UCRL-TR-209277

LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Probability Density and CFAR Threshold Estimation for Hyperspectral Imaging

G. A. Clark

January 28, 2005

Disclaimer

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Probability Density and CFAR Threshold Estimation for Hyperspectral Imaging

Grace A. Clark, Ph.D.
Lawrence Livermore National Laboratory
7000 East Ave., L-130, Livermore, CA 94550
(925) 423-9759 (Office), (925) 422-2495 (FAX),
clark9@llnl.gov

September 21, 2004

UCRL-TR-XXXXXX

Acknowledgments

The author gratefully acknowledges the important contributions to this work made by her colleagues. Dr. Randy S. Roberts (EE) and William D. Aimonetti (Comp) motivated and supervised this work as part of the Large Area Search Initiative (LASI) project, led by Robert Priest (NAI). Randy also provided the image for figure 2.1. The sponsors are DOE/NA20 and some additional work for others (WFO) sponsors. The project is part of the Proliferation Detection Systems (PDS) Program led by Bill Conaway, Q-Division, and the Remote Sensing Spectroscopy Program led by Bob Priest, Q-Division.

Abstract

The work reported here shows the proof of principle (using a small data set) for a suite of algorithms designed to estimate the probability density function of hyperspectral background data and compute the appropriate Constant False Alarm Rate (CFAR) matched filter decision threshold for a chemical plume detector. Future work will provide a thorough demonstration of the algorithms and their performance with a large data set.

The LASI (Large Aperture Search Initiative) Project involves instrumentation and image processing for hyperspectral images of chemical plumes in the atmosphere. The work reported here involves research and development on algorithms for reducing the false alarm rate in chemical plume detection and identification algorithms operating on hyperspectral image cubes. The chemical plume detection algorithms to date have used matched filters designed using generalized maximum likelihood ratio hypothesis testing algorithms [1, 2, 5, 6, 7, 12, 10, 11, 13].

One of the key challenges in hyperspectral imaging research is the high false alarm rate that often results from the plume detector [1, 2]. The overall goal of this work is to extend the classical matched filter detector to apply Constant False Alarm Rate (CFAR) methods to reduce the false alarm rate, or Probability of False Alarm P_{FA} of the matched filter [4, 8, 9, 12]. A detector designer is interested in minimizing the probability of false alarm while simultaneously maximizing the probability of detection P_D. This is summarized by the Receiver Operating Characteristic Curve (ROC) [10, 11], which is actually a family of curves depicting P_D vs. P_{FA} parameterized by varying levels of signal to noise (or clutter) ratio (SNR or SCR). Often, it is advantageous to be able to specify a desired P_{FA} and develop a ROC curve (P_D vs. decision threshold r_0) for that case. That is the purpose of this work.

Specifically, this work develops a set of algorithms and MATLAB implementations to compute the decision threshold r_0^* that will provide the appropriate desired Probability of False Alarm P_{FA} for the matched filter. The goal is to use prior knowledge of the background data to generate an estimate of the probability density function (pdf) [13] of the matched filter threshold r for the case in which the data measurement contains only background data (we call this case the null hypothesis, or H_0 [10, 11]. We call the pdf estimate $\widehat{f}(r|H_0)$). In this report, we use histograms and Parzen pdf estimators [14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. Once the estimate is obtained, it can be integrated to compute an estimate of the P_{FA} as a function of the matched filter detection threshold r. We can then interpolate $r\ vs.\ P_{FA}$ to obtain a curve that gives the threshold r_0^* that will provide the appropriate desired Probability of False Alarm P_{FA} for the matched filter. Processing results have been computed using both simulated and real LASI data sets. The algorithms and codes have been validated, and the results using LASI data are presented here.

Future work includes applying the pdf estimation and CFAR threshold calculation algorithms to the LASI matched filter based upon global background statistics, and developing a new adaptive matched filter algorithm based upon local background statisitcs. Another goal is to implement the 4-Gamma pdf modeling method proposed by Stocker et. al. [4] and comparing results using histograms and the Parzen pdf estimators.

Contents

List of Figures

Chapter 1

Introduction

The work reported here shows the proof of principle (using a small data set) for a suite of algorithms designed to estimate the probability density function of hyperspectral background data and compute the appropriate Constant False Alarm Rate (CFAR) matched filter decision threshold for a chemical plume detector. Future work will analyze performance using a large data set.

The LASI (Large Aperture Search Inititiative) Project involves instrumentation and image processing for hyperspectral images of chemical plumes in the atmosphere. The work reported here involves research and development on algorithms for reducing the false alarm rate in chemical plume detection and identification algorithms operating on hyperspectral image cubes. The chemical plume detection algorithms to date have used matched filters designed using generalized maximum likelihood ratio hypothesis testing algorithms [1, 2, 5, 6, 7, 12, 10, 11, 13].

One of the key challenges in hyperspectral imaging research is the high false alarm rate that often results from the plume detector [1, 2]. The overall goal of this work is to extend the classical matched filter detector to apply Constant False Alarm Rate (CFAR) methods to reduce the false alarm rate, or Probability of False Alarm P_{FA} of the matched filter [4, 8, 9, 12]. A detector designer is interested in minimizing the probability of false alarm while simultaneously maximizing the probability of detection P_D. This is summarized by the Receiver Operating Characteristic Curve (ROC) [10, 11], which is actually a family of curves depicting P_D vs. P_{FA} parameterized by varying levels of signal to noise (or clutter) ratio (SNR or SCR). Often, it is advantageous to be able to specify a desired P_{FA} and develop a ROC curve (P_D vs. decision threshold r_0) for that case. That is the purpose of this work.

Specifically, this work develops a set of algorithms and MATLAB implementations to compute the decision threshold r_0^* that will provide the appropriate desired Probability of False Alarm P_{FA} for the matched filter. The goal is to use prior knowledge of the background data to generate an estimate of the probability density function (pdf) [13] of the matched filter threshold r for the case in which the data measurement contains only background data (we call this case the null hypothesis, or H_0 [10, 11]. We call the pdf estimate $\widehat{f}(r|H_0)$. In this report, we use histograms and Parzen pdf estimators [14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. Once the estimate is obtained, it can be integrated to compute an estimate of the P_{FA} as a function of the matched filter detection threshold r. We can then interpolate $r\ vs.\ P_{FA}$ to obtain a curve that gives the threshold r_0^* that will provide the appropriate desired Probability of False Alarm P_{FA} for the matched filter. Processing results have been computed using both simulated and real LASI data sets. The algorithms and codes have been validated, and the results using LASI data are presented here.

This report is organized as follows. In Chapter 2, we give a brief summary of the theoretical background and the technical approach. Chapter 3 describes the probability density function estimation problem and some algorithms for solving it. Chapter 4 summarizes the MATLAB code for implementing the algorithms, and Chapter 5 describes a processing example using real LASI data. In Chapter 6 we discuss conclusions and future work.

Chapter 2

Theoretical Background and Approach

2.1 Detection Theory and Matched Filtering

This report will not provide a detailed summary of detection theory and matched filtering. Rather, we refer the reader to the bibliography. We do, however, summarize the discussion as follows.

Basic detection theory [10, 11] relies on the concepts of hypothesis testing, probability density functions and receiver operating characteristic (ROC) curves, as summarized in Figure (2.1). The detector used by the LASI project is a type of matched filter designed using generalized maximum likelihood ratio hypothesis testing algorithms [1, 2, 5, 6, 7, 12, 10, 11, 13].

Figure (2.2) summarizes the concept of a “Confusion Matrix ” or “Contingency Table ” , which is a useful way of defining classification results. The table shown is given for the special case in which the number of classes is two and the prior probabilites are equal.

Figure (2.3) by R. S. Roberts summarizes the matched filter equation used for detection [1, 2].

2.2 The CFAR Matched Filter Algorithm

One of the key challenges in hyperspectral imaging research is the high false alarm rate that often results from the plume detector [1, 2]. The overall goal of this work is to extend the classical matched filter detector to apply Constant False Alarm Rate (CFAR) methods to reduce the false alarm rate, or Probability of False Alarm P_{FA} of the matched filter [4, 8, 9, 12]. Figure (2.4) summarizes the concept behind CFAR methods. The CFAR technique is to specify a desired P_{FA} based upon some knowledge of the background statistics, and develop a ROC curve (P_D vs. decision threshold r_0) for that case. Given a set of known matched filter image exemplars r corresponding to the background part of a hyperspectral image cube, we can (1) Estimate the pdf $f(r|H_0)$ of the background and (2) Integrate $f(r|H_0)$ as depicted in the figure, letting the decision threshold r_0 vary over an appropriate range of values of r. This gives us an estimate of P_{FA} vs. r_0. We then wish to specify a desired P_{FA} and perform a table lookup to find the appropriate threshold r_0^* that will give us the desired P_{FA}.

