

Kernel-Based Anomaly Detection in Hyperspectral Imagery

Heesung Kwon and Nasser M. Nasrabadi
Army Research Laboratory ATTN: AMSRL-SE-SE,
2800 Powder Mill Road, Adelphi, MD 20783

Abstract

In this paper we present a nonlinear version of the well-known anomaly detection method referred to as the RX-algorithm. Extending this algorithm to a feature space associated with the original input space via a certain nonlinear mapping function can provide a nonlinear version of the RX-algorithm. This nonlinear RX-algorithm, referred to as the kernel RX-algorithm, is basically intractable mainly due to the high dimensionality of the feature space produced by the non-linear mapping function. However, in this paper it is shown that the kernel RX-algorithm can easily be implemented by *kernelizing* it in terms of *kernels* which implicitly compute dot products in the feature space. Improved performance of the kernel RX-algorithm over the conventional RX-algorithm is shown by testing several hyperspectral imagery for military target and mine detection.

1 Introduction

Anomaly detectors are pattern recognition schemes that are used to detect objects that might be of military interest. Almost all the anomaly detectors attempt to locate anything that looks different spatially or spectrally from its surroundings. In spectral anomaly detection algorithms, pixels (materials) that have a significantly different spectral signature from their neighboring background clutter pixels are identified as spectral anomalies. Spectral anomaly detection algorithms [1–5] could also use spectral signatures to detect anomalies embedded within a background clutter with a very low signal-to-noise ratio. In spectral anomaly detectors, no prior knowledge of the target spectral signature are utilized or assumed.

Most of the detection algorithms in the literature [1, 5–7] assume that the HSI data can be represented by the multivariate normal (Gaussian) distribution and under the Gaussianity assumption, the generalized likelihood ratio test (GLRT) is used to test the hypotheses to find the existence of a target in the image. The Gaussianity assumption has been used mainly because of mathematical tractability that allows the formation of widely used detection models, such as GLRT. However, in reality the HSI data might not closely follow the Gaussian distribution. Nevertheless, in

various fields of signal processing, GLRT is used to detect signals (targets) of interest in noisy environments.

In this paper we formulated a nonlinear version of the RX-algorithm by transforming each spectral pixel into a very high-dimensional feature space (could be infinite dimension) by a nonlinear mapping function. The spectral pixel in the feature space now consists of possibly the original spectral bands and a nonlinear combination of the spectral bands of the original spectral signature. Implementing the RX-algorithm in the feature space, the higher order correlations between spectral bands are exploited, thus resulting in a nonlinear RX-algorithm. However, this nonlinear RX-algorithm cannot be implemented directly due to the high dimensionality of the feature space. It is shown in Section 4 that because the RX-algorithm consists of inner products of spectral vectors, it is possible to implement a kernel-based nonlinear version of the RX-algorithm by using kernel functions, and their properties [8].

Kernel-based versions of a number of feature extraction or pattern recognition algorithms have recently been proposed [9–14]. In [12], a kernel version of principal component analysis (PCA) was proposed for nonlinear feature extraction and in [13] a nonlinear kernel version of the Fisher discriminant analysis was implemented for pattern classification. In [14], a kernel-based clustering algorithm was proposed and in [10] kernels were used as generalized dissimilarity measures for classification. Kernel methods have also been applied to face recognition in [9].

This paper is organized as follows. Section 2 provides an introduction to the RX-algorithm. Section 3 describes kernel functions and their relationship with the dot product of input vectors in the feature space. In Section 4 we show the derivation of the kernel version of the RX-algorithm. Experimental results comparing the RX-algorithm and the kernel-based RX-algorithm are given in Section 5. Finally, in Section 6 conclusion and discussion are provided.

2 Introduction to RX-ALGORITHM

Reed and Yu in [6] developed a GLR test, so called the RX anomaly detection, for multidimensional image data as-

suming that the spectrum of the received signal (spectral pixel) and the covariance of the background clutter are unknown. Let each input spectral signal be denoted by a vector $\mathbf{x}(n) = (x_1(n), x_2(n), \dots, x_J(n))^T$ consisting of J spectral bands. Define \mathbf{X}_b to be a $J \times M$ matrix of the M reference background clutter pixels. Each observation spectral pixel is represented as a column in the sample matrix \mathbf{X}_b

$$\mathbf{X}_b = [\mathbf{x}(1) \ \mathbf{x}(2) \ \dots \ \mathbf{x}(M)]. \quad (1)$$

