THESIS

ANOMALY DETECTION IN TERRESTRIAL HYPERSPECTRAL VIDEO USING VARIANTS OF THE RX ALGORITHM

Submitted by
Anthony N. Schwickerath
Department of Mathematics

In partial fulfillment of the requirements
For the degree of Master of Science
Colorado State University
Fort Collins, Colorado
Summer 2012

Master's Committee:

Advisor: Michael Kirby

Christopher Peterson
Charles Anderson

ABSTRACT

ANOMALY DETECTION IN TERRESTRIAL HYPERSPECTRAL VIDEO USING VARIANTS OF THE RX ALGORITHM

There is currently interest in detecting the use of chemical and biological weapons using hyperspectral sensors. Much of the research in this area assumes the spectral signature of the weapon is known in advance. Unfortunately, this may not always be the case. To obviate the reliance on a library of known target signatures, we instead view this as an anomaly detection problem. In this thesis, the RX algorithm, a benchmark anomaly detection algorithm for multi- and hyper-spectral data is reviewed, as are some standard extensions. This class of likelihood ratio test-based algorithms is generally applied to aerial imagery for the identification of man-made artifacts. As such, the model assumes that the scale is relatively consistent and that the targets (roads, cars) also have fixed sizes. We apply these methods to terrestrial video of biological and chemical aerosol plumes, where the background scale and target size both vary, and compare preliminary results. To explore the impact of parameter choice on algorithm performance, we also present an empirical study of the standard RX algorithm applied to synthetic targets of varying sizes over a range of settings.

ACKNOWLEDGEMENTS

First, I want to thank my advisor Michael Kirby. Over the course of the last year and a half, he has encouraged me in my investigation of anomaly detection. This is a class of problems that first interested me while I was a graduate student at University of Massachusetts, Amherst, and he responded enthusiastically when I first suggested picking this work back up. Since then, he has given me guidance when I was mired in the details and free rein to develop my background in a variety of related topics. Even though only a fraction of that research is contained in the pages that follow, it has provided a strong frame of reference for my current and future work in this field.

Thanks also go to Charles (Chuck) Anderson. Chuck was my first undergraduate research mentor in 1991. He encouraged me to continue my graduate education since the moment I left for industry over a decade ago. He has always made time to discuss both my coursework and research. I am grateful for his continued faith in me.

Josh Thompson was a frequent sounding board during the development of this work. Explaining background material and discussing obstacles with him over coffee proved essential to both my intellectual development and continued enthusiasm for this project.

While not included here, Chris Peterson's observations on the RX algorithm provided a key insight that I will be pursuing. His advice when I was trying to balance my course load with research demands also proved critical.

Chris Gittins' comments on an earlier draft of this paper were helpful both in polishing the paper and in developing my understanding of how the algorithms presented here are used in practice.

Last but not least, I am grateful for the love and encouragement of family and friends. Most of all, I am indebted to my wife Kristi. While I know it has not always easy, she has been supportive of my return to school. Without her, this would be a much more difficult journey.

This material is based in part upon work supported by the National Science Foundation under Grant Numbers DMS-0915262 and DMS-1120875. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The Defense Threat Reduction Agency has identified biological and chemical weapons as significant concerns in both domestic and overseas safety [4, 14]. Despite the Biological Weapons Convention of 1972 bringing an end to the production and stockpiling of biological weapons by most countries, there is still concern that terrorist organizations are actively working on producing them. Both biological and chemical weapons can spread beyond the initial release zone, increasing the range of their potential impact. Hence, identifying the release of these agents quickly and accurately is a crucial step in the prevention of a serious health threat.

Standoff detection is a method for detecting a substance safely at a distance. In the context of chemical and biological agents, this is the detection of the release of these airborne substances from a location potentially kilometers outside the release zone. This has the advantage of providing actionable intelligence before entering an area. It also allows a single sensor to monitor a large area.

Hyperspectral cameras are popular and effective sensors for performing standoff detection. The color cameras which we are all familiar with produce images of visible light separated into three spectral bands centered around red, green, and blue. In contrast, hyperspectral cameras produce images of light that, depending upon the sensor, may extend from infrared, through the visible spectrum, and into the ultraviolet and the number of spectral bands ranges from tens to hundreds. In both cases, the image is a record of both ambient light reflected off of objects and light that is emitted by sources in the scene.

State of the art methods for standoff detection of biological and chemical agents rely on classification algorithms and libraries of known spectra for the expected substances of concern. These classification algorithms have their roots in statistics, mathematics, and machine learning and can, in general, identify instances of an arbitrary number of classes [31, 13, 4]. For example, a single classifier might be able to discriminate glacial acetic acid, methyl salicylate, and triethyl phosphate, as well as the background class.

Unfortunately, it can be difficult to collect training data related to biological and chemical weapons released under field conditions. Spectra collected under lab conditions, while helpful, may be different from that observed in the field due to variation in temperature, humidity, atmospheric pressure, and masking effects of the other constituents of the atmosphere. Additionally, new and modified agents can be developed which have different and unknown signatures. As a consequence, training a classifier only on currently known examples may miss both existing and future forms.

The ideal detector would distinguish between known safe substances and those of unknown identity rather than trying to identify only known substances. This avoids the training data availability problem. It also means that the detector should still be sensitive to new or altered agents, since it is constructed without consideration for the set of known strains.

The scenario described above is generally referred to as an anomaly detection problem. An anomaly detection problem is a special two class labeling problem. Instead of having examples of both classes, we only have examples of one class: the nominal (also known as clutter or background) class. The goal is to determine whether a given datum falls into this class or not. If it does not, it is labeled an anomaly. While in general data can be any sort of feature including points, lines, regions, and motion tracks, in this thesis we will focus on labeling individual pixels.

This thesis demonstrates the application of a standard hyperspectral anomal detection technique, the RX algorithm [26], to aerosol release data sets from the Colorado State University Algorithms for Threat Detection Data Repository. We present the derivation and analysis of the RX algorithm in Chapter 2. We then discuss some challenges with this algorithm and the PCA-RX and NAPCA-RX extensions from the literature which attempt to address some of these difficulties. Chapter 2 closes with a description of four hyperspectral data sets to which the RX, PCA-RX, and NAPCA-RX algorithms will be applied. In Chapter 3, we address implementation details. We begin with computation of the test statistic and numerical stability issues. Then we address selection of the spatial template used in parameter estimation. Finally we present the technique for parallelizing RX, PCA-RX, and NAPCA-RX which we used in our implementation. The RX, PCA-RX, and NAPCA-RX algorithms are applied to both synthetic and real data and results are presented in Chapter 4. We then summarize the results and contributions of this thesis in Chapter 5. We close with future directions for this research.

CHAPTER 2

BACKGROUND

2.1 The RX Algorithm

The RX algorithm was originally proposed by Reed and Xiaoli [26]. It takes a statistically motivated approach that extends the single channel version presented by Chen and Reed in [9] to multispectral image data. Since it's introduction, it has been extensively used and extended and is repeatedly referred to in the literature as a benchmark hyperspectral anomaly detector. See [34, 17, 30, 19, 22] for a small sample of algorithms based upon the RX algorithm and [21] for a modern survey of hyperspectral anomaly detection and the RX algorithm's place in the field.

The algorithm has two key features. First, it is derived within a probability theoretic framework, making it defensible when the assumptions hold. Second, it can be shown to have a constant false alarm rate (CFAR) for a given decision threshold, independent of the covariance or signal-to-noise ratio of the image data. The following derivation and analysis follow and elaborate upon that provided by Reed and Xiaoli in [26].

2.1.1 Derivation

The basic idea behind the RX algorithm is that, for each pixel, we wish to compute a test statistic that will indicate if it is more likely that the pixel is an anomaly or that it is nominal. Since it is assumed that the statistical properties of both processes are unknown, we need to produce a reliable estimate for each. This is done by looking at the pixel values in two windows, one tightly centered around the pixel under test (PUT) and one further away. See Figure 2.1.

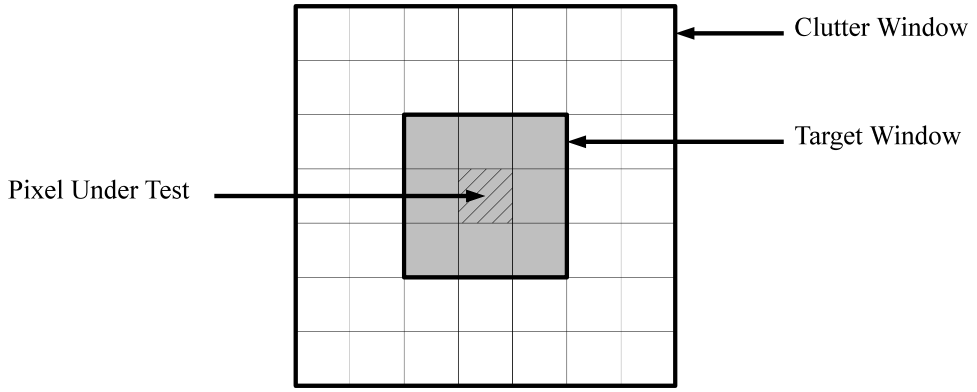


Figure 2.1: Example target and clutter windows surrounding a pixel under test (PUT).

For a particular PUT, consider the data as a $J \times N$ matrix $\mathbf{X}$ made up of raw image data collected from the two windows. Here, N is the number of pixels contained in the two windows (e.g., 49 for the window in Figure 2.1) and J is the number of spectral bands. We can think of $\mathbf{X}$ as a collection of random variables, where

$$\mathbf{x}(n) = [x_1(n), x_2(n), \ldots, x_J(n)]^T$$

is the pixel at site n relative to the PUT, is a column of the matrix

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(N)] \, .$$

Notice that we are looking at the pixels as a collection of data but are not explicitly considering the spatial relationship between those pixels.

Assuming a Gaussian data matrix $\mathbf{X}$ allows for a tractable statistical analysis . However, empirically this is generally not the case with raw image data. Nonetheless, it has been shown that, if we subtract the nonstationary local mean

$$\bar{\mathbf{X}} = \frac{1}{L^2} (\mathbf{X} * \mathbf{W})$$

where $\mathbf{W}$ is an $L \times L$ matrix of ones and $*$ is the convolution operator, then the data approaches a Gaussian distribution [16, 20]. They select L by minimizing the third moment

$$m_{3j} = \frac{1}{M} \sum_{i=1}^{M} (x_j(i) - \bar{x}_j(i))^3$$

where M is the number of pixels in a single band of the image as suggested in [16]. The spectral band under consideration is denoted by j. Matteoli, et al. [21] note that L should be smaller than the clutter window, since Reed and Xiaoli assume the mean to vary faster than the covariance. Observe that this formulation varies from the traditional definition in that it takes into account the nonstationary local mean and assumes independence of image bands.

Assume that we are given a spatial template that covers both of the windows,

$$\mathbf{s} = [s(1), s(2), \ldots, s(N)]$$

an $N \times 1$ column vector. In practice, this is a binary vector, with 1 in the target window and 0 in the clutter window. For example, the s that corresponds to Figure 2.1 is

$$\begin{aligned} \mathbf{s} = [&0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,\\ &0,0,0,0,0,0,0,0,0,0,0,0,0] \,. \end{aligned}$$

Think of this as the vectorized form of a spatial template that we will be sliding over the image, trying to find a match. It contains no spectral information. In practice, the target window is a small square that indicates the scale of the anomalies we wish to detect.

Similarly, assume that there is an unknown spectral signature

$$\mathbf{b} = [b_1, b_2, \ldots, b_J]^T \,.$$

Since this is a binary classifier (nominal and anomaly), we have two hypotheses for each pixel,

$$H_0 : \mathbf{x}(n) = \mathbf{x}^0(n), \qquad \forall n = 1, 2, \ldots, N,$$

$$H_1 : \mathbf{x}(n) = \mathbf{x}^0(n) + \mathbf{b}s(n), \qquad \forall n = 1, 2, \ldots, N.$$

Here, $\mathbf{x}^0(n)$ is produced by the nominal process, also referred to as background, clutter, or noise. Notice the way the n index is used; the data $\mathbf{x}$ and $\mathbf{x}^0$ are assumed to be aligned with the signal template s. We move the window over the data and determine which hypothesis is most likely for that position. We then label the PUT for that location.

Following [20, 9], it is assumed in [26] that, after subtracting the nonstationary mean, the clutter process is approximately Gaussian and independent (pixel-to-pixel). Let the (unknown) covariance matrix for this process at pixel n be given by

$$\mathbf{M} \doteq \mathbf{E}\left[\left(\mathbf{x}(n) - \mathbf{E}\mathbf{x}(n)\right) \left(\mathbf{x}(n) - \mathbf{E}\mathbf{x}(n)\right)^T \right].$$

Since the data is assumed to have a rapidly varying mean, we explicitly denote the mean vector for the distribution site n as $\mathbf{E}\mathbf{x}(n)$. The covariance matrix is assumed to be slowly varying, however, hence we denote it by a single $\mathbf{M}$ for every pixel in the window.

Recall that the probability density function (pdf) for a J-dimensional Gaussian random vector $\mathbf{x}(n)$ with covariance matrix $\mathbf{M}$ and mean $\mathbf{E}\mathbf{x}(n)$ is

$$f(\mathbf{x}(n)) = \frac{1}{(2\pi)^{J/2} |\mathbf{M}|^{1/2}} \exp\left(-\frac{1}{2} \left(\mathbf{x}(n) - E\mathbf{x}(n)\right)^T \mathbf{M}^{-1} \left(\mathbf{x}(n) - \mathbf{E}\mathbf{x}(n)\right) \right).$$

Since we don't know $\mathbf{b}$ or $\mathbf{M}$, our parameter space is given by

$$\Omega \doteq \left\{ [\mathbf{b}, \mathbf{M}] \mid \mathbf{M} > 0 \right\}.$$

The likelihood function for a specific $\mathbf{b}$ and $\mathbf{M}$ is given by

$$\begin{aligned} L(\mathbf{b},\mathbf{M}) &= \prod_{n=1}^{N} \frac{1}{(2\pi)^{J/2}\,|\mathbf{M}|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}(n)-\mathbf{E}\mathbf{x}(n)\right)^T \mathbf{M}^{-1}\left(\mathbf{x}(n)-\mathbf{E}\mathbf{x}(n)\right)\right) \\ &= \frac{1}{(2\pi)^{NJ/2}\,|\mathbf{M}|^{N/2}} \exp\left(-\frac{1}{2}\sum_{n=1}^{N}\left(\mathbf{x}(n)-\mathbf{E}\mathbf{x}(n)\right)^T \mathbf{M}^{-1}\left(\mathbf{x}(n)-\mathbf{E}\mathbf{x}(n)\right)\right) \\ &= \frac{1}{(2\pi)^{NJ/2}\,|\mathbf{M}|^{N/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)^T \mathbf{M}^{-1}\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)\right)\right) \\ &= \frac{1}{(2\pi)^{NJ/2}\,|\mathbf{M}|^{N/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\mathbf{M}^{-1}\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)^T\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)\right)\right). \end{aligned}$$

Let $\omega \subset \Omega$ corresponding to H_0 and $\Omega - \omega \subset \Omega$ corresponding to H_1. More specifically,

$$\begin{aligned} \omega &= \{[0,\mathbf{M}] \mid \mathbf{M} > 0\}, \\ \Omega - \omega &= \{[\mathbf{b},\mathbf{M}] \mid \mathbf{M} > 0, \mathbf{b} \neq 0\}. \end{aligned}$$

The generalized likelihood ratio (GLR) test considers the ratio of the maximum likelihood of H_1 versus the maximum likelihood of H_0.

$$\Lambda(\mathbf{x}) = \frac{\max_{[\mathbf{b},\mathbf{M}]\in\Omega-\omega} L(\mathbf{b},\mathbf{M})}{\max_{[\mathbf{b},\mathbf{M}]\in\omega} L(\mathbf{b},\mathbf{M})} \underset{H_0}{\overset{H_1}{\gtrless}} k$$

which, since $\mathbf{b} = \mathbf{0}$ in ω, is

$$\Lambda(\mathbf{x}) = \frac{\max_{[\mathbf{b},\mathbf{M}]\in\Omega-\omega} L(\mathbf{b},\mathbf{M})}{\max_{[\mathbf{0},\mathbf{M}]\in\omega} L(\mathbf{0},\mathbf{M})} \underset{H_0}{\overset{H_1}{\gtrless}} k$$

In other words, when $\Lambda(\mathbf{x}) \geq k$, we accept hypothesis H_1 ($\mathbf{x}$ is an anomaly), while we accept hypothesis H_0 ($\mathbf{x}$ is clutter) when $\Lambda(\mathbf{x}) < k$.

