KERNEL SPECTRAL MATCHED FILTER FOR HYPERSPECTRAL TARGET DETECTION

Nasser M. Nasrabadi and Heesung Kwon

US Army Research Laboratory 2800 Powder Mill Road, Adelphi, MD 20783

ABSTRACT

In this paper a kernel-based non-linear spectral matched filter is introduced for target detection in hyperspectral imagery. The proposed spectral matched filter is defined in a kernel feature space which is equivalent to a non-linear matched filter in the original input space. This non-linear spectral matched filter is based on the notion that performing matched filtering in the high dimensional feature space increases the separability of spectral data mainly because it exploits the higher order correlation between the spectral bands. It is also shown that the non-linear spectral matched filter can easily be implemented in terms of kernel functions using the so called *kernel trick* property of the *Mercer kernels*. The kernel version of the non-linear spectral matched filter is implemented and simulation results on hyperspectral imagery are shown to outperform the linear version.

1. INTRODUCTION

Target detection using linear matched filtering is a well known approach in detecting objects of interest in hyperspectral imagery [1]. However, the linear matched filter does not exploit the higher order statistical correlation between the spectral bands since it is only based on the second order statistics. The motivation behind designing the non-linear matched filter is to incorporate the higher order statistical correlation between the spectral bands in the design of the matched filter in order to improve the performance of the conventional linear matched filter.

Non-linear spectral matched filters can easily be developed by assuming a non-linear model where the input data is first converted into a high dimensional feature space by a certain non-linear mapping. However, to implement such a non-linear match filter in the feature space may not be computationally possible due to the high dimensionality of the feature space. Recently, using the ideas of kernel-based learning algorithms it has been shown in [2, 3, 4] that a number of linear algorithms can easily be extended to non-linear versions by implementing them in terms of kernel functions, thus avoiding the implementation of the algorithm in the feature space.

In this paper, we introduce a non-linear spectral matched filter in a kernel feature space and its corresponding kernel version. To convert a linear matched filter into a non-linear version, the matched filter problem is first formulated in a particular feature space by using a non-linear mapping which is associated with a kernel function. The matched filter expression in that feature space is then rewritten in terms of dot products and by using the so called kernel trick (see Eq. (8)), it is converted in terms of kernel functions. We refer to this process as kernelizing the expression for the non-linear matched filter and the resulting match filter is called the kernel spectral matched filter. This paper is organized as follows. Section 2 introduces the linear matched filter and the idea of kernel trick when using Mercer kernels is described in Section 3. In Section 4, non-linear matched filter is described which is reformulated in terms of the kernel function to obtain the kernel matched filter. Performance of the kernel matched filter on hyperspectral imagery is provided in Section 5 and conclusions are given in Section 6.

2. LINEAR MATCHED FILTER

Let the input spectral signal \mathbf{x} be $\mathbf{x} = [x(1), x(2), \dots, x(J)]^T$ consisting of J spectral bands. We can model each spectral observation as a linear combination of the target spectral signature and noise

$$\mathbf{x} = a\mathbf{s} + \mathbf{n},\tag{1}$$

where *a* is an attenuation constant (target abundance measure) when a = 0 no target is present and when a > 0 target is present, $\mathbf{s} = [s(1), s(2), \dots, s(J)]^T$ contains the spectral signature of the target and **n** contains the added background clutter noise.

We can design a linear matched filter $\mathbf{w} = [w(1), w(2), \dots, w(J)]^T$ such that the desired target signal **s** is passed through while the average filter output energy is minimized. Let us define **X** to be a $J \times N$ matrix of the N mean-removed (centered) independent reference pixels obtained from the input image. Let each centered observation spectral pixel to be represented as a column in the sample matrix $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N]$. The output of the filter for the input \mathbf{x}_i is given by

$$y(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i = \mathbf{x}_i^T \mathbf{w}.$$
 (2)

The average output power of the filter for the reference data ${\bf X}$ is given by

$$\frac{1}{N}\sum_{i=1}^{N}y(\mathbf{x}_{i})^{2} = \mathbf{w}^{T}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_{i}\mathbf{x}_{i}^{T})\mathbf{w} = \mathbf{w}^{T}\mathbf{C}\mathbf{w}, \quad (3)$$

where C is the covariance matrix of the reference data. The design of the matched filter w is equivalent to a constrained least squares minimization problem given by

$$\min_{\mathbf{w}} \{ \mathbf{w}^T \mathbf{C} \mathbf{w} \} \text{ subject to } \mathbf{s}^T \mathbf{w} = 1.$$
 (4)

