3-27-2008

Improved Feature Extraction, Feature Selection, and Identification Techniques That Create a Fast Unsupervised Hyperspectral Target Detection Algorithm

Robert J. Johnson

**IMPROVED FEATURE EXTRACTION, FEATURE
SELECTION, AND IDENTIFICATION TECHNIQUES
THAT CREATE A FAST UNSUPERVISED
HYPERSPECTRAL TARGET DETECTION ALGORITHM**

THESIS

Robert J Johnson, Captain, USAF

AFIT/GOR/ENS/08-07

IMPROVED FEATURE EXTRACTION, FEATURE SELECTION, AND
IDENTIFICATION TECHNIQUES THAT CREATE A FAST UNSUPERVISED
HYPERSPECTRAL TARGET DETECTION ALGORITHM

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Robert J Johnson, BA

Captain, USAF

March 2008

IMPROVED FEATURE EXTRACTION, FEATURE SELECTION, AND
IDENTIFICATION TECHNIQUES THAT CREATE A FAST UNSUPERVISED
HYPERSPECTRAL TARGET DETECTION ALGORITHM

Robert J Johnson, BA
Captain, USAF

Approved:

_____   _____
Kenneth W. Bauer, Jr., Ph.D. (Chairman)   date

_____   _____
Dr. J.O. Miller (Member)   date

Abstract

This research extends the emerging field of hyperspectral image (HSI) target detectors that assume a global linear mixture model (LMM) of HSI and employ independent component analysis (ICA) to unmix HSI images. Via new techniques to fully automate feature extraction, feature selection, and target pixel identification, an autonomous global anomaly detector, AutoGAD, has been developed for potential employment in an operational environment for real-time processing of HSI targets. For dimensionality reduction (initial feature extraction prior to ICA), a geometric solution that effectively approximates the number of distinct spectral signals is presented. The solution is based on the theory of the shape of the eigenvalue curve of the covariance matrix of spectral data containing noise. For feature selection, previously a subjective definition called significant kurtosis change was used to denote the separation between targets classes and non-target classes. This research presents two new measures, potential target signal to noise ratio (PT SNR) and max pixel score which is computed for each of the ICA features to create a new two-dimensional feature space where the overlap between target and non-target classes is reduced compared to the one dimensional kurtosis value feature space. Finally, after target feature selection, adaptive noise filtering, but with an iterative approach, is applied to the signals. The effect is a reduction in the power of the noise while preserving the power of the target signal prior to target identification to reduce false positive detections. A zero-detection histogram method is applied to the smoothed signals to identify target locations to the user. MATLAB code for the AutoGAD algorithm is provided.

To my family and my AFIT family.

I would like to especially thank Capt David Bethea and his family for their unconditional friendship throughout this journey through graduate school. From late nights at the kitchen table working through optimization and statistics problems to always being welcome for dinner and the unforgettable adventures of trying to watch a four year old while completing school work, I was treated like a member of their family. Thank you.

Acknowledgments

I would like to express my appreciation to my faculty advisor, Dr. Kenneth Bauer, for his confidence in my abilities, guidance, and ideas throughout the course of this thesis effort. I also appreciate the opportunities Dr. Miller gave me by encouraging me to brief this research to several DVs who visited AFIT to raise awareness as to its potential.

I would, also, like to Dr. Michael Eismann and Joe Meola at AFRL/RYJT for sharing their invaluable expertise in the field of HSI and allowing me to participate in their HSI data collection efforts. The experience gave me perspective into the HSI data collection and processing requirements. In addition, a special thanks goes to LtCol Tim Smetek for his time in familiarizing me with the tools necessary to operate on HSI data cubes.

Robert J. Johnson

Table of Contents

List of Figures

List of Tables

IMPROVED FEATURE EXTRACTION, FEATURE SELECTION, AND IDENTIFICATION TECHNIQUES THAT CREATE A FAST UNSUPERVISED HYPERSPECTRAL TARGET DETECTION ALGORITHM

I. Introduction

1.1 Background

Distinguishing man-made objects from natural objects in rural environments is of particular interest when man-made objects are targets within the military context. Hyperspectral sensors offer a passive means (as opposed to radar where an enemy can be alerted to surveillance) by which to identify targets by exploiting information within the electromagnetic spectrum (EM). The typical range of the EM spectrum exploited is the optical range which includes ultraviolet, visible, and infrared wavelengths as shown in Figure 1-1 below. Optical regions useful for remote sensing extend beyond the regions where photography can be used (Landgrebe, 2003: 13).
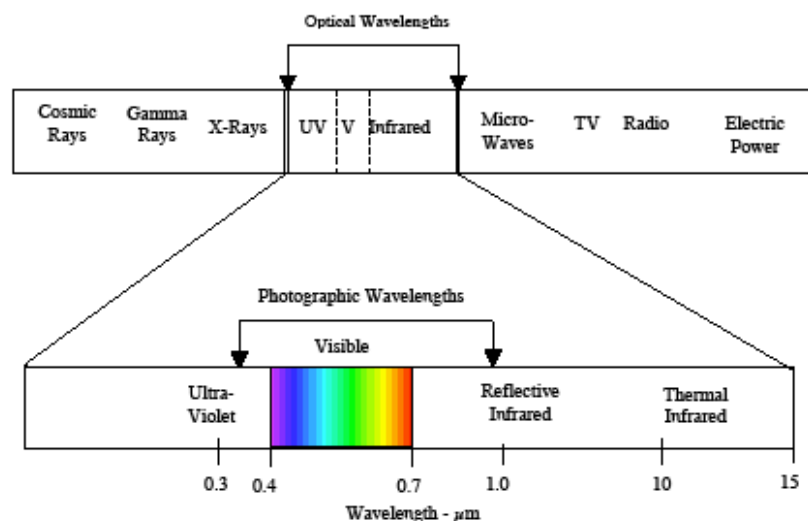
Figure 1-1. EM Spectrum (Landgrebe, 2003:14)

The specific physics and hardware specifications to collect hyperspectral images (HSI) is beyond the scope of this thesis, but a general explanation of the nature of an HSI image will be explained.

A particular area being imaged by an HSI sensor is divided into a raster grid with each grid cell or pixel corresponding to a rectangular sub-region of the image. The physical dimensions of the pixel correspond to the spatial resolution of the sensor which could range typically from some fraction of a meter to tens of meters (Smetek, 2007:16). Other than spatial resolution, the sensor can be characterized by the range of EM wavelengths it is capable of measuring and the smallest detectable wavelength difference (spectral resolution). Hyperspectral sensors typically record energy, measured in radiance (watts per square meter per steradian), over hundreds of discrete intervals called spectral bands (the width of the interval is determined by the spectral resolution) across some subset of the optical wavelengths. The sensor records the amount of energy radiated by each pixel over all the sensor's spectral bands.

It may not be possible to distinguish between natural foliage and camouflage netting concealing tanks or troop tents when examining a normal RGB (red green blue) digital image taken from a surveillance aircraft. However, when analyzing the same image taken with an HSI sensor that considers more than just the visible spectrum, the camouflage material will reflect electromagnetic energy different than the surrounding foliage and thus can be identified. This difference in the radiated energy can be observed when analyzing the spectral signatures of the objects. The spectral signature of a pixel would be a plot of the pixel's response across each of sensor's spectral bands.

It should be noted that whereas the HSI sensor collects data in units of radiance for each spectral band, data is usually converted to reflectance. When atmospheric correction is performed on the radiance data, the effects of path radiance (energy reflected by the atmosphere into the sensor that does not reach the ground) and skylight (energy that bounces off the atmosphere and is reflected by the ground) are removed. The resulting data is in terms of reflectance signatures and represent the relative amount of energy from the sun (without path radiance and skylight) that hits the Earth's surface and is reflected (Smetek, 2007: 12-13).

1.2 HSI Data Representation

Before proceeding further, the reader should be acquainted with how HSI data is organized for processing. HSI data is represented as a cube where the first two indices of (i, j, k) coordinate triple represent the spatial location of the pixel in the image and the third index defines the pixel's reflectance response in the kth spectral band. Figure 1-2 below is an example of an image cube.
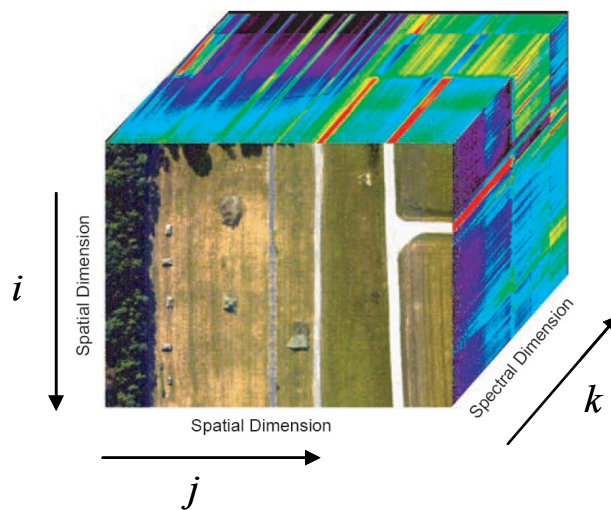
Figure 1-2. Hyperspectral Data Cube (Shaw and Manolakis, 2003: 13)

In order to perform operations on the data cube such as mean and covariance calculations

for example, the data cube is reshaped in to a data matrix. Say this particular image was

150 pixels by 200 pixels by 210 bands making a total of 30,000 pixels. Reshaping this

cube into a data matrix would result in a matrix with dimensions of 30,000 pixels by 210

bands. Figure 1-3 shows the process.
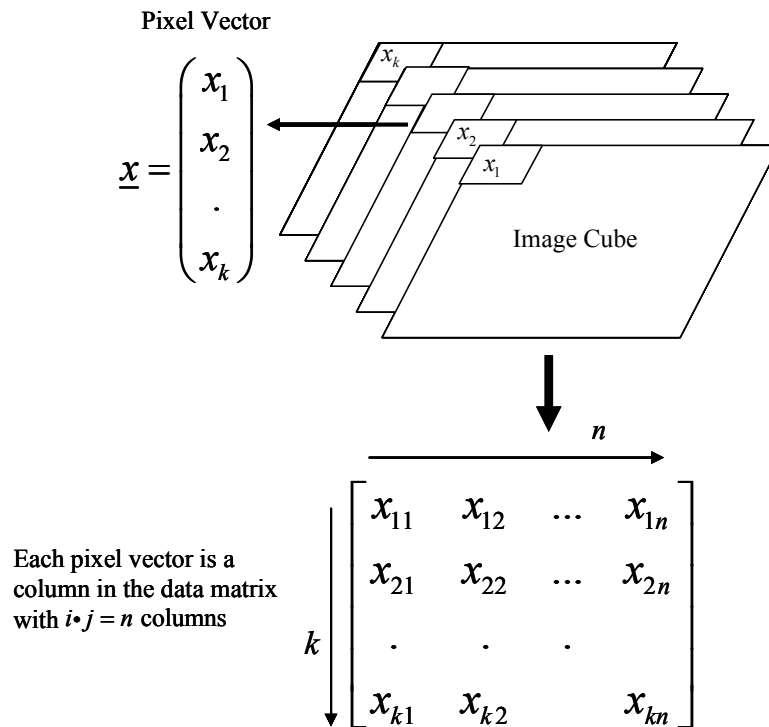
Figure 1-3. Reshaping Image Cube into Data Matrix

As shown in Figure 1-3 each pixel in the image cube can be considered a vector where

each element is a reflectance response in each of the k spectral bands. Starting from the

top left of the cube and moving down the first column $(j = 1)$ of the image, each pixel

vector in the first column is formed into a column in the data matrix. The process is

repeated starting with the second column and finishing with the jth column. The result is a data matrix with *k* rows (one for each spectral band) and *n* columns (one for each pixel observation). It should be noted that in some texts the data matrix is transposed from the one presented in Figure 1-3 to be *n* pixel observations by *k* spectral bands.

1.3 Approaches

In order to find targets two main methods are pursued. The first method is referred to as signature matching. Here, one compares pixel vectors (i.e. pixel signatures) to a library of target reflectance signatures to determine which pixels belong to targets of interest. Figure 1-4 below shows how spectral signatures of different materials may differ across the spectral bands of an HSI sensor.
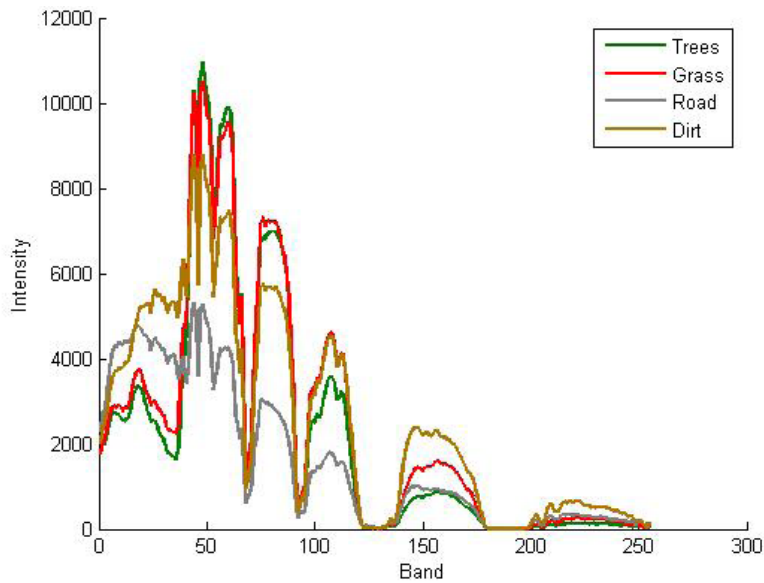
Figure 1-4. Spectral Signatures of Different Materials (Smetek, 2007:19)

The second method is referred to as anomaly detection. Although definitely not comprehensive, two common categories are (i) distribution based anomaly detectors (local or global) that assume a distribution (typically normal) of the pixel vectors and (ii) global linear mixture model detectors.

Distribution Based Anomaly Detectors

In local anomaly detection, a scanning window (which is some fraction of the size of the entire image) is moved over the image and centered on a pixel of interest. At each stop the local mean and covariance is calculated and if the centered pixel of interest is beyond some statistical distance threshold (typically a Mahalanobis distance) from the mean then it is nominated as an outlier. Such local anomaly detectors are referred to as local normal models, i.e. the pixel vectors are assumed to be distributed Gaussian (normal) with mean μ and covariance Σ. The RX detector is one such detector.

According to Stein, Beaven, Hoff, Winter, Schaum, and Stocker (2002:62), the local Gaussian model may not be a valid for hyperspectral data if relatively small regions contain multiple materials. Based on goodness-of-fit tests of a hyperspectral scene to a local normal model, the local normal model was rejected for more than 90% of the pixels. Thus, the local normal model may not capture the complexity of hyperspectral imagery. They assert that global mixture distributions will provide more accurate descriptions.

Global normal mixture models assume an image is made of *C* classes where each class has a separate multivariate Gaussian (normal) distribution with parameters μ_c and Σ_c for class *c*. The overall pdf of the scene is described by a Gaussian mixture distribution:

$$g(\underline{x}) = \sum_{c=1}^{C} \pi_c N(\underline{x}|\underline{u}_c, \Sigma_c); \ \pi_c \geq 0; \ \sum_{c=1}^{C} \pi_c = 1 \qquad (1.1)$$

where π_c is the probability of belonging to class c (Stein and others, 2002:63). Stein et al. use a stochastic expectation maximization (SEM) method to determine the location of the classes and their parameters. However, as Smetek points out, no guidance is given on how to determine the number of classes C (Smetek, 2007:48). Rather than the SEM approach, another way to determine the location of the classes is via the k-means clustering algorithm. However, the number of classes k is subjectively specified by the user (Smetek, 2007:49). Similar to the local anomaly detectors that identify anomalous pixels as statistical outliers in the local scanning window, the Gaussian mixture models identify anomalous pixels that are outliers for each of these clusters' distributions.

Global Linear Mixture Model Detectors

Rather than presuming some distribution of the pixel vectors in the scene, this approach assumes that each pixel vector observation is a convex combination of a deterministic number of endmembers, i.e. a finite number of distinct spectral signatures. The coefficients of the convex combination are interpreted as the fractional abundance of each endmember in a particular pixel. As will be explained in great detail in chapter 2, one must solve for the abundances in what is called unmixing the image. Target-like endmembers are identified based on properties of the abundance estimates (Stein and others, 2002:60). Given this research will employ the linear mixture model of HSI, a lengthy discussion of this model and a solution methodology to solve for the abundances via a technique called independent component analysis (ICA) will be presented in chapter 2.

1.4 Research Objectives

Based on the linear mixture model of HSI, this research seeks to develop a fast and truly autonomous (unsupervised) global anomaly detector, dubbed AutoGAD, to locate targets in an HSI image. Autonomous, implies the only input from the user is the image cube. The rest of the decisions to identify targets are made via the AutoGAD algorithm. By fast, the goal is to be able to process images in terms of minutes and even seconds in order to provide real-time targeting for operational employment.

As will be discussed in chapter 2 and chapter 3, feature extraction refers to reducing the dimensionality of HSI data by projecting the k dimensional data onto a lower dimensional orthogonal subspace (commonly via a technique referred to as principal components analysis) where much of the structure of the original data is maintained. The first objective of this thesis is to develop a way to effectively, efficiently, and autonomously determine the number of dimensions to project the data. As will be explained later, the correct number of dimensions to choose, refers to the number of distinct endmember signatures present in the image.

After reducing the dimensionality, the features (which correspond to the n pixels' scores on each of the new coordinate axes) will be projected again to a new set of axes that are not only orthogonal, but independent via a method previously mentioned called ICA. This projection is determined by an optimization scheme which will be rigorously developed in chapter 2. The second objective of this thesis will be to determine which objective function, that approximates a measure of independence, yields the least amount

of variability in the solution per each test image it is applied to. This will ensure a level of robustness to the new detector.

From the final set of features produced by ICA, a process called feature selection must choose which features are target like based on some statistical measure of target 'likeness'. Currently, features with high kurtosis values are used to nominate features that belong to the target class. However, the process of selecting the cutoff kurtosis value between target classes and non-target classes is a supervised procedure. As will be illustrated in chapter 3, attempts to automate the procedure is problematic and the overlap between non-target kurtosis values and target kurtosis values calls for the investigation of alternate measures of target 'likeness' that minimizes the overlap between the classes so that the cutoff decision between the classes can be automated. This is the third objective of this thesis.

As previously explained a feature is an n dimensional vector which represents each of the n pixels' scores on a particular axis in the ICA space. The selected target features, are in other words the selected target axes in the ICA space. The next step in the target detection process is to identify which pixels in each of the selected features are the target pixels. The fourth objective of this thesis is to find an unsupervised way to locate these pixels that reduce the amount of false positives detected.

1.5 Assumptions

Anomaly detectors, since they do not assume a priori knowledge of target signatures, must make a few assumptions about the nature of targets. Further the domain of effective application of anomaly detectors is limited. Targets in an image are assumed

to be a small, rare occurring class (as opposed to a large class like a forest) and have a spectral signature distinct from the rest of the spectral signatures present in the image. Man-made objects in large rural areas of large classes of natural occurring objects like dirt, grass, trees, bushes, gravel, sand, etc...meet the definition of targets. However, in an urban environment, most of the objects are man-made objects. Thus, the definition of small, rare occurring class and distinct spectral signature breaks down. For this reason the domain of the new anomaly detector is assumed to be rural environments.

Although not mentioned previously, an important initial processing requirement before executing AutoGAD is the removal of what are called the atmospheric absorption bands from the data. Atmospheric absorption bands are bands in which the energy at these wavelengths is almost entirely absorbed by the atmosphere. Therefore, the sensor detects primarily random noise at these wavelengths. Given the wavelengths of atmospheric absorption are known, the bands in the sensor referred to as the noise bands are assumed to be known. Further, in addition to the atmospheric absorption bands there may be other bands in the sensor that record significant noise, often at the extremes of the sensor's band range (Taitano, 2007:33-34). Before the application of the AutoGAD, it is assumed that the user has a priori knowledge of the 'good' non-noise bands in the sensor. For each different sensor that employs AutoGAD a separate input for the good bands will be required.

Finally, the set of test and validation images presented in chapters 3 and 4 are assumed to be a representative sample from the domain of rural HSI images with targets.

II. Literature Review and Critical Analysis of Current Practices

2.1 Linear Mixture Model (LMM) for HSI

As previously described, the LMM assumes each pixel vector is a "linear mixture of a discrete number of pure deterministic material spectra" (Chang, 2007:108). Terminology such as 'pure spectra', 'pure materials', or 'pure pixels' are often referred to as endmembers each having a characteristic spectral signature.

> The physical basis for the linear mixture model is that hyperspectral image measurements often capture multiple material types in an individual pixel and that measured spectra can be described as a linear superposition of the spectra of the pure materials from which the pixel is composed. The weights of the superposition correspond to the relative abundances of the various pure materials. This assumption of linear superposition is physically well-founded in situations, for example, where the sensor response is linear, the illumination across the scene is uniform, and there is no scattering (Chang, 2007:108).

Nonlinear mixing occurs when, for example, there is multiple scattering of light between elements in a scene.

> Despite the potential for nonlinear mixing in real imagery, the linear mixing model has been found to be a fair representation of hyperspectral data in many situations (Chang, 2007:108).

The LMM for a particular pixel observation i, where $i = 1,2,...,N$ observations, formulates as follows:

$$
\begin{pmatrix} x_{1j} \\ x_{2j} \\ \cdot \\ x_{Kj} \end{pmatrix} = s_{1j} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \cdot \\ \varepsilon_{K1} \end{pmatrix} + s_{2j} \begin{pmatrix} \varepsilon_{12} \\ \varepsilon_{22} \\ \cdot \\ \varepsilon_{K2} \end{pmatrix} + ... + s_{Pj} \begin{pmatrix} \varepsilon_{1P} \\ \varepsilon_{2P} \\ \cdot \\ \varepsilon_{KP} \end{pmatrix} + \begin{pmatrix} r_{1j} \\ r_{2j} \\ \cdot \\ r_{Kj} \end{pmatrix}
$$

$$\quad \underline{x}_j \qquad\qquad \underline{\varepsilon}_1 \qquad\qquad \underline{\varepsilon}_2 \qquad\qquad\quad \underline{\varepsilon}_P \qquad \underline{r}_j$$

$$or \qquad\qquad\qquad\qquad\qquad\qquad (2.1)$$

$$\underline{x}_j = \sum_{p=1}^{P} s_{pj}\underline{\varepsilon}_p + \underline{r}_j$$

$$\underline{x}_j = E_{KxP} \cdot \underline{s}_j + \underline{r}_j$$

where

$\underline{x}_j \equiv$ pixel column vector where each element is an intensity value (in terms of spectral reflectance) for one of K spectral bands for observation j, where $j = 1,...,N$.

$\underline{\varepsilon}_p \equiv$ column vector of E_{KxP} that represents the spectral signature for the pth endmember, where $p = 1, 2,..., P$. Each element is an intensity value for one of the K spectral bands.

$s_{pj} \equiv$ abundance fractions of the pth endmember in $\underline{x}_j$, the jth pixel vector observation. Note, since s_{pj} are abundance fractions,

$$\sum_{p=1}^{P} s_{pj} = 1 \text{ for } j = 1,...,n \text{ and } s_{pj} \geq 0 \text{ for } p = 1,...,P \text{ and } j = 1,...,N$$

$\underline{r}_j \equiv$ random vector representing additive Gaussian sensor noise with a mean of zero for the jth pixel vector observation

Note that in the case where a single pixel vector, $\underline{x}_j$, is an actual endmember, i.e. pure pixel, s_{pj} would be zero for all p except for the endmember occupying that particular

observation. Further, note the sum-to-one and non-negativity constraints on the abundance fractions.

Because of the random nature of $\underline{r}$, each measured $\underline{x}$ should be considered a realization of a random vector process. It may also be appropriate to consider the abundance fractions as a random component, but an inherent assumption in the LMM is that the endmember spectra, $\underline{\varepsilon}_P$, are deterministic (Chang, 2007:110).

If (2.1) is expressed in terms of an entire data matrix, X, where each column is a distinct observation of a pixel vector, one has:

$$
\begin{array}{c}
\begin{pmatrix}
x_{11} & x_{12} & \cdots & x_{1N} \\
x_{21} & x_{22} & & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
x_{K1} & x_{K2} & \cdots & x_{KN}
\end{pmatrix} =
\begin{pmatrix}
\varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1P} \\
\varepsilon_{21} & \varepsilon_{22} & & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\varepsilon_{K1} & \varepsilon_{K2} & \cdots & \varepsilon_{KP}
\end{pmatrix} \cdot
\begin{pmatrix}
s_{11} & s_{12} & \cdots & s_{1N} \\
s_{21} & s_{22} & & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
s_{P1} & s_{P2} & \cdots & s_{PN}
\end{pmatrix} + R_{KxN} \\
\quad \underline{x}_1 \quad \underline{x}_2 \qquad \underline{x}_N \qquad \underline{\varepsilon}_1 \quad \underline{\varepsilon}_2 \qquad \underline{\varepsilon}_P
\end{array}
$$

(2.2)

or

$$ X_{KxN} = E_{KxP} \cdot S_{PxN} + R_{KxN} $$

Notice that the elements of the jth column of S_{PxN}, represent the separate fractional abundances of the P endmembers signatures , $\underline{\varepsilon}_1, \underline{\varepsilon}_2, ..., \underline{\varepsilon}_P$ (column vectors of E), in observation j. Furthermore, the elements of pth row of S_{PxN}, represent the abundance fractions of a single endmember, $\underline{\varepsilon}_p$, in each of N observations. Thus, if one where to plot the values of the pth row of S_{PxN} (which is termed an abundance map) and the signal shows 'high' abundance values for observations, say observation 25 through 35, as

compared to the values for the other observations, then one could conclude that those pixel observations have a high concentration of the pth endmember. Each abundance row vector can be considered a feature in the image.

Another important fact to consider for the LMM is that the model is an additive only model. Given that E_{KxP} is matrix of endmember signatures where each element is a reflectance value (nonnegative) in a particular spectral band and the abundance matrix, S_{PxN}, contains nonnegative fractions, the LMM is a purely additive model (allowing no subtractions).

2.1.1 Abundance Maps and Target Detection

A Priori Knowledge of Endmember Matrix

If the endmember matrix E (i.e. total number of endmembers in the image and their respective spectral signatures) is known a priori then the problem is to solve for the abundance matrix, S_{PxN}, to unmix the hyperspectral image. The abundance matrix is generally solved using constrained least squares methods. The estimated abundance matrix row vectors recovered are usually formed into images and are termed the abundance maps, as described earlier, for each respective endmember (Chang, 2007:111). The abundance matrix row vector is simply reshaped into the original image pixel length by width and plotted on a grey scale (scaled between 0 and 1) to create an abundance map. Analysis of these abundance maps is the key to identifying targets of interest. Pixels with high abundance values in abundance map p corresponding to endmember p

(which is a signature of a particular target of interest or desirable endmember) identify
the location of that particular target. Figure 2-1 is an example of such an abundance map.

Figure 2-1. Abundance Map for Tank Endmember

No A Priori Knowledge of Endmember Matrix

Signature Matching

Of course, if one has no a priori knowledge of the endmember matrix, the
problem of solving for the endmember matrix and abundance matrix (termed unmixing)
in hyperspectral images is much harder. One must determine both the endmember matrix
and the abundance matrix from knowledge of just the observed matrix X. Such a problem
falls into the category of a much more general problem referred to as blind source
separation (BSS). The term blind indicates that little or no information is available on
either of the matrices that multiply to form the observed matrix X (Varshney and Arora,
2004:110). Two particular solutions applied to the problem of BSS, independent
component analysis (ICA) and nonnegative matrix factorization can be used to solve for
both matrices. One particular target identification algorithm, ICA-EEA (ICA-

Endmember Extraction Algorithm), performs the ICA algorithm to solve the BSS problem, unmixing the hyperspectral data set. Target candidate pixels are identified from each abundance map by choosing the pixel with the highest absolute abundance value from each map. These pixels are hypothesized to represent pure endmembers. The signatures of these pure endmembers, which are obtained from the corresponding pixel vectors from the original unmixed data matrix, X, are compared to the material signatures of targets of interest to identify which abundance maps may contain those targets of interest (Wang and Chang, 2006:623314-3).

Anomaly Detection

In the area of anomaly detection, one does not even know the signatures of the targets of interest and wishes to identify targets in terms of just anomalous occurrences in the overall image. The hypothesis is that target materials are relatively rare, producing abundance maps with relatively few intense pixels (Smetek, 2007:46). Abundance maps corresponding to these target materials will have a relatively constant dark background (representing very low values of abundance) with the few target pixels highlighted brightly (representing high abundance values). Such an abundance map represents a scenario with small objects and a large homogeneous background. In such a map, if one were to take the kurtosis value of the abundance vector that was used to form the map, the value would be high compared to an abundance map representing a large class, like for example a large sections of trees. Thus, one could select the maps with the highest kurtosis as the one containing information about small objects. Robila and Varshney make a determination of the number of maps to retain based on a scree graph of the kurtosis values. They state that a considerable slope change represents the border

between one class of maps, in our case target classes, to another, non-target classes (Robila and Varshney, 2002:177). An example of this small class (i.e. target) feature selection via the scree graph of kurtosis values will be presented later in this chapter in section 2.4. Once the frames with the highest kurtosis are selected, those pixels with abundance values higher than a determined threshold will be classified as targets in the target identification phase.

2.1.2 Dimensionality Reduction (Feature Extraction)

Spectral unmixing is usually performed within a subspace of the original K spectral bands. It has been proven that high-dimensional space is mostly empty, and multivariate data are usually in a lower dimensional structure. As a result, for any particular hyperspectral image, this K-dimensional data can be projected to a lower dimensional subspace without losing significant information and class separability (Landgrebe, 2003:270). This process is referred to as feature extraction. The features are the new variables that define the dimensions of the subspace. The focus is to increase the separation between classes within each feature (i.e. find projections where the new variables are uncorrelated) while reducing overall noise (Robila and Maciack, 2006:2). Further, since the number of endmembers, P, is not known, the dimension of the subspace is considered to be the number of distinct endmembers for the given image (Change, 2007:112). Determination of the number of endmembers via a dimensionality reduction technique called principal components analysis will be discussed next.

2.1.3 Principal Components Analysis (PCA)

PCA linearly transforms data into a subspace where the new variables (components) are decorrelated and ranked in decreasing order of variance. Consider a random hyperspectral pixel vector, $\underline{x}^T = (x_1, ..., x_K)$ in K spectral bands. To find the first principal component, one must find a projection of this vector

$$y_{(1)} = \underline{g}^T_{(1)} \underline{x} = g_{11}x_1 + g_{21}x_2 + ... + g_{K1}x_K \tag{2.3}$$

such that the variance of the projection, $\text{cov}(y_{(1)})$, is maximized subject to the projection, $\underline{g}_{(1)}$, being of unit norm.

$$\begin{array}{c} \text{MAX} \\ \underline{g}_{(1)} \end{array} \text{cov}\left(\underline{g}^T_{(1)}\underline{x}\right) = \underline{g}^T_{(1)} \Sigma_x \underline{g}_{(1)} \quad (\text{note: } \Sigma_x = \text{cov}(\underline{x})) \tag{2.4}$$

$$\text{s.t.} \quad \underline{g}^T_{(1)}\underline{g}_{(1)} = 1$$

If Σ_x positive definite, then to find $\underline{g}_{(1)}$ one must satisfy only the necessary condition, where the gradient of the Lagrangian is zero. The Lagrangian of (2.4) and gradient of the Lagrangian will be

$$L\left(\underline{g}_{(1)}, \lambda_{(1)}\right) = \underline{g}^T_{(1)} \Sigma_x \underline{g}_{(1)} - \lambda_{(1)}\left(\underline{g}^T_{(1)}\underline{g}_{(1)} - 1\right) \tag{2.5}$$

$$\nabla L = \begin{pmatrix} 2\Sigma_x\underline{g}_{(1)} - 2\lambda_{(1)}\underline{g}_{(1)} \\ \underline{g}^T_{(1)}\underline{g}_{(1)} - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.6}$$

Note that

$$2\Sigma_x\underline{g}_{(1)} - 2\lambda_{(1)}\underline{g}_{(1)} = 0 \Rightarrow \left(\Sigma_x - \lambda_{(1)}I\right)\underline{g}_{(1)} = 0 \tag{2.7}$$

In order to have a solution for $\underline{g}_{(1)}$ which is not the zero vector, one must have

$$\left|\Sigma_{\underline{x}} - \lambda_{(1)} I\right| = 0 \tag{2.8}$$

Thus, the solution, $\lambda_{(1)}$, for (2.8) is the largest eigenvalue of $\Sigma_{\underline{x}}$ which means $\underline{g}_{(1)}$ is the

normalized eigenvector associated with $\lambda_{(1)}$. The second principal component

$$y_{(2)} = \underline{g}^T_{(2)}\underline{x} = g_{12}x_1 + g_{22}x_2 + \ldots + g_{K2}x_K \tag{2.9}$$

is found by solving (2.4) with the additional constraint, $\underline{g}^T_{(1)}\underline{g}_{(2)} = 0$. Using the same

argument above it can be shown that $\underline{g}_{(2)}$ is the normalized eigenvector associated with

the second largest eigenvalue of $\Sigma_{\underline{x}}$, $\lambda_{(2)}$. The process is repeated until all K normalized

eigenvectors of $\Sigma_{\underline{x}}$ are found where each eigenvector is orthogonal to every other

eigenvector (Dillon and Goldstein, 1984:28). Now by definition of eigenvalues:

$$\Sigma_{\underline{x}}\underline{g}_{(1)} = \lambda_{(1)}\underline{g}_{(1)}$$
$$\Sigma_{\underline{x}}\underline{g}_{(2)} = \lambda_{(2)}\underline{g}_{(2)}$$
$$\vdots$$
$$\Sigma_{\underline{x}}\underline{g}_{(K)} = \lambda_{(K)}\underline{g}_{(K)}$$

$$\Rightarrow \left[\Sigma_{\underline{x}}\underline{g}_{(1)} \vdots \Sigma_{\underline{x}}\underline{g}_{(2)} \vdots \ldots \vdots \Sigma_{\underline{x}}\underline{g}_{(K)}\right] = \left[\lambda_{(1)}\underline{g}_{(1)} \vdots \lambda_{(2)}\underline{g}_{(2)} \vdots \ldots \vdots \lambda_{(K)}\underline{g}_{(K)}\right]$$

$$\Rightarrow \Sigma_{\underline{x}}\left[\underline{g}_{(1)} \vdots \underline{g}_{(2)} \vdots \ldots \vdots \underline{g}_{(K)}\right] = \left[\underline{g}_{(1)} \vdots \underline{g}_{(2)} \vdots \ldots \vdots \underline{g}_{(K)}\right]D, \text{ where } D = \begin{pmatrix} \lambda_{(1)} & & & \\ & \cdot & & \\ & & \cdot & \\ & & & \lambda_{(K)} \end{pmatrix}$$

$$\Rightarrow \Sigma_{\underline{x}} \cdot G = G \cdot D \tag{2.10}$$

Because G is a matrix of unit norm eigenvectors of a covariance matrix, G is orthonormal

and $G^T \cdot G = I$. Thus, left multiplying (2.10) by G^T, one has the result:

$$G^T \cdot \Sigma_{\underline{x}} \cdot G = D \qquad (2.11)$$

(2.11) is significant because the goal is to have new variables, $\left(y_{(1)}, y_{(2)}, ..., y_{(K)} \right) = \underline{y}\,'$,

that are uncorrelated. Notice

$$\text{cov}\left(\underline{y} \right) = \text{cov}\left(G^T \underline{x} \right) = G^T \cdot \Sigma_{\underline{x}} \cdot G = D \qquad (2.12)$$

Since D is a diagonal matrix the new variables (components) are uncorrelated with

the $\text{var}\left(y_{(i)} \right) = \lambda_i$, for $i = 1, ..., K$.

Now to address the issue of dimensionality reduction, the goal is to retain enough

of the $y_{(i)}$'s so that amount of variability explained is enough to preserve the significant

information and class separability as was explained previously in the citation from

Landgrebe. The total variance of the hyperspectral image can be defined:

$$\text{Total Variance} \equiv \text{trace}\left(\Sigma_{\underline{x}} \right) = \sum_{i=1}^{K} \sigma_{ii}^{\;2} \qquad (2.13)$$

Right multiplying (2.10) by G'

$$\Sigma_{\underline{x}} = G \cdot D \cdot G^T \qquad (2.14)$$

So

$$\sum_{i=1}^{K} \sigma_{ii}^{\;2} = \text{trace}\left(\Sigma_{\underline{x}} \right) = \text{trace}\left(G \cdot D \cdot G^T \right) = \text{trace}\left(G^T \cdot G \cdot D \right) = \text{trace}\left(D \right) = \sum_{i=1}^{K} \lambda_i \quad (2.15)$$

Thus,

$$\text{Total Variance} \equiv \sum_{i=1}^{K} \lambda_i \qquad (2.16)$$

Thus, (2.13) and (2.16) give the important result that the total variance of the original variables is equal to the sum of the variances for all K principal components (Dillon and Goldstein, 1984:29).

Determining Dimensionality

One way to determine what the reduced dimensionality of the image is to calculate a selected subset of components' percentage of variance explained. An analysis of their corresponding eigenvalues yields:

$$\frac{\sum_{i=1}^{P} \lambda_i}{\sum_{i=1}^{K} \lambda_i} = \% \text{ of variance explained by selected subspace} \qquad (2.17)$$

Robila and Varshney choose 99.9% in their cited research.

Another way to determine the dimensionality of the subspace comes from the assertion that covariance of the random hyperspectral pixel vector is made of the true signal covariance and noise. The true signal covariance is contributed by the true number, P, of distinct endmember signatures in the image. Beyond the Pth eigenvalue of $\Sigma_{\underline{x}}$, the eigenvalue distribution becomes constant with the value of the constant eigenvalues equal to the covariance of the noise term, $\underline{r}$ in (2.1), $\sigma^2 I$, i.e. white noise (Chang, 2007:112; Hyvärinen and others, 2001:131). As reproduced from Stocker et al., Figure 2-2 (a) shows a plot of eigenvalues from simulated data with 64 bands and 8 signals, i.e. a true dimensionality of 8, with added white noise of variance 1. Notice the constant floor at 1, σ^2, in Figure 2-2 (a). However in Figure 2-2 (b), due to finite-sample estimation errors, instead of a constant eigenvalue noise floor at σ^2, one can view a 'tilted

ramp' in the general vicinity of σ^2. The distribution of the noise eigenvalues follow approximately an asymptotic distribution called Silverstein's distribution, provided the eigenvalues are from a white noise covariance matrix formed from independently identically distributed noise vectors (Stocker and others, 2003: 654). If the covariance of the noise term, $\underline{r}$, is not white, i.e. $\text{cov}(\underline{r}) = \left[\sigma_1^2, \sigma_2^2, ..., \sigma_K^2 \right] \cdot I$, as depicted in Figure 2-2 (b), this nonwhite sensor noise added to the same signals imparts additional tilt into the sample eigenvalue curve (Stocker and others, 2003: 652). Fitting the white noise eigenvalues to the Silverstein distribution approximately locates the 'knee' in the eigenvalue curve that separates the signal and noise eigenvalues as shown in Figure 2-2 (c). However, for many hyperspectral sensors, noise is not white across the spectral range of the instrument, and no analogous closed form expression for the asymptotic eigenvalue density exists in the nonwhite noise case (Stocker and others, 2003: 657). Thus, locating the 'knee' is a more difficult problem due to the absence of a closed form theoretical distribution and requires some estimation of the distribution of the nonwhite noise eigenvalues. This is accomplished via an estimate of the spectrally varying noise level $\left[\sigma_1^2, \sigma_2^2, ..., \sigma_K^2 \right]$ extracted from scene data. Stocker et al. provide a parametric in-scene approximation to the spectral nonwhite noise characteristic to approximate $\left[\sigma_1^2, \sigma_2^2, ..., \sigma_K^2 \right]$ (Stocker and others, 2003: 657-664).

By locating the breakpoint in the eigenvalue curve between signal and noise of the covariance matrix of the data, and discarding those eigenvectors associated with the eigenvalues corresponding to noise, one can improve the signal to noise ratio of the data in the PCA space.
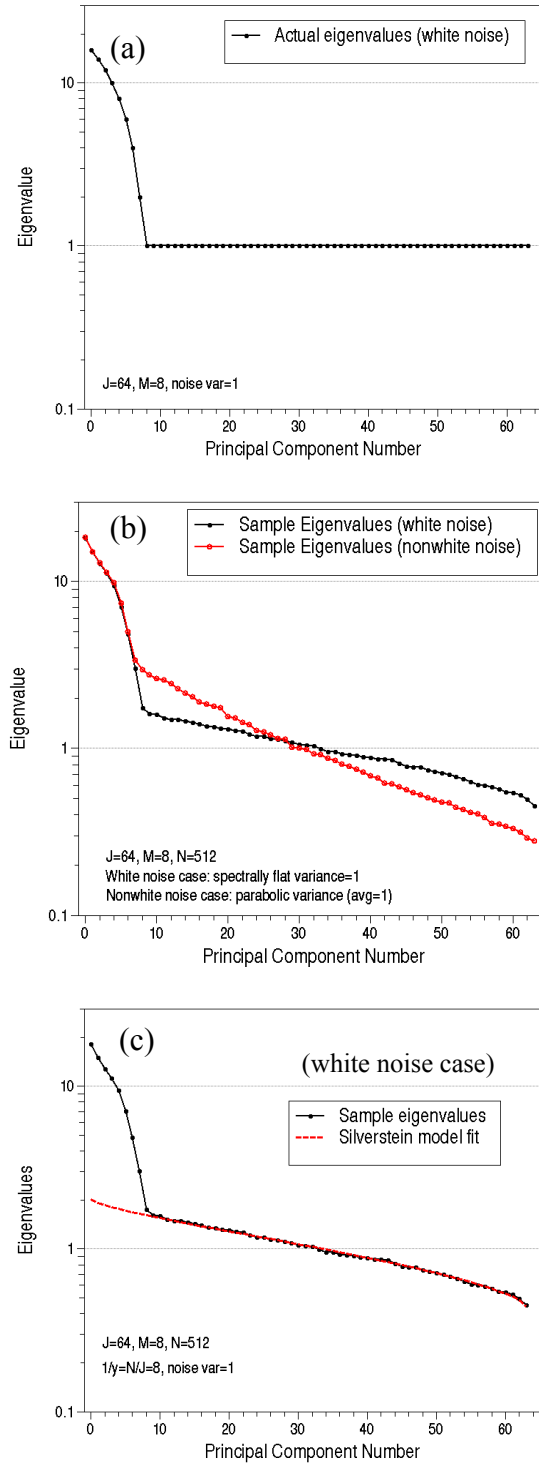
Figure 2-2. **(a) True eigenvalues, (b) sample eigenvalues, and (c) Silverstein model fit for simulated 64-band data with 8 signal modes and additive sensor noise (Stocker and others, 2003: 653, 657)**

Thus, after some decision is made on the number principal components to retain denoted by P, our original K-dimensional hyperspectral image is projected onto a P-dimensional orthogonal subspace by a linear transformation matrix defined by selecting the P eigenvectors (where P represents the number of retained eigenvectors) associated with the P eigenvalues in (2.17) and discarding the rest of the eigenvectors. This matrix of retained eigenvectors is applied to X shown in (2.18) below,

$$Y_{PxN} = \underbrace{\begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1K} \\ g_{21} & g_{22} & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ g_{P1} & g_{P2} & \cdots & g_{PK} \end{pmatrix}}_{G^T_{PxK}} \cdot \underbrace{\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x_{K1} & x_{K2} & \cdots & x_{KN} \end{pmatrix}}_{X_{KxN}} \tag{2.18}$$

where the pth row of G^T represents the pth retained eigenvector of Σ_x.

Recall the LMM presented in (2.2). After finding the lower dimensional subspace, the new form of the LMM model is

$$\begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ y_{P1} & y_{P2} & \cdots & y_{PN} \end{pmatrix} = \begin{pmatrix} \varepsilon'_{11} & \varepsilon'_{12} & \cdots & \varepsilon'_{1P} \\ \varepsilon'_{21} & \varepsilon'_{22} & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \varepsilon'_{P1} & \varepsilon'_{P2} & \cdots & \varepsilon'_{PP} \end{pmatrix} \cdot \begin{pmatrix} s'_{11} & s'_{12} & \cdots & s'_{1N} \\ s'_{21} & s'_{22} & & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ s'_{P1} & s'_{P2} & \cdots & s'_{PN} \end{pmatrix} + R'_{PxN}$$

$$or \tag{2.19}$$

$$Y_{PxN} = E'_{PxP} \cdot S'_{PxN} + R'_{PxN}$$

Notice the endmember matrix in (2.19) to be determined is now a square matrix which is advantageous when employing methods to solve for the endmember matrix and abundance matrix simultaneously. It is important to note that, whereas the jth column of X represented the jth pixel's signature across the K spectral bands, the jth column of Y in

(2.19) represents the jth pixel's signature across the P principal components. Thus, when solving for the endmember matrix in (2.19), the columns of this new square endmember matrix denoted by E' instead of E represent the signatures of the endmembers in the principal component space not signatures of the endmembers in terms of reflectance of the original K spectral bands. Thus, one loses the physical interpretability of the solved endmember matrix in this new space. Further, given that the principal component space is not nonnegative as is the case in the spectral reflectance space, the Y matrix and the E' matrix will have negative terms in (2.19) and thus no longer strictly conforms to the additive only LMM in (2.1) since subtractions will occur. However, based on this author's review of the myriad of literature available on dimensionality reduction of hyperspectral images, PCA is the most common tool used to effectively reduce the dimensionality of a hyperspectral image. Another approach called nonnegative PCA is suggested by several sources as a reduction technique that produces nonnegative components. This technique should be investigated and compared to PCA to see if the additional nonnegative restriction outperforms target detection algorithms using traditional PCA image compression as a preprocessing technique. However, this thesis due to its limited scope will employ traditional PCA.

2.1.4 Limitations of the LMM

Recall that a basic assumption in the LMM is that the endmembers are deterministic, i.e. within an endmember, the signature has no variability. Further, the assertion is that the mixtures of a small number of deterministic spectra (endmembers)

can be used to represent all of the non-noise variance in a hyperspectral image. However, there is natural signature variation to almost all materials that would be selected as endmembers in a hyperspectral image. Also, practically, there is a limit to how many endmembers can be used to represent a scene. Thus, any particular endmember may not necessarily represent just one material but a class of materials that are spectrally similar. Thus, the spectral signature of the endmember will have variation. Further, another source of endmember variation is illumination. A particular endmember in shade will have a different signature than the same endmember exposed to direct sunlight. Thus, the endmembers are not truly deterministic signature vectors, but random vectors. However, when the within endmember variance is small compared to the between endmember variance, the deterministic endmember assumption may remain relatively valid. The other assumption of linearity of the mixing fails if

> there is significant three-dimensional structure within a given pixel and where the optical energy makes multiple bounces between objects before exiting in the direction of the sensor (Chang, 2007:27).

However, if the mixing scale at each pixel is macroscopic, the linear model holds (Chang, 2007:149). Next, a solution to the LMM called independent component analysis will be presented followed by a discussion of how it approximately fits into the LMM of HSI. When discussing independent component analysis it is assumed we are still operating in the reduced PCA space.

2.2 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) introduced in the early 1980's is a multivariate data analysis method where, given a linear mixture of statistically

independent components, these components are recovered by solving for an unmixing matrix (Varshney and Arora, 2004:109). Whereas PCA finds the transform of the observed data that decorrelates the observed variables through the use of second-order statistics (i.e. a transform based on the eigenvectors of the covariance matrix), ICA utilizes higher-order statistics to find projections of the data where the components are independent, a stronger statement than uncorrelated. Throughout the past two decades from the initial development the ICA technique in the early 1980s, several solutions to the ICA problem have been presented. This thesis will focus on a computationally efficient solution, FastICA, developed by A. Hyvärinen, J. Karhunen, and E. Oja from the Neural Networks Research Centre at the Helsinki University of Technology, Finland. Their solution finds projections of the original data that maximize their nongaussianity. The following sections will formally develop the ICA solution to the BSS problem and justify how maximizing the component's nongaussianity is equivalent to minimizing their mutual information, a measure of the dependence of the components. The formal development of ICA in the subsequent sections follows the development given by A. Hyvärinen, J. Karhunen, and E. Oja in their text, Independent Component Analysis, copyright 2001. Any proofs of the theoretical results not provided by those authors will be provided by the author of this thesis and marked as such.

2.2.1 Formal Definition of ICA

Consider a situation in which one observes p random variables, $x_1, x_2, ..., x_P$, and these random variables are linear combinations of p be independent source signals (components), $s_1, s_2, ..., s_P$. Thus, we have:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jP}s_P \text{ for } j = 1, 2, \ldots, P \tag{2.20}$$

In matrix notation:

$$\underline{x} = A\underline{s} \tag{2.21}$$

where

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ . \\ x_P \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1p} \\ a_{21} & a_{22} & & . \\ . & . & . & . \\ a_{P1} & a_{P2} & \ldots & a_{PP} \end{pmatrix}, \quad \underline{s} = \begin{pmatrix} s_1 \\ s_2 \\ . \\ s_P \end{pmatrix}$$

Note that $\underline{x}$ is a random vector and $\underline{x}_i = A\underline{s}_i$ denotes a realization of that random vector, where $i = 1, 2, \ldots, N$ observations. Further, we can only observe realizations of the random vector $\underline{x}$, the mixed signals. The independent components which are random variables making up the vector, $\underline{s}$, are referred to as latent variables, since they cannot be directly observed (Hyvärinen and others, 2001:151). Further the mixing matrix, A, is also unknown. The model can also be written as:

$$\begin{pmatrix} x_1 \\ x_2 \\ . \\ x_P \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} \\ a_{21} \\ . \\ a_{P1} \end{pmatrix}}_{\underline{a}_1} s_1 + \underbrace{\begin{pmatrix} a_{12} \\ a_{22} \\ . \\ a_{P2} \end{pmatrix}}_{\underline{a}_2} s_2 + \ldots + \underbrace{\begin{pmatrix} a_{1P} \\ a_{2P} \\ . \\ a_{PP} \end{pmatrix}}_{\underline{a}_P} s_P \tag{2.22}$$

or

$$\underline{x} = \sum_{j=1}^{P} \underline{a}_j s_j$$

Finally, in terms of a sample data matrix of N observations ($N >> P$) where each column of X is a realization of the random vector $\underline{x}$ and each column of S is a realization of the random vector $\underline{s}$ one has

$$X_{PxN} = A_{PxP} S_{PxN} \qquad (2.23)$$

where

$$X_{PxN} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \cdot & \cdot & & \cdot \\ x_{P1} & x_{P2} & \cdots & x_{PN} \end{bmatrix}, \; S_{PxN} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \cdot & \cdot & & \cdot \\ s_{P1} & s_{P2} & \cdots & s_{PN} \end{bmatrix}$$

2.2.2 Assumptions of ICA

In order for the ICA model to hold, three assumptions must be true. The first assumption states that the independent components (ICs) one is estimating must in fact be statistically independent. Thus, for any of the elements $s_i, s_j \, (i \neq j)$ in the random vector $\underline{s}$ the value of s_i does not give any information on value of s_j. Formally, s_i, s_j are independent for all $(i \neq j)$ if and only if $f\left(s_1, s_2, ..., s_p\right) = f_1\left(s_1\right) f_2\left(s_2\right) ... f_p\left(s_p\right)$, i.e. the joint density of the components is equal to the product of the marginal densities.

The second assumption states that the ICs must have nongaussian distributions. Higher-order cumulants (e.g. skewness and kurtosis) are zero for Gaussian distributions, but this higher order information is essential for estimation of the ICA model. This assertion will be illustrated in a later section.

The third assumption is that the mixing matrix $\underline{A}$ is square and invertible. The solution strategy is to find the projection of the data that maximizes nongaussianity.

$$\underline{s} = W\underline{x} \tag{2.24}$$

This projection will be defined by the matrix, W. Thus, to recover the original mixing matrix, one would need to simply compute the inverse of W.

$$W^{-1}\underline{s} = \underline{x} \Rightarrow A = W^{-1} \tag{2.25}$$

Thus, a square, invertible mixing matrix is needed.

It is also convenient to assume that the observed data and the components have zero mean. If this is not the case, one can merely center the data by subtracting the mean (Hyvärinen and others, 2001:152-154). Note that centering the linearly mixed data results in the ICs also having zero mean.

2.2.3 Ambiguities of ICA

In the ICA model, two ambiguities exist. The first ambiguity is that the variances of the ICs cannot be determined. Referring to (2.22), this stems from the fact that any scalar multiplier α_j to one of the sources s_i could always be canceled by dividing the corresponding column $\underline{a}_{\bullet j}$ of A by the same scalar α_j.

$$\underline{x} = \sum_{j=1}^{p} \left(\frac{1}{\alpha_j} \underline{a}_j \right) \left(s_j \alpha_j \right) \tag{2.26}$$

Thus, one could just fix the magnitudes of the ICs. Since the components are random variables, one can assume that each has unit variance: $E\left(s_i^2\right) = 1$. In order to enforce unit variance on the ICs, the matrix $\underline{A}$ will be adapted to account for this restriction. This

adaptation will be explained in a later section. There is also the ambiguity of the sign of an independent component. The same argument for this ambiguity can be made from (2.26).

The second ambiguity is that the order of the components cannot be determined (Hyvärinen and others, 2001:154).

2.2.4 Example Problem of ICA

The following example is presented by A. Hyvärinen and E. Oja in 2000 in an article published in Neural Networks, volume 13, titled *Independent component analysis: algorithms and applications* covered on pages 414 through 415. The data used to replicate the figures were created by this author in Excel.

Let s_1 and s_2 be two ICs with the following marginal uniform PDF:

$$f_i(s_i) = \begin{cases} \dfrac{1}{2\sqrt{3}} & \text{if } -\sqrt{3} \leq s_i \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases} \tag{2.27}$$

Note that $E(s_i) = \displaystyle\int_{-\sqrt{3}}^{\sqrt{3}} s_i \frac{1}{2\sqrt{3}} ds_i = 0$ and $E\{s_i^2 - E\{s_i\}^2\} = E\{s_i^2\} = \displaystyle\int_{-\sqrt{3}}^{\sqrt{3}} s_i^2 \frac{1}{2\sqrt{3}} = 1$

So the mean and the variance of s_i are zero and one respectively conforming to the assumptions in the ICA model. Since s_1 and s_2 are independent

$$f(s_1, s_2) = f_1(s_1) \cdot f_2(s_2) = \begin{cases} \dfrac{1}{12} & \text{if } -\sqrt{3} \leq s_i \leq \sqrt{3} \\ 0 & \text{otherwise} \end{cases} \tag{2.28}$$

A depiction of the joint distribution of s_1, s_2, given 5000 samples generated in Excel, is shown below in Figure 2-3.
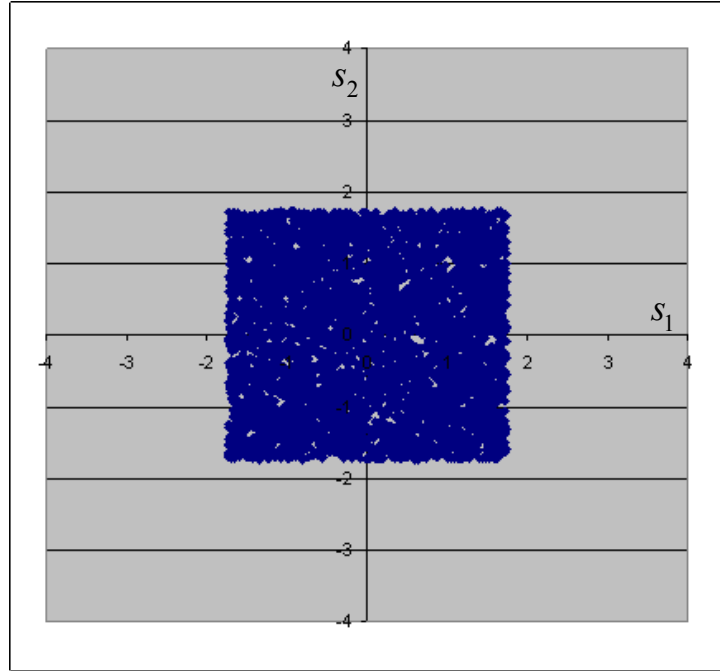
Figure 2-3. Joint Distribution of s_1 and s_2

Let $A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$ be the matrix to mix the two ICs. So $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$. Given we

have 5000 sample data points, the observed mixed matrix would be $X_{2x5000} = A \cdot S_{2x5000}$.

A depiction of the joint distribution of the observables x_1, x_2 is below in Figure 2-4:
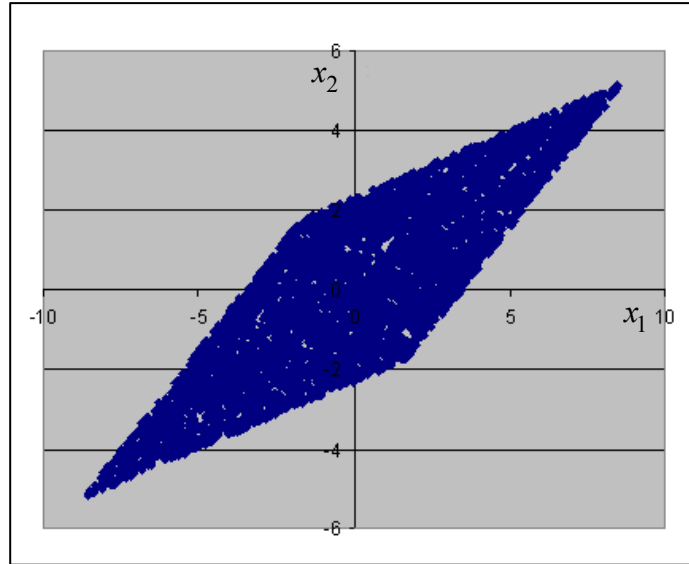
Figure 2-4. Joint Distribution of x_1 and x_2

Notice that the edges of the joint distribution of x_1 and x_2 in Figure 2-5 below are in the same direction as the columns of the mixing matrix A. One way to solve the ICA problem would be to estimate the joint density of the observables and then find the direction of the edges to estimate the mixing matrix A and thus solve for the ICs. However, this method would be computationally expensive. Further, the edges may not be as clearly defined as they are in a uniform distribution. For most distributions such edges cannot be found (Hyvärinen and others, 2001:156). A method that can compute the A matrix in a reliable and efficient manner is needed.
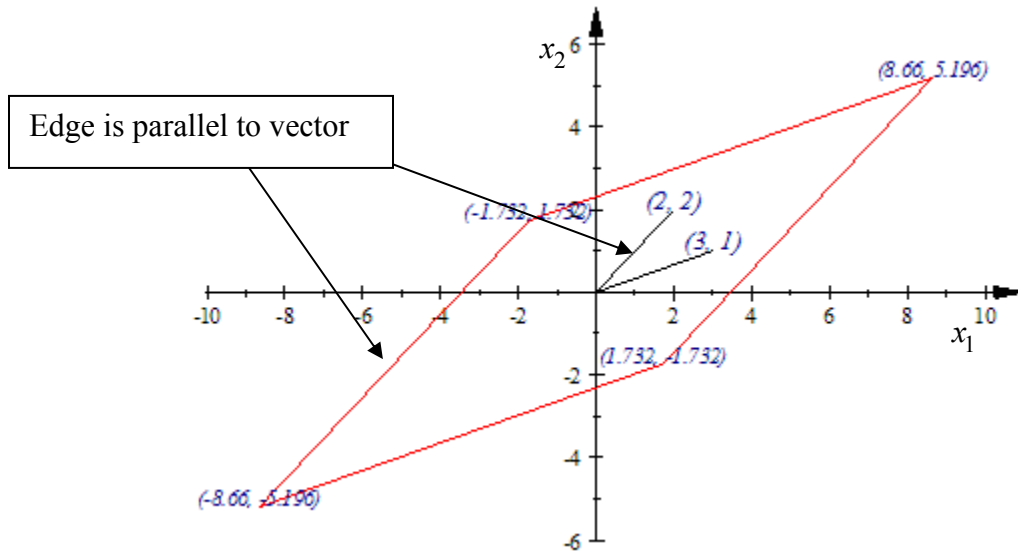
Figure 2-5. Edges of Joint Distribution of x_1 and x_2

2.2.5 Whitening Data

The term whitening, refers to centering sample data, decorrelating the variables, and scaling them to have unit variance. Note the only difference whitening has from PCA is that whitening in addition to decorrelating the data, scales the data. Thus, first and second order information (mean and variance) is removed from the data. The first step in the whitening processes is to center the data matrix, i.e. make the mean vector the zero vector. To center a sample data matrix, one must first find the mean vector of the data

$$\underline{m}_{px1} = n^{-1} \cdot X_{pxn} \cdot \underline{1}_{nx1} \tag{2.29}$$

where $\underline{1}_{nx1}$ is just a column vector of ones of size equal to the number of observations n. After finding the mean vector $\underline{m}_{px1}$, the following matrix operation subtracts the mean vector from every observation:

$$X_{centered} = X_{pxn} - \underline{m}_{px1} \cdot \left(\underline{1}_{nx1}\right)^T \tag{2.30}$$

From this point forward in this thesis, for any reference to a sample data matrix X, it is assumed that the matrix has been centered.

