windows
Size24 = Exl_Win * ComAdj
Size13M = Mean_Win * ((Mean_Win - Exl_Win)/2.0) ;window sizes for the 4 mean windows
Size24M = Exl_Win * ((Mean_Win - Exl_Win)/2.0)
MeanAdj = ((Cov_Win - Mean_Win)/2.0)                  ;common index adjustment for the mean window
WinCount = 0                                          ;count the slides to track progress
;************************************************************************************************
;                                DETECTION

StartItTime = double(systime(1))  ;start the clock on iterations

For j = 0l,NoYSlides-1 Do Begin   ;These increment to slide the window
For i = 0l,NoXSlides-1 Do Begin

   ;This defines the start/stop of the background window
   xstart1 = i & xstart24stop1 = i + ComAdj & xstart3stop24 = i + ComAdj + Exl_Win & xstop3 = i + Cov_win
   ystart123 = j & ystart4 = j + ComAdj + Exl_Win & ystop134 = j + Cov_Win & ystop2 = j + ComAdj

   If n_elements(CCov) EQ 0 Then Begin
      ;peel out the covariance window
      (*BackWindow)[*,0:Size13-1] = $
            transpose(reform(Image[xstart1:xstart24stop1-1,ystart123:ystop134-1,*],Size13,bands))
      (*BackWindow)[*,Size13:Size13+Size24-1]= $
            transpose(reform(Image[xstart24stop1:xstart3stop24-1,ystart123:ystop2-1,*],Size24,bands))
      (*BackWindow)[*,Size13+Size24:2.0*Size13+Size24-1] = $
            transpose(reform(Image[xstart3stop24:xstop3-1,ystart123:ystop134-1,*],Size13,bands))
      (*BackWindow)[*,2.0*Size13+Size24:2.0*Size13+2.0*Size24-1] = $
            transpose(reform(Image[xstart24stop1:xstart3stop24-1,ystart4:ystop134-1,*],Size24,bands))
      ;find pixels to exclude
      (*XWindow)[0:Size13-1] = $
            transpose(reform(XMap[xstart1:xstart24stop1-1,ystart123:ystop134-1],Size13))
      (*XWindow)[Size13:Size13+Size24-1]= $
            transpose(reform(XMap[xstart24stop1:xstart3stop24-1,ystart123:ystop2-1],Size24))
      (*XWindow)[Size13+Size24:2.0*Size13+Size24-1] = $
            transpose(reform(XMap[xstart3stop24:xstop3-1,ystart123:ystop134-1],Size13))
      (*XWindow)[2.0*Size13+Size24:2.0*Size13+2.0*Size24-1] = $
            transpose(reform(XMap[xstart24stop1:xstart3stop24-1,ystart4:ystop134-1],Size24))
```

```
     ;include only the good pixels in the covariance
     Include = Where(*XWindow EQ 0,N_Included)
     CovData = (*BackWindow)[*,Include]
     ;calculate and invert the covariance
     *BackCov = Correlate(CovData,/Covariance,/Double)
          If N_Included LT Bands $
            Then *BackCovI = Invert(*BackCov ## Transpose(*BackCov)) ## Transpose(*BackCov) $
            Else *BackCovI = Invert(*BackCov,/double)
   EndIf Else Begin
          ;invert the stationary covariance
          *BackCov = CCov
          *BackCovI = Invert(CCov,/double)
   EndElse