The two competing hypotheses that the RX-algorithm must distinguish are given by

$$\begin{aligned} \mathbf{H}_0 : \mathbf{x} &= \mathbf{n}, & \text{Target absent} \\ \mathbf{H}_1 : \mathbf{x} &= a\mathbf{s} + \mathbf{n}, & \text{Target present} \end{aligned} \quad (2)$$

where $a = 0$ under \mathbf{H}_0 and $a = 1$ under \mathbf{H}_1 , respectively. \mathbf{n} is a vector that represents the background clutter noise process, and \mathbf{s} is the spectral signature of the signal (target) given by $\mathbf{s} = [s_1, s_2, \dots, s_J]$. The target signature \mathbf{s} and background covariance C_b are assumed to be unknown. The model assumes that the data arises from two normal PDFs with the same covariance matrix but different means. Under \mathbf{H}_0 the data (background clutter) is modeled as $\mathcal{N}(0, C_b)$ and under \mathbf{H}_1 it is modeled as $\mathcal{N}(s, C_b)$. The background covariance C_b is estimated from the reference background clutter data. The estimated background covariance \hat{C}_b is given by

$$\hat{C}_b = \frac{1}{M} \sum_{i=1}^M (\mathbf{x}(i) - \hat{\boldsymbol{\mu}}_b)(\mathbf{x}(i) - \hat{\boldsymbol{\mu}}_b)^T, \quad (3)$$

where $\hat{\boldsymbol{\mu}}_b$ is the estimated background clutter sample mean given by

$$\hat{\boldsymbol{\mu}}_b = \frac{1}{M} \sum_{i=1}^M \mathbf{x}(i). \quad (4)$$

Assuming a single pixel target \mathbf{r} as the observation test vector, the expression for the RX-algorithm is given by

$$RX(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_b)^T \hat{C}_b^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_b). \quad (5)$$

3 Feature Space and Kernel Methods

Suppose the input hyperspectral data is represented by the data space ($\mathcal{X} \subseteq \mathcal{R}^J$) and \mathcal{F} be a feature space associated with \mathcal{X} by a nonlinear mapping function Φ

$$\begin{aligned} \Phi : \mathcal{X} &\rightarrow \mathcal{F}, \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}), \end{aligned} \quad (6)$$

where \mathbf{x} is an input vector in \mathcal{X} which is mapped into a potentially much higher dimensional feature space. Using the kernel trick (Equation 7), it allows us to implicitly compute

the dot products in \mathcal{F} without mapping the input vectors into \mathcal{F} ; therefore, in the kernel methods, the mapping Φ does not need to be identified. The kernel representation for the monomial dot products in \mathcal{F} is expressed as

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle \\ &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \end{aligned} \quad (7)$$

Equation 7 shows that the dot products in \mathcal{F} can be avoided and replaced by a kernel, a nonlinear function which can be easily calculated without identifying the nonlinear map Φ . Two commonly used kernels are the Gaussian RBF kernel: $k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{c})$ and Polynomial kernel: $((\mathbf{x} \cdot \mathbf{y}) + \theta)^d$.

4 Kernel RX-Algorithm

In this section, we remodel the RX-algorithm in the feature space by assuming the input data has already been mapped into a high dimensional feature space. The two hypotheses in the nonlinear domain are now

$$\begin{aligned} \mathbf{H}_{0_\Phi} : \Phi(\mathbf{x}) &= \Phi(\mathbf{n}), & \text{Target absent} \\ \mathbf{H}_{1_\Phi} : \Phi(\mathbf{x}) &= a_\Phi \Phi(\mathbf{s}) + \Phi(\mathbf{n}), & \text{Target present} \end{aligned} \quad (8)$$

The corresponding RX-algorithm in the feature space is

$$RX(\Phi(\mathbf{r})) = (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi})^T \hat{C}_{b_\Phi}^{-1} (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi}) \quad (9)$$

where \hat{C}_{b_Φ} and $\hat{\boldsymbol{\mu}}_{b_\Phi}$ are the estimated covariance and background clutter sample mean in the feature space, respectively, given by

$$\hat{C}_{b_\Phi} = \frac{1}{M} \sum_{i=1}^M (\Phi(\mathbf{x}(i)) - \hat{\boldsymbol{\mu}}_{b_\Phi})(\Phi(\mathbf{x}(i)) - \hat{\boldsymbol{\mu}}_{b_\Phi})^T \quad (10)$$

and

$$\hat{\boldsymbol{\mu}}_{b_\Phi} = \frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}(i)). \quad (11)$$

The nonlinear RX-algorithm given by Equation (9) is now in the feature space which cannot be implemented explicitly due to the non-linear mapping Φ which produces a data space of high dimensionality. In order to avoid implementing Equation (9) directly we need to kernelize (9) by using the kernel trick introduced in Section 3.

The estimated background covariance matrix can be represented by its eigenvector decomposition or spectral decomposition as given by

$$\hat{C}_{b_\Phi} = \mathbf{V}_\Phi \Lambda_b \mathbf{V}_\Phi^T, \quad (12)$$

where Λ_b is a diagonal matrix consisting of the eigenvalues and \mathbf{V}_Φ is a matrix whose columns are the eigenvectors of \mathbf{C}_{b_Φ} in the feature space. The eigenvector matrix \mathbf{V}_Φ is given by

$$\mathbf{V}_\Phi = [\mathbf{v}_\Phi^1, \mathbf{v}_\Phi^2, \dots], \quad (13)$$

where \mathbf{v}_Φ^j is the j th eigenvector with non-zero eigenvalue.