The solution to this quadratic minimization problem is known as the minimum variance method in the adaptive array processing literature [5, 6], which is also called the constrained energy minimization (CEM) filter [7] in the hyperspectral literature. The solution for the filter is given by

$$\mathbf{w} = \frac{\mathbf{C}^{-1}\mathbf{s}}{\mathbf{s}^T \mathbf{C}^{-1}\mathbf{s}}.$$
 (5)

The covariance matrix C is usually estimated from the input image (the estimated covariance matrix for the mean-removed reference data $\hat{\mathbf{C}} = \frac{1}{N} \mathbf{X} \mathbf{X}^T$). The output of the linear matched filter for a test input \mathbf{r} , given the estimated covariance matrix, is given by

$$y(\mathbf{r}) = \mathbf{w}^T \mathbf{r} = \frac{\mathbf{s}^T \hat{\mathbf{C}}^{-1} \mathbf{r}}{\mathbf{s}^T \hat{\mathbf{C}}^{-1} \mathbf{s}}.$$
 (6)

3. KERNEL METHODS AND KERNEL TRICK

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

$$\Phi: \mathcal{X} \to \mathcal{F}, \mathbf{x} \mapsto \Phi(\mathbf{x}), \tag{7}$$

where \mathbf{x} is an input vector in \mathcal{X} which is mapped into a potentially much higher - (could be infinite) - dimensional feature space.

Implementing any linear algorithm (i.e., matched filter) in the feature space is equivalent to performing a nonlinear version of that algorithm (i.e., non-linear matched filter) in the original data space. Due to the high dimensionality of feature space \mathcal{F} it is computationally not feasible to implement the algorithm in the feature space. The kernel trick given by (8) is then used 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 [8]. The kernel representation for the dot products in \mathcal{F} is expressed as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$$
(8)

where k is a kernel function in terms of the original data. There are a large number of kernels called Mercer kernels that have the kernel trick property, see [8] for detailed information about the properties of different kernels and kernel-based learning.

4. NON-LINEAR MATCHED FILTER AND KERNEL MATCHED FILTER

In this section, we show how to extend the linear matched filter to a non-linear version. Formulation of the matched filter in the feature space and its kernelization are also shown.

4.1. Introduction to Non-linear Matched filter

Consider the linear model of the input data in a kernel feature space which is equivalent to a non-linear model in the input space

$$\Phi(\mathbf{x}) = a_{\Phi} \Phi(\mathbf{s}) + \mathbf{n}_{\Phi}, \tag{9}$$

where Φ is the non-linear mapping that implicitly maps the input data into a kernel feature space, a_{Φ} is an attenuation constant (target abundance measure), the high dimensional vector $\Phi(\mathbf{s})$ contains the spectral signature of the target in the feature space, and vector \mathbf{n}_{Φ} contains the added noise in the feature space.

Using the constrained least squares approach that was explained in the previous section it can easily be shown that the equivalent matched filter \mathbf{w}_{Φ} in the feature space is given by

$$\mathbf{w}_{\Phi} = \frac{\hat{\mathbf{C}}_{\Phi}^{-1}\Phi(\mathbf{s})}{\Phi(\mathbf{s})^{T}\hat{\mathbf{C}}_{\Phi}^{-1}\Phi(\mathbf{s})},\tag{10}$$

where $\hat{\mathbf{C}}_{\Phi}$ is the estimated covariance of pixels in the feature space. The estimated covariance is given by $\hat{\mathbf{C}}_{\Phi} = \mathbf{X}_{\Phi} \mathbf{X}_{\Phi}^{T}$ assuming the sample mean has already been removed from each sample (centered), where $\mathbf{X}_{\Phi} = [\Phi(\mathbf{x}_1) \ \Phi(\mathbf{x}_2) \ \dots \ \Phi(\mathbf{x}_N)]$ is a full rank matrix whose columns are the mapped input reference data in the feature space.