   ;find the mean window pixels
   (*MeanWindow)[*,0:Size13M-1] = transpose(reform(Image$
          [xstart1+MeanAdj:xstart24stop1-1,ystart123+MeanAdj:ystop134-1 MeanAdj,*],Size13M,bands))
   (*MeanWindow)[*,Size13M:Size13M+Size24M-1]= transpose(reform(Image$
          [xstart24stop1:xstart3stop24-1,ystart123+MeanAdj:ystop2-1,*],Size24M,bands))
   (*MeanWindow)[*,Size13M+Size24M:2.0*Size13M+Size24M-1] = transpose(reform(Image$
          [xstart3stop24:xstop3-1-MeanAdj,ystart123+MeanAdj:ystop134-1-MeanAdj,*],Size13M,bands))
   (*MeanWindow)[*,2.0*Size13M+Size24M:2.0*Size13M+2.0*Size24M-1] = transpose(reform(Image$
          [xstart24stop1:xstart3stop24-1,ystart4:ystop134-1-MeanAdj,*],Size24M,bands))
   ;find the mean window pixels to exclude
   (*XMWindow)[0:Size13M-1] = transpose(reform(XMap$
          [xstart1+MeanAdj:xstart24stop1-1,ystart123+MeanAdj:ystop134-1-MeanAdj],Size13M))
   (*XMWindow)[Size13M:Size13M+Size24M-1]= transpose(reform(XMap$
          [xstart24stop1:xstart3stop24-1,ystart123+MeanAdj:ystop2-1],Size24M))
   (*XMWindow)[Size13M+Size24M:2.0*Size13M+Size24M-1] = transpose(reform(XMap$
          [xstart3stop24:xstop3-1-MeanAdj,ystart123+MeanAdj:ystop134-1-MeanAdj],Size13M))
   (*XMWindow)[2.0*Size13M+Size24M:2.0*Size13M+2.0*Size24M-1] = transpose(reform(XMap$
          [xstart24stop1:xstart3stop24-1,ystart4:ystop134-1-MeanAdj],Size24M))
   ;include only good pixels in the mean data
   Include = Where(*XMWindow EQ 0,N_Included)
   MeanData = (*MeanWindow)[*,Include]
   ;calculate the spectral mean
   For m = 0l,bands-1 Do (*BackMean)[0,m] = Mean((MeanData)[m,*])
   ;run detection on the test pixel
   If keyword_set(low_contrast) Then T = Target Else T = Target - *Backmean
   dT = reform(T,bands,1)
   x = reform(Image[i+CenPix,j+CenPix,*],1,bands) - *BackMean & xT = reform(x,bands,1)

   (*DetRes)[i+CenPix,j+CenPix] = (dT ## *BackCovI ## x)^2 / $
                                  ((dT ## *BackCovI ## T) * (1.0 + (xT ## *BackCovI ## x)))
    ;Keep track of progress
    WinCount = WinCount + 1
    If WinCount EQ floor(NoWindows/100.0) Then Begin
      HundTime = double(systime(1)) & TimeToHund = HundTime - StartItTime
      Print,Format='("RX is 1% complete, and will take approximately",$
                    1X,I8,1X,"seconds to complete.")',TimeToHund * 100.0
      Beep & wait,0.1 & WinCount = Pixels
    EndIf
EndFor
EndFor

ptr_free,BackWindow,MeanWindow,BackCov,BackCovI ;release pointers

;***********************************************************************************************
;                              RESULTS

EndTime = systime(1)
RunTime = EndTime - StartTime
print,format='("RX completed  ",2X,A24,2X,"and lasted",1X,I5,1X,"seconds.")',systime(),RunTime
Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5

Return, *DetRes
ptr_free,DetRes

END; RX
```

```
;+
;Class Mean Neighbor Guided Detection
;Jason E. West
;16 Nov 2004
;17 Jul 2005, modified to add Low Contrast keyword and allow for unclassified class in class map
;
;This program runs a sliding square ring across a classification image to derive a background from
;a the classes found in the ring, and then uses that background in the GLRT detector.  The user
;specifies a stand off distance away from the center detection pixel to form the ring.  The basis
;image and target, as well as the classification resulting image and an array of class covariances
;must be provided.  These can be obtained by running a classification algorithm and then the
;Class_Cov.pro code.  The result is a detection statistic image.
;
;*************************************************************************************************************
;INPUTS
;
;   Image              - A multi/hyperspectral image as [x,y,bands]
;   Target             - A target spectrum as [1,bands]
;   Class_Image     - A classification result image as [x,y] where values correspond to class
;   Class_CovMeans- An array of covariances and means as [bands+1,bands,classes] where the class
;                            mean is located [0,*,class]
;   Stand_Off         - The approximate side dimension of a target
;
;*************************************************************************************************************
;OUTPUT
;
;   DetRes      - The detection result
;
;*************************************************************************************************************
;KEYWORDS
;
;   Mode               - Set to use the mode class instead of linear mixtures of class stats
;   Low_Contrast   - Set when searching for low contrast targets to omit mean subtraction from d.
;
;*************************************************************************************************************
;-