Note that

$$\max_{[\mathbf{b},\mathbf{M}]\in\omega} L(\mathbf{b},\mathbf{M}) = \frac{1}{(2\pi)^{NJ/2}\,|\hat{\mathbf{M}}_0|^{N/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\hat{\mathbf{M}}_0^{-1}\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)^T\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)\right)\right)$$

$$
\begin{aligned}
&= \frac{1}{(2\pi)^{NJ/2}\,|\hat{\mathbf{M}}_0|^{N/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\hat{\mathbf{M}}_0^{-1}\hat{\mathbf{M}}_0\right)\right) \\
&= \frac{1}{(2\pi)^{NJ/2}\,|\hat{\mathbf{M}}_0|^{N/2}} \exp\left(-\frac{NJ}{2}\right) \\
\max_{[\mathbf{b},\mathbf{M}]\in\Omega-\omega} L(\mathbf{b},\mathbf{M}) &= \frac{1}{(2\pi)^{NJ/2}\,|\hat{\mathbf{M}}_b|^{N/2}} \exp\left(-\frac{1}{2}\operatorname{Tr}\left(\hat{\mathbf{M}}_b^{-1}\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)^T\left(\mathbf{X}-\mathbf{E}\mathbf{X}\right)\right)\right) \\
&= \frac{1}{(2\pi)^{NJ/2}\,|\hat{\mathbf{M}}_b|^{N/2}} \exp\left(-\frac{NJ}{2}\right)
\end{aligned}
$$

where

$$
\begin{aligned}
\hat{\mathbf{M}}_0 &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}(n)\mathbf{x}^T(n) \\
&= \frac{1}{N}\mathbf{X}\mathbf{X}^T \\
\hat{\mathbf{M}}_b &= \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_b(n)\mathbf{x}_b^T(n) \\
&= \frac{1}{N}\left(\mathbf{X}-\hat{\mathbf{b}}\mathbf{s}^T\right)\left(\mathbf{X}-\hat{\mathbf{b}}\mathbf{s}^T\right)^T \\
\hat{\mathbf{b}} &= \frac{\mathbf{X}\mathbf{s}}{\mathbf{s}^T\mathbf{s}}
\end{aligned}
$$

Hence, the GLR test becomes

$$
\begin{aligned}
&& \Lambda(\mathbf{X}) &= \frac{|\hat{\mathbf{M}}_0|^{N/2}}{|\hat{\mathbf{M}}_b|^{N/2}} \overset{H_1}{\underset{H_0}{\gtrless}} k \\
\Longrightarrow && \lambda(\mathbf{X}) &= \frac{|\hat{\mathbf{M}}_0|}{|\hat{\mathbf{M}}_b|} \overset{H_1}{\underset{H_0}{\gtrless}} c \\
&& &= \frac{\left|\frac{1}{N}\mathbf{X}\mathbf{X}^T\right|}{\left|\frac{1}{N}\left(\mathbf{X}-\hat{\mathbf{b}}\mathbf{s}^T\right)\left(\mathbf{X}-\hat{\mathbf{b}}\mathbf{s}^T\right)^T\right|} \\
&& &= \frac{\left|\mathbf{X}\mathbf{X}^T\right|}{\left|\left(\mathbf{X}-\frac{\mathbf{X}\mathbf{s}}{\mathbf{s}^T\mathbf{s}}\mathbf{s}^T\right)\left(\mathbf{X}-\frac{\mathbf{X}\mathbf{s}}{\mathbf{s}^T\mathbf{s}}\mathbf{s}^T\right)^T\right|}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\left|\mathbf{X}\mathbf{X}^T\right|}{\left|\mathbf{X}\mathbf{X}^T - \mathbf{X}\frac{\left(\mathbf{X}\mathbf{s}\mathbf{s}^T\right)^T}{\mathbf{s}^T\mathbf{s}} - \frac{\left(\mathbf{X}\mathbf{s}\mathbf{s}^T\right)}{\mathbf{s}^T\mathbf{s}}\mathbf{X}^T + \frac{\mathbf{X}\mathbf{s}\mathbf{s}^T}{\mathbf{s}^T\mathbf{s}}\frac{\left(\mathbf{X}\mathbf{s}\mathbf{s}^T\right)^T}{\mathbf{s}^T\mathbf{s}}\right|} \\
&= \frac{\left|\mathbf{X}\mathbf{X}^T\right|}{\left|\mathbf{X}\mathbf{X}^T - \frac{\mathbf{X}\mathbf{s}\mathbf{s}^T\mathbf{X}^T}{\mathbf{s}^T\mathbf{s}} - \frac{\mathbf{X}\mathbf{s}\mathbf{s}^T\mathbf{X}^T}{\mathbf{s}^T\mathbf{s}} + \frac{\mathbf{X}\mathbf{s}\mathbf{s}^T\mathbf{s}\mathbf{s}^T\mathbf{X}^T}{\mathbf{s}^T\mathbf{s}\mathbf{s}^T\mathbf{s}}\right|} \\
&= \frac{\left|\mathbf{X}\mathbf{X}^T\right|}{\left|\mathbf{X}\mathbf{X}^T - \frac{\left(\mathbf{X}\mathbf{s}\right)\left(\mathbf{X}\mathbf{s}\right)^T}{\mathbf{s}^T\mathbf{s}}\right|} \\
&= \frac{\left|\mathbf{X}\mathbf{X}^T\right|}{\left|\mathbf{X}\mathbf{X}^T\right|\left|\mathbf{I} - \left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\frac{\left(\mathbf{X}\mathbf{s}\right)\left(\mathbf{X}\mathbf{s}\right)^T}{\mathbf{s}^T\mathbf{s}}\left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\right|}
\end{aligned}
$$

Applying Sylvester's Determinant Theorem, we see that

$$
\begin{aligned}
&\Bigg|\mathbf{I} - \frac{1}{\mathbf{s}^T\mathbf{s}}\left(\left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\left(\mathbf{X}\mathbf{s}\right)\right)\left(\left(\mathbf{X}\mathbf{s}\right)^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\right)| \\
&\qquad = 1 - \frac{1}{\mathbf{s}^T\mathbf{s}}\left(\left(\mathbf{X}\mathbf{s}\right)^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\right)\left(\left(\mathbf{X}\mathbf{X}^T\right)^{-1/2}\left(\mathbf{X}\mathbf{s}\right)\right) \\
&\qquad = 1 - \frac{\left(\mathbf{X}\mathbf{s}\right)^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1}\left(\mathbf{X}\mathbf{s}\right)}{\mathbf{s}^T\mathbf{s}}.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
& & \lambda(\mathbf{X}) &= \frac{1}{1 - \frac{\left(\mathbf{X}\mathbf{s}\right)^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1}\left(\mathbf{X}\mathbf{s}\right)}{\mathbf{s}^T\mathbf{s}}} \\
&\Longrightarrow & r(\mathbf{X}) &= \frac{\left(\mathbf{X}\mathbf{s}\right)^T\left(\mathbf{X}\mathbf{X}^T\right)^{-1}\left(\mathbf{X}\mathbf{s}\right)}{\mathbf{s}^T\mathbf{s}} \underset{H_0}{\overset{H_1}{\gtrless}} r_0.
\end{aligned}
$$

In other words, the test statistic, r, essentially measures how far the mean of the local clutter distribution the average of the data in the target window is.[1] This is the squared Mahalanobis distance, since the data has already been mean-subtracted. Points of equal Mahalanobis distance describe an ellipsoid centered at the distribution mean and oriented according to the correlation matrix, M.

2.1.2 Analysis

How do we select r_0? We would like to be able to select it either based upon the false accept rate (FAR) or based upon the detection rate, which are denoted P_{FA} or P_D, respectively. Which of the two we want to start from depends upon the application. If we want to be fairly confident that what is marked as a detection actually is correct, we want a small P_{FA}. However, if the cost of a missed detection is high and a human operator is filtering results, P_D should be near 1.

The false alarm rate is the probability that we label a pixel $\mathbf{x}$ as H_1 ($r(\mathbf{x}) \geq r_0$) when it is actually H_0. The detection rate is the probability that we label pixel $\mathbf{x}$ H_1 when it actually is H_1. These probabilities can be stated mathematically as

$$\begin{aligned} P_{FA} &= P(r \geq r_0 \mid H_0) \\ &= \int_{r_0}^{1} f(r \mid H_0)\, dr \\ P_D &= P(r \geq r_0 \mid H_1) \\ &= \int_{r_0}^{1} f(r \mid H_1)\, dr. \end{aligned}$$

[1] "Average" is used loosely here. Assuming $\mathbf{s}$ is a binary vector, the vector $\frac{\mathbf{Xs}}{\mathbf{s}^T\mathbf{s}}$ is the average of the pixel values in the target window, since $\mathbf{Xs}$ gives the sum of the pixel values in the target window and $\mathbf{s}^T\mathbf{s}$ is the number of pixels in the target window. We are actually measuring the distance from $\frac{\mathbf{Xs}}{(\mathbf{s}^T\mathbf{s})^{1/2}}$.

To use these, we need to find $f(r \mid H_0)$ and $f(r \mid H_1)$, the probability density functions conditioned on the two hypotheses. To get there, we will transform the problem into one where the multivariate random variables are independent and normalized and then substitute back to get to the original distribution.

First, let us make some observations about the original data. Since we have already (locally) mean subtracted $\mathbf{x}(n)$,

$$\begin{aligned} E\left[\mathbf{x}(n) \mid H_0\right] &= E\left[\mathbf{x}^0(n)\right] \\ &= \mathbf{0} \end{aligned}$$

and

$$\begin{aligned} E\left[\mathbf{x}(n) \mid H_1\right] &= E\left[\mathbf{x}^0(n) + \mathbf{b}s(n)\right] \\ &= \underbrace{E\left[\mathbf{x}^0(n)\right]}_{=\mathbf{0}} + E\left[\mathbf{b}s(n)\right] \\ &= \mathbf{b}s(n). \end{aligned}$$

Given a hypothesis, the above expectations are true for every pixel site n in the clutter and target windows, hence the conditional expectations for the data matrix are $\mathbf{E}\left[\mathbf{X} \mid H_0\right] = \mathbf{0}$ and $\mathbf{E}\left[\mathbf{X} \mid H_1\right] = \mathbf{b}\mathbf{s}^T$.

Now consider the whitened data at pixel site n.

$$\begin{aligned} & \mathbf{z}(n) = \mathbf{M}^{-1/2}\mathbf{x}(n) \qquad \forall n = 1, 2, \ldots, N \\ \Longrightarrow \qquad & \mathbf{Z} = \mathbf{M}^{-1/2}\mathbf{X} \\ & \quad = \left[\mathbf{z}(1), \ldots, \mathbf{z}(N)\right] \end{aligned}$$

A side effect of $\mathbf{X}$ having the nonstationary local mean subtracted is that the data points (random variables) are nearly independent. In the following, we assume that they *are* actually independently distributed. In other words,

$$\mathrm{Cov}\left[z_i(m), z_j(n)\right] = \delta_{i,j}\delta_{m,n} \qquad \forall i, j = 1, \ldots, J \text{ and } \forall n, m = 1, \ldots, N$$

Since this is a linear transformation of the random variables $\mathbf{X}$,

$$\begin{aligned} E\left[\mathbf{Z} \mid H_0\right] &= \mathbf{0} \\ E\left[\mathbf{Z} \mid H_1\right] &= \mathbf{M}^{-1/2}\mathbf{b}\mathbf{s}^T \\ \mathrm{Cov}\left[\mathbf{z}(n) \mid H_0\right] &= \mathbf{I}_J \\ \mathrm{Cov}\left[\mathbf{z}(n) \mid H_1\right] &= \mathbf{I}_J. \end{aligned}$$

In other words, the random variables are not only independent of each other, they are also independent of each other independent of which hypothesis H_0 or H_1 is true.

Since we ultimately want to look at the distribution of $r \geq r_0$ given one of these hypotheses, let us rewrite r in terms of $\mathbf{Z}$.

$$\begin{aligned} r &= \frac{(\mathbf{X}\mathbf{s})^T \left(\mathbf{X}\mathbf{X}^T\right)^{-1} (\mathbf{X}\mathbf{s})}{\mathbf{s}^T\mathbf{s}} \underset{H_0}{\overset{H_1}{\gtrless}} r_0 \\ &= \frac{\mathbf{s}^T\mathbf{X}^T\mathbf{M}^{-1/2}\mathbf{M}^{1/2}\left(\mathbf{X}\mathbf{X}^T\right)^{-1}\mathbf{M}^{1/2}\mathbf{M}^{-1/2}\mathbf{X}\mathbf{S}}{\mathbf{s}^T\mathbf{s}} \\ &= \frac{(\mathbf{Z}\mathbf{s})^T \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1} (\mathbf{Z}\mathbf{s})}{\mathbf{s}^T\mathbf{s}} \underset{H_0}{\overset{H_1}{\gtrless}} r_0 \end{aligned}$$

If we normalize $\mathbf{s}$ with

$$\mathbf{s_1} = \frac{\mathbf{s}}{\left(\mathbf{s}^T\mathbf{s}\right)^{1/2}} \ ,$$

then

$$r = (\mathbf{Z}\mathbf{s_1})^T \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1} (\mathbf{Z}\mathbf{s_1}) \underset{H_0}{\overset{H_1}{\gtrless}} r_0 \ .$$

Recall that our ultimate goal is to produce the distribution for the single random variable r. Let us construct an orthonormal linear transformation that will combine all of the pixels associated with the indicator function $\mathbf{s_1}$ (and hence, $\mathbf{s}$) into a single J-dimensional random vector. Let $\mathbf{Q}$ be an $(N-1) \times N$ matrix with orthonormal row vectors, each also orthogonal with $\mathbf{s_1}^T$. In other words, $\mathbf{Q}\mathbf{s_1} = \mathbf{0}$ and $\mathbf{Q}\mathbf{Q}^T = \mathbf{I}_{(N-1)}$. Hence,

$$\begin{aligned} \mathbf{U} &\doteq \begin{bmatrix} \mathbf{s_1}^T \\ \mathbf{Q} \end{bmatrix}^T_{N \times N} \\ \Longrightarrow \qquad \mathbf{U}^T\mathbf{s_1} &= [1, 0, \ldots, 0]^T \ . \end{aligned}$$

Our data matrix, now whitened and then transformed (rotated) by this produces

$$\begin{aligned} \mathbf{V} &\doteq \mathbf{Z}\mathbf{U} \\ &= [\mathbf{v}(1), \ldots, \mathbf{v}(N)] \, . \end{aligned}$$

Observe that $\mathbf{v}(1)$ is the data in the target window mapped into a single J-dimensional random vector. Now we can rewrite our likelihood test in these more convenient coordinates as

$$\begin{aligned} r &= (\mathbf{Z}\mathbf{s_1})^T \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1} (\mathbf{Z}\mathbf{s_1}) \underset{H_0}{\overset{H_1}{\gtrless}} r_0 \\ &= \left(\mathbf{Z}\mathbf{U}\mathbf{U}^T\mathbf{s_1}\right)^T \left(\mathbf{Z}\mathbf{U}\mathbf{U}^T\mathbf{Z}^T\right)^{-1} \left(\mathbf{Z}\mathbf{U}\mathbf{U}^T\mathbf{s_1}\right) \\ &= \mathbf{v}^T(1) \left(\mathbf{V}\mathbf{V}^T\right)^{-1} \mathbf{v}(1) \ . \end{aligned}$$

Let us look at the statistics of the random variable $\mathbf{V}$. Specifically, consider the expectation given hypothesis H_1.

$$E\left[\mathbf{V} \mid H_1\right] = E\left[\mathbf{Z}\mathbf{U}^T \mid H_1\right]$$

$$
\begin{aligned}
&= E\left[\mathbf{M}^{-1/2}\mathbf{X}\mathbf{U}^T \mid H_1\right] \\
&= \mathbf{M}^{-1/2} E\left[\mathbf{X} \mid H_1\right] \mathbf{U}^T \\
&= \mathbf{M}^{-1/2}\mathbf{b}\mathbf{s}^T\mathbf{U}^T \\
&= \mathbf{M}^{-1/2}\mathbf{b}\mathbf{s_1}^T\mathbf{U}^T \left(\mathbf{s}^T\mathbf{s}\right)^{1/2} \\
&= \mathbf{M}^{-1/2}\mathbf{b}\left[1, 0, \ldots, 0\right]\left(\mathbf{s}^T\mathbf{s}\right)^{1/2} \\
&= \left[\mathbf{M}^{-1/2}\mathbf{b}\left(\mathbf{s}^T\mathbf{s}\right)^{1/2}, 0, \ldots, 0\right]
\end{aligned}
$$

From this, Reed and Xiaoli define the generalized signal-to-noise-ratio (GSNR) as

$$
\begin{aligned}
\text{GSNR} &\doteq E\left[\mathbf{v}^T(1) \mid H_1\right] E\left[\mathbf{v}(1) \mid H_1\right] \\
&= \left(\mathbf{M}^{-1/2}\mathbf{b}\left(\mathbf{s}^T\mathbf{s}\right)^{1/2}\right)^T \left(\mathbf{M}^{-1/2}\mathbf{b}\left(\mathbf{s}^T\mathbf{s}\right)^{1/2}\right) \\
&= \left(\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b}\right)\left\|\mathbf{s}\right\|^2 .
\end{aligned}
$$

Looking at our definition of r in terms of $\mathbf{V}$, notice that it is entirely composed of parts of $\mathbf{V}$ (either the whole thing or $\mathbf{v}(1)$). Break $\mathbf{V}$ into $\mathbf{v}(1)$ and $\mathbf{D} = [\mathbf{v}(2), \ldots, \mathbf{v}(N)]$,

$$
\begin{aligned}
\mathbf{V}\mathbf{V}^T &= \mathbf{v}(1)\mathbf{v}^T(1) + \sum_{n=2}^{N} \mathbf{v}(n)\mathbf{v}^T(n) \\
&= \mathbf{v}(1)\mathbf{v}^T(1) + \mathbf{\Phi}.
\end{aligned}
$$

where $\mathbf{\Phi} = \mathbf{D}\mathbf{D}^T$ is a non-singular $J \times J$ matrix. Since we are interested in r, we are actually interested in $\left(\mathbf{V}\mathbf{V}^T\right)^{-1}$,

$$
\begin{aligned}
\left(\mathbf{V}\mathbf{V}^T\right)^{-1} &= \left(\mathbf{v}(1)\mathbf{v}^T(1) + \mathbf{\Phi}\right)^{-1} \\
&= \left(\mathbf{I} - \frac{\mathbf{\Phi}^{-1}\mathbf{v}(1)\mathbf{v}^T(1)}{1 + \mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1)}\right)\mathbf{\Phi}^{-1}.
\end{aligned}
$$

Now substitute this into the equation for r,

$$\begin{aligned} r &= \mathbf{v}^T(1)\left(\mathbf{V}\mathbf{V}^T\right)^{-1}\mathbf{v}(1) \\ &= \mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1) \\ &= \mathbf{v}^T(1)\left(\mathbf{I} - \frac{\mathbf{\Phi}^{-1}\mathbf{v}(1)\mathbf{v}^T(1)}{1+\mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1)}\right)\mathbf{\Phi}^{-1}\mathbf{v}(1) \\ &= \frac{\mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1)}{1+\mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1)} \\ &= \frac{r_1}{1+r_1} \end{aligned}$$

where $r_1 = \mathbf{v}^T(1)\mathbf{\Phi}^{-1}\mathbf{v}(1)$. We can rewrite this as

$$r_1 = \|\mathbf{v}(1)\|^2\left(\frac{\mathbf{v}^T(1)}{\|\mathbf{v}(1)\|}\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\frac{\mathbf{v}(1)}{\|\mathbf{v}(1)\|}\right).$$

If we define $\xi = \dfrac{\mathbf{v}(1)}{\|\mathbf{v}(1)\|}$, the unit vector in the direction of $\mathbf{v}(1)$, then

$$r_1 = \|\mathbf{v}(1)\|^2\left(\xi^T\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\xi\right).$$