The matched filter in the feature space (10) is equivalent to a non-linear matched filter in the input space and its output for the input $\Phi(\mathbf{r})$ is given by

$$y(\Phi(\mathbf{r})) = \mathbf{w}_{\Phi}^{T} \Phi(\mathbf{r}) = \frac{\Phi(\mathbf{s})^{T} \hat{\mathbf{C}}_{\Phi}^{-1} \Phi(\mathbf{r})}{\Phi(\mathbf{s})^{T} \hat{\mathbf{C}}_{\Phi}^{-1} \Phi(\mathbf{s})}.$$
 (11)

Due to the high dimensionality of the feature space the expression (11) is not tractable. Therefore, it can not be directly implemented in the feature space. The expression (11) needs to be converted in terms of the kernel functions.

4.2. Kernel Matched Filter

In this subsection we show how to kernelize the matched filter in the feature space. The estimated background covariance matrix can be represented by its eigenvector decomposition or so called spectral decomposition given by

$$\hat{\mathbf{C}}_{\Phi} = \mathbf{V}_{\Phi} \Lambda \mathbf{V}_{\Phi}^{T}, \qquad (12)$$

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

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

where N is the maximum number of eigenvectors with non-zero eigenvalue.

The pseudoinverse of the estimated background covariance matrix can also be written in terms of its eigenvector decomposition as

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

Each eigenvector \mathbf{v}_{Φ}^{j} in the feature space, as shown in Appendix I, can be expressed as a linear combination of the input reference vectors in the feature space as shown by

$$\mathbf{v}_{\Phi}^{j} = \sum_{i=1}^{N} \beta_{i}^{j} \Phi(\mathbf{x}_{i}) = \mathbf{X}_{\Phi} \boldsymbol{\beta}^{j}, \qquad (15)$$

where $\beta^{j} = (\beta_{1}^{j}, \beta_{2}^{j}, \dots, \beta_{N}^{j})^{T}$ and for all the eigenvectors \mathbf{V}_{Φ} in the feature space

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

where $\mathcal{B} = [\beta^1 \ \beta^2 \ \dots \ \beta^N]$ are the eigenvectors of the kernel matrix (Gram) \mathbf{K} (see (21)) normalized by the square root of their corresponding eigenvalues, as shown in Appendix I. Substituting (16) into (14) yields

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

Inserting pseudoinverse (17) into (11) it can be rewritten as

$$y(\Phi(\mathbf{r})) = \frac{\Phi(\mathbf{s})^T \mathbf{X}_{\Phi} \mathcal{B} \Lambda^{-1} \mathcal{B}^T \mathbf{X}_{\Phi}^T \Phi(\mathbf{r})}{\Phi(\mathbf{s})^T \mathbf{X}_{\Phi} \mathcal{B} \Lambda^{-1} \mathcal{B}^T \mathbf{X}_{\Phi}^T \Phi(\mathbf{s})}.$$
(18)

The dot product term $\Phi(\mathbf{s})^T \mathbf{X}_{\Phi}$ in the feature space can be represented in terms of the kernel function which is referred to as its empirical kernel map [8]

$$\Phi(\mathbf{s})^T \mathbf{X}_{\Phi} = (\mathbf{k}(\mathbf{x}_1, \mathbf{s}), \mathbf{k}(\mathbf{x}_2, \mathbf{s}), \dots, \mathbf{k}(\mathbf{x}_N, \mathbf{s}))$$
(19)
= $\mathbf{K}^T (\mathbf{X}, \mathbf{s}) = \mathbf{K}_s^T$.

Similarly,

$$\Phi(\mathbf{r})^T \mathbf{X}_{\Phi} = (\mathbf{k}(\mathbf{x}_1, \mathbf{r}), \mathbf{k}(\mathbf{x}_2, \mathbf{r}), \dots, \mathbf{k}(\mathbf{x}_N, \mathbf{r}))$$
(20)
= $\mathbf{K}^T (\mathbf{X}, \mathbf{r}) = \mathbf{K}_{\mathbf{r}}^T$.