The second step is to find a transformation matrix V to decorrelate and scale the random vector $\underline{x}$ to have unit variance generating the 'whitened' random vector, $\underline{x}_w$. Thus,

$$\underline{x}_w = V\underline{x} \tag{2.31}$$

such that

$$Cov\left(\underline{x}_w\right) = \Sigma_{\underline{x}_w} = E\left(\underline{x}_w \cdot \underline{x}^T_{\ w}\right) = I_{pxp} \tag{2.32}$$

One way to compute V is to consider the eigenvalue decomposition of the covariance matrix , $\Sigma_{\underline{x}}$, presented in (2.14). Let V be the inverse square root of $\Sigma_{\underline{x}}$ in (2.14):

$$V = \Sigma_x^{-\frac{1}{2}} = G \cdot D^{-\frac{1}{2}} \cdot G^T \tag{2.33}$$

Thus,

$$\begin{aligned}
\Sigma_{\underline{x}_w} &= E\left(\underline{x}_w \cdot \underline{x}^T_{\ w}\right) = E\left(V\underline{x} \cdot \underline{x}^T V^T\right) \\
&= V \cdot E\left(\underline{x} \cdot \underline{x}^T\right) \cdot V^T = V \cdot \Sigma_{\underline{x}} \cdot V^T \\
&= \left(G \cdot D^{-\frac{1}{2}} \cdot G^T\right)\left(G \cdot D \cdot G^T\right)\left(G \cdot D^{-\frac{1}{2}} \cdot G^T\right)^T \\
&= G \cdot D^{-\frac{1}{2}} \cdot G^T \cdot G \cdot D \cdot G^T \cdot G \cdot D^{-\frac{1}{2}} \cdot G^T \\
&= G \cdot D^{-\frac{1}{2}} \cdot D \cdot D^{-\frac{1}{2}} \cdot G^T = G \cdot G^T = I \\
&\Rightarrow \Sigma_{\underline{x}_w} = I
\end{aligned} \tag{2.34}$$

So this choice of V whitens our centered random vector. The new form of the ICA model with a whitened sample data matrix is

$$X_{white} = V \cdot X = V(A \cdot S) = (V \cdot A) \cdot S = \tilde{A} \cdot S \qquad (2.35)$$

Recall that since the variances of the ICs are unknown, as explained earlier, the ICs are assumed to have unit variance (as a matter on convenience) and the mixing matrix would be adapted to enforce this assumption. A. Hyvärinen, J. Karhunen, and E. Oja prove in their text, Independent Component Analysis, that if one assumes that the components are of unit variance and given the data has been whitened, this implies that the mixing matrix is orthonormal. The proof below provided by the author of this thesis is similar to their proof. However, this author comes from the perspective that, one does not know that the components have unit variance. As stated in the ambiguities of ICA, another perspective is that one is adapting the mixing matrix to enforce this restriction. Unit variance is not known a priori, it is merely a convenient restriction enforced on the solved components. This author proves the converse of what the aforementioned authors proved in (2.36), i.e., assuming the mixing matrix is orthonormal and given the data has been whitened, this imposes unit variance and decorrelation of the components.

$$
\begin{aligned}
&\text{Given } \Sigma_{\underline{x}_w} = E\left(\underline{x}_w \cdot \underline{x}^T_{\ w}\right) = I \text{ from (2.34) and } \tilde{A}^T \cdot \tilde{A} = I \\
&\Rightarrow E\left(\underline{x}_w \cdot \underline{x}^T_{\ w}\right) = E\left(\tilde{A} \cdot \underline{s} \cdot \underline{s}^T \cdot \tilde{A}^T\right) = \tilde{A} \cdot E\left(\underline{s} \cdot \underline{s}^T\right) \cdot \tilde{A}^T = I \\
&\text{Now, } \tilde{A} \cdot E\left(\underline{s} \cdot \underline{s}^T\right) \cdot \tilde{A}^T = I \Rightarrow E\left(\underline{s} \cdot \underline{s}^T\right) = \tilde{A}^T \cdot I \cdot \tilde{A} = I \\
&\Rightarrow \text{Cov}\left(\underline{s}\right) = I
\end{aligned}
\qquad (2.36)
$$

If the mixing matrix $\tilde{A}$ were not orthonormal, one would need to estimate all the p^2 parameters of the mixing matrix. However, by whitening the data, the new mixing

matrix is assumed to be orthonormal and thus, contains only $\dfrac{p(p-1)}{2}$ degrees of freedom (Hyvärinen and others, 2001:160).

Recall our previous example of the linearly mixed data matrix X_{2x5000} whose joint distribution is depicted in Figure 3. If this matrix is whitened using the procedure just described one has

$$C_X = \begin{bmatrix} 13.3197 & 7.1298 \\ 7.1298 & 5.0907 \end{bmatrix}, G = \begin{bmatrix} -0.866 & 0.5001 \\ -0.5001 & -0.866 \end{bmatrix}, \text{ and } D = \begin{bmatrix} 17.4371 & 0 \\ 0 & 0.9733 \end{bmatrix} \quad (2.37)$$

where C_X is the sample covariance of X_{2x5000}, and G and D contain the eigenvectors (in columns) and eigenvalues of C_X respectively. So,

$$V = G \cdot D^{-\frac{1}{2}} \cdot G^T = \begin{bmatrix} 0.4331 & -0.3352 \\ -0.3352 & 0.82 \end{bmatrix} \quad (2.38)$$

Therefore,

$$X_w = \begin{bmatrix} 0.4331 & -0.3352 \\ -0.3352 & 0.82 \end{bmatrix} \cdot X_{2x5000} \quad (2.39)$$

A depiction of joint distribution of the whitened data is shown below in Figure 2-5.
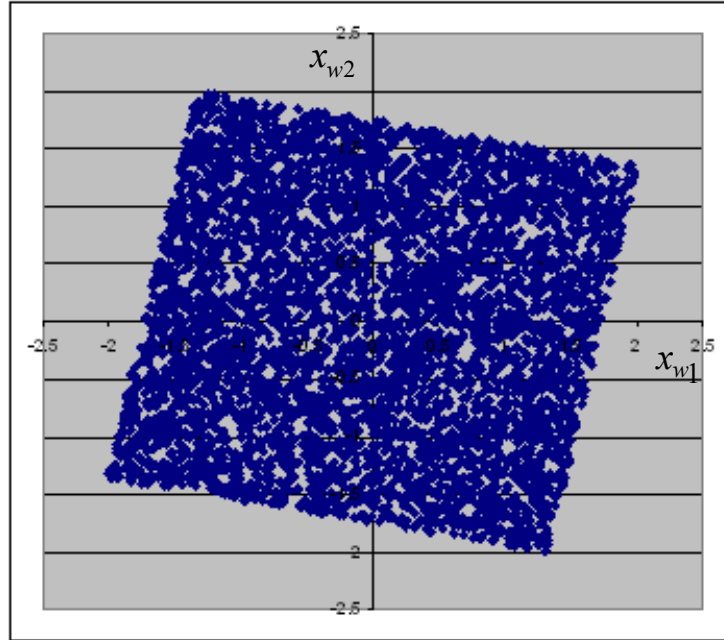
Figure 2-6. Joint Distribution of the Whitened Mixtures

Now that the linearly mixed data has been whitened, one must estimate the mixing

matrix, which as seen geometrically in Figure 2-6 given, will be an orthonormal

transformation matrix (i.e. rotation) given that the original components were of unit

variance. So, for a case where one has only 2 ICs, as in this example, the orthonormal

mixing matrix is determined by a single angle parameter, $\dfrac{p(p-1)}{2} = \dfrac{2(2-1)}{2} = 1$.

From this point forward when referring to a linearly mixed data set, it is assumed the data

has been centered and whitened.

2.2.6 Independent Components Cannot be Gaussian

Recall the assumption that the ICs cannot be Gaussian. The following discussion

offered by A. Hyvärinen, J. Karhunen, and E. Oja in their text Independent Component

Analysis motivates this assumption.

A p-dimensional random vector $\underline{s}$ is Gaussian has a joint PDF given by

$$f\left(\underline{s}\right)=\frac{1}{\left(2\pi\right)^{\frac{p}{2}}\left(\det C_s\right)^{\frac{1}{2}}}\exp\left(-\frac{1}{2}\left(\underline{s}-\underline{m}_s\right)^T C_s^{-1}\left(\underline{s}-\underline{m}_s\right)\right) \qquad (2.40)$$

where $\underline{m}_s$ is the mean vector and C_s is the covariance matrix of $\underline{s}$ and p is the dimension of the vector. If it's assumed that $\underline{m}_s$ is zero and $C_s = I$ as is the case if $\underline{s}$ is a vector of ICs with the assumptions previously mentioned then

$$f\left(\underline{s}\right)=\frac{1}{\left(2\pi\right)^{\frac{p}{2}}}\exp\left(-\frac{\underline{s}^T\underline{s}}{2}\right)=\frac{1}{\left(2\pi\right)^{\frac{p}{2}}}\exp\left(-\frac{\|\underline{s}\|_2^2}{2}\right) \qquad (2.41)$$

Consider the case with two independent Gaussian random variables whose distribution is illustrated in Figure 2-7 below.
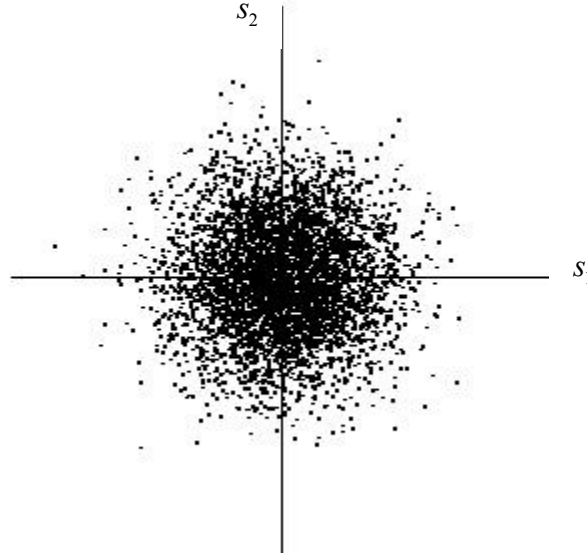
Figure 2-7. Distribution of Two Independent Gaussian RVs
(Hyvärinen and others, 2001:162)

Now it can be shown that the density of a linear transformation of $\underline{s}$ as in the ICA model $\underline{x} = \widetilde{A}\underline{s}$ is computed via the following relationship (Hyvärinen and others, 2001:36):

$$f_{\underline{x}}(\underline{x}) = \frac{1}{\left|\det\left(\widetilde{A}\right)\right|} \cdot f_{\underline{s}}\left(\widetilde{A}^{-1}\underline{x}\right) \qquad (2.42)$$

Further, assume that the mixing matrix $\widetilde{\underline{A}}$ is orthonormal because the observed data matrix has been whitened as described in the previous section in (2.36). Thus, $\widetilde{\underline{A}}^{-1} = \widetilde{\underline{A}}^{T}$. So the density of $\underline{x}$ is

$$f_x(\underline{x}) = \frac{1}{\left|\det\left(\widetilde{A}\right)\right|} \cdot f_{\underline{s}}\left(\widetilde{A}^{T}\underline{x}\right) = \frac{1}{\left|\det\left(\widetilde{A}\right)\right|} \cdot \frac{1}{(2\pi)} \exp\left(-\frac{\left\|\widetilde{A}^{T}\underline{x}\right\|_2^2}{2}\right) \qquad (2.43)$$

Now, since $\widetilde{A}$ is orthonormal, $\left\|\widetilde{A}^{T}\underline{x}\right\|_2^2 = \left\|\underline{x}\right\|_2^2$ and $\left|\det\left(\widetilde{A}\right)\right| = 1$. Thus (2.43) simplifies to

$$f_x(\underline{x}) = \frac{1}{(2\pi)} \exp\left(-\frac{\left\|x\right\|^2}{2}\right) \qquad (2.44)$$

Note the density $f_s(\underline{s})$ in (2.41) is identical to the density $f_x(\underline{x})$ in (2.44). Therefore, the orthonormal mixing matrix does not change the PDF. This is seen in the rotational symmetry of Figure 6. As a consequence, one could not infer any information about the mixing matrix after whitening a linearly mixed set of independent Gaussian random variables.

> The phenomenon that the orthogonal mixing matrix cannot be estimated for Gaussian variables is related to the property that uncorrelated jointly Gaussian variables are necessarily independent. Thus, the information on the independence of the components does not go any further than whitening…in the case of Gaussian independent components, we can only estimate the ICA model up to an orthogonal transformation. In other words, the matrix A is not identifiable for Gaussian independent components (Hyvärinen and others, 2001:162).

With linearly mixed Gaussian variables all one can do is whiten the data.

An important note for the case that some of the components are Gaussian and the rest nongaussian is:

> we can estimate all the nongaussian components, but the Gaussian components cannot be separated from each other. In other words, some of the estimated components will be arbitrary linear combinations of the Gaussian components. This means that in the case of just one Gaussian component, we can estimate the model, because the single Gaussian component does not have any other Gaussian components that it could be mixed with (Hyvärinen and others, 2001:163).

2.2.7 Measures of Nongaussianity

As mentioned previously, FastICA method finds a linear projection of the random vector $\underline{x}$ that maximizes its nongaussianity. Before justifying that this approach is equivalent to minimizing the degree of dependence between the projected components, a discussion will be presented on measures of nongaussianity.

Kurtosis

Kurtosis, widely used as a classic measure of nongaussianity, is the fourth-order cumulant of a random variable. With respect to the graph of the PDF, it can be viewed as a measure of 'peakedness'. Consider a random variable y, then the kurtosis of y is defined as

$$\text{kurt}(y) = \frac{E\{(y-\mu)^4\}}{\sigma^4} - 3 \tag{2.45}$$

If y has unit variance (2.45) simplifies to

$$\text{kurt}(y) = E\{y^4\} - 3 \tag{2.46}$$

For a Gaussian random variable of zero mean and unit variance, the 4th moment, $E\{y^4\}$ is 3. Hence, the kurtosis of a Gaussian random variable of zero mean and unit variance is

zero. For most nongaussian random variables kurtosis is non-zero. Subgaussian distributions are typically 'flat' (e.g. uniform distribution) and have negative kurtosis. Supergaussian distributions are typically 'spiky' (e.g. Laplacian distribution) and have positive kurtosis.

Suppose the random variable y is the projection of a linearly mixed random vector $\underline{x}$ defined by a weight vector $\underline{w}$, i.e. $y = \underline{w}^T \underline{x}$. Some optimization schemes search for a $\underline{w}$ that maximizes the absolute value of the $\mathrm{kurt}(\underline{w}^T \underline{x})$ to solve for one of the ICs. Such schemes are computationally simple due to the simplicity of the kurtosis calculation but suffer drawbacks. Kurtosis is very sensitive to outliers when it's estimated from a measured sample. Consider a sample of 1000 values from a random variable y with zero mean and unit variance. Say we have a single sample point that has a value of 10.

$$E\{y^4\} - 3 \approx \left(\frac{1}{1000}\sum_{i=1}^{1000} y_i^{\,4}\right) - 3 = \left(\frac{10^4}{1000} + \frac{y_2^{\,4}}{1000} + \frac{y_3^{\,4}}{1000} + \ldots + \frac{y_{1000}^{\,4}}{1000}\right) - 3 \geq \frac{10^4}{1000} - 3 = 7 \quad (2.47)$$

As one can see from (2.47), based on just one outlier the kurtosis of y will be at least 7, a large kurtosis value (Hyvärinen and others, 2001:182).

Entropy

Whereas kurtosis belongs within the field of estimation theory, or in other words, parametric statistics, an alternate way to measure nongaussianity is via a concept in information theory referred to as entropy. The entropy of a random variable relates to the information that the observations of a random variable gives or their degree of 'randomness' (Hyvärinen and others, 2001:182). The more random (unpredictable and unstructured) the variable is, the larger its entropy. This section will introduce, formally, the concept of entropy. Further, via examples generated by the author of this thesis, the

fact that the Gaussian distribution has the highest entropy of all distributions of the same mean and variance will be motivated. Finally, a scaled version of entropy called negentropy and it's computationally efficient approximation will be developed. This computationally efficient approximation for negentropy provides the basis for the objective function to be optimized to find the most nongaussian projections of the linearly mixed data, thus, approximately recovering the ICs.

Entropy for a discrete random variable is defined as follows:

$$H(x) = -\sum_i \log\left(P(x = a_i)\right)P(x = a_i)$$

$$\text{or} \tag{2.48}$$

$$H(x) = -E\left[\log\left(P(x = a_i)\right)\right]$$

It should be noted that usually the logarithm with base 2 is used, in which case the unit of entropy is called a bit (Hyvärinen and others, 2001:105). Let

$$f(p) = -p\log(p), \text{ for } 0 \le p \le 1 \tag{2.49}$$

Thus, rewriting (2.48)

$$H(x) = \sum_i f(p) \tag{2.50}$$

Below in Figure 2-8 is a graph of $f(p)$.
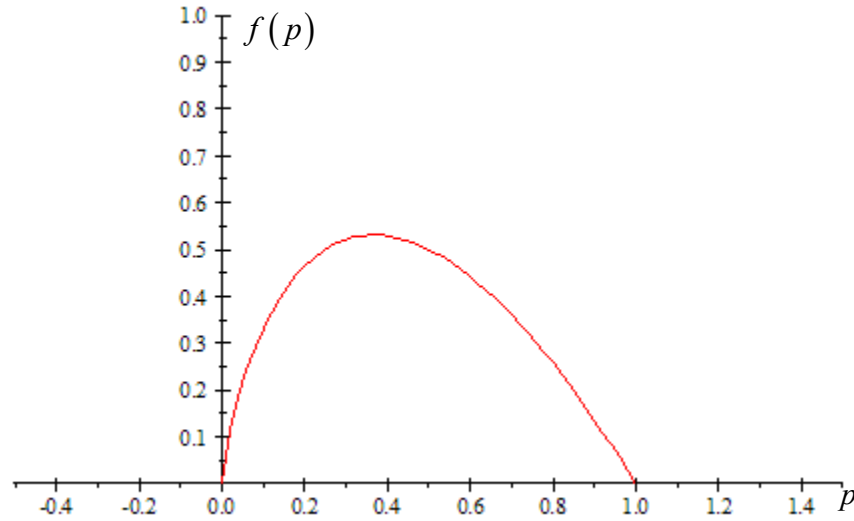
Figure 2-8. Graph of $f(p) = -p\log_2 p$ (Hyvärinen and others, 2001:106)

One can see that entropy is small if the probabilities, p, are close to 0 or 1 and large if the probabilities are in between. Consider a random variable that assumes three values with two probabilities all close to 0 (.001 and .001) and one that is close to one (.998).

$$H(x) = f(0.001) + f(0.001) + f(0.998) = 0.009966 + 0.009966 + 0.002883 = .022814 \quad (2.51)$$

For this case as shown in (2.51), the entropy is small as shown by the value of $H(x)$.

Intuitively this makes since given that the random variable almost always takes on the same value with probability of 0.998. Conversely, consider a random variable that takes on three values with equal probability.

$$H(x) = f\left(\frac{1}{3}\right) + f\left(\frac{1}{3}\right) + f\left(\frac{1}{3}\right) = 1.58496 \quad (2.52)$$

Here the variable is more 'random' and thus has a higher entropy value.

It has been shown that the Gaussian distribution holds the property of having the largest entropy among all random variables of unit variance (Hyvärinen and others,

2001:112). Rather than proving this rigorously, this assertion will be motivated by this

author via the computation of entropy of three standardized (zero mean and unit variance)

distributions, the Laplacian, uniform, and the Gaussian. First, the definition of entropy of

a continuous random variable x (which can be generalized to a random vector $\underline{x}$), termed

differential entropy is

$$H(x) = -\int \log(f(x)) \cdot f(x) dx$$
$$\text{or}$$
$$H(x) = -E\left[\log(f(x))\right] \tag{2.53}$$
$$\text{for a random vector the definition is the same}$$
$$H(\underline{x}) = -E\left[\log(f(\underline{x}))\right]$$

where $f(x)$ is the PDF of the random variable (vector) x. Figures 2-9, 2-10, and 2-11

below show the graphs of the PDFs and their respective entropies.

Laplacian distribution (supergaussian) $f(y) = \dfrac{1}{\sqrt{2}} \exp\left(-\sqrt{2}\,|y|\right)$

$$H(y) = -\int f(y) \log_2 f(y) dx = -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}\,|y|\right) \log_2\left(\frac{1}{\sqrt{2}} \exp\left(-\sqrt{2}\,|y|\right)\right) dy = 1.9427$$
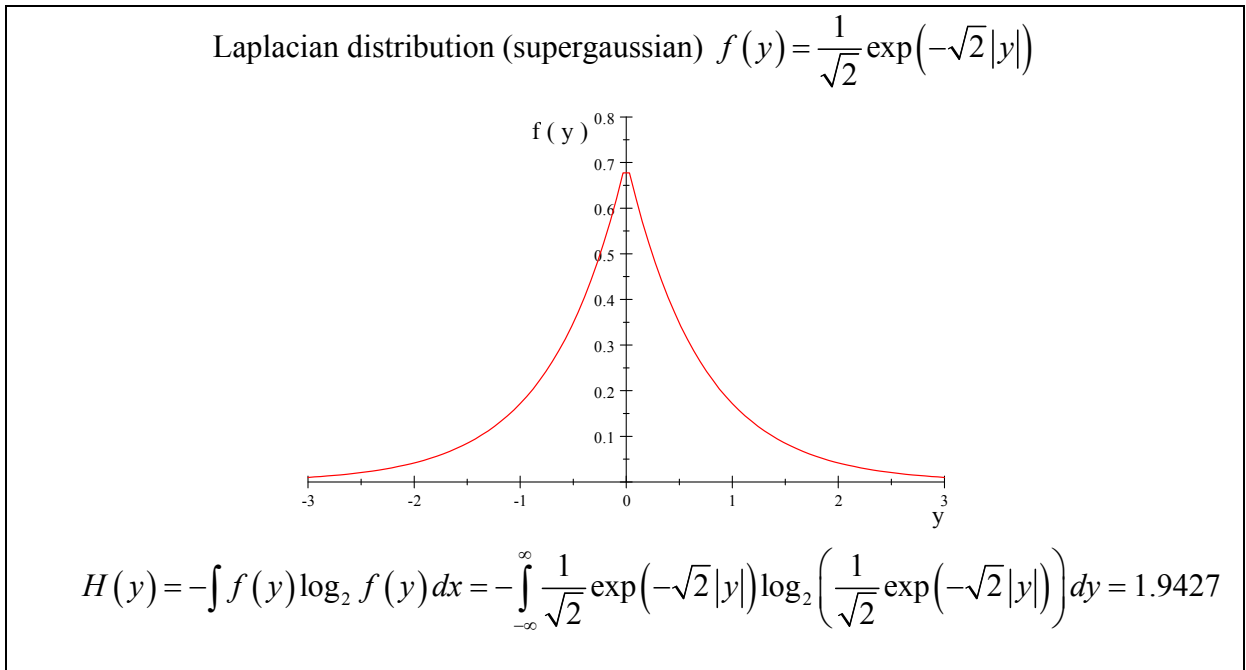
Figure 2-9. Laplace Distribution and its Entropy

Uniform distribution (subgaussian) $f(y) = \begin{cases} \dfrac{1}{2\sqrt{3}}, & \text{if } |y| \leq \sqrt{3} \\ 0, & \text{otherwise} \end{cases}$
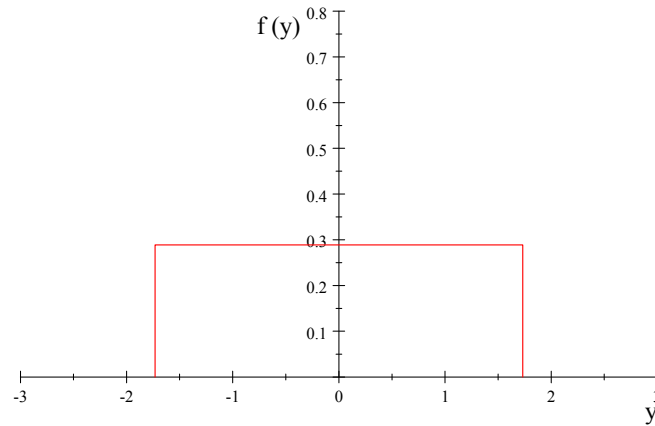
$$H(y) = -\int f(y)\log_2 f(y)\,dx = -\int_{-\sqrt{3}}^{\sqrt{3}} \frac{1}{2\sqrt{3}}\log_2\left(\frac{1}{2\sqrt{3}}\right)dy = 1.7925$$

Figure 2-10. Uniform Distribution and its Entropy

Gaussian distribution $f(y) = \dfrac{1}{\sqrt{2\pi}}\exp\left(-\dfrac{y^2}{2}\right)$
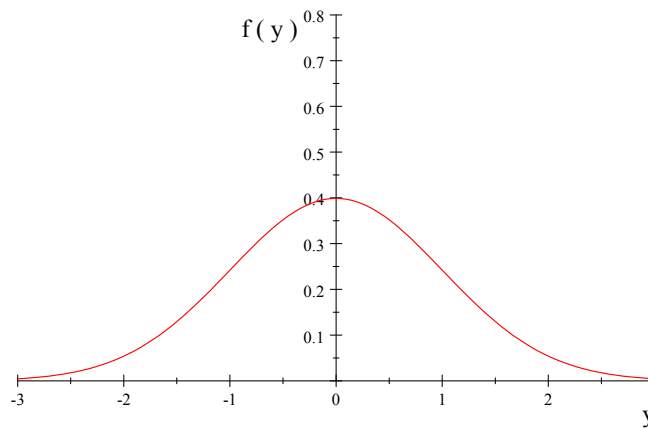
$$H(y) = -\int f(y)\log_2 f(y)\,dx = -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right)\log_2\left(\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{y^2}{2}\right)\right)dy = 2.0471$$

Figure 2-11. Gaussian Distribution and its Entropy

As shown, the Gaussian random variable has the highest value for entropy at 2.0471. The result of the maximum entropy property of the Gaussian distribution can be generalized to multidimensional spaces and arbitrary variances such that the multivariate Gaussian distribution has the maximum entropy among all distributions with the same covariance matrix (Hyvärinen and others, 2001:112). Thus, entropy could be used as a measure of nongaussianity of a random variable. The lower the variable's entropy the more nongaussian it is.

Negentropy

Negentropy is a quantity that scales a random variable's entropy such that the measure is zero for a Gaussian variable and always nonnegative. The result is that the less entropic (more nongaussian) a random variable is, the higher its negentropy value will be. Thus, for the ICA problem, one will need to find a vector $\underline{w}$ that maximizes the negentropy of $\underline{w}^T \underline{x}$. The definition of negentropy for a continuous random variable (vector) is

$$J(x) = H(x_{gauss}) - H(x) \tag{2.54}$$

where x_{gauss} is a Gaussian random variable (vector) with the same variance (covariance matrix) as x. From the three examples just mentioned, the negentropy of the uniform random variable would be

$$J(y_{uniform}) = H(y_{gauss}) - H(y_{uniform}) = 2.0471 - 1.7925 = 0.2546 \tag{2.55}$$

To use negentropy in practice, one would have to compute an integral involving a probability density function (that must be estimated in some fashion) which would be computationally complex. However, approximations of negentropy exist that are

computationally simple (integral computation not required) like the kurtosis based

measure of nongaussianity, but are robust to the presence of outliers, unlike the kurtosis

based measures.

First, an approximation of negentropy will be derived using high order moments

to approximate the density of the observed data (which is still plagued by the outlier

problem since it uses high order moments). However, this is done first as a motivation to

another approximation using nonpolynomial functions which are robust to the presence

of outliers.

Approximation of Negentropy Using the Gram-Charlier Expansion

Assuming the random variable x has been standardized (zero mean and unit

variance) an approximation of its density can be accomplished using a Taylor-like

expansion called the Gram-Charlier expansion of the PDF of x. This type of expansion is

belongs to a class called polynomial density expansions. If we make the assumption that

the distribution of x is 'near' the standardized Gaussian density, $f(v) = \dfrac{1}{\sqrt{2\pi}} \exp\left(\dfrac{-v^2}{2}\right)$,

the PDF of x can be approximated by

$$f(x) \approx \widehat{f}(x) = f(v)\left(1 + k_3(x)\frac{H_3(v)}{3!} + k_4(x)\frac{H_4(v)}{4!} + ...\right) \qquad (2.56)$$

where

$k_3(x) = E\{x^3\}$, skewness of observables

$k_4(x) = E\{x^4\} - 3$, kurtosis of observables

$H_i(v) \equiv$ Chebyshev-Hermite polynomials.

These polynomials are solutions to the differential

equation, $\dfrac{\partial^i f(v)}{\partial v^i} = (-1)^i H_i(v) f(v)$ The

polynomials form an orthonormal system meaning:

$$\int f(v) H_i(v) H_j(v) dv = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

The expansion of $\widehat{f}(x)$ in (2.56) has an infinite number of terms, but will be truncated after the fourth moment. The expansion starts at high order moments because x has been standardized to have zero mean and unit variance. In essence, one can see in that the estimated distribution of x is determined by its skewness and kurtosis. Now, this approximation for the PDF of x can be substituted into the definition of entropy giving

$$H(x) \approx -\int \widehat{f}(x) \log \widehat{f}(x) dv$$

$$= -\int \left[f(v) \left(1 + k_3(x) \frac{H_3(v)}{3!} + k_4(x) \frac{H_4(v)}{4!} \right) \right] \cdot \log \left[f(v) \left(1 + k_3(x) \frac{H_3(v)}{3!} + k_4(x) \frac{H_4(v)}{4!} \right) \right] dv$$

$$(2.57)$$

Letting $e = k_3(x) \dfrac{H_3(v)}{3!} + k_4(x) \dfrac{H_4(v)}{4!}$, (2.57) simplifies to

$$= -\int \left[f(v)(1+e) \right] \cdot \log \left[f(v)(1+e) \right] dv$$
$$= -\int \left[f(v)(1+e) \right] \cdot \left[\log f(v) + \log(1+e) \right] dv$$

$$(2.58)$$

Now if the PDF of x is close to Gaussian, e should be small. A simple approximation can then be used

$$\log(1+e) \approx e - \frac{e^2}{2} \qquad (2.59)$$

This further simplifies (2.58) giving

$$\approx -\int \left[f(v)(1+e) \right] \cdot \left[\log f(v) + e - \frac{e^2}{2} \right] dv \qquad (2.60)$$

Resubstituting the value of e

$$= -\int \left[f(v) \left(1 + k_3(x) \frac{H_3(v)}{3!} + k_4(x) \frac{H_4(v)}{4!} \right) \right] \cdot \left[\log f(v) + k_3(x) \frac{H_3(v)}{3!} + k_4(x) \frac{H_4(v)}{4!} - \frac{\left(k_3(x) \frac{H_3(v)}{3!} + k_4(x) \frac{H_4(v)}{4!} \right)^2}{2} \right] dv$$

$$(2.61)$$

Given that the Chebyshev-Hermite polynomials are orthonormal as defined in (2.56), the final simplification for entropy yields

$$H(x) \approx -\int f(v) \log f(v) \, dv - \frac{k_3(x)^2}{2 \cdot 3!} - \frac{k_4(x)^2}{2 \cdot 4!} \qquad (2.62)$$

Now

$$J(x) = H(x_{gauss}) - H(x) \approx -\int f(v) \log f(v) \, dv - \left(-\int f(v) \log f(v) \, dv - \frac{k_3(x)^2}{2 \cdot 3!} - \frac{k_4(x)^2}{2 \cdot 4!} \right)$$

$$= \frac{k_3(x)^2}{2 \cdot 3!} + \frac{k_4(x)^2}{2 \cdot 4!} = \frac{1}{12} k_3(x)^2 + \frac{1}{48} k_4(x)^2 = \frac{1}{12} E\{x^3\}^2 + \frac{1}{48} \left[E\{x^4\} - 3 \right]^2$$

So, finally the computationally simple approximation of negentropy is

$$J(x) \approx \frac{1}{12} E\{x^3\}^2 + \frac{1}{48} \left[E\{x^4\} - 3 \right]^2 \qquad (2.63)$$

As stated previously, this measure of nongaussianity will still be plagued by the outlier problem just as the kurtosis measure since this approximation for negentropy also uses higher-order cumulants, skewness and kurtosis. However, the form of (2.63) provides motivation for another approximation robust to the presence of outliers which will be discussed next.

Approximation of Negentropy Using Expectations of Nonpolynomial Functions

An alternate density approximation derived from a first-order approximation of the maximum entropy density for a continuous random variable (given a number of

simple constraints) results in a density expansion similar to the classic polynomial density expansion by Gram-Charlier. However, this approximation of entropy is more exact and more robust against outliers than the approximations based on the polynomial density expansions, without being computationally more expensive (Hyvärinen and others, 2001:116). The derivation of this alternate density expansion is outside the scope of this thesis, but can be thought of as a generalization of the higher-order cumulant approximation in (2.63) using expectations of general nonquadratic functions, or 'nonpolynomial moments' rather than $E(x^3)$ and $E(x^4)$, the standard moments used by Gram-Charlier. In general one replaces the polynomial functions x^3 and x^4 in (2.63) with other functions G_i. This alternate approximation of negentropy is

$$J(x) \approx k_1 \left(E\{G_1(x)\}\right)^2 + k_2 \left(E\{G_2(x)\} - E\{G_2(v)\}\right)^2 \qquad (2.64)$$

where

$$v \equiv \text{Gaussian variable of zero mean and unit variance}$$
$$k_1 \text{ and } k_2 \equiv \text{positive constants}$$

Notice if one takes $G_1 = x^3$ and $G_2 = x^4$, $J(x) \approx k_1 \left(E\{x^3\}\right)^2 + k_2 \left(E\{x^4\} - E\{v^4\}\right)^2$. Now since v is a standardized Gaussian random variable, $E\{v^4\} = 3$. So one has

$J(x) \approx k_1 \left(E\{x^3\}\right)^2 + k_2 \left(E\{x^4\} - 3\right)^2$, which is identical to (2.63) for the same choice of constants. In the case where only one nonquadratic function is used (2.64) simplifies to

$$J(x) \approx k \left(E\{G(x)\} - E\{G(v)\}\right)^2 \qquad (2.65)$$

This is merely a generalization of (2.63) if x is symmetric, in which case the first term in (2.63) is zero.

By choosing a function G that does not grow too fast, one can obtain approximations of negentropy more robust to the presence of outliers than (2.63). The following choices of G have proven to be effective:

$$G_a(y) = \frac{1}{a_1}\log\left(\cosh a_1 y\right) \tag{2.66}$$

$$G_b(y) = -\exp\left(\frac{-y^2}{2}\right) \tag{2.67}$$

where $1 \leq a_1 \leq 2$ is some constant, often taken to equal one (Hyvärinen and others, 2001:184). Figure 2-12 below shows G_a and G_b compared to x^4. The dashed line represents x^4, the dotted line G_a, and the solid line G_b. Notice the growth of the G's versus x^4.
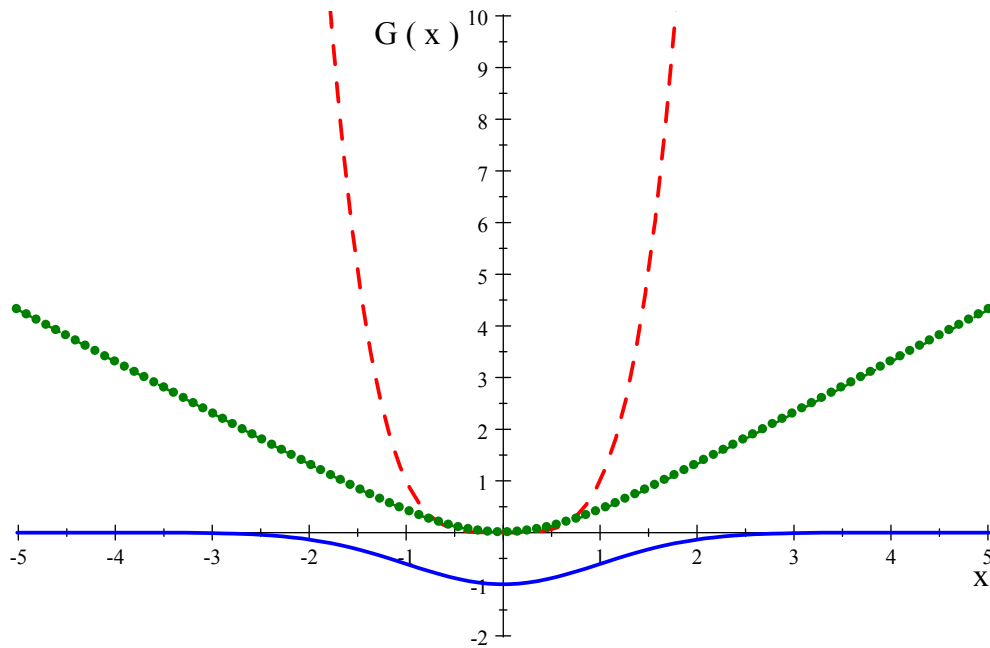
Figure 2-12. The functions G_a in (2.66), G_b in (2.67) given by the dotted and solid curve, respectively compared to x^4, the dashed curve (Hyvärinen and others, 2001:184).

In conclusion, one now has an approximation of negentropy (a measure of nongaussianity) that is computationally simple and has appealing statistical properties as robustness in the presence of outliers.

2.2.8 Connection between Mutual Information and Nongaussianity

As stated previously, this particular solution to the ICA problem finds projections of the linearly mixed data such the nongaussianity of the elements of the new random vector is maximized. This section will argue that in the case where our components are constrained to be uncorrelated and of unit variance, maximizing nongaussianity is equivalent to minimizing the mutual information (a measure of dependence) between the p random variables, s_i, $i = 1...p$.

For a p-dimensional random vector $\underline{s}$, the mutual information of its elements is defined as follows:

$$I\left(s_1, s_2, ..., s_p\right) = E\left[\log\left(\frac{f\left(\underline{s}\right)}{\prod_{i=1}^{p} f\left(s_i\right)}\right)\right] \tag{2.68}$$

where $f\left(\underline{s}\right)$ is the joint density and $f\left(s_i\right)$ are the marginal densities. When the components, s_i, are independent, the ratio inside the logarithm in(2.68) reduces to one and thus, the mutual information becomes zero. The converse of this statement is also true (Varshney and Arora, 2004:115). Varshney and Arora state the mutual information can be also defined in terms of the difference between the sum of the marginal (individual) entropies and the joint entropy. This author provides the proof of this

statement below in (2.69) by simplifying (2.68) using log identities and applying the

definition of entropy given in (2.53)

$$
\begin{aligned}
I\left(s_1, s_2, \ldots, s_p\right) &= E\left[\log\left(\frac{f(\underline{s})}{\prod_{i=1}^{p} f(s_i)}\right)\right] = E\left[\log(f(\underline{s})) - \log\left(\prod_{i=1}^{p} f(s_i)\right)\right] \\
&= E\left[\log(f(\underline{s}))\right] - E\left[\sum_{i=1}^{p} \log(f(s_i))\right] \\
&= E\left[\log(f(\underline{s}))\right] - \sum_{i=1}^{p} E\left[\log(f(s_i))\right]
\end{aligned}
$$

from (2.53)

$$
= -H(\underline{s}) + \sum_{i=1}^{p} H(s_i)
$$

$$
\Rightarrow I\left(s_1, s_2, \ldots, s_p\right) = \sum_{i=1}^{p} H(s_i) - H(\underline{s})
$$

(2.69)

Recall that the random vector of ICs, from (2.24), is defined as $\underline{s} = W\underline{x}$ where W is the

inverse of the mixing matrix A. It's been proven that for an invertible linear

transformation of $\underline{s}$ (Hyvärinen and others, 2001:109) that

$$
H(\underline{s}) = H(\underline{x}) + \log|\det W|
$$

(2.70)

Now, recall that if $\underline{x}$ has been whitened, the mixing matrix is constrained to be

orthonormal to enforce unit variance and decorrelation of the individual s_i as seen from

(2.36). Since W is the inverse of the mixing matrix, W is also orthonormal. Thus

$\log|\det W| = \log(1) = 0$. So for whitened data

$$
H(\underline{s}) = H(\underline{x})
$$

(2.71)

Substituting (2.71) into (2.69)

$$I\left(s_1, s_2, \ldots, s_p\right) = \sum_{i=1}^{p} H\left(s_i\right) - H\left(\underline{s}\right) = \sum_{i=1}^{p} H\left(s_i\right) - H\left(\underline{x}\right)$$

$$= \sum_{i=1}^{p}\left[H\left(s_{i\,gauss}\right) - J\left(s_i\right)\right] - H\left(\underline{x}\right), \text{ from definition of negenentropy in (2.54)}$$

(note: $s_{i\,gauss}$ is a Gaussian RV with mean = 0 and variance = 1)

$$= \sum_{i=1}^{p} H\left(s_{i\,gauss}\right) - \sum_{i=1}^{p} J\left(s_i\right) - H\left(\underline{x}\right)$$

(2.72)

now $\sum_{i=1}^{p} H\left(s_{i\,gauss}\right)$ and $H\left(\underline{x}\right)$ are constants

$$\Rightarrow I\left(s_1, s_2, \ldots, s_p\right) = \text{constant} - \sum_{i=1}^{p} J\left(s_i\right)$$

The result of (2.72), $I\left(s_1, s_2, \ldots, s_p\right) = \text{constant} - \sum_{i=1}^{p} J\left(s_i\right)$, stated by Hyvärinen,

Karhunen, and Oja in their text, "Independent Component Analysis" and proven here by

this author, shows that by maximizing the negentropy (which maximizes the

nongaussianity) of the individual components, $J\left(s_i\right)$, one is in effect minimizing the

mutual information (statistical dependency) between the s_i's. Thus, a rigorous

justification for finding projections of the data that maximize their nongaussianity to

approximately recover the ICs has been provided.

2.2.9 FastICA to Estimate One Component (One Unit FastICA)

Now that a suitable approximation for negentropy (a measure of nongaussianity)

has been derived, the remaining issue left to in order to solve the ICA problem is a

constrained optimization problem. The objective function to be maximized will be

(2.65), the approximation for negentropy using only one nonquadratic function,

$J\left(s_i\right) \approx k\left(E\left\{G\left(s_i\right)\right\} - E\left\{G\left(v\right)\right\}\right)^2$, where v is a Gaussian random variable with zero

mean and unit variance and G is either (2.66) or (2.67). This section will proceed to derive an iteration scheme that will recover a single independent component s_i of $\underline{s}$.

Now, since the random vector $\underline{s} = W\underline{x}$ from (2.24), any particular random variable of $\underline{s}$, $s_i = \underline{w}^T \underline{x}$, where $\underline{w}^T$ is a row of W, the inverse of the mixing matrix $\widetilde{A}$. This row of W defines the nongaussian projection of the random vector $\underline{x}$ whose value one wishes to maximize. Thus, rewriting (2.65) one has

$$J\left(\underline{w}^T \underline{x}\right) \approx k\left(E\left\{G\left(\underline{w}^T \underline{x}\right)\right\} - E\left\{G(v)\right\}\right)^2 \tag{2.73}$$

Recall that in order to enforce decorrelation and unit variance of the ICs, $\widetilde{A}$ is restricted to be orthonormal as seen from (2.36). Since W is the inverse of an orthonormal matrix i.e. $\underline{W} = \widetilde{A}^{-1}$, this implies $\underline{W} = \widetilde{A}^T$. So the rows of W are the columns of A. Thus, the norm of each row of W must be one, i.e. $\|\underline{w}\|_2^2 = 1$, which will be the constraint in the optimization problem. Further, since W is the inverse of an orthonormal matrix, W is also orthonormal.

Note that the maxima of the approximation of the negentropy of $\underline{w}^T \underline{x}$ in (2.73) are typically obtained at certain optima of $E\left\{G\left(\underline{w}^T \underline{x}\right)\right\}$ (Hyvärinen and others, 2001:189). It should be noted that since one is dealing with sample data one will compute this expectation by its estimate, i.e. $E\left\{G\left(\underline{w}^T \underline{x}\right)\right\} \approx \frac{1}{n}\sum_{i=1}^{n} G\left(\underline{w}^T \underline{x}(i)\right)$, where $\underline{x}(i)$ is a realization of the random vector $\underline{x}$, i.e. a column of the sample data matrix X defined in (2.23). Thus, we have the following optimization problem:

$$\max_{\underline{w}} \; E\left\{G\left(\underline{w}^T \underline{x}\right)\right\}$$

$$\text{s.t. } \left\|\underline{w}\right\|_2^2 = 1 \tag{2.74}$$

According to first-order necessary conditions for optimality, candidates for optima of (2.74) are obtained at the stationary points of the Lagrangian function, i.e. where gradient of the Lagrangian is zero. The Lagrangian of (2.74) is

$$L\left(\underline{w},\,\lambda\right) = E\left\{G\left(\underline{w}^T \underline{x}\right)\right\} - \lambda\left(\left\|\underline{w}\right\|_2^2 - 1\right) \tag{2.75}$$

The gradient of (2.75) denoted by $F\left(\underline{w}\right)$ below is

$$\frac{\partial L}{\partial \underline{w}} = F\left(\underline{w}\right) = E\left\{\frac{\partial G\left(\underline{w}^T \underline{x}\right)}{\partial \underline{w}}\right\} - \frac{\partial \lambda \left\|\underline{w}\right\|_2^2}{\partial \underline{w}} = E\left\{\underline{x} \cdot g\left(\underline{w}^T \underline{x}\right)\right\} - 2\lambda \underline{w} = 0 \tag{2.76}$$

where $g = G'$. In order the find the zeros of (2.76), one can use the following Newton iteration scheme followed by a normalization of the weight vector at each iteration

$$\underline{w}^+ = \underline{w} - F'\left(\underline{w}\right)^{-1} \cdot F\left(\underline{w}\right)$$

$$\underline{w} = \frac{\underline{w}^+}{\left\|\underline{w}^+\right\|} \tag{2.77}$$

Now

$$F'\left(\underline{w}\right) = E\left\{\underline{x}\underline{x}^T \cdot g'\left(\underline{w}^T \underline{x}\right)\right\} - \beta I \tag{2.78}$$

where $2\lambda = \beta$. A reasonable approximation to be made, given the random vector $\underline{x}$ is whitened (Hyvärinen and others, 2001:189) is

$$E\left\{\underline{x}\underline{x}^T \cdot g'\left(\underline{w}^T \underline{x}\right)\right\} \approx E\left\{\underline{x}\underline{x}^T\right\} \cdot E\left\{g'\left(\underline{w}^T \underline{x}\right)\right\} = E\left\{g'\left(\underline{w}^T \underline{x}\right)\right\} \cdot I \tag{2.79}$$

So the approximation to (2.78) becomes

$$F'(\underline{w}) \approx E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\bullet I - \beta I = \left(E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\} - \beta\right)I \tag{2.80}$$

Thus, the approximate Newton iteration scheme is

$$\underline{w}^+ = \underline{w} - \left[\left(E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\} - \beta\right)\bullet I\right]^{-1}\bullet\left(E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - \beta\underline{w}\right) \tag{2.81}$$

Since the approximation for $F'(\underline{w})$ given in (2.80) is a diagonal matrix with the same

element along the diagonal, matrix inversion is not needed at each iteration, a high

computational cost for the traditional Newton method. Thus, (2.81) simplifies to

$$\underline{w}^+ = \underline{w} - \frac{E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - \beta\underline{w}}{E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\} - \beta} \tag{2.82}$$

The algorithm can be further simplified by multiplying both sides of (2.82) by

$\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}$ as shown below in (2.83).

$$\underline{w}^+\bullet\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right) = \left(\underline{w} - \frac{E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - \beta\underline{w}}{E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\} - \beta}\right)\bullet\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right)$$

$$\Rightarrow \underline{w}^+\bullet\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right) = \beta\underline{w} - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\underline{w} + E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - \beta\underline{w}$$

$$\Rightarrow \underline{w}^+\bullet\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right) = E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\underline{w}$$

$$\Rightarrow \underline{w}^+ = \frac{1}{\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right)}\bullet\left(E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\underline{w}\right)$$

Let $\gamma = \dfrac{1}{\left(\beta - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\right)}$, note γ is a scalar

$$\Rightarrow \underline{w}^+ = \gamma\left(E\left\{\underline{x}\cdot g\left(\underline{w}^T\underline{x}\right)\right\} - E\left\{g'\left(\underline{w}^T\underline{x}\right)\right\}\underline{w}\right)$$

$$\tag{2.83}$$

Finally, the basic iteration scheme in one unit FastICA is (Hyvärinen and others, 2001:189):

$$\underline{w}^+ = E\left\{\underline{x} \cdot g\left(\underline{w}^T \underline{x}\right)\right\} - E\left\{g'\left(\underline{w}^T \underline{x}\right)\right\} \underline{w}$$

$$\underline{w} = \frac{\underline{w}^+}{\left\|\underline{w}^+\right\|} \tag{2.84}$$

Note that the scalar γ has been omitted because it would be eliminated in the subsequent normalization step. The function g and g' presented in (2.85) and (2.86) are the derivatives and second derivatives of (2.66) or (2.67) respectively. Choices for g and g' in the FastICA algorithm are shown below.

$$g_a(y) = \tanh(a_1 y)$$

$$g_a'(y) = a_1\left(1 - \tanh^2(a_1 y)\right) \tag{2.85}$$

$$g_b(y) = y \exp\left(\frac{-y^2}{2}\right)$$

$$g_b'(y) = (1 - y)^2 \exp\left(\frac{-y^2}{2}\right) \tag{2.86}$$

(for the case where one uses kurtosis to approximate negentropy)

$$g_c(y) = y^3$$

$$g_c'(y) = 3y^2 \tag{2.87}$$

Recall the whitened data matrix X_w whose joint distribution is depicted in Figure 5. If the iteration scheme in (2.84) is run (choosing $g_a(y)$) and using a random $\underline{w}$ as the initial solution, the algorithm converges (i.e. the dot product of the current and the previous iterate is close to one within some tolerance) to the following weight vector

$\underline{w}^T = \begin{bmatrix} 0.9824 & -0.1873 \end{bmatrix}$ depicted below in Figure 2-13. This vector represents the direction of maximum negentropy (nongaussianity).
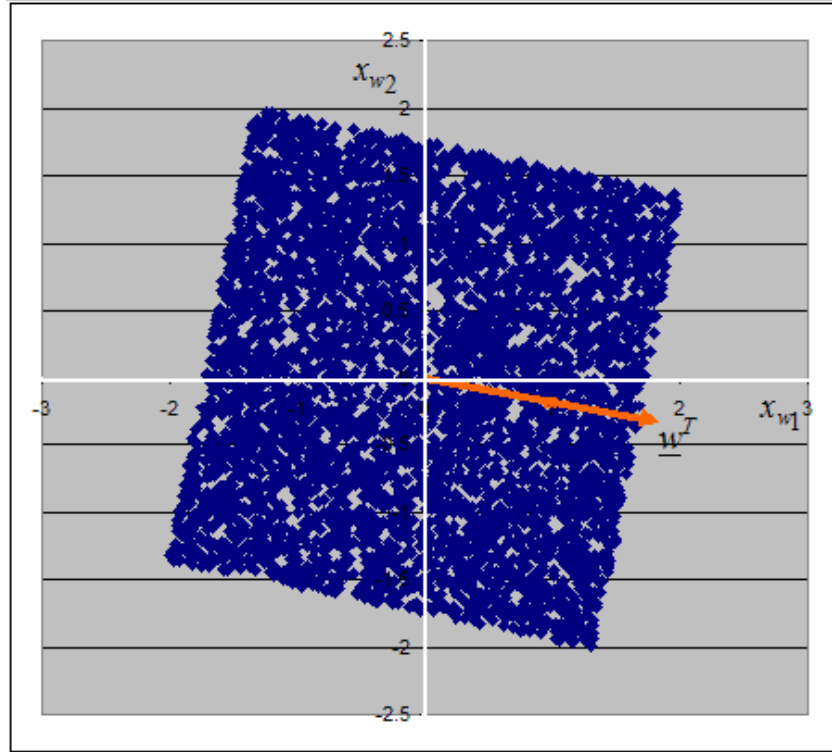
Figure 2-13. Direction of Maximum Negentropy (Nongaussianity)

Thus, $\underline{w}^T{}_{1x2} \cdot \underline{X}_{w_{2x5000}} = s_1$ would be the estimated recovered signal (i.e. realizations of the random variable across 5000 observations). To find the other vector defining the projection corresponding to the second independent component, one could just find a vector perpendicular to the first vector since we know that the vectors $\underline{w}^T$ are orthonormal in the whitened space. This leads to a brief discussion of how to estimate more than just one independent component simultaneously.

2.2.10 FastICA to Estimate Multiple Components

As stated previously, since the projection matrix W is the inverse of the orthonormal mixing matrix $\tilde{A}$, the rows (as well as the columns) of W are orthogonal and of unit length. Further i^{th} row of W defines the projection of the linearly mixed random vector $\underline{x}$ onto s_i as shown below in (2.88)

$$\underbrace{\begin{pmatrix} s_1 \\ s_2 \\ . \\ s_P \end{pmatrix}}_{\underline{s}} = \underbrace{\begin{pmatrix} w_{11} & w_{12} & ... & w_{1P} \\ w_{21} & w_{22} & ... & w_{2P} \\ . & . & . & . \\ w_{P1} & w_{P2} & ... & w_{PP} \end{pmatrix}}_{W} \bullet \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ . \\ x_P \end{pmatrix}}_{\underline{x}} \qquad (2.88)$$

For the rest of the discussion in this section let $\underline{w}_i^T$ represent the ith row of W.

One way to solve for multiple ICs would be to use the one unit FastICA algorithm in (2.84) to find a projection vector $\underline{w}_i^T$ and then use the algorithm again but add an additional constraint that the second projection vector, $\underline{w}_j^T$, must be orthogonal to the previous. The third would then have to be orthogonal to the first and second and so on. The Gram-Schmidt method could be used to orthogonalize the current projection vector estimate with all the previously estimated projection vectors at each iteration of the one unit FastICA algorithm, a process termed deflationary orthogonalization. The problem with this sequential orthogonalization approach is that each application of the one unit algorithm produces only an estimate of the true projection vector $\underline{w}_i^T$ and these estimation errors are accumulated in the subsequent estimates of other projection vectors during the orthogonalization process (Hyvärinen and others, 2001:195).

A better approach, rather than estimating the $\underline{w}_i^T$'s one by one, is to compute them

in parallel at each iteration of an algorithm. Then at each iteration, the p row vectors that

compose the matrix W will be orthogonalized meaning a new set of vectors will be found

that are orthogonal to each other, but also span the same space as the vectors of W at the

current iterate, i.e. an orthonormal basis for the space spanned by the original vectors.

In symmetric orthonormalization (orthogonalization and normalization to norm 1)

methods, none of the original vectors $\underline{w}_j^T$ are treated differently from the others as is the

case in deflationary orthogonalization. A classic approach to accomplish the

orthonormalization of W is the following equation (Hyvärinen and others, 2001:195):

$$W^+ = \left(W \cdot W^T\right)^{-\frac{1}{2}} W \qquad (2.89)$$

The result is a matrix where the $\underline{w}_i^T$'s are orthogonal and have unit norm. The FastICA

algorithm to find multiple ICs is presented below in (2.90).

1. Center and whiten the observed data.
2. Choose p, the number of ICs to estimate.
3. Choose the initial values for the $\underline{w}_i^T's$, $i = 1,...,p$, each of unit norm.
   Complete a symmetric orthogonalization of the matrix W as shown in
   (2.89)
4. For every $i = 1,...p$, let $\underline{w}_i^+ = E\left\{\underline{x} \cdot g\left(\underline{w}_i^T \underline{x}\right)\right\} - E\left\{g'\left(\underline{w}_i^T \underline{x}\right)\right\} \underline{w}_i$ (one
   unit ICA presented in (2.84)
5. Complete a symmetric orthogonalization of the matrix W as shown in
   (2.89)
6. If not converged, go back to step 4 (Hyvärinen and others, 2001:195).

$$(2.90)$$

Running the algorithm in (2.90) on the whitened data matrix $X_{W_{2 \times 5000}}$ whose distribution is

shown in Figure 5, the W matrix (demixing matrix) after convergence is

$$W = \begin{bmatrix} 0.982401 & -0.18732 \\ 0.185004 & 0.982840 \end{bmatrix}.$$ Thus, the two projection vectors are

$\underline{w_1}^T = (0.982401 \quad -0.18732)$ and $\underline{w_2}^T = (0.185004 \quad 0.982840)$. A plot of these vectors

on the whitened distribution is displayed in Figure 2-14 below.
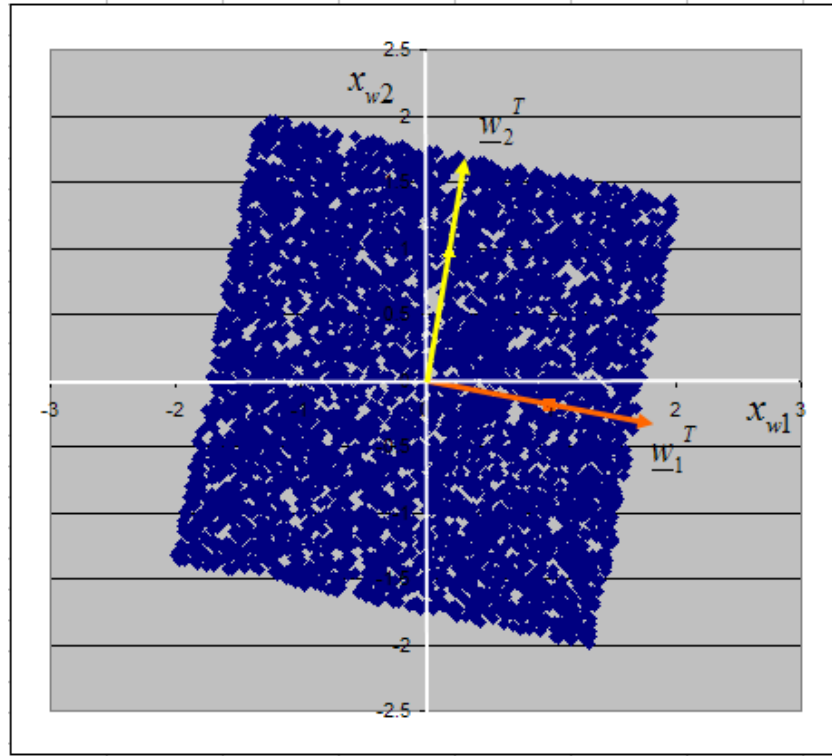
Figure 2-14. Projection Vectors to Recover Both of the Independent Components

Now that the technique of ICA has been rigorously developed, the next section

will proceed to discuss how the ICA model approximates the LMM of hyperspectral data

and can be used to solve for the endmember matrix and abundance matrix.

2.3 ICA Applicability to HSI

Recall that ICA utilizes higher-order statistics, third and fourth moments as shown

in (2.63) and nonpolynomial generalizations of third and fourth moments as shown in

(2.64), to find projections of the data that maximize these higher order statistics which in turn approximately recovers components (features) that are independent. Further, prior to running the ICA algorithm, first and second order statistics (mean and variance) are removed. A. Bell and T. Sejnowski write in an article titled *The 'Independent Components' of Natural Scenes are Edge Filters* from the Vision Research Journal in December 1997 that

> …second-order statistics correspond to the amplitude spectrum of a signal. The remaining information, higher-order statistics, corresponds to the phase spectrum. The phase spectrum is what we consider to be the informative part of a signal, since if we remove phase information from an image, it looks like noise, while if we remove amplitude information (for example, with whitening), the image is still recognizable. Edges and what we consider 'features' in images are suspicious coincidences in the phase spectrum (Bell and Sejnowski, 1997:3335).

Similarly, as stated by Q. Du, I. Kopriva, and H. Szu in an article titled *Independent-component analysis for hyperspectral remote sensing imagery classification* from the Optical Engineering journal in January 2006

> In contrast with many conventional techniques which use up to second order statistics only, ICA exploits higher-order statistics, which makes it more powerful in extracting irregular features in the data (Du and others, 2006:2).

Thus, for the purpose of extracting irregular features, which could possibly be targets in the military context, ICA appears to be well suited theoretically to accomplish this task.

2.3.1 ICA as a Solution to the LMM of HSI

One may question the interpretation of the mixing matrix and the ICs in the context of HSI. Recall the LMM of HSI given in (2.2),

$X_{KxN} = E_{KxP} \cdot S_{PxN} + R_{KxN}$, where X represents N observations of a K dimensional

random vector where the elements of the vector are reflectance responses in the K

spectral bands, E represents P endmember signatures in the K spectral bands, S

represents the abundance fractions of the P endmembers for each of the N

observations, and finally R represents Gaussian noise in each of the N

observations. Assume in the endmember matrix, there are as many endmembers

as there are spectral bands. Thus, E is a square matrix and one has

$X_{KxN} = E_{KxK} \cdot S_{KxN} + R_{KxN}$. Also, recall the LMM after dimensionality reduction to the

PCA space given in (2.19), $Y_{PxN} = E'_{PxP} \cdot S'_{PxN} + R'_{PxN}$. Now consider the ICA model

presented in (2.3), $X_{PxN} = A_{PxP} \cdot S_{PxN}$. Assume that $P = K$ if one is operating in the

spectral space rather than the PCA space when relating the LMM to the ICA model.

When operating in the PCA space (note that after PCA, the data is also whitened as

explained in section 2.2.5), Y_{PxN} in the LMM given by (2.19) is the same as X_{PxN} in the

ICA model. According to several sources, the mixing matrix A in the ICA model is

interpreted as the endmember matrix, E, and the matrix of ICs, S, in the ICA model is

interpreted as the abundance matrix, S, in the LMM (Chang, 2007:150; Chen, 2007:416;

Du and others, 2006:2-3; Nascimento and Dias, 2005:175-176; Sarigul and Alam,

2007:65650A-2; Varshney and Arora, 2004:125; Wang and Chang, 2006:1587). Notice,

that the ICA model does not include the Gaussian noise term R in the LMM. In a study

conducted by Nascimento and Dias, they conclude that ICA performance increases when

signal to noise ratio, SNR, increases (Nascimento and Dias, 2005:186). Recall the LMM

model in (2.1), $\underline{x} = E \cdot \underline{s} + \underline{r}$, where $\underline{r}$ represents system noise. SNR in the LMM as

defined by Nascimento and Dias is

$$SNR = 10 \cdot \log_{10} \frac{E\left[(E\underline{s})(E\underline{s})^T\right]}{E\left[\underline{r} \cdot \underline{r}^T\right]} = 10 \cdot \log_{10} \frac{\mathrm{cov}(E\underline{s})}{\mathrm{cov}(\underline{r})} \tag{2.91}$$

However, in completing a PCA dimensionality reduction prior to executing the

ICA algorithm, by discarding much of the eigenvectors associated with the 'noise'

eigenvalues prior to projecting onto the PCA space, the magnitude of the noise

component of the LMM, $E\left[r^T \cdot r\right] = \mathrm{cov}(\underline{r})$, is reduced thus increasing the SNR

in the PCA space.

Except for one Gaussian component, a key assumption in the ICA model is that

the ICs are distributed nongaussian. According to Neher and Srivastava who reference an

earlier study of HSI images conducted by Srivastava et al. titled *On advances in*

statistical modeling of natural images, it's well documented that pixel values in natural

images seldom follow Gaussian distributions (Neher and Srivastava, 2005:1365). Thus,

the *X* matrix (holding the pixel values) in the ICA model is most likely distributed

nongaussian. Since, the *X* matrix a linear combination of elements in the *S* matrix, this

implies that the ICs also are most likely distributed nongaussian. Thus, the nongaussian

nature of HSI conforms to one of the main constraints of the ICA model that the ICs are

nongaussian.

2.3.2 LMM Abundance Constraints Relaxed in ICA

So far, it appears that ICA is well suited for use in HSI and a good fit to the LMM of HSI. Thus, it seems to be an acceptable way to simultaneously solve for the endmember matrix and the abundance matrix given only knowledge of the X matrix. However, given the interpretation that the ICs are the abundance fractions in the LMM a few problems arise. As shown in the variable definitions in (2.1) the abundance fractions have two constraints. First for each realization of the random vector $\underline{s}$, since the elements of $\underline{s}$ are the fractional contribution of the endmembers to a particular pixel observation, those elements must sum to one, i.e.

$$\sum_{p=1}^{P} s_{pj} = 1 \ \text{ for } \ j = 1, 2, ..., N \tag{2.92}$$

Thus, the abundance fractions are not truly independent due to this constraint. Second, the abundance fractions are nonnegative. However, in the ICA model the ICs have zero mean, and thus any realization of $\underline{s}$ can have both positive and negative elements. This does not conform to the additive only LMM. Robila suggests in the text by Varshney and Arora modifying the LMM saying,

> it seems natural to modify the model by assuming that the abundance of one endmember in a specific pixel does not provide any information regarding the abundance of other endmembers for that pixel (Varshney and Arora, 2004:125).