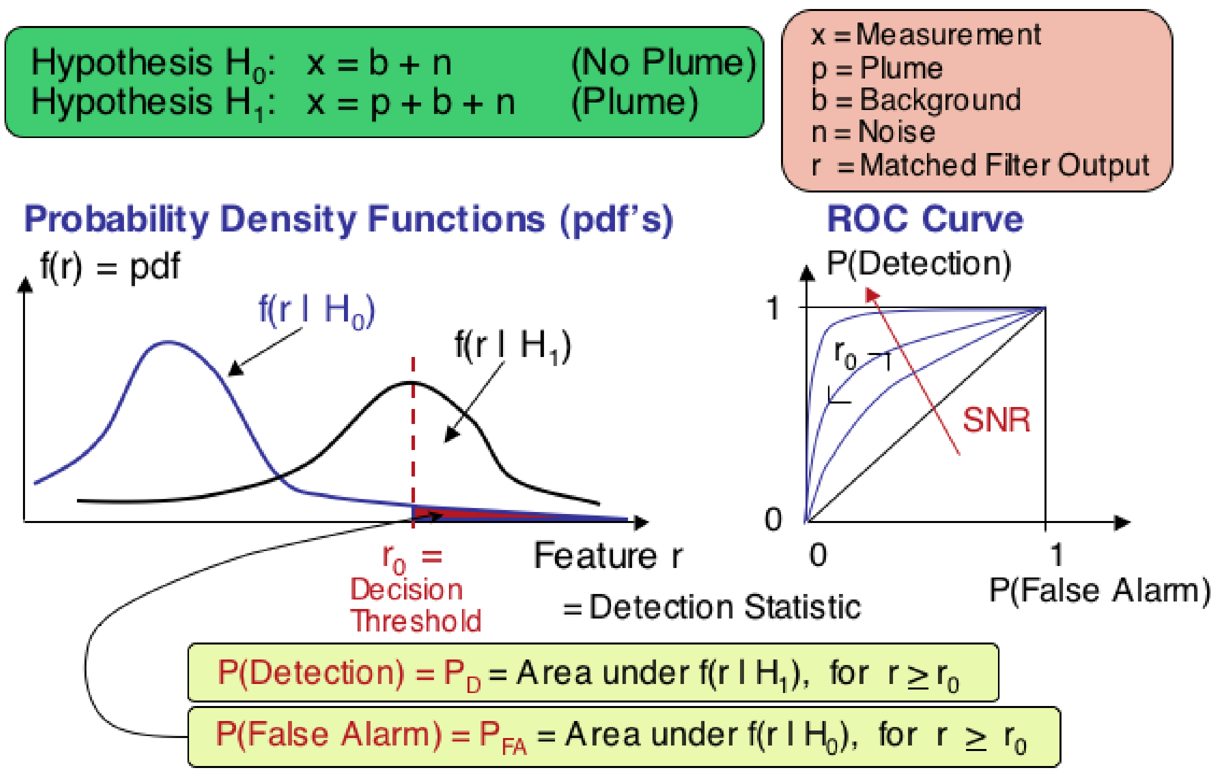


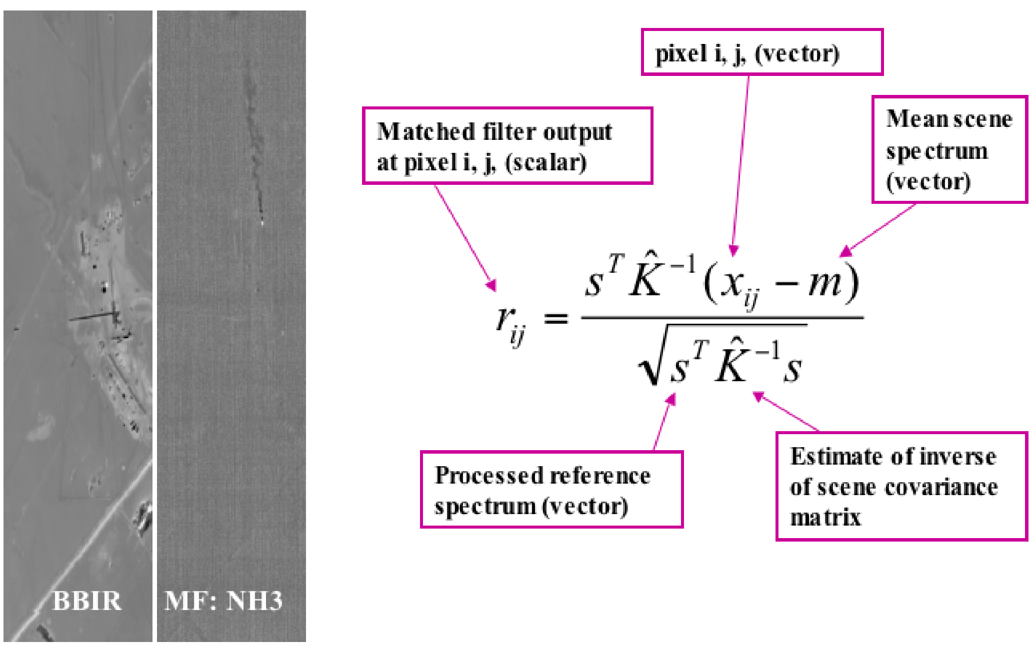
Figure 2.1: The elements of detection hypothesis testing are summarized here for the LASI application. The detector must make a decision as to whether or not a plume is present at each pixel in an image. Hypothesis H_0 is the case in which the measurement x corresponds to a part of the image in which there is no plume. H_1 represents the case in which a plume is present. Given estimates of the probability density functions, the probability of detection and probability of false alarm can be computed as a function of the decision threshold and the signal-to-noise ratio. The receiver operating characteristic curve (ROC) summarizes the detector performance.

Decision \ Actual	Event (E)	Background (BG)
Event (E)	P(E | E) = P(Detection)	P(E | BG) = P(False Alarm)
Background (BG)	P(BG | E) = P(Miss)	P(BG | BG) = P(Specificity)

Note: P(E | E) + P(BG | E) = 1 and P(E | BG) + P(BG | BG) = 1

P(Correct Classification) = .5 [P(E | E) + P(BG | BG)]

Figure 2.2: The Confusion Matrix or Contingency Table is another method for specifying detector performance based upon controlled experiments in which the correct performance is known. A ROC can be constructed from the table, without requiring knowledge of the probability density functions. In this figure, the symbol E denotes an event (a plume for the LASI application), and BG denotes background. Note that the equation for the probability of correct classification assumes the case in which the prior probabilities are equal.



$$r_{ij} = \frac{s^T \hat{K}^{-1}(x_{ij} - m)}{\sqrt{s^T \hat{K}^{-1} s}}$$

Figure 2.3: This figure, provided by R. S. Roberts, explains the components of the matched filter equation. The image on the left is an example of the measured data. The image on the right is the corresponding matched filter output image, showing the pixels detected as plume pixels.

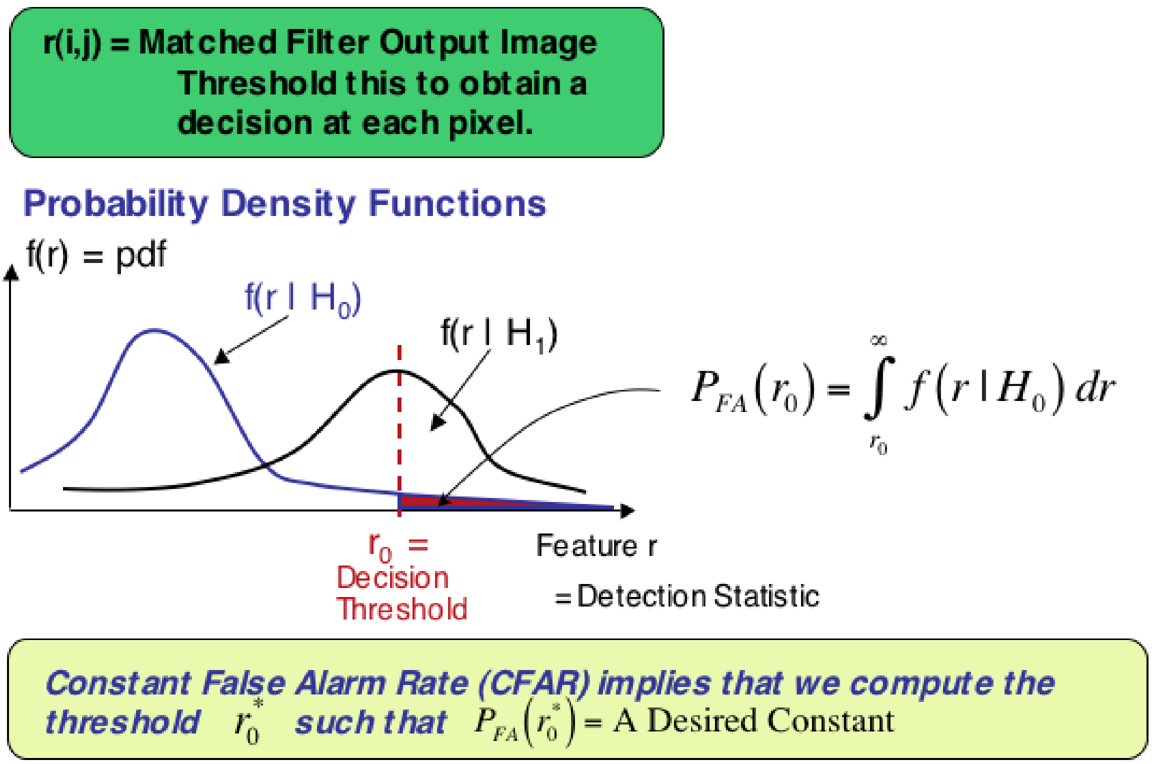


Figure 2.4: For a CFAR detector, the goal is to specify a desired P_{FA} for the detector and then compute a ROC curve (P_D vs. decision threshold r_0) for that case. In this way, we can attempt to minimize the detector's false alarm rate.

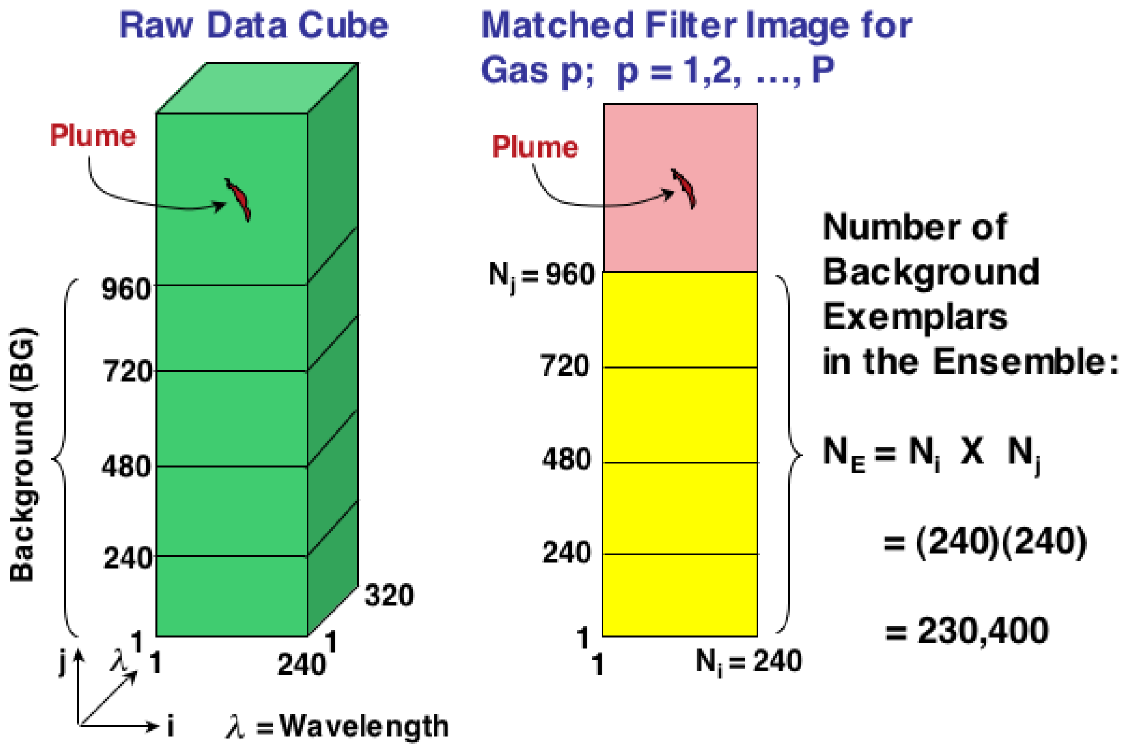


Figure 2.5: For calculating a CFAR threshold, we need a data set consisting of exemplars of the matched filter image corresponding to background-only (no plume). A reasonable amount of data to use is approximately 1000 lines of hyperspectral data. The drawing on the left shows that we can use a background region of a data cube consisting of four blocks of data (960 lines), where we pick a region of the image in which there exists background only (prior knowledge). The image on the right shows the corresponding matched filter image. We propose to use an ensemble of 230,400 pixels (exemplars) from the matched filter output image for comparison with the global statistics method normally used.

2.3 Calculating a CFAR Detection Threshold

Figure (2.5 describes the data set requirements for computing the CFAR threshold. The algorithm for computing the CFAR matched filter threshold is depicted in Figure (2.6).

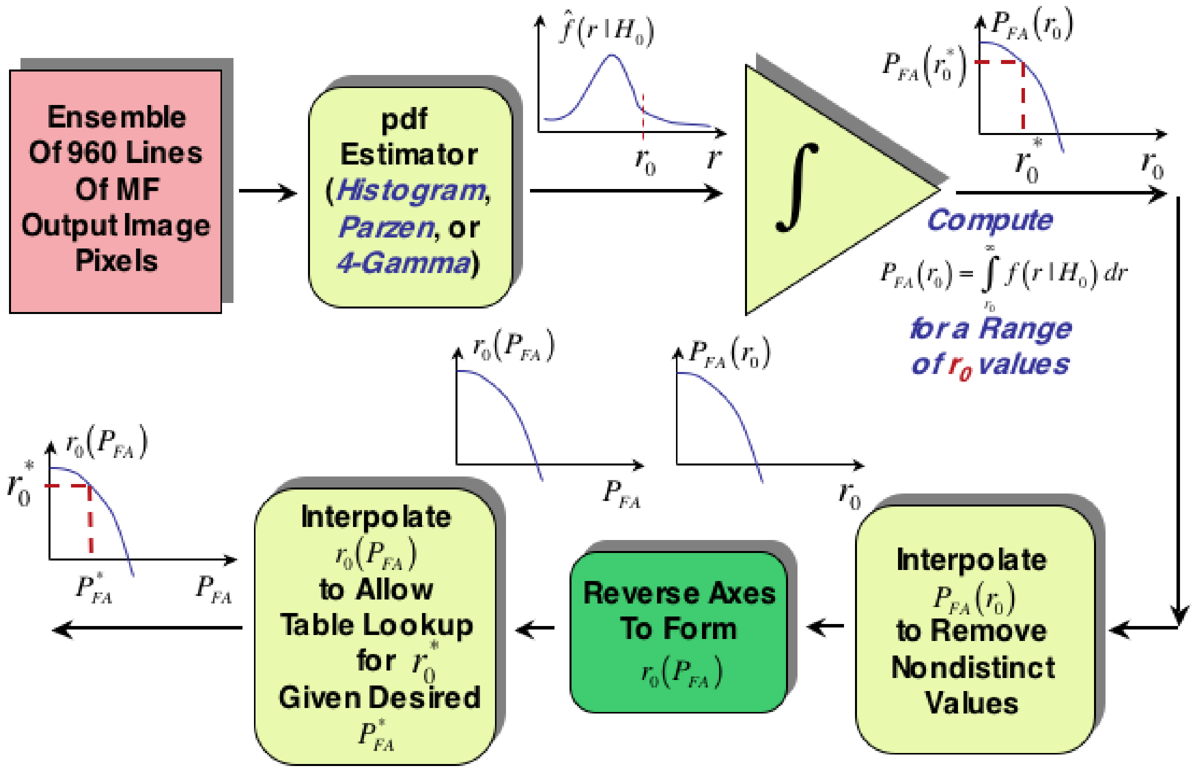


Figure 2.6: The algorithm for computing the matched filter decision threshold includes comparing three pdf estimation techniques.