The pseudoinverse of the estimated background covariance matrix can also be written as

$$\hat{\mathbf{C}}_{b_\Phi}^\# = \mathbf{V}_\Phi \Lambda_b^{-1} \mathbf{V}_\Phi^T. \quad (14)$$

Each eigenvector \mathbf{v}_Φ^j in the feature space can be expressed as a linear combination of the centered input vectors $\Phi_c(\mathbf{x}(i)) = \Phi(\mathbf{x}(i)) - \hat{\boldsymbol{\mu}}_{b_\Phi}$ in the feature space as shown by

$$\mathbf{v}_\Phi^j = \sum_{i=1}^M \beta_i^j \Phi_c(\mathbf{x}(i)) = \mathbf{X}_{b_\Phi} \boldsymbol{\beta}^j, \quad (15)$$

where $\mathbf{X}_{b_\Phi} = [\Phi_c(\mathbf{x}(1)) \ \Phi_c(\mathbf{x}(2)) \ \dots \ \Phi_c(\mathbf{x}(M))]$ and for all the eigenvectors

$$\mathbf{V}_\Phi = \mathbf{X}_{b_\Phi} \mathcal{B}, \quad (16)$$

where $\boldsymbol{\beta}^j = (\beta_1^j, \beta_2^j, \dots, \beta_M^j)^T$ and $\mathcal{B} = (\boldsymbol{\beta}^1, \boldsymbol{\beta}^2, \dots, \boldsymbol{\beta}^M)^T$ are shown in [12] to be the eigenvectors of the kernel matrix (Gram matrix) $\mathbf{K}(\mathbf{X}_b, \mathbf{X}_b)$ normalized by the square root of their corresponding eigenvalues.

Substituting Equation (16) into (14) yields

$$\hat{\mathbf{C}}_{b_\Phi}^{-1} = \mathbf{X}_{b_\Phi} \mathcal{B} \Lambda_b^{-1} \mathcal{B}^T \mathbf{X}_{b_\Phi}^T. \quad (17)$$

Inserting Equation (17) into (9) the nonlinear RX-algorithm can be rewritten as

$$\begin{aligned} RX(\Phi(\mathbf{r})) & \\ &= (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi})^T \mathbf{X}_{b_\Phi} \mathcal{B} \Lambda_b^{-1} \mathcal{B}^T \mathbf{X}_{b_\Phi}^T (\Phi(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi}). \end{aligned} \quad (18)$$

The dot product terms $\Phi(\mathbf{r})^T \mathbf{X}_{b_\Phi}$ in the feature space can be represented in terms of the kernel function:

$$\begin{aligned} \Phi(\mathbf{r})^T \mathbf{X}_{b_\Phi} & \\ &= \Phi(\mathbf{r})^T ([\Phi(\mathbf{x}(1)) \ \Phi(\mathbf{x}(2)) \ \dots \ \Phi(\mathbf{x}(M))]) \\ &\quad - \frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}(i)) \\ &= (k(\mathbf{x}(1), \mathbf{r}) \ k(\mathbf{x}(2), \mathbf{r}) \ \dots \ k(\mathbf{x}(M), \mathbf{r})) \\ &\quad - \frac{1}{M} \sum_{i=1}^M k(\mathbf{x}(i), \mathbf{r}) \\ &= \mathbf{k}(\mathbf{X}_b, \mathbf{r})^T - \frac{1}{M} \sum_{i=1}^M k(\mathbf{x}(i), \mathbf{r}) \equiv \mathbf{K}_r^T, \end{aligned} \quad (19)$$

where $\mathbf{k}(\mathbf{X}_b, \mathbf{r})^T$ represents a vector whose entries are the kernels $k(\mathbf{x}(i), \mathbf{r}), i = 1 \dots M$, and $\frac{1}{M} \sum_{i=1}^M k(\mathbf{x}(i), \mathbf{r})$ represents the scalar mean of $\mathbf{k}(\mathbf{X}_b, \mathbf{r})^T$. Similarly,

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{b_\Phi}^T \mathbf{X}_{b_\Phi} & \\ &= \frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}(i))^T \{[\Phi(\mathbf{x}(1)) \ \Phi(\mathbf{x}(2)) \ \dots \ \Phi(\mathbf{x}(M))]\} \\ &\quad - \frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}(i)) \\ &= \frac{1}{M} \sum_{i=1}^M (k(\mathbf{x}(i), \mathbf{x}(1)) \ k(\mathbf{x}(i), \mathbf{x}(2)) \ \dots \ k(\mathbf{x}(i), \mathbf{x}(M))) \\ &\quad - \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M k(\mathbf{x}(i), \mathbf{x}(j)) \\ &= \frac{1}{M} \sum_{i=1}^M \mathbf{k}(\mathbf{x}(i), \mathbf{X}_b) - \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M k(\mathbf{x}(i), \mathbf{x}(j)) \\ &\equiv \mathbf{K}_{\hat{\boldsymbol{\mu}}_b}^T. \end{aligned} \quad (20)$$