This seems to suggest interpreting the abundance fractions (or ICs), as unconstrained weights rather than nonnegative fractions.

In a study by Nascimento and Dias previously mentioned cited by Chang, if one assumes that the abundances are in fact dependent, the unmixing matrix W given by ICA might be far from the true one in the case where there are only a few number of

endmembers (Chang, 2007:163). However, abundance fraction dependency is reduced if the number of endmembers is increased. However, there will always be endmembers incorrectly unmixed (Chang, 2007:172).

For the sake of clarity, the interpretation of ICA model in the context of the LMM in the PCA space is given below.

$$
\begin{pmatrix}
y_{11} & y_{12} & \cdots & y_{1N} \\
y_{21} & y_{22} & & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
y_{P1} & y_{P2} & \cdots & y_{PN}
\end{pmatrix}
=
\begin{pmatrix}
a_{11} & a_{12} & \cdots & a_{1P} \\
a_{21} & a_{22} & & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
a_{P1} & a_{P2} & \cdots & a_{PP}
\end{pmatrix}
\cdot
\begin{pmatrix}
s_{11} & s_{12} & \cdots & s_{1N} \\
s_{21} & s_{22} & & s_{2N} \\
\cdot & \cdot & \cdot & \cdot \\
s_{P1} & s_{P2} & \cdots & s_{PN}
\end{pmatrix}
$$

$$or \qquad\qquad\qquad (2.93)$$

$$Y_{PxN} = A_{PxP} \cdot S_{PxN}$$

Signature of 2ⁿᵈ endmember in the *P* principal components of the PCA space is the 2ⁿᵈ column of the mixing matrix *A*.

N realizations(scores) of the 2ⁿᵈ independent component, s_2, are the *N* weights of the of the 2ⁿᵈ endmember in each of the *N* pixel vector observations.

2.4 Current Practices to Select Target Features from Unmixed Images

Now that this thesis explained in detail the LMM of HSI and an approximate solution methodology to unmix the LMM using ICA to solve for the endmember matrix and the abundance matrix, two images will be presented, unmixed and target abundance maps will be selected using current practices. A brief discussion of those current practices will be presented followed by the examples and analysis.

In the current literature several authors first perform PCA to reduce the dimensionality of the image and decide how many principal components to retain based

on a percentage of variability explained (usually in excess of 99%) or based on some estimate of the distribution of the covariance eigenvalues to determine where the cutoff between signal and noise eigenvalues occurs. Many times a simple scree graph (on a log scale) is visually examined to determine where the 'knee' occurs in the curve rather than a rigorous estimation of the eigenvalue distribution (i.e. Silverstein distribution). ICA is performed on the PCA transformed data matrix to solve for the ICs, or the rows of the abundance matrix in terms of the LMM. The individual rows of this matrix can be displayed as a series of abundance maps (see Figure 2-1) after normalized to a gray scale. To identify abundance maps containing targets, one ranks the maps in decreasing order according to the kurtosis value, KV, of the independent component that was used to form the map. If one assumes that targets represent a small class, i.e. are relatively rare in the image, the kurtosis will be high compared to abundance maps of larger classes. As mentioned previously, the cutoff between target and non-target is decided via a scree graph of the KV's. Just as in the scree graph to determine dimensionality, the first point where the slope significantly changes, the 'knee', distinguishes between target and non-target. A second criteria in addition to KV to distinguish targets suggested recently by Koo calculates a mean silhouette value, MSV, for each abundance map. First 2-class clustering is performed on each abundance map (K-means clustering, with K=2). For more information on K-means clustering see Dillon and Goldstein, pages 188-190. The idea is to find the abundance map that separates best into two classes, namely, background and target. After completing the K-means clustering, silhouette plots show how well the two classes are clustered. Each silhouette value is in the range of -1 and 1. When the classes are well clustered, the silhouette values should be close to 1. Thus, the

target map should have a MSV (mean of all the silhouette values for each cluster) close to 1. Koo states that KV and MSV act as a check-and-balance filtering process, since it is possible that two well separated non-target classes within a map may also produce a very high MSV. An abundance map containing the real targets should be among the features with the highest KV and have a MSV near 1. Koo's two-phased filtering method selects target maps that have a relatively high KV and a MSV close to 1 (Koo, 2007:46-47).

Below in Figure 2-15 are two HSI images image taken from the United States Air Force's Airborne Remote Sensing Program (ARES) using the Hyperspectral Digital Imagery Collection Equipment (HYDICE) sensor during the Forest Radiance I and Desert Radiance II data collection efforts. The sensor utilizes 210 narrow spectral bands covering the ultraviolet, visible and near-infrared portions of the electromagnetic spectrum (0.4 – 2.4 μm). Analysis by T. Smetek, determined which bands where atmospheric absorption bands and/or noise bands. After these bands were removed, 145 bands remain. The top row is the RGB images and the bottom row is the truth mask of the targets. These images were analyzed in the work by Koo and this section will attempt to reproduce his results using his methodology.
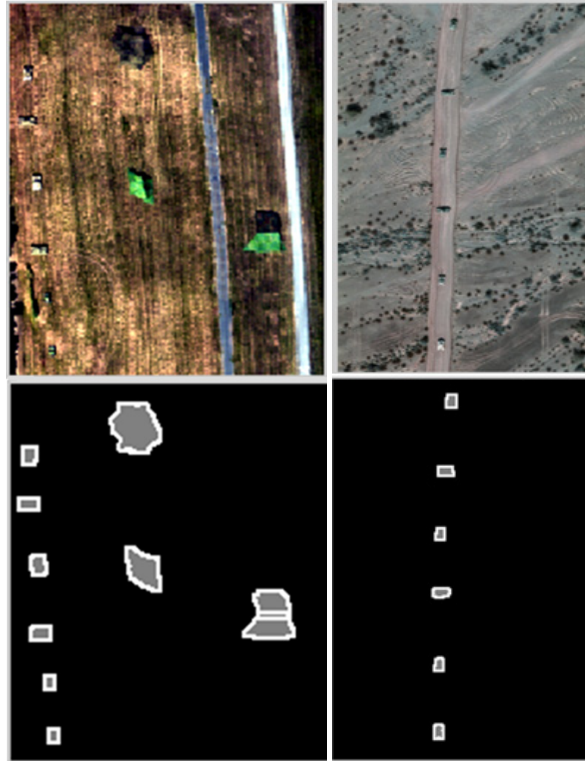
Figure 2-15. ARES 1F, ARES 1D with Truth Masks

Koo retained 12 and 14 principal components in ARES 1F and 1D respectively accounting for 99.83% and 99.84% respectively of the total variance. These dimensionality reduction decisions will be followed in an attempt to reproduce the results. In his thesis, Koo reports the following settings in FastICA.

- Symmetric Orthogonalization

- Function g of choice: $g = y^3$

  - Default. Uses kurtosis approximation of negentropy. g is the derivative of $G = y^4$ without the coefficient of 4

- Secondary refinement using function $g = \tanh(y)$

  - When initial convergence is achieved, the entire process will repeat using the refinement function for a second convergence

Note the g functions mentioned here refer to the choices
given in equations (2.85), (2.86), and (2.87) of this thesis.

- Initial starting point: Random

- Stopping criterion: 10^{-6}

Although not documented in Koo's research, via correspondence from Koo, the

following settings were used to calculate the K-means clustering:

- Distance measurement: squared Euclidean

- Method of choosing 2 initial cluster centroid positions: uniformly at random

- Action to take if cluster loses all its members: singleton (creates a new cluster consisting of the one point furthest from its centroid)

For each image in Figure 2-15, the ICs solved for by FastICA will be formed into

abundance maps and presented followed by the results from the K-means clustering and

mean silhouette plots. Next a summary table of the KVs, MSVs, and computation time

will precede a discussion of the results. Koo's results for the same image will then be

presented.
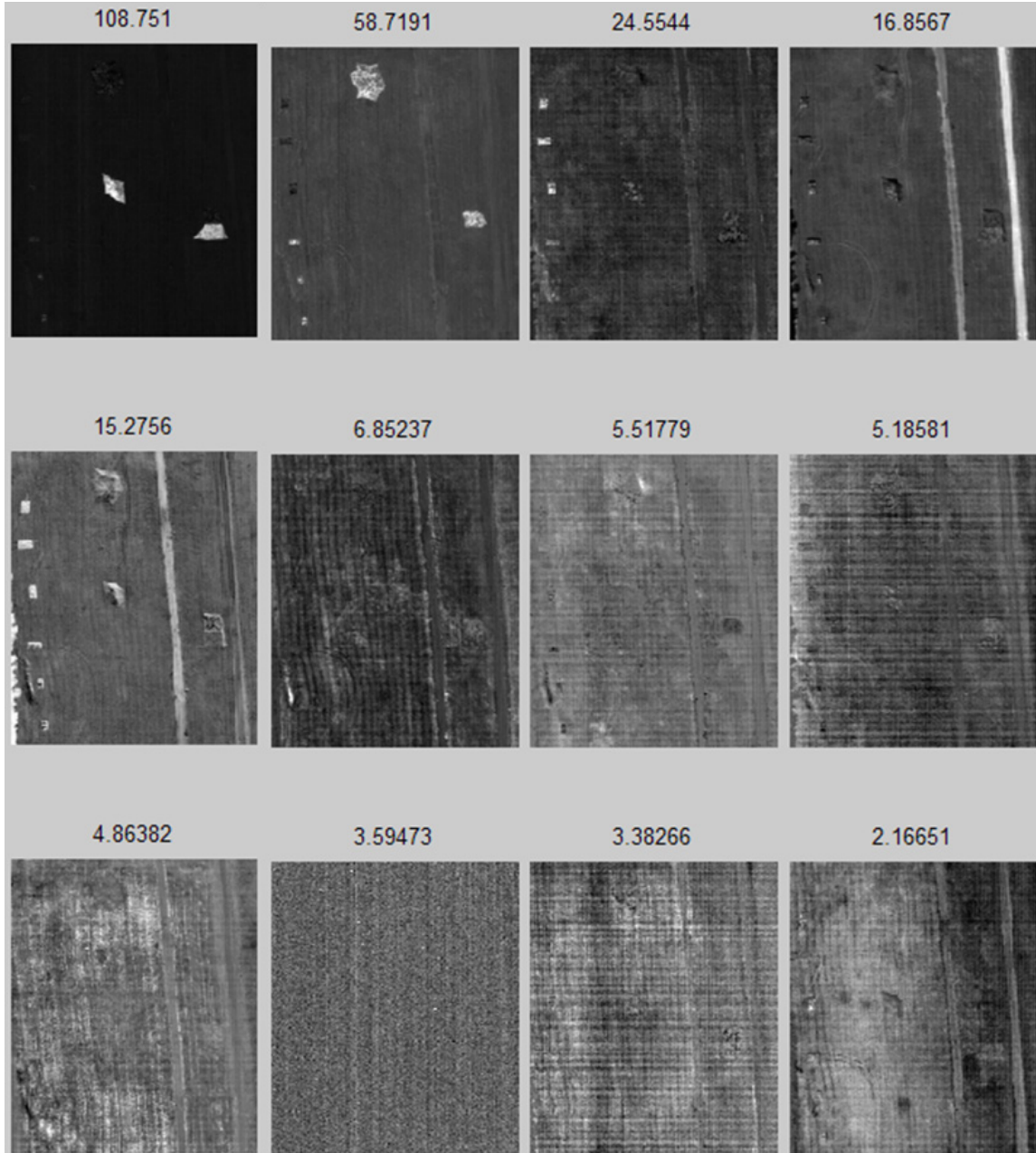
Figure 2-16. ARES 1F: Abundance Maps from ICs Sorted by Kurtosis Value
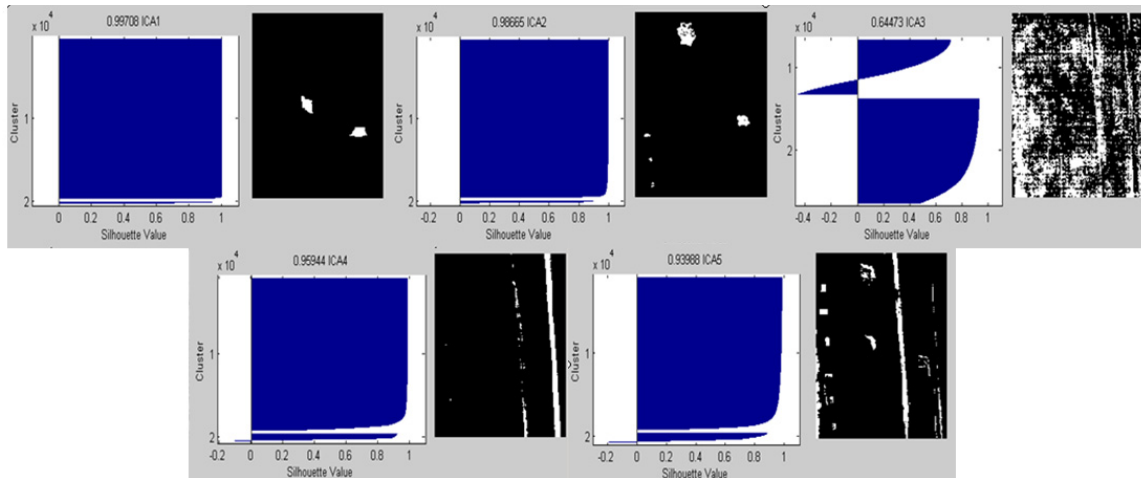
Figure 2-17. ARES 1F: Silhouette Plots (Top 5 Maps by KV) with MSV and Binary Images Produced by K-means to the Right of Each Plot

Table 2-1. ARES 1F Summary: IC's KV and MSV and overall Run Times

ARES 1F		
Maps Sorted by KV	**KV**	**MSV**
Target 1	108.7513	0.99708
Target 2	58.7191	0.98665
Target 3	24.5544	0.64473
4	16.8567	0.95944
5	15.2756	0.93988
6	6.8524	0.71471
7	5.5178	0.66855
8	5.1858	0.71813
9	4.8638	0.69517
10	3.5947	0.6991
11	3.3827	0.6932
12	2.1665	0.76627
Time to compute ICA and Sort ICs by KV	7.49 sec	
Time to compute K-means	2.09 sec	
Time to compute MSV	636.822 sec	

The first three abundance maps isolate the targets and the fifth map contains targets but the class is not separated from the non-targets classes of trees and road. The three abundance maps that isolate the targets have the three highest KVs. Further, abundance maps 1 and 2 have the highest MSV. However, the MSV of the third abundance map is not close to 1, but non-target maps 4 and 5 have MSVs higher than 0.9 as shown in Table 2-1. Further, note the computational expense of 636.8 seconds, in excess of 10 minutes, to calculate the 12 MSVs for this image. Notice the maps with the highest MSV, maps 1, 2, 4 and 5, cluster nicely into 2 classes (background and outlined objects) evidenced by the K-means binary images in Figure 2-17.

It should be noted that the K-means clustering algorithm is stochastic and therefore can yield different results from one run to the next. To ensure the K-means binary images in Figure 2-17 were not just a rare occurrence of the algorithm, it was re-run 20 times on the same ICs that FastICA yielded. The binary images in Figure 2-17 resulted in every run. Further the ICA algorithm is also stochastic, so it was re-run 10 times, followed by the K-means algorithm and, still, the binary images in Figure 2-17 resulted in every run.
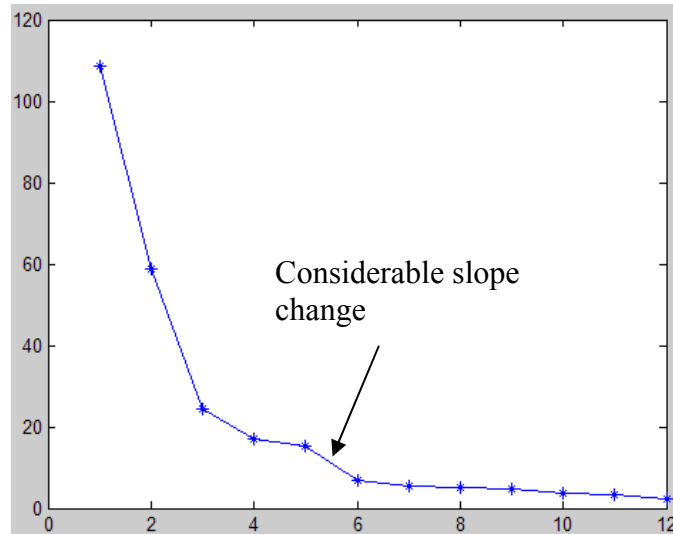
Figure 2-18. Scree Graph of Kurtosis Values of ARES 1F Maps

Analysis of a scree graph of the kurtosis values of the abundance maps (suggested technique by Robila) in Figure 2-18 shows that considerable change occurs between abundance maps 5 and 6. This considerable slope change is suggested to be the breakpoint between target maps and non-target maps. Thus, assuming potential target maps to be maps 1 though 5, Koo's two-phased filtering method that selects the maps with the highest KV and MSV would incorrectly select maps 4 and 5 as target maps and incorrectly omit abundance map 3. The results given by Koo for this image are shown below in Table 2-2 and Figure 2-19. As previously mentioned, due to the stochastic nature of the FastICA algorithm and K-means clustering, results can differ. As seen in Table 2-2, the two-phase filtering in Koo's research is successful at identifying the target maps. Note, that the abundance maps in Figure 2-16 and Figure 2-19 are quite similar. This thesis's results successfully reproduced Koo's results for target maps 1 and 2, but not for target map 3. Perhaps Koo's result was a positive chance occurrence that

validated his method or there are alternate K-means settings that yield a MSV close to 1

for target map 3.

Table 2-2. Koo Results for ARES 1F (Koo, 2007:81)

Features Sorted by Kurtosis	Kurtosis Value	MSV
1	104.4346635	0.9962
2	36.04489407	0.9762
3	22.1025491	0.9771
4	15.66723472	0.8633
5	15.42182144	0.5428
6	13.39154261	0.7272
7	12.8694271	0.6119
8	8.48488004	0.6863
9	4.70540516	0.5382
10	3.715898291	0.6813
11	3.545306664	0.6897
12	2.329904951	0.7506
* Red Values Indicate the Target Features		

Figure 2-19. ARES 1F: Abundance Maps From ICs Sorted by KV from Koo (Koo, 2007:82)
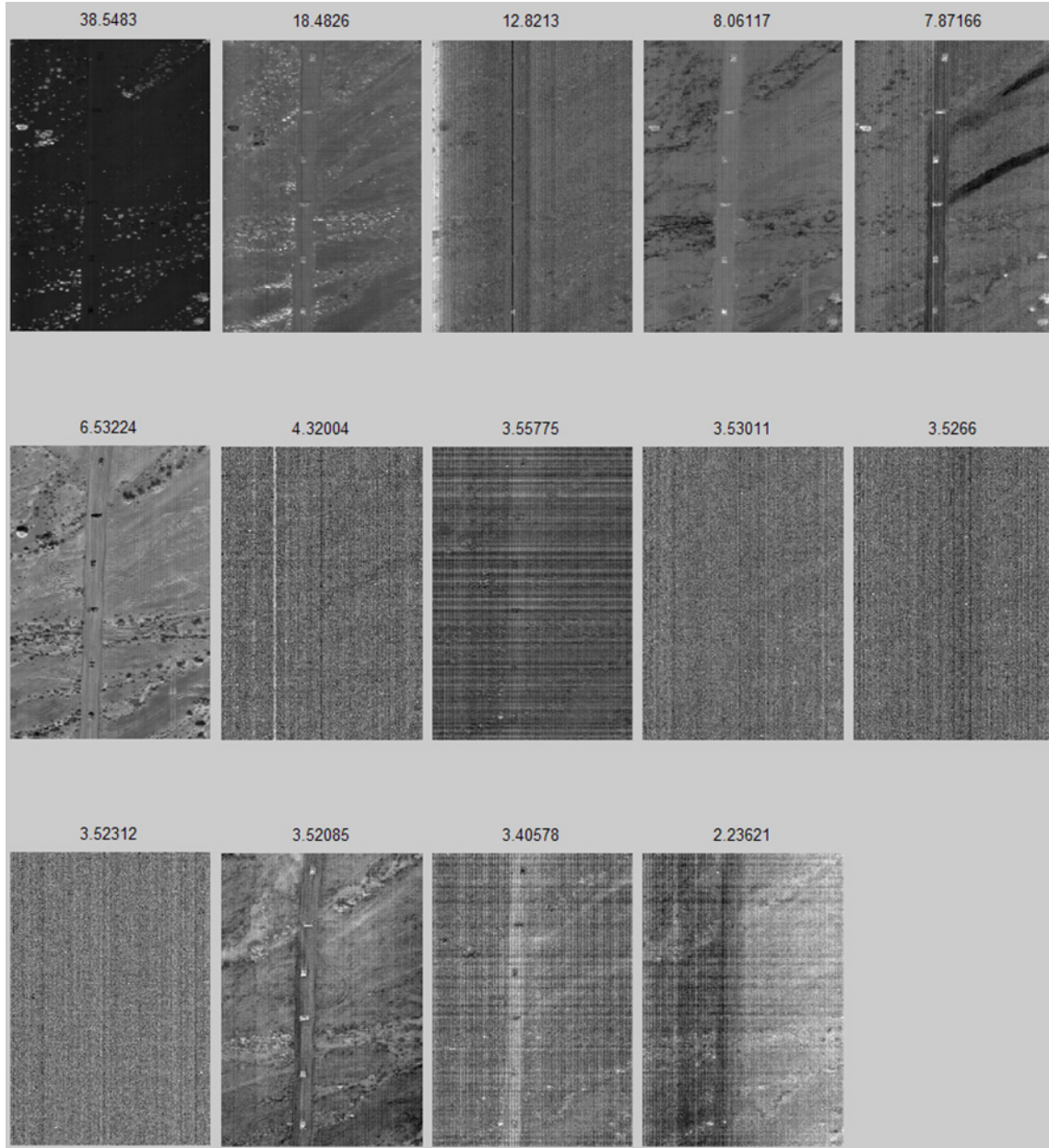
Figure 2-20. ARES 1D: Abundance Maps from ICs Sorted by Kurtosis Value Using Secondary Refinement

Notice, maps 4 and 5 in Figure 2-20 highlight the targets. However, the target

class is not separated from non-target classes that exist in the bottom right and upper left

of the maps. FastICA was run 10 times and each time the target class did not separate out into its own map as shown in maps 4 and 5 (in order of KV) in Figure 2-20. Instead of running ICA twice, using $g = y^3$ first and then running ICA again using $g = \tanh(y)$ on the solved ICs (i.e. the secondary refinement previously mentioned), ICA was run 10 times using just $g = \tanh(y)$ with no secondary refinement. The same negative results were observed as in Figure 2-20. Thus, the secondary refinement technique is superfluous here. Finally, ICA was run 10 times using $g = y^3$ with no secondary refinement. Each time the target class was isolated successfully in a single abundance map (map number 4 sorted by KV) as shown in Figure 2-21.
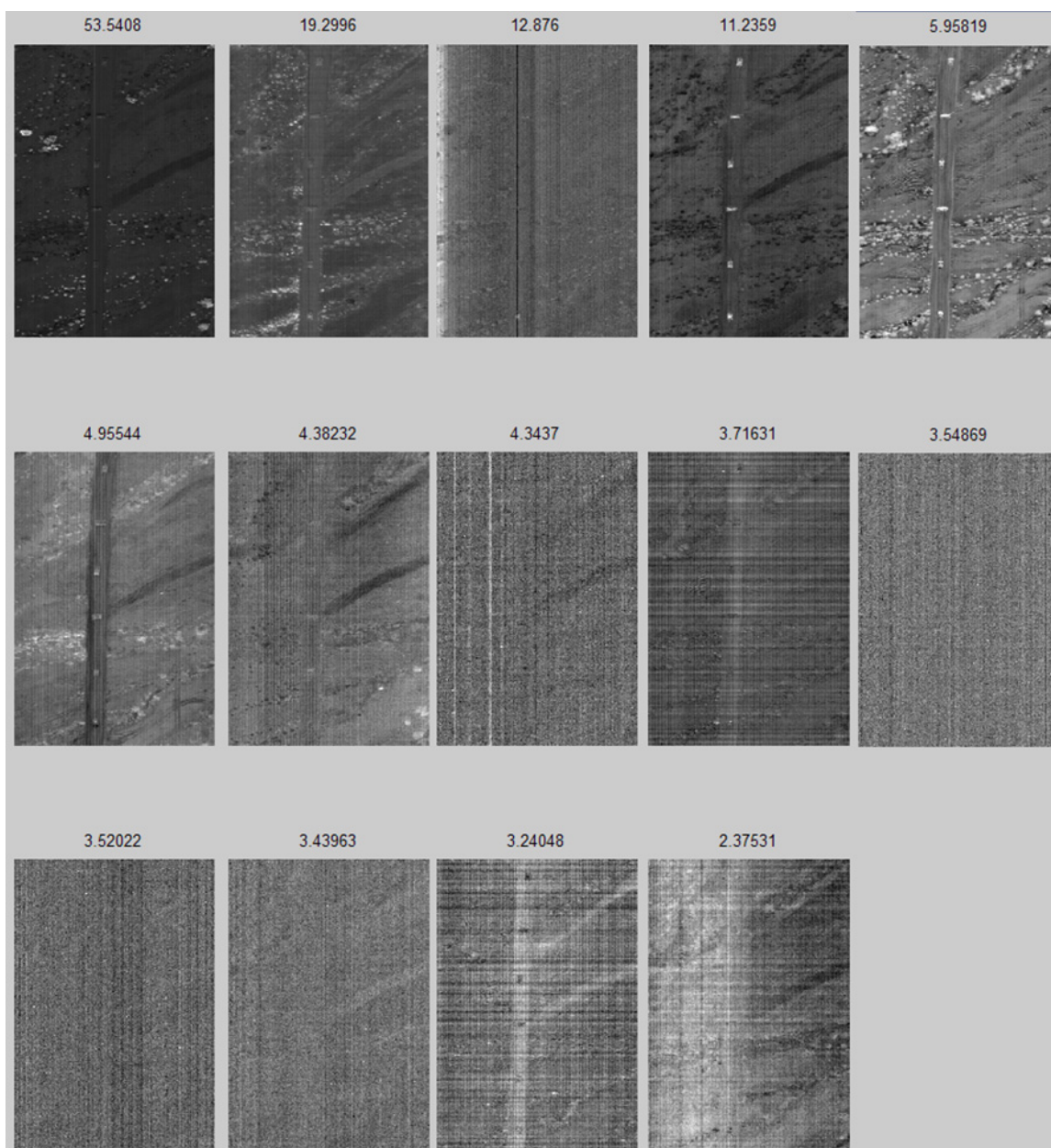
Figure 2-21. ARES 1D: Abundance Maps from ICs Sorted by Kurtosis Value Using $g = y^3$ without Secondary Refinement
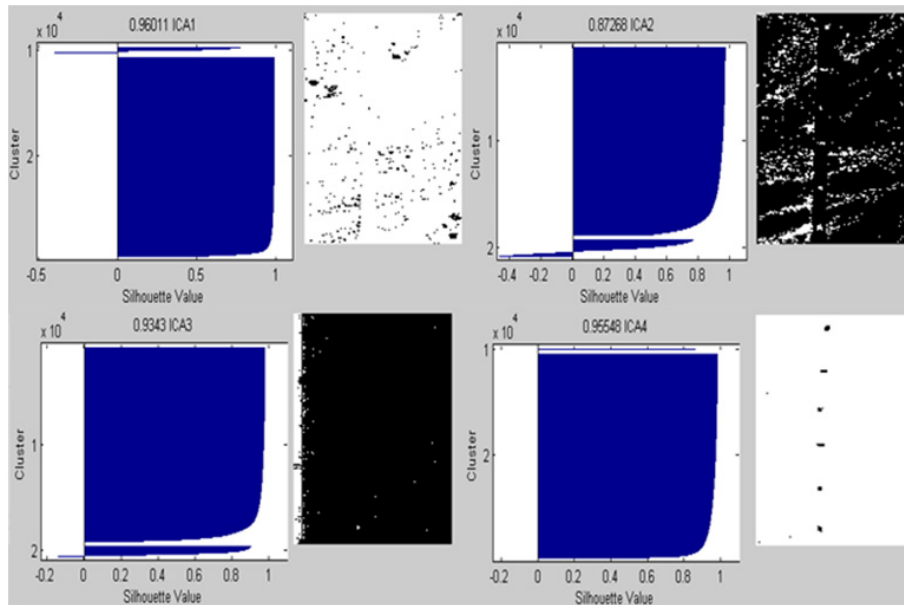
Figure 2-22. ARES 1D: Silhouette Plots (Top 4 by KV) with MSV and Binary Images Produced by K-means to the Right of Each Plot

Table 2-3. ARES 1D Summary: IC's KV and MSV and overall Run Times

ARES 1D		
Maps Sorted by KV	KV	MSV
1	53.5408	0.9601
2	19.2996	0.8727
3	12.876	0.9343
Target 4	11.2359	0.9555
5	5.9582	0.7276
6	4.9554	0.6377
7	4.3823	0.69
8	4.3437	0.6807
9	3.7163	0.6976
10	3.5487	0.6968
11	3.5202	0.6987
12	3.4396	0.7015
13	3.2405	0.7008
14	2.3753	0.7529
Time to compute ICA and Sort ICs by KV	8.57 sec	
Time to compute K-means	5.28 sec	
Time to compute MSV	2724.14 sec	45.4 min

Contrary to ARES 1F, the target map is not the map with the highest KV. Although, as shown in the scree graph of KVs in Figure 2-23, a considerable slope change occurs between maps 5 and 4. Thus, maps 1-4 would be considered candidate target maps by this criteria, keeping the true target map. As see in Table 2-3 and Figure 2-22, applying the second filtering technique, maps 1, 3, and 4 have MSVs in excess of 0.9 and cluster nicely into 2 classes. Since all are among the highest KV maps, they would be selected as target maps by the two-phase filtering technique. In the 10 runs of FastICA using $g = y^3$, maps 1 and 3 sorted by kurtosis yielded high MSVs in excess of 0.9. Further, in the 10 runs using the secondary refinement, and in the 10 runs just using $g = \tanh(y)$, maps 1 and 3 yielded MSVs higher than 0.9. Thus, the result that maps 1 and 3 would be classified as target maps using the two-phase filtering technique is not a rare occurrence. Again, as shown in Table 2-3, note the computational expense of 2724.14 seconds or 45.4 minutes to calculate the 14 MSVs for this image.
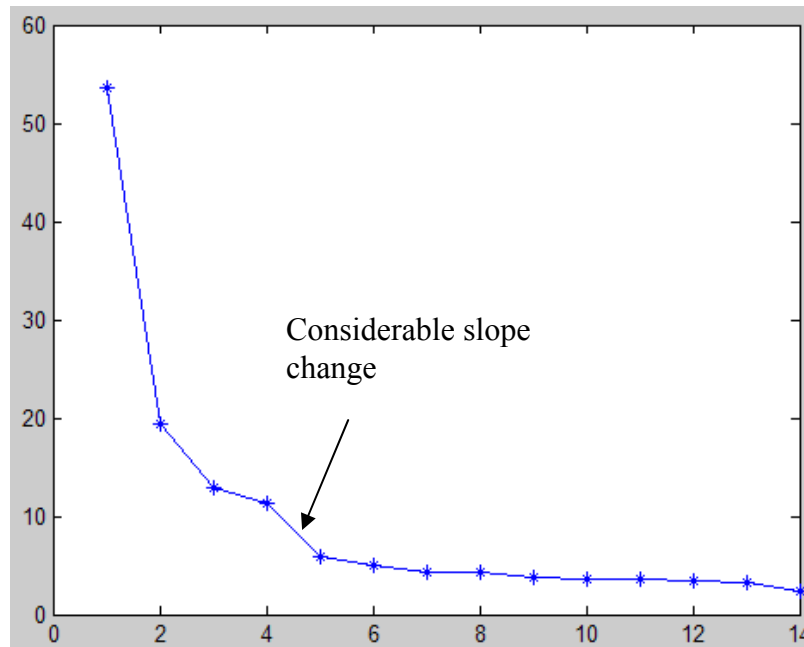
Figure 2-23. Scree Graph of Kurtosis Values of ARES 1D Maps

Examining the incorrectly selected non-target maps 1 and 3 a few observations can be made. Map 1 appears to be a large rock or large bush class. The large rocks or bushes (difficult to determine from the RGB image) in the upper left and lower right along with the smaller rocks or bushes throughout the image constitute a class with a spectral signature significantly different from the desert sand background. Further, of the 57,909 pixels in this image, approximately 2,100 pixels belong to this class or 3.6% of the image. Thus, this class also represents a small class, meeting the assumptions underlying an anomaly detector. Hence, one would expect a high MSV and KV for this class. Map 3 shows a thick bright line on the far left of the image which appears to be some kind of artifact from the HSI sensor. However, this is only speculation. Regardless, it is also distinct from its background and is a small class resulting in high a MSV and KV.

Also, of important note is that the clustering of map 4 (true target) resulting in a MSV of 0.9555 occurred 80% of the time in an experiment repeating K-means clustering 20 times on this independent component (map 4). As shown in Figure 2-24 below, 5 out of the 20 runs resulted in a MSV of 0.66 and the corresponding binary image. This highlights the stochastic nature of the K-means clustering algorithm.
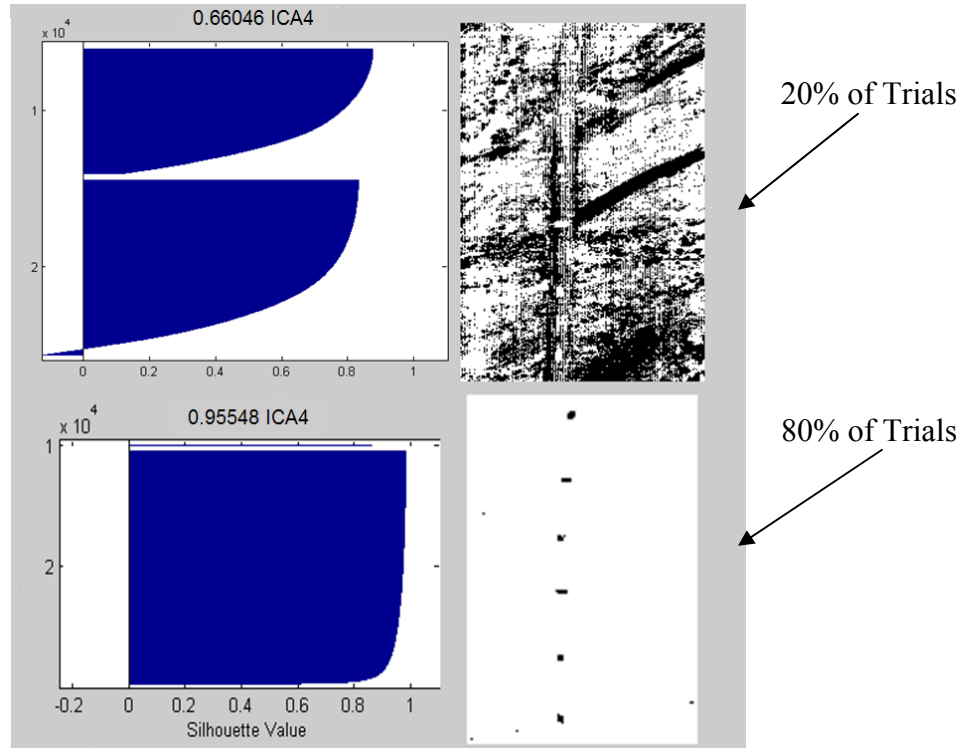
**Figure 2-24. Map 4 (True Target) MSV and Binary Image Produced
in 20% and 80% of the Trials**

The results given by Koo for this image are shown in Table 2-4 and Figure 2-25. As shown in Table 2-4, Koo's two-phase filtering approach correctly selected map 4 as the only map with a MSV in excess of 0.9.

Table 2-4. Koo Results for ARES 1D (Koo, 2007:72)

Features Sorted by Kurtosis	Kurtosis Value	MSV
1	27.38693961	0.4724
2	14.64158718	0.4621
3	12.26042991	0.4013
4	10.42031576	0.9482
5	9.95437817	0.6651
6	6.20765919	0.7077
7	5.806268984	0.6768
8	4.782459699	0.6480
9	4.741022114	0.6921
10	4.369351272	0.6815
11	3.820212294	0.6920
12	3.570611975	0.6976
13	3.536070957	0.6962
14	2.299170966	0.7633
* Red Values Indicate the Target Features		
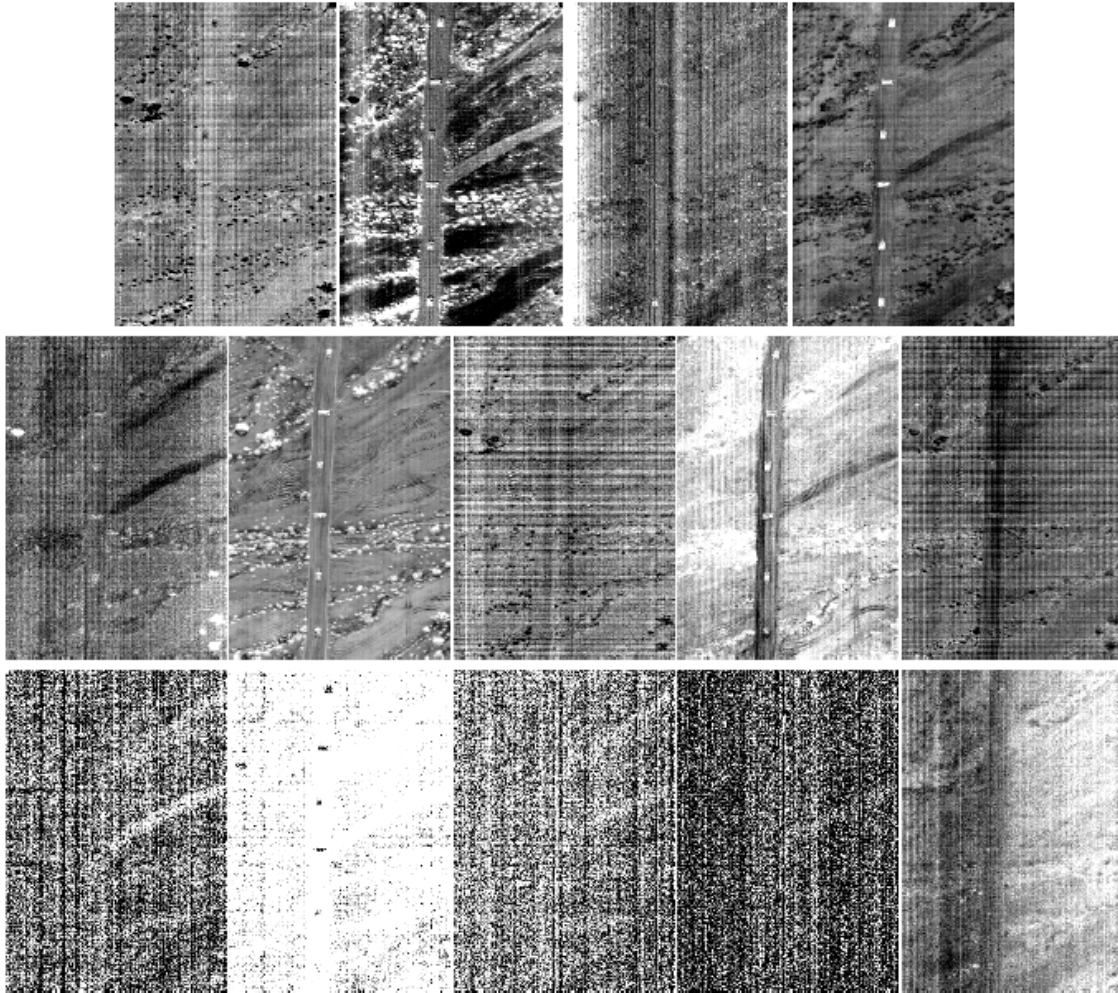
**Figure 2-25. ARES 1D: Abundance Maps From ICs Sorted by KV
from Koo (Koo, 2007:73)**

Notice map 1 (top left) in Figure 2-25, the negative of map 1 in Figure 2-21, does not isolate the class of rocks or bushes quite as well as evidenced visually (more noise) and by its KV of 27.4 versus the map 1 that occurred 30 times in Figure 2-21 with a KV of 53.5. Map 3 in Figure 2-25 is quite similar to map 3 in Figure 2-21 showing the same artifact of a thick bright line on the left. Figure 2-25 map 3 KV is 12.26 and Figure 2-21 map 3 KV is 12.88. However, K-means clustered this map differently in Koo's research causing it to have a low MSV. One must conclude that either the results Koo achieved

were either a positive chance occurrence where FastICA did not did not produce a well isolated rock/bush feature enabling his method to work well or some alternate settings in FastICA, not documented, were used that consistently unmixed the image in the fashion presented in Figure 2-25. The settings used by this author consistently unmixed the image as either the way it is presented in Figure 2-20 using the secondary refinement or Figure 2-21 using just $g = y^3$. Further for map 3, since it has a similar KV in both Koo's results and this thesis's results, either this was also chance occurrence where K-means did not cluster well or alternate settings in K-means were used than the ones reported to this author via correspondence.

Conclusions on Current Practices to Identify Target Maps

Ranking ICs (i.e. row vectors that comprise the abundance matrix) by kurtosis value as suggested in a myriad of current literature and using a scree graph as suggested by Robila to identify the break point between potential target classes and non-target classes appears to be a relatively effective initial processing technique to identify target maps from these example HSI images. However, identifying the point of considerable slope change via inspection could be problematic and not reproducible from one analyst to the next. A quantifiable, repeatable algorithm is needed to identify the breakpoint in the kurtosis scree plot. Koo's novel approach of using K-means clustering, followed by a mean silhouette calculation appears to be inconsistent at nominating the correct target maps when comparing results of this research to Koo's. Thus, given the inconsistent results with just two example images, the new two-phase filtering approach does not appear to be a robust method. Further, the computational cost of computing the MSV for the images is high, 10 and 45 minutes for ARES 1F and ARES 1D respectively. Thus,

even if it identified target maps consistently, it is not feasible as a real time target detector. Approaches to nominate target maps that work consistently and quickly in addition to or instead of the KV sort are still desired.

2.5 Current Practices to Identify Target Pixels from Selected Target Maps

Assuming one is able to determine which of the abundance maps the target maps are, the next problem to solve is the identification of which pixels in the maps are the target pixels. Recall that each abundance map corresponds to a row vector in the abundance matrix that has been converted to a gray scale (normalized to values between 0 and 1) and reshaped into the original image pixel length by width. If one where to take the raw abundance matrix row vector corresponding to a particular abundance map and graph its elements, the signal in Figure 2-26 would be observed. This signal represents the pixel scores on this particular independent component (abundance matrix row vector). One can observe pixel scores that rise above the background. These pixels represent the pixels corresponding to the targets in the map to the right in Figure 2-26. The problem is to locate the threshold that separates background pixels from target pixels. Note that due to the ambiguity of the sign of the independent component, this signal may sometimes be reversed with the signal pointing down. To alleviate this, one can just reverse the sign of the signal if the absolute value of the minimum pixel is larger than the max pixel (practiced by this author), or as suggested by in the citation by Robila, one can calculate the skewness of the signal and reverse the sign if skewness is negative.
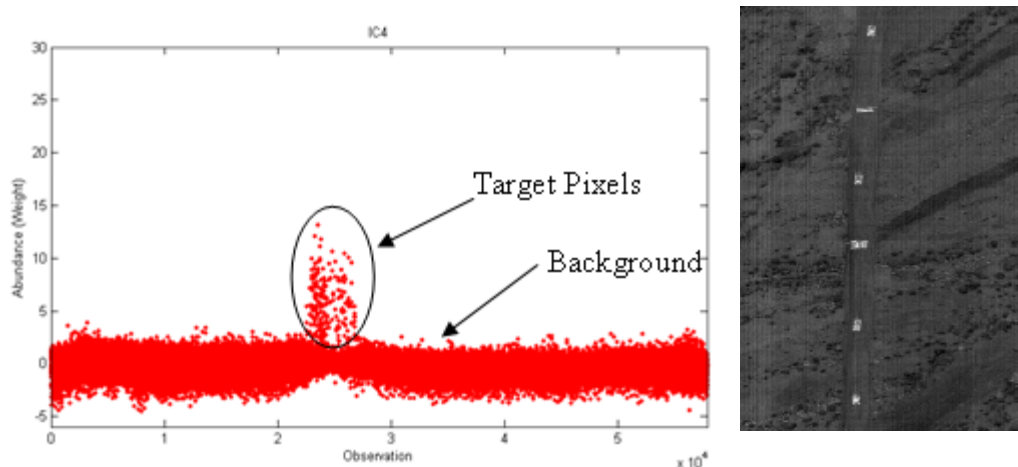
**Figure 2-26. ARES 1D: Abundance Row Vector Elements
Corresponding to the Abundance Target Map**

According to Koo's research, a threshold at a certain percentile of the data was

used. For map 4 in ARES 1D, shown above in Figure 2-26, Koo tested the 98th, 99th,

99.5th, and 99.8th to find the percentile that yielded zero false positives according to truth

information. He found that the 99.8 percentile (all scores above 3.87 classified as target)

yielded no false positives (Koo, 2007:76-78). For ARES 1F, for the first three maps by

KV, he found that the 98.8th, 98.7th, 99.8th percentile, scores above 2.385, 2.911, and

5.592 for maps 1, 2 and 3 respectively yielded no false positives. As one can see from

these results, the percentile threshold to separate target from background is different from

image to image. Thus, one cannot simply set one particular percentile threshold to be

used for all images and effectively identify targets. Further these image dependent

percentile thresholds were supervised decisions made with full a priori knowledge of the

targets which is not a practical approach for an unsupervised algorithm. The decision for

the score above which to separate target and non-target must be determined dynamically

and unsupervised, i.e. some piece of information in the abundance row vector yields a

clue as to how to choose this score autonomously if one is to hope to create an unsupervised detection algorithm.

One such approach to determine the score that separates target from non-target is to create a histogram of the chosen target independent component and find the first point in the tail of the histogram where the number of pixels in a particular bin is zero (Chiang and others, 2001:1383). This represents the first point of clear separation from the scores of the pixels that comprise potential targets to those that comprise objects that are in the background. Given the ambiguity of the sign of the ICs, one must determine in which direction of the signal (extreme negative scores or extreme positive scores) the target pixels are concentrated. To make this determination, Robila calculates the skewness of the signal (Robila and Varshney, 2002:178). Positive skewness means targets are concentrated in the positive direction (right tail of the histogram) and vice versa for negative skewness. The scores in Figure 2-26 have a positive skew so the right tail of the histogram will be analyzed.

To illustrate this approach, a histogram is created in Figure 2-27 using the scores from Figure 2-26. Not mentioned in the cited article by Chiang is the bin width to use to create the histogram. For this example bins were used ranging from -15 to 15 with a bin width of 0.05. Recall that ICA standardizes scores to have mean of zero and variance of one. Thus, the scores are concentrated heavily around zero and we would expect background pixels to be concentrated around this mean. To identify the point that distinguishes the background from target, we zoom in on the right tail in Figure 2-27 (b) and the first point with a bin value of zero, 4.1, is chosen as the threshold.
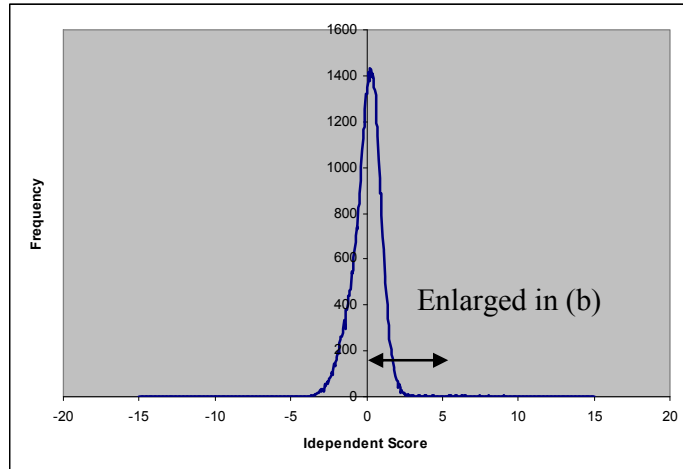
(a)
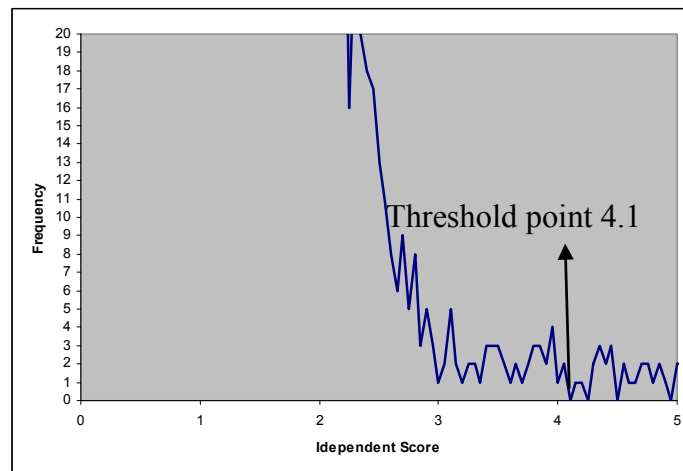
(b)
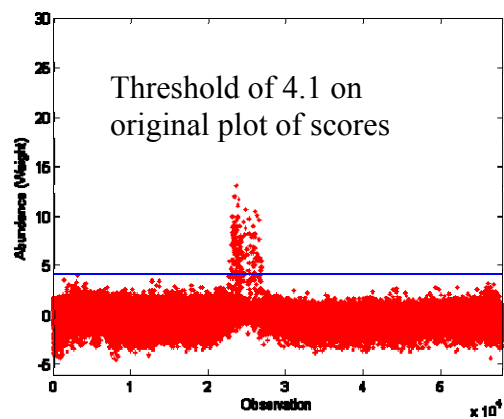
(c)

Figure 2-27. Histogram of Scores (a) and (b) from Map 4 of ARES 1D and Threshold Determined (c)

Applying this threshold one gets the following binary image below. Given truth knowledge, we can see a false positive rate of zero and a true positive rate (percentage of true target pixels identified) of 0.65.
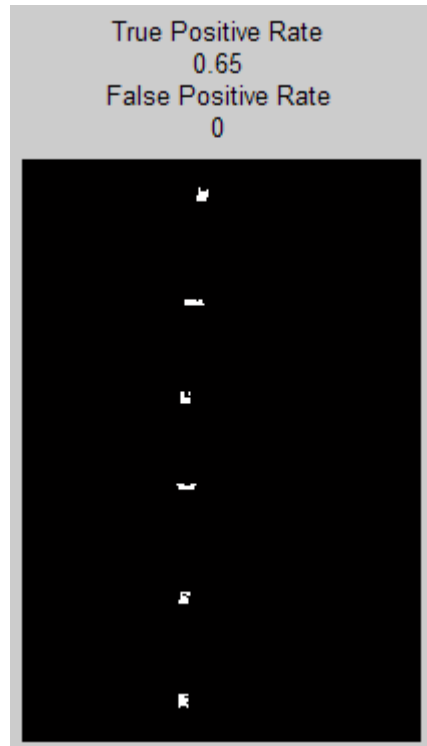
Figure 2-28. ARES 1D: Target Detection Binary Image using Histogram Method with Bin Width of 0.05

However, what if a histogram bin width size of 0.1 is used or width of 0.01? Figure 2-29 shows the results of each. Using a bin width size of 0.1, the first bin to have zero pixels was 8.7. For a bin width of 0.01 the first bin to have zero pixels was 2.37.

True Positive Rate 0.835938	True Positive Rate 0.0978723
False Positive Rate 0.0016131	False Positive Rate 0

Bin Width of 0.01	Bin Width of 0.1
Threshold 2.37	Threshold 8.7
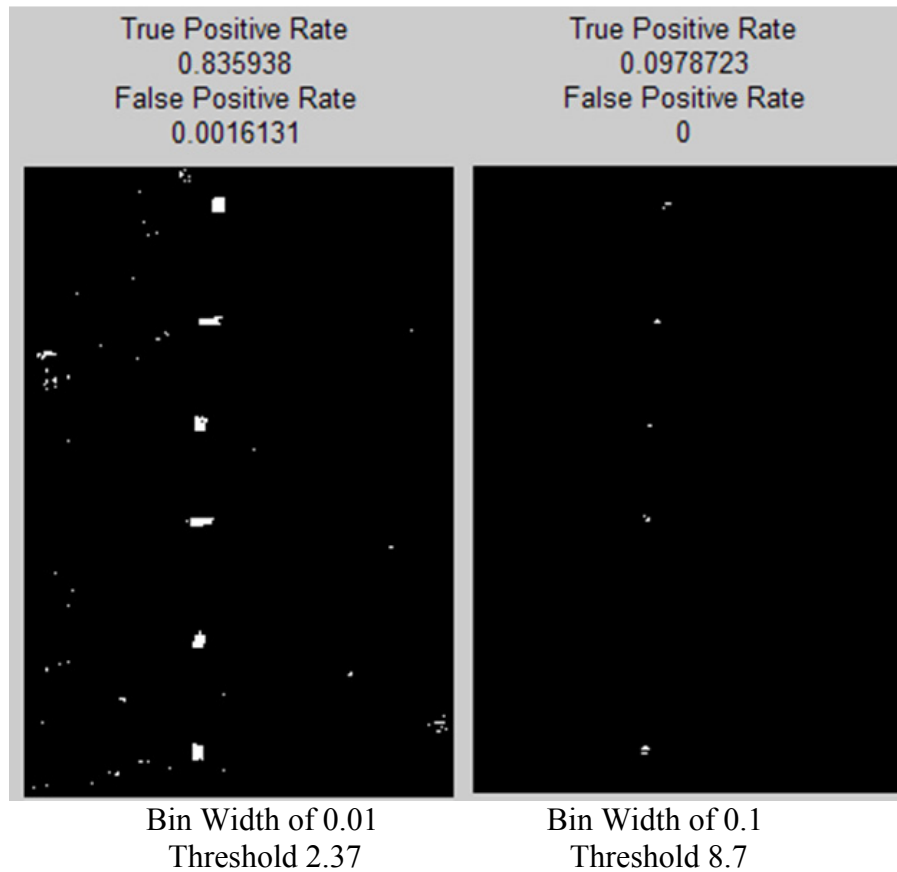
**Figure 2-29. ARES 1D: Target Detection Binary Image using
Histogram Method with Bin Widths of 0.1 and 0.01**

Thus, how well the histogram method works is sensitive to the choice of bin width. One

questions to ask is whether or not there is a bin width that works well at determining the

threshold for all images. For this image, a width of 0.05 appears to work reasonably well.

Using a bin width of 0.01, 0.05, and 0.1 was tested on the top 3 maps sorted by

KV from ARES 1F. Figure 2-30 below shows the binary images resulting from applying

this method to each of the three maps and combining the binary maps to form a single

detection image for each bin width. Again, a bin width of 0.05 appears to work

reasonably well at determining the threshold between target and background.

True Positive Rate	True Positive Rate	True Positive Rate
0.957848	0.882516	0.45183
False Positive Rate	False Positive Rate	False Positive Rate
0.00546782	0.000678449	3.3873e-005

| 0.01 Bin Width | 0.05 Bin Width | 0.1 Bin Width |
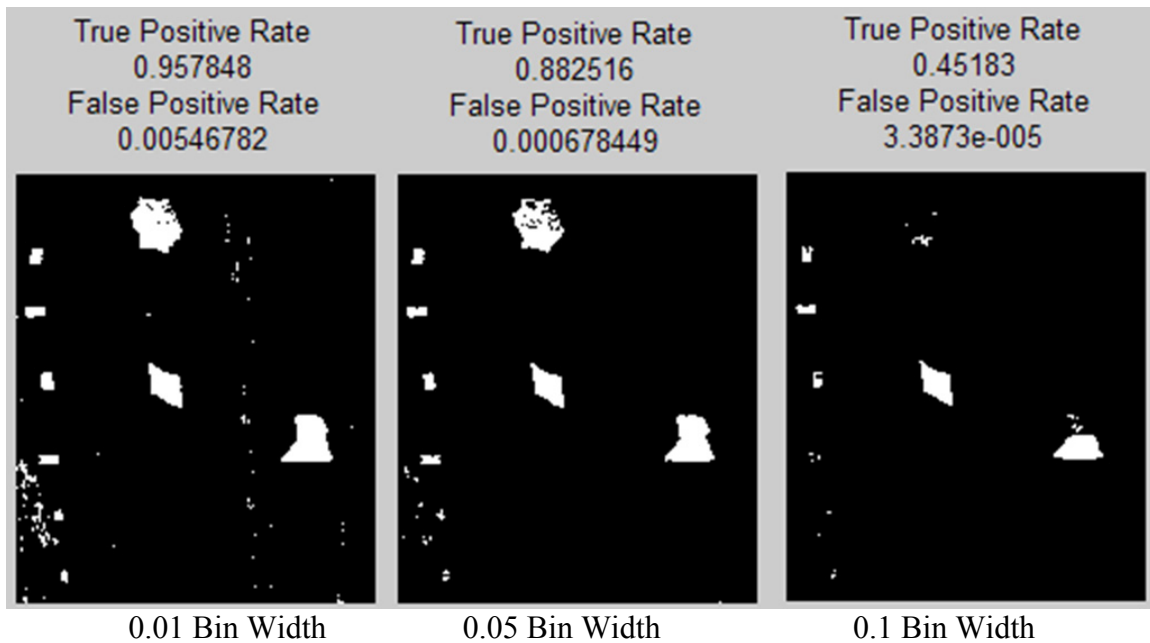
Figure 2-30. ARES 1F: Target Detection Binary Image using Histogram Method with Bin Widths of 0.01, 0.05, and 0.1

In Figure 2-31, the thresholds determined using the histogram method for maps 1, 2, and 3 are shown. The lower thresholds are associated with using a bin width of 0.01 and highest thresholds are associated with a bin width of 0.1. Notice for the second map, bin width choice could have high variability in threshold selection for some signals.
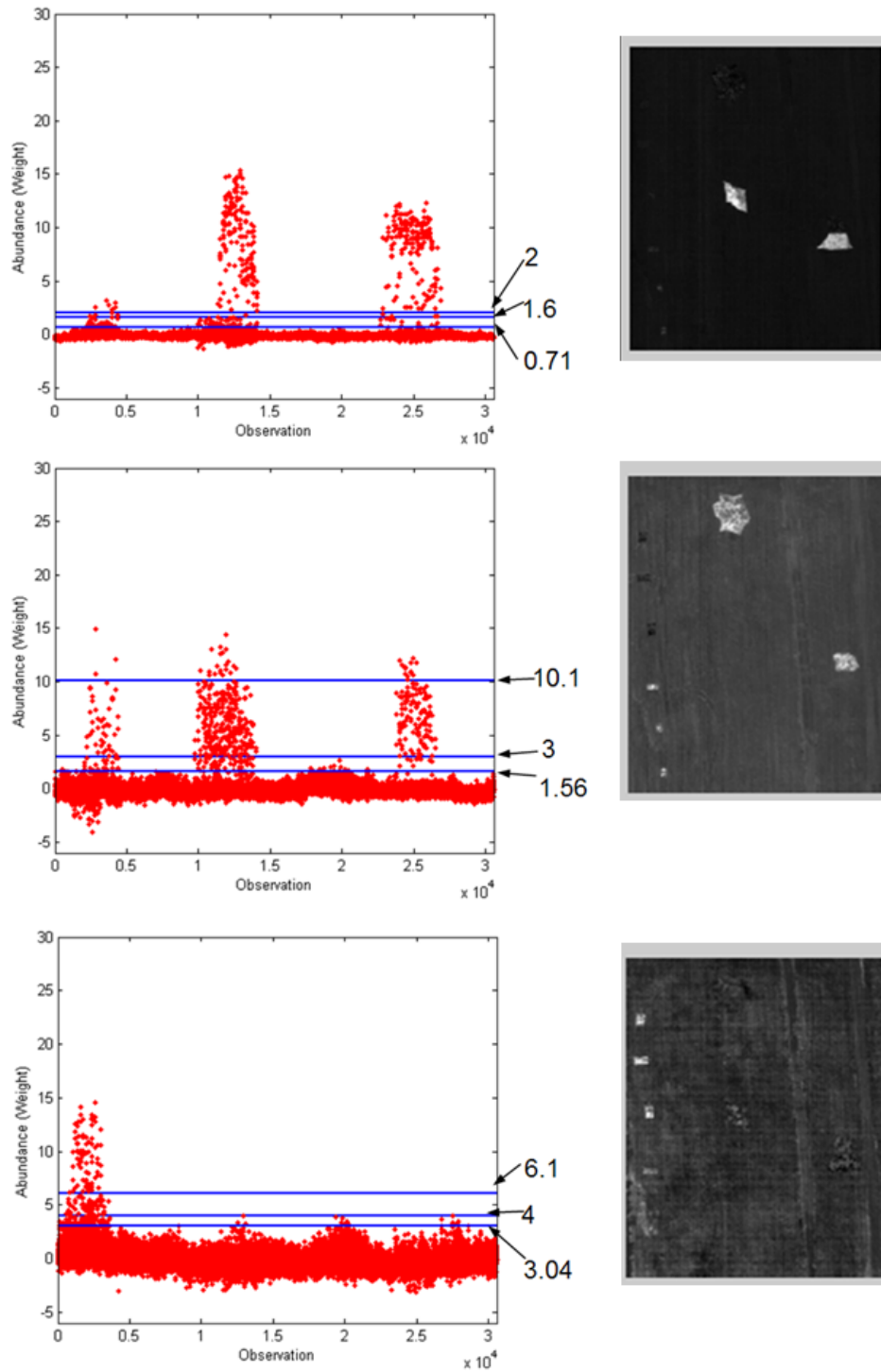
Figure 2-31. ARES 1F: Target Thresholds for Maps 1, 2, & 3 Using Histogram Method with Bin Widths of 0.01, 0.05, and 0.1

Conclusions on Current Practices to Identify Target Pixels

As shown from Koo's work, a particular constant percentile threshold to apply to all images would not be an effective way to identify target pixels due to the inherent difference in independent component scores (i.e. signals) from one image to the next. The zero-detection histogram method (term coined by Chiang et al.), i.e. choosing the first point with a bin value of zero, dynamically chooses the threshold with the critical decision being histogram bin width size. For these two images a bin width of 0.05 works reasonably well. However, as evidenced in Figure 2-31 in the second map, since bin width choice can yield high variability in threshold determination, more test images are needed to see the robustness of the 0.05 bin width choice.

III. Methodology and Test Image Experimentation

3.1 Process Flow for Detector

As previously mentioned, HSI data is made into an image cube of length by width by reflectance values per spectral band (really just a data matrix of three dimensions) and then reshaped into a two dimensional data matrix of bands by pixel observations. After this arrangement of the data, an HSI detection algorithm can be broken down into four general phases.