Function CMNG, Image, Target, Class_Image, Class_CovMeans, Stand_Off, &
             Low_Contrast=Low_Contrast, Mode=Mode
;***************************************************************************************************
;Size up and define variables

StartTime = double(systime(1))                     ;start the clock
isize=size(image)                                  ;size the image
ix=isize[1]
iy=isize[2]
bands=isize[3]
pixels=ix * iy
scc = size(class_covmeans)                         ;determine # of classes
classes = scc[3]
Win = (2.0 * Stand_Off) + 3                        ;window side size
BRPix = 8 * (Stand_Off + 1)                        ;number of pixels in background ring
BackRing = intarr(BrPix+Classes+1)                 ;The sliding ring
DetPix = dblarr(1,bands)                           ;The detection pixel
BackCov = dblarr(bands,bands)                      ;The covariance derived from BackRing
BackCovI = dblarr(bands,bands)                     ;The inverted covariances
BackMean = dblarr(1,bands)                         ;The mean derived from BackRing
DetRes = dblarr(ix,iy)                             ;The detection results
NoXSlides = ix - Win - 2                           ;how many slides in the x direction
NoYSlides = iy - Win - 2                           ;how many in y
NoWindows = (NoXSlides+1) * (NoYSlides+1)   ;how many total slides

If Keyword_Set(Mode) Then Begin
for i = 0,classes-1 do BackCovI[*,*,i] = Invert(Class_CovMeans[1:Bands,*,i],/double)
EndIf
WinCount = 0
```

```
;****************************************************************************************************
;Slide window/calculate backgroud/run detector

StartItTime = double(systime(1))                     ;start the clock on iterations

For j = 0,NoYSlides-1 Do Begin                     ;increment the sliding the window position
For i = 0,NoXSlides-1 Do Begin

   ;gather pixels into the background ring
   BackRing[0:win-1] = Class_Image[i,j:j+win-1]
   BackRing[win:2*win-1] = Class_Image[i+win-1,j:j+win-1]
   BackRing[2*win:3*win-3] = Class_Image[i+1:i+win-2,j]
   BackRing[3*win-2:BRPix-1] = Class_Image[i+1:i+win-2,j+win-1]
   BackRing[BRPix:BRPix+classes] = indgen(classes+1)    ;be sure to include each integer

   H_BR = Histogram(BackRing) - 1   ;histogram and subtract the indgen-added occurance

   ;eliminate null class values and select class 1 in case of full null backgrounds.
   H_BR[0] = 0
   If Total(H_BR) EQ 0 Then H_BR[1] = 1

   If Keyword_Set(Mode) Then Begin
   Winner = min(where(H_BR EQ Max(H_BR))) ;min incase there is a tie
   BackMean = Class_CovMeans[0,*,Winner-1]
   EndIf Else Begin

   BackCov[*,*] = 0.0 & BackMean[*,*] = 0.0

   ;Sum the class means and covariances in ratios
   For k = 1,classes-1 Do Begin
      BackCov =  BackCov + ((H_BR[k] / double(BRPix)) * Class_CovMeans[1:bands,*,k-1])
      BackMean =  BackMean + ((H_BR[k] / double(BRPix)) * Class_CovMeans[1:bands,*,k-1])
   EndFor

   ;find the center pixel and run detector
   DetPix = reform(Image[i+Stand_Off+1,j+Stand_Off+1,*],1,bands)
   x = DetPix - BackMean
   xT = reform(x,bands,1)
   IF keyword_set(Low_Contrast) Then d = Target Else d = Target - BackMean
   dT = reform(d,bands,1)
   IF KeyWord_Set(Mode) Then SI = BackCovI[*,*,Winner-1] Else SI = Invert(BackCov,/double)

   DetRes[i+Stand_Off+1,j+Stand_Off+1] = (dT ## SI ## X)^2.0 / ((dT ## SI ## d) * (1.0 + (xT ## SI ## x)))