Also using the properties of the Kernel PCA [12], as shown in Appendix I, we have the relationship

$$\hat{\mathbf{K}}_b^{-1} = \frac{1}{M} \mathcal{B} \Lambda_b^{-1} \mathcal{B}^T, \quad (21)$$

where we denote the estimated centered Gram matrix $\hat{\mathbf{K}}_b = \hat{\mathbf{K}}(\mathbf{X}_b, \mathbf{X}_b) = (\hat{\mathbf{K}})_{ij}$ the $M \times M$ kernel matrix whose entries $k(\mathbf{x}_i, \mathbf{x}_j)$ are the dot products $\langle \Phi_c(\mathbf{x}_i), \Phi_c(\mathbf{x}_j) \rangle$ and M is the total number of background clutter samples which can be ignored. Substituting (19), (20), and (21) (without $\frac{1}{M}$) into (18) the kernelized version of the RX-algorithm is given by

$$RX_{\mathbf{K}}(\mathbf{r}) = (\mathbf{K}_r^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_b}^T)^T \hat{\mathbf{K}}_b^{-1} (\mathbf{K}_r^T - \mathbf{K}_{\hat{\boldsymbol{\mu}}_b}^T) \quad (22)$$

which can now be implemented with no knowledge of the mapping function Φ . The only requirement is a good choice for the kernel function k . Note that $\hat{\mathbf{K}}_b$ is the centered Gram matrix, as shown in [8]. The centered $\hat{\mathbf{K}}_b$ is given by

$$\hat{\mathbf{K}}_b = (\mathbf{K}_b - \mathbf{1}_N \mathbf{K}_b - \mathbf{K}_b \mathbf{1}_N + \mathbf{1}_N \mathbf{K}_b \mathbf{1}_N), \quad (23)$$

where \mathbf{K}_b is the Gram matrix before centering and the elements of the $N \times N$ matrix $(\mathbf{1}_N)_{ij} = 1/N$.

5 Simulation Results

In this section, we apply both the kernel RX- and conventional RX-algorithms to two HYDICE images – the Forest Radiance I (FR-I) image and the Desert Radiance II (DR-II) image – and the hyperspectral mine image, as shown in Fig. 1. FR-I includes total 14 targets and DR-II contains

6 targets along the road; all the targets are military vehicles. The hyperspectral mine image contains a total of 33 surface mines. A HYDICE imaging sensor generates 210 bands across the whole spectral range ($0.4 - 2.5 \mu m$), but we only use 150 bands by discarding water absorption and low signal to noise ratio (SNR) bands; the bands used are the 23rd–101st, 109th–136th, and 152nd–194th. The hyperspectral mine image consists of 70 bands whose spectral range spans $8 - 11.5 \mu m$.

Gaussian RBF kernel, $\mathbf{k}(\mathbf{x}, \mathbf{y}) = \exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{c})$, was used to implement the kernel RX-algorithm; the value of c was set to 40. All the pixel vectors in the test image are first normalized by a constant, which is a maximum value obtained from all the spectral components of the spectral vectors in the corresponding test image, so that the entries of the normalized pixel vectors fit into the interval of spectral values between zero and one. The rescaling of pixel vectors was mainly performed to effectively utilize the dynamic range of Gaussian RBF kernel.

The kernel matrix $\hat{\mathbf{K}}_b$ can be estimated either globally or locally. The global estimation must be performed prior to detection and normally needs a large amount of data samples to successfully represent all the background types present in a given data set. In this paper, to globally estimate $\hat{\mathbf{K}}_b$ we need to use all the spectral vectors in a given test image. A well-known data clustering algorithm, k -means [15], is used on all the spectral vectors in order to generate a significantly less number of spectral vectors (centroids) from which $\hat{\mathbf{K}}_b$ is estimated. By using a small number of distinct background spectral vectors a manageable kernel matrix is generated where a more efficient kernel RX-algorithm is now implemented. The number of the representative spectral vectors obtained from the k -means procedure was set to 600, which means the number of centroids generated by the k -means was 600.

For local estimation of $\hat{\mathbf{K}}_b$ we use local background samples, which are from the neighboring area of the pixel being tested. For each test pixel location, a dual concentric rectangular window is used to separate a local area into two regions – the inner-window region (IWR) and the outer-window region (OWR), as shown in Fig. 2; the local kernel matrix and the background covariance matrix are calculated from the pixel vectors in the OWR. The test pixel vector \mathbf{r} was obtained from the IWR.

The dual concentric windows naturally divide the local area into the potential target region – the IWR – and the background region – the OWR – whose local statistics in the original and nonlinear feature domain are compared using the conventional RX- and kernel RX- algorithms, respectively. The size of the IWR is set to enclose targets to be detected whose approximate size is based on prior knowledge of the range, field of view (FOV), and the dimension of the biggest target in the given data set. Similarly, the size of the OWR is set to include sufficient statistics from

the neighboring background. The size for the dual windows used were 5×5 and 13×13 pixel areas, respectively. The size of the OWR was set to include a sufficient number of spectral vectors to generate the kernel matrix $\hat{\mathbf{K}}_b$.

Figs. 3, 4, and 5 show the anomaly detection results of both the kernel RX and the conventional RX using the local dual window applied to the FR-I and DR-II images and the hyperspectral mine image, respectively. The kernel RX detected most of the targets and mines with a few false alarms while the conventional RX generated much more false alarms and missed some targets; especially, in the case of FR-I the conventional RX missed 7 successive targets from the left. For both the HYDICE images and the mine image the kernel RX showed significantly improved performance over the conventional RX.