First, the dimensionality (determined by the number of spectral bands) must be reduced (referred to as feature extraction), most commonly via PCA. Some decision is made as to the number of principal components retained (which will be assumed to be the number of different spectrally distinct endmembers in the image) via percentage of variability explained or some estimation of the noise floor of the eigenvalues, as previously mentioned in the citation from Stocker et al.. Further, the data is also whitened as described in section 2.2.5. An effective technique with simple implementation that takes into account the theoretical shape of the eigenvalue curve is needed to automate this initial phase.

Second, assuming a linear mixture model, some optimization algorithm is used to solve for the endmember matrix and the abundance matrix, i.e. to unmix the image into the separate endmembers that make up the image. The abundance of a particular endmember in each of the observations, i.e. pixels, lies in a row of the abundance matrix. In other words, each row of the abundance matrix represents a feature in the image. These abundance rows are reshaped into the original image's length by width and plotted

on a grey scale to form abundance maps. The abundance maps show, visually, the location of the endmember it represents. This phase can be considered an extension of the feature extraction phase that finds projections of the data that further separate the classes.

Third, from these rows of the abundance matrix, those rows which represent target features must be selected. Selecting the rows that are 'target like' is referred to as the target feature selection phase. As previously mentioned, one technique to determine which abundance rows are the target features, is to identify for each row the pixel with the highest abundance score. The pixel from each row with the highest score represents what some refer to as a pure pixel relative to its endmember class. The spectral signature for each of these pixels in the original HSI data matrix is compared to spectral signatures of targets of interest. Those pixels whose signatures are closest to targets of interest tell which rows from the abundance matrix are rows that correspond to target features. In the case where target signatures are unknown, as is the case with this thesis, one must determine the target features based solely on the information in the data. In this domain, one must make some assumptions about target characteristics. Targets are rare in the image (represent a small class) and have a spectral signature significantly different from the signatures of the other classes in the image. Currently, the kurtosis values of abundance rows significantly higher than kurtosis values of other abundance rows are used to nominate the rows that are the target features. However, what is lacking is a clear repeatable mathematical definition of what constitutes a significantly high kurtosis relative to the other abundance rows. Analysis of a scree graph of kurtosis values to find the significant slope change via inspection is problematic. Further, except for Koo's

work, other statistical measures have not been investigated that could identify target features. These are a few improvements in the target feature selection phase that are needed in the field of HSI global anomaly detection that uses the LMM.

The fourth phase is the target identification phase. Once the abundance rows corresponding to targets are selected, one must identify the location of the targets in the abundance maps formed from the selected abundance matrix row vectors. As explained in section 2.5, one such method is the histogram method that selects the first bin with a zero value as the separation point between target and non-target. While a bin width of 0.05 appears to be a good choice as shown in Figures 2-28 and 2-30, choice of bin width needs to be tested on a larger image set to gain insight as to the most robust choice. Further, as shown in figure 2-30, the middle image, false positive declarations are still made even though the correct target features (maps 1, 2, and 3 sorted by KV) were selected. The threshold set was low enough to include some background pixels as target pixels in the selected target signals. Perhaps, smoothing the selected target signals prior to target identification will improve the histogram method's performance.

Below in Figure 3-1 is a simple flow diagram of the target detection process explained. The goal of this research effort is to fully automate the target detection process via development of robust, repeatable, and programmable set of decisions at each stage in the process outlined below.

Feature Extraction I	**Feature Extraction II**	**Feature Selection**	**Identification**
PCA dimensionality reduction followed by whitening	Solve for abundance matrix to unmix image via ICA (optimization)	Select target features based on some measure of target characteristics	Identify which pixels are targets in the selected features
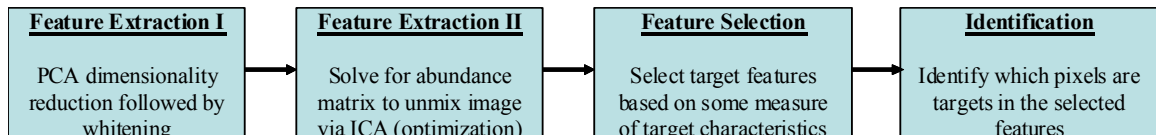
Figure 3-1. Process Flow for Target Detection

3.1.1 Test Images

The following HYDICE images in Figure 3-2 will be used in the following sections as representative samples of HSI images from non-urban, rural environments. They will be used in the development of improved methods at each phase of the detection process. As previously mentioned, atmospheric absorption bands have been removed leaving 145 bands.

ARES 1F	ARES 2F	ARES 1D	ARES 2D
10 Targets	30 Targets	6 Targets	46 Targets
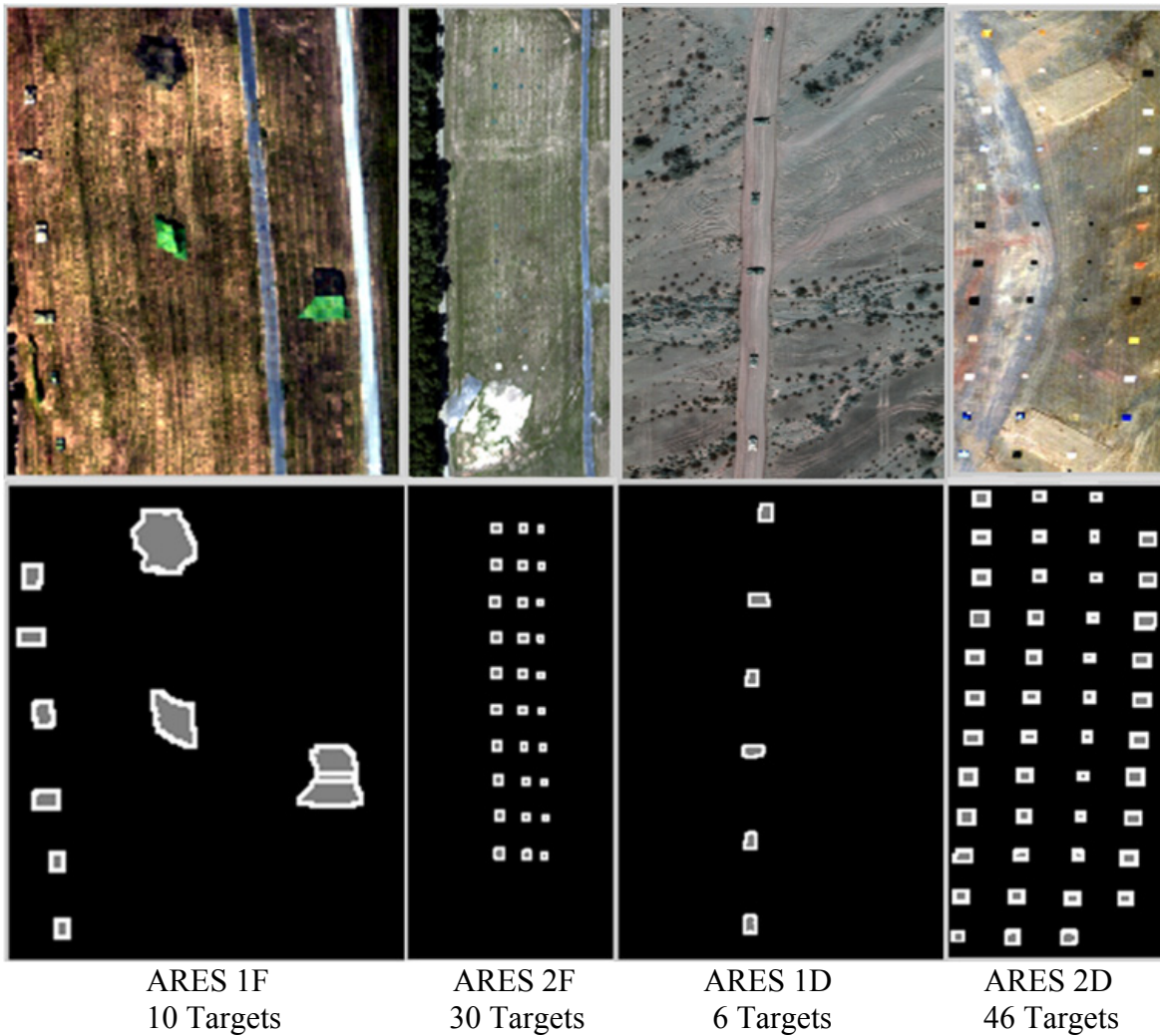
Figure 3-2. HYDICE HSI Test Images
('F' denotes Forest Radiance and 'D' denotes Desert Radiance)

3.1.2 Measures of Detector Performance

Performance of the detector will be measured based on three standards, false positive fraction, FPF, true positive fraction, TPF, and of the total number of pixels detected the percentage that are truly targets. In order to define FPF and TPF more clearly, one must have the definitions of the following four measures calculated at the pixel level.

True Positive (TP) - A target pixel is correctly declared a target pixel

False Positive (FP) - A non-target pixel is falsely declared a target pixel

True Negative (TN) - A non-target pixel is correctly declared a non-target pixel

False Negative (FN) - A target pixel is incorrectly declared a non-target pixel

Based on the definitions above, $\text{TPF} = \dfrac{\text{TP}}{\text{TP} + \text{FN}}$ and $\text{FPF} = \dfrac{\text{FP}}{\text{FP} + \text{TN}}$. TPF simply stated is of all the target pixels present, what percentage of pixels detected were declared as target pixels. Likewise, FPF simply stated is of all the non-target pixels present, what percentage were declared as target pixels. The third measure of performance, of the total number of pixels detected, what percentage are targets pixels, would be calculated as

percent $\text{TGT} = \dfrac{\text{TP}}{\text{TP} + \text{FP}}$. These scores will be calculated from the truth masks in Figure 3-2. The highlighted areas represent the location of true target pixels. Note the white line around the targets in the truth masks. These denote regions of indifference meaning no penalties occur for false declarations (FP or FN) or true declarations (TP or TN).

Worth noting before proceeding further is the prior probability of a pixel being a target pixel in each of these images. For ARES 1F, 1,980 out of 30,560 pixels are targets or a prior probability of 0.065. For ARES 2F, 1,528 pixels out of the total 47,424 pixels

are targets. Thus, the prior probability is 0.032. For ARES 1D, 672 out of 57,909 pixels are targets or a prior probability of 0.012. Finally, for ARES 2D, 2465 out of 22,360 pixels are targets, or a prior probability of 0.11.

3.1.3 FastICA Settings and Computer Specifications

The following settings in FastICA were used to conduct all experimentation for the remainder of this thesis:

- Symmetric Orthogonalization

- Function g of choice: $g = y^3$ (denoted by pow3 in FastICA)

    o The decision to use pow3 vs. tanh will be justified in section 3.3

- Initial starting point: Normal random matrix with elements that have zero mean and unit variance

    o This matrix refers to the W matrix described in (2.88) and (2.89). It is the default starting point in FastICA.

- Maximum number of iterations: 1,000

- Step size: 1

- Stabilized version of algorithm selected

    o This allows the step size to be momentarily halved if the program detects that the algorithm is stuck between two points

    o Also if no convergence has been achieved after an eighth of the number of max iterations (125 iterations) the step size is halved for the remainder of the iterations

    > *This author modified the FastICA code. Originally if no convergence had been achieved after half the number of max iterations (500 iterations) the step size would be halved for the remainder of the iterations. This was found, empirically, to be too long to wait to permanently change the step size. Line 390 of the fpica.m file was changed to an eighth of the max iterations.*

- Stopping criterion: 10^{-5}

It should be noted that all experiments in this thesis were completed on an Intel Xeon 5160 dual core computer with dual processors speeds of 3.00 GHz and 2.99 GHz and 3.00 GB of RAM.

3.2 Dimensionality Reduction (Feature Extraction) Automation

As mentioned previously two options that can be used to determine the number of principal components to retain after accomplishing PCA, is to retain enough components to explain a particular percentage of variability or where the eigenvalue noise floor occurs in the covariance matrix of the data. To illustrate the pitfalls of using a set percentage of variability to reduce the dimension, consider a truncated version of ARES 2F shown below in Figure 3-3.

Figure 3-3. Truncated Version of ARES 2F

Table 3-1 shows the number of components to keep for each choice of percentage of variability explained. Notice that for just a small change in percentage of variability explained 99.1% to 99.9% the number of components changes from 8 to 56. For a simple image such as this, with background of mainly grass and dirt and small panel targets, 56 components or endmembers is much more than its true dimensionality. For this image, 99.32% or 10 components was found to be sufficient at isolating the targets into separate abundance maps after performing an ICA. However, a choice of 99.32% variability was insufficient at isolating the targets into separate abundance maps with ARES 1F. 99.78% or 9 components were needed before all targets were isolated into their own abundance map.

**Table 3-1. ARES 2F: Truncated Percentage of Variability Explained
per Number of Components to Retain**

ARES 2F Truncated	
No. Components	% Variability Explained
8	99.12%
11	99.39%
19	99.60%
36	99.80%
56	99.90%

**Table 3-2. ARES 1F: Percentage of Variability Explained
per Number of Components to Retain**

ARES 1F	
No. Components	% Variability Explained
4	98.99%
5	99.42%
6	99.61%
10	99.80%
21	99.90%

If one were to use 99.78% on ARES 2F truncated, then 33 components would be retained, significantly more than required. Further, although retaining more components may attenuate the dependency that exists between the abundance row vectors and aids in ICA effectiveness, conversely, retaining significantly more components than needed reduces the signal to noise ratio and the effectiveness of ICA as explained in section 2.3.1. This results since one would be retaining more of the eigenvalues associated with noise. Further, the more components one retains, the slower FastICA will perform due to having to compute more ICs. Thus, even with just two examples finding a constant choice of percentage of variability explained to reduce the dimensionality for any image is problematic. Percentage of variability to retain to effectively reduce dimensionality is image dependent.

Recall the discussion on pages 2-12 through 2-14 regarding the distribution of the eigenvalues associated with the covariance of the random noise vector in the LMM. A least square fit to the Silverstein distribution (assuming white noise) approximately locates the location of the 'tilted ramp' on the eigenvalue curve to locate the breakpoint, (i.e. 'knee') between signal and noise eigenvalues. Finding the breakpoint in the nonwhite noise case can be more difficult given the lack of a closed form expression for the distribution of the nonwhite noise eigenvalues.

3.2.1 Max Euclidean Distance from Log-Scale Secant Line

A simpler approach developed by this author, not considered in the current literature, is a simple optimization scheme to locate the 'knee' in the eigenvalue curve.

In Figure 3-4 are the white and nonwhite noise cases again from Figure 2-2 (b) that was reproduced from page 653 of the Stocker citation. However, the figure has been modified to illustrate this new method.
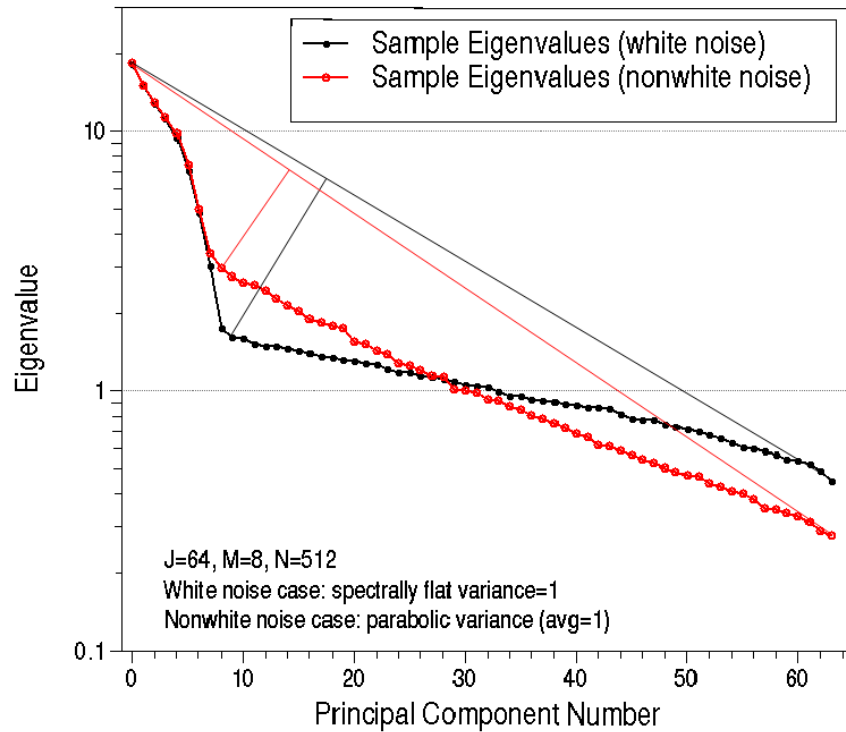
**Figure 3-4. Finding Breakpoint between Signal and Noise
Eigenvalues of White Noise and Nonwhite Noise Using
Max Euclidean Distance from Log-Scale Secant Line**

As shown from Figure 3-4, if one finds the Euclidean distance from each eigenvalue coordinate pair on the curve to the line that connects the endpoints of the curve (deemed secant line), the eigenvalue with the maximum normal distance to the secant line locates the breakpoint relatively well. It should be noted that the vertical component of all the coordinate pairs for each eigenvalue is considered to be the base 10 logarithm of the original eigenvalue in calculating the Euclidean distances from each coordinate pair to the secant line. For both the white and nonwhite cases this simple rule

appears to locate the 'knee' relatively well. This method is by no means a rigorous

solution to find the true breakpoint between signal and noise eigenvalues, but offers a

simple dimensionality approximation easy to implement in computer code to

autonomously determine dimensionality. Further, for the purposes of target detection,

locating the exact breakpoint between signal and noise eigenvalues may not be necessary.

Finding the general location of the knee may prove relatively effective if the sensitivity of

the decision in general vicinity of the knee is low with regard to keeping enough

dimensions to unmix the image into enough classes such that targets are isolated.

The true dimensionality of a hyperspectral image, in terms of the LMM, is the

number of endmembers in the image, or the total number of distinct classes of objects in

the image. In the case of a HSI target detection algorithm using the LMM, practically,

one would need to retain at least enough principal components so that after solving the

LMM via ICA, target classes are separated from non-targets classes in abundance maps.

Consider for example ARES 1F. An experiment was conducted varying the number of

principal components kept from just 1 component to 9 components, the point at which all

targets were isolated from non-targets in abundance maps after running FastICA on the

reduced PCA space.
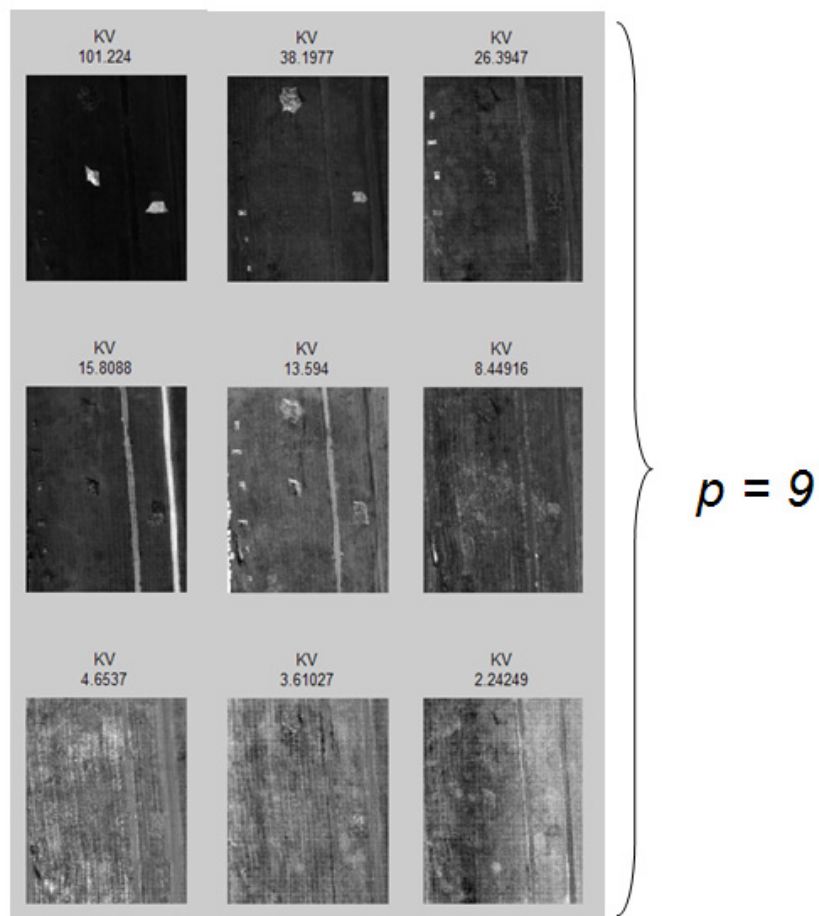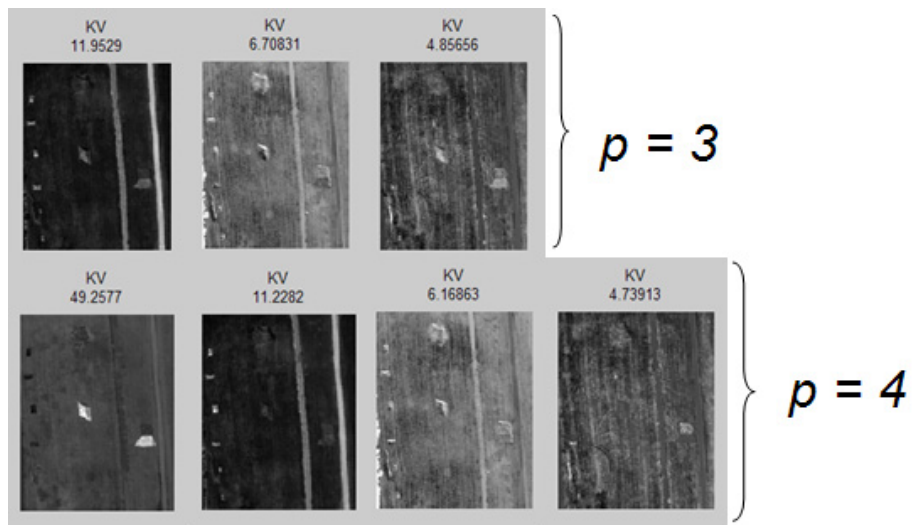
Figure 3-5. Minimum Dimensionality Needed for ARES 1F to Isolate Targets

Figure 3-5 on the previous page shows the abundance maps labeled with kurtosis value (KV) that result from keeping 3, 4, and 9 principal components and then executing FastICA. For a dimensionality of 3, the first map from the left shows vehicles and tarps (the targets) convoluted with road. For a dimensionality of 4, some of the tarps are isolated in their own abundance map as seen in the map with a KV of 49. However, the vehicles are still convoluted with the road. By a dimensionality of 9, one can see the 10 targets are isolated in the 3 abundance maps with the highest KVs. Thus, the minimum dimensionality needed for purposes of target detection for ARES 1F is 9. It should be emphasized that 9 is not necessarily the true dimensionality of the image. It is merely argued that at least 9 components are needed to separate out targets after running FastICA.

Below in Figure 3-6 is the eigenvalue curve for ARES 1F on a log scale and the point where the Euclidean distance from the log-scale secant line is maximum.
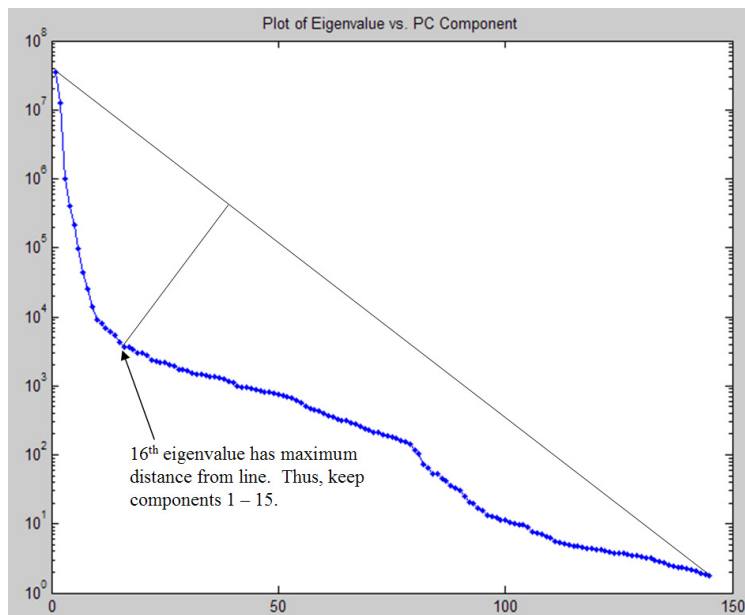
Figure 3-6. ARES 1F: Eigenvalue Curve with Max Euclidean Distance from Log Scale Secant Line Occurring at 16th Eigenvalue

This geometric solution would keep 15 components rather than the 9 suggested in Figure

3-5. Note the abundance maps underlined (dotted line) in Figure 3-7 below.
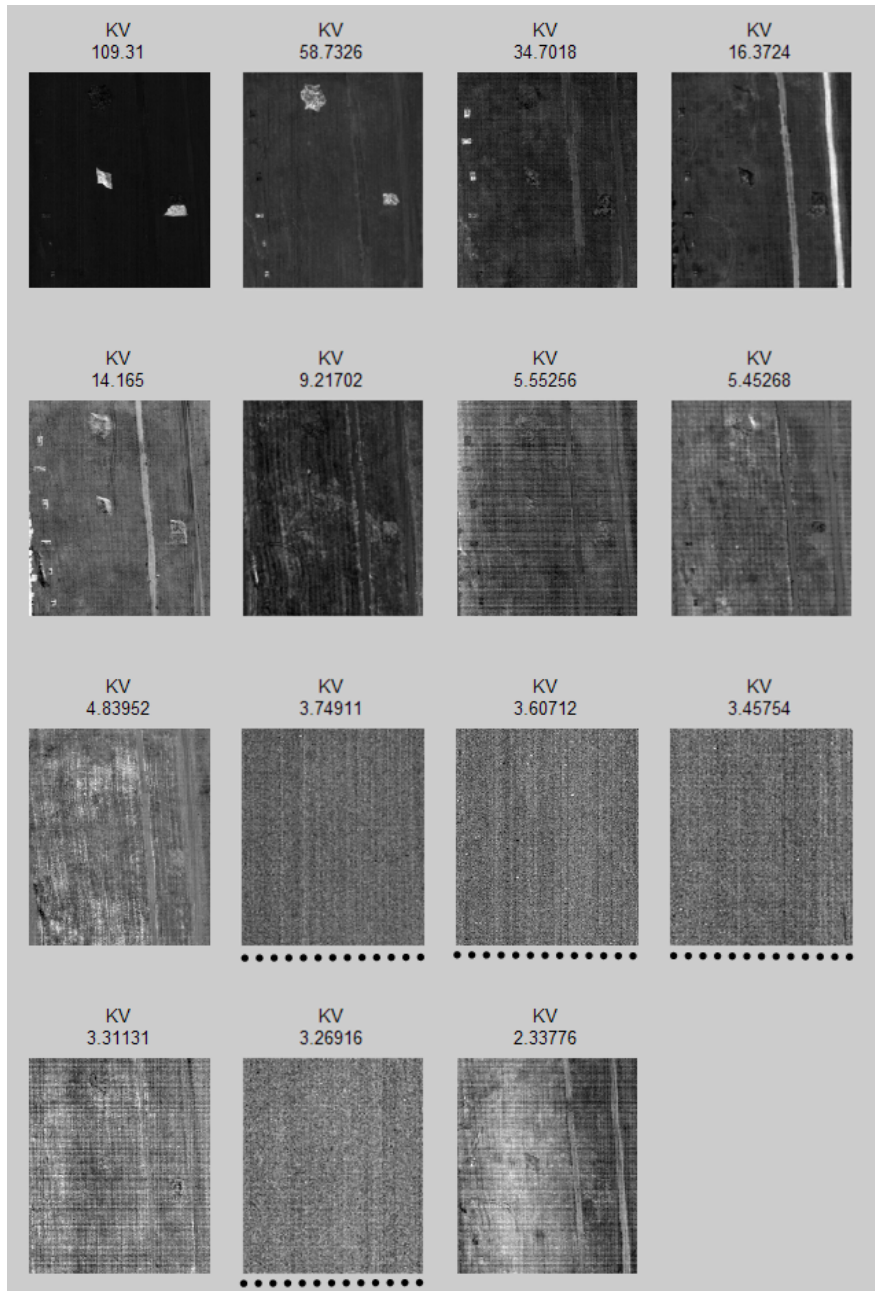
**Figure 3-7. ARES 1F: Abundance Maps from 15 ICs that
Result from New Dimensionality Decision - 3/15 Isolate All 10 Targets**

It seems that retaining 15 components results in 4 abundance maps that appear to be entirely noise. However, retaining these additional 6 components does not effect clear target isolation in the 3 abundance maps with the highest KVs in Figure 3-7.

Further worth noting is the increase in target signal strength (using kurtosis as a measure of signal strength) from a dimensionality of 9 to 15. With a dimensionality of 9, the kurtosis values associated with the target abundance maps are 101, 38, and 26. However, with a dimensionality of 15, the kurtosis values increase to 109, 59, and 35 respectively. Thus, despite adding the additional maps that are only noise, using this dimensionality decision, targets are still clearly separated but with stronger signals. For the purpose of target detection, this new method of deciding dimensionality appears effective for this image.

Including some eigenvalues associated with noise can, as seen in this example, improves target signal strength. This effect appears to agree with the research conducted by Nascimento and Dias where they state that increasing the number of endmembers improves ICA performance due to attenuating the dependency that truly exists between the components. Recall that in the LMM the abundance fractions, which are the independent components in ICA, are not truly independent due to the sum-to-one and non-negativity constraints imposed on the fractions. This dimensionality decision increased the number of endmembers from 9 (enough to separate targets) to 15 which includes 4 noise endmembers (the dotted underlined maps in Figure 3-7). However, their research also states that ICA decreases as SNR decreases. Thus, retaining too many noise endmembers could eventually degrade the performance of ICA due to reducing the SNR of the LMM as defined in equation 2.91. Further, retaining more and more dimensions

will increasingly slow the rate of convergence of ICA due to having to compute more and more ICs.

The next section will use the maximum distance from the eigenvalue curve to the log-scale secant line to reduce the dimensionality of the test images. For the remainder of this thesis, this new method will be referred to as the MDSL (Max Distance Secant Line) decision for dimensionality reduction. Effectiveness will be judged on whether or not the MDSL decision was sufficient at isolating all targets into separate abundance maps.

3.2.2 Dimensionality Assessment for Test Images

Figure 3-8 shows the log-scale eigenvalue curves of the covariance matrices for ARES 2F (a), ARES 1D (b), and ARES 2D (c) along with the respective MDSL decisions. Figures 3-9, 3-10, and 3-11 show the abundance maps formed from the ICs for each image that result from the MDSL dimensionality decisions from Figure 3-8. Images are sorted by kurtosis value, labeled above each map.
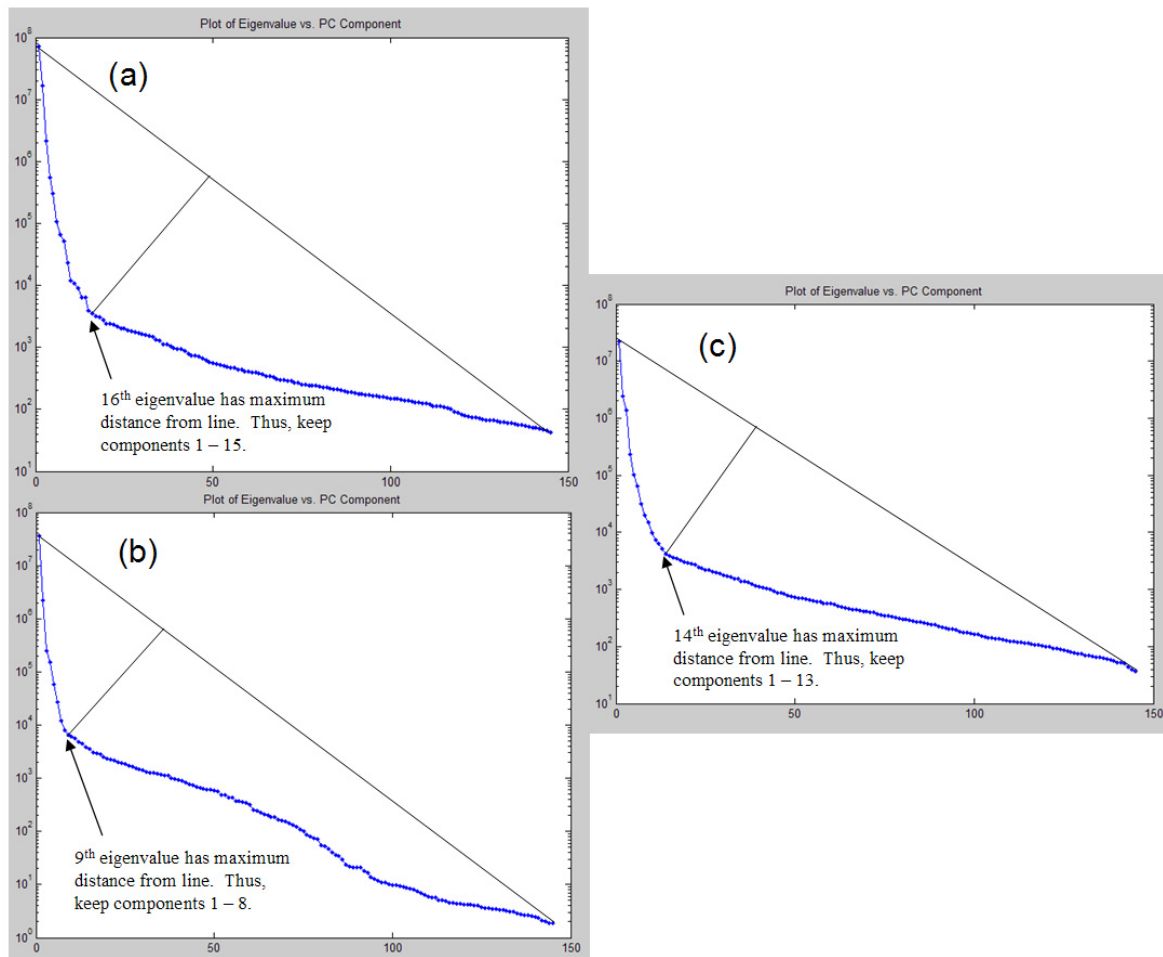
Figure 3-8. ARES 2F (a), ARES 1D (b), and ARES 2D (c) Dimensionality Decisions via MDSL
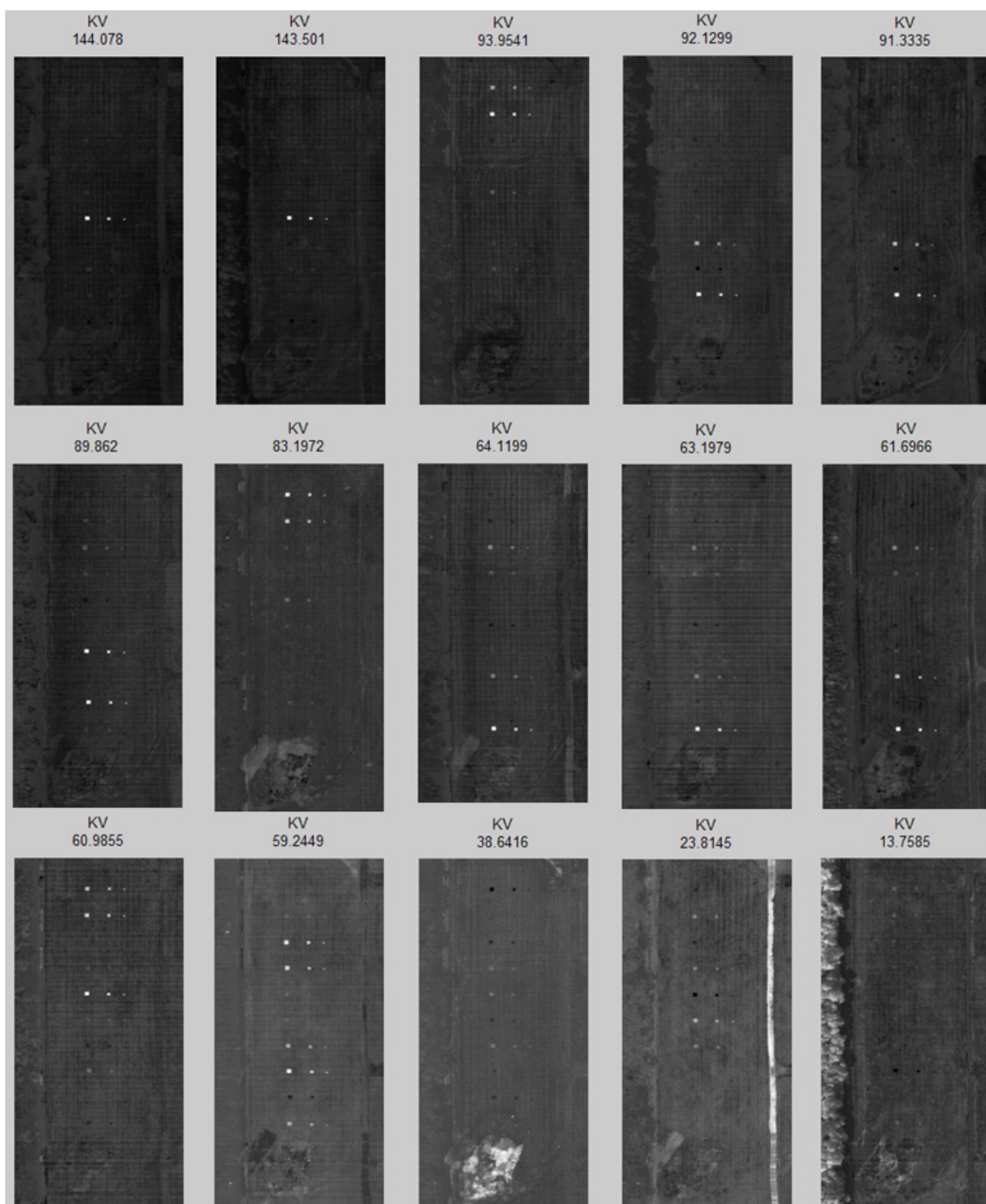
**Figure 3-9. ARES 2F: Abundance Maps from 15 ICs via MDSL Decision
Top 12 Maps by KV Isolate All 30 Targets**
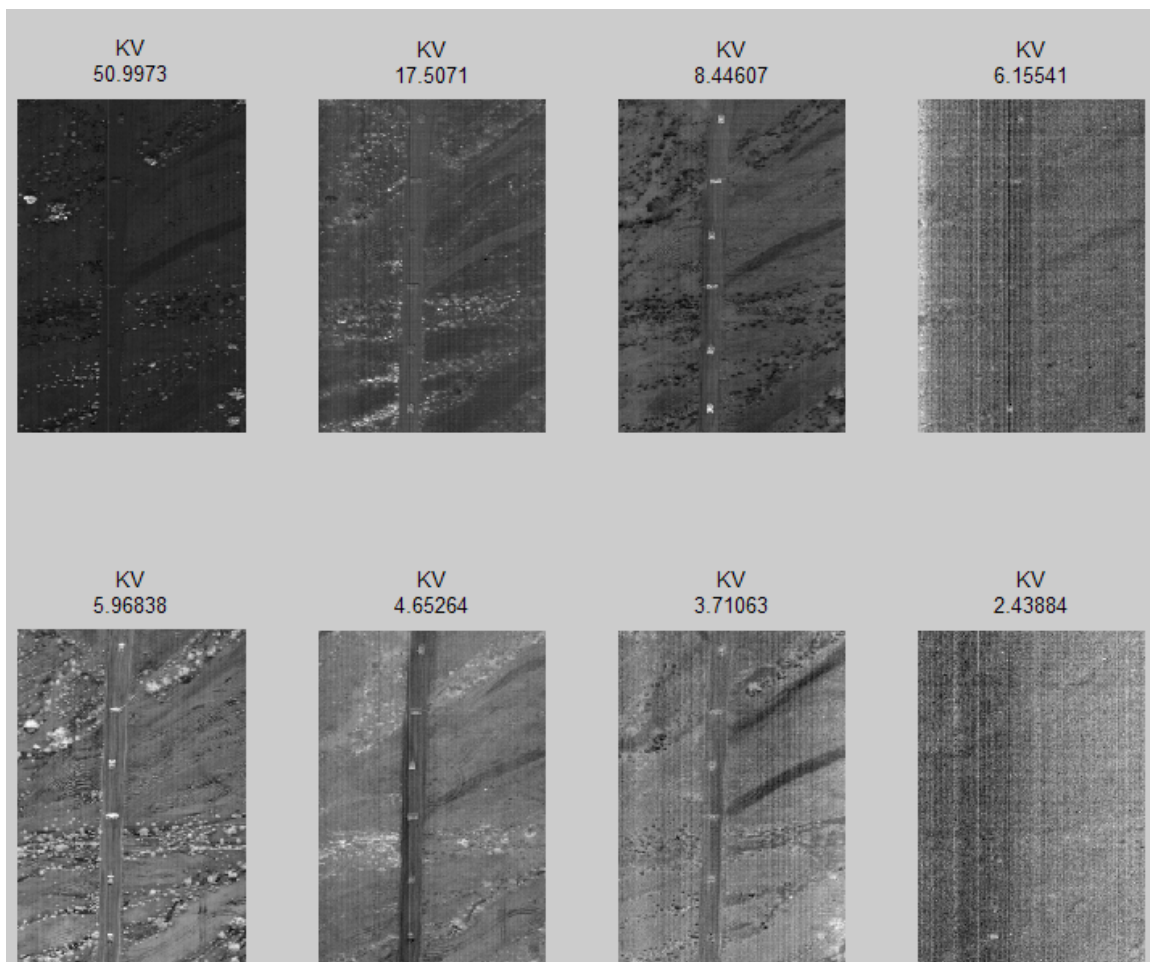
**Figure 3-10. ARES 1D: Abundance Maps from 8 ICs via MDSL Decision
3rd Map by KV Isolates All 6 Targets**
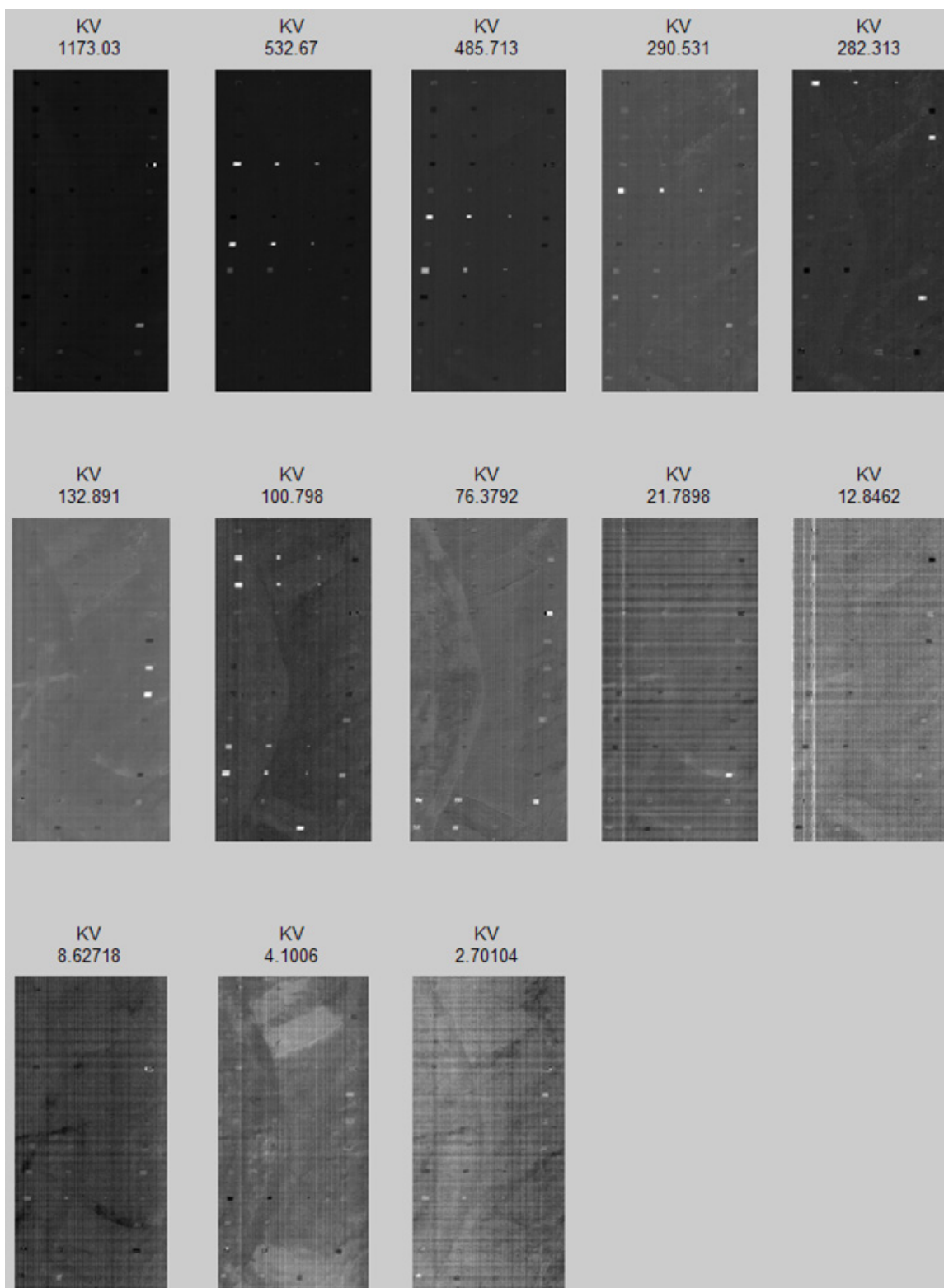
**Figure 3-11. ARES 2D: Abundance Maps from 13 ICs via MDSL Decision
Top 9 Maps and Map 11 by KV Isolate All 46 Targets**

Given that the MDSL decision resulted in sufficient dimensionality to isolate all

targets using ICA, this technique appears to be an effective autonomous dimensionality

reduction method. However, there can be cases where the shape of the eigenvalue curve

does not conform to the theoretical shape an eigenvalue curve of a covariance matrix of a

spectral data matrix under the LMM. Recall, that of the 210 spectral bands, 145 were

retained after removing the atmospheric absorption bands. Consider the case if the

absorption bands were left. Notice the shape of the eigenvalue curve of the covariance

matrix for ARES 2D with all 210 bands shown in Figure 3-12.
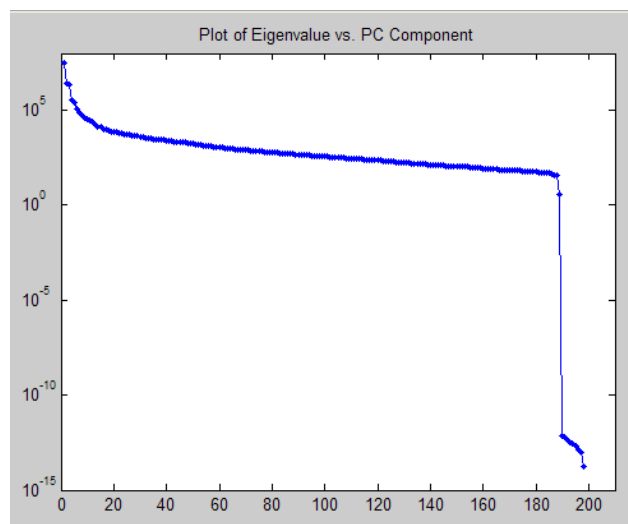
**Figure 3-12. ARES 2D: Eigenvalues of Covariance
Matrix of Spectral Data with all 210 Bands**

One can still observe a tilted ramp except for the eigenvalues near the end of the curve.

Due to including the absorption bands, these eigenvalues are very close to zero compared

the rest of the eigenvalues on the tilted ramp. Some are even negative in value (which

cannot be shown due to the log scale) still very close to zero due to computer precision

error. One would not want to connect the log-scale secant line to the endpoints of this curve. Doing so would bias the MDSL calculation away from the true 'knee'. As a protection against a pathological case such as this, MDSL method will check the right endpoint and successively move the endpoint up from which to construct the secant line until the eigenvalue is greater than 10^{-4} (merely a nominal decision point).

Worth noting is that in Figures 3-9 and 3-11 some of the same targets appear in more than one map. In some cases a target with a weak abundance in one map may have a stronger abundance in another where it is duplicated. Thus, this can be a positive result. Redundancy does not negatively affect the detection process so long as maps with duplicated targets which are kept during the target feature selection phase do not have significant noise than can result in false positives during the target identification phase. As seen with Figure 3-11, maps 9 and 11 sorted by KV can be considered redundant. Circled targets appear in other maps. The histogram method with a bin size of 0.05 (previously described in section 2.5) was applied keeping maps 1-9 and 11, and then applied keeping only maps 1-8 to check for ill effects. Results are shown in Figure 3-13. Across the three measures of performance, no significant detection difference occurs from keeping the redundant maps.

TPF = 0.958333	TPF = 0.958261
FPF = 0.00188212	FPF = 0.0015148
% TGT = 0.93086	% TGT = 0.943493
Histogram Bin Width = 0.05	Histogram Bin Width = 0.05

Keeping 10 Maps   *Keeping 8 Maps*
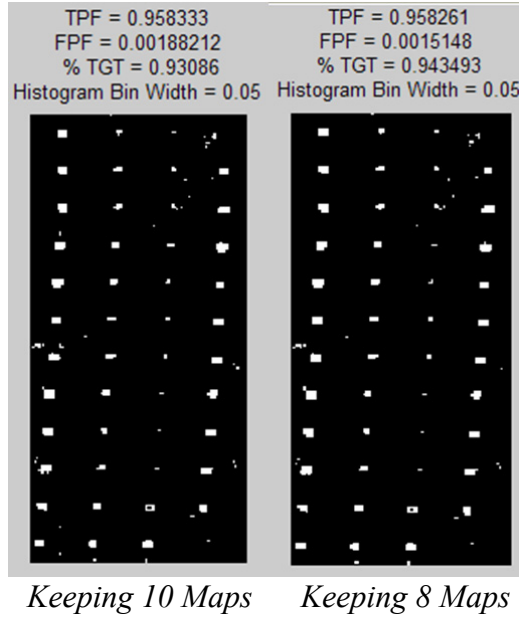
**Figure 3-13. ARES 2D: Target Identification Comparing
10 Target Maps Selected to 8 Target Maps Selected**

3.3 Variability Analysis to Choose FastICA Objective Function

Before continuing any further analysis, it would be prudent to check the

variability from one run to the next for each of the test images given the random nature of

FastICA. Recall FastICA finds projections of the data, $y = \underline{w}^T \underline{x}$ defined by $\underline{w}$, that

maximize the projected data's negentropy. Kurtosis can be used to approximate

negentropy where $G(y) = y^4$ for the G function in (2.65). Thus, the g function for this

choice in the fixed point iteration scheme given in (2.84) would be $g = y^3$, where g is the

derivative of kurtosis. Also, nonpolynomial moments can be used such as

$G(y) = \dfrac{1}{a_1} \log(\cosh a_1 y)$ for the G function in (2.65) to approximate negentropy.

Thus, $g = \tanh(a_1 y)$ in the fixed point iteration scheme given in (2.84), where g is

derivative for this choice of G. These choices correspond to the 'pow3' and 'tanh' settings in FastICA respectively. Further, recall that FastICA uses a random initial W matrix (the matrix described in (2.88) and (2.89)) as the staring point in the algorithm. Thus, results can vary for each run on the same image. The FastICA settings described in section 3.1.3 will be held constant except for the choice of g function. The results for the FastICA g functions, 'pow3' and 'tanh', will be contrasted. Further, the MDSL dimensionality decisions described in the previous section will be used for each image. Tables 3-3 through 3-6 show the mean and variance of the kurtosis values for the abundance maps sorted by kurtosis value for each test image using the 'pow3' and 'tanh' FastICA settings.

Table 3-3. ARES 1F: 100 Runs Using FastICA Settings of pow3 vs. tanh

ARES 1F 100 Runs (pow3)			ARES 1F 100 Runs (tanh)		
Abundance Map	**Mean KV**	**Var KV**	**Abundance Map**	**Mean KV**	**Var KV**
Map 1	109.31	7.34E-27	Map 1	108.74	3.81E-07
Map 2	58.73	1.42E-08	Map 2	58.76	1.82E-05
Map 3	34.70	5.16E-07	Map 3	25.71	6.76E-04
Map 4	16.37	5.27E-06	Map 4	16.88	2.96E-05
Map 5	14.16	1.04E-05	Map 5	15.28	1.81E-04
Map 6	9.22	1.69E-07	Map 6	7.51	5.03E-03
Map 7	5.55	1.71E-07	Map 7	5.68	4.16E-05
Map 8	5.45	2.32E-07	Map 8	5.62	9.30E-04
Map 9	4.84	5.20E-06	Map 9	4.85	1.83E-04
Map 10	3.75	1.12E-07	Map 10	3.58	5.44E-05
Map 11	3.61	6.36E-09	Map 11	3.55	5.69E-05
Map 12	3.46	3.86E-07	Map 12	3.46	1.83E-05
Map 13	3.31	8.43E-06	Map 13	3.41	2.17E-04
Map 14	3.27	9.08E-07	Map 14	3.25	1.27E-03
Map 15	2.34	1.85E-08	Map 15	2.17	1.29E-05

tanh variance per map	
pow3 variance per map	
Map 1	5.19E+19
Map 2	1.28E+03
Map 3	1.31E+03
Map 4	5.61E+00
Map 5	1.74E+01
Map 6	2.98E+04
Map 7	2.43E+02
Map 8	4.01E+03
Map 9	3.53E+01
Map 10	4.86E+02
Map 11	8.95E+03
Map 12	4.75E+01
Map 13	2.58E+01
Map 14	1.40E+03
Map 15	6.97E+02

For ARES 1F the variance for each map sorted by kurtosis is quite low, between 10^{-27} and 10^{-6} for the pow3 solved maps and 10^{-7} and 10^{-3} for tanh solved maps. Despite both settings having low variances, when examining the ratio of the variances for each map between tanh and pow3, tanh variances are between 1 and 4 orders of magnitude more (except for the one case of 19 orders of magnitude) than the pow3 run variances. Thus, although both give relatively stable results in terms of low variability, pow3 is the more stable of the two for this image.

Table 3-4. ARES 2F: 100 Runs Using FastICA Settings of pow3 vs. tanh

ARES 2F 100 Runs (pow3)		
Abundance Map	**Mean KV**	**Var KV**
Map 1	166.64	361.88
Map 2	157.99	513.43
Map 3	111.27	589.22
Map 4	106.53	598.41
Map 5	91.64	126.31
Map 6	87.17	166.00
Map 7	79.84	119.79
Map 8	63.99	33.54
Map 9	62.50	20.74
Map 10	60.28	23.32
Map 11	57.43	53.82
Map 12	52.40	112.93
Map 13	34.67	65.17
Map 14	18.04	58.94
Map 15	8.77	16.20

ARES 2F 100 Runs (tanh)		
Abundance Map	**Mean KV**	**Var KV**
Map 1	403.04	66057.20
Map 2	318.35	37592.56
Map 3	260.96	21271.20
Map 4	150.27	4914.49
Map 5	115.19	1227.69
Map 6	78.16	1129.89
Map 7	70.70	1087.70
Map 8	59.50	986.36
Map 9	28.87	150.32
Map 10	21.39	69.68
Map 11	17.27	69.35
Map 12	12.84	36.08
Map 13	10.23	22.79
Map 14	8.23	14.24
Map 15	4.83	5.07

$\dfrac{\text{tanh variance per map}}{\text{pow3 variance per map}}$	
Map 1	1.83E+02
Map 2	7.32E+01
Map 3	3.61E+01
Map 4	8.21E+00
Map 5	9.72E+00
Map 6	6.81E+00
Map 7	9.08E+00
Map 8	2.94E+01
Map 9	7.25E+00
Map 10	2.99E+00
Map 11	1.29E+00
Map 12	3.20E-01
Map 13	3.50E-01
Map 14	2.42E-01
Map 15	3.13E-01

In contrast to ARES 1F, ARES 2F has more run to run variability for the 15 maps sorted.

Kurtosis values range from a variance of 16 to 598 for pow3 and 5 to 66,057 for tanh.

Except for the last four maps, tanh's variance is between 1.3 and 183 times more per map

than pow3's variance for the map of the same KV rank. Thus, again for this image, pow3

yields more consistent results in terms of lower variability.

Table 3-5. ARES 1D: 100 Runs Using FastICA Settings of pow3 vs. tanh

ARES 1D 100 Runs (pow3)		
Abundance Map	**Mean KV**	**Var KV**
Map 1	51.00	4.39E-06
Map 2	17.50	1.77E-05
Map 3	8.46	1.66E-04
Map 4	6.15	1.45E-06
Map 5	5.97	8.63E-07
Map 6	4.64	1.13E-04
Map 7	3.72	3.86E-05
Map 8	2.44	3.42E-07

ARES 1D 100 Runs (tanh)		
Abundance Map	**Mean KV**	**Var KV**
Map 1	37.58	3.38E-02
Map 2	15.69	3.73E-01
Map 3	8.11	2.42E-01
Map 4	6.46	5.72E-03
Map 5	6.26	2.32E-03
Map 6	6.11	9.74E-02
Map 7	3.81	8.66E-02
Map 8	2.31	4.78E-04

$\dfrac{\text{tanh variance per map}}{\text{pow3 variance per map}}$	
Map 1	7.68E+03
Map 2	2.11E+04
Map 3	1.46E+03
Map 4	3.93E+03
Map 5	2.69E+03
Map 6	8.59E+02
Map 7	2.24E+03
Map 8	1.40E+03

Similar to ARES 1F, variability is low for the kurtosis values of each sorted map but with pow3 variances consistently lower.

Table 3-6. ARES 2D: 100 Runs Using FastICA Settings of pow3 vs. tanh

ARES 2D 100 Runs (pow3)			ARES 2D 100 Runs (tanh)		
Abundance Map	**Mean KV**	**Var KV**	**Abundance Map**	**Mean KV**	**Var KV**
Map 1	1172.90	1.27E-02	Map 1	776.36	45.93
Map 2	532.67	1.56E-07	Map 2	529.28	25.34
Map 3	485.71	3.63E-05	Map 3	476.07	402.83
Map 4	290.40	1.35E-02	Map 4	280.91	21.94
Map 5	282.31	8.48E-06	Map 5	242.03	50.10
Map 6	132.80	6.66E-03	Map 6	151.93	1.13
Map 7	100.61	2.51E-02	Map 7	79.45	1.15E-02
Map 8	76.37	1.33E-04	Map 8	38.11	1.58
Map 9	22.44	3.06E-01	Map 9	17.12	3.39E-02
Map 10	12.78	3.24E-03	Map 10	16.25	1.80E-02
Map 11	8.62	9.46E-05	Map 11	15.69	2.03E-02
Map 12	4.10	6.73E-07	Map 12	5.58	6.78E-02
Map 13	2.70	4.02E-06	Map 13	2.38	5.95E-04

$\dfrac{\text{tanh variance per map}}{\text{pow3 variance per map}}$	
Map 1	3.62E+03
Map 2	1.62E+08
Map 3	1.11E+07
Map 4	1.62E+03
Map 5	5.91E+06
Map 6	1.69E+02
Map 7	4.59E-01
Map 8	1.19E+04
Map 9	1.11E-01
Map 10	5.57E+00
Map 11	2.15E+02
Map 12	1.01E+05
Map 13	1.48E+02

Finally, for ARES 2D, yet again the same trend is evident that pow3 yields unmixed images with lower variability.

Thus, in conclusion, from these four images, choosing the setting in FastICA that maximizes negentropy using the kurtosis approximation ('pow3' setting) rather than the nonpolynomial moment approximation ('tanh' setting) results in lower run to run variability. With respect to creating a detector, an objective function that yields more consistent results in unmixing the image into separate classes, would be the robust choice.

3.4 New Target Feature Filters (Feature Selection)

As previously mentioned in the arena of global anomaly detectors that employ ICA to solve the LMM, the kurtosis value of the abundance matrix row vectors is used to identify the which rows represent target features. Those maps with significantly higher kurtosis values than the rest are deemed as target maps. For this method there two issues:

1. What is a quantifiable definition of significantly higher kurtosis value?

2. Is there an absolute kurtosis value threshold that can be used to determine if a image even has any targets? In an image that has no targets, there will still invariably be frames with kurtosis values higher than the rest. Without a target kurtosis value threshold, for any image, the highest kurtosis maps will be deemed target maps regardless of whether or not an image even has target like anomalies.

A computer algorithm tasked to analyze these images has no a priori knowledge of target information and must make a decision as to the cutoff between target and non-target classes via some quantifiable definition of significant change between non-target classes of kurtosis values and target classes of kurtosis values. For the four test images, a scree plot of each map's kurtosis values (as computed for each image in sections 3.2.1 and 3.2.2) will be presented in Figures 3-13 through 3-16. For each image, two possible definitions of significant change in kurtosis value to decide between target and non-target classes will be compared to truth information. Results illustrate that attempting to quantify this breakpoint is problematic.

3.3.1 Kurtosis Value Filter Problematic
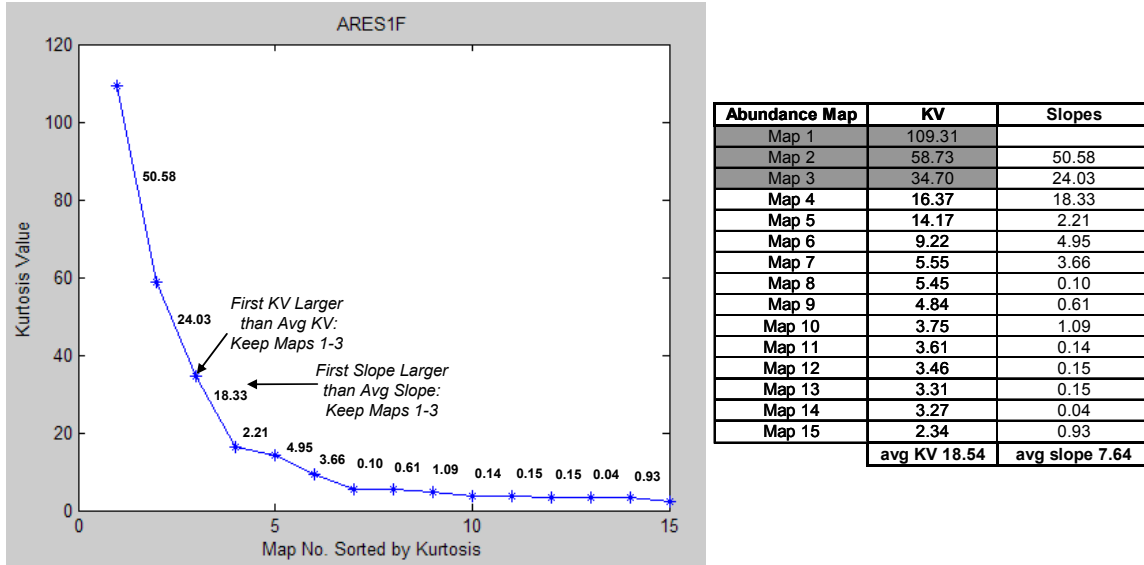
Figure 3-14. ARES 1F: KV Scree Plot. Target Maps Highlighted in Gray

Abundance Map	KV	Slopes
Map 1	109.31	
Map 2	58.73	50.58
Map 3	34.70	24.03
Map 4	16.37	18.33
Map 5	14.17	2.21
Map 6	9.22	4.95
Map 7	5.55	3.66
Map 8	5.45	0.10
Map 9	4.84	0.61
Map 10	3.75	1.09
Map 11	3.61	0.14
Map 12	3.46	0.15
Map 13	3.31	0.15
Map 14	3.27	0.04
Map 15	2.34	0.93
	avg KV 18.54	avg slope 7.64

As shown in Figure 3-14, two definitions are used to quantify significant kurtosis change. One option may be to choose the target candidate maps with KVs higher than the average KV for all maps. Another option, starting from the right of the plot, could identify the first slope larger than the average of all the slopes between the sorted KVs. All maps above this significant slope would be chosen as target candidate maps. As shown in Figure 3-13, this clear definition of significant change between non-target classes and target classes correctly picks the maps that belong to the target class. An analyst visually analyzing this graph could arguably make the same judgment keeping maps 1 through 3 due to the noticeable jump from map 4 to map 3.