Defining $e = \xi^T\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\xi$, we get

$$r_1 = \|\mathbf{v}(1)\|^2 e.$$

Since $\|\xi\| = 1$ and $\mathbf{D}$ is unitary, there exists a $\mathbf{U}_1$ unitary such that

$$\mathbf{U}_1\xi = [1, 0, \ldots, 0]^T.$$

Now define $\mathbf{H} = \mathbf{U}_1\mathbf{D} = \mathbf{U}_1[\mathbf{v}(2), \ldots, \mathbf{v}(N)]$. Then

$$\begin{aligned} e &= \xi^T\left(\mathbf{D}\mathbf{D}^T\right)^{-1}\xi \\ &= [1, 0, \ldots, 0]\left(\mathbf{H}\mathbf{H}^T\right)^{-1}[1, 0, \ldots, 0]^T. \end{aligned}$$

So now partition $\mathbf{H}$ as we did with $\mathbf{U}$,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_A^T \\ \mathbf{H}_B \end{bmatrix}.$$

To find e, we must find $\left(\mathbf{H}\mathbf{H}^T\right)^{-1}$. We can do this by multiplying out it out using the block form of $\mathbf{H}$,

$$\begin{aligned} \left(\mathbf{H}\mathbf{H}^T\right)^{-1} &= \begin{bmatrix} \mathbf{h}_A^T\mathbf{h}_A & \mathbf{h}_A^T\mathbf{H}_B^T \\ \mathbf{H}_B\mathbf{h}_A & \mathbf{H}_B\mathbf{H}_B^T \end{bmatrix} \\ &\doteq \begin{bmatrix} \mathbf{R_{AA}} & \mathbf{R_{AB}} \\ \mathbf{R_{BA}} & \mathbf{R_{BB}} \end{bmatrix}. \end{aligned}$$

Using the helpful inversion formula found in [15, page 472] yields

$$\begin{aligned} \mathbf{R_{AA}} &= \left(\mathbf{h}_A^T\mathbf{h}_A - \mathbf{h}_A^T\mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)\mathbf{H}_B\mathbf{h}_A\right)^{-1} \\ &= \left(\mathbf{h}_A^T\left(\mathbf{I} - \mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B\right)\mathbf{h}_A\right)^{-1} \\ &= \frac{1}{\mathbf{h}_A^T\left(\mathbf{I} - \mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B\right)\mathbf{h}_A} \\ &= \frac{1}{\mathbf{h}_A^T\mathbf{P}_1\mathbf{h}_A} \end{aligned}$$

where $\mathbf{P}_1 = \mathbf{I}_N - \mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B$. Since $\mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B$ is clearly a projection operator ($\mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B\mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B = \mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B$), so is $\mathbf{P}_1$. Therefore, $\mathbf{P}_1^2 = \mathbf{P}_1$. By the argument alluded to in [9], $\mathrm{Tr}\left(\mathbf{P}_1\right) = N - J$ and $\mathbf{P}_1$ has $N - J$ unity eigenvalues and $J - 1$ zero eigenvalues. Specifically,

$$\begin{aligned} \mathrm{Tr}\left(\mathbf{P}_1\right) &= \mathrm{Tr}\left(\mathbf{I}_N - \mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B\right) \\ &= \mathrm{Tr}\left(\mathbf{I}_N\right) - \mathrm{Tr}\left(\mathbf{H}_B^T\left(\mathbf{H}_B\mathbf{H}_B^T\right)^{-1}\mathbf{H}_B\right) \end{aligned}$$

$$= N - J.$$

All eigenvalues being either unity or zero follows from $\mathbf{H}_B$ being orthonormal.

Hence, there exists an eigenvalue factorization

$$\begin{aligned}\mathbf{U}_2^T\mathbf{P}_1\mathbf{U}_2 &= \mathbf{\Lambda}_1 \\ &= \begin{bmatrix}\mathbf{I}_{N-J} & \mathbf{0} \\ \mathbf{0} & \mathbf{0_{J-1}}\end{bmatrix}.\end{aligned}$$

Given $\mathbf{v}(1)$ and $\mathbf{P}_1$, e^{-1} is a random variable,

$$\begin{aligned}\frac{1}{e} &= \mathbf{h}_A^T\mathbf{P}_1\mathbf{h}_A \\ &= \eta^T\eta \\ &= \sum_{i=1}^{N-J}\eta_i^2\end{aligned}$$

where $\eta = \mathbf{\Lambda}_1^{1/2}\mathbf{U}_2^T\mathbf{\Lambda}_1^{1/2}$. Hence, given $\mathbf{v}(1)$ and $\mathbf{P}_1$, $\eta_1, \ldots, \eta_{N-J}$ are independent and identically distributed standard normal random variables. In other words,

$$P\left(\eta_1, \ldots, \eta_{N-J} \mid \mathbf{v}(1), \mathbf{P}_1\right) = N\left(\mathbf{0}, \mathbf{I}_{N-J}\right).$$

Notice that η is independent of $\mathbf{v}(1)$ and $\mathbf{P}_1$. Hence,

$$P\left(\eta_1, \ldots, \eta_{N-J}\right) = N\left(\mathbf{0}, \mathbf{I}_{N-J}\right)$$

From this, we get that e^{-1} is χ^2 distributed, since it is the sum of the squares of $N - J$ standard, normal, independent random variables.

Now we can rewrite r_1 in terms of $\mathbf{v}(1)$ and η as

$$\begin{aligned} r_1 &= \frac{\mathbf{v}^T(1)\mathbf{v}(1)}{\eta^T\eta}, \\ &= \frac{\sum_{j=1}^{J} v_j^2(1)}{\sum_{i=1}^{N-J} \eta_i^2}. \end{aligned}$$

Notice that $\mathbf{v}(1) \perp\!\!\!\perp \eta$, $\mathbf{v}(1) \in \Re_J\left[\mathbf{I}_J; \mathbf{E}\left[\mathbf{v}(1) \mid H_i\right]\right]$, and $\eta \in \Re_{N-J}\left[\mathbf{I}_{N-J}; \mathbf{0}_{N-J}\right]$. In other words, since the entries are uncorrelated normal random variables, each with the same variance, $\mathbf{v}(1)$ is a J-dimensional Rayleigh random vector. The covariance of $\mathbf{v}(1)$ is $\mathbf{I}_J$ and the mean is $\mathbf{E}\left[\mathbf{v}(1) \mid H_i\right]$ assuming the original $\mathbf{x}$ belongs to class $i \in \{0, 1\}$. Similarly, η is an $(N-J)$-dimensional Rayleigh random vector with covariance $\mathbf{I}_{N-J}$ and mean $\mathbf{0}_{N-J}$. Since we are considering the quotient

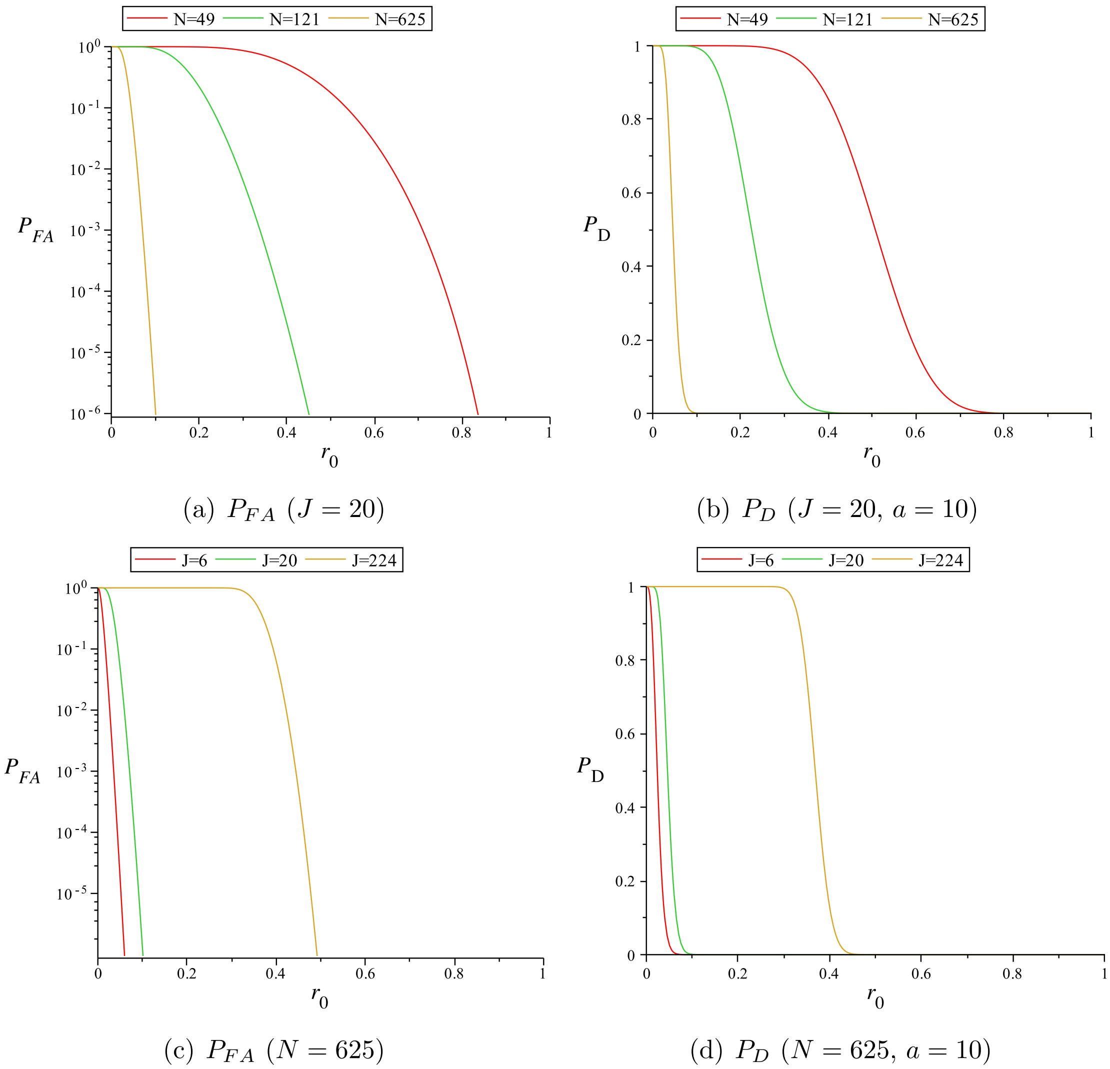
$$r_1 = \left(\frac{|\mathbf{v}(1)|}{|\eta|}\right)^2 ,$$

we can apply [24, page 52, corollary 2] and the observation that $|\mathbf{E}\left[\mathbf{v}(1) \mid H_i\right]|^2 = a$.

$$f(r_1 \mid H_1) = \frac{r_1^{(J-2)/2} e^{-a/2}}{B\left(\frac{N-J}{2}, \frac{J}{2}\right)(1+r_1)^{N/2}} {}_1F_1\left(\frac{N}{2}; \frac{J}{2}; \frac{ar_1}{2(1+r_1)}\right)$$

where $B(\alpha, \beta)$ is the Beta function, ${}_1F_1(\alpha; \beta; x)$ is the confluent hypergeometric function, and a is the generalized signal-to-noise ratio (GSNR). Recalling that $r = \dfrac{r_1}{1+r_1}$ yields

$$\begin{aligned} f(r \mid H_1) &= \frac{\Gamma\left(\frac{N}{2}\right)}{\Gamma\left(\frac{N-J}{2}\right)\Gamma\left(\frac{J}{2}\right)} (1-r)^{(N-J-2)/2} r^{(J-2)/2} e^{-a/2} {}_1F_1\left(\frac{N}{2}; \frac{J}{2}; \frac{ar}{2}\right), \\ f(r \mid H_0) &= \frac{\Gamma\left(\frac{N}{2}\right)}{\Gamma\left(\frac{N-J}{2}\right)\Gamma\left(\frac{J}{2}\right)} (1-r)^{(N-J-2)/2} r^{(J-2)/2}. \end{aligned}$$



(a) P_{FA} $(J = 20)$

(b) P_D $(J = 20,\ a = 10)$

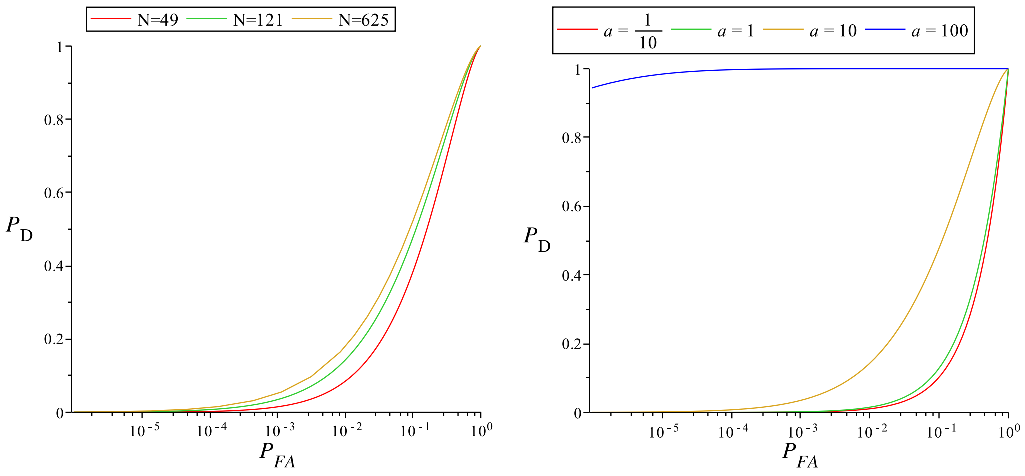
(c) P_{FA} $(N = 625)$

(d) P_D $(N = 625,\ a = 10)$

Figure 2.2: P_{FA} and P_D versus r_0.

The form of $f(r \mid H_0)$ follows directly from $f(r \mid H_1)$, since H_0 implies $a = 0$. This says that r subject to H_1 is noncentral beta-distributed and r subject to H_0 is standard beta-distributed.

It also says that the false alarm rate and detection rate for a given r_0 are dependent upon N, the size of the spatial template, and J, the number of spectral bands, but not upon $\mathbf{M}$, the covariance matrix. In other words, the RX algorithm is what is known as a constant false alarm rate (CFAR) detector. This is convenient, since the covariance matrix can change throughout the image and also from frame to frame. We can select r_0 once, though, and know that the theoretical false alarm and detection rates will not change.



(a) ($J = 20$, $a = 10$)

(b) ($N = 121$, $J = 20$)

Figure 2.3: Theoretical receiver operating characteristic (ROC) curves for the RX algorithm.

Figure 2.2 shows the choice of threshold (r_0) versus the false alarm rate (P_{FA}) and detection rate (P_D). Since these change with varying N (spatial template window pixel count), J (spectral bands), and a (GSNR), we show these for some different values. Notice that using more pixels in the spatial template window improves the estimate of the statistics and reduces the threshold necessary for a given false alarm or detection rate.

Figure 2.3 shows the theoretical receiver operating characteristic (ROC) curves for some of the same values of N, J, and a. While adding more samples to the spatial window does improve performance, it is not a large improvement. A greater GSNR, on the other hand, significantly improves the detector performance. Note that this signal-to-noise ratio is the ratio of the signal signature to the clutter process. This suggests that we may improve our performance by applying a preprocessing step that attenuates the clutter and accentuates the non-clutter portion of the signal.

2.1.3 Extensions

Central to the RX algorithm is computation of the r test statistic for each pixel. This involves computing the inverse of the $J \times J$ matrix $\mathbf{XX}^T$. Whether directly computing the inverse or using the LU decomposition method described in Section 3.1, computing this part of the test statistic can generally be counted as requiring $O(J^3)$ time.[2] As a consequence, when moving from $J = 20$ (see FPISDS in Section 2.3.2) to $J = 200$ (see AVIRIS in Section 2.3.1), the cost of computing the r statistic at each pixel can increase by a factor of 1000.

A number of extensions to the RX algorithm have been proposed, many using a dimension reducing transform on the hyperspectral image data. By reducing the number of spectral bands from J to J', where $J' < J$, computational costs can be significantly reduced. Also, since N, the number of pixels covered by the spatial template, must be larger than J and, in practice, much larger than J, this has the additional advantage of reducing the size of the windows over which statistics must be computed.

Some extensions map $\mathbb{R}^J$ into (possibly lower dimensional) spaces which may impact the performance of the RX algorithm. These methods may alter the generalized signal-to-noise ratio, and hence the detection rate (P_D) for a given false alarm rate (P_{FA}).

In general, we have two primary options for where to perform the dimension reduction. We can perform dimension reduction on the image as a whole. In this case, we find the optimal[3] map using all of the data from the image. If we opt for this approach, we can apply the transformation to the image prior to the standard RX algorithm. A variation on this is to generate the "optimal" map based upon a uniformly drawn subset of the image data. This may be useful, if the dimension reduction method scales poorly with the number of data points, but will not necessarily result in the same map we would construct using the entire data set.

[2] This is a standard cost of computing the LU decomposition of a $J \times J$ matrix. This is true, even if we take advantage of the symmetric nature of the covariance matrix and use Cholesky factorization. Bunch and Hopcroft [8] present an $O(J^{\log_2 7}$ method for computing the LU decomposition of a $J \times J$ matrix based on a fast matrix multiplication algorithm proposed by Strassen [29]. This can be further improved by substituting a faster matrix multiplication algorithm.

[3] This is optimal in the sense defined by the specific dimension reduction method.

Alternatively, we can construct an optimal map for the block we construct around each pixel. This is reasonable, since we are only making comparisons within a block, not between blocks. This also guarantees that, in an image where there may be multiple distinct regions, the map used is always optimal for the region over which we are concerned. Unfortunately, this amounts to constructing a new map for every pixel, which means a possible increase in computational cost on the order of the number of pixels over the whole image method given above. The exact difference depends upon the dimension reduction method employed.

Since we are operating on video, we have a third option open to us. We can construct an optimal map based upon the initial image(s) in the sequence and apply this map to all subsequent images. This may be reasonable, if the video has similar illumination properties. In practice, this could mean recalibrating the system periodically. Depending upon the cost of computing the initial transform, this may significantly reduce the computational cost, though it may also impact the quality of the result.

2.2 Dimension Reducing Transforms

Principal components analysis (PCA) [6, 33], noise adjusted principal components analysis (NAPCA) [6, 18], kernel PCA (KPCA) [17, 12], compressive-projection PCA (CP-PCA) [10], locally linear embedding (LLE) and robust locally linear embedding (RLLE) [19] have been used in the literature to transform hyperspectral data from $\mathbb{R}^J$ to another space (or manifold). In this section we will present PCA and NAPCA, two linear dimension reducing transforms, that is, transforms which map a space $\mathbb{R}^J$ into a space $\mathbb{R}^K$, where $K < J$.