Figs. 6 and 7 show the ROC curves for the detection results for FR-I and DR-II images, as shown in Figs. 3 and 4, using the kernel RX and the conventional RX based on the local dual window. Figs. 6 and 7 also include the ROC curves for the kernel RX based on the global kernel matrix. The global method for the kernel RX provided slightly improved performance over the local method for the HYDICE images that were tested. Fig. 8 shows the the ROC curves for the detection results for the hyperspectral mine image, as shown in Fig. 5, using the kernel RX and the conventional RX based on the local dual window. Note that the kernel RX significantly outperformed the conventional RX at lower false alarm rates.

6 Conclusions

We have extended the RX-algorithm to a nonlinear feature space by kernelizing the corresponding nonlinear GLRT expression. The GLRT expression of the kernel RX is similar to the conventional RX, but every term in the expression is in kernel forms which can be readily calculated in terms of the input data in the original space. The kernel RX showed superior detection performance over the conventional RX given the HYDICE images tested. This is mainly because the high order correlations between the spectral bands are exploited by the kernel RX.

Appendix I

In this Appendix derivation of Kernel PCA and its properties providing the relationship between the covariance matrix and the corresponding Gram matrix are presented. Our goal here is to prove expression (21). To drive the Kernel PCA consider the background clutter covariance matrix in feature space for the centered data $\mathbf{X}_{b_g} =$

$$\begin{bmatrix} \Phi_c(\mathbf{x}_1) & \Phi_c(\mathbf{x}_2) & \dots & \Phi_c(\mathbf{x}_M) \end{bmatrix} \\ \hat{\mathbf{C}}_{b_\Phi} = \mathbf{X}_{b_\Phi} \mathbf{X}_{b_\Phi}^T. \quad (24)$$

The PCA eigenvectors are computed by solving the eigenvalue problem

$$\begin{aligned} \lambda \mathbf{v}_\Phi &= \mathbf{C}_{b_\Phi} \mathbf{v}_\Phi \\ &= \frac{1}{M} \sum_{i=1}^M \Phi_c(\mathbf{x}_i) \Phi_c(\mathbf{x}_i)^T \mathbf{v}_\Phi \\ &= \frac{1}{M} \sum_{i=1}^M \langle \Phi_c(\mathbf{x}_i), \mathbf{v}_\Phi \rangle \Phi_c(\mathbf{x}_i). \end{aligned} \quad (25)$$

where \mathbf{v}_Φ is an eigenvector in \mathcal{F} with a corresponding nonzero eigenvalue λ . Equation (25) indicates that any eigenvector \mathbf{v}_Φ with corresponding $\lambda \neq 0$ are spanned by the input data $\Phi_c(\mathbf{x}_1), \dots, \Phi_c(\mathbf{x}_M)$ - i.e.

$$\mathbf{v}_\Phi = \sum_{i=1}^M \beta_i \Phi_c(\mathbf{x}_i) = \mathbf{X}_{b_\Phi} \boldsymbol{\beta}, \quad (26)$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_M)^T$. Substituting (26) into (25) and multiplying with $\Phi_c(\mathbf{x}_n)^T$, $n = 1, \dots, M$, yields

$$\begin{aligned} \lambda \sum_{i=1}^M \beta_i \langle \Phi_c(\mathbf{x}_n), \Phi_c(\mathbf{x}_i) \rangle \\ &= \frac{1}{M} \sum_{i=1}^M \beta_i \Phi_c(\mathbf{x}_n) \Phi_c(\mathbf{x}_i) \Phi_c(\mathbf{x}_i)^T \sum_{i=1}^M \Phi_c(\mathbf{x}_i) \\ &= \frac{1}{M} \sum_{i=1}^M \beta_i \langle \Phi_c(\mathbf{x}_n), \sum_{j=1}^M \Phi_c(\mathbf{x}_j) \rangle \langle \Phi_c(\mathbf{x}_j), \Phi_c(\mathbf{x}_i) \rangle, \end{aligned} \quad (27)$$

for all $n = 1, \dots, M$.

We denote by $\mathbf{K}_b = \mathbf{K}(X_b, X_b) = (\mathbf{K})_{ij}$ the $M \times M$ kernel (Gram) matrix whose entries are the dot products $\langle \Phi_c(\mathbf{x}_i), \Phi_c(\mathbf{x}_j) \rangle$. Equation (25) can now be rewritten as

$$M \lambda \boldsymbol{\beta} = \mathbf{K}_b \boldsymbol{\beta}, \quad (28)$$

where $\boldsymbol{\beta}$ turn out to be the eigenvectors with nonzero eigenvalues of the kernel matrix \mathbf{K}_b , as shown in [12]. Note that each $\boldsymbol{\beta}$ need to be normalized by the square root of its corresponding eigenvalue.