;****************************************************************************************************
;Keep track of progress

   WinCount = WinCount + 1
   If WinCount EQ floor(NoWindows/10) Then Begin
   HundTime = double(systime(1)) & TimeToHund = HundTime - StartItTime
   Print,Format='("CMNG is 1% complete, and will take approximately",1X,I8,1X,$
                      "seconds to complete.")',TimeToHund * 100.0
   WinCount = Pixels
   Beep & wait,1.0
   EndIf

EndFor
EndFor

;****************************************************************************************************
;Return and End

EndTime = double(systime(1))
RunTime = EndTime - StartTime
print,format='("A_RX completed  ",2X,A24,2X,"and lasted",1X,I5,1X,"seconds.")',systime(),RunTime

Return, DetRes
Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5 & Beep & wait,0.5
END ;CMNG
```

```
;+
;Class Mean & Covariance Estimator with Statistical Distance and Target Exclusion
;Jason E West
;28 Feb 2004

;****************************************************************************************************
;                              FUNCTION
;
;   This program calculates the class statistics of an HSI image given a classmap.
;   Statistical distance exclusion and target exclusion pre-filtering can also be
;   performed.
;
;****************************************************************************************************
;                              INPUTS
;
;   Image              - the full image as [bands,pixels]
;   ClassImage         - the classified image as [x,y] where values are 1,2,3...K
;                        corresponding with classes
;   Exclude            - the fraction of pixels to exclude from each class, i.e. 0.1 means
;                        10% of the outer pixels will be excluded
;   XMap               - target exclusion map where 1 denotes pixles to be excluded
;
;****************************************************************************************************
;                              OUTPUT
;
;   Result             - class means and covariances as [bands+1,bands,classes]
;                        where [0,*,*] are class means
;
;****************************************************************************************************
;                              KEYWORDS
;
;   Normality          - Set to run the Chi-Squared test for MVN along with the calculation
;
;****************************************************************************************************
;                              USES
;   Chi-Squared.pro
;-

Function Class_MeanCov_SDEX,Image,ClassImageIn,Exclude,XMap,Normality=Normality

SImage = Size(image)
pixels = SImage[2] & bands = SImage[1] & classes = Max(ClassImageIn)

ClassImage = ClassImageIn
If n_elements(XMap) GT 0 Then Begin
   Targs = Where(XMap GT 0)
   ClassImage[Targs] = 0
EndIf

ClassCovs = ptr_new(dblarr(bands,bands,classes)) & ClassMeans = ptr_new(dblarr(1,bands,classes))

For i = 1,Classes Do Begin

    CPixels = Where(ClassImage EQ i,PixCount)
    If PixCount GT 0 Then Begin
        TCovImage =Image[*,CPixels] & TMean = dblarr(bands,1)
        TempCov = Correlate(TCovImage,/Covariance,/Double) & ITempCov = Invert(TempCov)
        For j=0,Bands-1 Do TMean[j,0] = Mean(TCovImage[j,*])
    EndIf Else Print,'Class'+string(i)+'has no remaining pixels'

    If Exclude GT 0 Then Begin
        StatDist = ptr_new(dblarr(PixCount))
        For j = 0l,PixCount-1 Do Begin
              (*StatDist)[j] = ((TCovImage[*,j]-TMean[*,0]) ## ITempCov ## transpose(TCovImage[*,j]-TMean[*,0]))
        EndFor

        PixExl = floor(PixCount * Exclude) & ExlSort = reverse(Sort(*StatDist,/L64))
        CovImage = Image[*,ExlSort[0:PixExl]]
        (*ClassCovs)[*,*,i-1] = Correlate(CovImage,/Covariance,/Double)
        For j=0,Bands-1 Do (*ClassMeans)[0,j,i-1] = Mean(CovImage[j,*])
    EndIf Else Begin
        CovImage = TCovImage
        Print,'No pixels excluded by SDE.  Printed MVN results are both for Xmap exclusion.'
    EndElse