2.2.1 Principal Components Analysis

Given a zero-mean data set in $\mathbb{R}^J$, the idea behind principal components analysis is to find a linear subspace $\mathbb{R}^K$ $(K < J)$ which captures the maximum variance in the data. In other words, consider the set of N zero-mean, m-dimensional column data vectors $\mathbf{X} = \left[\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\right]$. The principal components are the basis vectors of this subspace in order of decreasing variance. The first principal component $\mathbf{y}^{(1)}$ solves the problem

$$\begin{aligned} &\arg\max_{\mathbf{y}} \ \|\mathbf{X}\mathbf{y}\|_2^2 \\ &\text{subject to } \ \mathbf{y}^T\mathbf{y} = 1 \ . \end{aligned}$$

In other words, $\mathbf{y}$ is the (unit) eigenvector of the covariance matrix of $\mathbf{X}$ corresponding to the largest eigenvalue. In fact, the second principal component is the (unit) eigenvector of the covariance matrix corresponding to the second largest eigenvalue. The eigenvectors of the covariance matrix $\mathbf{X}\mathbf{X}^T$ are also the left singular vectors $\mathbf{U}$ in the singular value decomposition (SVD) of $\mathbf{X}$, as noted throughout the literature, including in [25, 28, 27].

When using PCA to perform dimensionality reduction from $\mathbb{R}^n$ to $\mathbb{R}^m$, one simply projects the n-dimensional data onto the first m principal components. This preserves the most variance possible in an m-dimensional linear subspace over the class of all orthonormal linear mappings.

2.2.2 Noise Adjusted Principal Components Analysis

When applied to real data containing noise, PCA produces results that do not necessarily decrease in quality (increase in noise) with increasing dimension (K) of the subspace. To address this issue, Green, et al. [11] produced what they term the maximum noise fraction (MNF), and which is elsewhere [27] referred to as noise-adjusted PCA (NAPCA) .

In the context of hyperspectral imagery, assume each pixel to be a random vector $\mathbf{X}(z)) = \mathbf{S}(z) + \mathbf{N}(z)$, the sum of a signal component $\mathbf{S}(z)$ and a noise component $\mathbf{N}(z)$. Define the noise fraction of component i to be

$$\frac{\mathrm{Var}\left[N_i(z)\right]}{\mathrm{Var}\left[X_i(z)\right]}$$

As the name MNF implies, we want the linear transformation

$$\begin{aligned}\mathbf{Y}(z) &= \mathbf{A}^T\mathbf{X}(z)\\ &= \mathbf{A}^T\mathbf{S}(z) + \mathbf{A}^T\mathbf{N}(z)\end{aligned}$$

that solves the problem

$$\begin{aligned}\arg\max_{\mathbf{A}} \ & \frac{\mathrm{Var}\left[\mathbf{A}^T\mathbf{N}(z)\right]}{\mathrm{Var}\left[\mathbf{A}^T\mathbf{X}(z)\right]}\\ \text{subject to} \ & \mathbf{A}^T\mathbf{A} = \mathbf{I}\end{aligned}$$

Since we do not know the covariance of the noise $\langle \mathbf{N}(x), \mathbf{N}^T(x)\rangle$, Green, et al. suggest approximating it with the $\langle \mathbf{X}(x), \mathbf{X}(x+\Delta)\rangle$.

Originally, this was cast as the solution of two PCA problems. Roger [27] shows that solving these is equivalent to solving the generalized singular value decomposition problem.

2.3 Data Sets

We are interested in two types of data sets. First, aerial data sets, one of which we describe in Section 2.3.1. This is the type of data that the RX algorithm and its variants were designed to interpret. These data sets are characterized by small variations in distance between the landscape and the sensor. The anomalies typically sought are solid manmade artifacts like cars and roads.

We also are interested in terrestrial video. Specifically, we are interested in examples of biological and chemical agent standoff detection scenarios. The distance between the sensor and landscape vary greatly throughout an image in one of these sequences. Rather than detecting solid artifacts, we are want to identify gaseous and aerosol plumes which vary in shape and transparency throughout a video sequence. We describe three such data sets in Sections 2.3.2, 2.3.3, and 2.3.4.

2.3.1 Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)

AVIRIS is a typical aerial push-broom hyperspectral sensor. Its sensor points down from an airplane and records a single row of pixels in 224 spectral bands. As the plane moves forward, successive rows of data are collected, producing a single long image. Data from this sensor is often used to demonstrate hyperspectral anomaly detection algorithms in the literature. See [5, 12, 10, 7, 33] for examples. Even though it is not used to collect data in the domain we are interested in, it is included to show the RX algorithm working on its intended source material.

2.3.2 Fabry-Pérot Interferometer Sensor (FPISDS)

For this data set, a Fabry-Pérot interferometer is used as a spectrometer in the FPIS sensor [3]. The images produced are 256 × 256 pixels in 20 spectral bands. An image was recorded every 2 seconds. The events collected are chemical plumes produced using explosives. Five trials were provided, each with a single release of one of the following chemicals: glacial acetic acid, methyl salicylate, and triethyl phosphate.

2.3.3 Johns Hopkins Applied Physics Lab (JHAPL)

This data set uses between one and three FTIR longwave infrared sensors to record a sequence of images. The images are 128×320 pixels in 129 spectral bands and are collected every 5 seconds. The events collected are bursts of gas or combinations of liquid and gas. Events were collected from between one and three locations. Each sequence contains a single release of a single chemical. Chemicals used include isopropanol, ammonia, and sulfur hexafluoride.

2.3.4 Frequency Agile LIDAR (FAL)

Unlike any of the other data sets used in this thesis, the FAL data set is collected with an active rather than a passive sensor [1]. Instead of recording the reflected ambient light and naturally occurring emissions, it illuminates and excites the objects in the scene with a laser. It is also unique among these in that it produces a data cube whose axes are time, distance, and frequency, rather than time, x, y, and frequency.

The FAL sensor uses a laser to emit a periodic burst of coherent light at a specifically selected frequency. The light follows a ray into the scene, reflecting off of and exciting particles, biological agents, and objects. Reflected light will return with the same frequency as it was generated at. The laser can also excite objects which it comes in contact with (such as the mitochondria of specific strains of bacteria), causing them to fluoresce at different frequencies. The FAL sensor contains a spectrograph, which records both of these responses. In addition, since the laser light is emitted periodically, the time of flight is also recorded, resulting in a measure of the distance to the objects producing the response.

The events recorded are releases of dust, exhaust, smoke, spore simulant, and viral simulant in the Joint Ambient Breeze Tunnel (JABT), at a distance of over 1 km from the sensor. The sensor records 19 spectral bands, distance data is collected at 625 samples, and the laser bursts occur once per second. Each data collection run is between 900 and 1200 seconds long.

CHAPTER 3

IMPLEMENTATION DETAILS

The derivation of the RX algorithm is theoretically clear. In contrast, the implementation of the algorithm poses a range of challenges, in particular, dealing with sensitivity to parameter selection. The RX algorithm relies on an appropriate selection of the mean window size (L) and spatial template ($\mathbf{s}$). We have already addressed mean window size Section 2.1.1. We address the spatial template in Section 3.2.

At the heart of the RX algorithm is the r test statistic. A direct implementation can be numerically unstable. It also relies on $\mathbf{X}\mathbf{X}^T$ being full rank. We address these issues and provide a stable and efficient solution in Section 3.1.

Processing video sequences introduces the added challenge of a realtime requirement. Because frames arrive at regular intervals, detection in the current image must be completed before the next arrives. In Section 3.3, we describe the parallel implementation which we successfully applied to our video sequences.

3.1 Computing the Test Statistic

Recall that the test statistic used by the RX algorithm is

$$r(\mathbf{X}) = \frac{(\mathbf{X}\mathbf{s})^T \left(\mathbf{X}\mathbf{X}^T\right)^{-1} (\mathbf{X}\mathbf{s})}{\mathbf{s}^T\mathbf{s}}$$

Computing the inverse of a matrix should generally be avoided, if the result can be computed without it. More numerically stable methods exist [32, lecture 22]. Observe that we can rewrite

$$r(\mathbf{X}) = \frac{(\mathbf{X}\mathbf{s})^T \mathbf{y}}{\mathbf{s}^T\mathbf{s}}$$

where $\left(\mathbf{X}\mathbf{X}^T\right)\mathbf{y} = \mathbf{X}\mathbf{s}$. We can use the LU factorization of $\mathbf{X}\mathbf{X}^T$ ($\mathbf{L}\mathbf{U} = \mathbf{X}\mathbf{X}^T$ where $\mathbf{L}$ is lower triangular and $\mathbf{U}$ is upper triangular) and find Y using forward substitution to solve $\mathbf{L}\mathbf{z} = \mathbf{X}\mathbf{S}$ and then backward substitution to solve $\mathbf{U}\mathbf{y} = \mathbf{z}$ [15, section 3.5]. Alternately, we can take advantage of $\mathbf{X}\mathbf{X}^T$ being a symmetric matrix and use the Cholesky factorization ($\mathbf{X}\mathbf{X}^T = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is lower triangular with non-negative diagonal entries) in the same way.

It should be noted that it is possible for $\mathbf{X}\mathbf{X}^T$ to be (nearly) singular. In other words, the condition number may be (very large) infinite. The condition number is the ratio of the eigenvalues with the largest and the smallest magnitude. [1] If the condition number is too large, we do not attempt to label the pixel. Another option is to use the pseudoinverse[2] in place of $\left(\mathbf{X}\mathbf{X}^T\right)^{-1}$ when $\mathbf{X}\mathbf{X}^T$ is (nearly) singular.

3.2 The Spatial Template

The form of the spatial template $\mathbf{s}$ embodies assumptions about the scale of the anomalies and homogeneity of the "clutter".

Historically, RX and its variants are applied to (scanned) aerial imagery. In other words, the distance from the sensor to the landscape varies little when compared to the actual distance. Therefore, the scale is preserved in all directions. Therefore, it reasonable to use square spatial templates in this case. A circular template could also be used, but this complicates the implementation and reduces the amount of data available for parameter estimation.

With terrestrial imagery, the distance to the sensor can be anywhere from very small at the bottom of the image to effectively infinite above the horizon. If we were to use the

[1] Matlab approximates the condition number rather than explicitly performing the eigen factorization.

[2] The Moore-Penrose pseudoinverse can be computed using the SVD of $\mathbf{M}\mathbf{M}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. In this case, the pseudoinverse $(\mathbf{M}\mathbf{M})^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^T$, where $\mathbf{\Sigma}^+$ is the pseudoinverse of $\mathbf{\Sigma}$. We can compute $\mathbf{\Sigma}^+$ by taking the transpose of $\mathbf{\Sigma}$ and replacing the non-zero (or larger than some small ϵ) with their reciprocal [23].

same square spatial templates as in the literature, our clutter model could end up covering vastly different scales at the bottom and the top. Also, when we move across the horizon, we could end up with a significant change in the estimated covariance matrix, violating the assumption of a slowly varying covariance matrix. As a consequence, using a short and wide spatial filter keeps the parameter estimation drawn from terrain that is roughly the same distance from the sensor. We can do even better if we only draw clutter data from the left and right of the pixel under test, but not above or below.

A second issue still remains, which exists even when using aerial images. If the size and shape of the target does not exactly match what we are observing, we can end up with either target bleeding into the clutter estimation or with clutter falling into the signal parameter estimation. Since this method relies on accurate parameter estimation, this can result in incorrect labeling of pixels. This issue has been addressed in the literature [5, 21] by having a guard window between the target window and the clutter window. Data points in the guard window are ignored when computing r. Since the spatial coherence is not explicitly used in the derivation of the RX algorithm, this does not invalidate any of the theory that supports the RX algorithm.

3.3 Opportunities for Parallelism

The RX algorithm is trivially parallelizable. The classification of each pixel only depends on a small window of pixels and not on the classification of any other pixel. As a consequence, if we have four processing units, we can process a quarter of the pixels on each one. In a shared memory architecture, like a modern multi-core system, we do no even need to concern ourselves with passing only the portion of the data cube that will be needed. To take advantage of cache coherency, though, we should divide the cube with the grain, just as we should loop through the pixels with the grain. For example, in a C implementation operating on a four core system, we should pass each processing unit a quarter of the rows. In Matlab or Fortran, however, we should pass each unit a quarter of the columns.

The RX algorithm and many of the dimension reduction techniques are built upon standard linear algebra operations. In addition to this coarse-grain parallelism already noted, we can also exploit fine-grain parallel implementations of standard linear operations. Depending upon the system architecture and the size of the matrices involved, these may result in an improvement in performance, though they can also result in a decrease in performance in some cases. Some modern linear algebra libraries, such as ATLAS, automatically attempt to optimize their performance based upon the system they are executing on.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Experimental Protocol

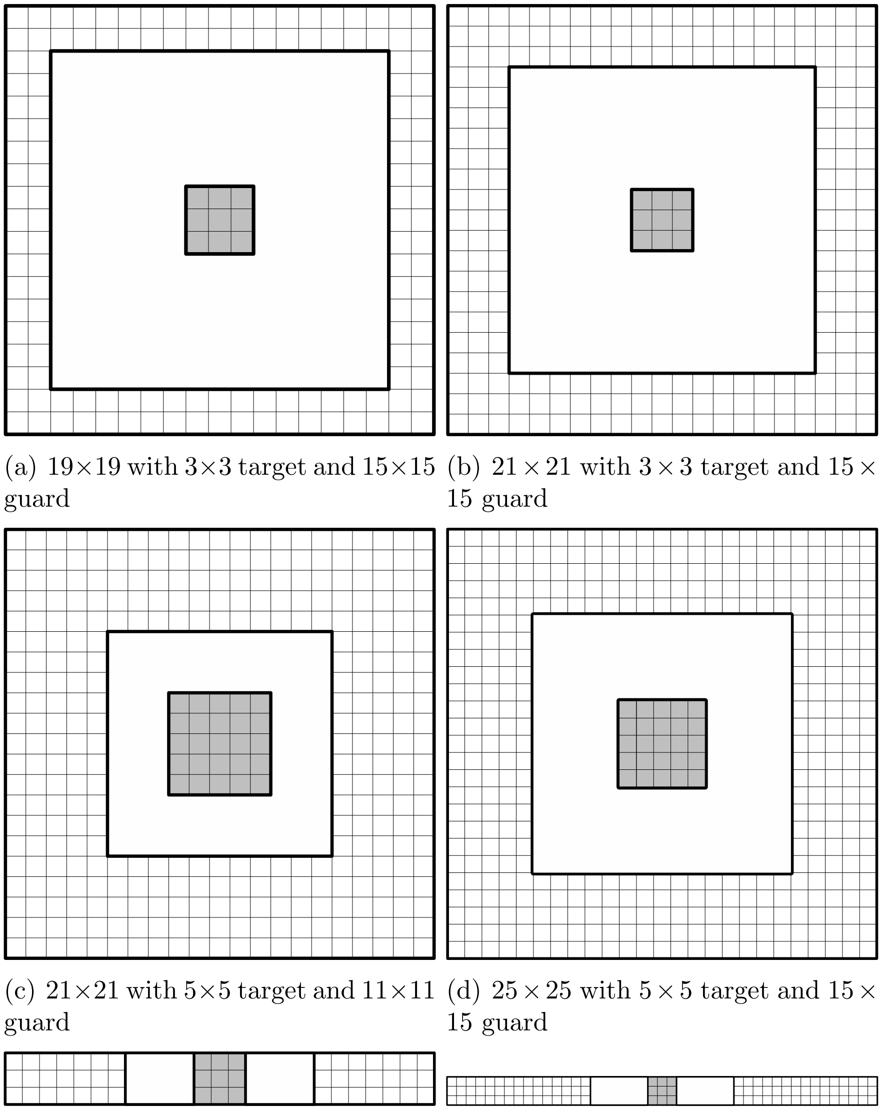
Since the goal of this study is to explore the performance of the RX algorithm and some of its descendants on terrestrial plume data, we do no more preprocessing on the images than is called for by the particular algorithm. Where indicated, we do replace extreme outliers (both maxima and minima) with local averages. We do not use background subtraction or any other techniques which have proven useful with other detection and classification techniques, though.

We read the raw data one frame at a time.[1] We then apply the selected algorithm to it (including any associated dimension reduction). Then we write the label map to disk.

We compute the local mean over a 5×5 window. When computing over different size windows, the distribution tends to be bimodal, with the upper mode decreasing in size with larger window sizes. However, the window size quickly becomes larger than the tiles used by the RX algorithm. Matteoli [21] notes that this should be avoided, hence the smaller, suboptimal window size.

The spatial windows used with the RX algorithm for all terrestrial (FPISDS, JHAPL, and FAL) data sets are shown in Figure 4.1. These were the best performing of a larger set of windows applied to the synthetic data. Both square and horizontal rectangular templates were used. The former were more traditional, while the later were included to restrict the range of depth covered in the terrestrial images.

[1] In the case of FAL data, where there is a single data cube that records the full timeline, we process an entire experiment in the same way that we would process a single frame from any of the other data sets.



(a) 19×19 with 3×3 target and 15×15 guard

(b) 21 × 21 with 3 × 3 target and 15 × 15 guard

(c) 21×21 with 5×5 target and 11×11 guard

(d) 25 × 25 with 5 × 5 target and 15 × 15 guard

(e) 3 × 25 with 3 × 3 target and 3 × 11 guard

(f) 3 × 45 with 3 × 3 target and 3 × 15 guard

Figure 4.1: Spatial templates used in FPISDS, JHAPL, and FAL tests.

RX was run on the original number of spectral bands for the data set. PCA-RX and NAPCA-RX reduced the data to 6 bands before applying the RX algorithm.

The processing times reported were collected on a Matlab implementation running with twelve processing threads (1 thread per core) on modern COTS (commercially available, off-the-shelf) computers. [2] Parallel implementations of linear algebra routines produced worse performance on the test systems, so single-threaded versions were employed. Per frame processing times are wall clock time and include reading the image from network storage as well as all processing necessary to produce a label image.