Furthermore, we assumed that the data was centered in the feature space, however, we cannot center the data in the high dimensional feature space because we do not have any knowledge about the non-linear mapping Φ . Therefore, we have to start with the original uncentered data and the resulting Gram matrix \mathbf{K}_b needs to be properly centered. As shown in [12], the centered Gram matrix $\hat{\mathbf{K}}_b$ can be obtained from the uncentered Gram Matrix \mathbf{K}_b by

$$\hat{\mathbf{K}}_b = (\mathbf{K}_b - \mathbf{1}_M \mathbf{K}_b - \mathbf{K}_b \mathbf{1}_M + \mathbf{1}_M \mathbf{K}_b \mathbf{1}_M), \quad (29)$$

where $(\mathbf{1}_M)_{ij} = 1/M$ is an $M \times M$ matrix. From the definition of PCA in the feature space (25) and the Kernel PCA (28) we can now write the eigenvector decomposition of the background covariance matrix and Gram matrix as

$$\hat{\mathbf{C}}_{b_\Phi} = \mathbf{V}_\Phi \Lambda_b \mathbf{V}_\Phi^T \quad (30)$$

and

$$\hat{\mathbf{K}}_b = \mathcal{B} \Omega_{\mathbf{K}_b} \mathcal{B}^T, \quad (31)$$

respectively. Using pseudoinverse matrix properties [16] the pseudoinverse background covariance matrix $\hat{\mathbf{C}}_{b_\Phi}^\#$ and inverse Gram matrix $\hat{\mathbf{K}}_b^{-1}$ can also be written as

$$\hat{\mathbf{C}}_{b_\Phi}^\# = \mathbf{V}_\Phi \Lambda_b^{-1} \mathbf{V}_\Phi^T \quad (32)$$

and

$$\hat{\mathbf{K}}_b^{-1} = \mathcal{B} \Omega_{\mathbf{K}_b}^{-1} \mathcal{B}^T, \quad (33)$$

respectively. From the relationship between the eigenvalues of covariance matrix in the feature space and the Gram matrix described in (28)

$$\Lambda_b = \frac{1}{M} \Omega_{\mathbf{K}_b} \quad (34)$$

where Λ_b is a diagonal matrix with its diagonal elements being the eigenvalues of $\hat{\mathbf{C}}_{b_\Phi}$ and $\Omega_{\mathbf{K}_b}$ is a diagonal matrix with diagonal values equal to the eigenvalues of the Gram matrix $\hat{\mathbf{K}}_b$. Substituting (34) into (33) we obtain the relationship

$$\hat{\mathbf{K}}_b^{-1} = \frac{1}{M} \mathcal{B} \Lambda_b^{-1} \mathcal{B}^T \quad (35)$$

where M is a constant representing the total number of background clutter samples which can be ignored.

References

- [1] D. W. J. Stein, S. G. Beaven, L. E. Hoff, . Winter, E. M, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Processing Mag.*, vol. 19, pp. 58–69, 2002.
- [2] D. W. J. Stein, "Stochastic compositional models applied to subpixel analysis of hyperspectral imagery," in *Proc. SPIE*, July 2001, vol. 4480, pp. 49–56.
- [3] H. Kwon, S. Z. Der, and N. M. Nasrabadi, "Adaptive anomaly detection using subspace separation for hyperspectral images," *Optical Engineering*, vol. 42, no. 11, pp. 3342–3351, Nov. 2003.
- [4] C.-I. Chang and S.-S. Chiang, "Anomaly detection and classification for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 40, no. 6, pp. 1314–1325, June 2002.

- [5] X. Yu and I. S. Reed, "Comparative performance analysis of adaptive multispectral detectors," *IEEE Trans. Signal Process.*, vol. 41, no. 8, pp. 2639–2656, 1993.
- [6] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoustics, Speech and Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.
- [7] D. Manolakis and G Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 29–43, Jan. 2002.
- [8] B Schölkopf and A. J. Smola, *Learning with Kernels*, The MIT Press, 2002.
- [9] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithm," *IEEE Trans. Neural Networks.*, vol. 14, no. 1, pp. 117–126, 2003.
- [10] Paclik P. Pekalska, E. and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *J. of Machine Learning*, vol. 2, pp. 175–211, 2001.
- [11] A. Ruiz and E. Lopez-de Teruel, "Nonlinear kernel-based statistical pattern analysis," *IEEE Trans. Neural Networks.*, vol. 12, pp. 16–32, 2001.
- [12] B Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," *Neural Computation*, , no. 10, pp. 1299–1319, 1999.
- [13] G. Baudat and F Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, , no. 12, pp. 2385–2404, 2000.
- [14] M. Girolami, "Mercer kernel-based clustering in feature space," *IEEE Trans. Neural Networks.*, vol. 13, no. 3, pp. 780–784, 2002.
- [15] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [16] G. Strang, *Linear algebra and its applications*, Harcourt Brace & Company, 1986.