Chapter 3

Density Estimation

3.1 The Histogram

The histogram of a random variable r is a very simple estimate of a probability density function $f(r)$, based upon measured samples (exemplars) from an experiment. The range of the random variable r is divided (quantized) into bins. The estimate $\widehat{f}(r)$ of the pdf is constructed by counting the number of occurrences of the random variable that lie within each bin. A histogram plot of $\widehat{f}(r)$ is constructed in which the abscissa is the bin number and the ordinate is the number of occurrences of the random variable in each bin. The reader is referred to [14, 29, 30] for a thorough discussion of histograms. For this project, we used the histogram as one of the estimates of interest for $f(r)$.

A key issue in the use of histograms is how to choose the number of bins to use. A practical rule of thumb has been developed by Sturgess [28] using a combination of theory and empirical studies. If we let N_E equal the number of exemplars in the data set and N_r equal the number of bins to use for random variable r, then the rule of thumb is

$$N_r = log_2[N_E] + 1 \tag{3.1}$$

This rule of thumb has been found to be useful by the author. However, other considerations such as ease of visualizing the histogram by visual inspection may be the overriding consideration in choosing the bin size.

3.2 Parzen Kernel pdf Estimation

We also used a more sophisticated pdf estimator of the kernel type [14]. We chose the Parzen kernel type estimator because of its generality, ease of use, and robustness as demonstrated by a wide variety of application observed by the author [14, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]. The Parzen kernel pdf estimator is the basis for the Probabilistic Neural Network (PNN) proposed by Donald F. Specht [15, 17, 18]. The PNN is actually a Bayes optimal classifier which uses pdf estimates based upon the Parzen kernel. The most commonly- used kernel is a Gaussian-shaped kernel.

We assume that we are given a data set (or training set) consisting of exemplars from the decision class of interest. For the LASI project, in which we wish to estimate the P_{FA}, this means we have a set of data samples known to have been collected from the background plus noise portion of a matched filter image resulting from processing a hyperspectral data cube.

We chose a set of samples from the data set indicated in Figure (3.1). The data came from a cube called “rr

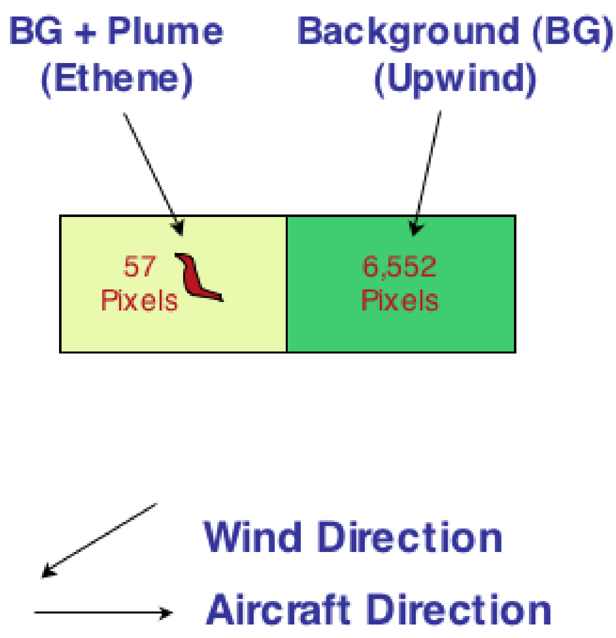


Figure 3.1: Here, we depict the LASI data set actually used to demonstrate the CFAR threshold estimation algorithm. Pixels from the data file "rr mfcube " were divided manually into two subsets, one consisting of "Background (BG) " pixels (actually background plus noise), and the other consisting of "Background plus Plume " pixels (actually background plus noise plus plume).

mfcube " with header file "rr mfcube.hdr " created by R. S. Roberts and G. A. Clark. The data were collected in a controlled experiment in which ethene was released into the atmosphere. We defined the background part of the scene to be upwind from the ethene plume. Each pixel value in the background image was mapped to a single element x_i in a feature vector $\underline{X}$ where $i = 1, 2, \ldots, m$ is the feature index such that $\underline{X} = [x_1, x_2, \ldots, x_m]^T$.

The essence of the Parzen kernel pdf estimator can be summarized as follows. Let us define the following symbols:

$i =$ Training pattern or exemplar index, where $i = 1, 2, \ldots, m$

$m =$ Number of training or exemplar patterns in the training set

$\underline{X} =$ Pattern or feature vector under test, $\underline{X} = [x_1, x_2, \ldots, x_m]^T$.

$\underline{X}_i =$ i-th training vector in the training set

$\sigma =$ Smoothing parameter for the pdf estimator (to be chosen by the user)

$p =$ Dimension of the feature vector $\underline{X}$ (The dimension of $\underline{X}$ is p X 1)

Given a set of training vectors $\underline{X}_i$ from the training set, the Parzen kernel pdf estimate is given by:

$$\widehat{f}(\underline{X}) = \frac{1}{(2\pi)^{\frac{p}{2}}\sigma^p} \sum_{i=1}^{m} exp\left[-\frac{(\underline{X}-\underline{X}_i)^T(\underline{X}-\underline{X}_i)}{2\sigma^2}\right] \tag{3.2}$$

The kernel for this estimator is a Gaussian-shaped function centered at the exemplar value $\underline{X}_i$ = in multi-dimensional space. The width of the Gaussian-shaped kernel is specified by the smoothing parameter σ, which clearly is the standard deviation of the Gaussan kernel. The pdf estimate is the superposition, or summation of all of these kernels summed over the training data set consisting of m samples.

The smoothing parameter choice is very important to the quality of the pdf estimate. As $\sigma \to 0$, the pdf estimate becomes the same as the Nearest Neighbor estimate. Also, the estimated pdf has distinct modes corresponding to the locations of the training samples. As $\sigma \to \infty$ the pdf estimate experiences broad smoothing and interpolation. In addition, the estimated pdf approaches Gaussian and the PNN equals the hyperplane (linear) classifier.

3.3 Automatic Selection of the Smoothing Parameter

Smoothing parameter selection is generally an ad hoc, manual, empirical, time-consuming activity [14]. We are, therefore interested in automating the process. The MATLAB software written for this project allows the user to choose whether she/he wishes to choose the parameter manually, or automatically. We implemented an automatic scheme suggested by J. B. Cain [16]. This scheme provides a reasonable estimate of the smoothing parameter, given the training data set. If one wishes to adjust σ manually, one can use the Cain algorithm as an initial condition for starting the manual search.

The Cain algorithm was originally intended for use with the Probabilistic Neural Network (PNN) [15], which is a Bayes classifier, based upon pdf estimates from a Parzen pdf estimator. For that reason, the discussion of the Cain algorithm dwells upon the classification problem. For this report, however, we are not using the Parzen pdf estimator as part of a classifier, so the number of classes is one.

Cain's algorithm is based upon the observation that the pdf estimate at a point (a member of the training set, or set of exemplar samples), should be significantly influenced by more than one exemplar, but not by a large number of exemplars. It is clear that the larger the number of exemplars and the denser the exemplars, the smaller the smoothing parameter σ must be for best performance in estimating the pdf. In the Cain algorithm, σ is set to a constant times the average distance between exemplars in the same class.

Let i denote the exemplar index, and k denote the class index (for our case, $k = 1$). Let ρ_i denote the $i - th$ exemplar from the training set. Let C_k denote the $k - th$ class, and $|C_k|$ denote the number of exemplars in the $k - th$ class.

Let d_i denote the distance between exemplar pattern ρ_i and the nearest exemplar in the class C_k. We can then define the minimum distance between exemplar patterns in class C_k as follows:

$$\widehat{d}_{avg}[k] = \frac{1}{|C_k|} \sum_{\rho_i \in C_k} d_i \tag{3.3}$$

Finally, we assign the smoothing parameter for class C_k to be:

$$\sigma_k = g \bullet \widehat{d}_{avg}[k] \tag{3.4}$$

where $1.1 \leq g \leq 1.4$. The range of g was determined empirically by Cain, and the author has also found it to be useful in a variety of applications [19, 20, 21, 22, 23, 24, 25, 26, 27].

The Cain algorithm is a two-pass algorithm and is summarized as follows:

3.3.1 First Pass

The first pass is nearly identical to the training method used for the PNN:
(1) Present all training patterns (exemplars) ρ_i to the the Parzen estimator.
(2) After all exemplars are presented, set constant the number of exemplars $|C_k|$ in each of the k classes.

3.3.2 Second Pass

Assign the smoothing parameter for class C_k to be [16]:

$$\sigma_k = g \bullet \widehat{d}_{avg}[k] \tag{3.5}$$

where $1.1 \leq g \leq 1.4$, and $\widehat{d}_{avg}[k]$ is given by Equation (3.3).

Chapter 4

MATLAB Code

The MATLAB code for implementing the algorithms is depicted in Figure (4.1).

4.1 Quadrature for Calculating the Pfa

For quadrature, a short MATLAB function was written implementing the the “extended ” or “composite ” Trapezoidal Rule algorithm, which is accurate and practical [31, 32]. Given a discrete function f_k, where k is the sample index for the abscissa, and the sampling period h, then the integration result I_k at the $k-th$ iteration can be found using the formula:

$$I_k = I_{k-1} + \frac{h}{2}[f_{k-1} + f_k] \tag{4.1}$$

This is a recursion and a “closed ”formula, in the sense that it uses the values of the function at the endpoints.

4.2 Interpolating the $P_{FA}\ vs.\ r0$ Curve and the $r0\ vs.\ P_{FA}$ Curve

Once the estimate of $f(r|H_0)$ is obtained, it can be integrated to compute an estimate of the P_{FA} as a function of the matched filter detection threshold $r0$. We can then interpolate $r0\ vs.\ P_{FA}$ to obtain a curve that gives the threshold r_0^* that will provide the appropriate desired Probability of False Alarm P_{FA} for the matched filter. Unfortunately, one additional step is required, because the $P_{FA}\ vs.\ r0$ curve is not monotonic for the data we examined. This is a problem because the interpolation algorithms require that the function to be intepolated must be monotonic. Therefore, we procede in a two- step procedure: (1) Intepolate the $P_{FA}\ vs.\ r0$ curve using a greatly reduced sampling period (about a factor of one or two orders of magnitude) to ensure that the curve is monotonic. (2) Interpolate the $r_0\ vs.\ P_{FA}$ curve.

For function interpolation, we used a MATLAB function called “interp1 ” [36]. We used the cubic spline algorithm option, with good results.