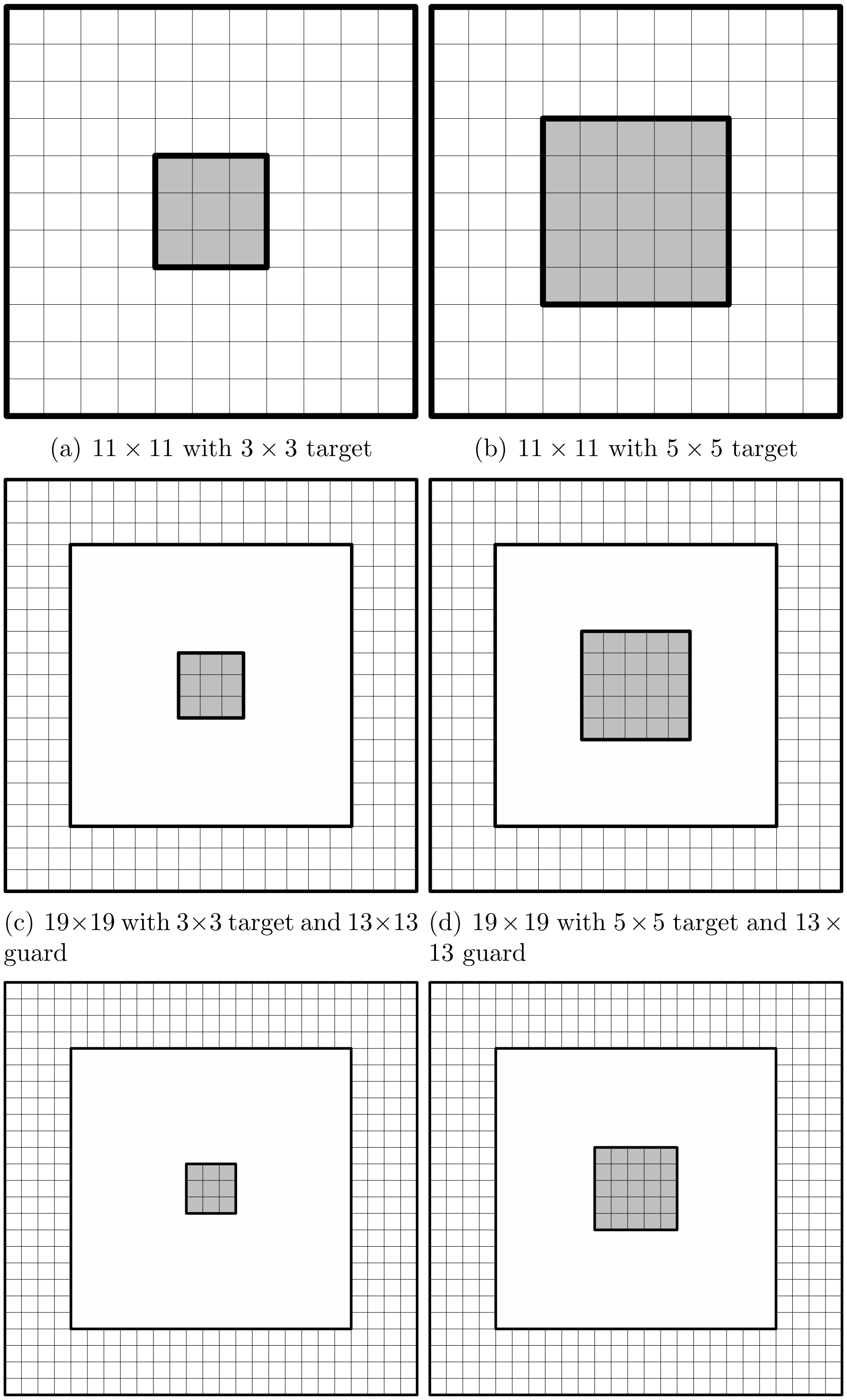
4.2 Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)

To test the operation of our implementations of the RX, PCA-RX, and NAPCA-RX algorithms, we applied it to the Moffett Field dataset [2], a subset of which was used in [10]. This data set is freely available and covers terrain similar to that in the various San Diego data sets which are frequently seen in the literature.[3] The first three principal components of this data cube mapped to red, green, and blue are shown in Figure 4.3(a). The local mean is computed over a 9×9 window, chosen because it is sufficiently local for all of the spatial template sizes. P_{FA} values of 10^{-3}, 10^{-4}, and 10^{-5} are used. The spatial templates shown in Figure 4.2 were used.

Example results for the spatial template in Figure 4.2(c) and $P_{FA} = 10^{-4}$ are shown in Figure 4.3. The first observation we can make is that, because the number of spectral bands in the raw image (224) is so large, many of the windows are too small to use with the basic RX algorithm. Even the ones which are large enough still have such large thresholds that no pixels fall above this. More experimentation is necessary to determine if the fault is in the selection of the mean window size or if much larger clutter windows are necessary.

[2] Two hardware configurations were used. The first contains four AMD Opteron 6174, 12 core CPUs running at 2.2 GHz. The other has eight AMD Opteron 7276, 8 core CPUs running at 2.3 GHz. All systems had 512 GB RAM and gigabit ethernet and were running CentOS 6 Linux. Since neither memory nor processing resources were fully utilized, similar results are expected on any system with at least 12 cores and sufficient free memory to hold the source image, dimension reduced image, mean subtracted image, and label image. In practice, this is less than 4 GB on even the large AVIRIS image.

[3] Optical color images of San Diego are used in [9, 26]. A subset of an AVIRIS data set of San Diego is used in [7, 33].



(a) 11 × 11 with 3 × 3 target

(b) 11 × 11 with 5 × 5 target

(c) 19×19 with 3×3 target and 13×13 guard

(d) 19 × 19 with 5 × 5 target and 13 × 13 guard

(e) 25×25 with 3×3 target and 17×17 guard

(f) 25×25 with 5×5 target and 17×17 guard

Figure 4.2: Spatial templates used in AVIRIS tests.

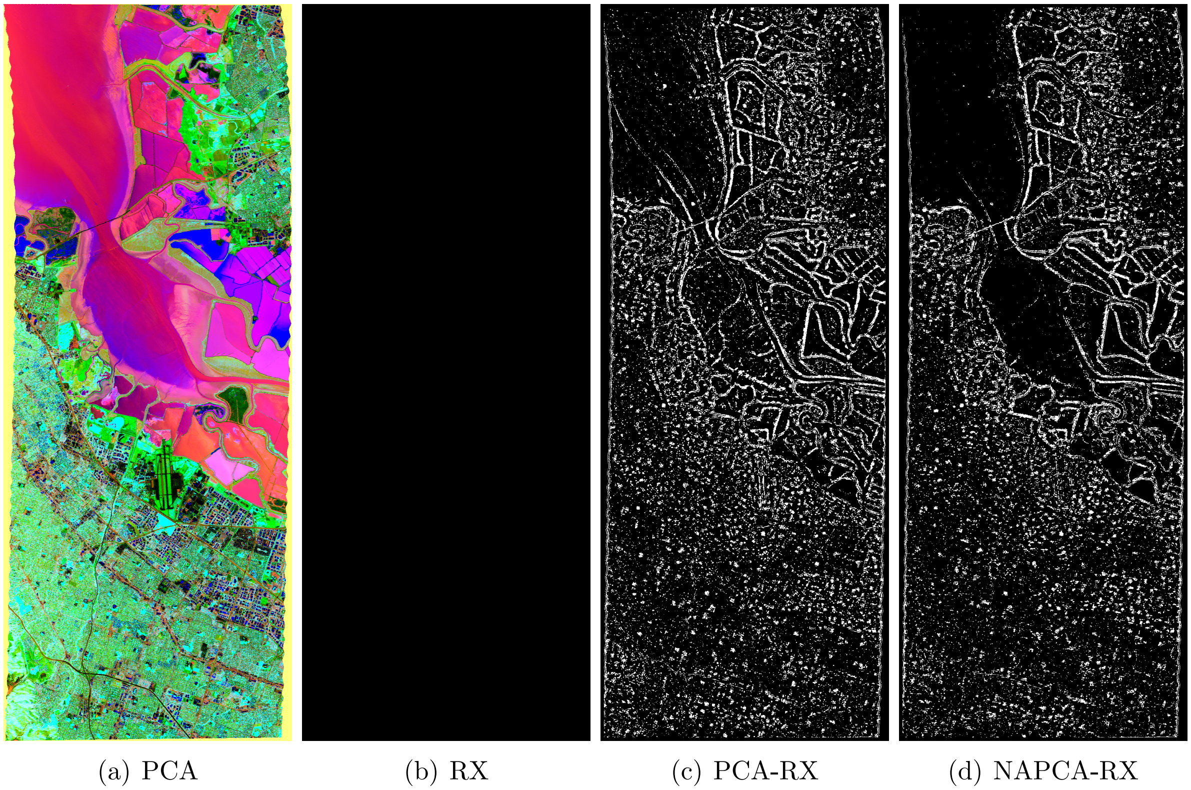
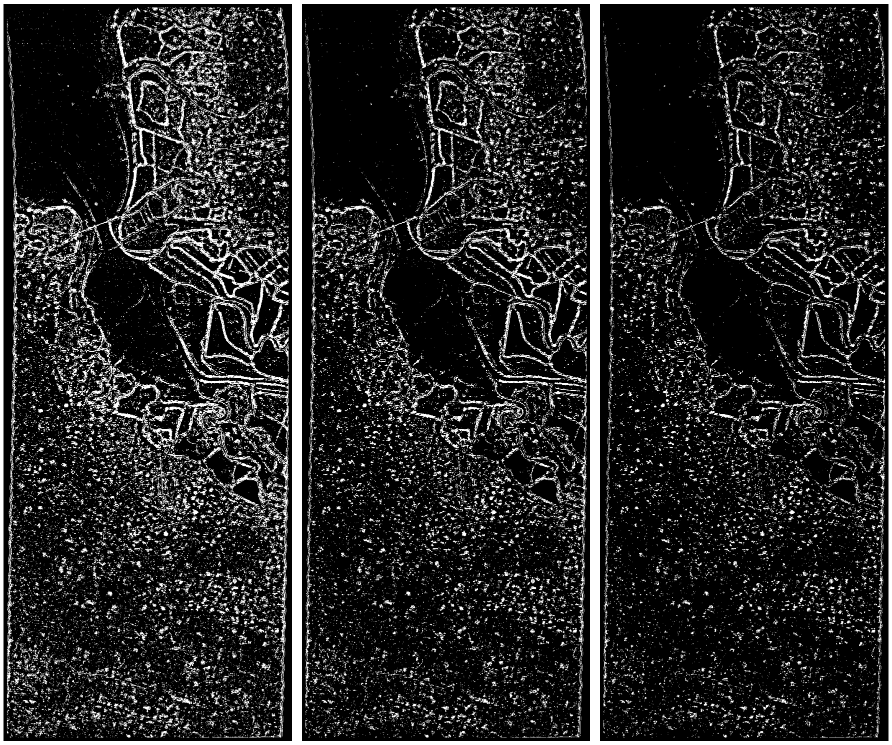

(a) PCA (b) RX (c) PCA-RX (d) NAPCA-RX

Figure 4.3: Moffett Field results using 19×19 spatial template with 3×3 target and 13×13 guard and $P_{FA} = 10^{-4}$.

Now we will turn our attention to PCA-RX and NAPCA-RX which, by virtue of utilizing only six bands, produce reasonable thresholds for all of the spatial templates. PCA-RX produces a noisier label image that NAPCA-RX. This is especially noticeable in the bay, which shows few returns in NAPCA-RX, but many scattered through it in the label image produced by PCA-RX. However, PCA-RX also identifies the runways immediately below the bay, while NAPCA-RX does not.

Figure 4.4 shows the result on NAPCA-RX of varying P_{FA} from 10^{-3} to 10^{-5}. While there are fewer spurious detections with higher P_{FA}, it is not a drastic difference. This suggests that there is a significant difference between the natural features and human features and hence that performance is not sensitive to P_{FA} selection.



(a) $P_{FA} = 10^{-3}$ (b) $P_{FA} = 10^{-4}$ (c) $P_{FA} = 10^{-5}$

Figure 4.4: NAPCA-RX applied to Moffett Field using 19 × 19 spatial template with 3 × 3 target and 13 × 13 guard for different values of P_{FA}.

4.3 Synthetic

To investigate the "ideal" performance of the algorithms, we generated simple synthetic data. 128 × 128 images in 20 spectral bands were generated. The background was drawn from a uniform distribution over the interval $[0, 1]$. Then a square of ones was added to the center of the image. The size of the square ranged from 1 × 1 to 25 × 25, to test the response of the algorithm to features that ranged from smaller than the target window to overlapping the guard window and finally overlapping the clutter window. Initial tests indicated that the results were dependent upon the size of the local mean window, so we also varied its size from 3 × 3 to 25 × 25.

Figure 4.5, shows the response of the standard RX algorithm using the 25 × 25 spatial template. The mean window size and target size are varied to show responses smaller than the target window, equal to the target window, in the guard window, and in the clutter window. What we see is that the response is weak or non-existant when the target is smaller than the target window. When it is equal to the target window or in the guard window, we get a response, and when it is in the clutter window, we get identifications outside of the target, rather than within the target. The best response is when the size of the mean window is within the guard window and worst when it is within the target window.

Even so, response on a target that is exactly the size of the target window is still incomplete. This is both due to the way in which local mean subtraction is performed and the amount of clutter included in the target window on the boundaries. This illustrates one of the difficulties in predicting how the RX algorithm will perform on a given scene. Since we are really classifying the central pixel in a target block, the boundaries of a target can be unclear.

Figure 4.6 shows empirically derived ROC curves for the corresponding templates in Figure 4.1. These were generated by varying the size of the $L \times L$ of the window over which the local mean is computed. Each line represents a different size target ranging from 1 to 25. The threshold r_0 was selected based upon a theoretical $P_{FA} = 10^{-3}$. The templates chosen were the ones which did not exceed an empirical false alarm rate of 10^{-2} (e.g., the ones which remained near the correct order of magnitude).

Figures 4.7 and 4.8 show how the empircal rate of false alarm and rate of detection vary as the size of the $L \times L$ local mean window varies. For target sizes for which the templates are capable of performing well, the maximum detection rate is generally reached when L is between 7 and 11. The rate of false alarm is approximately the theoretical value (10^{-3} in this range. In the following experiments, L was selected to be 9 for templates (a), (b), (d), and (e) and 11 for templates (c) and (f) based upon inspection of these plots.

4.4 Fabry-Pérot Interferometer Sensor (FPISDS)

Figure 4.9 shows the first three noise-adjusted principal components mapped to red, green, and blue for two images from a FPISDS sequence. This is included to give the reader a sense for the content of the scene. The first image contains no plume. The second image contains the plume, visible on the horizon just left of the mountain. Figure 4.10 shows RX, PCA-RX, and NAPCA-RX using the 3×39 spatial template in Figure 4.1(f) applied to the two images. Figure 4.11 shows the same images but using the 21×21 spatial template in Figure 4.1(b). Each has a 3×3 target window and a 15 pixel wide guard window.

First, let us look at the plume. In the rightmost images, it appears just to the left of the mountain. This is visible in all of the label images produced using the 3×45 spatial template. It is clearest when using the basic RX algorithm on the raw set of spectral data. PCA-RX and NAPCA-RX both produce a sparser set of detections. At this point, it cannot be said whether this indicates a lack of discriminating information in the dimension reduced data or false positives when using all spectral bands. Using the 21×21 spectral template produces a strong horizon line which masks much of the plume.

Notice that, even though the most extreme outliers have been replaced by local averages, remaining outliers result in blocks of detections. This demonstrates that, when the data does not fit the nice smooth distribution assumed by the model, artifacts can result. These occur in both the basic RX and PCA-RX algorithms, but not in the NAPCA-RX results, demonstrating that NAPCA can remove outliers, while PCA may fail to.

We see a related problem along the horizon, especially when using the square 21×21 spatial template. Because the clutter window covers both sky and ground, two distinctly different classes, we end up with detections all along the horizon. The rectangular 3×45 template rejects much of the horizon, since its clutter window looks only at other points along the horizon. This fails at the mountain on the far right. Here, the left half of the clutter window covers sky, while the right covers land.

Table 4.1: Average time (in seconds) per frame of RX variants applied to FPISDS sequences. Templates are identified as in Figure 4.1.

	(a)	(b)	(c)	(d)	(e)	(f)
RX	1.7799	1.8491	1.9713	2.0136	1.7134	1.7288
PCA-RX	1.4343	1.4250	1.4156	1.4182	1.4698	1.4035
NAPCA-RX	1.5008	1.4518	1.4770	1.4454	1.4986	1.4322

Both templates respond to the coarse texture in the foreground. With the variation in scale, items like rocks in the foreground and a gas plume on the horizon cover a similar number of pixels in the image plane. PCA-RX and NAPCA-RX both produce more detections in the foreground than does basic RX. This may be due to the removal of low-variance components of the signal in the clutter window, or it may be due to the difference in the threshold resulting from the difference in J, the number of bands.

It should be noted that this sequence produced the clearest result in the Fabry-Perot data set. In general, the background of the other sequences looked similar. However, there were no clear plumes visible in any of them. This may be due to differences in the scale of the plume or the signature of the other chemicals being released. Further investigation is necessary to determine whether any of the RX variants tested are capable of detecting the events in these other sequences.

Table 4.1 shows average time per frame for each combination of algorithm and spatial template. These times include reading in the image from network storage, removing dropouts, and performing any dimension reduction, as well as applying the RX algorithm. The video is collected at the rate of one frame every two seconds, so in all cases, the algorithm executes in real time on a COTS (commercially available off-the-shelf) system. In addition, since the rectangular window contains fewer pixels, it runs faster than any of the square templates. This indicates that design of the spatial template can not only improve detection, but can also be used to improve processing time.

4.5 Johns Hopkins Applied Physics Lab (JHAPL)

The JHAPL data provides some large, clear examples of plumes. Additionally, the events which were produced using explosions provide readily visible ejecta. These are more clearly visible than the gas-propelled plumes. In Figures 4.12 and 4.13, results on a pre-event image and an image from during an explosion-driven event are shown. The spatial extent of the event is much larger than in the FPISDS data and occurs below the horizon line.

In this case, performance on the raw data is significantly worse than on FPISDS. Due to the large number of spectral bands, the rectangular window suffers from the same problem observed with the AVIRIS data; the threshold is too high, resulting in no detections. Using a larger square spatial template, we are able to detect a small number of pixels in the event, but the responses to extreme pixel values dominates the result. Both PCA-RX and NAPCA-RX perform better, in part due to the reduced number of bands. In addition, NAPCA cleans up the extreme values as we saw with the FPISDS sequence. NAPCA-RX also detects more of the event boundary than does PCA-RX. Notice that, since inside plume, the clutter window contains a greater quantity of data within the plume, these are not identified as outliers in any of the sequences.