Addressing issue 2, in terms of a simple absolute KV threshold to determine if an image even has targets, from this image, one might choose a KV of 20. All non-target maps have a KV less than 20. Thus, when analyzing another image, if no maps had a KV

above 20, the conclusion could be made that no targets are in the image. However, as

will be seen in subsequent examples, this conclusion would be naïve.

Abundance Map	KV	Slopes
Map 1	144.08	
Map 2	143.50	0.58
Map 3	93.95	49.55
Map 4	92.13	1.82
Map 5	91.33	0.80
Map 6	89.86	1.47
Map 7	83.20	6.66
Map 8	64.12	19.08
Map 9	63.20	0.92
Map 10	61.70	1.50
Map 11	60.99	0.71
Map 12	59.24	1.74
Map 13	38.64	20.60
Map 14	23.81	14.83
Map 15	13.76	10.06
	avg KV 74.90	avg slope 9.31
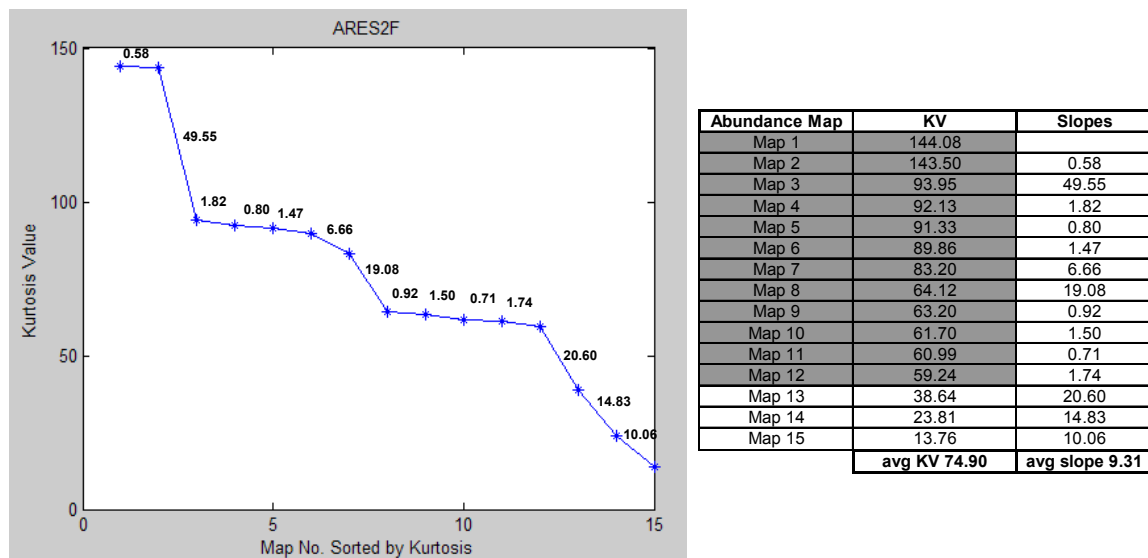
Figure 3-15. ARES 2F: KV Scree Plot. Target Maps Highlighted in Gray.

Unlike ARES 1F, identifying the point of separation between classes is not as

clear. Both definitions of separation between classes fail to identify correctly maps 1

through 12 as target maps. Further, notice the shape of this plot. Even an analyst, having

no a priori target information could arguably have a difficult time determining the

breakpoint visually. Several noticeable jumps occur, from maps 13 to 12, then 8 to 7, and

finally from 3 to 2.

As far as issue 2, a couple of non-target maps have a KV greater than 20. Thus,

the KV threshold conclusion from the previous image would be incorrect. Further, note

that for ARES 2F, one non-target map (map 13) has a KV of 38.64. ARES 1F has a

target map with a KV of 34.70. Thus, with just two examples, it is apparent that non-

target classes can have KVs just as high as or higher than target class KVs. This phenomena will appear in ARES 1D.

Abundance Map	KV	Slopes
Map 1	51.00	
Map 2	17.51	33.49
Map 3	8.45	9.06
Map 4	6.16	2.29
Map 5	5.97	0.19
Map 6	4.65	1.32
Map 7	3.71	0.94
Map 8	2.44	1.27
	avg KV 12.48	avg slope 6.94
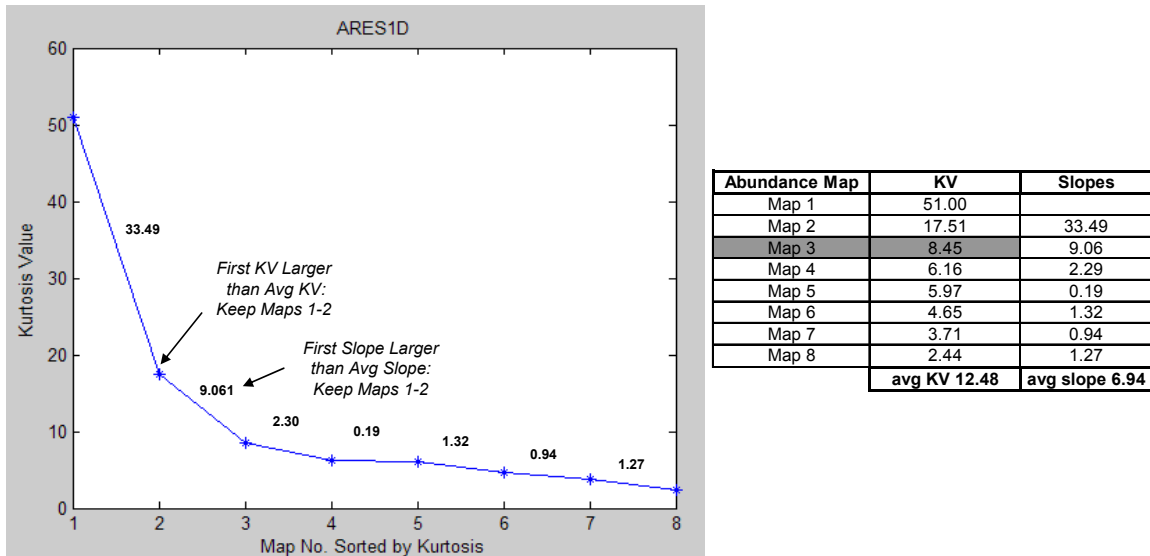
Figure 3-16. ARES 1D: KV Scree Plot. Target Map Highlighted in Gray.

For ARES 1D, the target map (map 3) does not even have the highest KV. Further, both definitions of significant change would not even include map 3 in the target class. Also, notice the much lower KV of this target map relative to the KVs of the target maps from the previous two images.

Addressing issue 2 again, this image would force the lower bound KV threshold to determine if an image even has targets to 8.
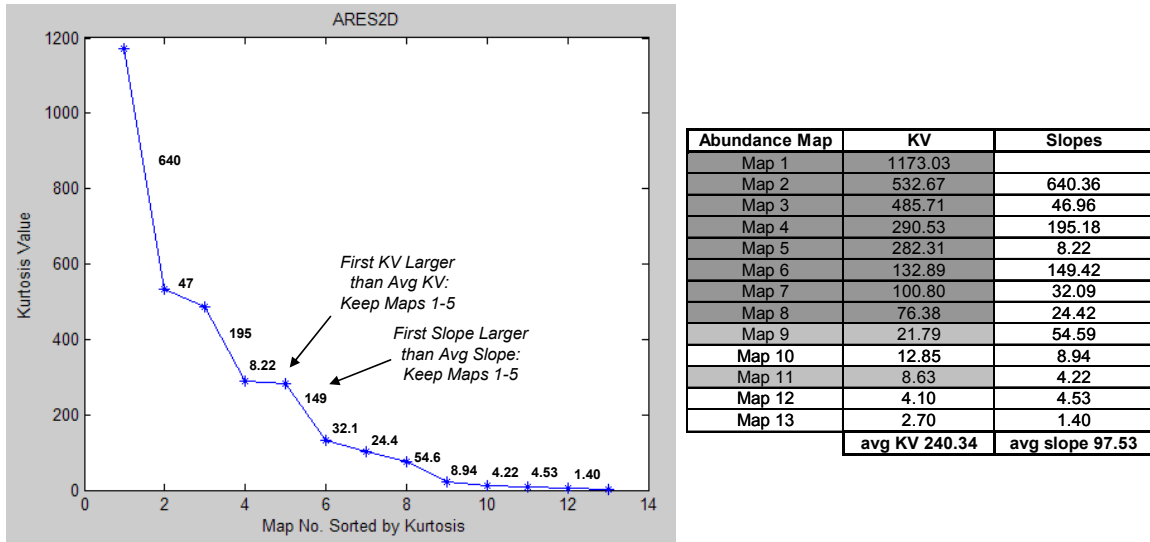
Figure 3-17. ARES 2D: KV Scree Plot. Target Maps Highlighted in Gray. Lighter Highlighted Maps (9 and 11) Denotes Indifference.

Again, these definitions of significant slope change fail to correctly identify maps 1 through 8. An analyst could arguably choose maps 1 through 8, given the slope from map 9 to 8 is 54.6, substantially higher than the previous four slopes of 8.94, 4.22, 4.53, and 1.4. Perhaps another definition of significant slope change would correctly choose 54.6 as the significant change. However, this research will not attempt to further refine the definition of significant KV change because regardless of finding a better definition of significant KV change, from these four examples the second issue explained on page 3-28 remains unsolved.

By defining a target as a rare occurrence in the image (a small class) and having a spectral signature significantly different from the signatures of the other classes, an anomaly detector will invariably include some objects that meet this criteria that are not intended to be included as target classes. For example, in an image with a few boulders and tanks in a scene of primarily sand, dust, and gravel, an anomaly detector might

conclude the boulders and tanks are targets. This is an inescapable reality. However, an anomaly detector viewed as a preprocessing algorithm prior to the application of a signature matching detector, should eliminate as many pixels as possible that could be considered potential targets without eliminating the true targets. Thus, the work load of a signature matching algorithm is reduced due to having fewer pixels to compare to a library of target signatures of interest. In order to eliminate as many pixels as possible, without eliminating the true targets, based on sample images of the terrain, one would want to set a threshold no higher than the lowest value (KV for this example) of a target class in that terrain across the sample images. Figure 3-18 shows the smallest uncertainty region that does not eliminate any true target maps. Setting the KV threshold near the lowest mean target KV of 8 as a rule of whether or not an image even has targets would include 9 non-target maps across these four test images.
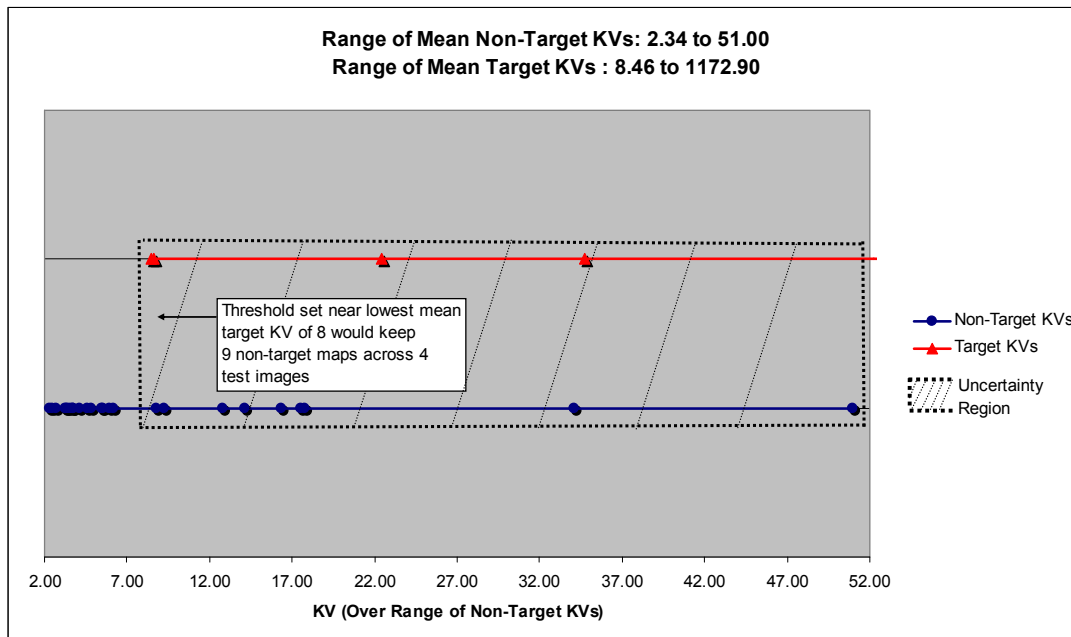
**Figure 3-18. Smallest Possible Uncertainty Region that
Includes all Target Maps in KV Feature Space**

3-34

As an illustrative example, Figure 3-19 below shows an image with no targets and the results from unmixing the image using ICA and using KV to determine if targets are present. Based on the scree plot of KV values, the first map would be classified as target. Further, based on a KV threshold setting of 8 to determine if the image even has targets, the first sorted map with a KV of 15.88 (road) would still be considered a target. Thus, filters are needed that better separate targets from non-targets than the KV filter.
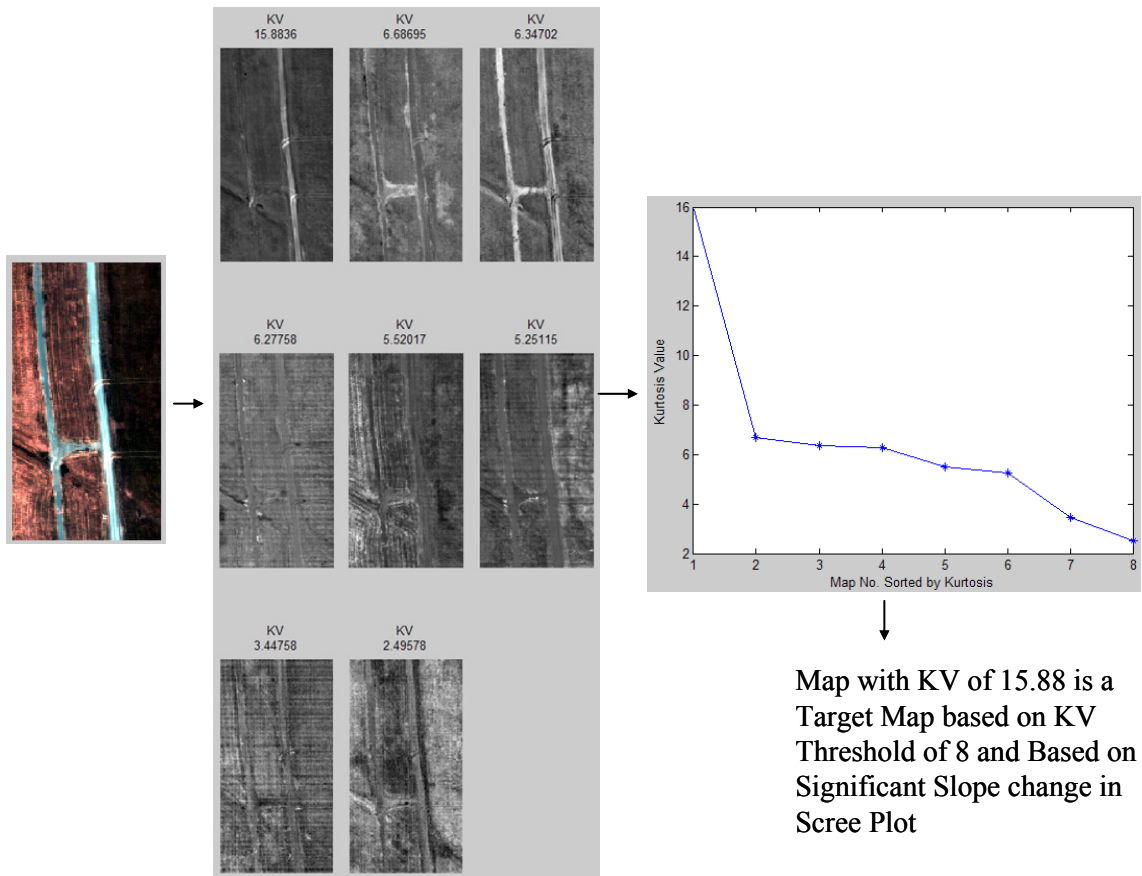
Figure 3-19. Image with No Targets

3.3.2 Max Pixel Score and Potential Target Signal to Noise Ratio Filter

Rather than using kurtosis value to rank solved abundance maps, two other potential filters developed by this author to select target maps that are not in the current literature are the map's max pixel score and potential target signal to noise ratio.

Max Pixel Score

Recall that the original data matrix is centered (zero mean) and ICs are standardized to have unit variance. Thus, for a particular independent component a pixel's absolute value score of 1, 2, 3, etc... on that component represents 1, 2, 3, etc... standard deviations above the mean score for that component. Perhaps ICs that have at least one pixel score above a certain consistent threshold could be used to nominate which ICs belong to the target class. One would expect target maps to have higher pixel scores than maps that isolate larger background classes. The top object in Figure 3-20 shows the plots of the pixel scores for each independent component. These plots will be referred to as IC signals from this point forward. Also, on a side note, so that the reader has a common reference when referring to map number in presented abundance maps and corresponding IC signals, maps will be presented in order of kurtosis value as in the previous sections. Figure 3-20's maps are still in order of kurtosis value. Notice the target signals in the first three plots have pixel scores higher than the plots of non-target classes. Thus, it appears ranking by max pixel score has potential.

Potential Target Signal to Noise Ratio

Another filter to nominate target maps will be called potential target signal to noise ratio (PT SNR). In the top three plots (the target plots), notice the difference between target pixel variability and background pixel variability.
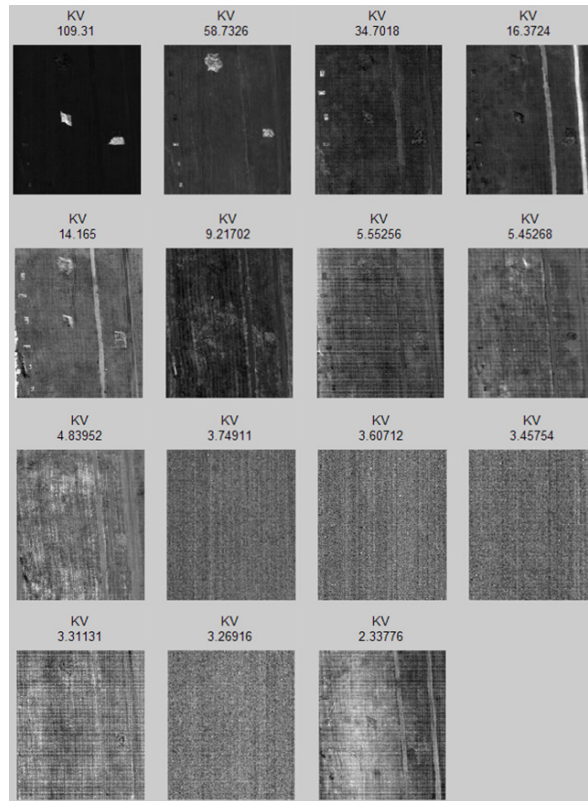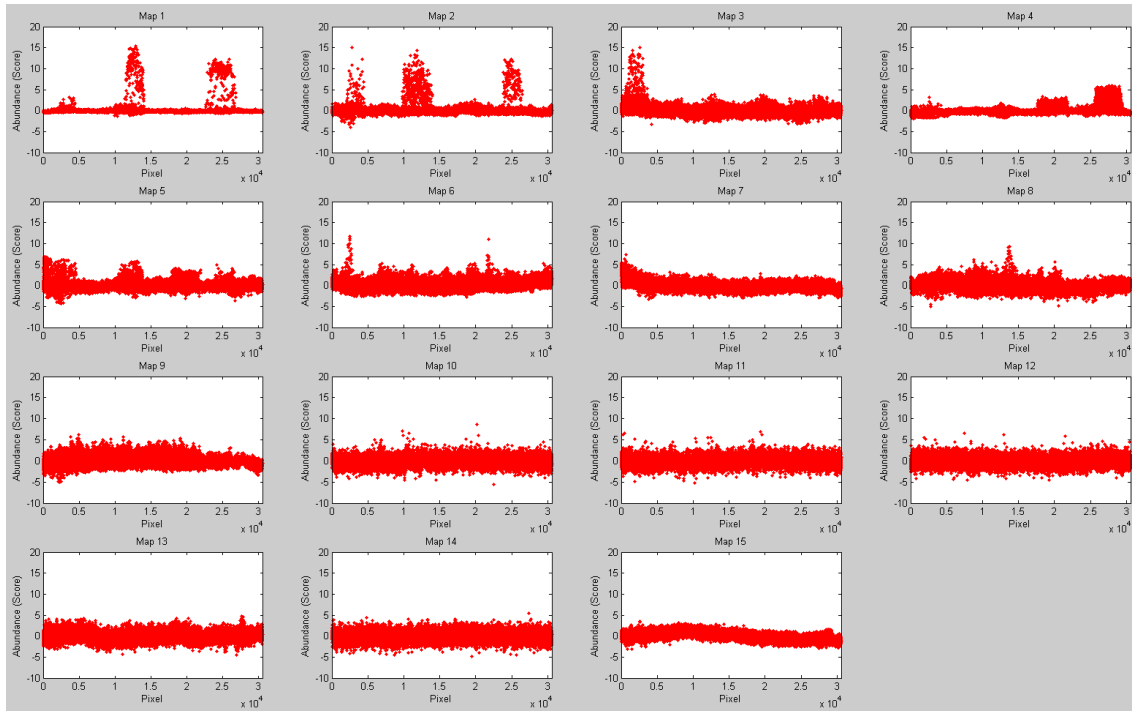
Figure 3-20. ARES 1F: Independent Component Signal Plots in Order of KV with Abundance Maps Directly Below

Pixel variability can be thought of as a measure of signal power. The signal to noise ratio is a measure of a signal's power to the power of the noise (background) and then is converted to decibels. The reason this author includes the wording 'potential target' with SNR is that here one is interested in the power of only those pixels that could be defined as potential target pixels. Thus, for a particular signal, the potential target signal to noise ratio would be computed as follows:

$$PT\ SNR_{dB} = 10 \cdot \log_{10}\left(\frac{\text{power(potential target signal)}}{\text{power(background)}}\right)$$
$$= 10 \cdot \log_{10}\left(\frac{\text{var(potential target signal)}}{\text{var(background)}}\right) \qquad (3.1)$$

A crucial problem to solve in order to calculate the PT SNR for each IC signal is to locate the separation point (partition line) between potential target signal and background. Thus, one needs a clear mathematical definition potential target pixels. An obvious choice is the zero-detection histogram method mentioned in section 2.5. According to Chiang, Chang, and Ginsberg, the outliers in the signal caused by small targets create ripples in the tails. This method defines potential target pixels as those pixels that come after the first empty (zero point) bin from the histogram constructed from the signal. An initial decision point with this method considered by this author is to use a bin width of 0.05. Figures 3-20 and 3-21 closely examine map 1's (target) signal and map 4's (non-target, road) signal using the zero-detection histogram method.

Given targets represent a small class, they comprise only small percentage of pixels in proportion to the total number of pixels in an image. Defining targets as a small class results in some telling characteristics of the frequency distribution constructed from the target class IC signal as shown in Figure 3-21.
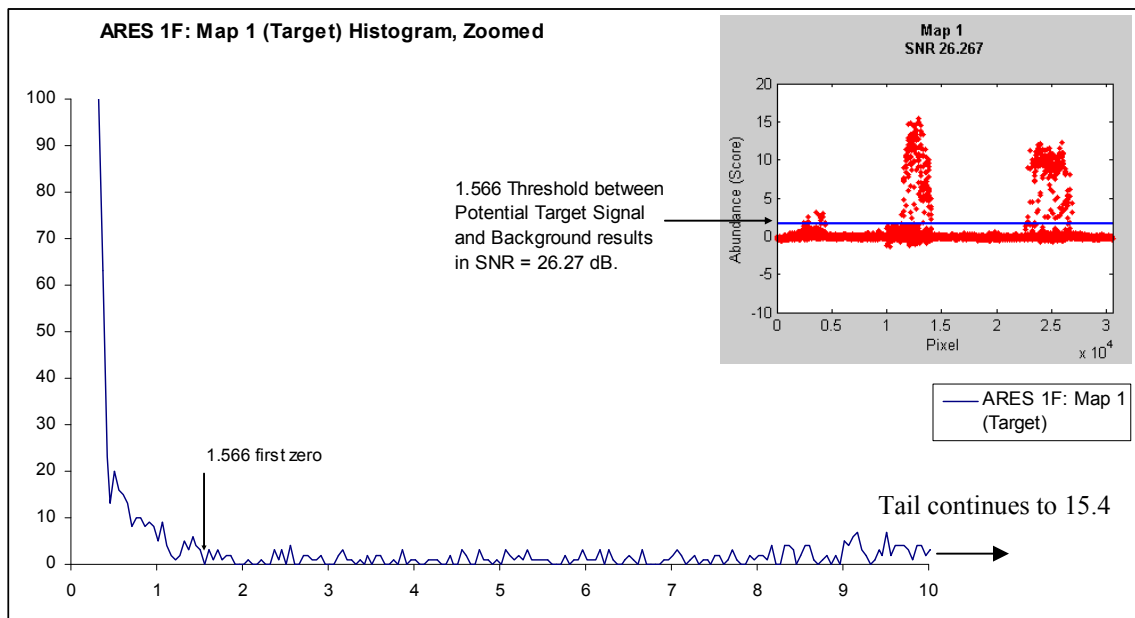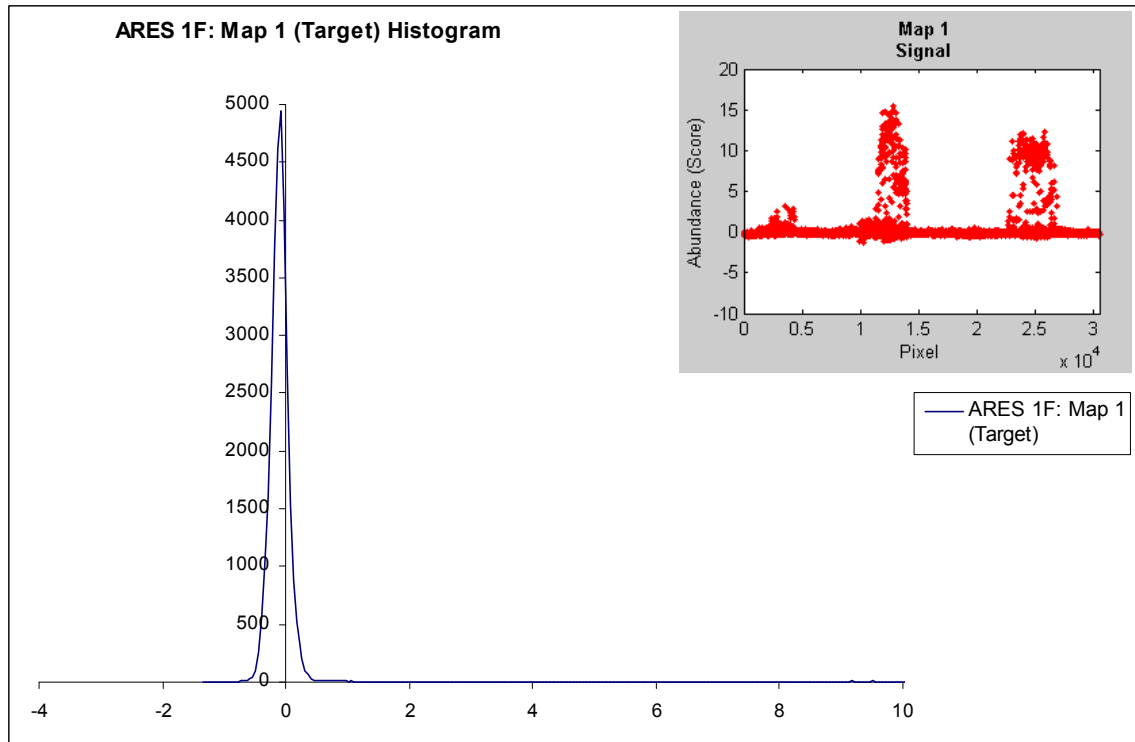
Figure 3-21. ARES 1F: Map 1 Potential Target Threshold Determination

The frequency distribution drops sharply to a thin long tail in the direction of the target

signal. The distribution drops sharply due to the high concentration of the background

pixels about the mean of zero. The tail is long due to the high scores of the outlier target pixels. The tail is thin due to the outlier target pixels representing, as just described, only a small proportion of the total pixels. Thus, the resulting density of the tail will be quite low. These characteristics contribute to the success of the zero-detection histogram method in detecting the breakpoint between potential target and background for the purposes of calculating PT SNR.

For this particular example, after applying the zero-detection histogram method, the variance of the pixels above the threshold are 10.50 and the variance of the pixels below the threshold are 0.0248. Thus, calculating the PT SNR one has

$$10 \cdot \log_{10}\left(\frac{10.5}{0.0248}\right) = 26.27 \text{ dB}.$$

In contrast to the characteristics of a frequency distribution constructed from a signal that isolates target, notice the characteristics of the frequency distribution constructed from a signal that isolates a larger class, such as the road in map 4 as shown in Figure 3-22. The pixels are less concentrated about the mean (lower peak in the distribution) and a fatter tail in the direction of the road signal. The peak is lower and the tail is fatter, due to the outlier road pixels representing a large class, a more substantial portion of the overall number of pixels. Thus, as evidenced in the zoomed in portion of Figure 3-22, the tail is 'fat' with the first zero-detection not occurring until, nearly, the very end of the tail. Therefore, the zero-detection histogram method determines this signal as having almost no pixels that would be considered potential targets. For an independent component signal representing a large class, this methods results in a quite low PT SNR, -37.83 dB as detailed in Figure 3-22. In conclusion, this new filter shows

promise at drawing a clearer separation between target maps and non-target maps than

the KV filter.
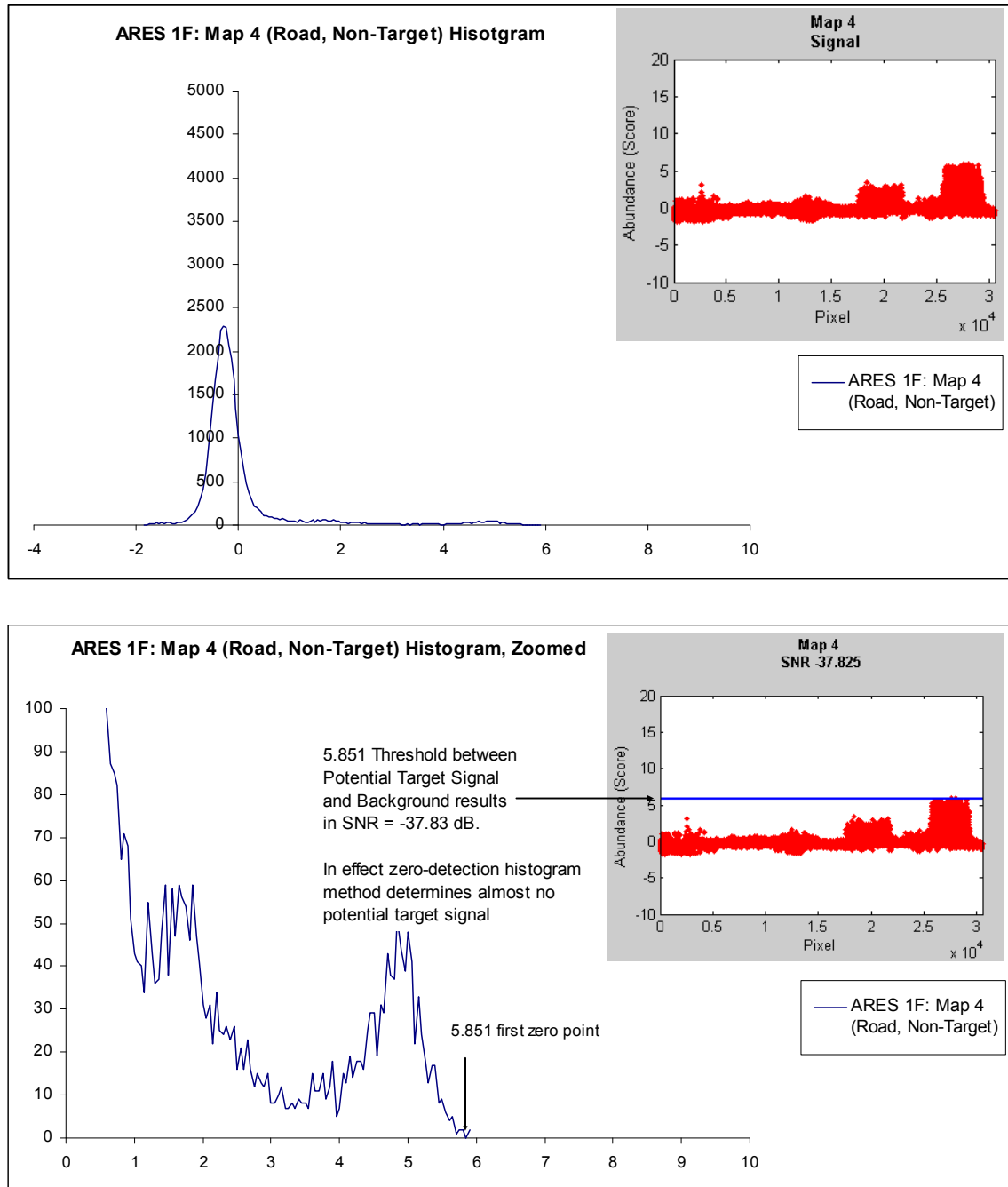
Figure 3-22. ARES 1F: Map 4 Potential Target Threshold Determination

Figure 3-23 shows the breakpoint decisions between background and potential target for the remaining IC signals and the corresponding PT SNR and max pixel score. Note that the target maps have the highest PT SNR and max pixel scores. The non-target map with the highest PT SNR and max pixel score of 7.12 dB and 11.6 respectively is map 6.
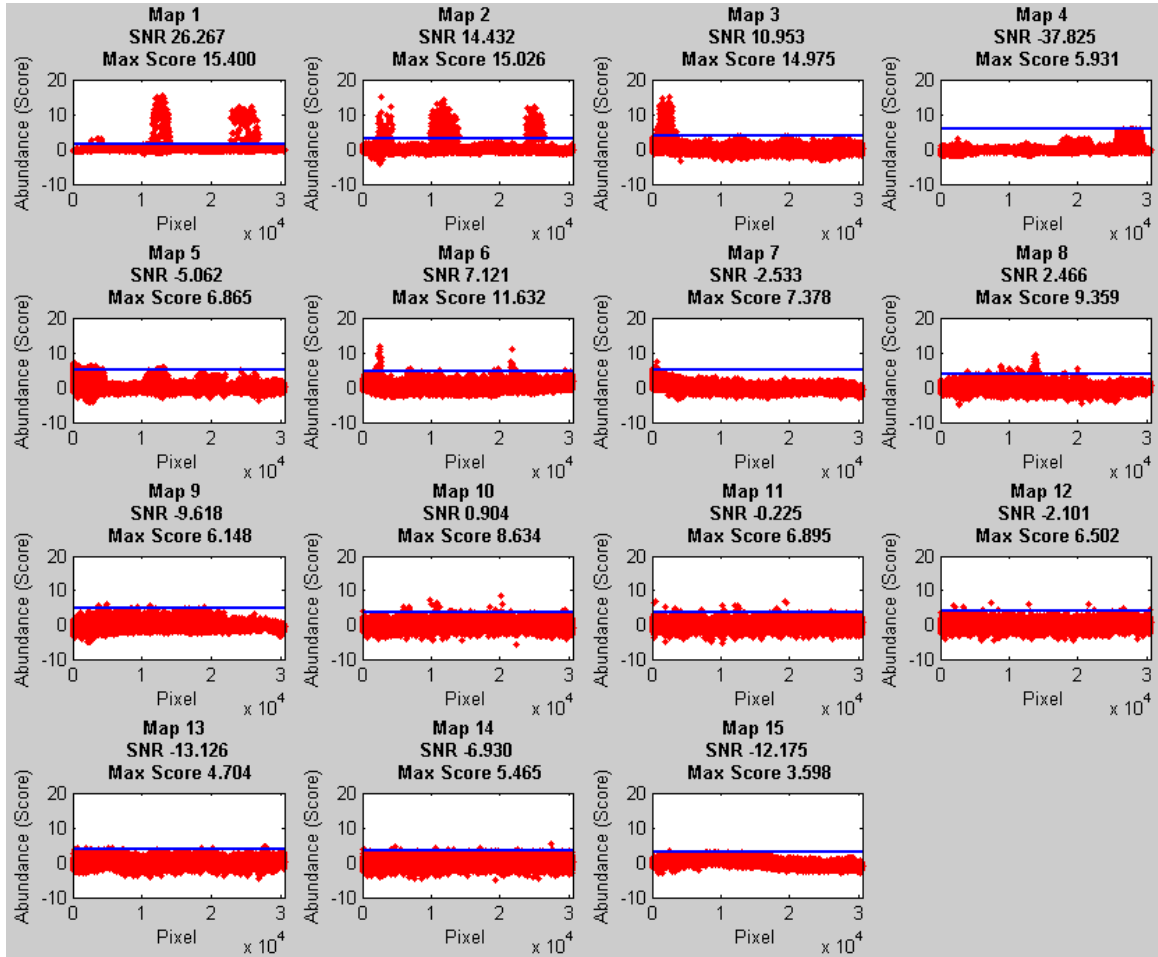
**Figure 3-23. ARES1F: PT SNR and Max Pixel Score for Each Signal
with Potential Target Threshold Lines**

Notice map 6 highlights a small patch of vegetation pointed out in Figure 3-24. This small patch meets the assumptions of targets (small, rare class). Although, not as strong

as the other true target signals, such a phenomena could invariably produce false

positives depending on the target threshold decision for the PT SNR and max pixel score.
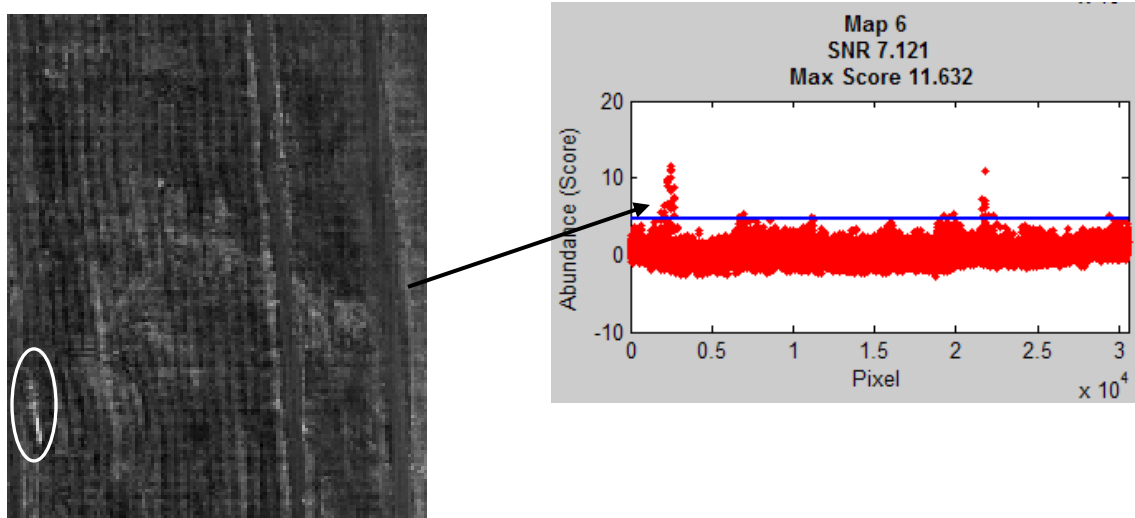
Figure 3-24. ARES 1F: Highest SNR and Max Pixel Score of Non-target Map

For the remaining test images, these two new filters will be calculated for each IC

signal to test if a clearer separation exists between targets and non-targets using these

measures. The experiment will be replicated 100 times to check for variability in each of

the new filters for each IC signal in each test image. Summarized results will appear in

subsequent tables for each of the test images. The maps will be sorted in descending

order according to PT SNR. Map numbers will stay consistent with the numbers given

according to the KV ranking.

Table 3-7. ARES 1F: New Target Feature Filters Summary 100 Reps
Target Maps Highlighted in Gray

Abundance Map	Mean KV	Var KV	Mean Max Score	Var Max Score	Mean PT SNR	Var PT SNR
Map 1	109.31	7.34E-27	**15.40**	1.00E-10	**26.27**	5.10E-29
Map 2	58.73	1.27E-09	**15.03**	6.32E-08	**14.43**	1.27E-29
Map 3	34.70	8.75E-09	**14.98**	2.73E-09	**10.95**	5.10E-29
Map 6	9.22	7.35E-08	**11.63**	2.66E-08	**7.12**	2.23E-09
Map 8	5.45	2.38E-07	**9.36**	2.78E-05	**2.63**	2.60E-02
Map 10	3.75	1.12E-07	**8.65**	2.26E-04	**0.97**	4.92E-03
Map 11	3.61	9.94E-10	**6.89**	3.34E-05	**-0.17**	1.13E-02
Map 12	3.46	4.01E-07	**6.50**	2.40E-05	**-1.99**	1.34E-02
Map 7	5.55	1.64E-07	**7.38**	6.15E-06	**-2.54**	7.55E-05
Map 5	14.17	3.66E-09	**6.86**	9.66E-09	**-5.06**	8.30E-08
Map 14	3.27	8.91E-07	**5.47**	7.96E-05	**-6.96**	5.36E-04
Map 9	4.84	5.31E-06	**6.15**	3.21E-05	**-9.52**	1.37E-01
Map 15	2.34	1.72E-08	**3.60**	5.02E-08	**-12.18**	7.16E-05
Map 13	3.31	8.62E-06	**4.73**	6.24E-04	**-13.28**	1.66E-02
Map 4	16.37	2.12E-08	**5.93**	2.93E-09	**-37.78**	2.15E-03

Table 3-8. ARES 2F: New Target Feature Filters Summary 100 Reps
Target Maps Highlighted in Gray

Abundance Map	Mean KV	Var KV	Mean Max Score	Var Max Score	Mean PT SNR	Var PT SNR
Map 1	165.82	385.20	**28.52**	0.17	**21.06**	0.27
Map 2	157.35	543.28	**27.19**	0.60	**19.72**	2.08
Map 4	107.71	581.35	**25.40**	3.74	**18.83**	1.98
Map 3	111.52	537.88	**26.00**	2.64	**18.72**	2.57
Map 5	92.80	73.50	**23.57**	1.87	**17.71**	0.38
Map 6	87.76	129.30	**22.90**	1.44	**17.27**	0.93
Map 7	79.67	113.61	**22.45**	2.98	**15.94**	1.35
Map 8	63.76	21.82	**22.33**	0.57	**15.70**	0.07
Map 9	62.64	13.33	**22.63**	1.38	**15.63**	0.46
Map 10	60.57	14.30	**21.26**	0.39	**15.34**	0.29
Map 11	58.28	45.11	**19.59**	8.31	**13.07**	48.16
Map 12	52.26	119.11	**17.00**	13.71	**8.68**	97.15
Map 14	17.76	58.41	**9.01**	8.80	**-9.61**	106.79
Map 13	34.13	62.94	**10.02**	5.21	**-12.55**	92.07
Map 15	8.76	14.26	**5.76**	5.52	**-18.83**	340.65

Table 3-9. ARES 1D: New Target Feature Filters Summary 100 Reps
Target Map Highlighted in Gray

Abundance Map	Mean KV	Var KV	Mean Max Score	Var Max Score	Mean PT SNR	Var PT SNR
Map 1	51.00	3.19E-06	**18.52**	4.06E-06	**8.04**	2.32E-01
Map 3	8.46	1.59E-04	**11.74**	1.37E-03	**4.56**	1.77E-01
Map 2	17.50	1.44E-05	**12.04**	3.85E-06	**1.52**	1.18E-03
Map 6	4.64	1.17E-04	**6.77**	1.87E-04	**-5.34**	2.09E-03
Map 8	2.44	2.49E-07	**4.62**	1.40E-06	**-5.59**	3.11E-02
Map 4	6.15	1.08E-06	**6.29**	7.29E-07	**-10.25**	5.44E-03
Map 7	3.72	2.96E-05	**5.48**	8.14E-05	**-10.81**	1.14E+00
Map 5	5.97	7.04E-07	**5.66**	9.50E-07	**-18.25**	1.76E-04

Table 3-10. ARES 2D: New Target Feature Filters Summary 100 Reps
Target Maps Highlighted in Gray.
Lighter Highlighted Maps (9 and 11) Denote Indifference.

Abundance Map	Mean KV	Var KV	Mean Max Score	Var Max Score	Mean PT SNR	Var PT SNR
Map 2	532.67	1.48E-07	**30.05**	2.57E-07	**34.70**	1.48E-07
Map 3	485.71	3.52E-05	**30.38**	6.59E-07	**28.23**	5.17E-08
Map 1	1172.90	1.25E-02	**58.24**	1.56E-07	**25.73**	1.16E-05
Map 5	282.31	6.97E-06	**24.97**	8.06E-07	**23.71**	2.49E-08
Map 4	290.39	1.32E-02	**28.29**	3.40E-06	**20.74**	7.41E-03
Map 6	132.79	6.51E-03	**24.18**	3.08E-07	**18.32**	4.62E-02
Map 7	100.61	2.46E-02	**16.98**	1.24E-04	**17.14**	1.70E-04
Map 8	76.37	1.23E-04	**18.82**	2.76E-05	**15.54**	2.02E-03
Map 9	22.46	2.99E-01	**14.91**	1.22E-02	**12.95**	1.79E-02
Map 11	8.62	7.87E-05	**15.65**	7.79E-05	**10.91**	6.88E-02
Map 12	4.10	6.31E-07	**8.71**	7.64E-06	**8.41**	3.66E+00
Map 13	2.70	3.83E-06	**7.02**	1.21E-05	**1.94**	1.57E-03
Map 10	12.78	3.15E-03	**7.61**	1.61E-04	**-5.60**	2.47E-02

First, notice in all four images that the majority of the 25 non-target signals have a negative mean PT SNR. Only 7 out of the 25 non-target signals have positive mean PT SNR. Particularly, large classes such as road, ARES 1F map 4, and forest, ARES 2F map 15, have the most negative mean PT SNR. For the 26 target signals, 24 have a mean PT SNR greater than 10 dB. Of those target maps below a mean of 10 dB, ARES 1D map 3 and ARES 2F map 12 have mean PT SNRs of 4.56 dB and 8.68 dB respectively.

With regard to the other filter, max pixel score, 21 out of 25 non-target signals have a mean max pixel score less than 10. Of the 4 non-target signals that have a mean

max pixel score greater than 10, only 3 have a mean PT SNR that are positive with mean

PT SNR's of 8.04 dB (ARES 1D map 1, large rock or bush feature), 7.12 dB (ARES 1F

map 6, small patch of vegetation feature) and 1.52 dB (ARES 1D map 2 small numerous

rocks or bushes feature). All 26 target signals have a mean max pixel score above 10.

Figure 3-25 plots all 49 IC signal values for the four test images in the new

feature space defined by the max pixel score filter and the PT SNR filter. Notice, in this

new space it is possible to define an uncertainty region with only 2 non-target maps, a

substantial improvement over the KV feature space where the smallest uncertainty region

included 9 non-target maps. The non-target maps in the uncertainty region are ARES 1D

map 1 and ARES 1F map 6.
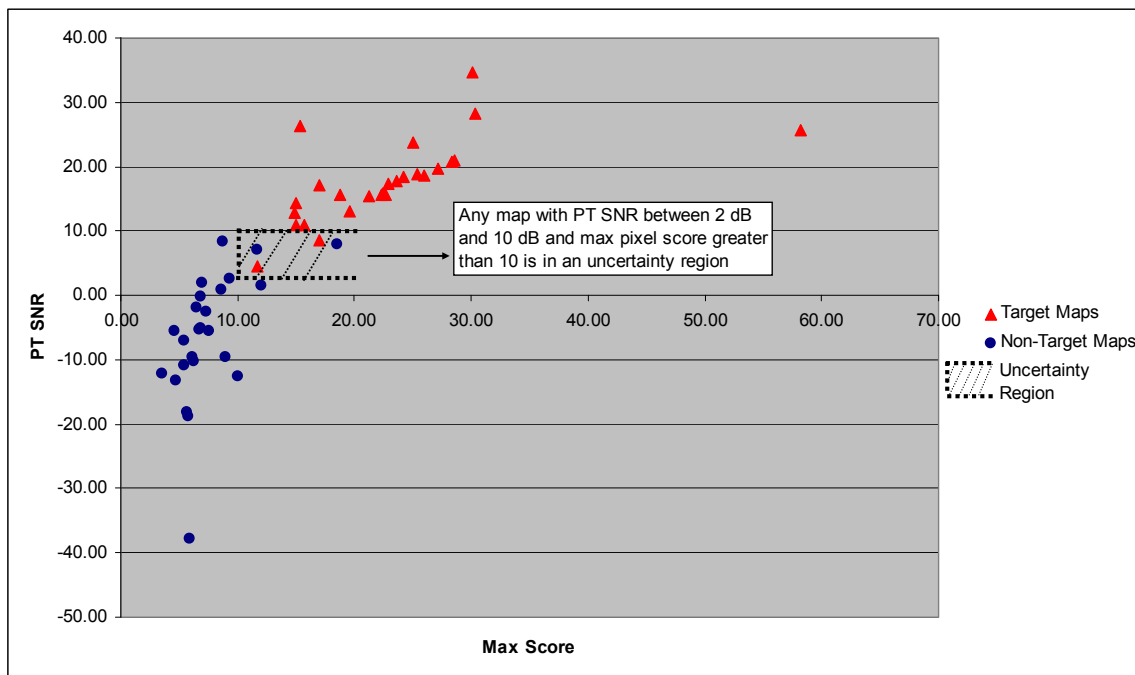
**Figure 3-25. Smallest Possible Uncertainty Region in Max Score and
PT SNR Feature Space that Includes all Target Maps**

Other than defining the smallest uncertainty region that contains all target maps based on the training test set's target and non-target signal values in this two-dimensional feature space, one could use a quadratic discriminant function trained on these values. The quadratic discriminant function defines as follows (Bauer, 2007:88):

$$d_i^Q(\underline{X}_0) = -\frac{1}{2}\ln|S_i| - \frac{1}{2}(\underline{X}_0 - \underline{\mu}_i)^T S_i^{-1}(\underline{X}_0 - \underline{\mu}_i) + \ln(P_i) \tag{3.2}$$

where

$d_i^Q \equiv$ quadratic discriminant score for the ith population denoted ρ_i

$\underline{X}_0 \equiv$ exemplar in question (i.e. the values of a map in the feature space)

$S_i \equiv$ population i sample covariance

$\underline{\mu}_i \equiv$ mean vector for population i in feature space

$P_i \equiv$ prior probability of being in population i

$\underline{X}_0$ belongs to ρ_i if $d_i^Q(\underline{X}_0) = MAX\left(d_1^Q(\underline{X}_0),...,d_g^Q(\underline{X}_0)\right)$

Assuming multivariate normality, using Bayes' rule, the posterior probability of belonging to ρ_i with K populations calculates as follows (Bauer, 2007:95):

$$\Pr(\rho_i|\underline{X}_0) = \frac{\dfrac{1}{(2\pi)^{\frac{P}{2}}|S_i|}\exp\left[-\frac{1}{2}(\underline{X}_0 - \underline{\mu}_i)^T S_i^{-1}(\underline{X}_0 - \underline{\mu}_i)\right]}{\displaystyle\sum_{j=1}^{K}\dfrac{1}{(2\pi)^{\frac{P}{2}}|S_j|}\exp\left[-\frac{1}{2}(\underline{X}_0 - \underline{\mu}_j)^T S_i^{-1}(\underline{X}_0 - \underline{\mu}_j)\right]} \tag{3.3}$$
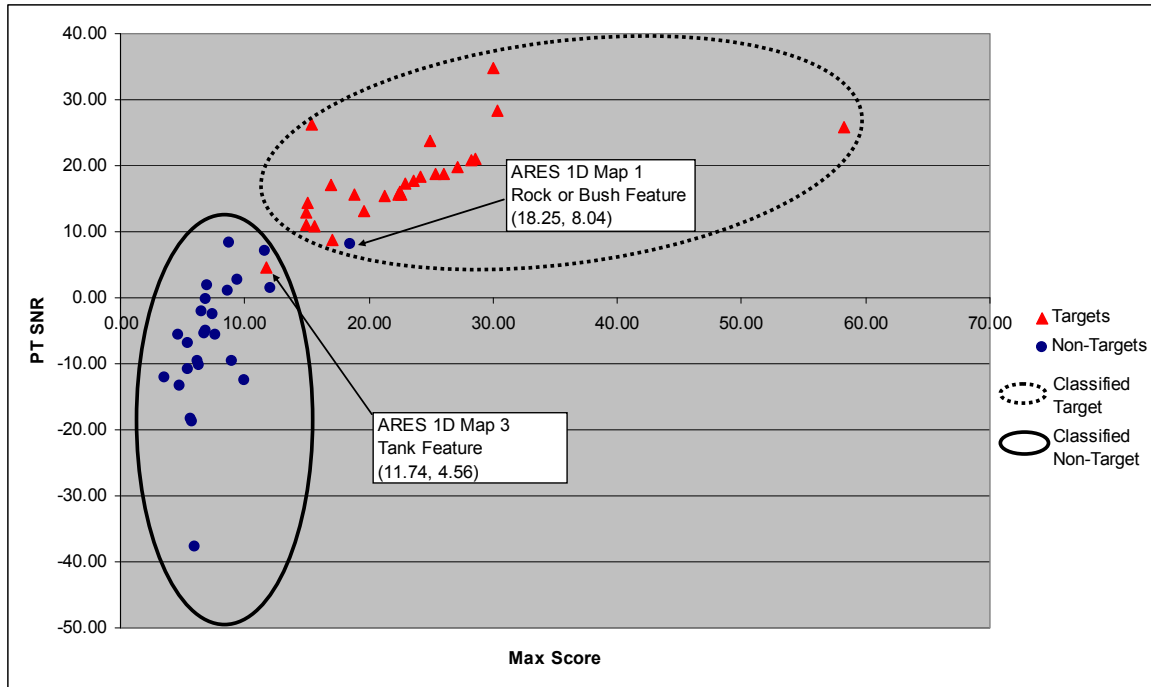
Figure 3-26. Group Classifications Based on Quadratic Discriminant Score with Misclassifications Labeled

Figure 3-26 shows the results of using the quadratic discriminant function to classify the target and non-target maps. Misclassified maps are labeled. The posterior probability of ARES 1D Map 1, a non-target map, belonging to the target population calculates to 0.99 and ARES 1D map 3, a target map, belonging to the non-target population calculates to 0.85.

Wishing to err on the side of including some non-target maps so that no target maps are misclassified as non-target maps given the view of an anomaly detector as a preprocessing algorithm prior to execution of a signature algorithm, all maps that fall into the uncertainty region in Figure 3-25 will be classified as targets in subsequent validation images. Thus, any map with a PT SNR greater than 2 dB and a max pixel score greater

than 10 will be classified as a target. Use of the trained quadratic discriminant function will not be employed.

3.4 Target Pixel Identification Improvements via Iterative Adaptive Noise Filtering

After identifying which IC signal corresponding to the displayed abundance maps isolate potential targets, one must determine from those signals which pixels are targets pixels. As previously discussed the zero-detection histogram method appears from the test images to be well suited to identify which pixels belong to small, rare objects. However, the target signals can have significant noise that can result in several false positive detections. One technique not exploited in the current literature to improve target pixel identification is an adaptive filter. First, one should reshape the ICA signal into the original images pixel length by width, so that targets are again clustered in their true physical location. An adaptive filter has the desirable property of smoothing 'heavily' the part of the signal where the variance is close to overall system noise and smoothing 'little' the part of the signal where the variance is significantly higher than overall system noise. Thus, after completing the adaptive smoothing, the effects of noise during detection are minimized while not significantly reducing the target signals. The effect should improve target detection in terms of increasing TPF and percent TGT while reducing FPF, the measures of performance explained in section 3.1.2.

Let N and M be the pixel length by width respectively of a moving smoothing window. At each stop the mean score and variance of the pixel neighborhood is calculated via (3.3) below.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$

$$\sigma^2 = \left[\frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) \right] - \mu^2$$

(3.3)

where

$a(n_1, n_2) \equiv$ a pixel at location (n_1, n_2) in neighborhood η

$\mu \equiv$ mean pixel score in neighborhood η

$\sigma^2 \equiv$ variance of pixel score in neighborhood η

The current pixel score is replaced with the smoothed or filtered score via (3.4) below.

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} \left[a(n_1, n_2) - \mu \right]$$

(3.4)

where

$b(n_1, n_2) \equiv$ new pixel score

$v^2 \equiv$ system noise variance (noise power level for the signal)

Note: if v^2 is not known the algorithm uses the average of all the locally estimated variances as the estimate of v^2

Consider an example of a neighborhood where σ^2 is high in relation to v^2, i.e. a neighborhood with potential targets. Let $\sigma^2 = 50$, $v^2 = 1$ and $\mu = 15$. Suppose a pixel score in the neighborhood is $a(n_1, n_2) = 12$. Then

$b(n_1, n_2) = 15 + \frac{50 - 1}{50}(12 - 15) = 15 - .98 \cdot 3 = 12.06$. Thus, the new pixel score is nearly the same as the old pixel score. Conversely, consider a neighborhood where σ^2 is similar in relation to v^2, i.e. a neighborhood that could be considered background. Let $\sigma^2 = 1.4$, $v^2 = 1$ and $\mu = 1.9$. Suppose a pixel score in the neighborhood

is $a(n_1, n_2) = 5$, an errant pixel score compared to the mean of its local neighborhood.

Then $b(n_1, n_2) = 1.9 + \dfrac{1.4 - 1}{1.4}(5 - 1.9) = 1.9 - .286 \cdot 3.1 = 1.013$. Thus, the possibly noise

pixel with a score of 5 is reduced substantially to 1.013. Hence, the algorithm is

classified as an adaptive filter due to its changing level of filtering based on local

behavior of the signal. This algorithm exists in the Image Processing Toolbox of

MATLAB named 'wiener2' denoting the algorithm as an adaptive Wiener filter.

An important decision for this algorithm is choice of window size. A window

size too large, could adversely affect target signals by reducing their score. This happens

due to the fact that the larger the window size the smaller the neighborhood variance. As

previously described, for neighborhoods with smaller variance or variance closer to the

system noise variance, the adaptive filter applies more smoothing. Thus, target pixels

will be smoothed more than desired. A smaller window will prevent this occurrence.

However, with a small window, neighborhoods with just background pixels will not be

smoothed as much since smaller neighborhoods will undoubtedly have larger variance

and an adaptive algorithm applies little smoothing to neighborhoods with larger

variances. In order to substantially smooth the background but not target pixels, this

author suggests that the answer is to use a small window size, but repeat the algorithm,

i.e. an iterative smoothing, to continually reduce the power of the background while

maintaining the power of the target signal.

Consider for example ARES 1D map 3 signal presented below in Figure 3-27.

A small window size of 3 x 3 was used and the wiener2 algorithm was applied 20 times.