    If Keyword_Set(Normality) Then Begin
        Temp1 = Chi_Squared(TCovImage,100,/NoPlot,/Metric)&print,String(i)+'Class MVN is'+String(Temp1)
        Temp2 = Chi_Squared(CovImage,100,/NoPlot,/Metric)&print,String(i)+'Class SDE MVN is'+String(Temp2)
    EndIf

    ptr_free,StatDist

    Print,Format='("There are",1X,I0,1X,"pixels in class",1X,I0,1X,"and",1X,I0,1X,$
                        "of them were excluded.")',string(PixCount),string(i),string(PixCount*Exclude)

EndFor

Result = dblarr(Bands+1,Bands,Classes)
Result[0,*,*] = *ClassMeans
Result[1:Bands,*,*] = *ClassCovs
ptr_free,ClassCovs,ClassMeans

Return,Result

END ;Class_MeanCov_SDEX

;+
;Chi-Squared Test for Multivariate Normality (MVN)
;Jason West
;30 Nov 2004

;*********************************************************************************************
;                               FUNCTION
;This function will generate a Chi-Squared plot to test MVN of a data set (Data) with a given
;covariance (Covar), and will return the normalized chi-squared goodness of fit value for the plot.
;
;*********************************************************************************************
;                              INPUT
;
;   Data         - A mulitvariate data set as [bands,pixels]
;   MeanCov  - The mean and covariance of that data set (will be calculated if not included)
;   dF            - The degrees of freedom of the data
;
;*********************************************************************************************
;                              OUTPUT
;
;   R              - The chi-squared goodness of fit measure or the data for the plot
;
;*********************************************************************************************
;                              KEYWORDS
;
;   NoPlot      - set this if you don't want to display the chi-squared plot
;   Metric      - set this to return only the GOF metric
;-

Function Chi_Squared,Data,dF,MeanCov,NoPlot=NoPlot,Metric=Metric,Verbose=Verbose

sData = Size(data)
pixels = sData[2] & bands = sData[1]

If n_elements(MeanCov) EQ 0 Then Begin
MeanCov[1:bands,*] = Correlate(Data,/Covariance,/Double)
MeanCov[0,*] = dblarr(bands,1) & For i = 0,bands-1 Do Mean_Vec[i,0] = Mean(Data[i,*])
EndIf

ICov = Invert(MeanCov[1:bands,*])
```

```
;Find the statistical distance values
R2 = dblarr(pixels)
For j = 0l,pixels-1 Do R2[j] = (Data[*,j]-MeanCov[0,*]) ## ICov ## transpose((Data[*,j]-MeanCov[0,*]))

;Rank order the R2 results
Sorter = Sort(R2) & SR2 = dblarr(pixels) & For j = 0l,pixels-1 Do SR2[j] = R2[Sorter[j]]

;Then find the corresponding Chi_Squared quantiles
Chi = dblarr(pixels) & For i = 1l,pixels Do Chi[i-1] = ChiSqr_CVF(((pixels - i + 0.5) / pixels),dF)

;Plot it unless the NoPlot keyword is set
If KeyWord_Set(NoPlot) Then Begin
EndIf Else Begin
    device,decomposed=0 & loadct,0 & window,/free,xsize=500,ysize=500,title='Chi-Squared Plot'
    plot,Chi,SR2,psym=3,color=0,background=255,title='Chi-Squared Plot',xtitle=$
                                                    'Chi-Squared Quantiles',ytitle='Data Distribution'
EndElse

;Print the Chi-Squared goodness of fit metric
If Keyword_Set(Verbose) Then Print,'The GOF Metric is' + String(Total(((SR2-Chi)^2.0)/Chi) / double(pixels))

Res = dblarr(2,pixels) & Res[0,*] = Chi & Res[1,*] = SR2

If Keyword_Set(Metric) Then Return,Total(((SR2-Chi)^2.0)/Chi) / double(pixels) Else Return,Res