Once again, the rectangular template handles the variation in scale better than the square one does. With the square template, we once again see detections along the horizon. Additionally, the large square window produces artifacts such as double lines. This is similar to some of the responses seen with the synthetic data, when large targets interact with the clutter window, causing non-target pixels to be identified as anomalies.

There is a great deal of variety in the quality of the results on the JHAPL data set. While the examples selected show a clear response, some of the other images show little or no response. In some cases this may be due to the small scale of the release. In others, terrain features such as rocks dominate the images, much as they did in the FPISDS data. While some of the results are encouraging, more experimentation is necessary to determine whether the algorithms can be tuned to work well on the others.

Table 4.2: Average time (in seconds) per frame of RX variants applied to JHAPL sequences. Templates are identified as in Figure 4.1.

	(a)	(b)	(c)	(d)	(e)	(f)
RX	8.3798	9.6864	10.5579	11.4630	5.7872	6.3462
PCA-RX	3.0611	4.6438	4.4955	4.6921	3.0842	3.0837
NAPCA-RX	3.2327	3.2335	3.2436	3.3094	3.2526	3.2558

The JHAPL data was collected at the rate of one frame every five seconds. Table 4.2 contains the average time per frame for each algorithm-template pair. Unlike the results when the algorithms were applied to FPISDS sequences, in this case the basic RX algorithm is not running in realtime. It is only off by less than a factor of three, though. It may be possible to make up for this difference with a number of standard techniques. This does demonstrate another way in which the RX algorithm does not scale well with increased spectral bands; increasing spectral bands increases the execution time significantly. In this case, the dimension-reducing algorithms, while having more work upfront, see a much smaller increase in computation time.

4.6 Frequency Agile LIDAR (FAL)

The FAL data provides some of the clearest results. Some of this is due to the more controlled environment of the Joint Ambient Breeze Tunnel. Release events are clearly visible in all of the sequences. See Figures 4.14 and 4.15 for two examples. The results using the rectangular spatial template in Figure 4.1(f) (3 in the temporal and 45 in the distance direction) are much noisier than those using the 25 × 25 template in Figure 4.1(d). This suggests that emphasizing information in the distance direction (few time samples) does not provide a high quality model for the clutter process. Incorporating more temporal information improves the performance.

Due to the slow sample rate (1 Hz), the inclusion of more temporal information–forward as well as backward looking–means that results will be delayed, though. It is worth inves-

Table 4.3: Average time (in seconds) per sequence of RX variants applied to FAL. Templates are identified as in Figure 4.1.

	(a)	(b)	(c)	(d)	(e)	(f)
RX	13.6990	14.9191	16.1091	17.4537	11.5407	12.6413
PCA-RX	10.6244	10.8803	11.1746	11.4557	10.1185	10.3273
NAPCA-RX	10.3725	10.6287	10.9053	11.1244	9.8735	9.9881

tigating a spatial-temporal template that only looks backward in time, as this would be necessary to produce timely results in the field.

The length of each sequence varies, but each are on the order of 1000 seconds. Average execution times for processing an entire sequence are given in Table 4.3. Clearly, this implementation runs significantly faster than real time. While dimension reduction has some up front costs, it ultimately does result in faster run times than the basic RX algorithm.

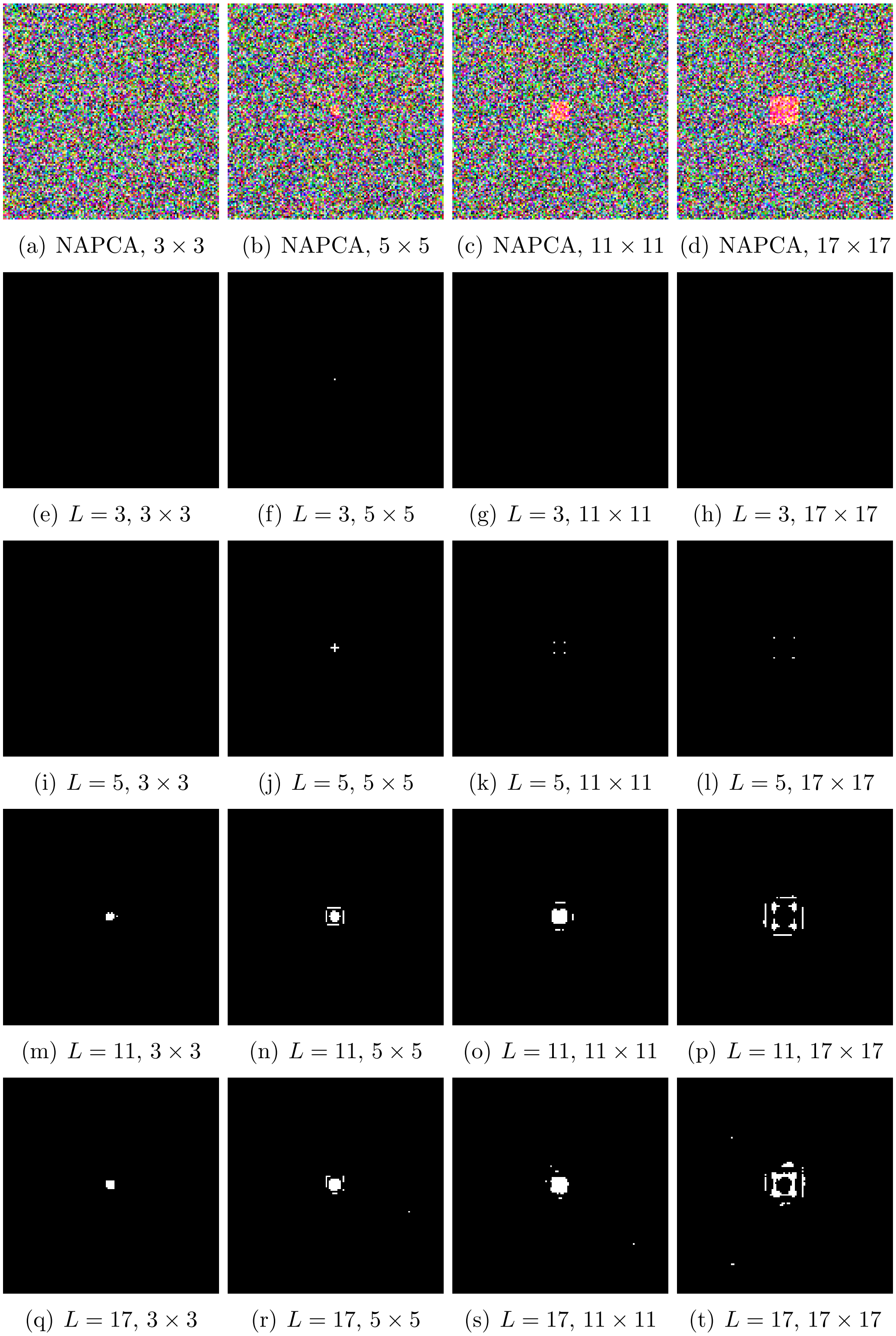
4.7 General Observations

One of the themes which ran through all of the testing is that the exact parameter selection can significantly impact the performance of the algorithm. This includes both the selection of the spatial template s, as well as the choice of the local mean window, L. The choice of P_{FA} was the one exception we saw, although this was only tested on a single data set.

Additionally, outlier data can result in anomaly responses in a string of images. Even identifying these points and replacing them with the mean of neighboring pixel values does not completely remove the problem. Especially when applying the RX algorithm to the full hyperspectral image, the outliers can produce responses in the surrounding pixels, resulting in small boxes that are a function of both the size of the mean window and of the target mask.

In some sequences, landscape features such as the horizon can dominate the results, obscuring the release event. It is likely that tuning the spatial template and mean window size could improve stationary feature rejection. Additionally, it has been suggested that performing background subtraction based upon early frames could also aid in stationary feature rejection.

The original papers applying dimensionality reduction emphasized the necessity of doing so to keep them computationally tractable. This did not appear to be a great of an issue on modern computers. However, running RX on images with more than a hundred spectral bands proved problematic for other reasons. A primary problem was in justifying a spatial template that was large enough to balance the number of spectral bands while not ignoring local variations in the spectral makeup of the data. As already stated, further investigation into this issue is necessary.



(a) NAPCA, 3×3 (b) NAPCA, 5×5 (c) NAPCA, 11×11 (d) NAPCA, 17×17

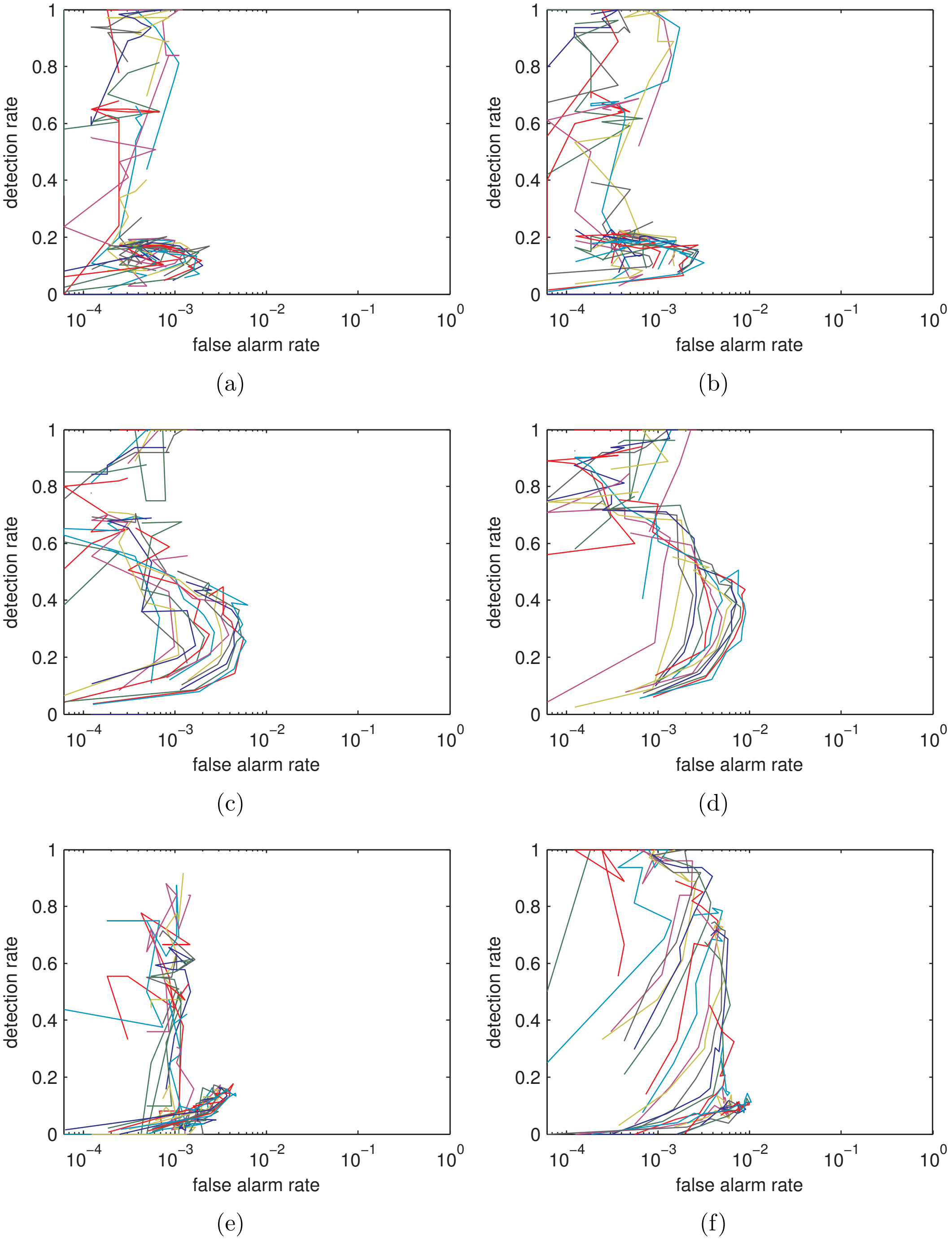
(e) $L = 3$, 3×3 (f) $L = 3$, 5×5 (g) $L = 3$, 11×11 (h) $L = 3$, 17×17

(i) $L = 5$, 3×3 (j) $L = 5$, 5×5 (k) $L = 5$, 11×11 (l) $L = 5$, 17×17

(m) $L = 11$, 3×3 (n) $L = 11$, 5×5 (o) $L = 11$, 11×11 (p) $L = 11$, 17×17

(q) $L = 17$, 3×3 (r) $L = 17$, 5×5 (s) $L = 17$, 11×11 (t) $L = 17$, 17×17

Figure 4.5: Response on synthetic targets of varying sizes (N) compared with mean window (L). All results using standard RX with the 25×25 spatial template.



(a)

(b)

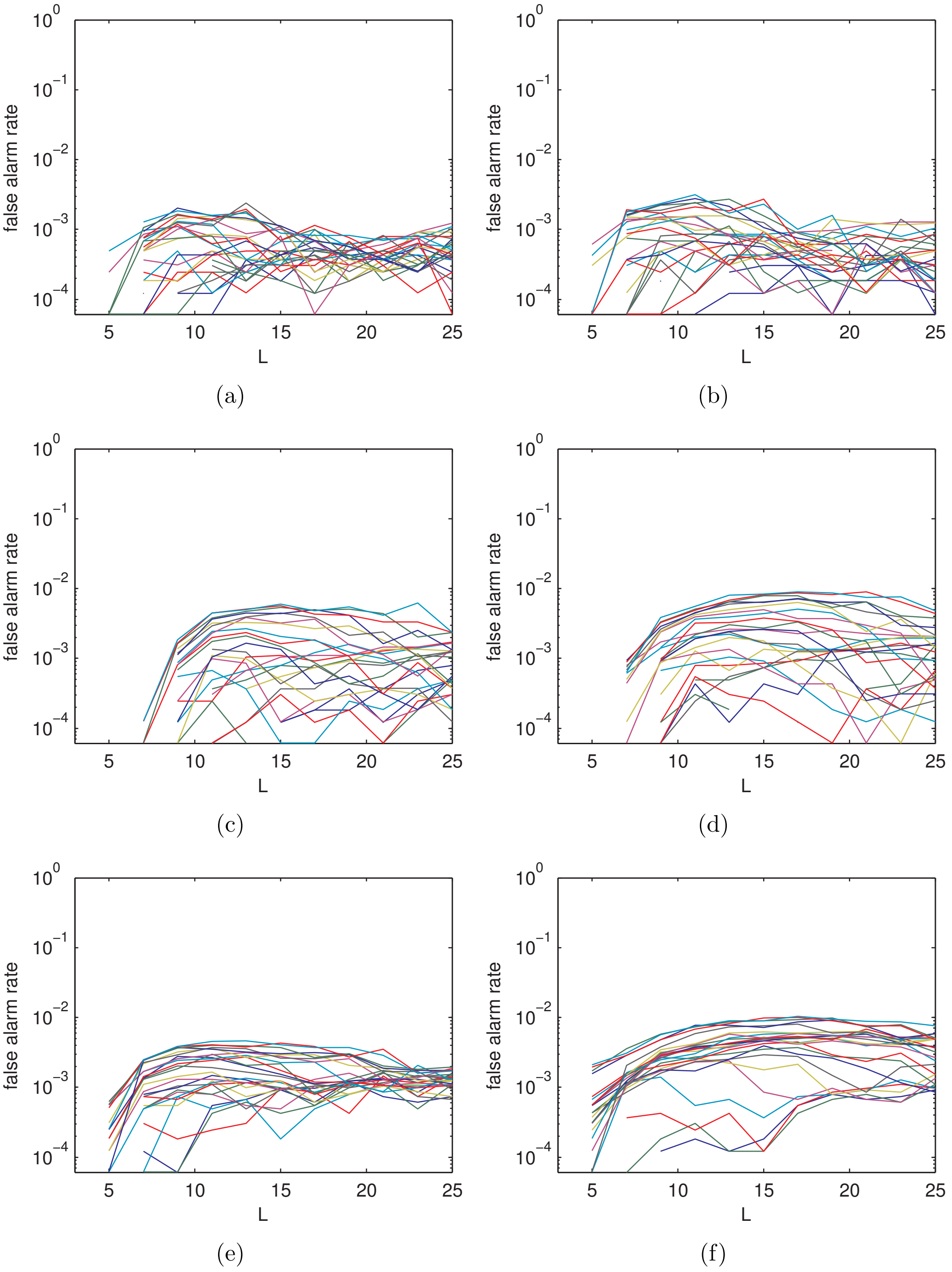
(c)

(d)

(e)

(f)

Figure 4.6: Empirically derived ROC curves for the templates in Figure 4.1. Local mean window size (L) is allowed to vary between 3 and 25. Each line represents a different target size from 1×1 to 25×25.