Notice the power of the target signal was not adversely affected, but the power of the

background was reduced substantially as predicted. Notice further the effect to the

abundance map. The background is now near black with target pixels remaining white.
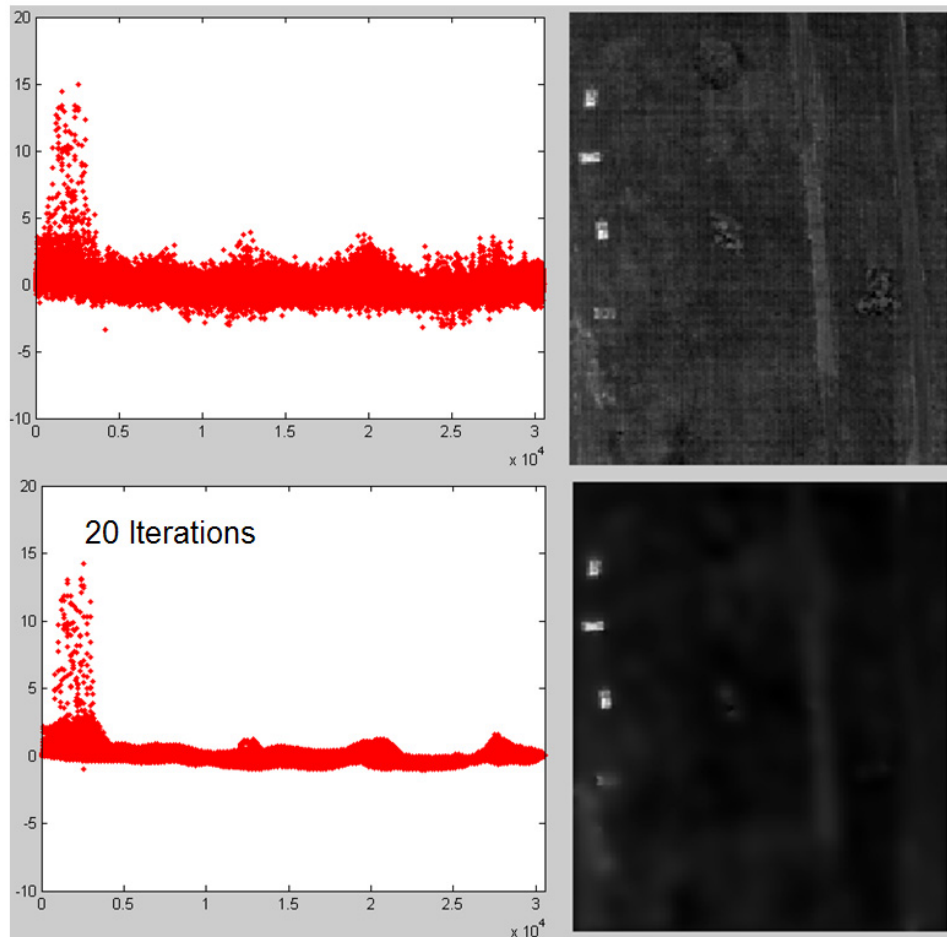
Figure 3-27. ARES 1F: Map 3 Results after Applying
Adaptive Noise Filter with 3 x 3 window size 20 times

Recall the uncertainty region defined in Figure 3-25. To further improve detector

performance in this region, maps that fall into this category will be smoothed more

aggressively since the PT SNR of these maps is lower. Low SNR being indicative of

weaker target signal and stronger noise signal, completing more iterations of the adaptive

filter should improve detector performance of true target maps in the uncertainty region

by reducing the presence of the stronger background power. Figure 3-28 illustrates the

benefit of increased iterations of the adaptive noise filter in this region.
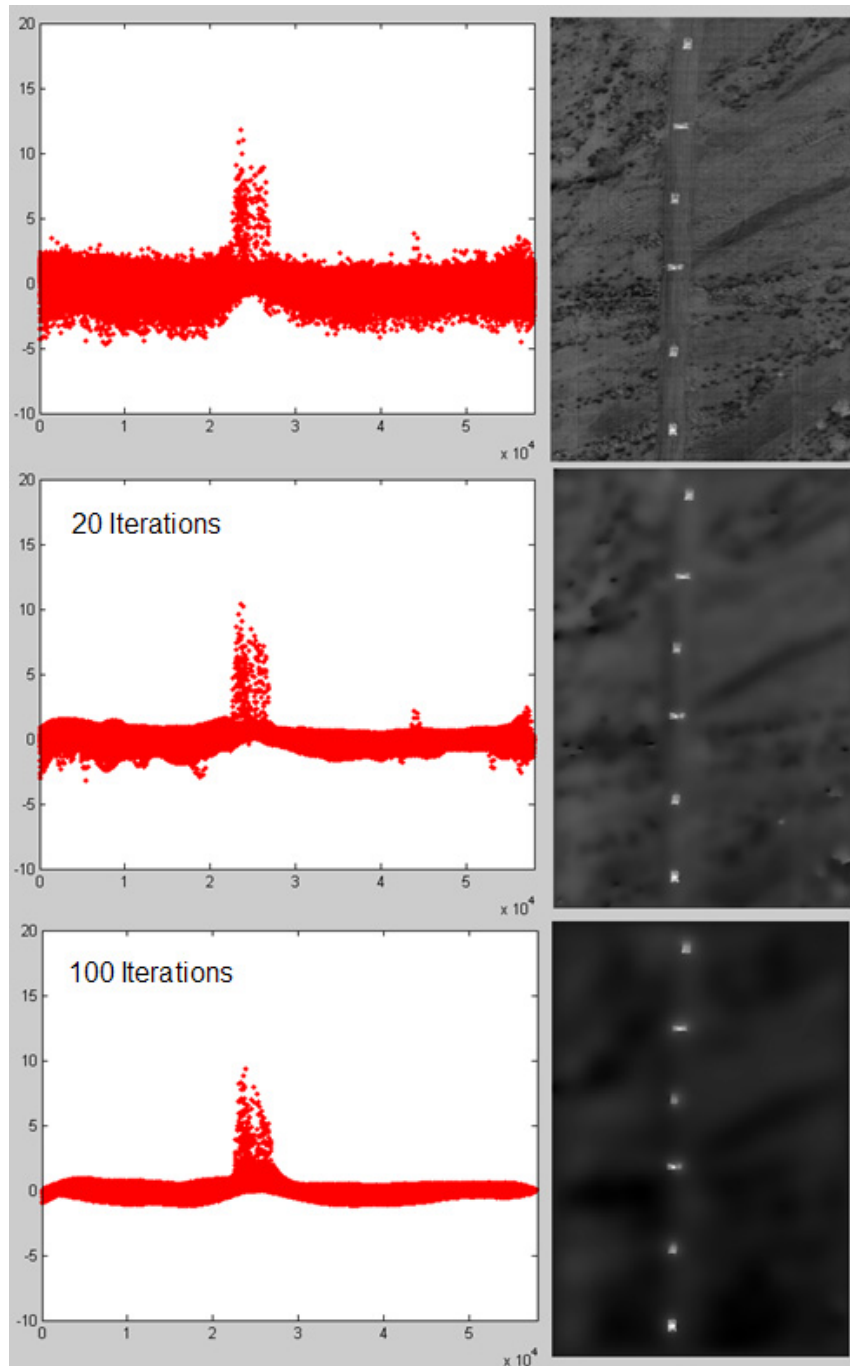
**Figure 3-28. ARES 1D: Map 3 Results after Applying
Iterative Adaptive Noise Filter with 3 x 3 window size 20 and 100 times**

After target feature selection for ARES 1D, maps 1 and 3 are selected as potential target maps based on meeting the criteria of having PT SNRs above 2 dB and max pixel scores above 10. Figure 3-29 details the target identification process without adaptive noise filtering using the zero-detection histogram method to set the threshold between background pixels and target pixels. On the left are the IC signals corresponding to the abundance maps in the middle. The line in the signals represents threshold determined by the zero-detection histogram with a bin width of 0.05. The right of the figure presents the pixels identified as potential targets.
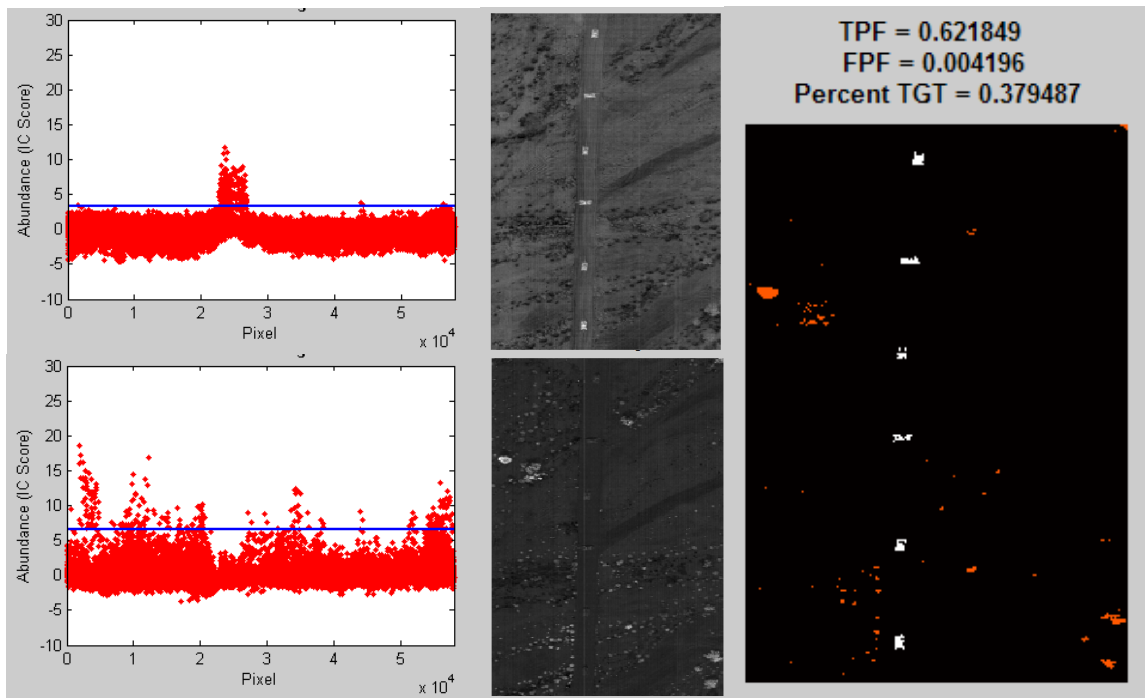
Figure 3-29. ARES 1D: Target Identification without Iterative Adaptive Noise Filtering, False Positives in Red

For contrast Figure 3-30 details the target identification with iterative (100 iterations) adaptive noise filtering. As predicted, TPF improved from 71.2% to 84.0%, FPF decreased from 0.42% to 0.19%, and percent TGT increased from 41.6% to 65.6%.

Notice that for the retained non-target map, the adaptive noise filtering removed all

signals except the strongest signals corresponding to the upper right and lower left large

rock or bush feature. The numerous smaller rocks or bushes were removed being

smoothed into the background. In conclusion, only the strongest, rare, and small class

signals remain, a desirable effect for a global anomaly detector that precedes a signature

matching algorithm.
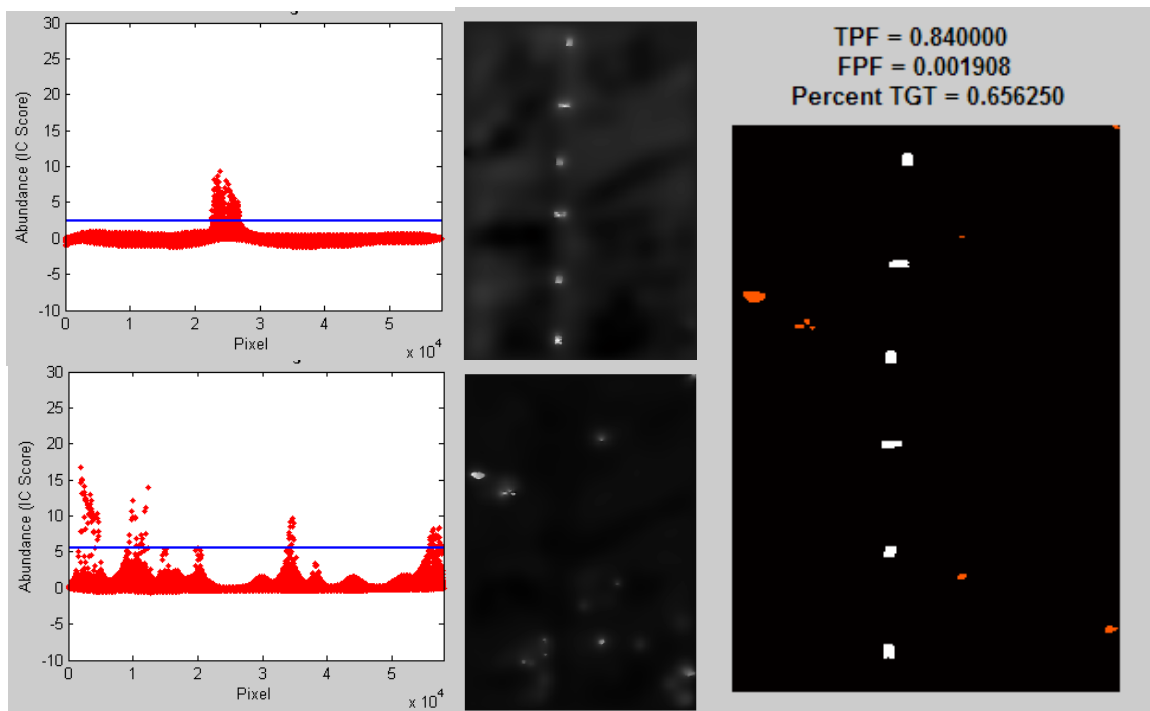
**Figure 3-30. ARES 1D: Target Identification with Iterative
Adaptive Noise Filtering, False Positives in Red
100 Iterations Due to Both Classes Being in Uncertainty Region**

The following figures show target identification for the remaining test images

without and with iterative adaptive noise filtering.

TPF = 0.882516
FPF = 0.002069
Percent TGT = 0.939901

TPF = 0.962298
FPF = 0.001664
Percent TGT = 0.956289

9/9 Targets

9/9 Targets
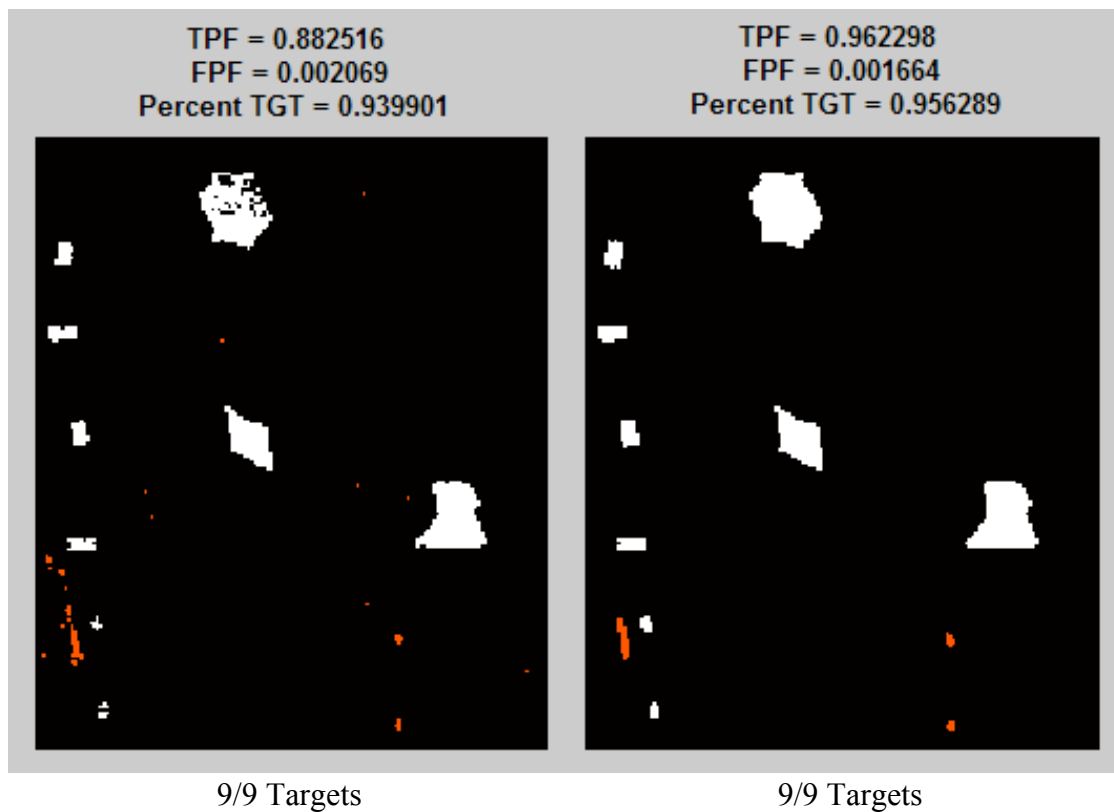
**Figure 3-31. ARES 1F: Target Identification without (left) and with (right)
Iterative Adaptive Noise Filtering, False Positives in Red**

TPF = 0.891813
FPF = 0.002761
Percent TGT = 0.701149

TPF = 0.953846
FPF = 0.000702
Percent TGT = 0.918519

30/30 Targets   29/30 Targets
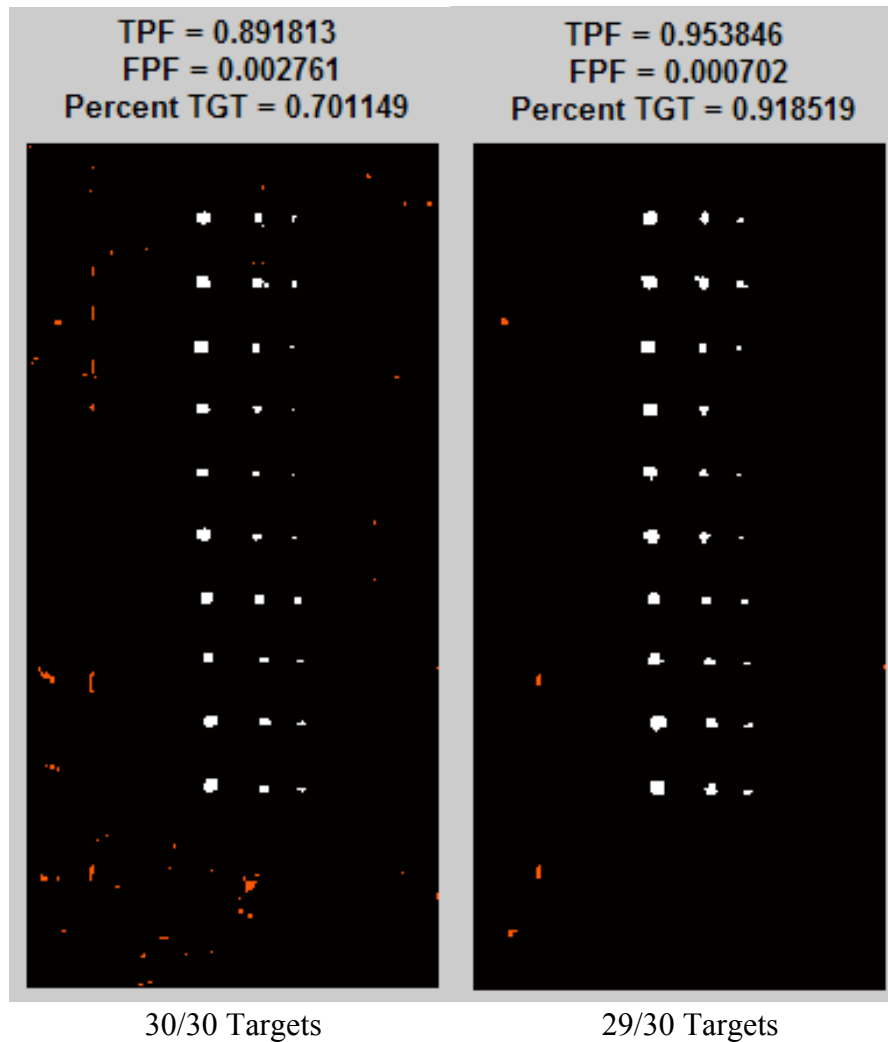
**Figure 3-32. ARES 2F: Target Identification without (left) and with (right)
Iterative Adaptive Noise Filtering, False Positives in Red**
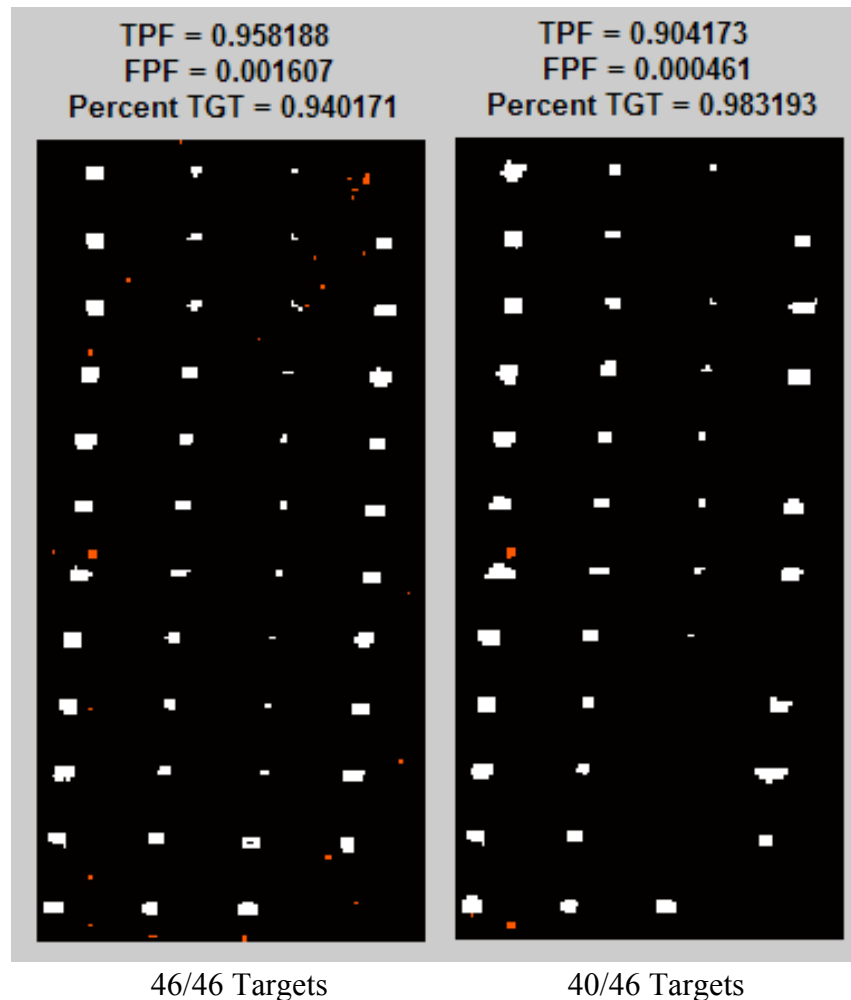
Figure 3-33. ARES 2D: Target Identification without (left) and with (right) Iterative Adaptive Noise Filtering, False Positives in Red

For all cases FPF reduced and percent TGT increased. With the exception of ARES 2D, TPF increased. The reduction in TPF in ARES 2D resulted from the elimination of some of the true targets during smoothing. Also, although TPF improved for ARES 2F, one target was eliminated during smoothing. For some images with extremely small targets (only a few pixels in size), this could be an unavoidable tradeoff. Otherwise, the user may wish to perform less or no smoothing if expected targets are extremely small.

3.5 New Unsupervised Target Detection Algorithm (AutoGAD)

This section details the proposed unsupervised target detection algorithm in
Figure 3-34 based on the results from the test images of the preceding sections. New
contributions from this author to the field of global anomaly detection using ICA to solve
the LMM include:

1. Autonomous dimensionality determination via MDSL.

2. Autonomous selection of target features based on a PT SNR filter which uses
   the zero-detection histogram method to determine separation between
   potential targets pixels and background pixels. After this determination,
   pixels above threshold are used to calculate potential target signal power and
   below threshold are used to calculate background noise power.

3. Use of a secondary autonomous target feature filter, max pixel score, whose
   threshold must also be met in addition to the PT SNR filter. Thus, a two
   dimensional feature space is created rather than the one dimensional KV
   feature space.

4. Use of an iterative adaptive noise (coined IAN) filtering technique on selected
   target feature signals prior to using the zero-detection histogram method to
   autonomously identify target pixels.

The algorithm will be referred to as the Autonomous Global Anomaly Detector or
AutoGAD. The algorithm is detailed in Figure 3-34.
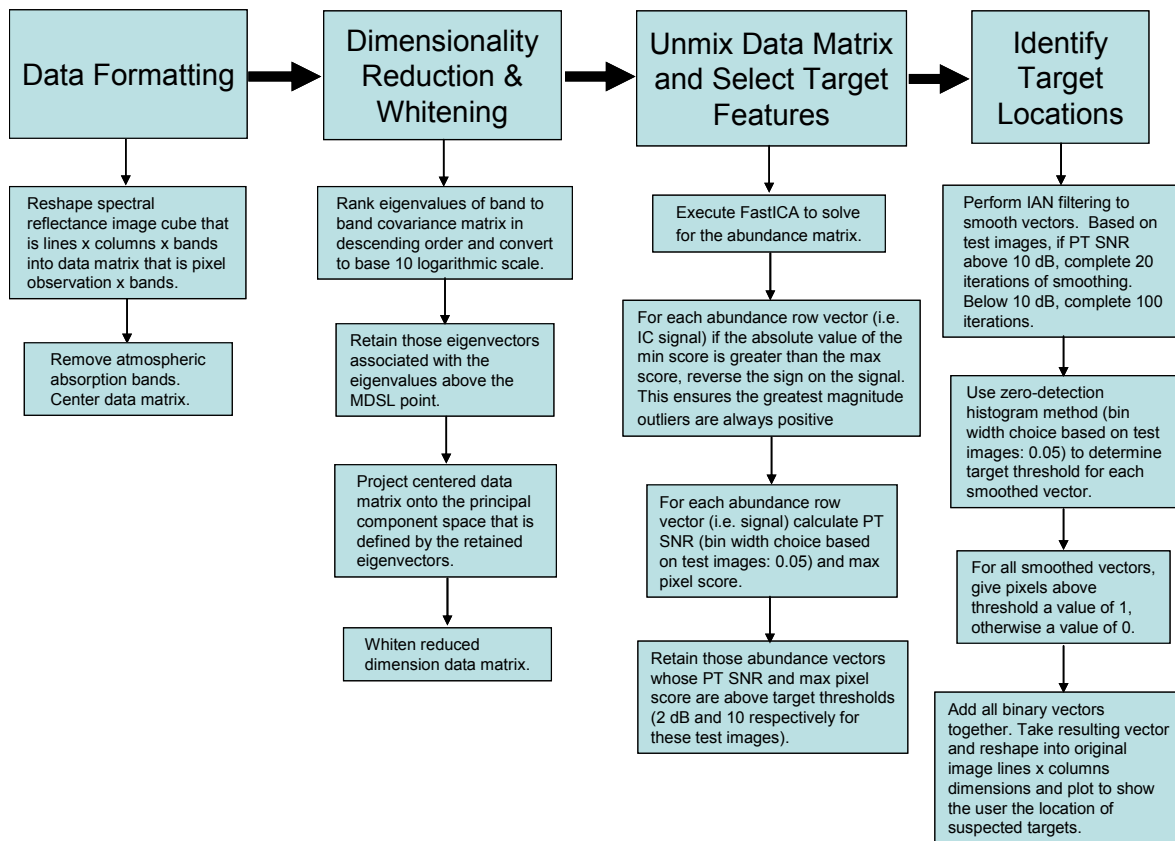
Figure 3-34. AutoGAD Target Detection Algorithm

The AutoGAD algorithm in Figure 3-34 was applied to each of the test images 100 times to check for variability across the three measures of performance and computational time. An additional measure of performance included will be the number of targets detected out of the total present for each iteration. A target will be considered detected if at least one pixel from the target was detected. Table 3-11 details the results. Since computational time is influenced by file size, Table 3-12 details the total number of elements (pixels x bands) of each image before and after dimensionality reduction.

Table 3-11. AutoGAD Algorithm Results for Test Images (100 Reps)

	Mean TPF	Var TPF	Mean FPF	Var FPF	Mean Percent TGT	Var Percent TGT	Mean Time (sec)	Var Time	No. TGTs Detected
ARES 1F	0.96	4.03E-30	0.0017	1.19E-36	0.96	6.10E-31	4.74	0.609	9 / 9
ARES 2F	0.96	2.06E-05	0.0012	3.49E-07	0.87	3.08E-03	41.01	24.968	29 / 30
ARES 1D	0.84	4.03E-30	0.0025	1.28E-07	0.60	1.31E-03	6.64	0.284	6 / 6
ARES 2D	0.90	2.05E-06	0.0002	1.29E-08	0.99	1.72E-05	2.99	0.060	40 / 46

	Min TPF	Max TPF	Min FPF	Max FPF	Min Percent TGT	Max Percent TGT
ARES 1F	0.96	0.96	0.0017	0.0017	0.96	0.96
ARES 2F	0.95	0.97	0.0004	0.0028	0.74	0.95
ARES 1D	0.84	0.84	0.0019	0.0028	0.56	0.66
ARES 2D	0.90	0.91	0.0001	0.0005	0.98	1.00

Table 3-12. Test Image Statistics

ARES 1F	
Total No. of Elements	
Original Image	30,560 pixels x 210 bands = 6,417,600
After Removal of Absortion Bands	30,560 pixels x 145 bands = 4,431,200
After PCA (MDSL Decision)	30,560 pixels x 15 bands = 458,400

No. Target Pixels = 1980

ARES 2F	
Total No. of Elements	
Original Image	47,424 pixels x 210 bands = 9,959,040
After Removal of Absortion Bands	47,424 pixels x 145 bands = 6,876,480
After PCA (MDSL Decision)	47,424 pixels x 12 bands = 569,088

No. Target Pixels = 1528

ARES 1D	
Total No. of Elements	
Original Image	57909 pixels x 210 bands = 12,160,890
After Removal of Absortion Bands	57909 pixels x 145 bands = 8,396,805
After PCA (MDSL decision)	57909 pixels x 8 bands = 463,272

No. Target Pixels = 672

ARES 2D	
Total No. of Elements	
Original Image	22,360 pixels x 210 bands = 4,695,600
After Removal of Absortion Bands	22,360 pixels x 145 bands = 3,242,200
After PCA (MDSL Decision)	22,360 pixels x 13 bands = 290,680

No. Target Pixels = 2465

As shown in Table 3-11 on the previous page, the AutoGAD algorithm yields relatively consistent results in terms of low variability across TPF, FPF, and percent TGT. Also, the number of targets detected for each image remained the same for each run. Further, note the low computational times, all under a minute and for three of the images under 10 seconds. Recall the computational times of 636 and 2,724 seconds (10.6 and 45.4 minutes) for ARES 1F and ARES 1D respectively that resulted from computing the MSV in Koo's two-phase filtering approach. Further, the two-phase filtering approach could not be validated to consistently nominate the same target features. According to Taitano, who used a local anomaly detector on ARES 1D, his Iterative RX detector (using a suggested local scanning window size of 21) took on average 290 seconds (4.83 minutes) per iteration. Retaining 5 principal components, his algorithm required 33 iterations to converge to a solution. Thus, the average run time was 9,570 seconds (159.9 minutes) (Taitano, 2007:85). Further, his solution for ARES 1D had a TPF of 0.166 and an FPF of 0.01 (Taitano, 2007:51). AutoGAD's run time for ARES 1D was on average **6.64 seconds** and had a mean **TPF of 0.84** and **FPF of 0.0025**. AutoGAD represents a substantial leap in performance in terms of accuracy and run time based on this comparison.

In chapter four, the AutoGAD algorithm will be applied to each of the validation images. Further, sensitivity analysis will be performed to check the sensitivity of the zero-detection histogram method in the neighborhood of the 0.05 bin width decision for all test and validation images. The method is used in two phases of the detection process. First in the target feature selection phase, this method is used to determine the breakpoint between potential target and background for the PT SNR

calculation. Second, the method is used to identify the target pixels in the selected target features. If the neighborhood around this decision point in the target feature selection and target identification phases across the test and validation images is not volatile in terms of variability in the response (TPF, FPF, percent TGT), then one might conclude that the decision is possibly a robust one for images other than the ones available for this research.

IV. Results and Analysis

4.1 Validation Images

The AutoGAD algorithm will be applied to the images in Figure 4-1 and 4-2. ARES 4F in Figure 4-1 will be especially challenging due to partially hidden and some completely hidden targets under the tree line. Note, the images in Figure 4-2 are absent of targets. Since it is likely most target searches in rural environments will survey areas with no true targets, images with no targets should be included in the testing of the AutoGAD algorithm.

ARES 3F
20 Targets

ARES 4F
29Targets
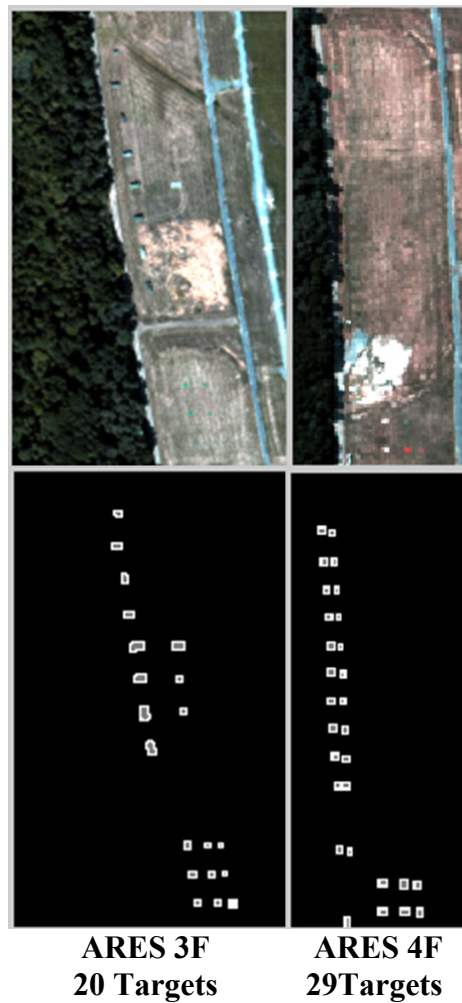
Figure 4-1. HSI Validation Images with Targets

ARES 1C   **ARES 2C**
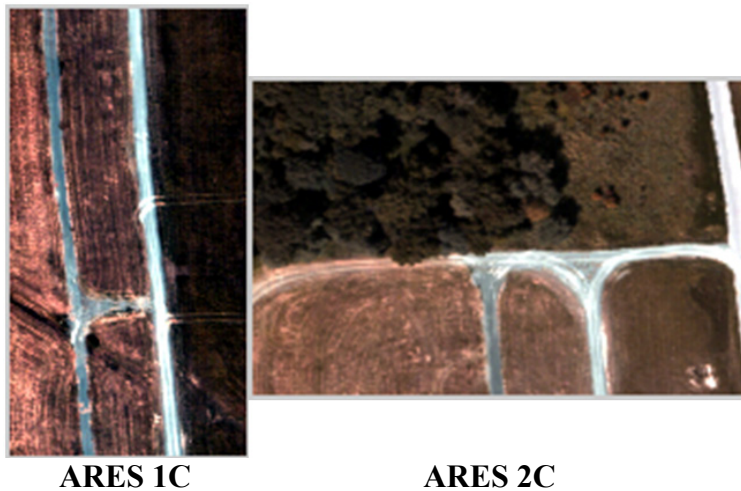
Figure 4-2. HSI Validation Images without Targets
('C' denotes Clear of Targets)

4.2 Overview of Validation Image Results Presentation

For each of the validation images, the section 4.3 will detail:

1. The dimensionality decision via MDSL.

2. The abundance maps and independent component signals with decision line between potential target and background via zero detection histogram method. PT SNR and max pixel score will display above each signal.

> *Note, maps are no longer presented in order of KV.*
> *Maps will be presented in the random ICA solution*
> *order.*

3. Target signals and target abundance maps after IAN filtering.

4. Binary target identification map without and with IAN filtering with TPF, FPF, and percent TGT labeled.

Section 4.4 will include:

1. Results from 100 replications to check for variation in TPF, FPF, and percent TGT.

2. All validation image abundance vector's values in the PT SNR and max pixel score feature space in the same scale as the test image feature space in Figure 3-26.

3. Sensitivity analysis of the histogram bin width decision for target feature selection and target identification.

4.3 Validation Image Results

Results for ARES 3F
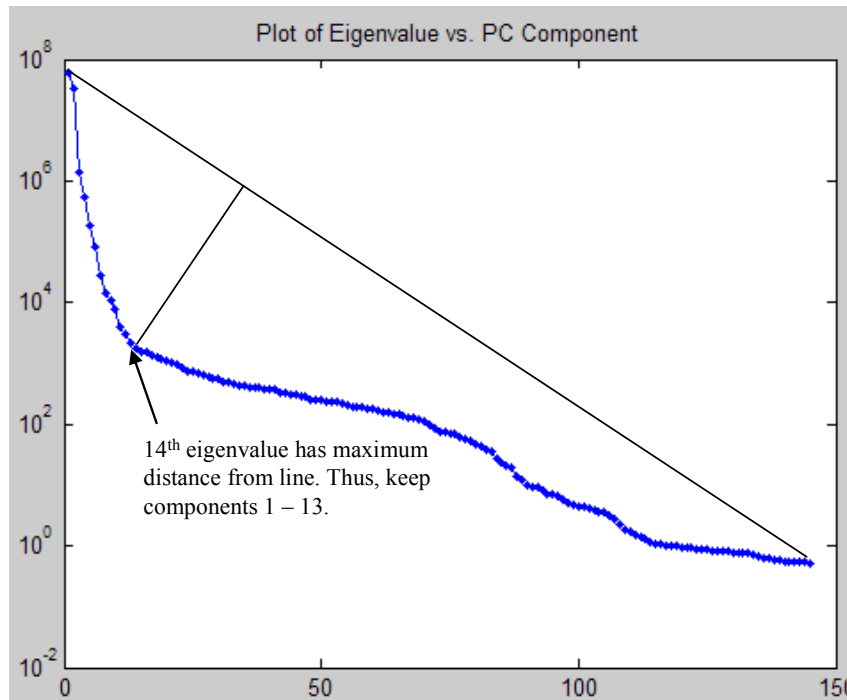
Figure 4-3. ARES 3F: Dimensionality Decision via MDSL

As in the test images, the shape of the eigenvalue curve on a log scale reveals the tilted ramp that locates the eigenvalues of the covariance matrix of the noise term in the LMM. The MDSL method effective locates the 'knee' and as evidenced by Figure 4-4, this dimensionality determination is sufficient to separate target classes from non-target classes after unmixing the reduced dimension data set.
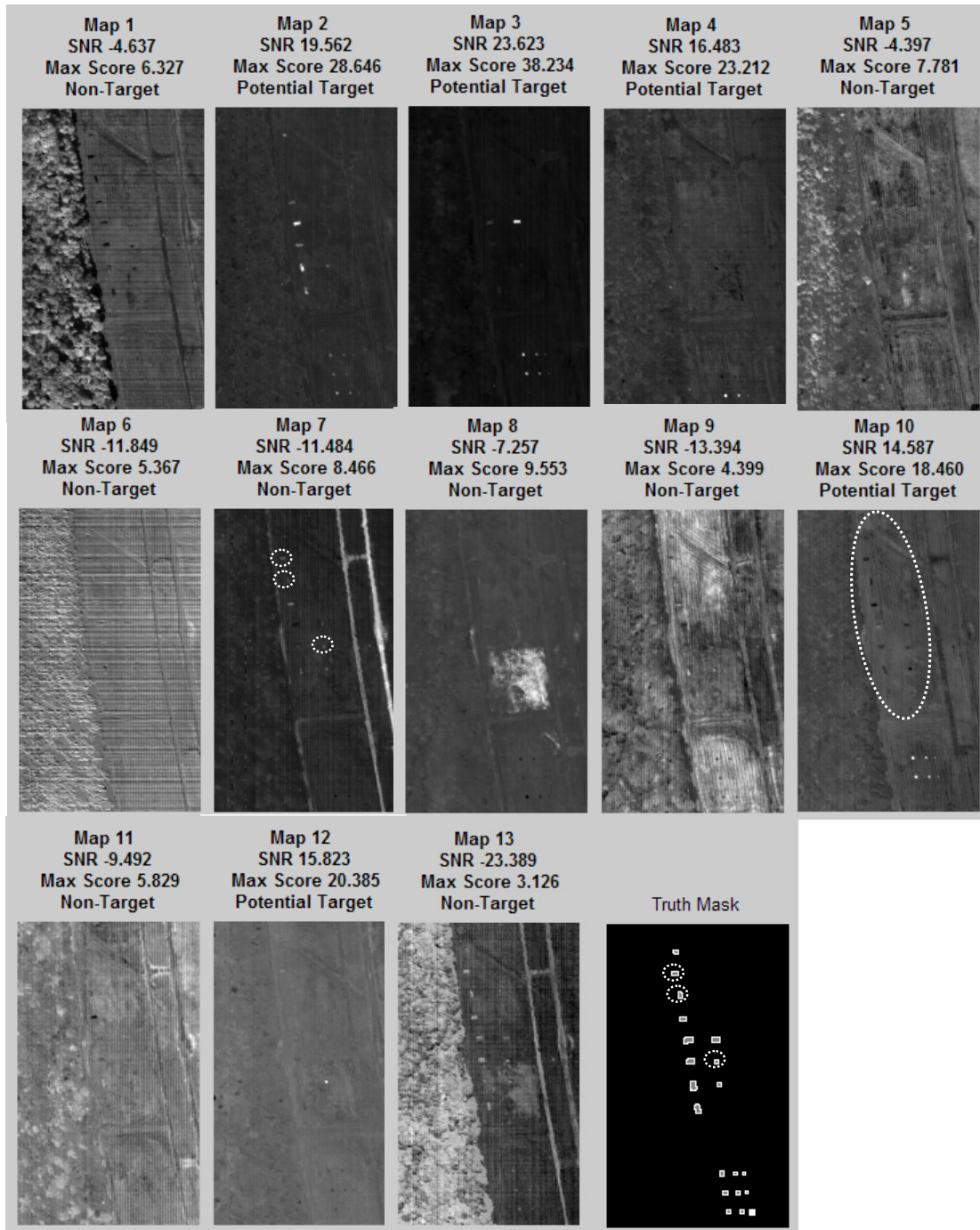
Figure 4-4. ARES 3F: Abundance Maps from 13 ICs via MDSL Decision
Maps 2, 3, 4, 10, and 12 isolate 17 out of 20 targets when considering only positive outliers (white appearance). Targets not isolated are circled on truth mask and in map 7 where they are convoluted with the non-target class of road. Circled targets in map 10 illustrate negative outlier targets. Considering positive and negative outliers, 20 out of 20 targets are isolated.

Notice during the target feature selection phase in Figure 4-4, AutoGAD correctly selected the maps that isolate target features based on the new filters of PT SNR and max pixel score. Non-target classes had PT SNRs below 0 dB, while target classes had PT SNRs well above the PT SNR threshold of 2 dB and max pixel score of 10.

Further, notice map 10 isolates some of the panel targets in the lower part of the image denoted by their white pixel appearance with respect to the rest of the image. Thus, these pixels have positive outlier scores on the corresponding signal plot. Circled in map 10 are target pixels that appear near black, some of which are the same targets that are convoluted with the road feature as circled in map 7 when considering only positive outlier signal scores. These pixels have outlying negative scores on the corresponding signal plot for map 10. Thus, one has an IC with target pixels that are both positive and negative outliers on the signal. This example illustrates one of the pitfalls with ICA not conforming to the non-negativity constraint of the LMM. Recall that after the application of FastICA, due to the ambiguity of the sign of the scores, the author's proposed algorithm reorients each signal such that the highest absolute magnitude score is made positive. During target identification only the side of the signal with the highest absolute magnitude score is considered when using the zero-detection histogram to determine the target threshold. Perhaps after IAN filtering, thresholding can be accomplished to check for positive and negative outliers. The algorithm in Figure 3-34 will be modified to compare results of thresholding just the positive side of the signal and thresholding both sides.
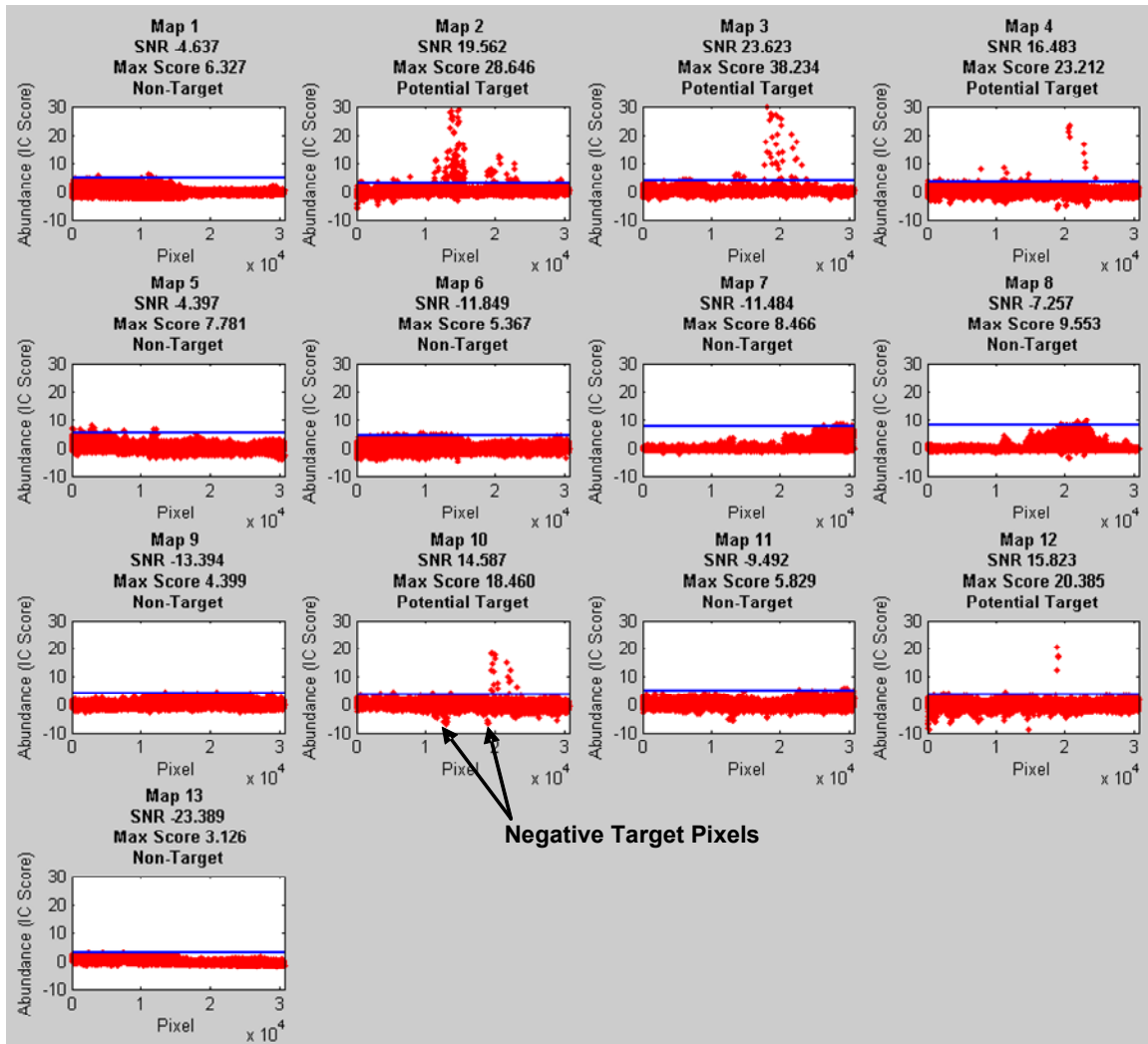
**Figure 4-5. ARES3F: PT SNR and Max Pixel Score for Each Signal
with Potential Target Threshold Lines**

Figure 4-5 illustrates the zero-detection histogram method effectively determining

the breakpoint between outlier pixels (potential targets) and the background. In the 5

selected signals denoted by the potential target label, notice visually the difference in

variability between the pixels above the threshold versus those below. Hence those

signals have PT SNRs higher than the PT SNRs of the other signals. Further notice in

map 7 and 8 the 'bulge' in the corresponding signal plots. These bulges correspond to

the large classes of road and bare earth patch respectively in map 7 and 8 in Figure 4-4. Given these are large classes, as discussed previously, the tails of the frequency distribution will be 'fatter' relative to those signals with targets. Thus, the first zero bin does not occur until the near the end of the tail as evidenced by the breakpoint in the signal plot not occurring until the top of the bulge. Thus, as in the test images, large non-target classes have low PT SNR.
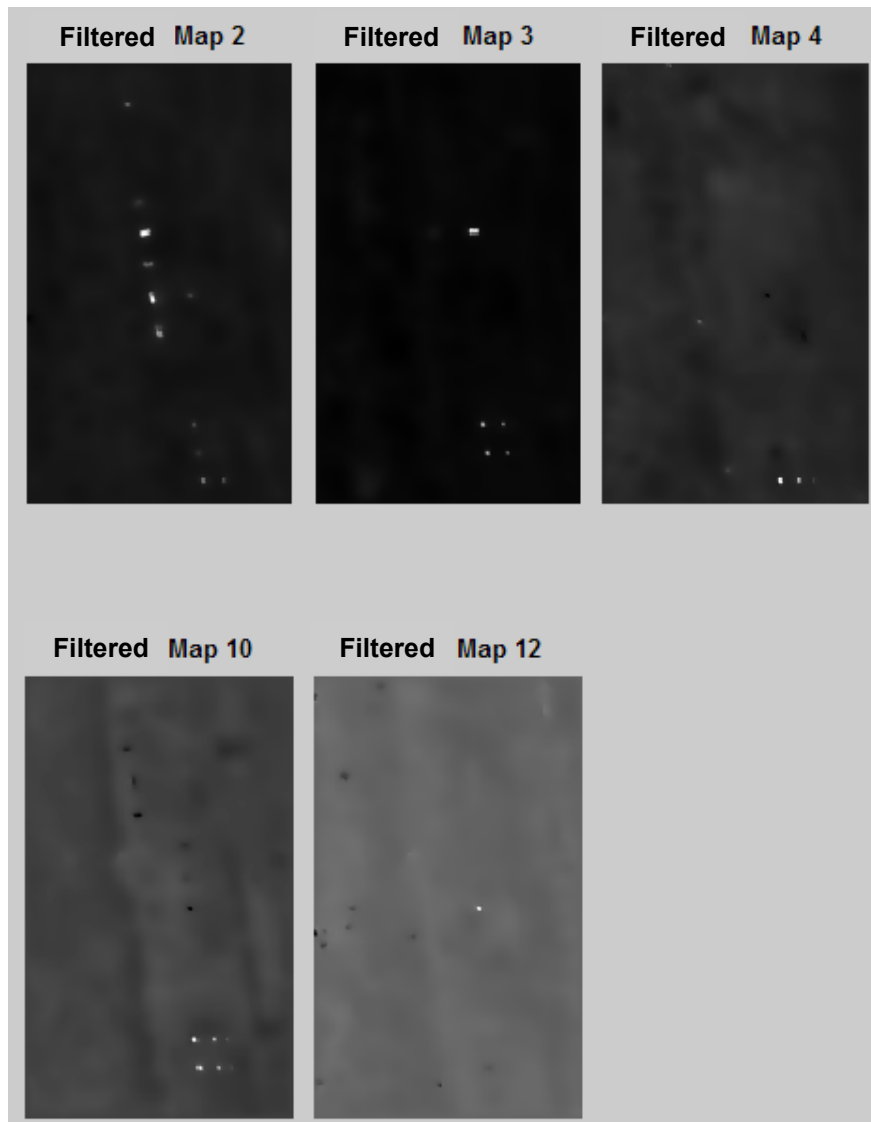
Figure 4-6. ARES 3F: Target Abundance Maps after 20 iterations of IAN Filtering

As intended, notice in the abundance maps in Figure 4-6 on the previous page and below in the signal plots in Figure 4-7 how after IAN filtering much of the detail associated with the background has been 'smoothed' out while target detail is still apparent.
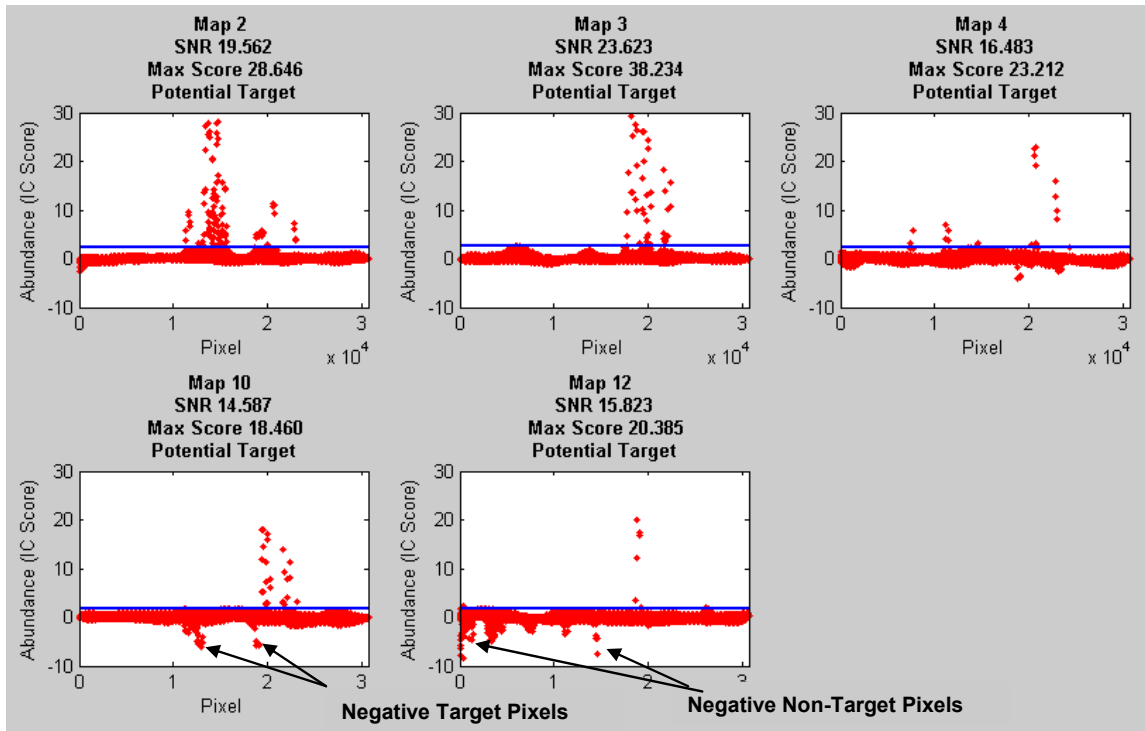
Figure 4-7. ARES 3F: Target Signals after IAN Filtering with Positive Signal Target Identification Threshold

Also shown in Figure 4-7, in addition to target pixels that can be negative outliers, thresholding to check for both positive and negative outliers in selected target feature signals will include more false positive detections for situations where non-target pixels are also negative outliers. Positive and negative thresholding using the zero-detection histogram method is detailed in Figure 4-8. The additional thresholding on the negative

side will include the negative outlying target pixels in the signal for map 10 but also

include false positive pixels corresponding to the negative outliers in the signal for map

12.
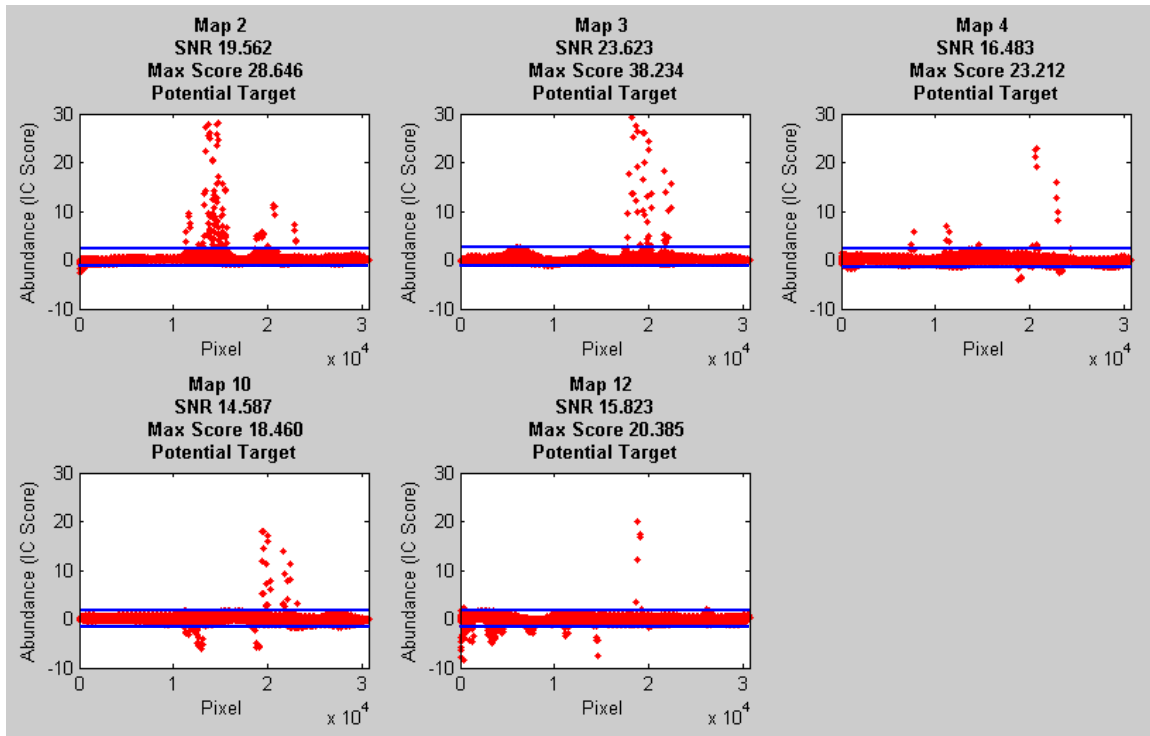
**Figure 4-8. ARES 3F: Target Signals after IAN Filtering with
Positive and Negative Signal Target Identification Threshold**
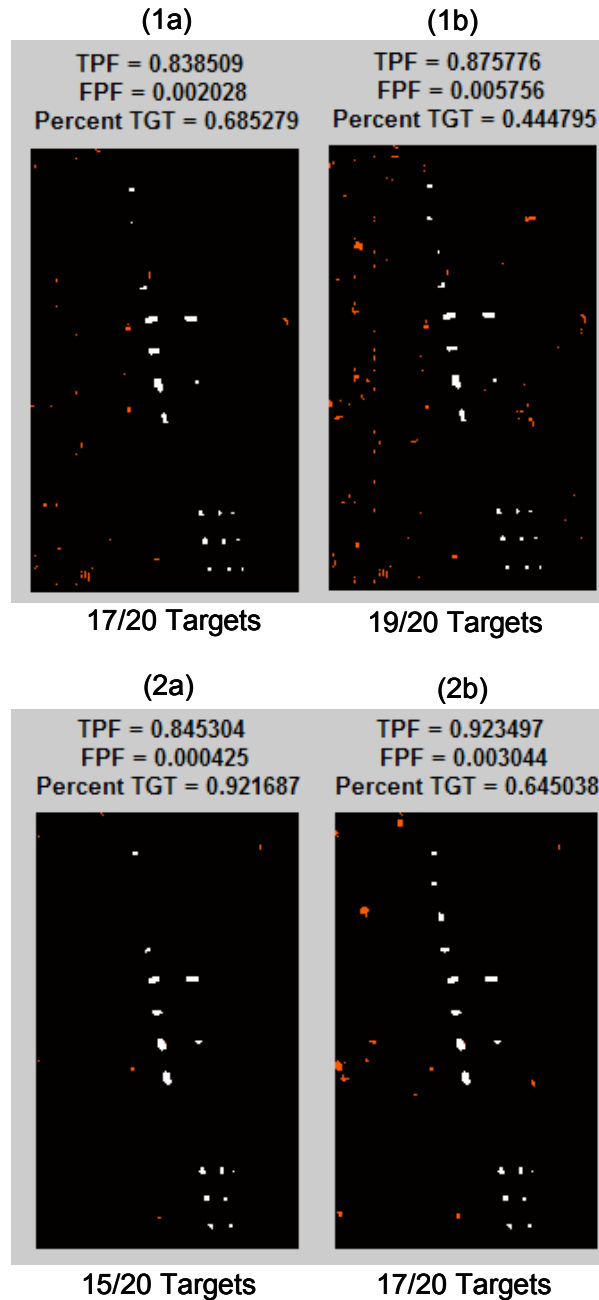
**Figure 4-9. ARES 3F: Target Identification, False Positives in Red
(1) without and (2) with IAN Filtering
(a) Positive Threshold (b) Positive and Negative Threshold**

As shown in Figure 4-9, IAN filtering reduces FPF and increases TPF and percent TGT

when comparing (1) images to (2) images. Although, TPF increased, 2 smaller panel

targets (lower right) where eliminated during the smoothing in (2) images. Further, when comparing (a) images to (b) images, thresholding on both sides results in an increase in the number of targets detected and TPF, but increases FPF and decreases percent TGT as discussed when analyzing the signal plots in Figure 4-7. The user should be aware of these tradeoffs when employing the algorithm and choose whether or not to threshold on both sides and/or employ IAN filtering depending the user's priorities concerning the measures of performance, TPF, FPF, percent TGT, and number of targets detected.

Results for ARES 4F
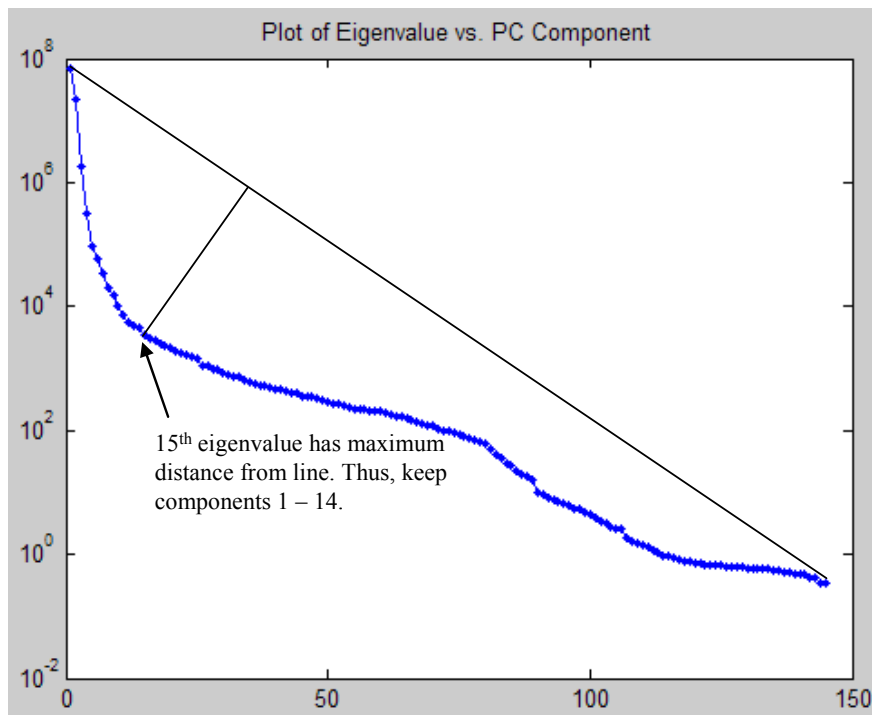
Figure 4-10. ARES 4F: Dimensionality Decision via MDSL

As with ARES 3F, again the eigenvalue curve reveals the tilted ramp and the MDSL method effectively locates the knee.
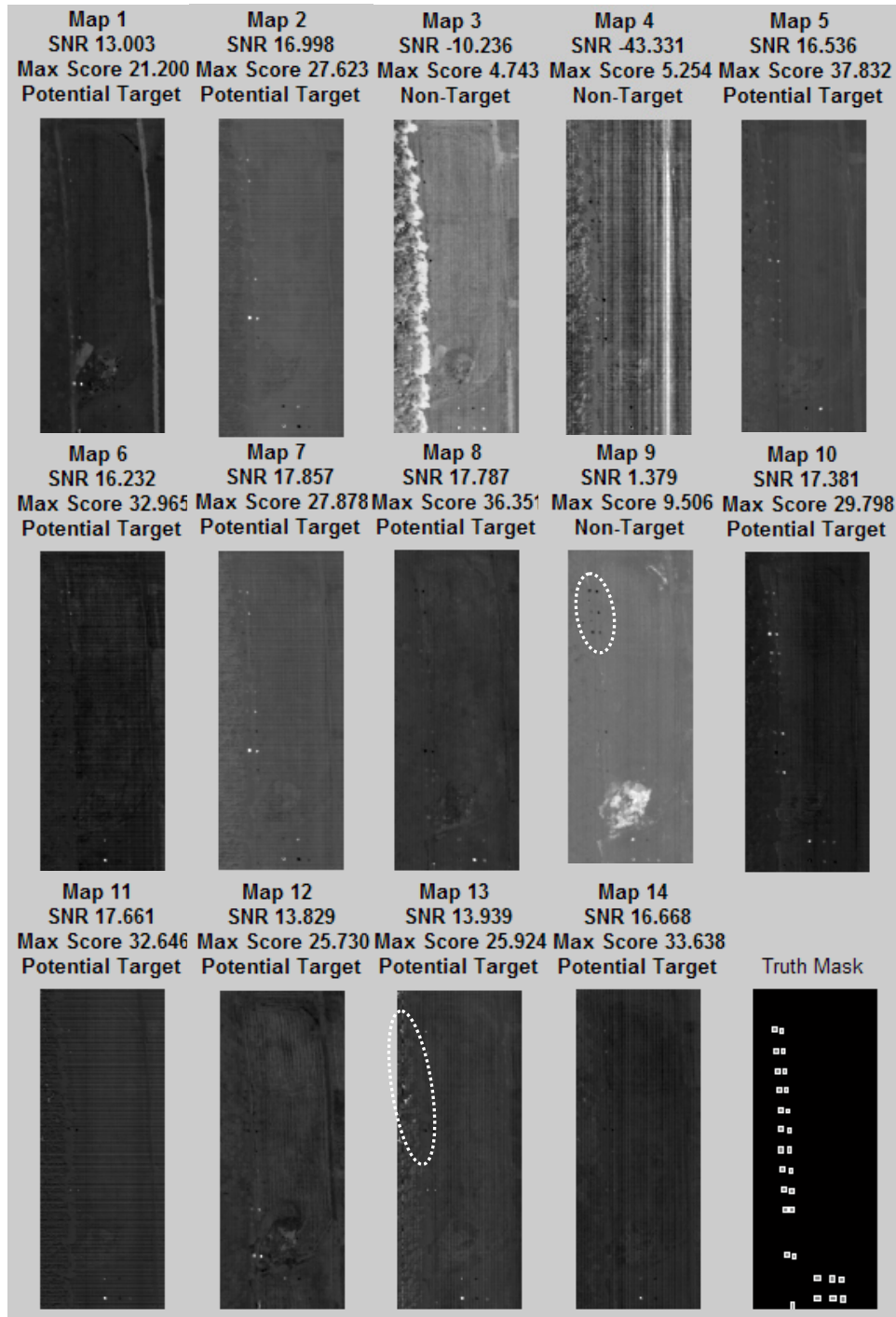
Figure 4-11. ARES 4F: Abundance Maps from 15 ICs via MDSL Decision
Maps 1, 2, 5, 6-8, 10-12, and 14 isolate 29 out of 29 targets. Although, some targets have weak positive signal as evidenced by the low white target pixel intensity in the gray scale. Although not selected as a potential target map due to large class of bare earth, circled in map 9 are negative outlier target pixels. Further, circled in map 13 are intense pixels corresponding to trees, a non-target class convoluted with target classes in the same map.