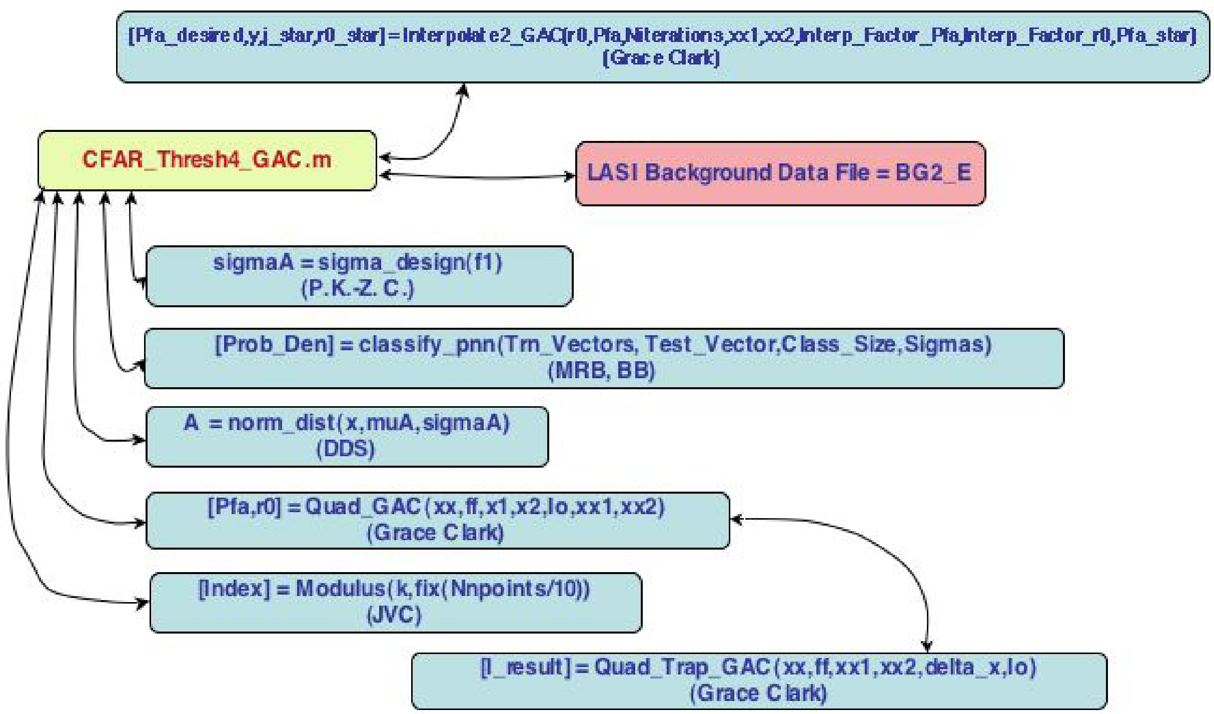


Figure 4.1: The block diagram for the MATLAB code that implements the algorithms is given in this figure. The basic code, the input data file and the various functions are given.

Chapter 5

Processing Results for a Controlled Experiment with LASI Data

The algorithms described above were tested on simulated data and real LASI data. The simulated data were used for validating the algorithm, and the validation results are not reported here. A low-order example of the algorithm performance using LASI data is presented next.

The data reported here were taken from a set of data samples known to have been collected from the background plus noise portion of a matched filter image resulting from processing a hyperspectral data cube.

We chose a set of samples from the data set indicated in Figure (3.1). The data came from a cube called "rr mfcube " with header file "rr mfcube.hdr " created by R. S. Roberts and G. A. Clark. The data were collected in a controlled experiment in which ethene was released into the atmosphere. We defined the background part of the scene to be upwind from the ethene plume. Each pixel value in the background image was mapped to a single element x_i in a feature vector $\underline{X}$ where $i = 1, 2, \ldots, m$ is the feature index such that $\underline{X} = [x_1, x_2, \ldots, x_m]^T$.

The processing results using the algorithms specified above are depicted in Figures (5.1), (5.2), (5.3), (5.4), (5.5), (5.6), (5.7), (5.8), (5.9), (5.10) (5.11), and (5.12).

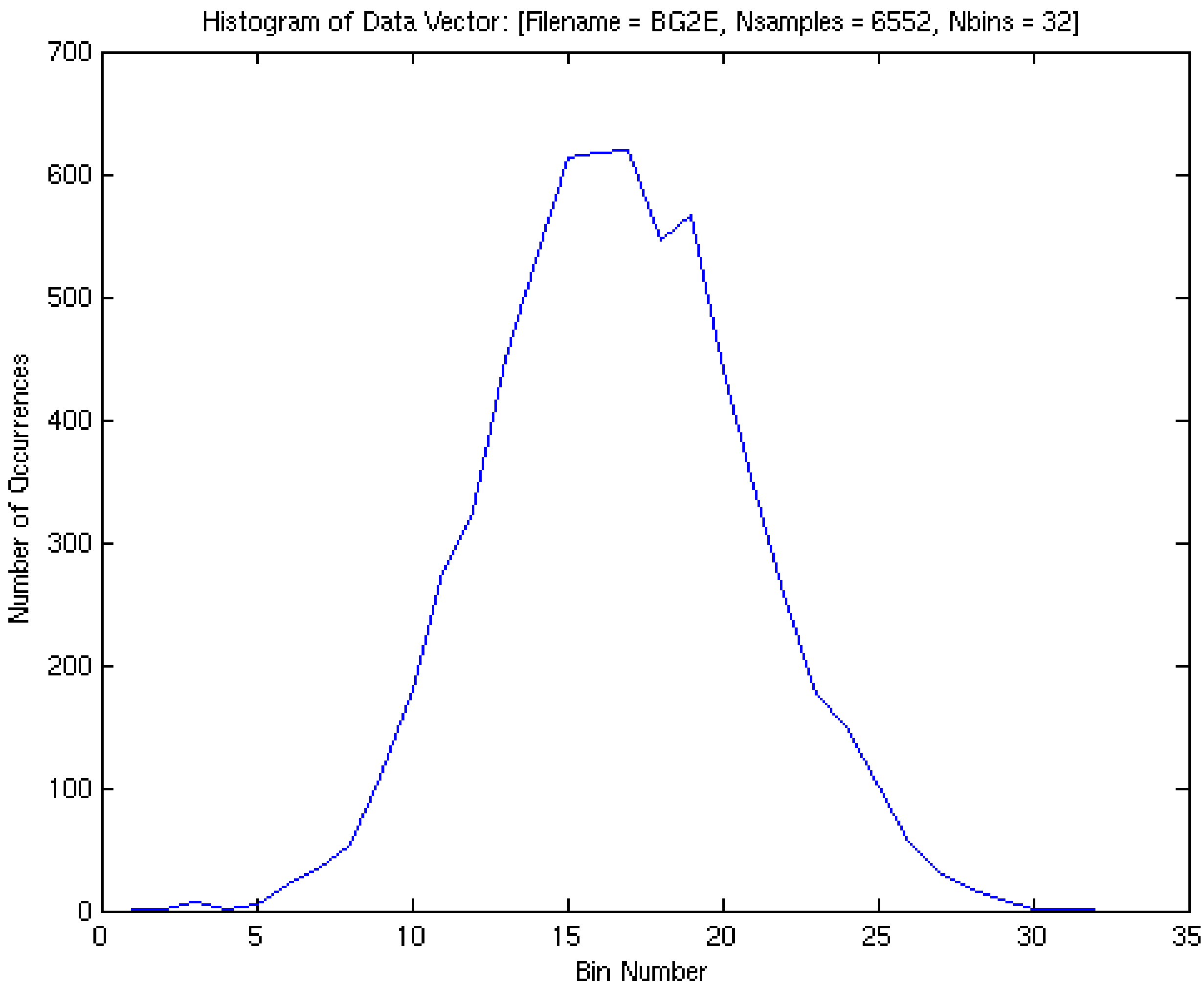


Figure 5.1: A histogram is plotted for the full $BG2E$ background data set depicted in Figure (3.1). The number of samples in the data set is 6552, and the number of bins used for the histogram is 32.

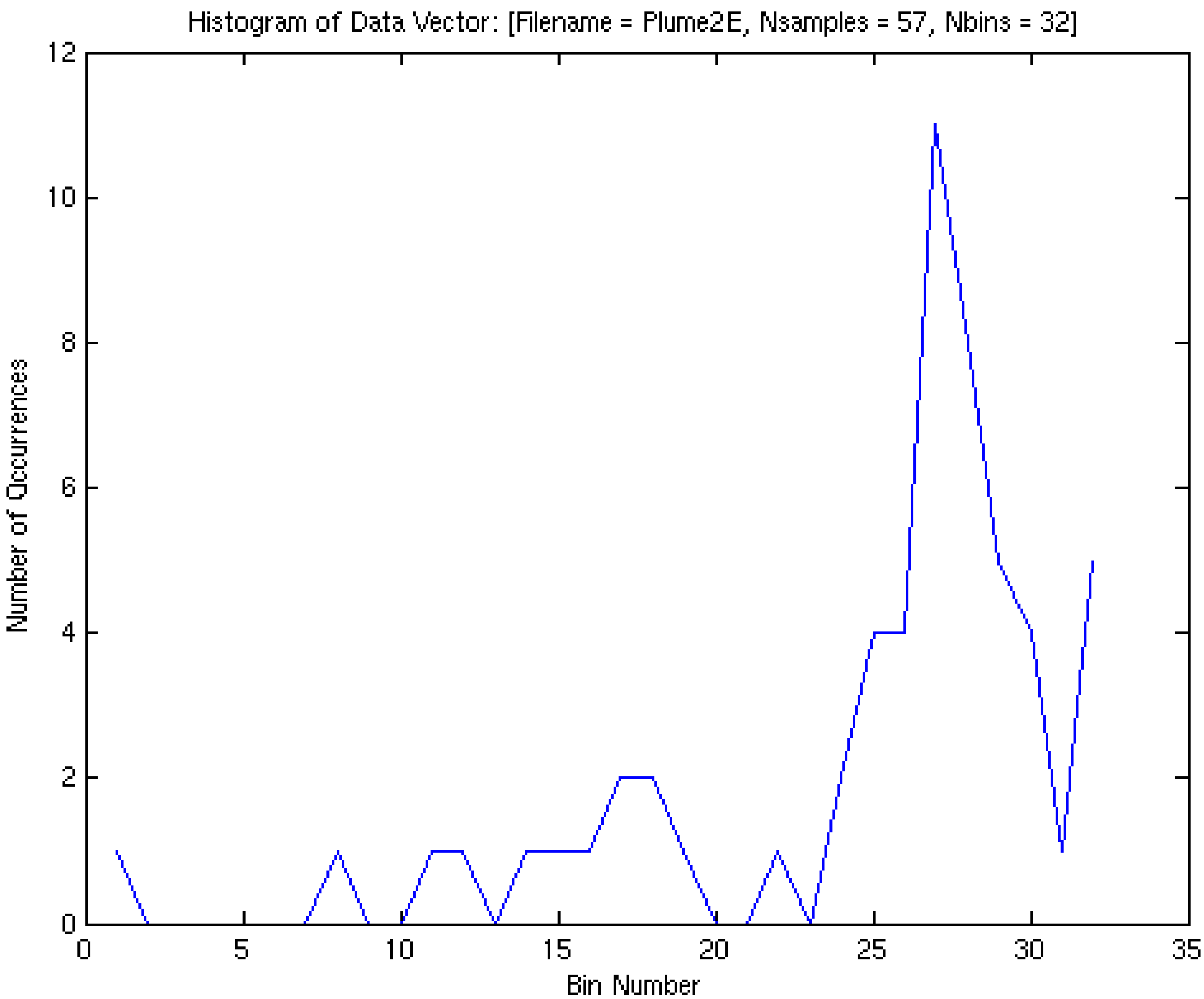


Figure 5.2: A histogram is plotted for the $Plume2E$ data set depicted in Figure (3.1). The number of samples in the data set is 57, and the number of bins used for the histogram is 32.