(a)

(b)

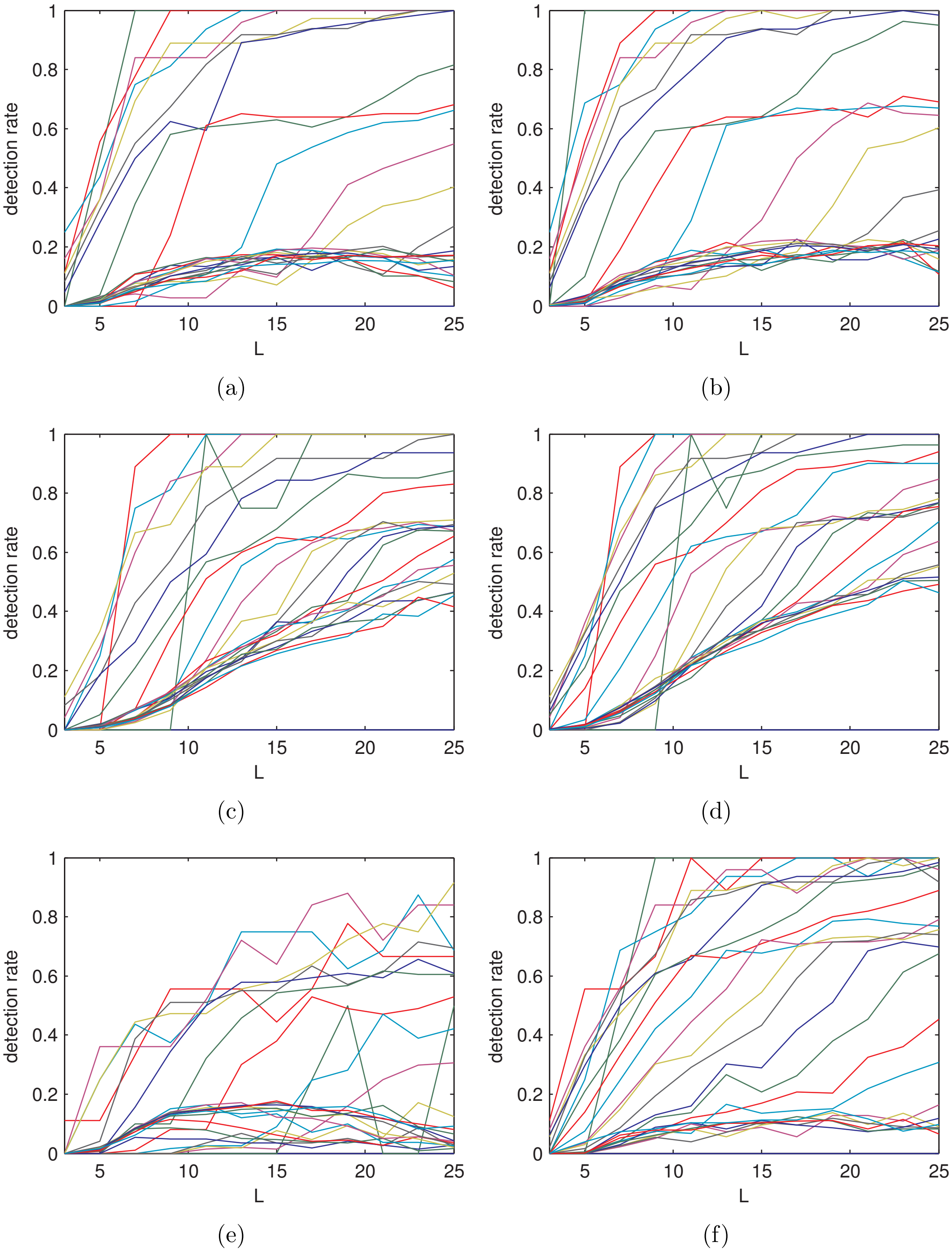
(c)

(d)

(e)

(f)

Figure 4.7: Empirically derived P_{FA} versus local mean window size (L) for the templates in Figure 4.1 . Each line represents a different target size from 1 × 1 to 25 × 25.



(a)

(b)

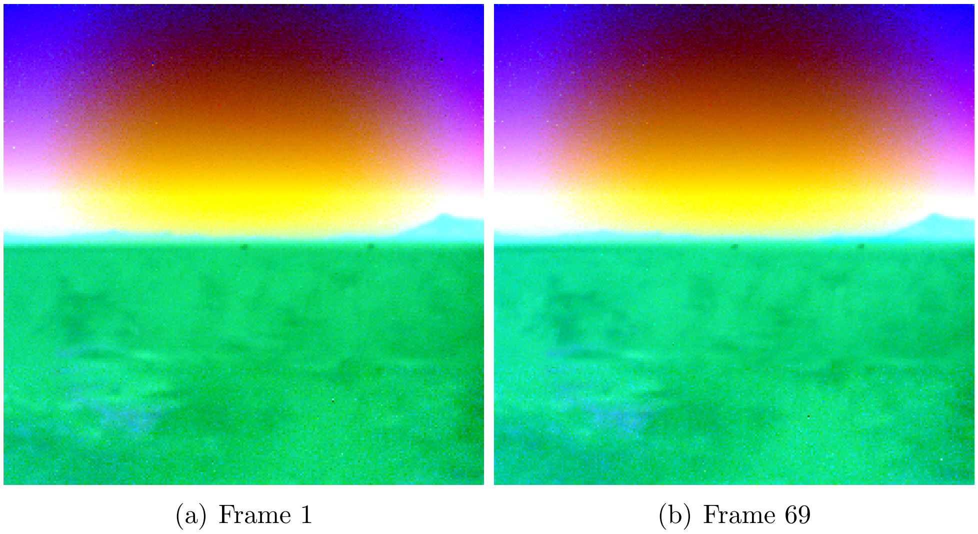
(c)

(d)

(e)

(f)

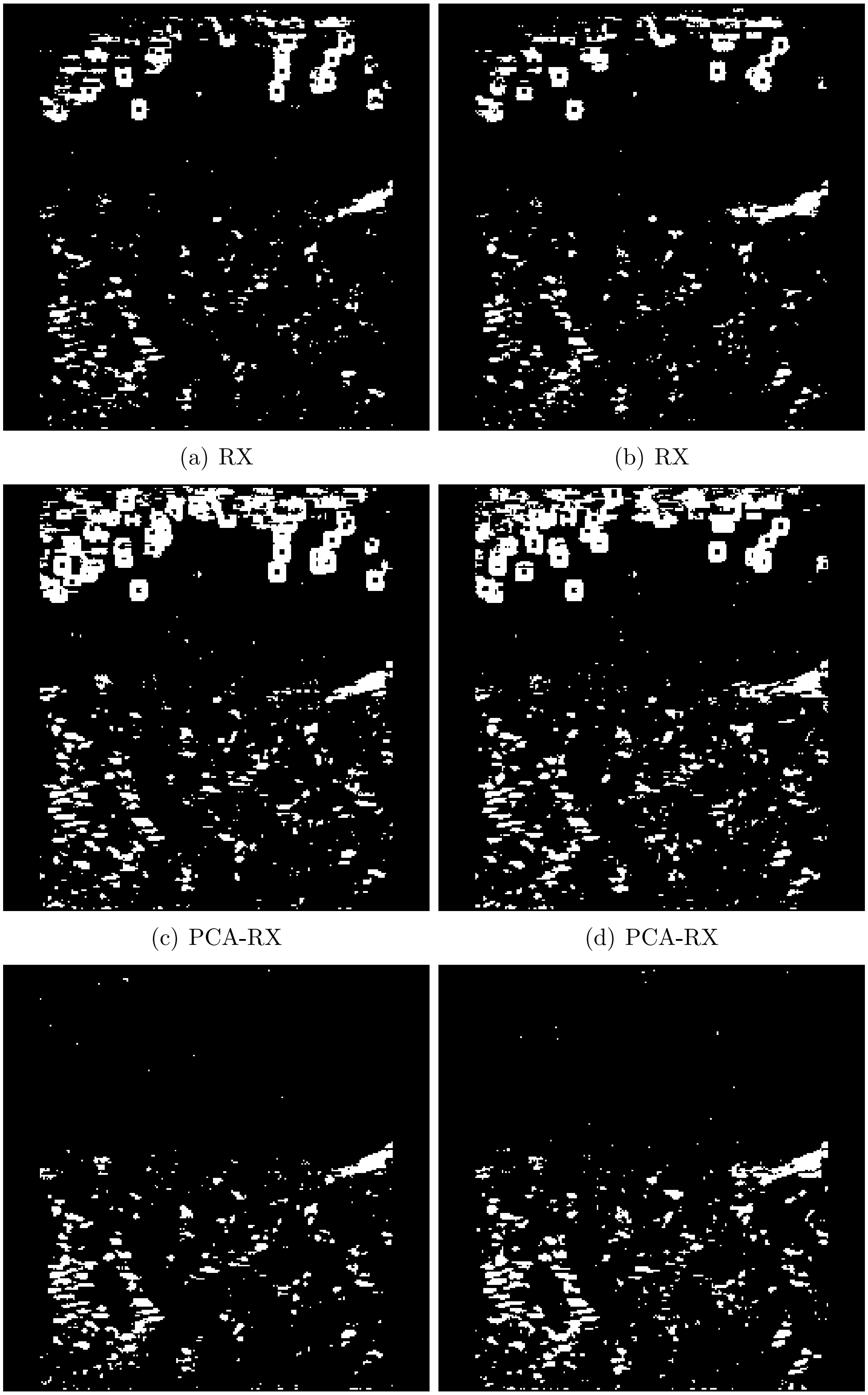
Figure 4.8: Empirically derived P_D versus local mean window size (L) for the templates in Figure 4.1. Each line represents a different target size from 1 × 1 to 25 × 25.



(a) Frame 1

(b) Frame 69

Figure 4.9: First three noise-adjusted principal components of two frames from a FPISDS sequence.



(a) RX

(b) RX

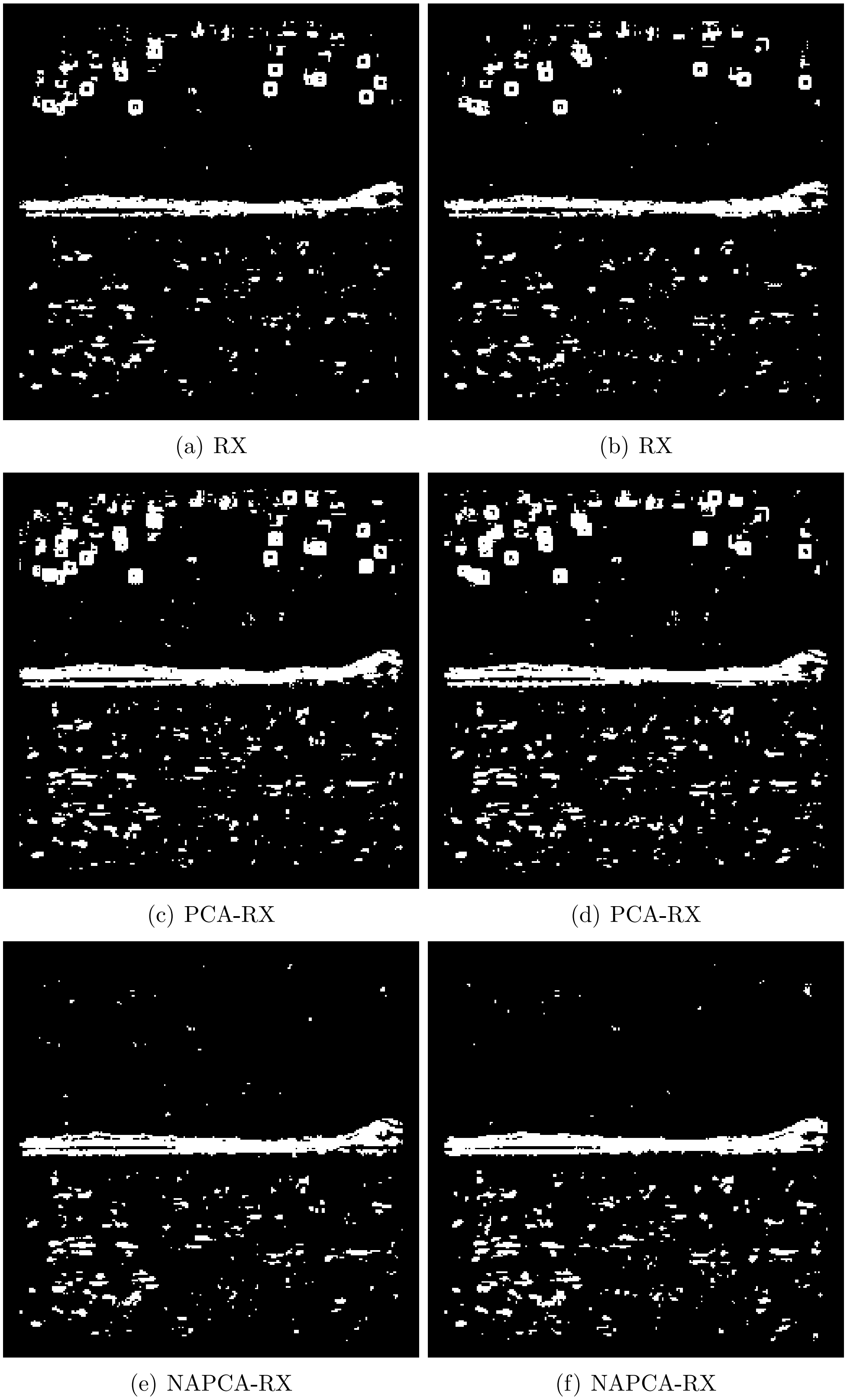
(c) PCA-RX

(d) PCA-RX

(e) NAPCA-RX

(f) NAPCA-RX

Figure 4.10: RX variants applied to two images from a FPISDS sequence, using the 3 × 45 template with 3 × 3 target and 3 × 15 guard windows.



(a) RX

(b) RX

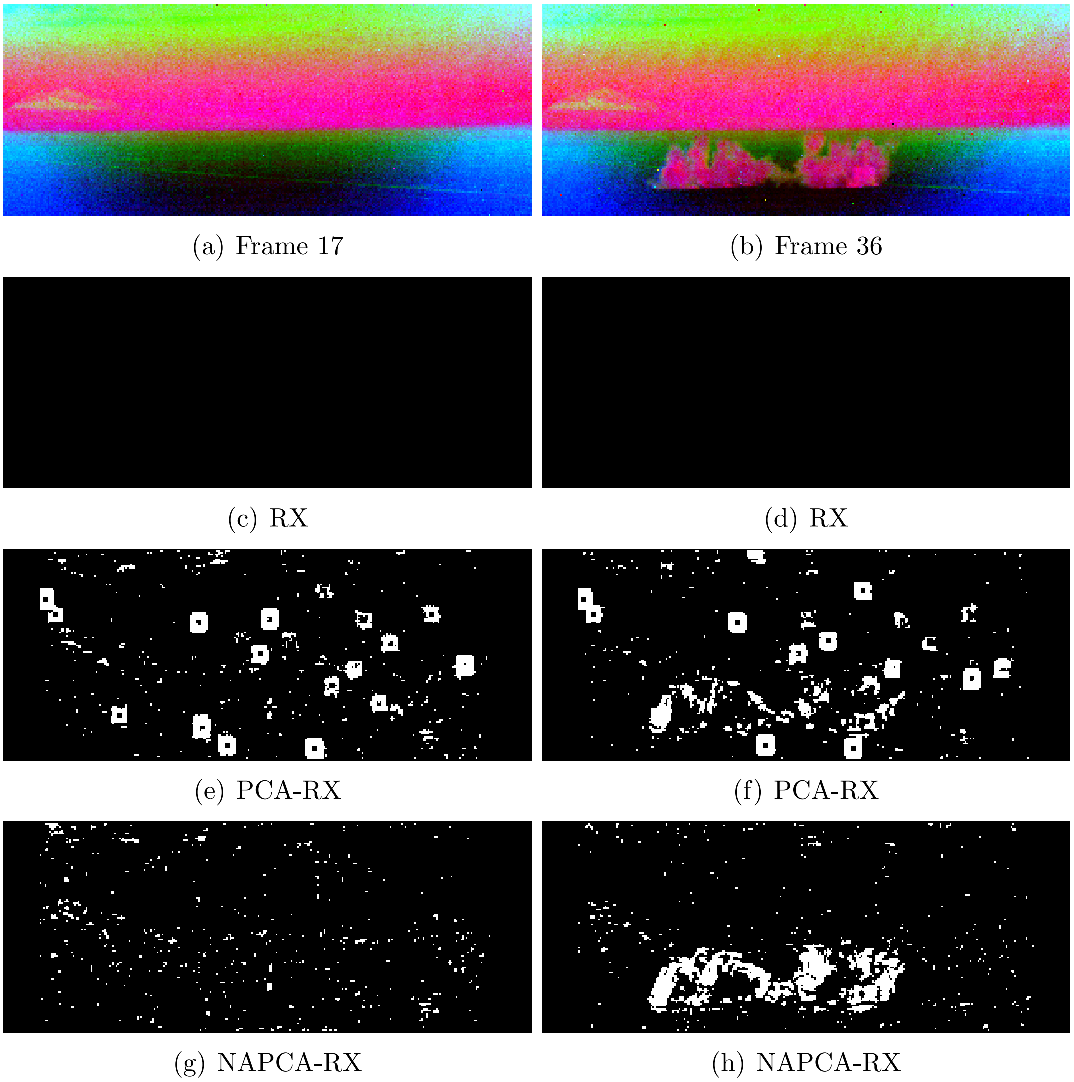
(c) PCA-RX

(d) PCA-RX

(e) NAPCA-RX

(f) NAPCA-RX

Figure 4.11: RX variants applied to two images from a FPISDS sequence, using the 21×21 template with 3×3 target and 15×15 guard windows.



(a) Frame 17

(b) Frame 36

(c) RX

(d) RX

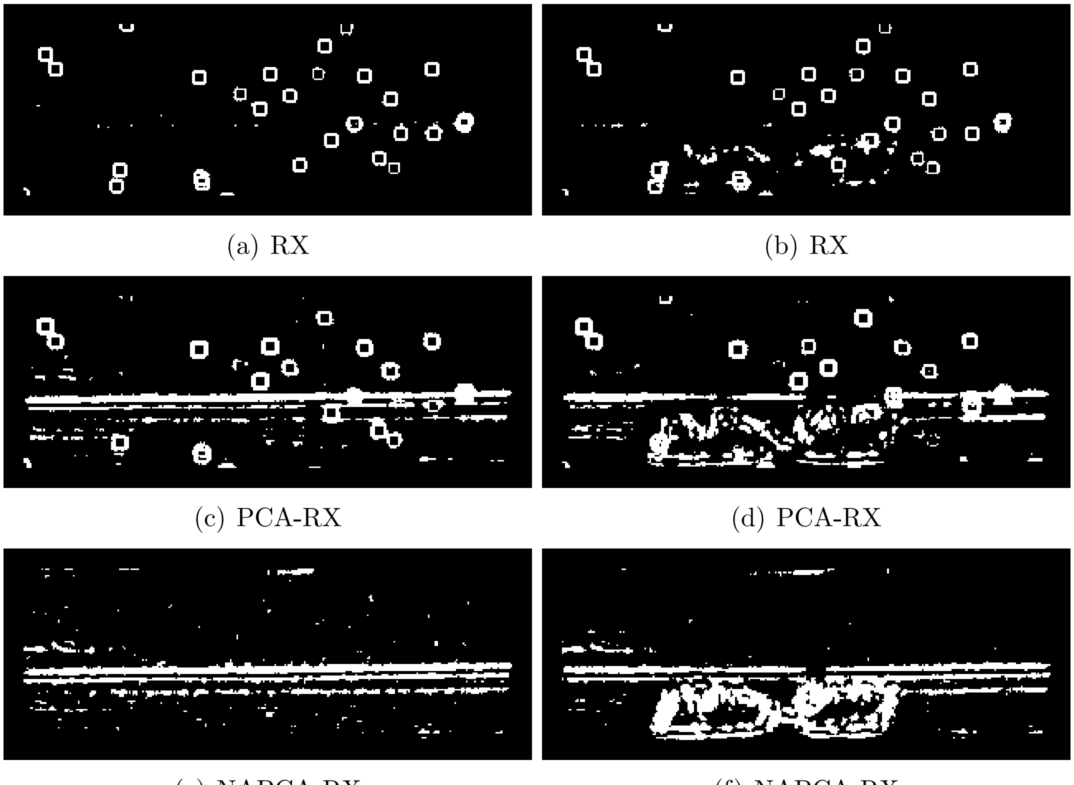
(e) PCA-RX

(f) PCA-RX

(g) NAPCA-RX

(h) NAPCA-RX

Figure 4.12: (a) and (b) show the first three noise-adjusted principal components of two frames from a JHAPL sequence. (c)-(h) show RX variants applied to these two images using the 3 × 45 template with 3 × 3 target and 3 × 15 guard windows.