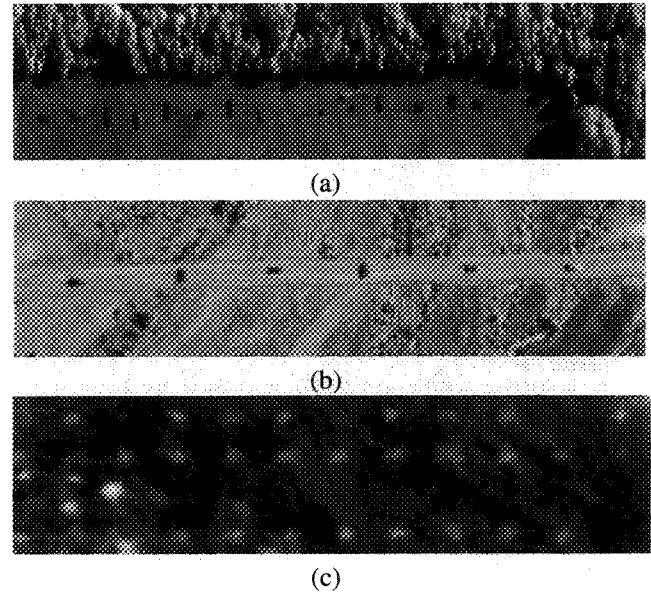


Figure 1: Sample band images (48th) from HYDICE images and mine image. (a) the Forest Radiance I image, (b) the Desert Radiance II image and (c) the hyperspectral mine image.

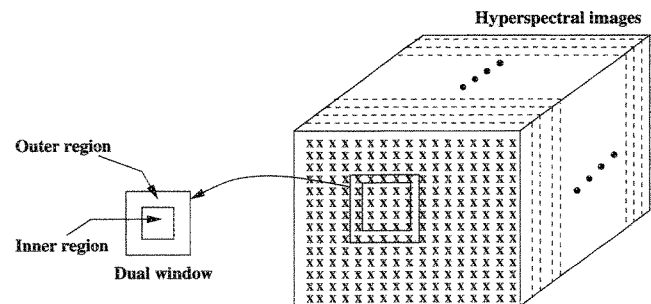


Figure 2: Example of the dual concentric windows in the hyperspectral images.

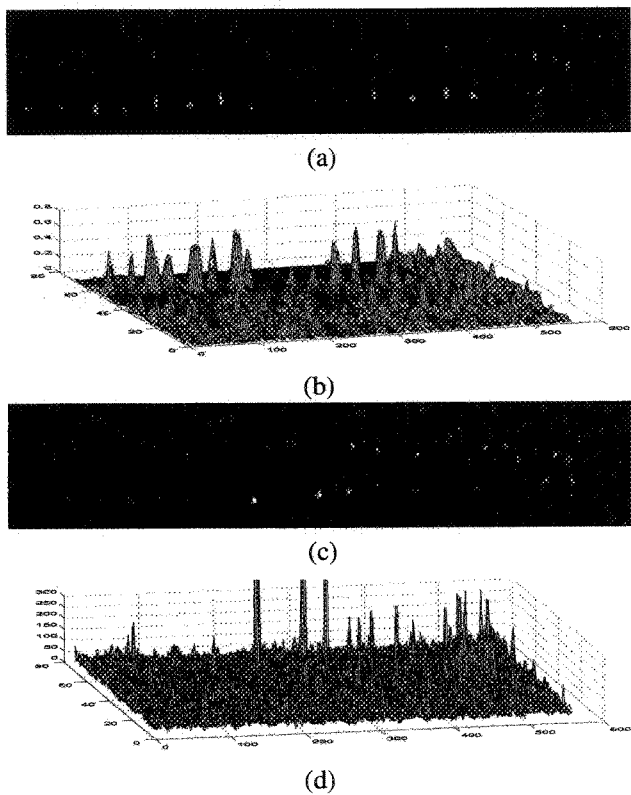


Figure 3: Detection results for the Forest Radiance I image using the kernel RX-algorithm and conventional RX-algorithm based on the local dual window. (a) Kernel RX, (b) 3-D plot of (a), (c) RX, and (d) 3-D plot of (c).

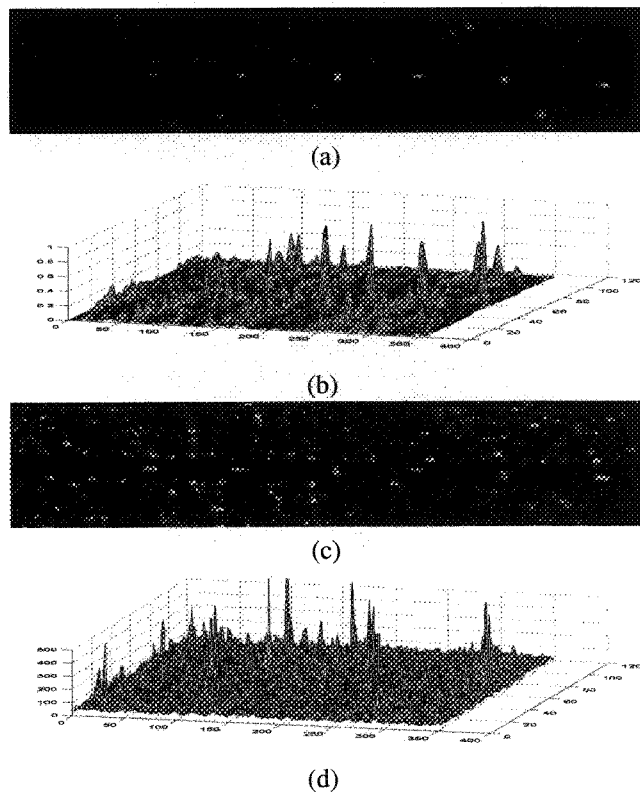


Figure 4: Detection results for the Desert Radiance II image using the kernel RX-algorithm and conventional RX-algorithm based on the local dual window. (a) Kernel RX, (b) 3-D plot of (a), (c) RX, and (d) 3-D plot of (c).