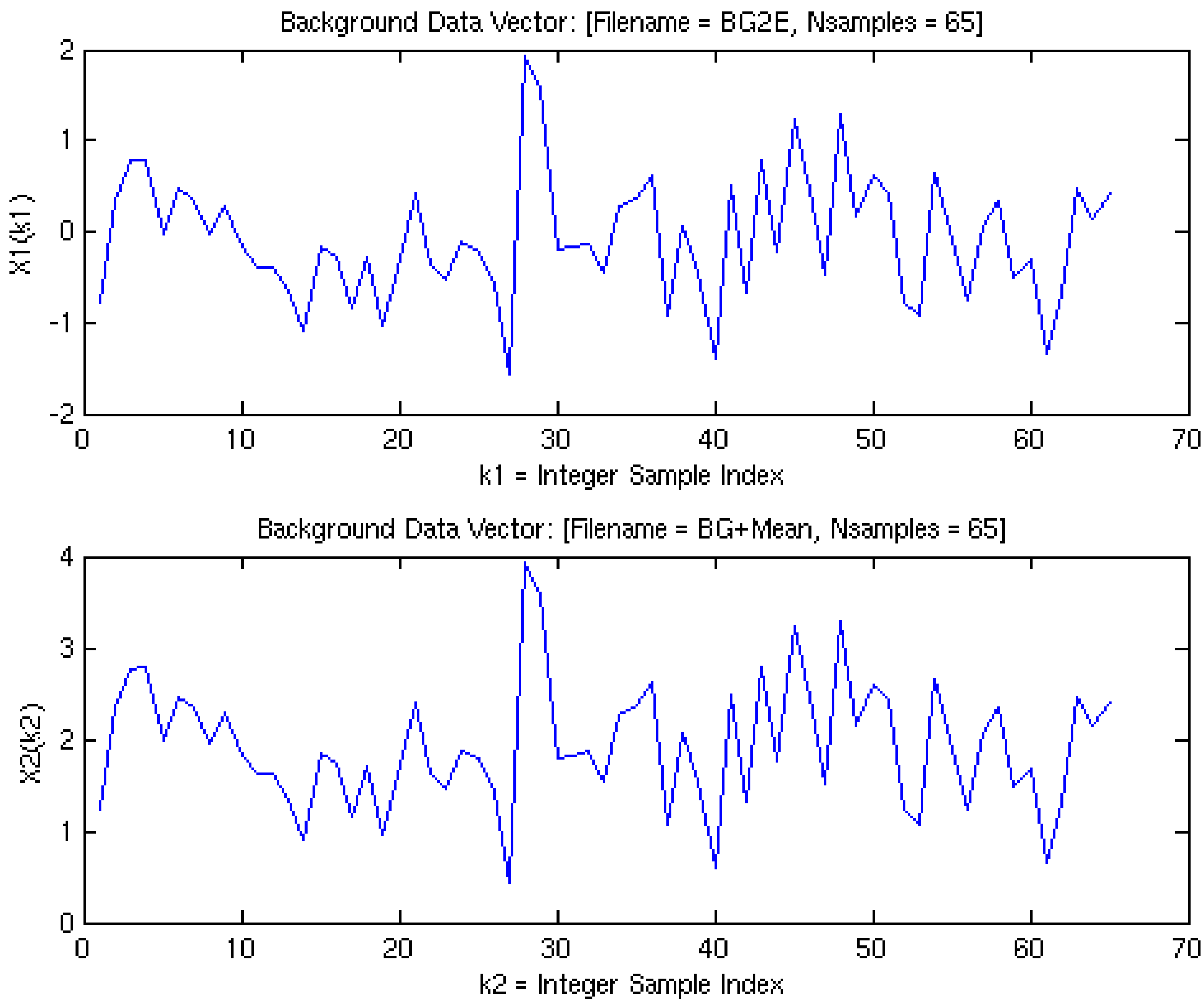


Figure 5.3: For visualization purposes, a subset of only 65 samples from the raw LASI matched filter image background data are plotted plotted as vectors. The first figure shows the raw background data. The bottom figure shows the raw background data with an artificial mean or 2.0 added to it. This was done as a demonstration of the PNN and its use (see the MATLAB code documentation).

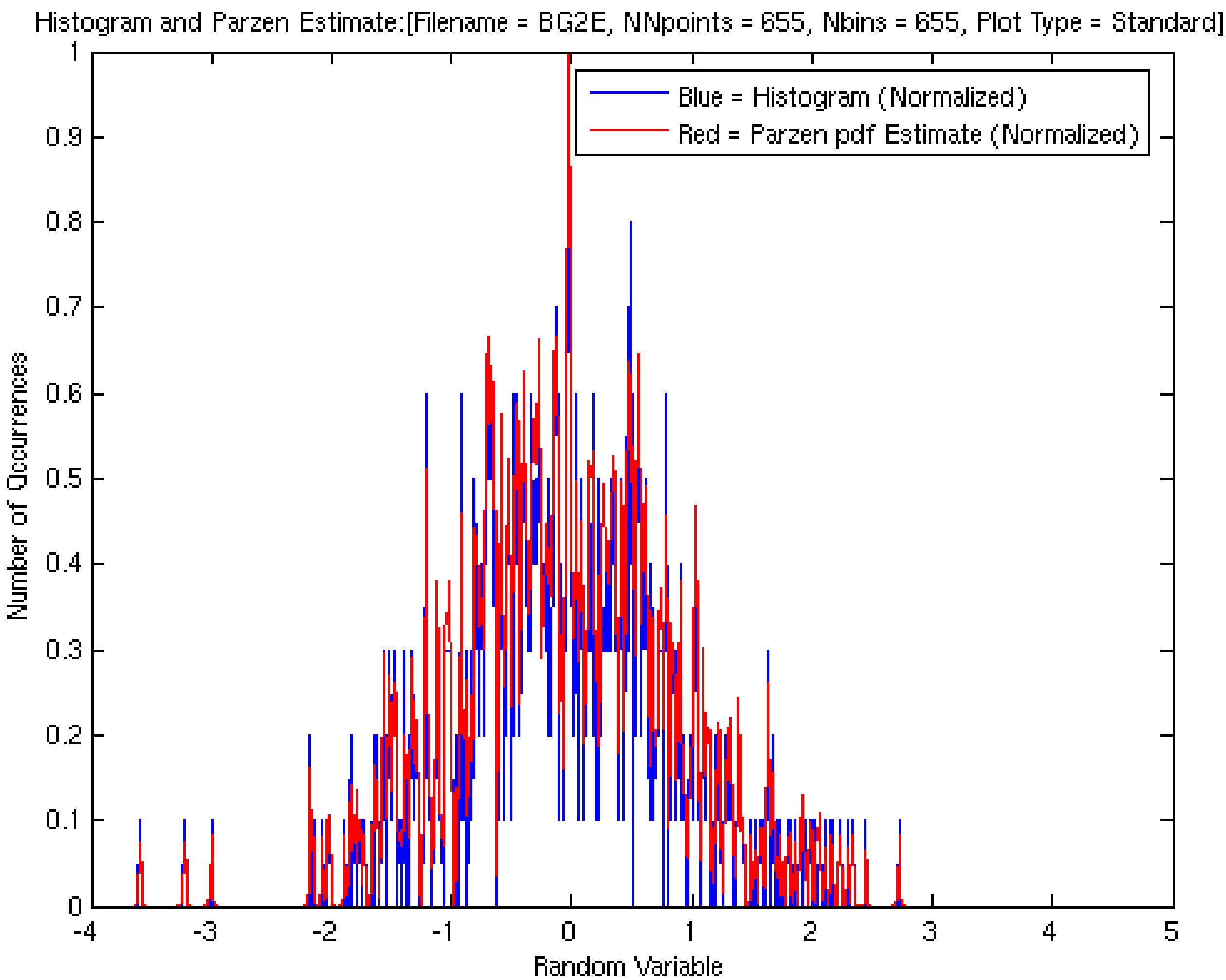


Figure 5.4: The Parzen pdf estimate and the histogram are overlayed for the background data set BG2E. We see that the estimates are similar. The number of data samples used was about one tenth of the full data set available, 655 points. For the histogram, the number of bins used was equal to the number of data samples (655 points). We did this for easy comparison with the Parzen estimator. The smoothing parameter $\sigma = .0072$ for the Parzen kernel pdf estimate was chosen using the automatic algorithm proposed by Cain [16].

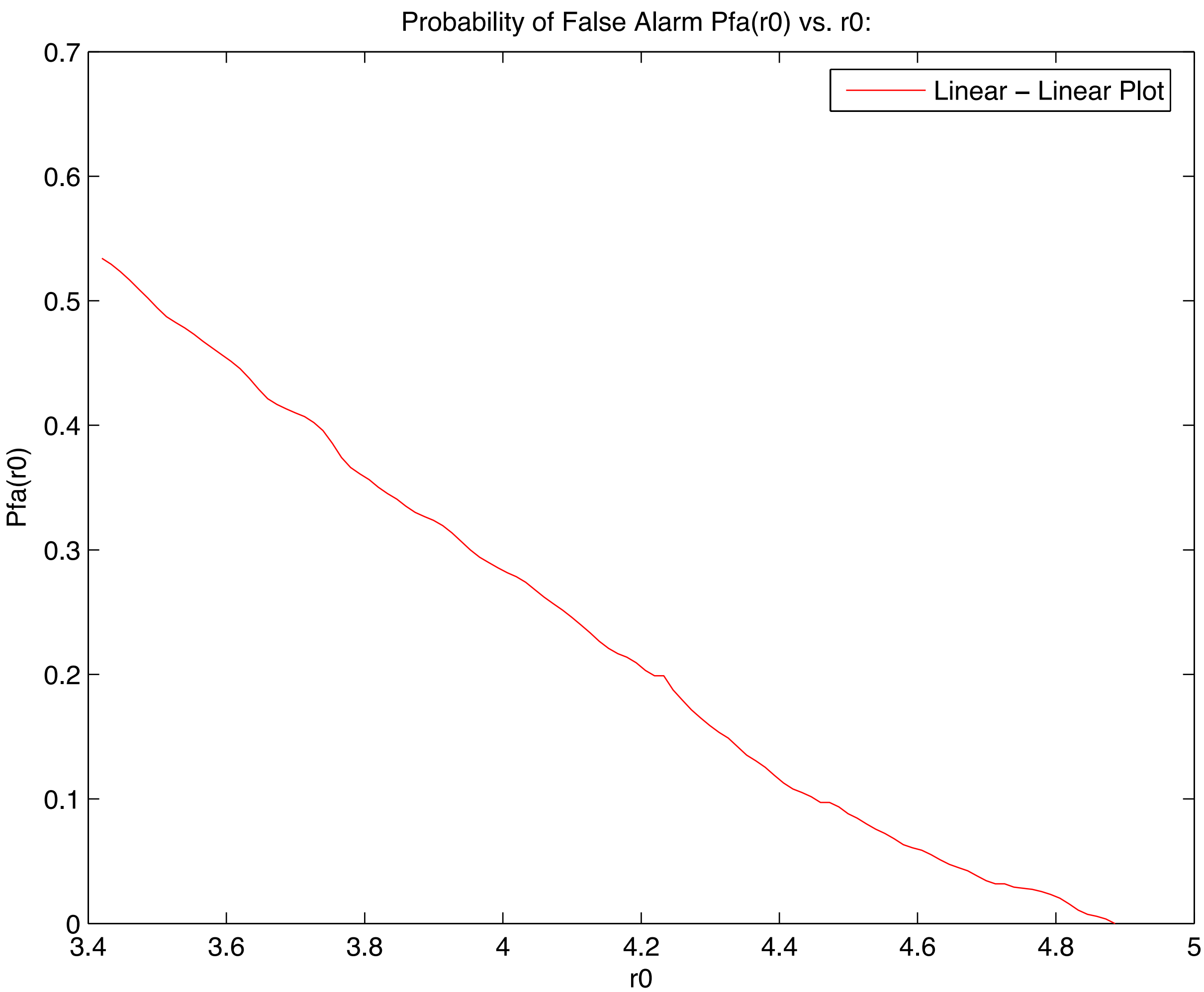


Figure 5.5: P_{FA} vs. r_0 is plotted using a linear-linear scale. This is the result of integrating the upper tail of the pdf estimate for multiple values of the decision threshold r_0. These data contain <u>nondistinct</u> values of P_{FA}.