(a) RX

(b) RX

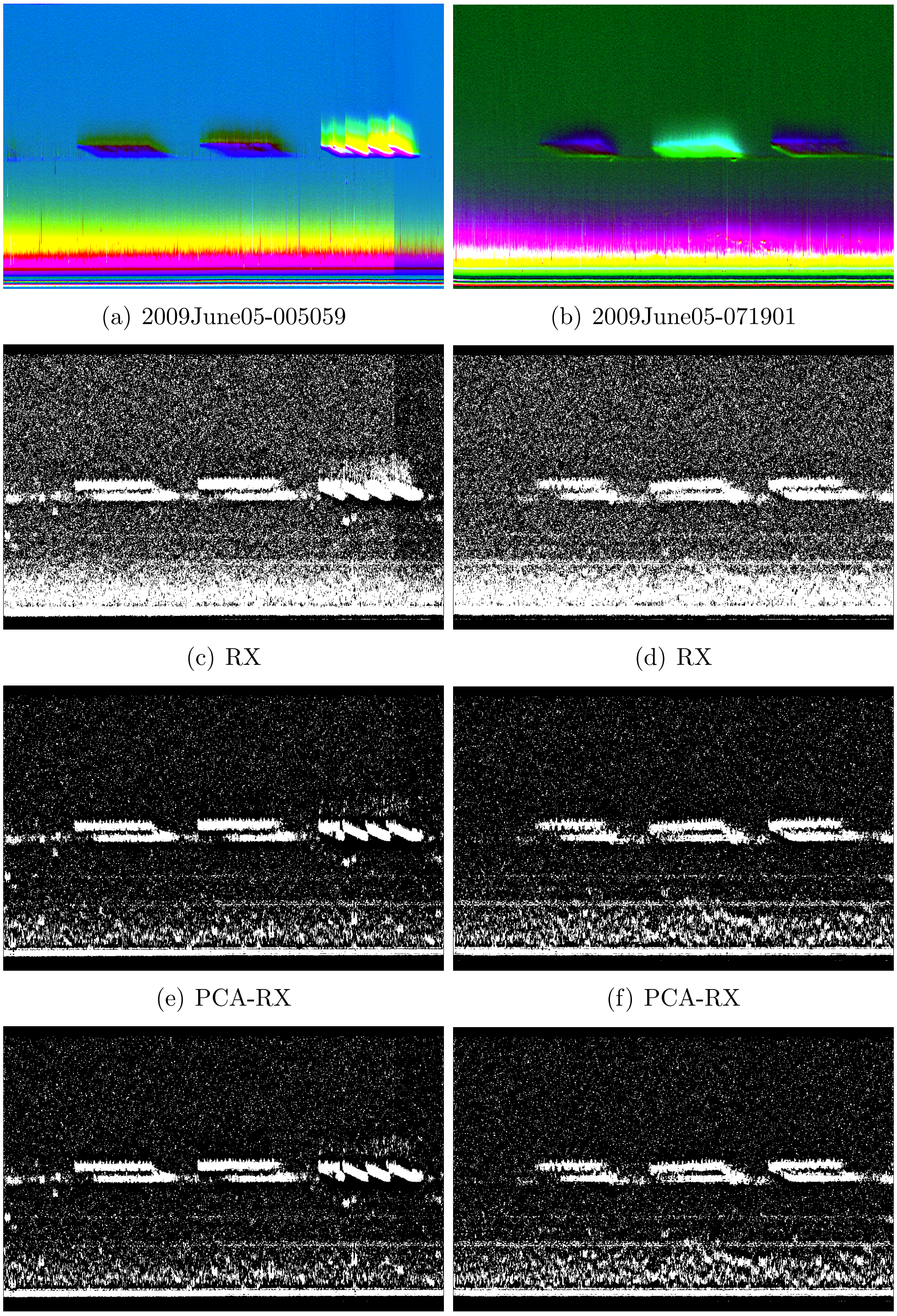
(c) PCA-RX

(d) PCA-RX

(e) NAPCA-RX

(f) NAPCA-RX

Figure 4.13: RX variants applied to two images from a JHAPL sequence, using the 25 × 25 template with 5 × 5 target and 15 × 15 guard windows.



(a) 2009June05-005059

(b) 2009June05-071901

(c) RX

(d) RX

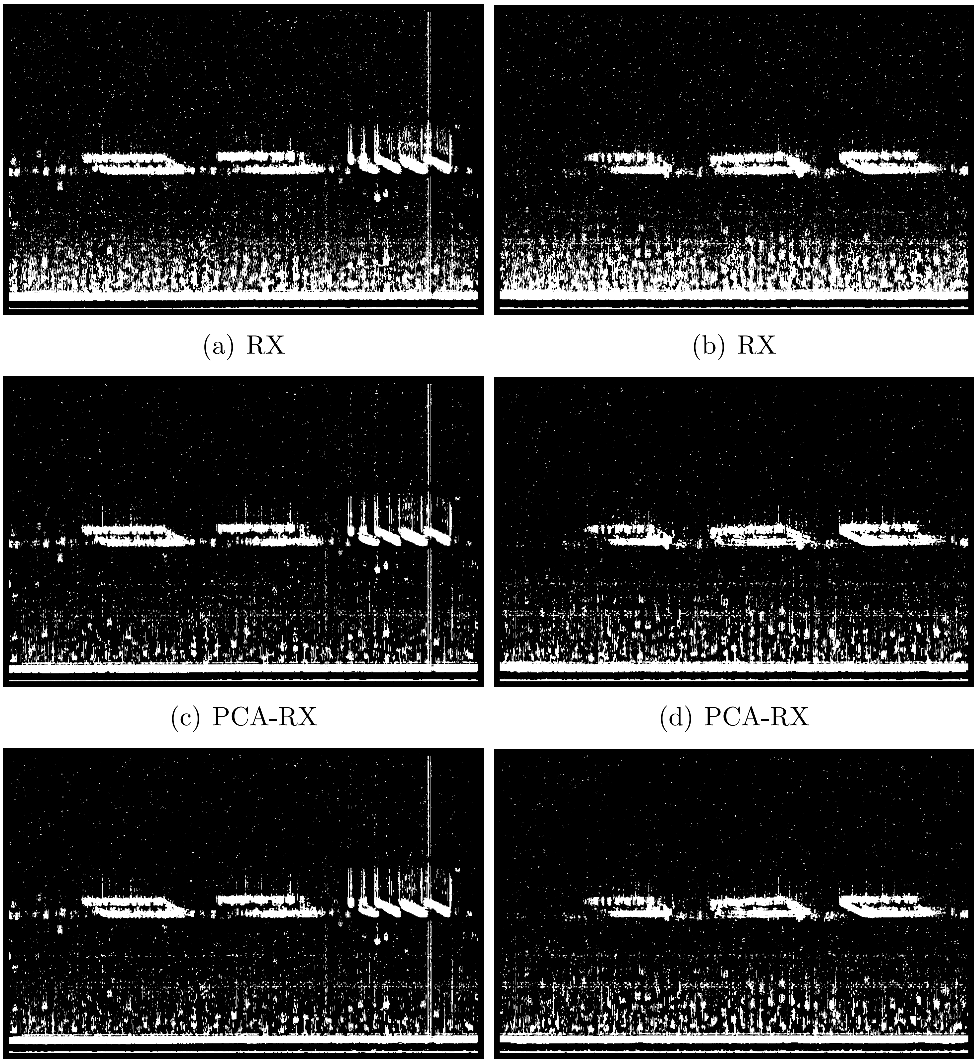
(e) PCA-RX

(f) PCA-RX

(g) NAPCA-RX

(h) NAPCA-RX

Figure 4.14: (a) and (b) show the first three noise-adjusted principal components of two FAL data cubes. (c)-(h) show results from RX variants applied to these data cubes using the 3×45 template with 3×3 target and 3×15 guard windows. The horizontal axis is time and the vertical axis is distance.



(a) RX

(b) RX

(c) PCA-RX

(d) PCA-RX

(e) NAPCA-RX

(f) NAPCA-RX

Figure 4.15: RX variants applied to two FAL data cubes, using the 25×25 template with 5×5 target and 15×15 guard windows. The horizontal axis is time and the vertical axis is distance.

CHAPTER 5

CONCLUSION

5.1 Summary

In this thesis, we have presented the theoretical foundation of the RX algorithm, a standard hyperspectral anomaly detection algorithm, as well as PCA-RX and NAPCA-RX, two extensions from the literature. Four data sets currently interest were introduced, three in the standoff detection arena. This was followed by a description of some implementation details. We then presented our results applying the three algorithms to a standard aerial data set to demonstrate the correctness of our implementation. We also applied the algorithms to a synthetic data set to study the effect of template design and mean window size on detection rate and false alarm rate for varying target sizes. Finally, we provided results of these three algorithms applied to three terrestrial data sets to test the method's ability to identify the release of an airborne substance. The algorithms were able to detect chemical and biological plumes in some sequences.

This work makes four contributions to the literature. We have presented

- a description of our realtime, parallel implementation of the RX, PCA-RX, and NAPCA-RX algorithms;
- an empirical study of the effect of local mean window size, spatial template design, and target size using synthetic data;
- the application of the RX, PCA-RX, and NAPCA-RX algorithms to terrestrial hyperspectral video of chemical and biological plume releases collected with both passive and active sensors;

- and our use of a novel rectangular spatial template to improve false alarm and detection rates in terrestrial imagery.

5.2 Future Work

The terrestrial data used in this thesis does not include a ground truth label image. As such, the interpretation of our results are somewhat subjective. Moving forward, we should make use of classifiers in the literature and the known spectra which accompany some of the data sets to produce a target label image. This will allow us to statistically compare the performance of the algorithms.

There are also some additional extensions to the RX algorithm that warrant testing on standoff detection data sets. Most notably, the wavelet-RX techniques, which may reduce our dependency on selecting the correct template for the spatial size. Further experimentation with window shape and size also appears to be warranted. In addition, we intend to take advantage of the temporal nature of the FPISDS and JHAPL data by computing the clutter covariance based upon earlier "calibration" frames, rather than using neighboring pixels. The data sets sometimes have known bad pixels, either due to dust on the lens or other damage. Extending RX to take into an account an data validity mask should help combat some of the artifacts seen in the initial tests. All of these goals are all still in the context of simply identifying substance releases without regard to the actual nature of the substance.

In addition to these, it will also be interesting to incorporate data representing known substances into the clutter data. This could be done either by including examples selected from the data set or by creating synthetic data by creating linear combinations of the clutter data already being used with calibration spectra or spectra that were unmixed from background clutter. This last method would allow examples identified by a human operator as nominal to be ignored in future classifications.

There is still the question of how to construct a well-balanced set of clutter data of this nature. The percentage of data representing each class must be determined to construct an appropriate model. There is also the question of the extent to which the spectral signature must be unmixed from the background clutter before making use of it in other parts of the image plane.

We are now armed with the family of RX-based algorithms, a speedy implementation that permits for future experimentation with a number of other extensions, and a roadmap that should carry us through to a demonstration on the more difficult plume identification problem.

BIBLIOGRAPHY

[1] *Description of the Frequency Agile Lidar(FAL) Bio Standoff Detection Data Obtained Using the Joint Ambient Breeze Tunnel (JABT)*, Tech. report.

[2] *Moffett Field AVIRIS Data*, Retrieved 13 March 2012, from `ftp://popo.jpl.nasa.gov/pub/free\_data/f080611t01p00r07rdn\_c.tar.gz`.

[3] *Fabry-Pérot Interferometer Sensor Data Set Algorithm Development Data Set*, Tech. report, 2009.

[4] *Laser Based Stand-Off Detection of Biological Agents*, Tech. report, NATO, Neuilly-sur-Seine Cedex, France, 2010.

[5] N Acito, G Corsini, and M Diani, *Adaptive detection algorithm for full pixel targets in hyperspectral images*, IEEE Proceedings - Vision, Image, and Signal Processing **152** (2005), no. 6, 731.

[6] N. Acito, G. Corsini, and M. Diani, *Computational load reduction for anomaly detection in hyperspectral images: An experimental comparative analysis*, 2007 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2007, pp. 3206–3209.

[7] Mohsen Zare Baghbidi, Kamal Jamshidi, Ahmad Reza Naghsh, and Saeid Homayouni, *Improvement of anomaly detection algorithms in hyperspectral images using discrete wavelet transform*, Signal & Image Processing : An International Journal **2** (2011), no. 4, 13–25.

[8] James R Bunch and John E Hopcroft, *Triangular factorization and inversion by fast matrix multiplication*, Mathematics of Computation **28** (1974), no. 125, 231–236.

[9] Jiah Chen and Irving Reed, *A detection algorithm for optical targets in clutter*, IEEE Transactions on Aerospace and Electronic Systems **AES-23** (1987), no. 1, 46–59.

[10] James E. Fowler, Qian Du, Wei Zhu, and Nicolas H. Younan, *Classification performance of random-projection-based dimensionality reduction of hyperspectral imagery*, 2009 IEEE International Geoscience and Remote Sensing Symposium, IEEE, 2009, pp. V–76–V–79.

[11] A.A. Green, Mark Berman, Paul Switzer, and M.D. Craig, *A transformation for ordering multispectral data in terms of image quality with implications for noise removal*, IEEE Transactions on Geoscience and Remote Sensing **26** (1988), no. 1, 65–74.

[12] Yanfeng Gu, Ying Liu, and Ye Zhang, *A selective kernel PCA algorithm for anomaly detection in hyperspectral imagery*, 2006 IEEE International Conference on Acoustics Speed and Signal Processing Proceedings **2** (2006), no. 1, II–725–II–728.

[13] Eitan Hirsch and Eyal Agassi, *Detection of gaseous plumes in IR hyperspectral images using hierarchical clustering*, Applied Optics **46** (2007), no. 25, 6368—-6374.

[14] Rebecca J. Hopkins, Stephen J. Barrington, Michael J. Castle, Karen L. Baxter, Nicola V. Felton, Joseph Jones, Clare Griffiths, Virginia Foot, and Kit Risbey, *UV-LIF lidar for standoff BW aerosol detection*, vol. 7484, SPIE, September 2009 (en).

[15] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, New York, USA, 1985.

[16] B. R. Hunt and T. M. Cannon, *Nonstationary assumptions for gaussian models of images*, IEEE Transactions on Systems, Man, and Cybernetics **6** (1976), no. 12, 876–882.

[17] H. Kwon and N.M. Nasrabadi, *Kernel RX-algorithm: a nonlinear anomaly detector for hyperspectral imagery*, IEEE Transactions on Geoscience and Remote Sensing **43** (2005), no. 2, 388–397.

[18] Zhenlin Liu, Yanfeng Gu, and Ye Zhang, *Comparative analysis of feature extraction algorithms with different rules for hyperspectral anomaly detection*, 2010 First International Conference on Pervasive Computing, Signal Processing and Applications, no. 1, IEEE, September 2010, pp. 293–296.

[19] Li Ma, Melba M Crawford, and Jinwen Tian, *Anomaly detection for hyperspectral images using local tangent space alignment*, 2010 IEEE International Geoscience and Remote Sensing Symposium, IEEE, July 2010, pp. 824–827.

[20] Avigdor Margalit, *Adaptive detection of stationary optical and IR targets using correlated scenes*, Ph.d. dissertation, University of Southern California, 1984.

[21] Stefania Matteoli, Marco Diani, and Giovanni Corsini, *A tutorial overview of anomaly detection in hyperspectral images*, IEEE Aerospace and Electronic Systems Magazine **25** (2010), no. 7, 5–28.

[22] Asif Mehmood and Nasser M Nasrabadi, *Kernel wavelet-Reed-Xiaoli: an anomaly detection for forward-looking infrared imagery*, Applied Optics **50** (2011), no. 17, 2744.

[23] Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA, 2000.

[24] Kenneth S. Miller, *Multidimensional Gaussian Distributions*, John Wiley and Sons, Inc, New York, New York, USA, 1964.

[25] B. Moore, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Transactions on Automatic Control **26** (1981), no. 1, 17–32.

[26] Irving S. Reed and Xiaoli Yu, *Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution*, IEEE Transactions on Acoustics, Speech, and Signal Processing **38** (1990), no. 10, 1760–1770.

[27] R.E. Roger, *A faster way to compute the noise-adjusted principal components transform matrix*, IEEE Transactions on Geoscience and Remote Sensing **32** (1994), no. 6, 1194–1196.

[28] G. W. Stewart, *On the early history of singular value decomposition*, SIAM Review **35** (1993), no. 4, 551–566.

[29] Volker Strassen, *Gaussian elimination is not optimal*, Numerische Mathematik **356** (1969), no. 4, 354–356.

[30] Yuri P. Taitano, Brian A. Geier, and Kenneth W. Bauer, *A locally adaptable iterative RX detector*, EURASIP Journal on Advances in Signal Processing **2010** (2010), no. 1, 341908.

[31] Jean-Marc Thériault, Eldon Puckrin, and James O Jensen, *Passive standoff detection of Bacillus subtilis aerosol by Fourier-transform infrared radiometry*, Applied Optics **42** (2003), no. 33, 6696–6703.

[32] Lloyd N. Trefethen and III Bau, David, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, USA, 1997.

[33] Yu-lei Wang, Chun-Hui Zhao, and Ying Wang, *Anomaly detection using subspace band section based RX algorithm*, 2011 International Conference on Multimedia Technology, IEEE, July 2011, pp. 3436–3439.

[34] Xiaoli Yu, Irving S. Reed, and Alan D. Stocker, *Comparative performance analysis of adaptive multispectral detectors*, IEEE Transactions on Signal Processing **41** (1993), no. 8, 2639–2656.