END;Chi_Squared

;+
;Reciever Operator Characteristic (ROC) Curve Calculator
;Written by: Adam Cisz
;20Jan05
;Edits: Jason West, 16 May 05 add keywords
;
;;********************************************************************************
;                          FUNCTION
;
;     This program counts the number of false alarms that occur for the detection of
;     each target pixel in an image.  This data can then be used to create ROC curves.
;
;*******************************************************************************
;                          INPUTS
;
;     tdet   - detection results from previous algorithm (results in x,y format)
;     t_truth  - truth image (x,y format)(guard pixels = -1, target pixels > 0)
;
;*******************************************************************************
;                          OUTPUT
;
;     2 column array where first column lists percentage of targets detected and
;     second column lists percentage of false alarms encountered before detecting
;     each target pixel.
;
;*******************************************************************************
;                         KEYWORDS
;
;    sam     - set if low detection statistics correspond to a probable target
;    all_pix  - Set to perform analysis on all target pixels
;    (full/sub/glare/shadow)_pix - Set to perform analysis on full/sub/glare/shadow pixels only
;-

function roc_calc, tdet, t_truthin, sam=sam, &
       full_pix=full_pix,sub_pix=sub_pix,all_pix=all_pix,glare_pix=glare_pix,shadow_pix=shadow_pix

t_truth = t_truthin
; Define regions
targ    = where(T_Truth GT 0, targ_count) & full    = where(T_Truth EQ 8, full_count)
glare   = where(T_Truth EQ 6, glare_count) & shadow  = where(T_Truth EQ 4, shadow_count)
sub     = where(T_Truth EQ 2, sub_count) & guard   = where(T_Truth EQ -1,guard_count)
back    = where(T_Truth EQ 0, back_count)
```

```
If Keyword_Set(full_pix) Then Begin
    If Sub_Count GT 0 Then T_Truth[Sub] = -1
    If Glare_Count GT 0 Then T_Truth[Glare] = -1
    If Shadow_Count GT 0 Then T_Truth[Shadow] = -1
    res=transpose(dindgen(full_count,2))+1
    res[1,*] = 0
EndIf

If Keyword_Set(sub_pix) Then Begin
    If Full_Count GT 0 Then T_Truth[Full] = -1
    If Glare_Count GT 0 Then T_Truth[Glare] = -1
    If Shadow_Count GT 0 Then T_Truth[Shadow] = -1
    res=transpose(dindgen(sub_count,2))+1
    res[1,*] = 0
EndIf

If Keyword_Set(glare_pix) Then Begin
    If Full_Count GT 0 Then T_Truth[Full] = -1
    If Sub_Count GT 0 Then T_Truth[Sub] = -1
    If Shadow_Count GT 0 Then T_Truth[Shadow] = -1
    res=transpose(dindgen(glare_count,2))+1
    res[1,*] = 0
EndIf

If Keyword_Set (shadow_pix) Then Begin
    If Full_Count GT 0 Then T_Truth[Full] = -1
    If Sub_Count GT 0 Then T_Truth[Sub] = -1
    If glare_Count GT 0 Then T_Truth[glare] = -1
    res=transpose(dindgen(shadow_count,2))+1
    res[1,*] = 0
EndIf

If Keyword_Set (all_pix) Then Begin
   res=transpose(dindgen(targ_count,2))+1
   res[1,*] = 0
EndIf

tsize = size(tdet)                   ;calculate total number of pixels
tsize=tsize(4)

;Redefine regions
targ    = where(T_Truth GT 0, targ_count) & full     = where(T_Truth EQ 8, full_count)
glare   = where(T_Truth EQ 6, glare_count) & shadow  = where(T_Truth EQ 4, shadow_count)
sub     = where(T_Truth EQ 2, sub_count) & guard   = where(T_Truth EQ -1,guard_count)
back    = where(T_Truth EQ 0, back_count)

norm = tsize-targ_count-guard_count
no_fas = 0L
no_tgs = 0L
If Keyword_Set(sam) Then rank=sort(Tdet, /L64) else rank=reverse(sort(Tdet, /L64))

For i=0L, tsize-1 Do Begin
    If t_truth(rank[i]) GT 0 Then Begin
       res(1,no_tgs)=no_fas
       no_tgs=no_tgs+1
    EndIf Else If t_truth(rank[i]) eq 0 Then no_fas=no_fas+1
    If no_tgs EQ targ_count Then i=tsize-1
EndFor

res(0,*)=res(0,*)/(targ_count) & res(1,*)=res(1,*)/(norm)

Return, res[1,*]

END ;Roc_Calc
```