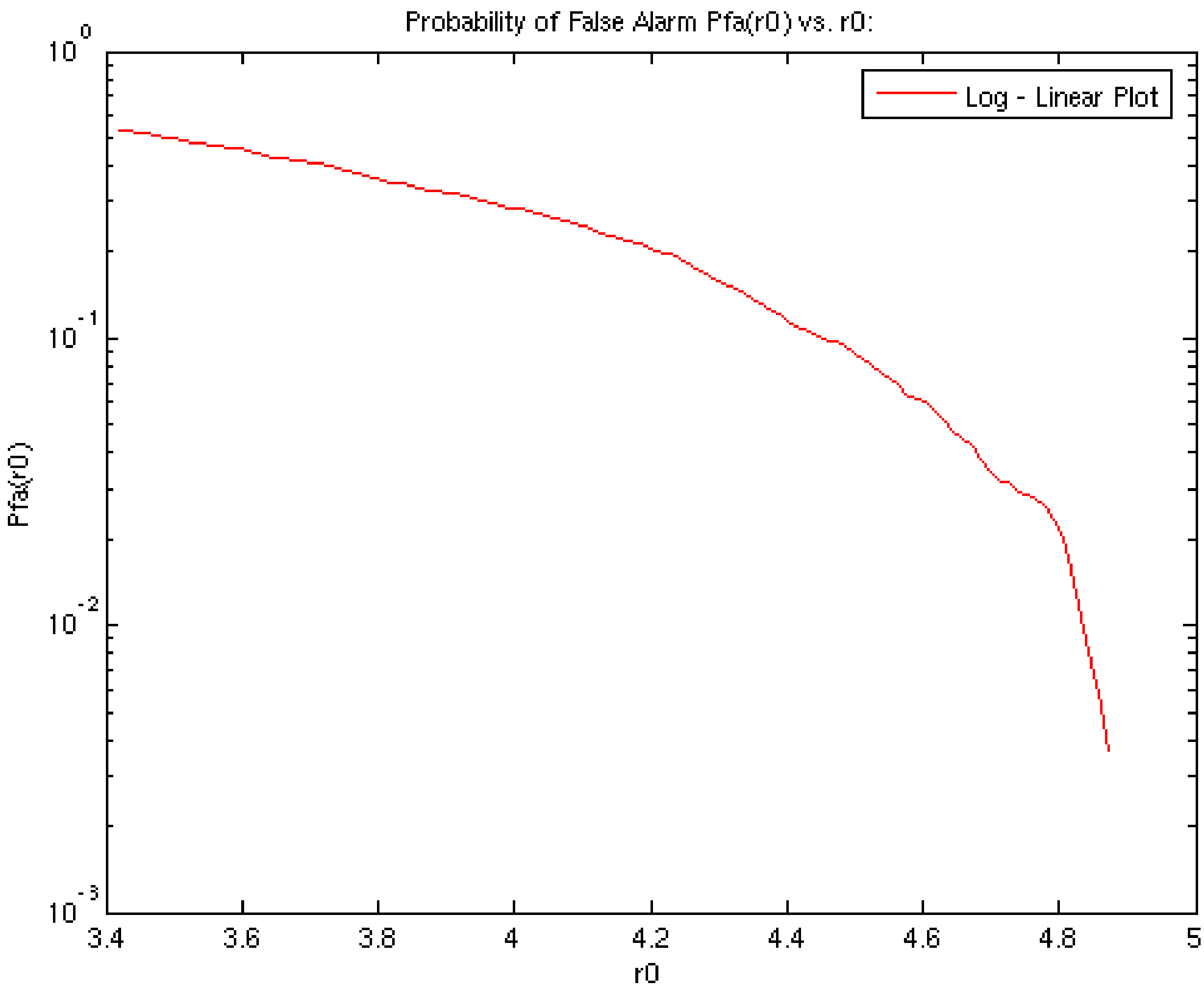


Figure 5.6: P_{FA} vs. r_0 is plotted using a log-linear scale. This is the result of integrating the upper tail of the pdf estimate for multiple values of the decision threshold r_0. These data contain nondistinct values of P_{FA}.

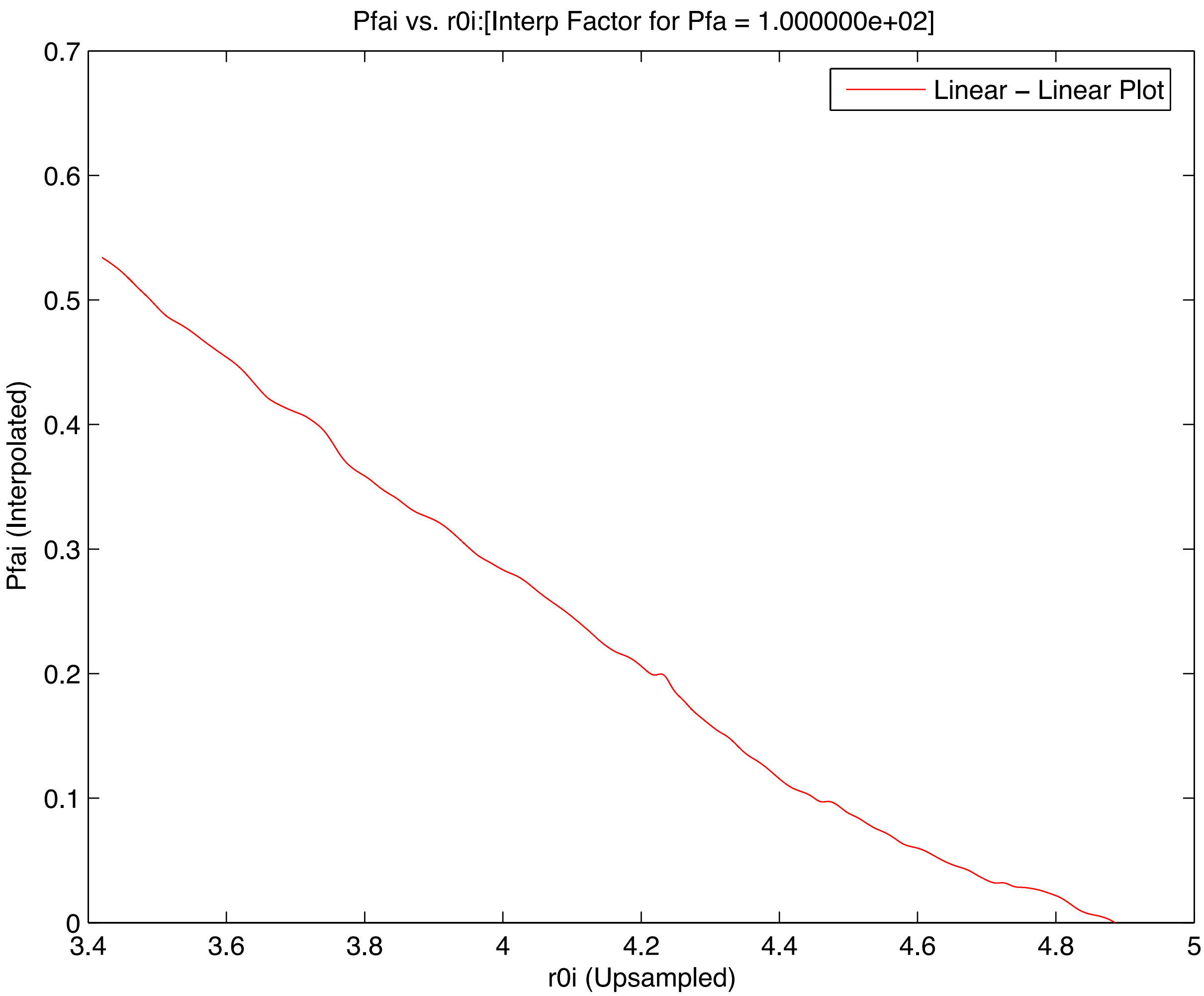


Figure 5.7: The interpolated version of P_{FA} vs. r_0 is plotted using a linear-linear scale. This is the result of integrating the upper tail of the pdf estimate for multiple values of the decision threshold r_0. These data DO NOT contain nondistinct values of P_{FA} because the interpolation removed them. The interpolation resampling factor used was 100.

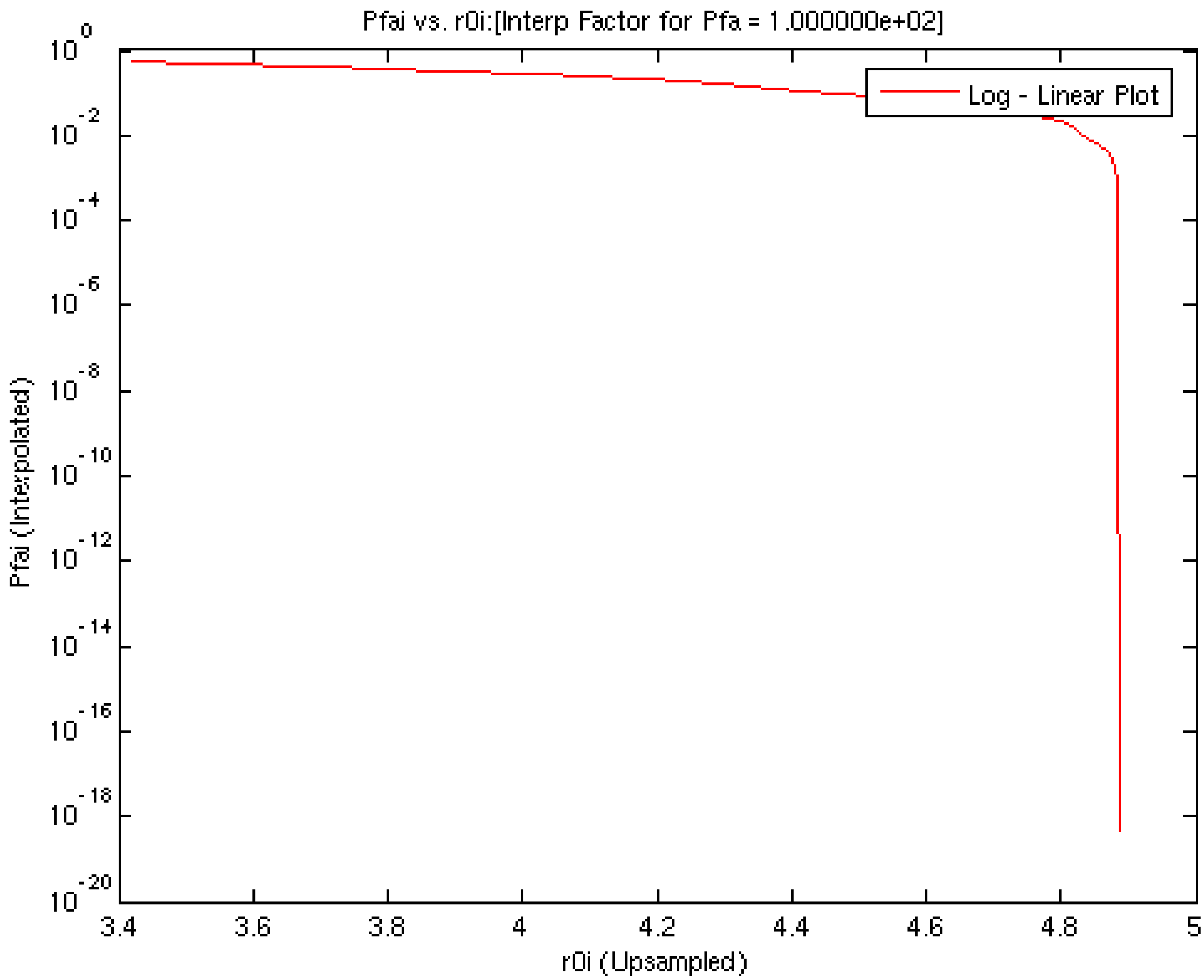


Figure 5.8: The interpolated version of P_{FA} vs. r_0 is plotted using a log-linear scale. This is the result of integrating the upper tail of the pdf estimate for multiple values of the decision threshold r_0. These data DO NOT contain nondistinct values of P_{FA} because the interpolation removed them. The interpolation resampling factor used was 100.