AutoGAD correctly selects the maps that isolate target features based on the PT SNR and max pixel score filter except for map 13. Although, some target's pixels are highlighted in the map, some intense pixels corresponding to trees are also isolated. This will contribute to more false positives during target identification. Also, notice in map 9, a non-target feature of bare earth, targets appear as negative outliers. However, this feature was not selected due to PT SNR begin calculated by considering the power (potential target pixel variability) of the positive part of the signal only. Whereas AutoGAD was modified to threshold on both sides of the signal for ARES 3F, this modification only applied to those maps kept during target feature selection. During target feature selection PT SNR is only calculated for the positive outliers in the IC signal. Perhaps, a future modification to AutoGAD would also calculate the PT SNR of the negative side of the signal so that maps with negative outliers are not necessarily discluded from entry into the target identification phase as was the case here due to the PT SNR of the positive side falling below the PT SNR threshold. If the PT SNR of one side was greater than that of the other side and above the PT SNR threshold for targets, then that map could be selected as a target map. During the target identification phase, the side with the higher PT SNR would only considered during thresholding to locate target pixels. This endeavor will be left to subsequent researchers employing the AutoGAD algorithm as an area of future research. For this research effort, PT SNR will continue to be calculated from the positive side of the signal which is determined during initial signal processing to be the side with the highest absolute magnitude score.
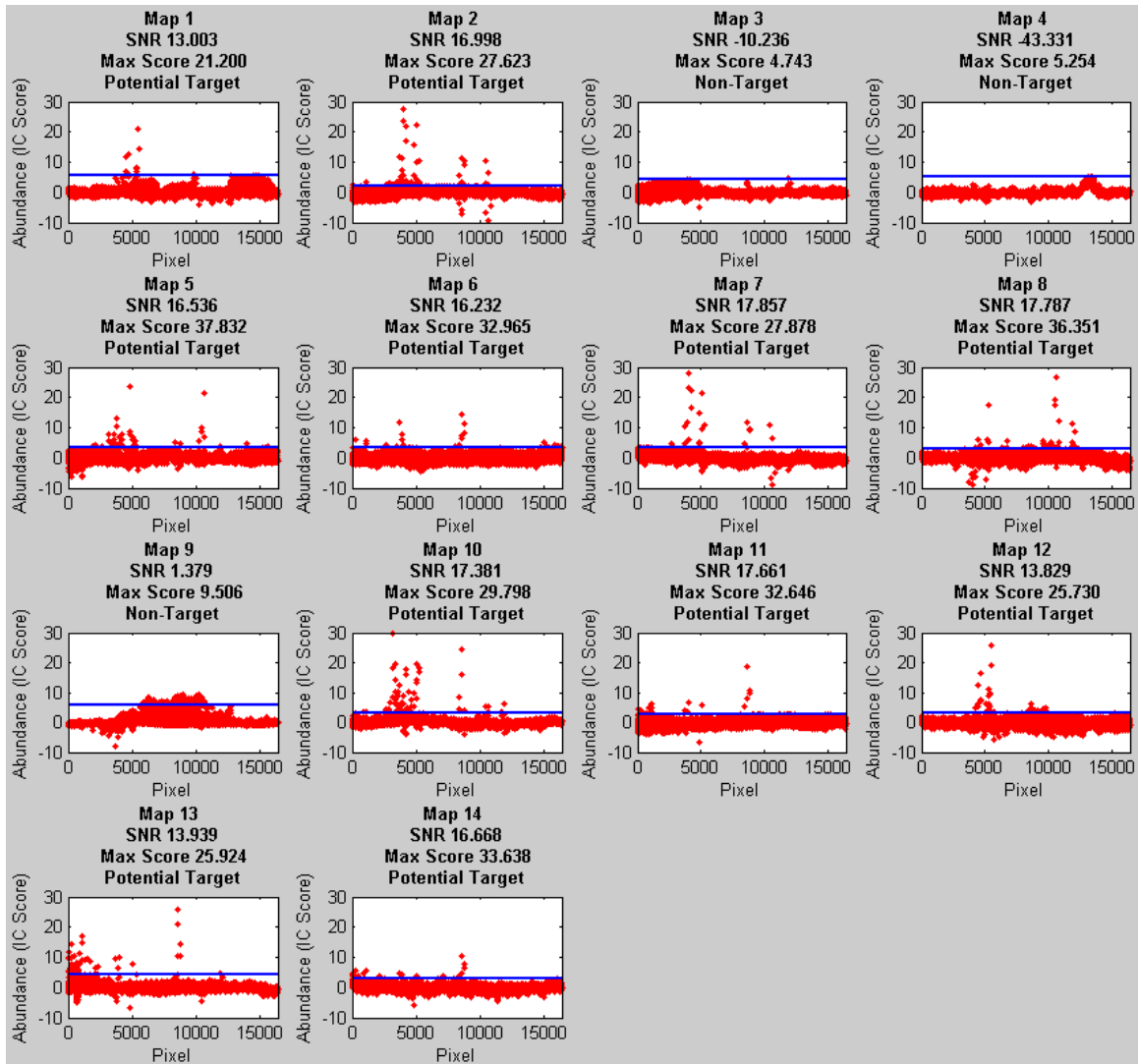
**Figure 4-12. ARES 4F: PT SNR and Max Pixel Score for Each Signal
with Potential Target Threshold Lines**

As with ARES 3F, Figure 4-12 illustrates the zero-detection histogram method again effectively determining the breakpoint between outlier pixels (potential targets) and the background. Notice in the signal corresponding to map 13, the tree outlier pixels on the left of the signal.
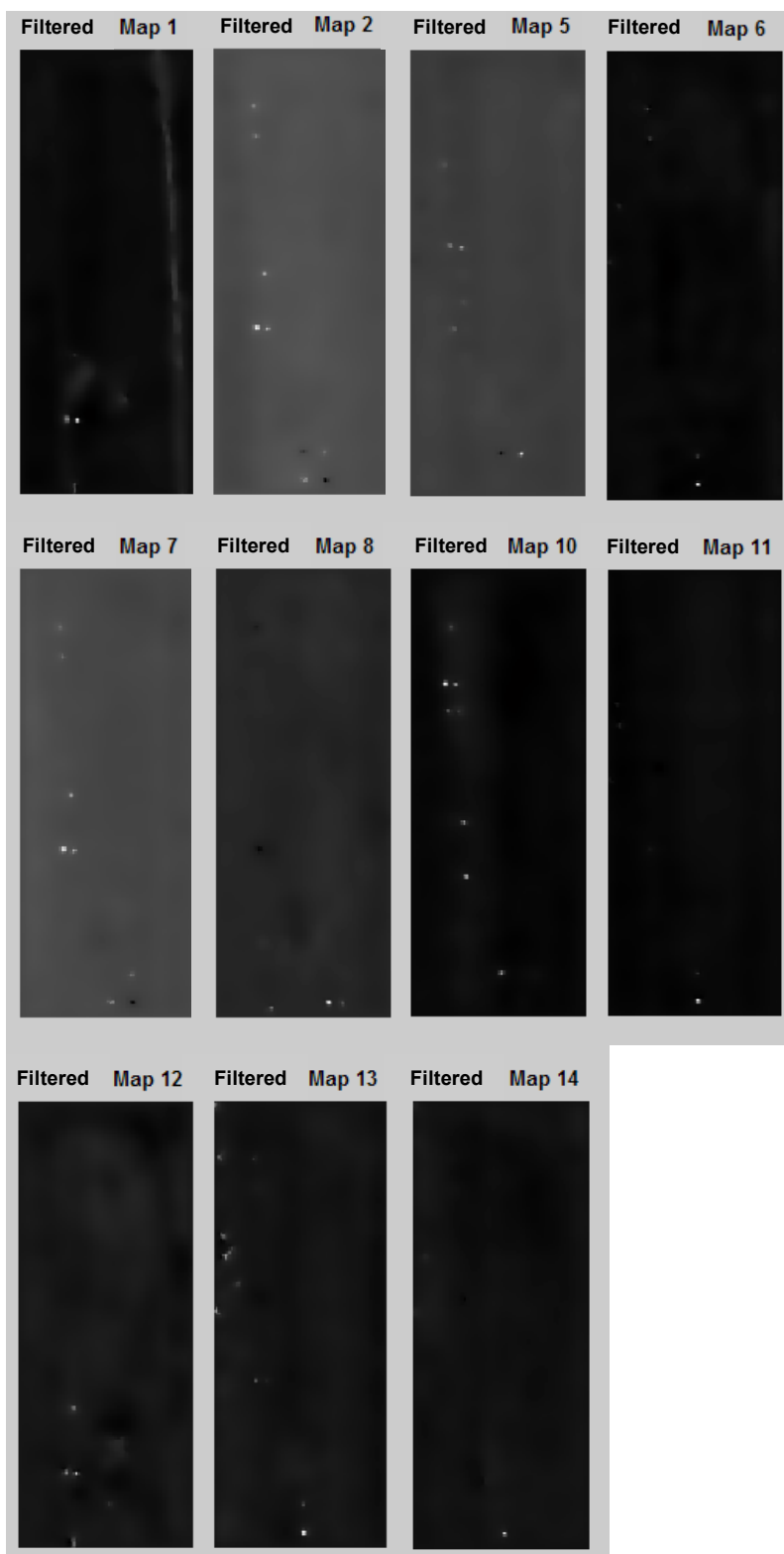
**Figure 4-13. ARES 4F: Target Abundance Maps after IAN Filtering**

As with ARES 3F, notice in the abundance maps in Figure 4-13 on the previous page and below in the signal plots in Figure 4-14 how after IAN filtering much of the background detail has been filtered out while target detail is still apparent. However, notice in map 13 the variability (power) of the tree outliers in the left side of the signal in Figure 4-14 was significant enough to not be smoother out with IAN filtering. One can also see the intense pixels remain in the corresponding abundance map in Figure 4-13.
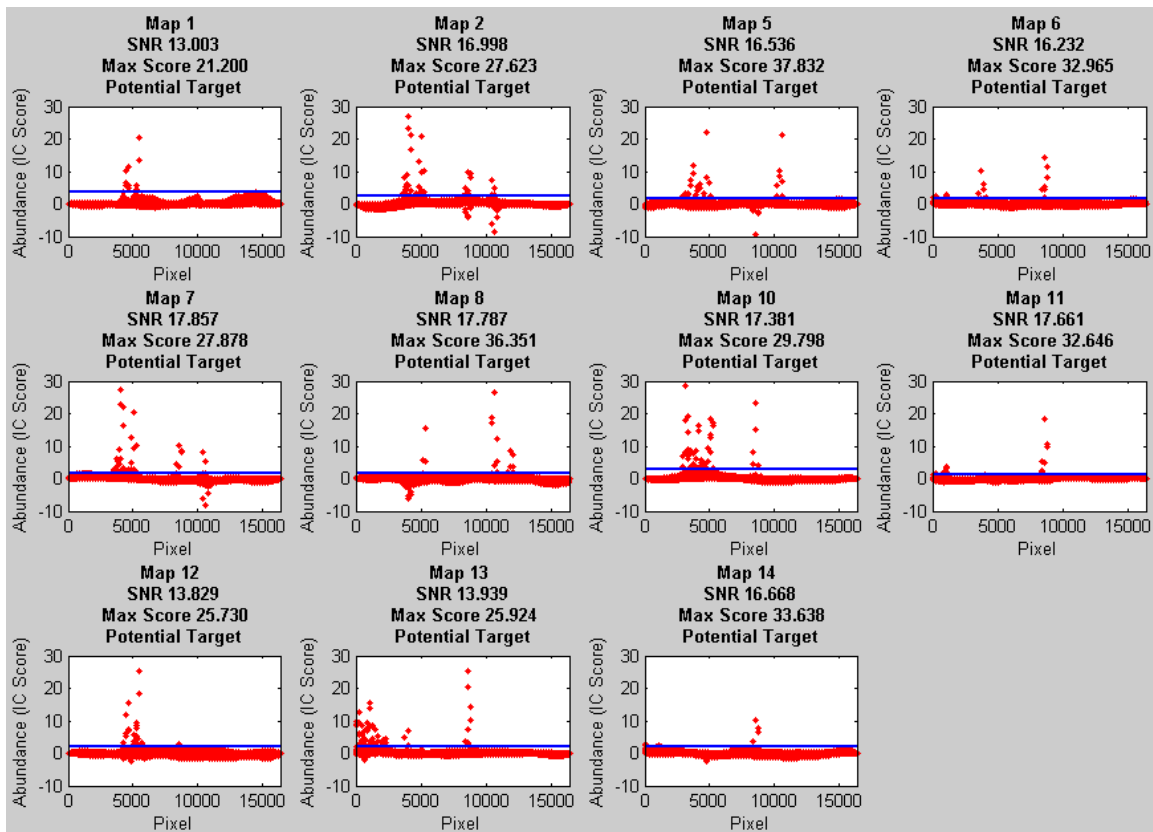

Figure 4-14. ARES 4F: Target Signals after IAN Filtering with Positive Signal Target Identification Thresholds
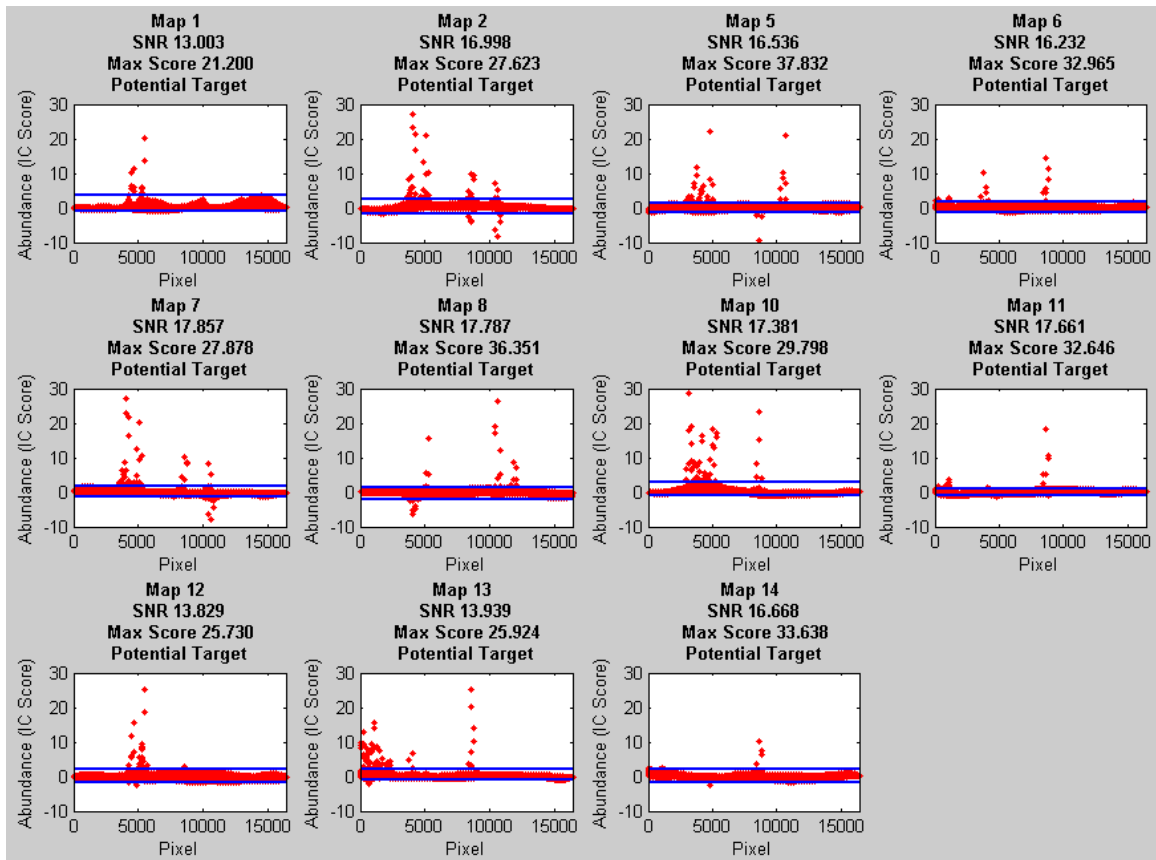
**Figure 4-15. ARES 4F: Target Signals after IAN Filtering with
Positive and Negative Signal Target Identification Thresholds**

As with ARES 3F, thresholding on both sides of the smoothed signal was accomplished to check for positive and/or ill effects.
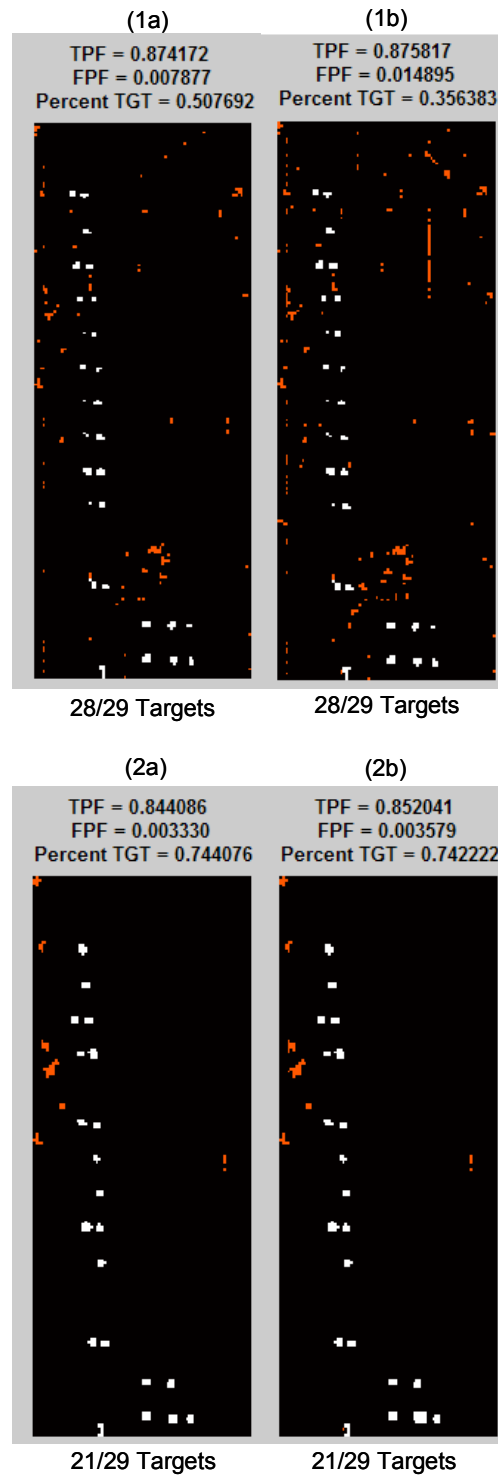
**Figure 4-16. ARES 4F: Target Identification, False Positives in Red
(1) without and (2) with IAN Filtering
(a) Positive Threshold (b) Positive and Negative Threshold**

As with ARES 3F in Figure 4-16, IAN filtering reduces FPF and increases percent TGT when comparing (1) images to (2) images. TPF decreased due to some panel targets, as before, being eliminated during smoothing. When comparing (2a) images to (2b) images, thresholding on both sides results in no difference in performance. However, with no IAN filtering, comparing (1a) to (1b), thresholding on both sides increases false positives, but does not increase the number of targets detected.

It should be noted that AutoGAD's result of locating 21/29 targets with a TPF of 0.85 and FPF of 0.003 (2a and 2b) is a significantly positive result in light of the many of the targets being hidden under the tree line. More on the significance of this result will be discussed in chapter 5.

Results for ARES 1C
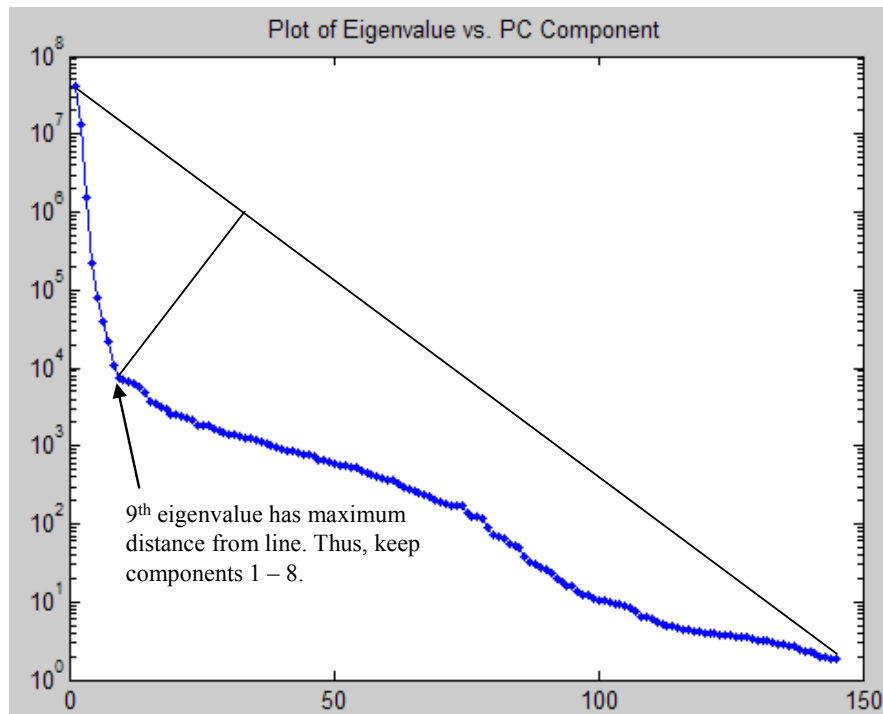
Figure 4-17. ARES 1C: Dimensionality Decision via MDSL

In Figure 4-17, the MDSL method presents another successful location of the knee in the eigenvalue curve determining 9 endmembers.
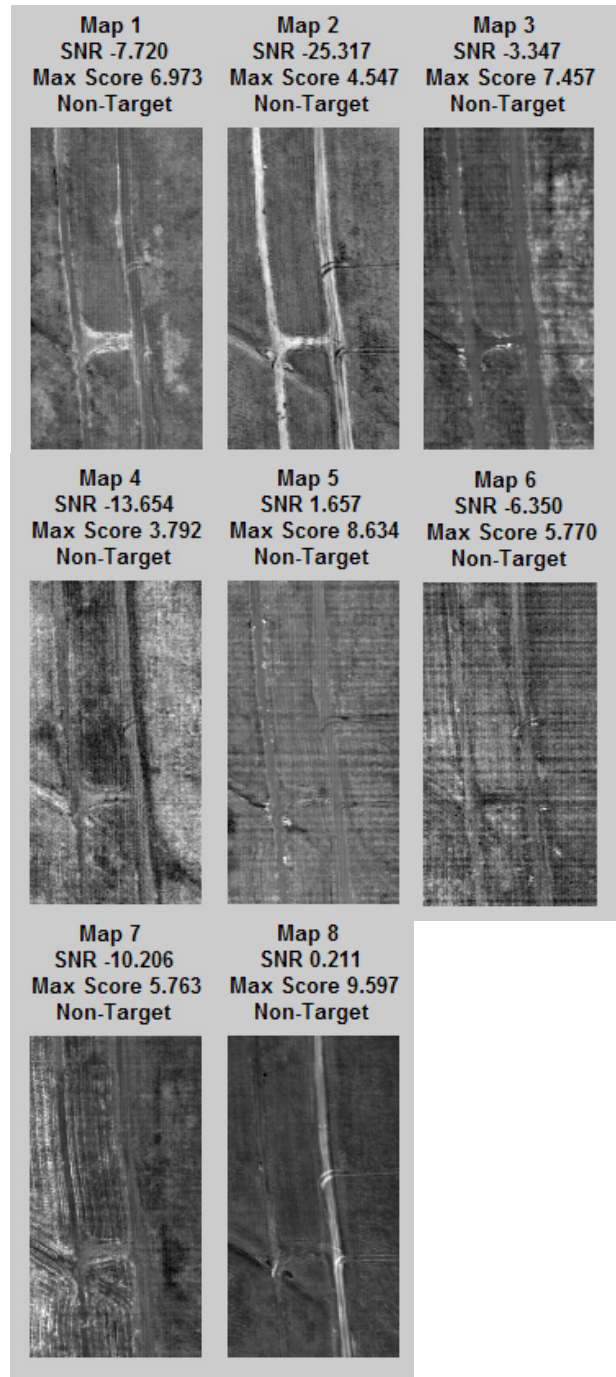
Figure 4-18. ARES 1C: Abundance Maps from 8 ICs via MDSL Decision
No maps are above PT SNR and max pixel score thresholds

Figure 4-18 and 4-18 illustrate a significant positive result of AutoGAD's ability to recognize an image that has no targets based on the PT SNR and max pixels score filters. Recall, by using the KV filter, having set the threshold for target detection to the lowest KV value for an image with targets, as explained in Figure 3-19, the road would have been deemed a target. However, based on the feature space characterized by the two new filters, the road feature falls into the non-target class. Thus, during feature selection AutoGAD eliminates all maps as potential targets.
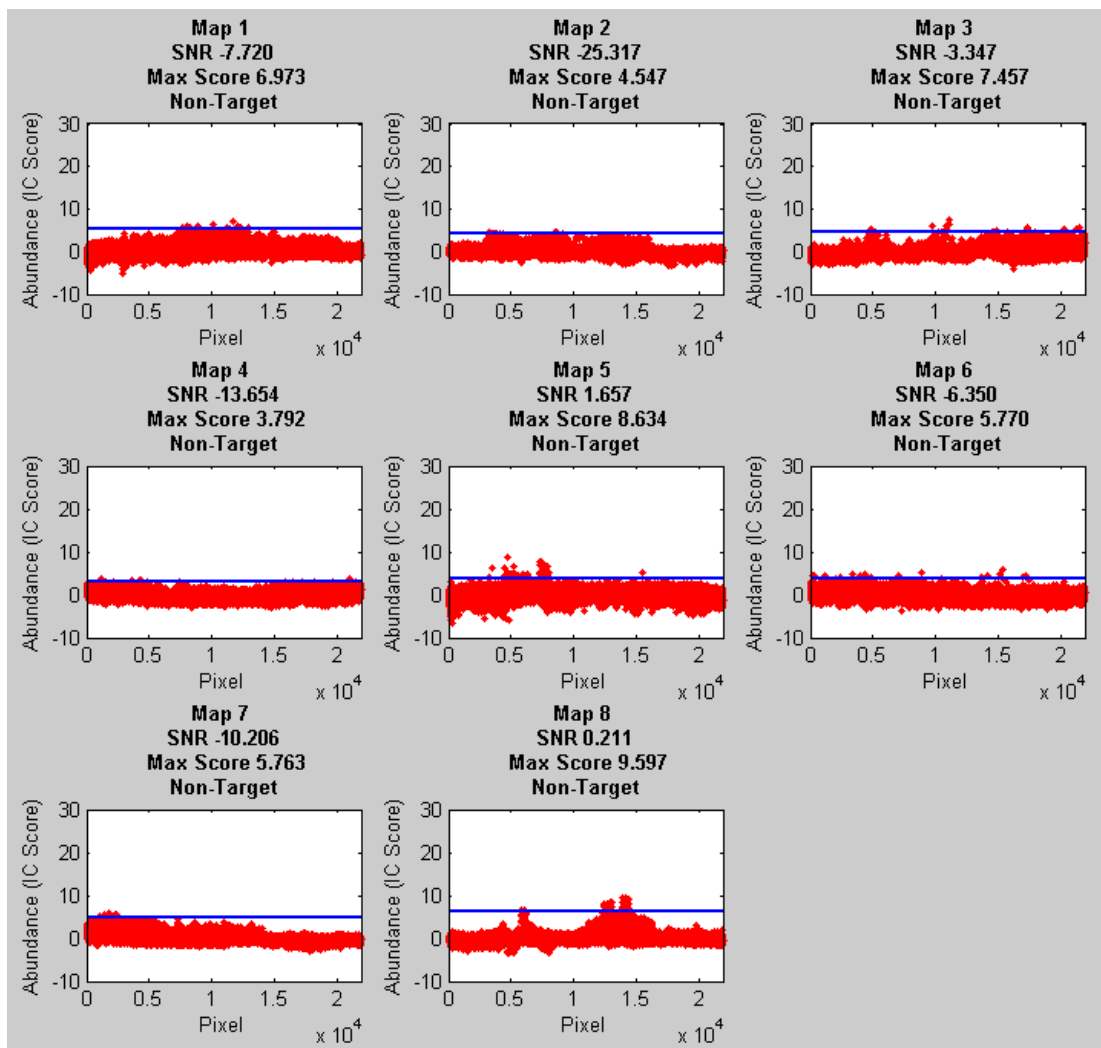
Figure 4-19. ARES 1C: PT SNR and Max Pixel Score for Each Signal with Potential Target Threshold Lines
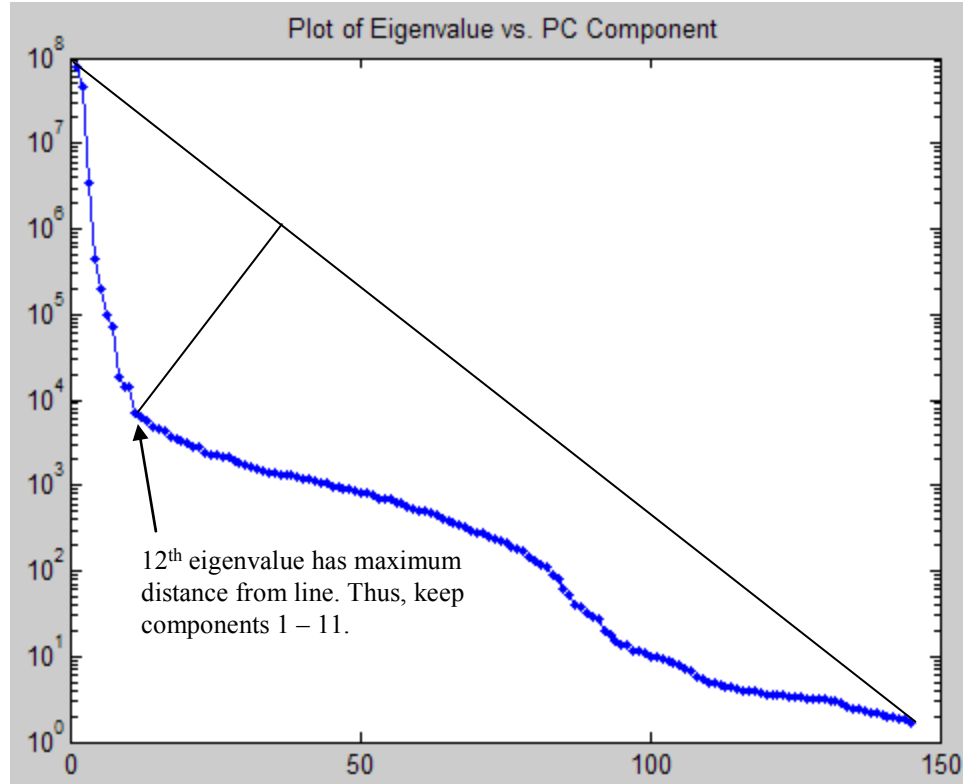
Figure 4-20. ARES 2C: Dimensionality Decision via MDSL

Having now applied the MDSL method to eight sample HSI images, it appears that if the shape of the eigenvalue curve of the covariance matrix of the spectral data conforms to the theoretical shape that occurs under the LMM (see Stocker citation) where noise eigenvalues lie on a tilted ramp and eigenvalues representing the distinct spectral signals in the image lie above the knee, then the MDSL method locates this point effectively. As with any new method, the success of the MDSL method should be compared to other means of approximating the breakpoint between noise and signal eigenvalues over many more HSI images. Based on this subset of images, this method is fast and effective and thus, indicates promise.
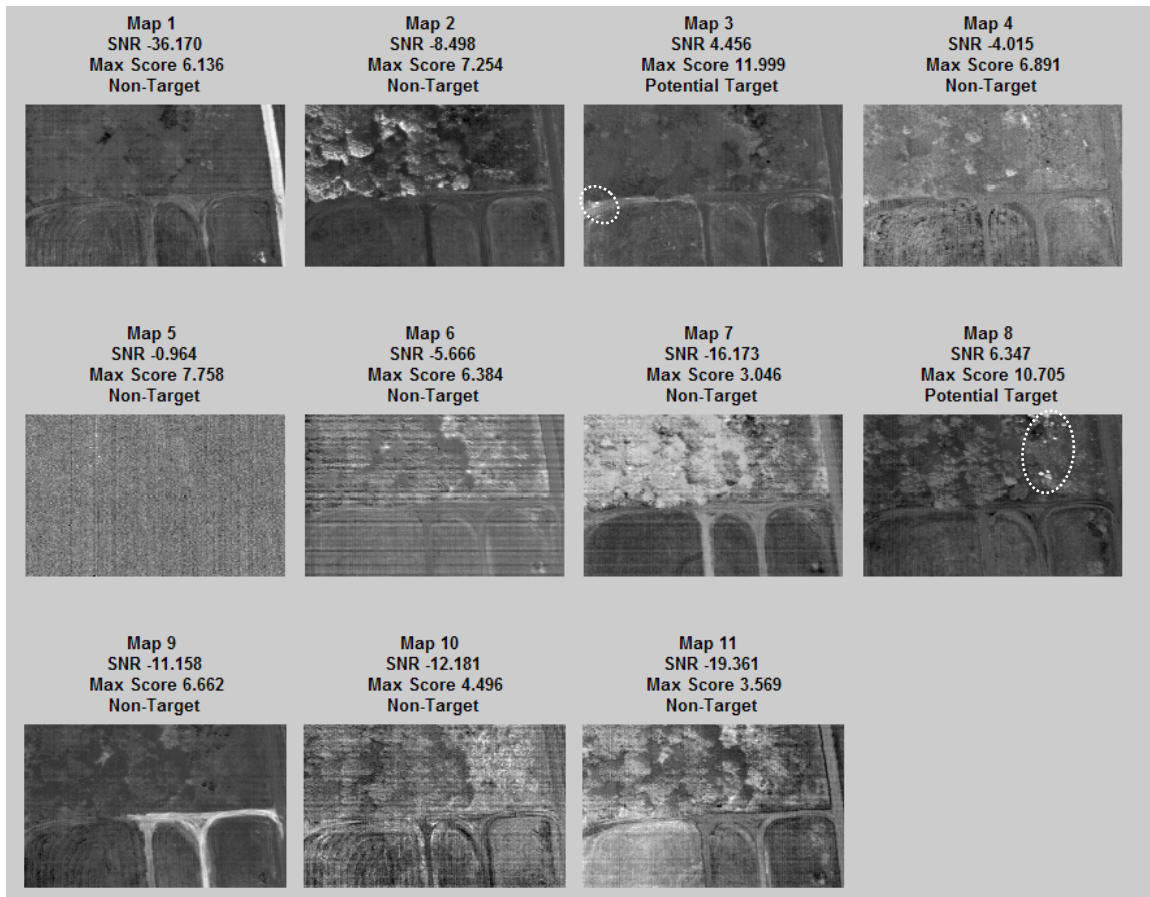
Figure 4-21. ARES 2C: Abundance Maps from 11 ICs via MDSL Decision
Maps 3 and 8 are above PT SNR and max pixel score thresholds and thus, are falsely declared target features. Circled in these maps are the anomalies causing the false selection.

Unfortunately, AutoGAD selects two non-target anomalies as potential targets based in PT SNR and max pixels score. Map 8 highlights small, rare, and spectrally unique bushes, the same definition for targets in the eyes of an anomaly detector. Map 3 highlights what appears to be some disturbed dust/earth on the dirt road. Although, not a target, such a detection could indicate recent road use. Regardless, this result highlights the need for fusion with a signature matching algorithm to eliminate these false positives as non-man-made objects.
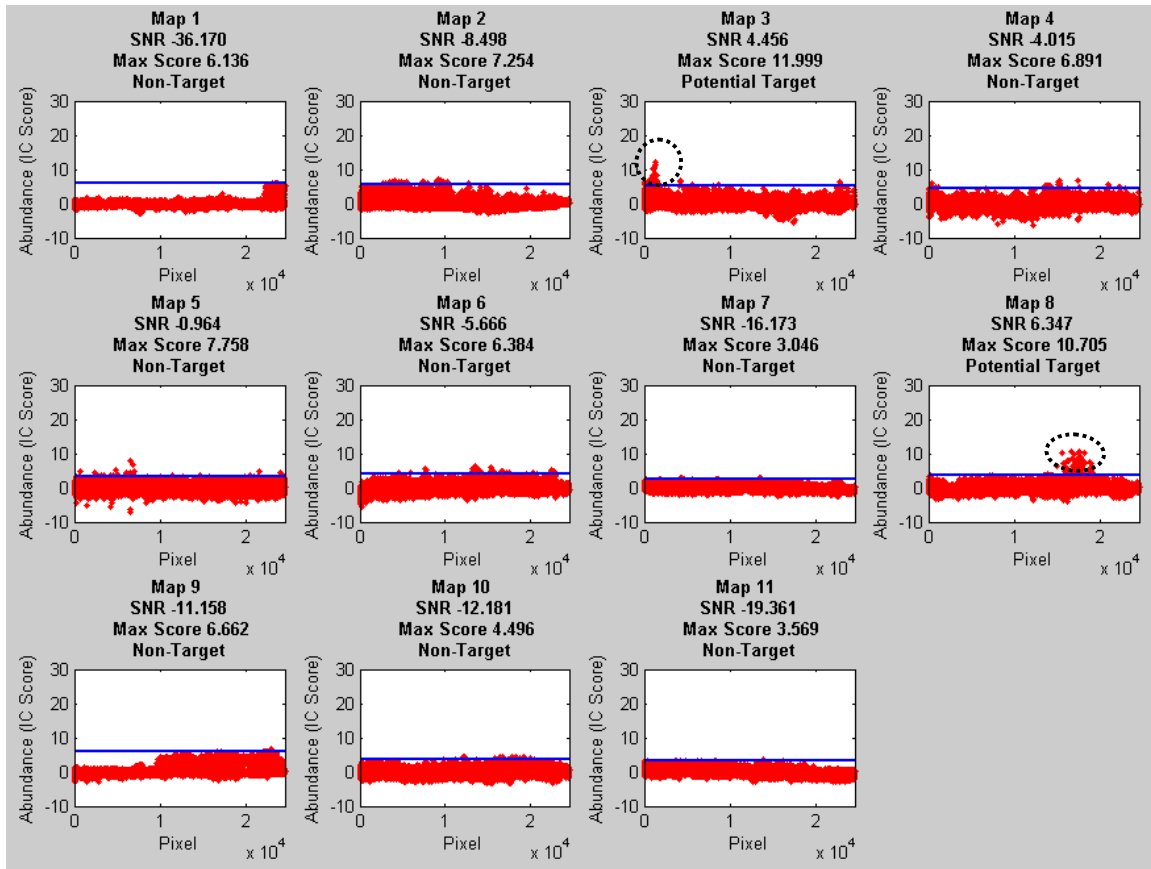
Figure 4-22. ARES 2C: PT SNR and Max Pixel Score for Each Signal with Potential Target Threshold Lines

Circled in these signals are the anomalies causing the above threshold PT SNRs and max pixel scores.
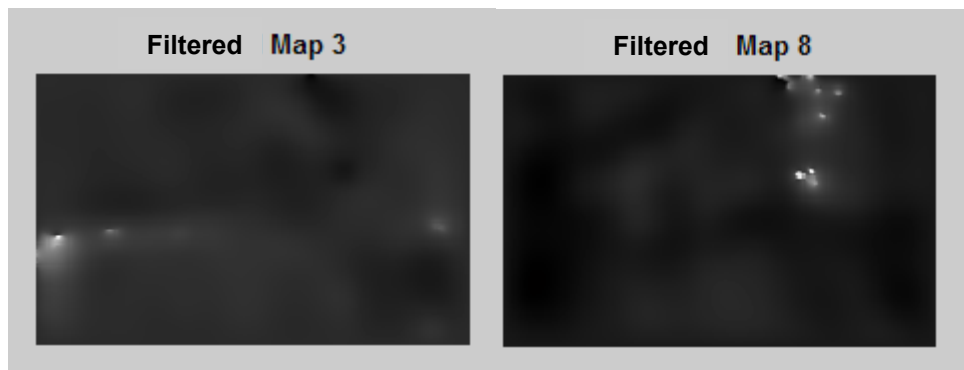
Figure 4-23. ARES 2C: False Target Abundance Maps after IAN Filtering

After IAN filtering the non-target signal power is strong enough to remain and be detected during AutoGAD's target identification phase.
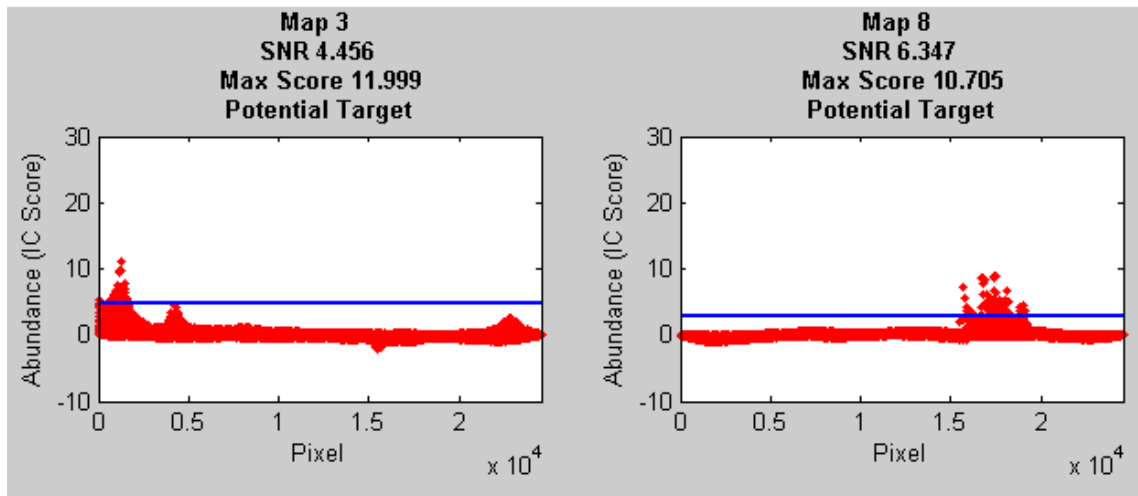
Figure 4-24. ARES 2C: False Target Signals after IAN Filtering with Positive Signal Target Identification Thresholds
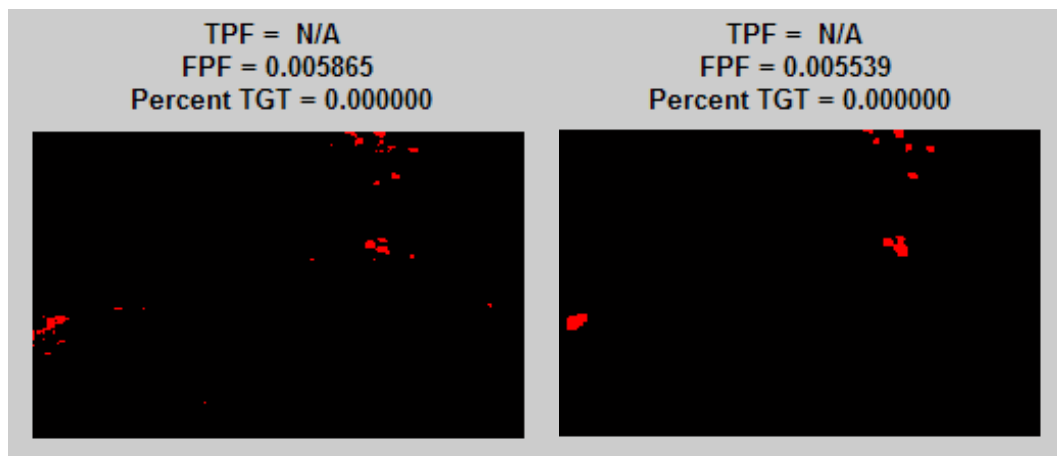
Figure 4-25. ARES 2C: Target Identification, False Positives in Red without (left) and with (right) IAN Filtering

4.4 Critical Analysis

This section will provide some more in depth analysis of AutoGAD's performance. First this research will present a comparison of the feature space of the validation images to the feature space of the test images to check for similarities. Second, as with the test images, 100 runs were completed for each validation image to test the run-to-run variability of AutoGAD using FastICA across the measures of performance. Finally, sensitivity analysis will be conducted on the zero-detection histogram bin width decision used during the target feature selection phase to calculate PT SNR and target pixel identification phase to determine target pixels.

4.4.1 Feature Space Comparison

Figure 4-26 shows the similarity of the feature spaces for the test and validation images. In the validation image feature space, no target classes fell in the uncertainty region, but two non-target classes, labeled in the figure, lie in the region. More images in multiple rural environments need to be tested using the PT SNR and max pixel score filters to further characterize the overlap between the classes.
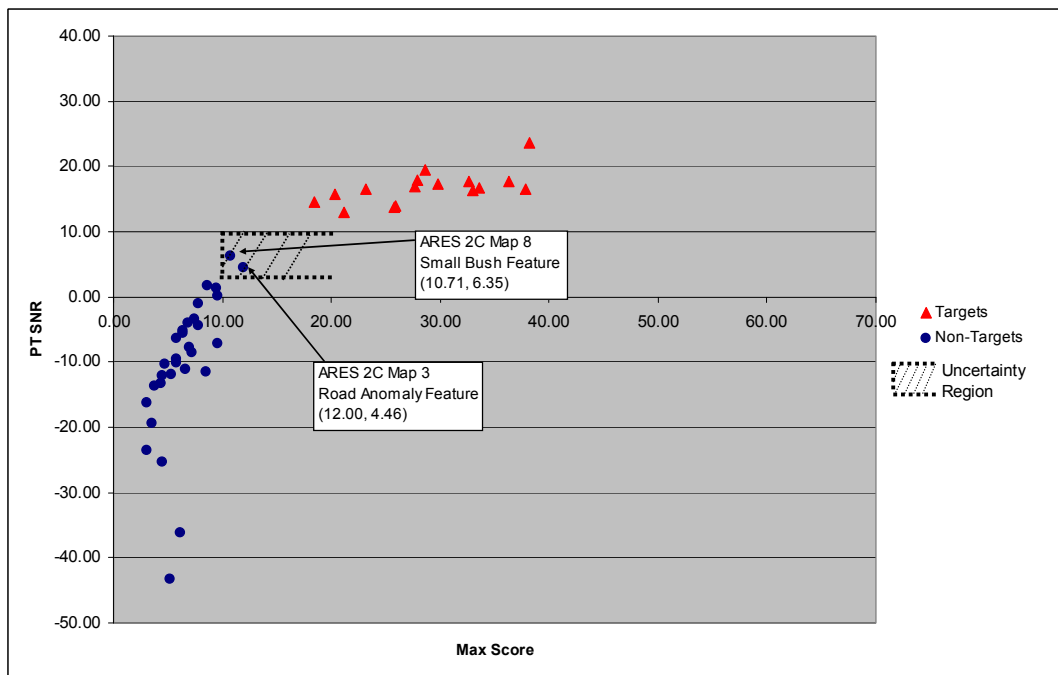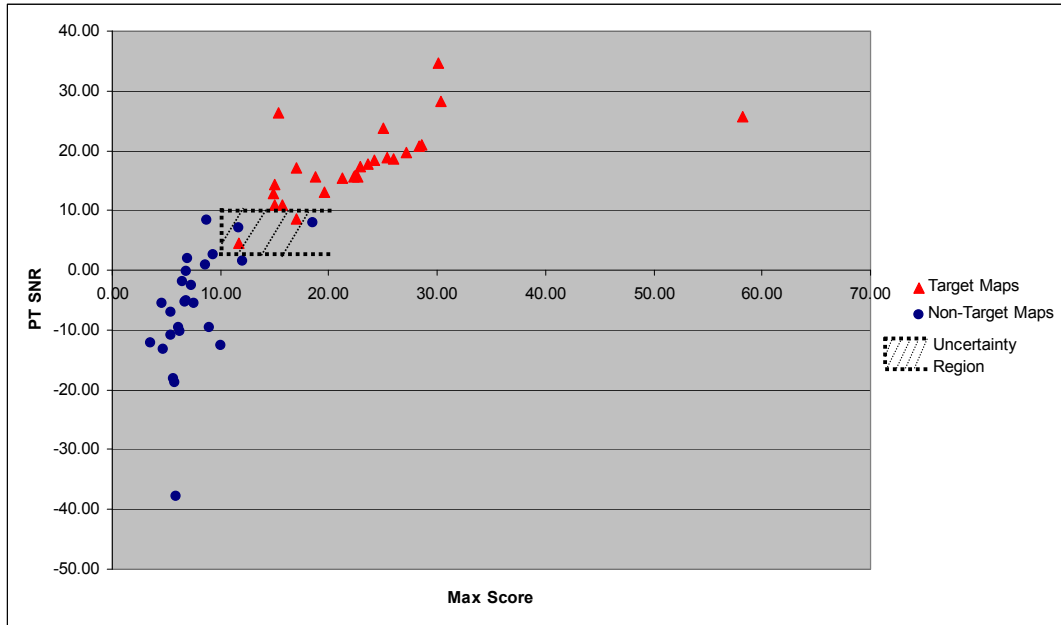
Figure 4-26. Max Score and PT SNR Feature Space for Test Images (top) and Validation Images (bottom)

4.4.2 Variability Analysis

As with the test images, note the low run-to-run variability across the measures of performance. Further the number of targets detected remained consistent for each run. It should be noted that the source of variability is from FastICA, the algorithm that has a random initial starting point. This is the reason for choosing the 'pow3' setting in FastICA since it was observed to have the minimum amount of run-to-run variance compared to 'tanh'. The other features of AutoGAD are deterministic. Hence if one fixes the an initial staring point for FastICA by using the same position in the random number stream, AutoGAD will return the exact same results for the image regardless of the number of runs.

Table 4-1. AutoGAD Results for Validation Images (100 Reps)

	Positive Outlier Thesholding								
	Mean TPF	Var TPF	Mean FPF	Var FPF	Mean Percent TGT	Var Percent TGT	Mean Time (sec)	Var Time	No. TGTs Detected
ARES 3F	0.85	5.53E-05	0.0004	5.66E-10	0.92	3.15E-06	4.99	1.10	15 / 20
ARES 4F	0.84	8.99E-06	0.0039	1.04E-06	0.71	0.0029	8.26	9.35	21 / 29
ARES 1C	N/A	N/A	0	0	N/A	N/A	1.44	0.17	N/A
ARES 2C	N/A	N/A	0.0051	5.15E-07	N/A	N/A	3.28	2.97	N/A

	Min TPF	Max TPF	Min FPF	Max FPF	Min Percent TGT	Max Percent TGT
ARES 3F	0.84	0.86	0.0004	0.0005	0.92	0.93
ARES 4F	0.83	0.84	0.0028	0.0054	0.64	0.78
ARES 1C	N/A	N/A	0	0	N/A	N/A
ARES 2C	N/A	N/A	0.0044	0.0083	N/A	N/A

	Positive and Negative Oultlier Thresholding								
	Mean TPF	Var TPF	Mean FPF	Var FPF	Mean Percent TGT	Var Percent TGT	Mean Time (sec)	Var Time	No. TGTs Detected
ARES 3F	0.92	3.86E-06	0.0029	1.46E-07	0.66	0.0010	5.00	0.87	17 / 20
ARES 4F	0.85	6.59E-06	0.0042	1.35E-06	0.71	0.0033	9.02	6.79	21 / 29

	Min TPF	Max TPF	Min FPF	Max FPF	Min Percent TGT	Max Percent TGT
ARES 3F	0.92	0.93	0.0020	0.0032	0.62	0.73
ARES 4F	0.84	0.85	0.0030	0.0061	0.63	0.77

Of important note are the low computation times for the images. Mean times are all under 10 seconds. Again, as with the test images, image files sizes are presented in Table 4-2. Reducing the image size via PCA using the MDSL decision significantly lowers the file size for each of the images enabling fast computation times for FastICA and AutoGAD's target feature selection, IAN filtering, and then identification.

Table 4-2. Validation Image Statistics

	ARES 3F
	Total No. of Elements
Original Image	30,736 pixels x 210 bands = 6,454,560
After Removal of Absortion Bands	30,736 pixels x 145 bands = 4,456,720
After PCA (MDSL Decision)	30,736 pixels x 13 bands = 399,568

No. Target Pixels = 459

	ARES 4F
	Total No. of Elements
Original Image	16,400 pixels x 210 bands = 3,444,000
After Removal of Absortion Bands	16,400 pixels x 145 bands = 2,378,000
After PCA (MDSL Decision)	16,400 pixels x 14 bands = 229,600

No. Target Pixels = 448

	ARES 1C
	Total No. of Elements
Original Image	21,924 pixels x 210 bands = 4,604,040
After Removal of Absortion Bands	21,924 pixels x 145 bands = 3,178,980
After PCA (MDSL Decision)	21,924 pixels x 8 bands = 175,392

No. Target Pixels = 0

	ARES 2C
	Total No. of Elements
Original Image	24,552 pixels x 210 bands = 5,155,920
After Removal of Absortion Bands	24,552 pixels x 145 bands = 3,560,040
After PCA (MDSL Decision)	24,552 pixels x 8 bands = 196,416

No. Target Pixels = 0

4.4.3 Sensitivity Analysis

As stated previously, a key decision is choice of bin width when using the zero-detection histogram method to determine the breakpoint between background and potential target pixels to calculate PT SNR during feature selection and when using the method to determine target pixels during the identification phase. The choice of bin width need not be the same for both phases, although AutoGAD uses a default choice of 0.05 for both phases.

The PT SNR and max pixel score feature space was indeed based on a bin width choice of 0.05. One may ask how the PT SNR values for the target and non-target classes change for different choices of bin width. As the choice of bin width is increasingly different from 0.05, then the PT SNR threshold of 2 dB may no longer be an effective one and thus the uncertainty region characterized in Figure 4-26 may become an incorrect characterization of the overlap between the classes. Fortunately, max pixel scores do not change based on choice of bin width during feature selection. The max pixel score is just based on the IC signals produced from ICA. No secondary calculation on the signal is required as is the case for PT SNR. Thus, the second criteria of the abundance vector having to have a max score of at least 10 to be classified as a potential target class acts as a good balance to the variability in PT SNR values that may change with choice of bin width. Ultimately, using these two filters, one would hope that over a decision space for bin widths during target feature selection, AutoGAD performance will be consistent, i.e. the same target features will be selected for perturbations about 0.05. Also, over the decision space for bin widths during target pixel identification on the selected target features, one hopes AutoGAD performance to be relatively consistent, i.e.

the variance of TPF, FPF, and percent TGT is low. For the target feature selection phase, rather than analyzing the change in PT SNR for each map and for each image across the bin width decision space, the bottom line is (as in the target pixel identification phase) change in TPF, FPF, and percent TGT. Further, total performance for each phase can be captured with just TPF and percent TGT. Perfect performance would be:

(a) Of all the target pixels present, all were detected (TPF = 1)

(b) Of all the pixels detected, all were target pixels (percent TGT = 1)

Thus, the question is, over the decision space for each phase (target feature selection and target pixel identification), what is the variability of TPF and percent TGT? For ease of variance analysis, this research will project the two-dimensional response vector (percent TGT, TPF) to one dimension and analyze the variability in this one dimensional response. One can have a perfect TPF, TPF = 1, but to do so may sacrifice percent TGT. In other words, all the target pixels could be detected but the threshold was set so low that several non-target pixels were detected as well the making percentage of pixels detected that are targets low and vice versa. Thus, one can consider TPF and percent TGT as competing objective functions. One way to project competing objective functions to R^1 in multicriteria optimization is to use the vector's distance from the ideal point as the new response. In this case, the ideal point is (1,1). This is one of several techniques in the field of multicriteria optimization to order a set of vector valued responses. For more information the reader can consult the Ehrgott citation in the bibliography. An implicit assumption in this projection is that each objective function response, TPF and percent TGT, is of equal importance or weight to the user. This may not necessarily be the case for this application, but for the purposes of analyzing the

variability of the response in one dimension, this research will make the assumption of equal weighting.

Variability of the TPF, Percent TGT Vector's Distance from Ideal Point

Figure 4-27 illustrates AutoGAD's response for ARES 3F in the two dimensional space (top) versus the one dimensional space (bottom) when varying the histogram bin width for target pixel identification from 0.01 to 0.1 incrementing by 0.001, thus creating 91 data points. In the top chart one can see that starting from left to right TPF stays roughly the same, but percent TGT increases. Then one reaches a point where percent TGT stays roughly the same, but TPF decreases. The region between these two points is considered the trade-off space between the two objective function values in the field of multicriteria optimization.

The scale for the bottom chart in Figure 4-27 on the vertical axis is from 0 (best case) to $\sqrt{2}$ (worst case) distance from the ideal point. Due to some coordinate pairs being nearly identical in the 2-D space, it is not possible to visualize all 91 data points in the top chart. However, in the bottom chart all 91 data points are visible and give insight as to the variability about the default bin width decision of 0.05. After projection to the 1-D space that represents each coordinate pair's Euclidean distance from the ideal point, one can see in a neighborhood about the default decision of 0.05 little variability in the response. Specifically, from a bin width of 0.038 to 0.069, except for one point at 0.043, the response is nearly horizontal indicating little variability in the response on this range.
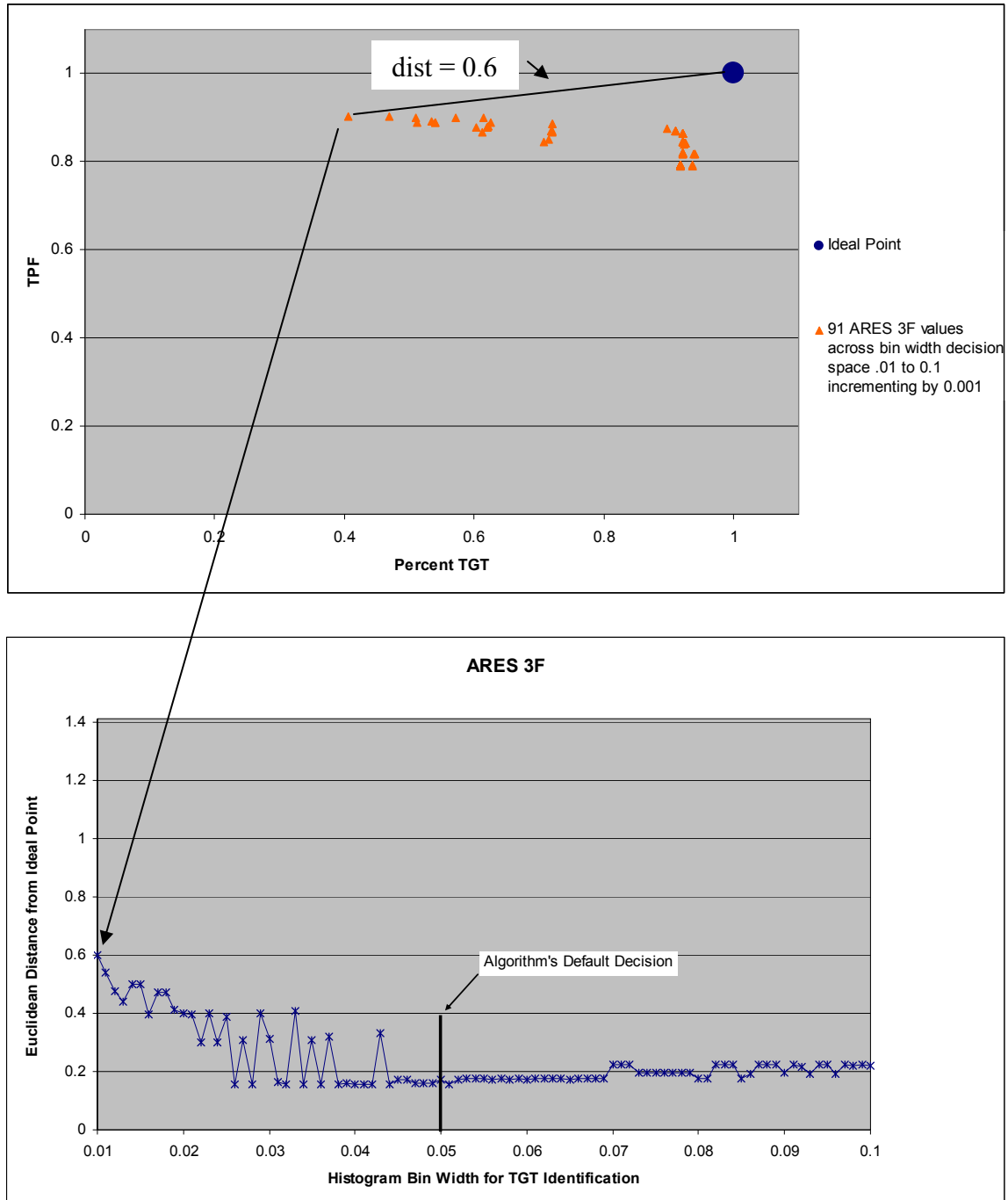
Figure 4-27. Comparison of 2-D Response (top) to 1-D Response (bottom)

On this range the mean percent TGT is 0.92 with a variance of 0.001 and mean TPF is

0.85 with a variance of 0.0001. Considering the one dimensional response, the mean

Euclidean distance from the ideal point on this range is 0.17 with a variance of 0.0009.