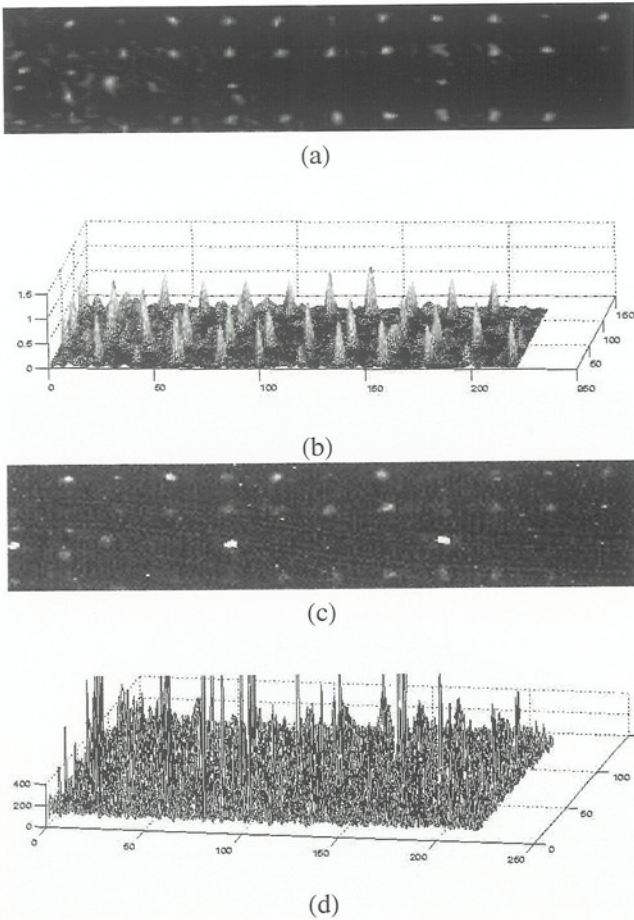


Figure 5: Detection results for the mine image using the kernel RX-algorithm and conventional RX-algorithm based on the local dual window. (a) Kernel RX, (b) 3-D plot of (a), (c) RX, (d) 3-D plot of (c).

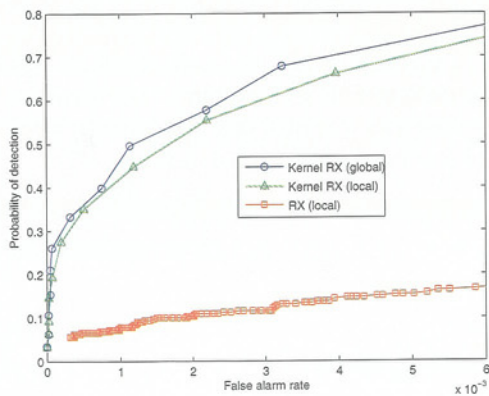


Figure 6: ROC curves obtained by the kernel RX-algorithm based on the global and local kernel matrices and the conventional RX-algorithm based on the local covariance matrix for the Forest Radiance I image.

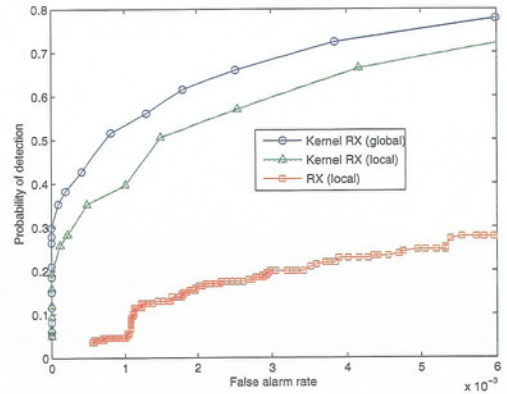


Figure 7: ROC curves obtained by the kernel RX-algorithm based on the global and local kernel matrices and the conventional RX-algorithm based on the local covariance matrix for the Desert Radiance II image.

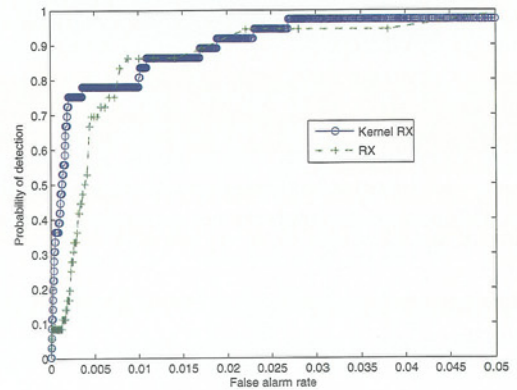


Figure 8: ROC curves obtained by the kernel RX-algorithm and the conventional RX-algorithm based on the local dual window for the hyperspectral mine image.