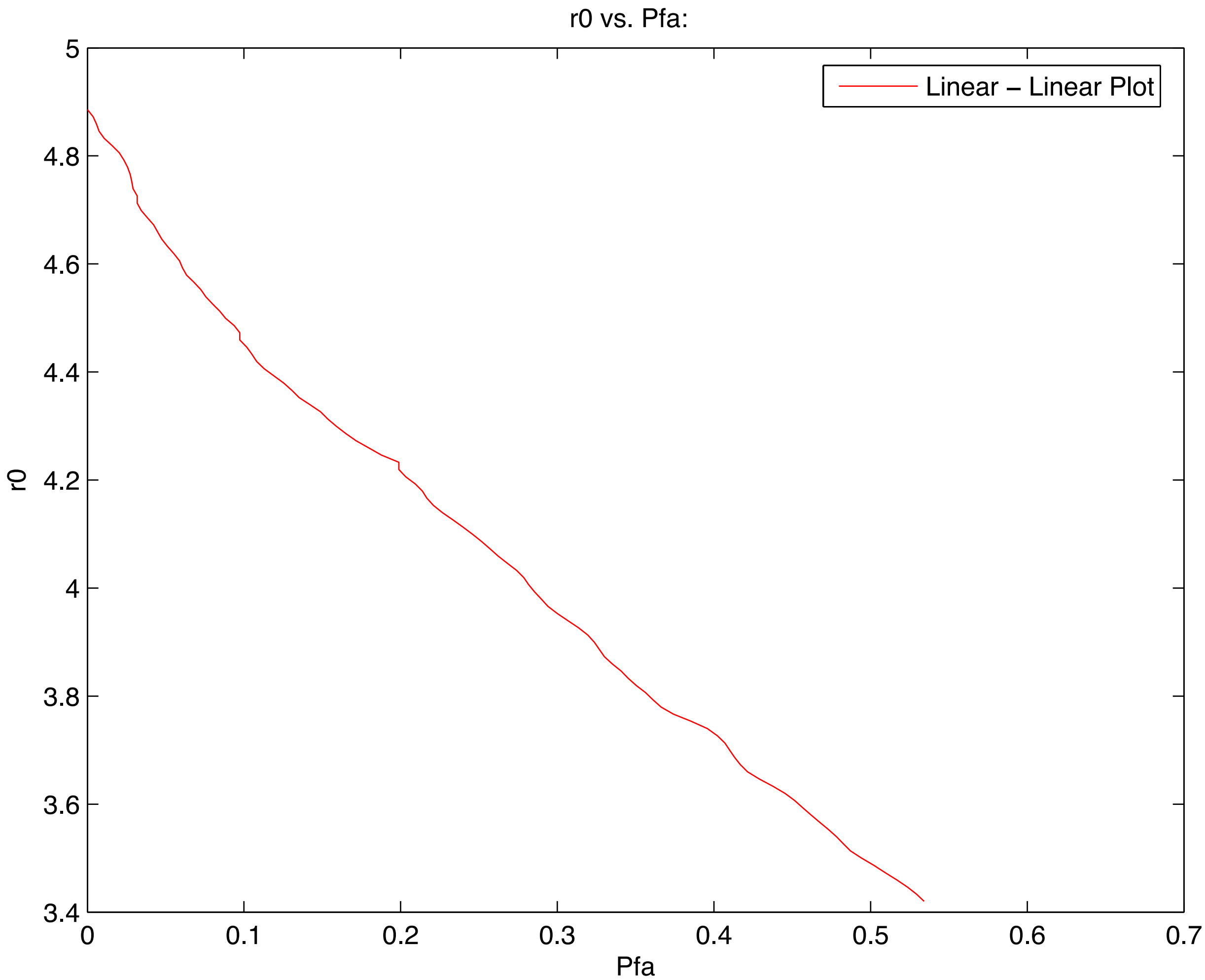


Figure 5.9: A linear-linear plot of r_0 vs. P_{FA} is depicted here.

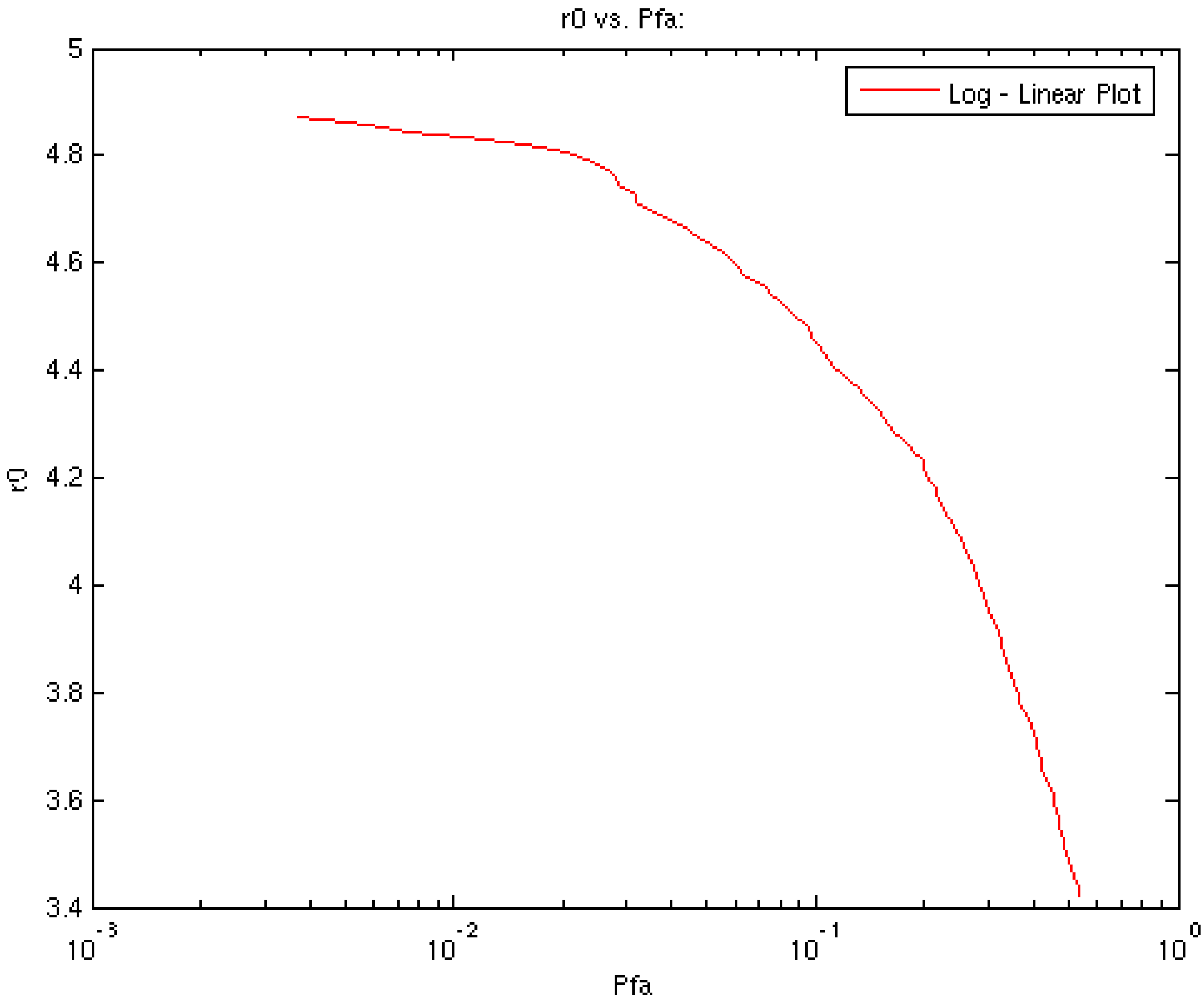


Figure 5.10: A Log-linear plot of r_0 vs. P_{FA} is depicted here.

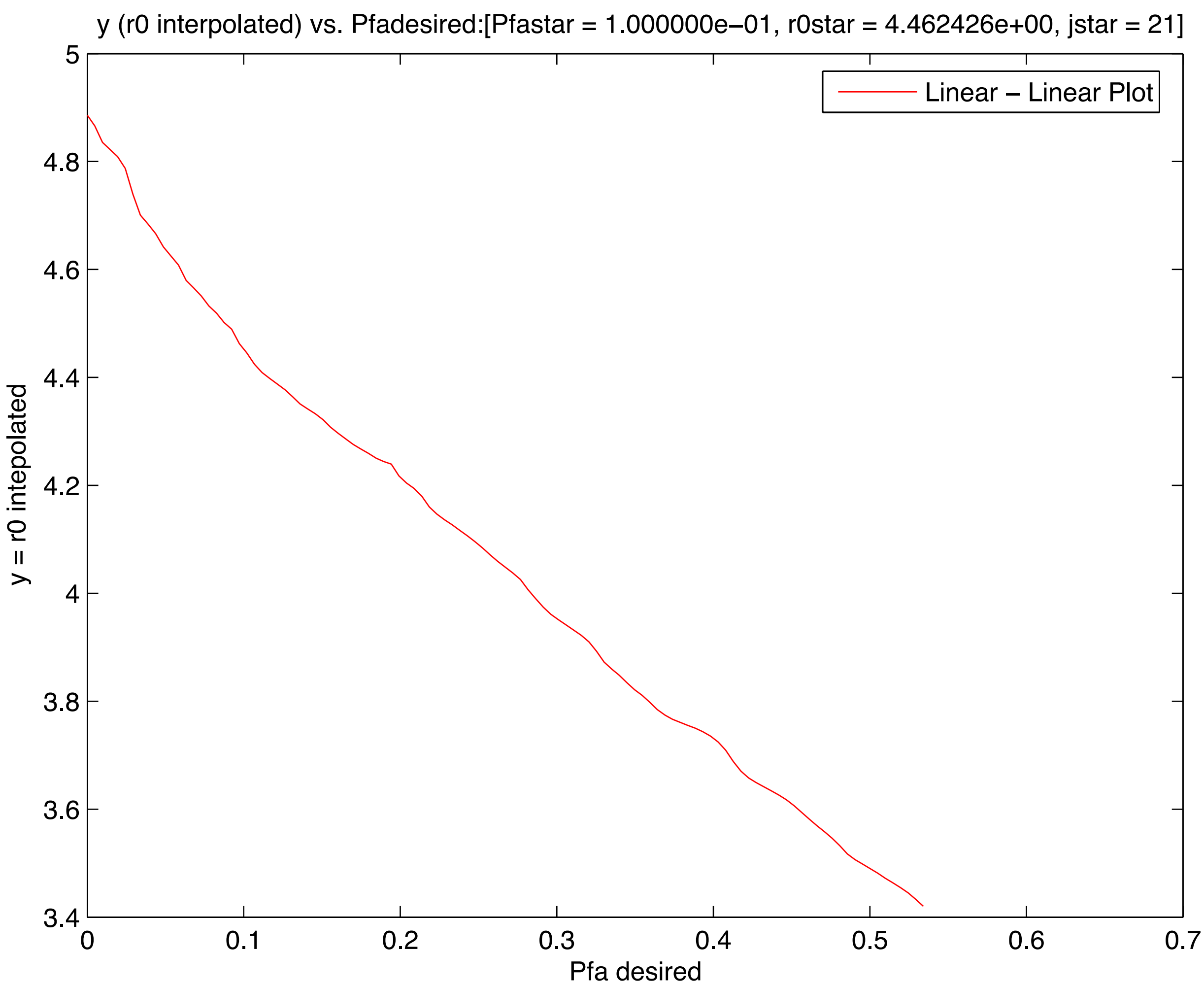


Figure 5.11: Linear-linear plot of the interpolated r_0 vs. the interpolated P_{FA}. An example of determining the decision threshold is demonstrated here. Referring to the figure title, we see that we asked the algorithm for a desired $P_{FA} = P^*_{FA} = Pfastar = .1$, and the algorithm returned the appropriate $r^*_0 = r0star = 4.462$.