Further, this region is also has the lowest distance from the ideal point over the decision

space range. This low region in the bottom chart corresponds to the set of coordinate

pairs in R^2 with the shortest distance from the ideal point. Ideally, for each image one

would want the range of low variability about and near 0.05 to be substantial relative to

the total range tested and the values to be low signifying short distance from the ideal

point.

The next section will show visually AutoGAD's 1-D response for each image

(test and validation) during the target feature selection phase as the bin width is varied

from 0.01 to 0.1 by 0.001 increments. Then for the following section, the same will be

presented for the target pixel identification phase. For each section, while varying the bin

width for that particular phase, the bin width for the other phase will be held at the default

value of 0.05. Note, so that the minor run-to-run variability in ICA does not influence the

variability analysis of bin width choice for each phase in the detection process, the initial

random matrix input to ICA was fixed to a particular state in MATLAB. In MATLAB

the initial state (or position in the random number stream) can be held constant.
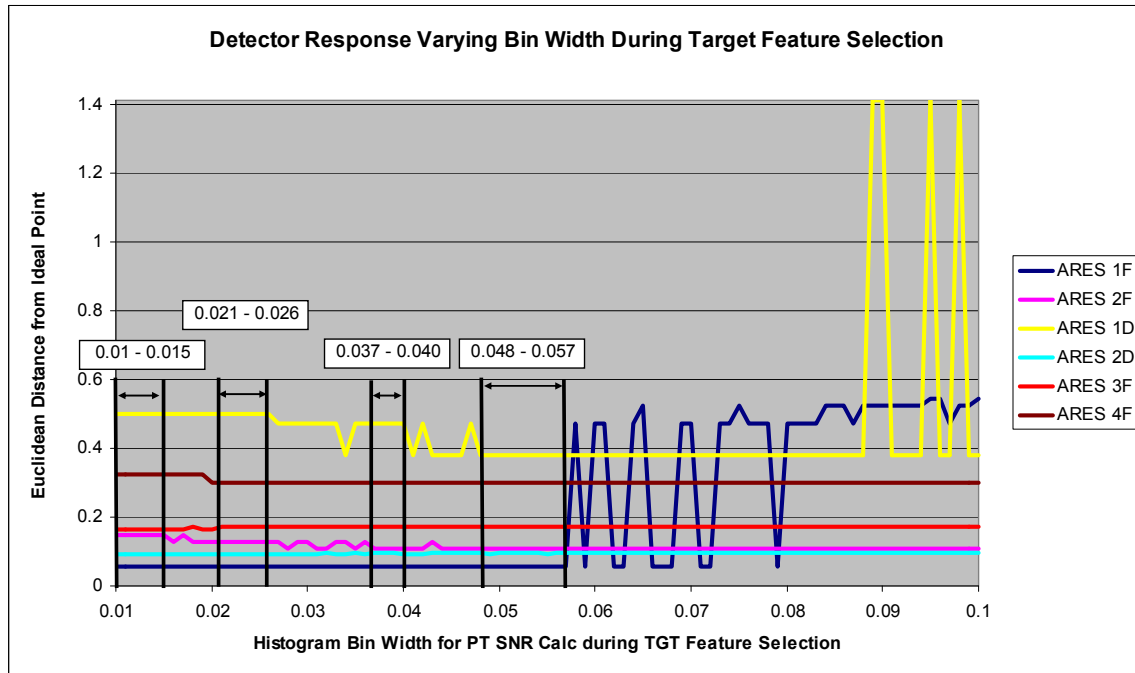
Figure 4-28. AutoGAD's 1-D Response during Feature Selection across all Images with Targets

The 4 largest regions with zero variability across all images over the decision space with the lowest Euclidean distance across all images are labeled.

Notice that except for ARES 1D and ARES 1F, the response for the images is constant over much of the decision space range. Even for ARES 1D and ARES 1F the response is constant over a substantial portion of the range. Labeled in Figure 4-28 are four regions of zero variance across all six images. This is by no means all the regions of zero variance, just the substantial regions on the left side of the decision space where the distance from the ideal point is the lowest across all images. The largest region across all six images with zero variability and shortest distance from the ideal point is from 0.048 – 0.057 bin width. This makes sense since the feature space was characterized (i.e. the PT SNR threshold was decided) using a bin width of 0.05. However, substantial regions of

algorithm stability other than at the point where the detector was designed yields some insight as to the algorithm's robustness across this bin width decision space.

On a side note, this way of viewing a detector's 1-D response (Euclidean distance from the ideal point) over some decision space for the detector is useful in comparing performance of a detector from one image to the next, much like a ROC curve. The image that has the lowest area under this new curve in Figure 4-28 can be considered the image across the sample images where the detector performs the best. Further when comparing detectors, instead of ranking the detectors by the one that yields the largest area under a ROC curve for a particular image, perhaps this method offers another way to compare detectors by ranking them according to those with the smallest area under the Euclidean distance from the ideal point curve.

Figure 4-29 details AutoGAD's response across the two images without targets. Notice the variance in the response is zero over the entire bin width range. Also, for these images, FPF was the response used since TPF and percent TGT have no values in images with no targets.

Detector Response Varying Bin Width During Target Feature Selection
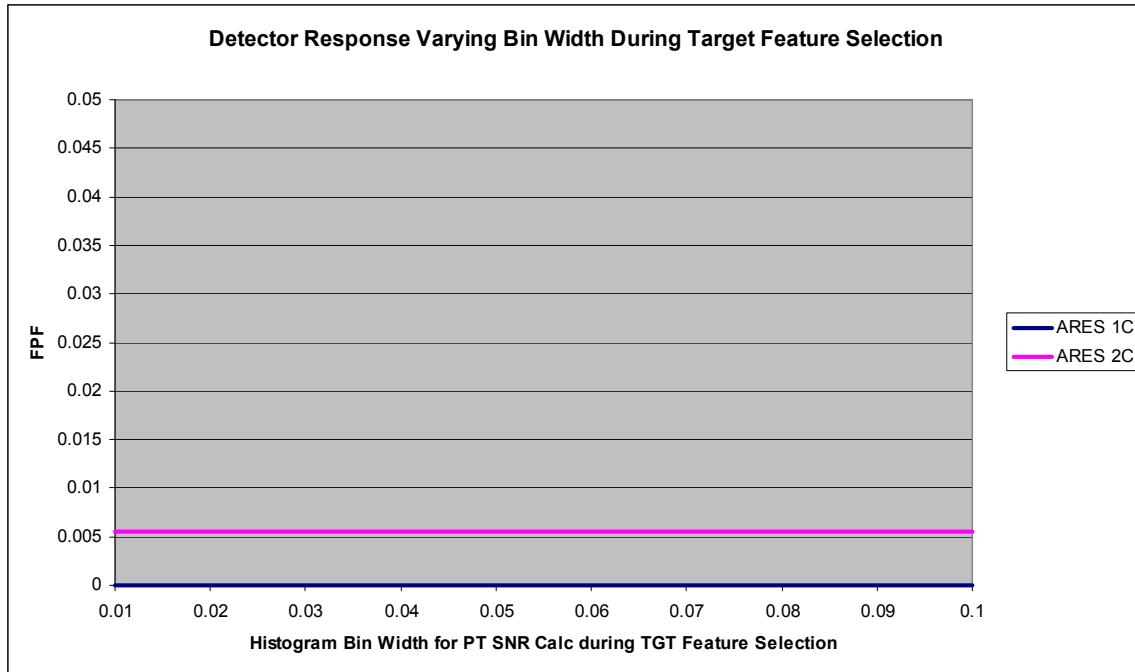
Figure 4-29. AutoGAD's 1-D Response across all Images without Targets

Summary Target Feature Selection Phase Variability Analysis

The reason the response lines are predominantly horizontal is that as one makes small changes to the bin width for the PT SNR calculation, the value will not change significantly because PT SNR calculation considers thousands of pixels. Plus as mentioned previously, the constancy of max pixel score over the decision space balances the variability in PT SNR. However, one can observe areas of significant jumps in Figure 4-28. These represent points where the change in PT SNR became significant enough to alter what features where selected as targets. These points are especially clear for ARES 1D and ARES 1F. As stated previously, areas of zero variance other than near the region the detector was designed is a positive result and suggests robustness.

Relationship between Bin Width and Threshold Location

So that the reader has an understanding on the effect that bin width choice and IAN filtering has on TPF and percent TGT during the target identification phase, Figure 4-30 is offered using ARES 4F as an example.

Although the response in percent TGT and TPF is not 'smooth' the predominant trend can be described as follows:

As bin width increases from 0.01 to 0.1, of all the pixels detected, the percentage that are targets increases. In other words, the number of false positives decreases. This is due to the threshold increasing on the IC signal as bin width increases. The is expected because the wider the bin width, the lower fidelity one has when creating the frequency distribution and thus the zero-point bin will occur further out in the tail. As a result the calculation of the threshold will be higher. Conversely, for a thinner bin width, the fidelity of the frequency distribution is higher and as such the zero-point bin will occur sooner in the tail. When comparing the top chart in Figure 4-30 to the bottom chart, notice the effect IAN filtering has on percent TGT. The curve is shifted upwards denoting an overall increase in this performance measure across the decision space. Thus, false positive detections are substantially decreased.

Not as pronounced, but the opposite trend is observed for TPF as bin width increases. Again due to the threshold increasing as bin width increases, the number of true target pixels detected out of the total number of true target pixels present will decrease. Notice the effect of IAN filtering has on TPF. By applying the smoothing a slight drop in TPF performance can be observed in Figure 4-30. Recall ARES 4F has small panel targets and as explained previously the smoothing process has been observed

to eliminate some of the smaller panel targets in the sample images, thus the slight

reduction in the TPF curve. Depending on the users priorities, increasing percent TGT

performance at the expense of small reduction in TPF may be acceptable especially for

cases where the cost of false positives is high.
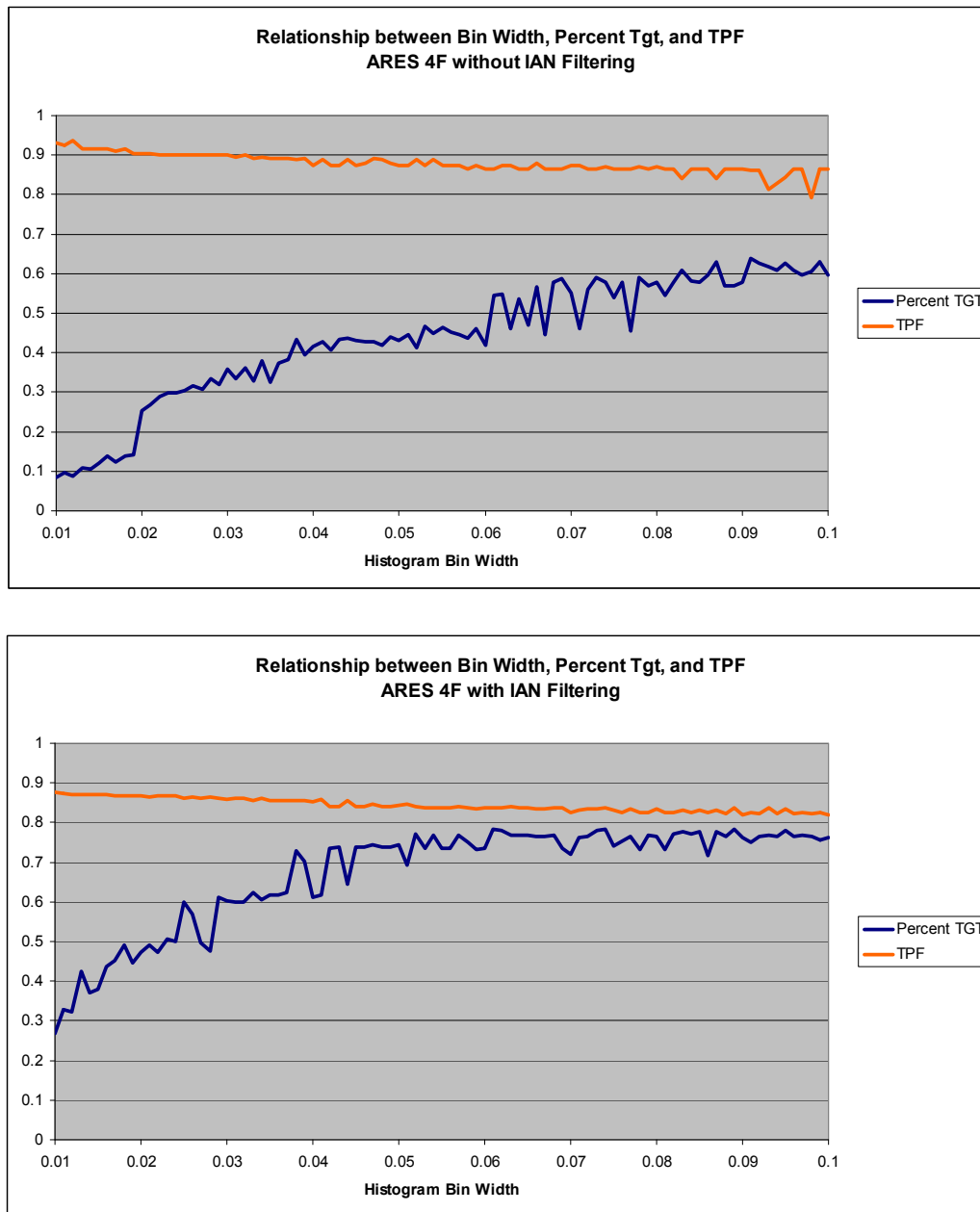
**Figure 4-30. AutoGAD's Response in Percent TGT and TPF
across Histogram Bin Width for Target Identification (ARES 4F)**

Now that the reader hopefully has a better understanding of the dynamics between bin width, percent TGT and TPF, sensitivity analysis for the target pixel identification phase will be presented in the 1-D response space.

Target Pixel Identification Phase Variability Analysis

Considering Figure 4-31, it is expected that AutoGAD's response will more variable in the target identification phase since we are now looking at the variability in the response as we change the threshold not for feature selection but for pixel identification. For feature selection the response will change only when as explained before the PT SNR calculation is significantly different enough to alter what features (i.e. abundance vectors) where selected as targets. Here, for each minor threshold change, TPF and percent TGT changes at the pixel level rather than the feature level. Thus, one can observe a much more noise (i.e. more variable) response. Again, the top chart details the response without IAN filtering and the bottom with IAN filtering.

First, consider the bottom chart with IAN filtering. The region with the lowest distance from the ideal point and the least amount of variability in the response across all six images occurs on the bin width range from 0.045 to 0.064. On this range, ARES 1F, 2D, 2F and 3F have the least variation compared to ARES 4F and ARES 1D as noticed by the near smooth horizontal line. Now, consider AutoGAD's performance without IAN filtering on the same range. Notice the considerable increase in volatility in the response for ARES 1F, 2D, and 2F. Further notice the trend of a decrease in distance from the ideal point when using IAN filtering. Table 4-3 details quantitatively what is observed visually in Figure 4-31 on the range from 0.045 to 0.064. Across all images except ARES 1D the variance in the response decreased after applying IAN filtering. Also,

4-40

except for ARES 2D the distance from the ideal point decreased after applying IAN filtering.
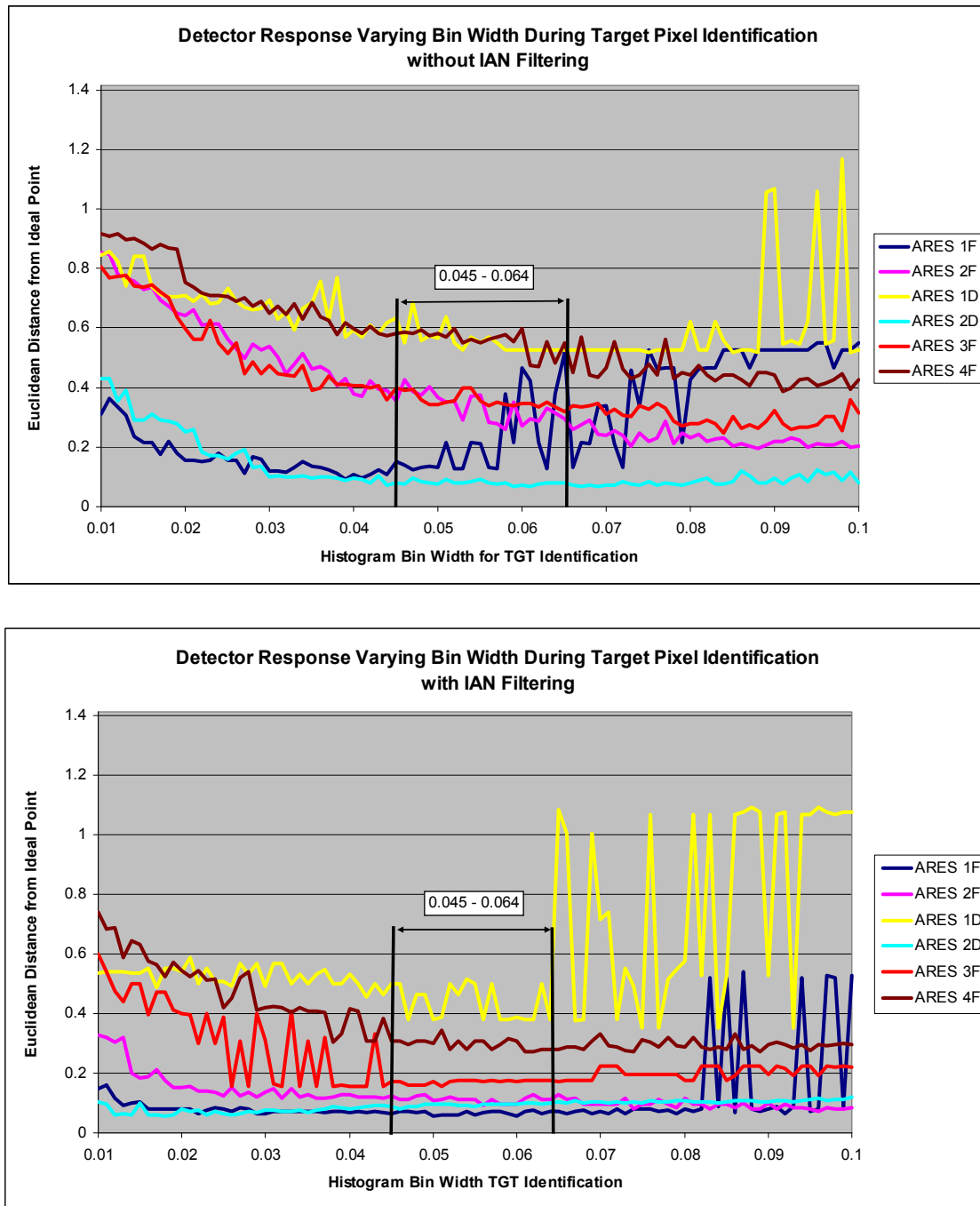
Figure 4-31. AutoGAD's 1-D Response during Identification across all Images with Targets

Table 4-3. Comparison of Mean and Variance of Euclidean Distance from Ideal Point without and with IAN Filtering on the Range from 0.045-0.064

	without IAN		with IAN		Difference	
	Mean	Var	Mean	Var	Mean Diff	Var Diff
ARES 1F	0.21	0.012426	0.07	0.00004	-0.14	-0.01239
ARES 2F	0.34	0.002312	0.11	0.00009	-0.22	-0.00222
ARES 1D	0.56	0.001893	0.44	0.00333	-0.12	0.00143
ARES 2D	0.08	0.000048	0.09	0.00003	0.01	-0.00002
ARES 3F	0.36	0.000544	0.17	0.00004	-0.19	-0.00050
ARES 4F	0.56	0.001487	0.30	0.00033	-0.26	-0.00116

Summary Target Pixel Identification Phase Variability Analysis

Across these sample images, the range of bin widths with the least amount of response variability and lowest distance from the ideal point is a bin width for identification of 0.045 – 0.064 when using IAN filtering. Thus, AutoGAD's default choice of 0.05 for target pixel identification appears to be a robust one. Reiterating, AutoGAD's feature space was characterized based on a 0.05 bin width to calculate PT SNR. Thus, stability directly about 0.05 was expected for the target feature selection variability analysis.   There were areas of stability, other than the region directly around 0.05, which was a positive result. It should be noted that the target identification phase was not designed around a bin width of 0.05. It was merely chosen based on pilot runs from chapter 2 when analyzing the zero-detection histogram method as a current technique for target identification. Bin width choice during target identification has no influence on the feature space characterization since the target identification phase succeeds the target feature selection phase.   However, intuition that the 0.05 bin width choice was a good choice for target identification from the pilot runs in chapter 2 has been proven here to be the robust choice especially after IAN filtering based on the

sample image responses seen in the bottom chart of Figure 4-31. More images are

needed to further validate this choice of bin width during AutoGAD's target pixel

identification phase.

V. Discussion

5.1 Limitations

Note that this detector's development, testing, and validation occurred over 8 HSI images in rural environments (forest and desert) where the targets consisted of tanks, smaller vehicles, tents and/or tarps, and different colored panels. Non-target classes consisted of roads, trees, large rocks, sand, gravel, dirt, grass, various sized and colored bushes, etc... The sensor was the HYDICE sensor and 145 of the 210 bands available were utilized prior to dimensionality reduction. Thus, the PT SNR and max pixel score target thresholds were based on the feature space characterization of the separation between target and non-target classes in these environments, with this sensor, retaining those 145 bands. AutoGAD most likely will require calibration if used with different sensors in different environments. More testing should be conducted with images from different sensors from more environments with more non-target classes and target classes to further define the PT SNR and max pixel score feature space of targets and non-targets under a myriad of different operating conditions. It may be found that a desert calibration is different from a woodland calibration. Thus, depending on the environment different thresholds are used when employing AutoGAD operationally.

5.2 Contributions to the Field of HSI Target Detection

Based on this subset of HSI images, this research made the following contributions that were not found after a review of the current literature in the field:

1. Using the kurtosis approximation of negentropy (the 'pow3' setting) as the objective function, this author illustrated that FastICA yields the least amount of

variability compared to using the non-polynomial approximation of negentropy (the 'tanh' setting).

2. Based on the theory of the shape of the eigenvalue curve of the covariance matrix of spectral data under the LMM, a simple and as shown effective unsupervised dimensionality determination can be made via a new process provided by this author coined the MDSL method. This method finds the eigenvalue with maximum distance from the log-scale secant line to identify the 'knee' in the curve that approximately separates the noise eigenvalues from the signal eigenvalues.

3. The need to subjectively analyze a scree graph of kurtosis values of independent component signals has been eliminated by using new target feature filters suggested by this author, PT SNR and max pixel score. Target classes and non-target classes have less overlap in this new feature space when comparing the overlaps from Figures 3-17 and 3-24. .

4. The zero-detection histogram method suggested by Chiang, Chang, and Ginsberg was validated by this author as an effective technique at locating the breakpoint between background and potential targets for use in calculating PT SNR during target feature selection and for identifying outlier target pixels during target identification. Further, a bin width of 0.05 (based on the test and validation images) was demonstrated to be the robust choice when employing this method.

5. False positives during target identification were reduced using an adaptive noise filter, but with an iterative approach (suggested by this author and coined IAN filtering) using a small scanning window. Thus, target signal power is preserved, while background signal power is repeatedly reduced, in essence an effective signal noise smoothing technique.

6. Projecting a detector's vector response (TPF, percent TGT) to its Euclidean distance from the ideal point (1,1) (an idea borrowed from multicriteria optimization) is offered as a way to grade a detector's overall performance across a user defined decision space as an alternative to a ROC curve.

5.3 Future Research

In order to calibrate the AutoGAD algorithm to other HSI scenarios and with

other HSI sensors, after determining the atmospheric absorption bands for each sensor,

one could apply a robust parameter design to find the settings for the algorithm across a

range of hyperspectral images that minimize the distance from the ideal point of [1

(TPF), 1 (percent TGT)], while also minimizing the variability of this response. Specific

factors for consideration are detailed below. As a beginning search of the space, the

center points for the control factors could be AutoGAD's default settings detailed in

Figure 3-34.

Control factors:

(1) Histogram bin width for target feature selection

(2) PT SNR threshold

(3) Max pixel score threshold

(4) Histogram bin width for target identification

(5) Number of iterations of IAN filtering to complete on selected target

features prior to target identification

Noise factor:

Hyperspectral images

Response: Euclidean distance from ideal point [1 (TPF), 1 (percent TGT)]

Another improvement to this effort discussed in chapter 4, could be to modify the

AutoGAD algorithm to include more intelligent target feature selection by calculating the

PT SNR of the negative side of the independent component signal as well.

One challenging extension could be an investigation of methods other than ICA

that do conform to the non-negativity and sum-to-one constraints to solve for the

abundance matrix in the LMM. One such possible method is non-negative matrix

factorization (NMF). However, given the large file sizes inherent in HSI images, some

dimensionality reduction must be accomplished prior to employing NMF. Normal PCA

is not an option due to projecting the original data to a space that is not non-negative.

Thus, in addition to exploring NMF, a new technique, non-negative PCA should be explored.

A final recommendation is a fusion of AutoGAD with a signature matching algorithm. As discussed, natural objects can meet the assumptions of an anomaly detector and be falsely classified as targets. Thus, after preprocessing an rural HSI image with AutoGAD, nominated pixels' spectral signatures could be compared to a library of man-made targets of interest to further eliminate false positives.

5.4 Conclusion

Recall at the outset of chapter 4 the statement of the added difficulty ARES 4F would provide given targets hidden under the tree line. Despite the few false positive hits observed even after employing IAN filtering as denoted in Figure 4-16 (2a) and (2b), AutoGAD successfully detected hidden targets under the tree line with an impressive 0.85 TPF and 0.003 FPF. Regardless of missing some of the targets, an operator analyzing this intelligence product produced from AutoGAD, would be alerted to an obvious linear pattern of objects under the tree line. Such awareness may not have been known prior to the application of the detector, analyzing a transmitted RGB image only. This result is significant in light of the Air Force coined acronym TUT, Tanks Under Trees, referring to the ever present problem that targets hidden under foliage presents. Further, this application of AutoGAD took only 9 seconds. This speed of this detection provides promise for AutoGAD as a target detection algorithm that could be employed in UAVs with hyperspectral sensors so that the aircraft has the ability to process its imagery onboard in real time and relay potential coordinates and binary images like (2a) to

operators. The size of such a binary file presented in Figure 4-16 for ARES 4F is 16.4

kb. The alternative would be to transmit the entire HSI data matrix for the operator to

process at the ground control station. Capability provided by an algorithm such as

AutoGAD could give UAVs on board processing ability that would drive down the data

transfer rate requirement from the UAV to the control station, given the file, after

processing for this example is only 16.4 kb (the binary target location file). The

following quote from the OSD Unmanned Arial Vehicles Roadmap 2000-2025 gives

perspective on this issue.

> The most fundamental, technology-driven decision facing UAV
> planners early in the 2000-2025 timeframe is whether to migrate towards
> an air-centric (processor based) or a ground-centric (communications
> based) architecture. In the case of the former, relatively autonomous
> UAVs with minimal ground infrastructure and direct downlinks to users
> will be the norm. For the latter, UAVs will be remoted "dumb" sensors
> feeding a variety of sensory data into a centralized ground node which
> builds a detailed, integrated picture for the users. Hybrid architectures, in
> which processing is begun on the aircraft and completed on the ground
> and transmission requirements are reduced by using recorders and/or data
> compression techniques, are used by today's reconnaissance aircraft. This
> architecture exists because the capabilities of current processors and data
> links are inadequate by themselves to handle the amount of data generated
> by today's sensor suites. Data compression techniques are the most
> prevalent workaround for insufficient onboard processing speed and data
> link data rate constraints.
>
> **At some future point, sufficient onboard processing power for
> the worst case information processing requirement, such as streaming
> video of ultra spectral imaging (thousands of spectral bands), will be
> reached. At that point the answer, vice the data that provided it, will
> become the driver for the data link's capacity, downsizing its
> requirement drastically. As an illustration, a future UAV system
> searching for "tanks under trees" (TUT) with a hyper-spectral
> imaging sensor would process and exploit its imagery onboard in real
> time, then relay the coordinates and certainty of identification of all
> tank suspects found over a 9.6 kbps link, simultaneously with the
> UAVs health and status. This becomes an air-centric (processor
> driven) architecture, in which UAVs become highly autonomous
> extensions of man, drawing their own conclusions onboard and**

distributing their answers directly to users (Office of the Secretary of Defense, 2001:55).

The last portion of the quotation highlighted by this author emphasizes the point that if sufficient processing power (and not just CPU power but intelligent software such as AutoGAD) is available on UAVs with HSI sensors, the requirement for data link capacity is drastically reduced. The technology potentially offered by AutoGAD offers planners the option of migrating towards an air-centric (processor based) architecture rather than ground-centric (communications) architecture that requires higher data transfer rates. Further, by removing the need to relay an entire HSI data matrix to the operator, who must process the data at the control station to make a determination for targeting, the ability of a UAV to process the data via artificial intelligence in real time and transmit a small file with a determination of target locations (with some confidence attached) dramatically improves capability during time critical targeting scenarios.

Even if not employed as an on board processing algorithm on a UAV with an HSI sensor, AutoGAD shows potential as an effective target detection algorithm for use by intelligence analysts. The full MATLAB code for AutoGAD will be provided in the hopes that researchers will further test AutoGAD to validate its effectiveness and refine the algorithm. Further, this author hopes operators in the intelligence community will use the algorithm and that it provides additional capability.

In closing, the end product delivered by this author is a fast, truly autonomous (unsupervised) global anomaly detection algorithm. Dimensionality determination, target feature selection, and target identification have been improved and fully automated. Its speed and automation make it of practical use in an operational environment.

Appendix. MATLAB Code

```matlab
%**********************************************************************%
%AutoGAD version 1.0                                                  %
%                                                                     %
%                                                                     %
%Hyperspectral Autonomous Global Anomaly Detector (AutoGAD)           %
%Using FastICA                                                        %
%                                                                     %
%Author: Capt Robert Joseph Johnson                                   %
%Feb 2008                                                             %
%**********************************************************************%
clear all;
close all;
clc;


%Tactical Decisions By User-------------------------------------------
functn=2;%objective function in ICA to use.  Options [1=tanh, 2=pow3]
orthogonalization=1;%find ICs in parallel (symm) or one by one (delf).
%Options [symm=1, defl=2]
dim_adjustment=0;%how much to adjust max distance log scale secant line (MDLS)
%dimensionality decision
max_score_thresh=10;%threshold above which decision is made to declare target
bin_width_SNR=.05;%bin width when using zero-detection histogram method to
%determine breakpoint between background and potential targets for
%calculating potential target SNR (PT SNR)
PT_SNR_thresh=2;%2;%threshold above which decision is made to declare target
bin_width_ident=.05;%bin width when using zero-detection histogram method to
%determine breakpoint between background and targets for identifying target
%pixels from selected target signals
threshold_both_sides=0;%1=identifiy outliers on both sides of IC signal,
%0=identify ouliers on side with highest magnitude scores only
clean_sig=1;%0 = no signal smoothing, 1 = signal smoothing prior to target
%identification
smooth_iter_high=100;%number of iterations to complete for iterative smoothing
%of low SNR object
smooth_iter_low=20;%20;%number of iterations to complete for iterative smoothing
%of high SNR object
low_SNR=10;%Threshold decision for choosing smooth_iter_low or smooth_iter_high
window_size=3;%image window size for smoothing
show_plots=2;%1=yes, 2=no
%---------------------------------------------------------------------

switch num2str(functn)
    case '1'
        funct='tanh';
    case '2'
        funct='pow3';
end

switch num2str(orthogonalization)
    case '1'
        orthog='symm';
    case '2'
        orthog='defl';
end


%--------------Solicit User Input to Load HSI Image File---------------
display('This program requires the Image Processing Toolbox for MATLAB.');
```

A-1

```matlab
display('Make sure your version of MATLAB has this toolbox.');
display('   ');
display('Make sure you have in your working directory the all the files for');
display('FastICA and the Center_and_PCA.m file');
display('   ');
display('The first several lines in the AutoGAD algorithm detail default');
display('settings for AutoGAD.  If you would like to experiment');
display('changing these settings, hit ctrl c to interrupt this run.  Open');
display('up AutoGAD in the the editor and make changes.');
display('   ');
display('Please hit enter');
display('  ');
answer=input('');
display('Enter you image cube file name to be processed.');
display('File should be in .mat format ');
display('  ');
display('!Make sure to put it in single quotes!')
display('!Make sure the image cube is in the same directory as this code!');
display('  ');
temp1=input('');
temp2=struct2cell(load(temp1));
im_cube=temp2{1};
display('   ');
display ('Enter truth mask');
display('  ');
display('If you do not have a truth mask and this is a real target search');
display('with no truth knowledge, enter 0');
display('  ');
temp3=input('');
if temp3~=0;
    temp4=struct2cell(load(temp3));
    truth=temp4{1};
end
clear temp1
clear temp2
clear temp4
clc;
display('   ');
display('Please enter the good bands for this HSI sensor');
display('These are the bands that are NOT the atmospheric absorption bands');
display('   ');
display('If this the the 210 band HYDICE sensor, LtCol Tim Smetek concluded');
display('that the good_bands = [5:72, 78:85, 92:99, 116:134,158:199]');
display(' ');
display('If this is HYDICE data and you would like to keep these bands, type');
display('1 and hit enter');
display('   ');
display('If this is not HYDICE data or you do not want to keep those bands');
display('just hit enter and then enter the bands you wish to keep');
display(' ');
answer=input('');
if answer==1
    good_bands=[5:72,78:85,92:99,116:134,158:199];
else
    good_bands=input('good_bands = ');
end
%-------------------------------------------------------------------------


%-----------Ask User if they want to see color image----------------------
display('  ');
```

```matlab
display('Do you want to see a RGB image of your HSI file?');
display('  ');
display('If so, enter 1. If not just hit enter.           ');
display('   ');
answer=input('');
if answer==1
    Red=input('Please enter the band number for red, HYDICE is 50        ');
    display(' ');
    Green=input('Please enter the band number for green, HYDICE is 29     ');
    display(' ');
    Blue=input('Please enter the band number for blue, HYDICE is 22       ');

    R=im_cube(:,:,Red);
    G=im_cube(:,:,Green);
    B=im_cube(:,:,Blue);

    %Borrowed from Lt Col Tim Smetek, lines 142 - 163, offer a way to make
    %an RGB image look better.  The following lines are used in conjunction
    %with the mat2gray function to perform a 2% linear stretch on the image
    %data

    m1=size(R,1);
    n=size(R,2);
    low_id=floor(0.02*m1*n);
    hi_id=floor(0.98*m1*n);

    r_vec=reshape(R,m1*n,1);
    r_vec=sort(r_vec);
    r_vec=double(r_vec);
    min_R=r_vec(low_id);
    max_R=r_vec(hi_id);

    g_vec=reshape(G,m1*n,1);
    g_vec=sort(g_vec);
    g_vec=double(g_vec);
    min_G=g_vec(low_id);
    max_G=g_vec(hi_id);

    b_vec=reshape(B,m1*n,1);
    b_vec=sort(b_vec);
    b_vec=double(b_vec);
    min_B=b_vec(low_id);
    max_B=b_vec(hi_id);

    %The IPT function mat2gray to scales the values in each matrix between 0
    %and 1.  This is necessary because the matrices are of type double and
    %imshow requires double value matrices to be scaled between 0 and 1

    R=mat2gray(double(R),[min_R max_R]);
    G=mat2gray(double(G),[min_G max_G]);
    B=mat2gray(double(B),[min_B max_B]);

    %**Now stack the three matrices into a 3D array and display the image
    RGB=cat(3,R,G,B);
    figure (1)
    imshow(RGB,[]);
    title('True Color Image');
    impixelinfo;
    %**Turn-on the interactive pixel value utility
    clear R G B
    clear RGB
```

```
    clear r_vec g_vec b_vec
end
%-----------------------------------------------------------------------


tic;
%----Resize Image Cube into matrix where each row is a pixels-------------
%----------------signature in the spectral bands-------------------------
dims=size(im_cube,3);
num_pixels=size(im_cube,1)*size(im_cube,2);
num_lines=size(im_cube,1);
num_col=size(im_cube,2);

%**Place all the pixel vectors into a single matrix where each row
%corresponds to a pixel vector
data_matrix=zeros(num_pixels,dims);
data_matrix_truth=zeros(num_pixels, 1);
for x=1:dims
    data_matrix(:,x)=reshape(im_cube(:,:,x),num_pixels,1);
end
clear im_cube;
%If HSI cube is too large for MATLAB since MATLAB converts variables to
%double precision, this will make file smaller so that MATLAB can
%operate on it.
if num_pixels*dims > 25*10^6
    data_matrix=single(data_matrix);
end
%%%%%%%%%%%%%%%%%

if temp3~=0;
    data_matrix_truth=reshape(truth,num_pixels,1);
end
%-----------------------------------------------------------------------


%----------Keep bands that are not atmospheric absorption bands-----------
data_matrix_new=data_matrix(:,good_bands);
dims=size(data_matrix_new,2);
clear data_matrix;
%-----------------------------------------------------------------------


%-----------------------------Perform PCA--------------------------------
[Ac,Lc,TotVarCompC,YscorC]=Center_and_PCA(data_matrix_new);
%Function written by Capt Johnson does PCA on covariance matrix

Lplot=diag(Lc);
%checks for eigenvalues 10^-4 and smaller and moves the endpoint of the
%eigenvalue curve to the point where eigenvalues are greater than 10^-4
%so that the MDSL method in the next section is not biased by pathological
%cases where the endpoint of the log scale eigenvalue curve has extremely
%small endpoints and grossly alters the theoretical shape of the curve that
%should arise for eigenvalues of covariance matrices of spectral data
%that follow the LMM
while Lplot(dims)<=10^-4;
    dims=dims-1;
end
L=log10(Lplot);
clear data_matrix_new;
clear Ac;
clear Lc;
```

A-4

```matlab
%-------------------------------------------------------------------------

%--------------------Dimensionality Assessment--------------------------
%slope of line connecting endpoints of scree plot of eigenvalues
m_slope = (L(1)- L(dims))/(1-dims);

%calculate Euclidean distances from scree plot curve to line connecting
%endpoints
Eqdist=[];
for i=1:dims
    x_int=(L(i) - L(1)+ m_slope + (i/m_slope))/(m_slope + (1/m_slope));
    y_int=L(1)+ m_slope*(x_int-1);
    Eqdist(i)=sqrt((i-x_int)^2+(L(i)-y_int)^2);
end
%find the point on the log scale eigencurve curve with the largest distance
%from the line connecting the endpoints
[max_Eqdist, index_dim]=max(Eqdist);
reduced_dim = index_dim;
k=reduced_dim-1;
k=k+dim_adjustment;
percent_var=TotVarCompC(k,1);
Y=YscorC(:,1:k);
clear YscorC;
if show_plots==1
    figure(3);
    semilogy(Lplot(1:dims), '.-');
    title({'Plot of Eigenvalue vs. PC Component',...
        sprintf('Dimensionality = %i',k)},'fontweight','b');
end
%-------------------------------------------------------------------------

%--------------------Perform ICA on reduced PCA space--------------------
[icasig, A, W]=fastica(Y','approach',orthog, 'g', funct, 'epsilon',...
    .00001, 'stabilization','on', 'verbose','off');
icasig=icasig';
%If an IC score has a high signals, make them always positive
for j=1:k
    if abs(min(icasig(:,j)))>max(icasig(:,j))
        icasig(:,j)=-icasig(:,j);
    end
end
clear Y
%-------------------------------------------------------------------------

%------------------Find the Kurtosis of Each Signal----------------------
kurt=zeros(k,1);
for j=1:k
    kurt(j)=abs(kurtosis(icasig(:,j)));
end
%this statistic will not be used in AutoGAD, but is included if the user
%wishes to compare this value to the PT SNR and max pixel score values
%-------------------------------------------------------------------------

%-------------------Find the Max Score of Each Signal--------------------
maxim=zeros(k,1);
for j=1:k
    maxim(j)=max(icasig(:,j));
end
```

```matlab
%-------------------------------------------------------------------------

%-------------------Find the PT SNR of each signal----------------------
for j=1:k
    bins=[];
    freq=[];
    bins=min(icasig(:,j)):bin_width_SNR:max(icasig(:,j));
    freq=hist(icasig(:,j),bins);
    count=1;
    %find the bin that is at center (mean) of the ICA signal
    for i=1:size(bins,2)
        if bins(i)<0
            count=count+1;
        else
            break
        end
    end
    %from the center of the signal keep counting until the first zero
    %bin is found
    count1=1;
    for i=count+1:size(freq,2)
        if freq(i)> 0;
            count1=count1+1;
        else
            break
        end
    end
    %if there are no zero bins until the very end of the tail then the
    %threshpoint is set above the last data point
    if count+count1 > size(bins,2)
        thresh_pt(j)=max(icasig(:,j));
    else
    %otherwise use the first bin were a zero value occurs
        thresh_pt(j)=bins(count+count1);
    end
end
PT_SNR=zeros(k,1);
for j=1:k
    potent_target=[];
    potent_bkrd=[];
    %find the indices of those pixels greater than threshold
    ind = icasig(:,j)>thresh_pt(j);
    %store those pixels greater than threshold in vector
    potent_target=icasig(ind,j);
    if size(potent_target,1)==0
        potent_target=0;
    end
    %find the indices of those pixels less than threshold
    ind2 = icasig(:,j)<=thresh_pt(j);
    %store those pixels less than threshold in vector
    potent_bkrd=icasig(ind2,j);
    power_target(j)=var(potent_target);
    power_bkrd(j)=var(potent_bkrd);
    PT_SNR(j)=10*log10(power_target(j)/power_bkrd(j));
end
%-------------------------------------------------------------------------
one=ones(num_pixels,1);
```

```matlab
%---Plot Abundance Maps from ICs Frames with PT SNR and Max Pixel Score----
if show_plots==1
    figure (4)
    d=ceil(sqrt(k));
    for j=1:k
        subplot(d,d,j)
        ICsig(:,:,j)=reshape(icasig(:,j),num_lines,num_col);
        ICsig_grey(:,:,j)=mat2gray(double(ICsig(:,:,j)));
        imshow(ICsig_grey(:,:,j));
        if maxim(j)>=max_score_thresh && PT_SNR(j)>=PT_SNR_thresh
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',j,PT_SNR(j)...
                ,maxim(j)),'Potential Target'},'fontweight','b');
        else
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',j,PT_SNR(j)...
                ,maxim(j)),'Non-Target'},'fontweight','b');
        end
    end
    clear ICsig;
    clear ICsig_grey;
%-------------------------------------------------------------------------


%------------------------Plot IC signals----------------------------------
    figure(5)
    PT_SNR_line=[];
    for j=1:k
        PT_SNR_line(:,j)=thresh_pt(j)*one;
    end
    for j=1:k
        subplot(d,d,j)
        plot(icasig(:,j),'.', 'MarkerEdgeColor','r');
        hold on
        plot(PT_SNR_line(:,j),'LineWidth',2);
        xlabel('Pixel');
        ylabel('Abundance (IC Score)');
        if maxim(j)>=max_score_thresh && PT_SNR(j)>=PT_SNR_thresh
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',j,PT_SNR(j)...
                ,maxim(j)),'Potential Target'},'fontweight','b');
        else
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',j,PT_SNR(j)...
                ,maxim(j)),'Non-Target'},'fontweight','b');
        end
        axis([0,num_pixels,-10,30]);
    end
    clear PT_SNR_line
%-------------------------------------------------------------------------
end

%----------------Keep only Those Signals Above Both Thresholds-------------
ind_max=[];
ind_SNR=[];
ind_both=[];
ind_max = maxim>=max_score_thresh;
ind_SNR = PT_SNR>=PT_SNR_thresh;
ind_both=ind_max+ind_SNR;
[rind,cind]= find(ind_both==2);
if size(rind,1)==0
    display('NO TARGETS')
    target_sig=zeros(num_pixels,1);
    target_vec=zeros(num_pixels,1);
else
```

```matlab
        target_sig=icasig(:,rind);
end
clear icasig;
num_tgt_maps=size(target_sig,2);
for j=1:num_tgt_maps
    tgt_sig_map(:,:,j)=reshape(target_sig(:,j),num_lines,num_col);
end
%-------------------------------------------------------------------------
if size(rind,1)~=0
%-------------Show Abundance Maps of Retained Target Signals---------------
    if show_plots==1
        d=ceil(sqrt(num_tgt_maps));
        tgt_gray=[];
        figure(6)
        for j=1:num_tgt_maps
            subplot(d,d,j);
            tgt_sig_map_gray(:,:,j)=mat2gray(tgt_sig_map(:,:,j));
            imshow(tgt_sig_map_gray(:,:,j));
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',rind(j),...
                PT_SNR(rind(j)),maxim(rind(j))),'Potential Target'},...
                'fontweight','b');
        end
        clear tgt_sig_map_gray
    end
    clear target_sig
%-------------------------------------------------------------------------


%----Clean (IAN Filtering) target Signals prior to Identification----------
    if clean_sig==1
        for j=1:num_tgt_maps
            if PT_SNR(rind(j))<low_SNR
                for c=1:smooth_iter_high
                    [tgt_sig_map(:,:,j)]=wiener2(tgt_sig_map(:,:,j), ...
                        [window_size,window_size]);
                end
            else
                for c=1:smooth_iter_low
                    [tgt_sig_map(:,:,j)]=wiener2(tgt_sig_map(:,:,j), ...
                        [window_size,window_size]);
                end
            end
        end
    end
%-------------------------------------------------------------------------


%------------------Plot IAN Filtered Target Maps------------------------------
        if show_plots==1
            for j=1:num_tgt_maps
                clean_map_gray(:,:,j)=mat2gray(tgt_sig_map(:,:,j));
            end
            figure(7)
            for j=1:num_tgt_maps
                subplot(d,d,j);
                imshow(clean_map_gray(:,:,j));
                title(sprintf('Filtered Map %i',rind(j)),'fontweight','b');
            end
        end
    end
%-------------------------------------------------------------------------
```

```matlab
%----------Identify Target Pixels from Selected Target Maps----------------
%-------------------------------------------------------------------------
    target_sig_clean=[];
    for j=1:num_tgt_maps
        target_sig_clean(:,j)=reshape(tgt_sig_map(:,:,j), num_pixels, 1);
    end
    for j=1:num_tgt_maps
        bins=[];
        freq=[];
        bins=(min(target_sig_clean(:,j))):bin_width_ident:...
            max(target_sig_clean(:,j));
        freq=hist(target_sig_clean(:,j),bins);
        count=1;
        %find the bin that is at center (mean) of the ICA signal
        for i=1:size(bins,2)
            if bins(i)<0
                count=count+1;
            else
                break
            end
        end
        %from the center of the signal keep counting until the first zero
        %bin is found
        count1=1;
        for i=count+1:size(freq,2)
            if freq(i)> 0;
                count1=count1+1;
            else
                break
            end
        end
        %if there are no zero bins until the very end of the tail then the
        %threshpoint is set above the last data point
        if count+count1 > size(bins,2)
            thresh_pt_ident(j)=max(target_sig_clean(:,j))+.01;
        else
        %otherwise use the first bin were a zero value occurs
            thresh_pt_ident(j)=bins(count+count1);
        end
    end

    target=zeros(num_pixels, num_tgt_maps);
    for j=1:num_tgt_maps
        ind_tgt = target_sig_clean(:,j)>thresh_pt_ident(j);
        target(:,j)=ind_tgt;
    end

    target_vec=zeros(num_pixels,1);
    for j=1:num_tgt_maps
        target_vec=target_vec + target(:,j);
    end
end
%-------------------------------------------------------------------------
%Checks both sides of the selected target signals for target pixels if user
%specified this option
if threshold_both_sides==1
    target_sig_clean_left=-target_sig_clean;
    for j=1:num_tgt_maps
        bins=[];
        freq=[];
```

```matlab
            bins=(min(target_sig_clean_left(:,j))):bin_width_ident:...
                max(target_sig_clean_left(:,j));
            freq=hist(target_sig_clean_left(:,j),bins);
            count=1;
            for i=1:size(bins,2)
                if bins(i)<0
                    count=count+1;
                else
                    break
                end
            end
            count1=1;
            for i=count+1:size(freq,2)
                if freq(i)> 0;
                    count1=count1+1;
                else
                    break
                end
            end
            if count+count1 > size(bins,2)
                thresh_pt_ident_left(j)=max(target_sig_clean_left(:,j))+.01;
            else
                thresh_pt_ident_left(j)=bins(count+count1);
            end
        end


    target_left=zeros(num_pixels, num_tgt_maps);
    for j=1:num_tgt_maps
        ind_tgt = target_sig_clean_left(:,j)>thresh_pt_ident_left(j);
        target_left(:,j)=ind_tgt;
    end

    target_vec_left=zeros(num_pixels,1);
    for j=1:num_tgt_maps
        target_vec_left=target_vec_left + target_left(:,j);
    end
    target_vec=target_vec+target_vec_left;
end
%-------------------------------------------------------------------------
target_pic = reshape(target_vec,num_lines,num_col);
%-------------------------------------------------------------------------
%-------------------------------------------------------------------------

%----------Plot Target Signals with Calculated Thresholds-----------------
if show_plots ==1
    if size(rind,1)~=0
        d=ceil(sqrt(num_tgt_maps));
        linetrh_ident=[];
        for j=1:num_tgt_maps
            linetrh_ident(:,j)=thresh_pt_ident(j)*one;
        end
        if threshold_both_sides==1
            for j=1:num_tgt_maps
                linetrh_ident_left(:,j)=-thresh_pt_ident_left(j)*one;
            end
        end
        figure(8)
        for j=1:num_tgt_maps
            subplot(d,d,j)
            plot(target_sig_clean(:,j),'.', 'MarkerEdgeColor','r');
```

```matlab
            hold on
            plot(linetrh_ident(:,j),'LineWidth',2);
            if threshold_both_sides==1
                hold on
                plot(linetrh_ident_left(:,j),'LineWidth',2);
            end
            xlabel('Pixel');
            ylabel('Abundance (IC Score)');
            title({sprintf('Map %i \n SNR %4.3f \n Max Score %4.3f',rind(j),...
                PT_SNR(rind(j)),maxim(rind(j))),'Potential Target'},...
                'fontweight','b');
            axis([0,num_pixels,-10,30]);
        end
        clear linetrh_ident
    end
    clear one
end
%-------------------------------------------------------------------------

%-------Grade Performance of AutoGAD if Truth Mask was Provided------------
if temp3~=0;
%--------------------------Confusion Matrix Calculation--------------------
    ConfusMat=[];
    ConfusMat(1,1)=0; %(TP)
    ConfusMat(1,2)=0; %(FP)
    ConfusMat(2,1)=0; %(FN)
    ConfusMat(2,2)=0; %(TN)

    for i=1:num_pixels
        if target_vec(i,1)>= 1 && data_matrix_truth(i,1) >= 1
            ConfusMat(1,1)=ConfusMat(1,1)+1;
        else
            if target_vec(i,1)>= 1 && data_matrix_truth(i,1) == 0
                ConfusMat(1,2)=ConfusMat(1,2)+1;
            else
                if target_vec(i,1)== 0 && data_matrix_truth(i,1) == 1
                    ConfusMat(2,1)=ConfusMat(2,1)+1;
                else
                    if target_vec(i,1)== 0 && data_matrix_truth(i,1) == 0 || 2
                        ConfusMat(2,2)=ConfusMat(2,2)+1;
                    end
                end
            end
        end
    end

    APER = (ConfusMat(1,2)+ConfusMat(2,1))/(num_pixels);
    TPF = ConfusMat(1,1)/(ConfusMat(1,1)+ConfusMat(2,1));
    FPF = ConfusMat(1,2)/(ConfusMat(1,2)+ConfusMat(2,2));
    Perc_tgt = ConfusMat(1,1)/(ConfusMat(1,1)+ConfusMat(1,2));
%-------------------------------------------------------------------------
```

```matlab
%------------------Show Target Locations to the User---------------------
    target_vec_color=zeros(num_pixels,1);
    for i=1:num_pixels
        if target_vec(i,1)>=1 && data_matrix_truth(i,1)>=1
            target_vec_color(i,1)=4;
        elseif target_vec(i,1)>=1 && data_matrix_truth(i,1)==0
            target_vec_color(i,1)=2;
        end
    end
    target_pic_color = uint8(reshape(target_vec_color,num_lines,num_col));
    if size(rind,1)~=0
        figure(9)
        imshow(mat2gray(target_pic_color));
        colormap('Hot')
        title(sprintf('TPF = %4.6f \n FPF = %4.6f \n Percent TGT = %4.6f',...
            TPF, FPF,Perc_tgt),'fontweight','b');
        impixelinfo;
    elseif size(rind,1)==0
        figure(9)
        imshow(target_pic);
        title('No Targets Detected')
    end
    figure (2)
    imshow(truth,[]);
    title('Truth Mask');
    impixelinfo;
else
    if size(rind,1)~=0
        figure(9)
        imshow(target_pic)
        title({'Suspected Target Pixels'});
        impixelinfo;
    elseif size(rind,1)==0
        figure(9)
        imshow(target_pic)
        title({'No Targets Detected'});
        impixelinfo;
    end
end
time=toc
```

```matlab
%PCA Data Analysis
%by Capt Robert Joseph Johnson

%--------------------------------------------------------------------------
%----------------------------Variables-------------------------------------
% X = n observations x p variables data matrix
% Xd = centered data matrix
% C = covariance of X
% Lc = diagonal matrix of eigenvalues of C
% Ac = matrix of eigenvectors of C
% YscorC = matrix of component scores if input is C
%--------------------------------------------------------------------------


%-------------------Function and Initial Variable Definitions--------------

function [Ac,Lc,TotVarCompC,YscorC]=Center_and_PCA(X)
n = size(X,1);
p = size(X,2);
one = ones(n,1);

%center data_matrix
Xbar = mean(X);
Xd = X - one*Xbar;
%--------------------------------------------------------------------------


%--------------------PCA based on Cov(x) = C-------------------------------
C=cov(X);

%----Sort in descending order Eigenvalues and associated Eigenvectors------
[ac,lc] = eig(C);
[lc_ordered, ord_value_lc] = sort(diag(lc), 'descend');
Lc = diag(lc_ordered);
for i=1:p
    Ac(:,i)=ac(:,ord_value_lc(i));
end
%--------------------------------------------------------------------------


%-------Calculate percentage of total variance for each compnent----------
TotVar = trace(Lc);
TVC = 0;
TotVarCompC=[];
for i=1:p
    TVC = TVC + Lc(i,i)/TotVar;
    TotVarCompC = [TotVarCompC;TVC,i];
end
%------------------------------------------------------------------
YscorC = Xd*Ac; %Component Scores
```

Bibliography

Bauer, Kenneth W. Course Notes, OPER 685, Applied Multivariate Analysis I, Department of Operational Sciences, Air Force Institute of Technology, Wright-Patterson AFB OH. March 2007: 88, 95

Bell, Anthony J. and Terrence J. Sejnowski. "The Independent Components of Natural Scenes are Edge Filters," *Vision Research*, Vol. 37, No. 23: 3335 (April 1997).

Chang, Chein-I. *Hyperspectral Data Exploitation*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2007.

Chen, C. H. *Signal and Image Processing for Remote Sensing*. Boca Raton, Florida: Taylor and Francis Group, 2007.

Chiang, Shao-Shan, Chein-I Chang, and Irving Ginsberg. "Unsupervised Target Detection in Hyperspectral Images Using Projection Pursuit," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 39, No. 7: 1383 (July 2001).

Dillon, William R. and Matthew Goldstein. *Multivariate Analysis*. New York: John Wiley & Sons, Inc., 1984.

Du, Qian, Ivica Kopriva, and Harold Szu. "Independent-component analysis for hyperspectral remote sensing imagery classification," *Optical Engineering* 45(1), 017008: 2-3 (January 2006).

Ehrgott, Matthias. *Multicriteria Optimization*. Berlin, Heidelburg, and New York: Springer, 2005.

Hyvärinen, Aapo and others. *Independent Component Analysis*. New York: John Wiley & Sons, Inc., 2001.

Hyvärinen, Aapo and Erkki Oja. "Independent component analysis: algorthims and applications," *Neural Networks*, 13: 414-415 (2000).

Koo, Robert. *Feature Extraction Using Principal and Independent Component Analysis for Hyperspectral Imagery*. MS thesis, AFIT/GOR/07M-16. Department of Operational Sciences, Air Force Institute of Technology (AU), Wright Patterson AFB OH, May 2007.

Landgrebe, David A. *Signal Theory Methods in Multispectral Remote Sensing*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2003.

Nascimento, José, M. P. and José M. Bioucas Dias. "Does Independent Component Analysis Play a Role in Unmixing Hyperspectral Data?," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 1: 175-176, 186 (January 2005).

Neher, Robert and Anuj Srivastava. "A Bayesian MRF Framework for Lebeling Terrain Using Hyperspectral Imaging," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 6: 1365 (June 2005).

Office of the Secretary of Defense. *Unmanned Arial Vehicles Roadmap 2000 –2025.* April 2001 http://www.acq.osd.mil/usd/road.doc.

Robila, Stefan A. and Lukasz Maciack. "New Approaches for Feature Extraction in Hyperspectral Imagery" *Systems, Applications and Technology Conference, 2006. LISAT 2006. IEEE Long Island*: 2 (5 May 2006).

Robila, Stefan A. and Pramod K. Varshney. "Target Detection in Hyperspectral Images Based on Independent Component Analysis" *Proceedings of SPIE*, Vol. 4726: 176-178 (2002).

Sarigul, E. and M. S. Alam. "Hyperspectral Target Detection Using Independent Component Analysis Based Linear Mixture Model" *Proceedings of SPIE*, Vol. 6565: 65650A-2 (2007).

Shaw, Gary and Dimitris Manolakis. "Signal Processing for Hyperspectral Image Exploitation." *IEEE Signal Processing Magazine*, 19:13 (January 2002).

Smetek, Timothy E. *Hyperspectral Imagery Target Detection Using Improved Anomaly Detection and Signature Matching Methods.* Air Force Institute of Technology (AU), Wright Patterson AFB OH, May 2007.

Stein, David W.J., Scott G. Beaven, Lawrence E. Hoff, Edwin M. Winter, Alan P. Schaum and Alan D. Stocker. "Anomaly Detection for Hyperspectral Imagery," *IEEE Signal Processing Magazine,* 19:58-63 (January 2002).

Stocker, Alan D., Eskandar Ensafi, and Clark Oliphint. "Applications of eigenvalue distribution theory to hyperspectral processing," *Proceedings of SPIE*, Vol. 5093: 652-654, 657-664 (2003).

Taitano, Yuri. *Hyperspectral Imagery Target Detection Using the Iterative RX Detector.* MS thesis, AFIT/GOR/07M-25. Department of Operational Sciences, Air Force Institute of Technology (AU), Wright Patterson AFB OH, May 2007.

Varshney, Pramod K. and Manoj K. Arora. *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data.* New York: Springer, 2004.

Wang, Jing and Chein-I Chang. "Applications of Independent Component Analysis (ICA) in Endmember Extraction and Abundance Quantification for Hyperspectral Imagery" *Proceedings of SPIE*, Vol. 6233: 623314-3 (2006).

Wang, Jing and Chein-I Chang. "Independent Component Analysis-Based Dimensionality Reduction With Applications in Hyperspectral Image Analysis" *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 44, No. 6: 1587 (June 2006).

Vita

Captain Robert J. Johnson graduated valedictorian from Morristown Hamblen High School East in Morristown, Tennessee. He entered undergraduate studies at Carson-Newman College, Tennessee where he graduated salutatorian and summa cum laude with a Bachelor of Arts degree in Mathematics in May 2000. Upon graduation he worked as a high school mathematics teacher in Raleigh and Durham, North Carolina. In April 2003 he was selected to attend United States Air Force Officer Training School and was commissioned as a second lieutenant on June 27, 2003.

His first assignment was at Tyndall AFB, FL as a missile analyst responsible for designing flight profiles and analyzing missile telemetry from live fire testing involving F-15s, F-16s, and F-22s firing the AIM-7 and AIM-120 air-to-air missiles. In August 2006, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology to pursue a Master of Science degree in Operations Research. Upon graduation in March 2008, he will be assigned to the DLA Office of Operations Research and Resource Analysis (DORRA) in Richmond, Virginia.

REPORT DOCUMENTATION PAGE		*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 02-25-2008	2. REPORT TYPE **Master's Thesis**	3. DATES COVERED *(From – To)* Jul 2007 - Mar 2008
4. TITLE AND SUBTITLE IMPROVED FEATURE EXTRACTION, FEATURE SELECTION, AND INDETIFICATION TECHNIQUES THAT CREATE A FAST UNSUPERVISED HYPERSPECTRAL TARGET DETECTION ALGORITHM		5a. CONTRACT NUMBER
		5b. GRANT NUMBER
		5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Johnson, Robert, J., Captain, USAF		5d. PROJECT NUMBER
		5e. TASK NUMBER
		5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765		8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GOR/ENS/08-07
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally left blank		10. SPONSOR/MONITOR'S ACRONYM(S)
		11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research extends the emerging field of hyperspectral image (HSI) target detectors that assume a global linear mixture model (LMM) of HSI and employ independent component analysis (ICA) to unmix HSI images. Via new techniques to fully automate feature extraction, feature selection, and target pixel identification, an autonomous global anomaly detector, AutoGAD, has been developed for potential employment in an operational environment for real-time processing of HSI targets. For dimensionality reduction (initial feature extraction prior to ICA), a geometric solution that effectively approximates the number of distinct spectral signals is presented. The solution is based on the theory of the shape of the eigenvalue curve of the covariance matrix of spectral data containing noise. For feature selection, previously a subjective definition called significant kurtosis change was used to denote the separation between targets classes and non-target classes. This research presents two new measures, potential target signal to noise ratio (PT SNR) and max pixel score which computed for each of the ICA features to create a new two-dimensional feature space where the overlap between target and non-target classes is reduced compared to the one dimensional kurtosis value feature space. Finally, after target feature selection, adaptive noise filtering, but with an iterative approach, is applied to the signals. The effect is a reduction in the power of the noise while preserving the power of the target signal prior to target identification to reduce false positive detections. A zero-detection histogram method is applied to the smoothed signals to identify target locations to the user. Matlab code for the AutoGAD algorithm is provided.

15. SUBJECT TERMS

Hyperspectral Imagery (HSI), Unsupervised Target Detection, Feature Extraction, Principal Component Analysis (PCA), Autonomous Dimensionality Reduction, Linear Mixture Model (LMM), Independent Component Analysis (ICA), Feature Selection, Potential Target Signal to Noise Ratio, Max Pixel Score, Target Identification, Iterative Adaptive Noise Filtering

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Kenneth W. Bauer (ENS)
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	244	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636, ext 4631; e-mail: Kenneth.Bauer@afit.edu
U	U	U			