HYPERSPECTRAL ANOMALY DETECTION USING KERNEL RX-ALGORITHM

Heesung Kwon and Nasser M. Nasrabadi

U.S. Army Research Laboratory, 2800 Powder Mill Rd., Adelphi, MD 20783-1197

Abstract

In this paper we present a nonlinear version of the well-known anomaly detection method, referred to as the RX-algorithm, by extending this algorithm in a feature space associated with the original input space via a certain nonlinear mapping function. An expression for the nonlinear form of the RX-algorithm is derived which is basically intractable mainly due to the high dimensionality of the feature space. We convert the nonlinear RX expression into kernels which implicitly compute dot products in the nonlinear domain. The proposed kernel RX-algorithm is applied to hyperspectral images for anomaly detection. Improved performance of the kernel RX over the conventional RX is shown for the HYDICE (HYperspectral Digital Imagery Collection Experiment) images tested.

1. INTRODUCTION

Anomaly detectors are pattern recognition schemes that are used to detect objects that might be of military interest. Almost all 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 significant different spectral signature from their neighboring background clutter pixels are identified as spectral anomalies. Anomaly detection algorithms [1, 2, 3, 4] use the spatial as well as 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. However, most of the algorithms in the literature assume that the data has a Gaussian probability distribution and the generalized likelihood ratio test (GLRT) is used to test the hypotheses that a target is in the image.

In [2], an spectral anomaly detection was developed for detecting targets of unknown spectral distribution against a background with unknown spectral covariance. This algorithm is now commonly referred to as the RX anomaly detector which has been successfully applied to many hyperspectral target detection applications [3, 4]. It is now considered as the benchmark anomaly detection algorithm for multispectral/hyperspectral data. The RX-algorithm is a constant false alarm rate (CFAR) adaptive anomaly detector which is derived from the generalized likelihood ratio test.

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 [5].

In fact, using kernel functions no explicit knowledge of the actual nonlinear mapping is necessary which means the RX-algorithm is not computed explicitly in the feature space. This property is the major advantage of the kernel-based methods that reduce a nonlinear algorithm to a linear one in some high-dimensional feature space. Kernel-based versions of a number of feature extraction or pattern recognition algorithms have recently been proposed [6, 7, 8].

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 [2] developed a GLR test, so called the RX anomaly detection, for multidimensional image data assuming 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 $\mathbf{x}(n) = (x_1(n), x_2(n), \ldots, x_J(n))^T$ 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 $\mathbf{X}_b$

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

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

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

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 $\mathbf{s} = [s_1, s_2, \ldots, s_J]$. 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

Correspondence:Email:nnasraba@arl.army.mil

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

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

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

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

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

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

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

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

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 monomial 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

$$\begin{aligned} k(\mathbf{x}_i, \mathbf{x}_j) &= < \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) > \\ &= \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j). \end{aligned} \tag{7}$$

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

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

The corresponding RX-algorithm in the feature space is

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

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

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

and

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

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

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

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

$$\mathbf{V}_\Phi = [\mathbf{v}_\Phi^1, \mathbf{v}_\Phi^2, \ldots], \tag{13}$$

where $\mathbf{v}_\Phi^j$ is the jth eigenvector with non-zero eigenvalue.

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

$$\hat{C}_{b_\Phi}^{-1} = \mathbf{V}_\Phi \Lambda_b^{-1} {\mathbf{V}_\Phi}^T. \tag{14}$$

Each eigenvector $\mathbf{v}_\phi^j$ 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

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

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

$$\mathbf{V}_\Phi = \mathbf{X}_{b_\Phi}\mathcal{B}, \tag{16}$$

where $\boldsymbol{\beta}^j = (\beta_1^j, \beta_2^j, \ldots, \beta_M^j)^T$ and $\mathcal{B} = (\boldsymbol{\beta}^1, \boldsymbol{\beta}^2, \ldots, \boldsymbol{\beta}^M)^T$ are shown in [6] 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

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

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

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

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:

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

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,

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

Also using the properties of the Kernel PCA [6], we have the relationship

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

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 $< \Phi_c(\mathbf{x}_i),\Phi_c(\mathbf{x}_j) >$ 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

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

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 [5]. The centered $\hat{\mathbf{K}}_b$ is given by

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

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 the HYDICE image - the Forest Radiance I (FR-I) image, as shown in Fig. 1 - that includes total 14 targets (military vehicles). The HYDICE imaging sensors generate 210 band images across the whole spectral range (0.4 – 2.5 μm), but we use only 150 band images by discarding water absorption and low signal to noise ratio (SNR) bands; the bands used are the 23rd–101st, 109th–136th, and 152nd–194th.

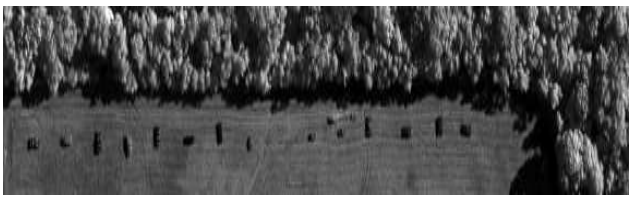


Fig. 1. Sample band image (48th) from the Forest Radiance I image.

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.