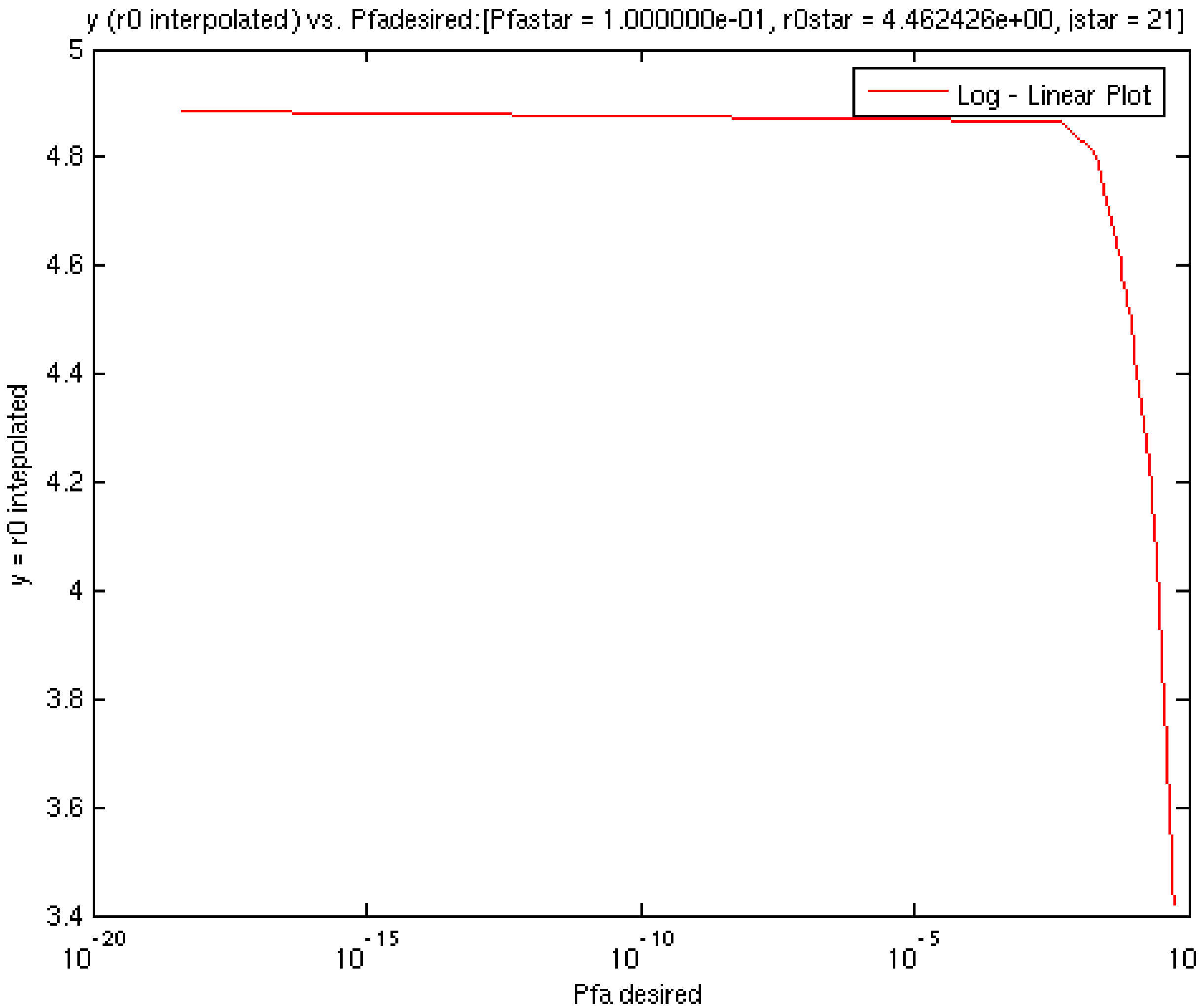


Figure 5.12: Log-linear plot of the interpolated r_0 vs. the interpolated P_{FA}. An example of determining the decision threshold is demonstrated here. Referring to the figure title, we see that we asked the algorithm for a desired $P_{FA} = P^*_{FA} = Pfastar = .1$, and the algorithm returned the appropriate $r^*_0 = r0star = 4.462$.

Chapter 6

Conclusions

We developed algorithms and MATLAB software for computing a CFAR matched filter decision threshold for the LASI project data. We demonstrated proof of principle for the algorithms using a small LASI data set. Future work includes implementing the algorithms in the IDL/ENVI language, applying the pdf estimation and CFAR threshold calculation algorithms to the LASI matched filter based upon a full-size set of global background statistics, and developing a new adaptive matched filter algorithm based upon local background statisitcs [3]. Along with this is a plan to perform a "backward" integration for the P_{FA}, rather than the forward integration used in this report [3]. Another goal is to implement the 4-Gamma pdf modeling method proposed by Stocker et. al. [4] and comparing results using histograms and the Parzen pdf estimators.

Bibliography

[1] R. S. Roberts, *A Change Detection Approach to Improved Gas Detection and Identification*, Lawrence Livermore National Laboratory informal presentation document, 2004 Technical Information Exchange Meeting, Office of Nonproliferation Research and Engineering, NA-22, 2004.

[2] R. S. Roberts, *Falcon Talon Summer Campaigns*, Lawrence Livermore National Laboratory informal presentation document, DSED Update, July 17, 2003.

[3] R. S. Roberts, *Personal communication*, 2004.

[4] A. Stocker, E. Ensafi, S. Beaven, D. Stein, *Joint Algorithm Exploitation (JALEX): Fusion of Spectral Anomaly Detection Statistics*, Report, Space Computer Corporation, Los Angeles, CA, and SPAWAR Systems Center, San Diego, CA, March 2001.

[5] A. D. Stocker and A. P. Schaum, *Hyperspectral Image Classification and Detection Using the Stochastic Mixing Model*, Proceedings IRIS CCD Conference, Monterey, CA, Space Computer Corporation, Santa Monica, CA and Naval Research Laboratory, Washington, D.C., October 1997.

[6] P. V. Villeneuve and A. D. Stocker, *HIRIS Algorithm Physics Model Description*, Version 2.3, Space Computer Corporation, Los Angeles, CA, June 2, 2000.

[7] A. D. Stocker, *HIRIS Baseline Algorithms for Level 3 Analysis*, Version 2.0, Space Computer Corporation, Los Angeles, CA September 12, 2000.

[8] X. Yu, I. S. Reed, A. D. Stocker, *Comparative Performance Analysis of Adaptive Multispectral Detectors*, IEEE Trans. Signal Processing, Vol., 41, No. 8, 1993.

[9] I. S. Reed and X. Yu, *Adaptive Multiple-Band CFAR Detection of an Optical Pattern with Unknown Spectral Distribution*, IEEE Trans. Acoustics, Speech and Signal Processing, Vol. 38, No. 10, October 1990.

[10] H. L. Van Trees, *Detection, Estimation and Modulation Theory, Part I Detection, Estimation and Linear Modulation Theory*, John Wilery and Sons, 1968.

[11] A. Whalen, *Detection of Signals in Noise* Academic Press, 1971.

[12] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume II Detection Theory*, Prentice Hall, 1998.

[13] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 1965.

[14] B. W. Silverman, *Density Estimation for Statisitcs and Data Analysis*, Chapman Hall/CRC, 1986.

[15] D. F. Specht, *Probabilistic Neural Networks*, Neural Networks, Vol. 3, pp. 109-118, 1990.

[16] J. Bibb Cain, *An Improved Probabilistic Neural Network and its Performance Relative to Other Models*, SPIE Vol. 1295 Applications of Arificial Neural Networks, 1990.

[17] D. F. Specht, *A General Regression Neural Network*, IEEE Trans. Neural Networks, Vol. 2, No. 6, November 1991.

[18] D. F. Specht, *Probabilistic Neural Networks and the Polynomial Adaline as Complementary Techniques for Classification*, IEEE Trans. Neural Networks Vol. 1, No. 1, March 1990.

[19] M. E. Glinsky, G. A. Clark, Peter K.-Z. Cheng, K. R. Sandhya Devi, J., H. Robinson, and G. E. Ford, *Automatic Event Picking in Prestack Migrated Gathers Using a Probabilistic Neural Network*, Geophysics, Vol. 66, No. 5 (September-October, 2001), pp. 1488-1496.

[20] G. A. Clark, S. K. Sengupta, W. D. Aimonetti, F. Roeske and J. G. Donetti, *Multispectral Image Feature Selection for Land Mine Detection*, Lawrence Livermore National Laboratory report UCRL-JC-124375-Rev.1, IEEE Trans. Geoscience and Remote Sensing, January, 2000, pp. 304-311.

[21] G. A. Clark, *The Revelations of Acoustic Waves*, Science and Technology Review, Lawrence Livermore National Laboratory, Lawrence Livermore National Laboratory, UCRL-52000-99-5, U. S. Government Printing Office 1999/783-046-80013, May, 1999.

[22] P. C. Schaich, G. A. Clark, K.-P. Ziock, S. K. Sengupta, *Automatic Image Analysis for Detecting and Quantifying Gamma-Ray Sources in Coded Aperture Images*, IEEE Transactions on Nuclear Science, Vol. 43, No. 4, August, 1996.

[23] G. A. Clark, D. D. Scott, et. al., *Heart Valve Classification Using Opening and Closing Sounds*, Imaging Sciences Workshop, Center for Advanced Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 21-22, 1996.

[24] G. A. Clark, M. E. Glinsky, K. R. S. Devi, J. H. Robinson, P. K.-Z. Cheng, G. E. Ford, *Automatic event picking in pre-stack migrated gathers using a probabilistic neural network*, Society of Exploration Geophysicists (SEG) International Exposition and 66th Annual Meeting, Denver, Colorado, November 10-15, 1996.

[25] G. A. Clark, B. C. Bowman, N. Boruta and G. H. Thomas, *Classification of Heart Valve Condition Using Acoustic Measurements*, Imaging Sciences Workshop, Center for Signal and Imaging Sciences, Lawrence Livermore National Laboratory, November 15-16, 1994.

[26] G. A. Clark, M. C. Axelrod and D. D. Scott, *Classification of Heart Valve Sounds from Experiments in an Anechoic Water Tank* Lawrence Livermore National Laboratory report UCRL-JC-134634, June 1, 1999.

[27] B. C. Bowman, G. A. Clark, G. H. Thomas, and N. Boruta, *Classification of Bjork-Shiley Convexo-Concave Heart Valve Single-Leg Separations from Acoustic Measurements*, Lawrence Livermore National Laboratory report UCRL-JC-119858, 1994.

[28] S. K. Sengupta, *Personal Communication*, Lawrence Livermore National Laboratory, 1991.

[29] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973.

[30] T. Y. Young and K-S Fu, *Handbook of Pattern Recognition and Image Processing*, Academic Press, 1986.

[31] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes*, Cambridge University Press, 1986.

[32] F. A. Acton, *Numerical Methods That Work*, Harper and Row, 1970.

[33] G. Bachman, L. Narici, *Functional Analysis*, Academic Press, 1966.

[34] B. Noble, *Applied Linear Algebra*, Prentice-Hall, 1969.

[35] R. V. Churchill *Complex Variables and Applications*, McGraw-Hill, 1960.

[36] *MATLAB Reference Manual*, The Mathworks, Natick Massachussetts, 1993.