For each pixel location in the test image, 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 background kernel matrix $\hat{\mathbf{K}}_b$, the covariance matrix $\hat{\mathbf{C}}_b$, and the background mean $\hat{\boldsymbol{\mu}}_b$ were obtained locally based on the pixel vectors in the OWR. The test

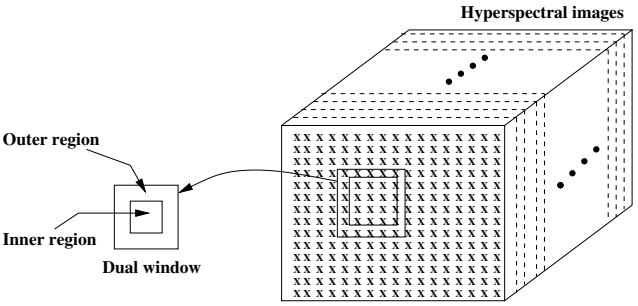
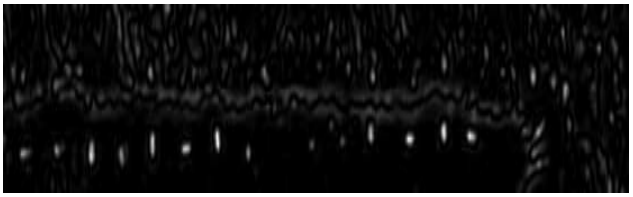


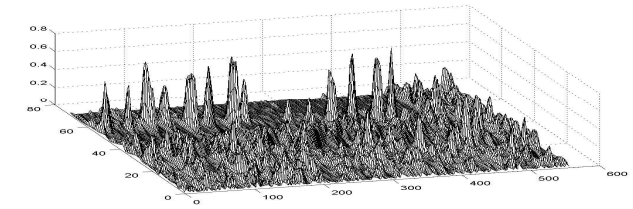
Fig. 2. Example of the dual concentric windows.

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. The size of the IWR was set to enclose targets to be detected. The target size is approximated using the 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 was set to include the neighboring background. The size for the dual windows used were 5×5 and 13×13 pixel areas, respectively.

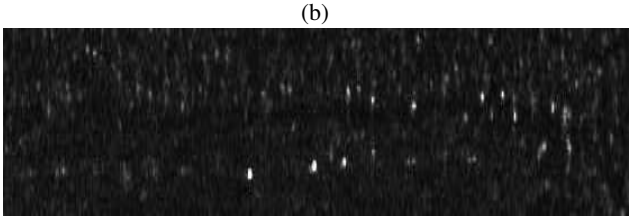
Fig. 3 shows the detection results. The kernel RX detected most of the targets with a few false alarms while the conventional RX generated more false alarms and missed some targets. The receiver operating characteristics (ROC) curves representing detection probability P_d versus false alarm rates N_f, were also generated to provide quantitative performance comparison between the two algorithms, as shown in Fig. 4: the kernel RX showed superior detection performance over the conventional RX. Every target pixel inside the target regions is considered as a target candidate to be detected. P_d becomes one only when all the individual target pixels within a target are detected; perfect detection is, therefore, difficult to achieve and the values of P_d for both the kernel RX and conventional RX are usually less than one.



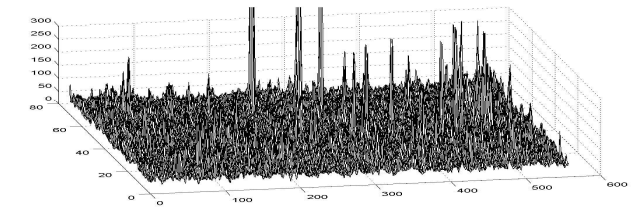
(a)



(b)



(c)



(d)

Fig. 3. Detection results for the Forest Radiance I images using kernel-RX and conventional RX. (a) Kernel-RX, (b) 3-D plot of (a), (c) RX, and (d) 3-D plot of (c).

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 image tested. This is mainly because the high order correlations between the spectral bands are exploited by the kernel RX.

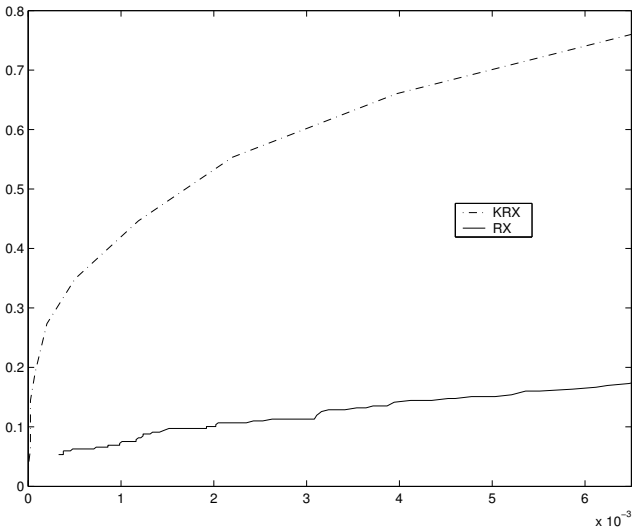


Fig. 4. ROC curves obtained by kernel-RX and RX for the Forest Radiance I image.

7. REFERENCES

[1] 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.

[2] 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.

[3] C.M. Stellman, G.G. Hazel, F. Bucholtz, J.V. Michalowicz, A. Stoker, and W. Scaaf, "Real-time hyperspectral detection and cuing," *Optical Engineering*, vol. 39, no. 7, pp. 1928–1935, July 2000.

[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] B Schökopf and A. J. Smola, *Learning with Kernels*, The MIT Press, 2002.

[6] B Schökopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," *Neural Computation*, , no. 10, pp. 1299–1319, 1999.

[7] G. Baudat and F Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Computation*, , no. 12, pp. 2385–2404, 2000.

[8] H. Kwon and N. M. Nasrabadi, "Hyperspectral target detection using kernel spectral matched filter," in *IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum*, July 2004, to appear.