Also using the properties of the Kernel Principal Component Analysis (PCA) as shown in Appendix I, we have the relationship

$$\mathbf{K}^{-1} = \frac{1}{N} \mathcal{B} \Lambda^{-1} \mathcal{B}^{T}, \qquad (21)$$

where $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X}) = (\mathbf{K})_{ij}$ is an $N \times N$ kernel matrix whose entries are the dot products $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Substituting (19), (20), and (21) into (18) the kernelized version of the matched filter is given by

$$y(\mathbf{K}_{\mathbf{r}}) = \frac{\mathbf{K}(\mathbf{X}, \mathbf{s})^T \mathbf{K}^{-1} \mathbf{K}(\mathbf{X}, \mathbf{r})}{\mathbf{K}(\mathbf{X}, \mathbf{s})^T \mathbf{K}^{-1} \mathbf{K}(\mathbf{X}, \mathbf{s})} = \frac{\mathbf{K}_{\mathbf{s}}^T \mathbf{K}^{-1} \mathbf{K}_{\mathbf{r}}}{\mathbf{K}_{\mathbf{s}}^T \mathbf{K}^{-1} \mathbf{K}_{\mathbf{s}}}$$
(22)

which can now be implemented with no knowledge of the mapping function Φ .

Note that the kernel matrix \mathbf{K} and the empirical kernel maps $\mathbf{K}_{\mathbf{s}}^{T}$ and $\mathbf{K}_{\mathbf{r}}^{T}$ in (22) need to be properly centered because the sample mean cannot be directly removed in the feature space due to the high dimensionality of \mathcal{F} . The resulting centered $\hat{\mathbf{K}}$ is shown in [8] to be given by $\hat{\mathbf{K}} = (\mathbf{K} - \mathbf{1}_N \mathbf{K} - \mathbf{K} \mathbf{1}_N + \mathbf{1}_N \mathbf{K} \mathbf{1}_N)$, where the elements of the $N \times N$ matrix $(\mathbf{1}_N)_{ij} = 1/N$. The centered kernel matched filter output for (22) is now given by

$$y(\hat{\mathbf{K}}_{\mathbf{r}}) = \frac{\hat{\mathbf{K}}_{\mathbf{s}}^{T}\hat{\mathbf{K}}^{-1}\hat{\mathbf{K}}_{\mathbf{r}}}{\hat{\mathbf{K}}_{\mathbf{s}}^{T}\hat{\mathbf{K}}^{-1}\hat{\mathbf{K}}_{\mathbf{s}}},$$
(23)

where

$$\hat{\mathbf{K}}_{\mathbf{s}}^{T} = \mathbf{K}_{\mathbf{s}}^{T} - \frac{1}{N} \sum_{i=1}^{N} \mathbf{K}(\mathbf{x}_{i}, \mathbf{s})$$

and

$$\hat{\mathbf{K}}_{\mathbf{r}}^{T} = \mathbf{K}_{\mathbf{r}}^{T} - \frac{1}{N} \sum_{i=1}^{N} \mathbf{K}(\mathbf{x}_{i}, \mathbf{r}).$$

5. EXPERIMENTAL RESULTS

In this section, we implemented both the proposed kernel matched filter detector (KMFD) described by (23) and the conventional matched filter detector (MFD) described by (6) to detect targets of interest (military vehicles) in the HYDICE images. The 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 spectral bands used are the 23rd–101st, 109th–136th, and 152nd–194th.

We implemented KMFD with three different kernel functions, each kernel function being associated with its corresponding feature space. The three different kernels used were i) the Gaussian



Fig. 1. Sample band (48th) from the Desert Radiance II image.



Fig. 2. Detection results for the Desert Radiance II image using the kernel matched filter detectors (KMFD) and the matched filter detector (MFD). (a) KMFD with the Gaussian RBF kernel, (b) KMFD with the inverse multiquadric kernel, (c) KMFD with the polynomial kernel and (d) MFD in the original input domain.



Fig. 3. ROC curves obtained from the detection results for the Desert Radiance II image shown in Fig.2.

RBF kernel, $\exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{30})$, ii) inverse multiquadric kernel, $\frac{1}{\sqrt{\|\mathbf{x}-\mathbf{y}\|^2+1}}$, and iii) 5th order polynomial kernel, $((\mathbf{x} \cdot \mathbf{y}) + 1)^5$.

The HYDICE image from the Desert Radiance II data collection was used to test both the kernel-based and conventional matched filter detectors. The Desert Radiance II (DR-II) image contains 6 targets located in the dirt road, as shown in the sample spectral band in Fig. 1. The targets in the HYDICE image are all military vehicles. In the experiments the target spectral signature s is obtained by averaging the target samples collected from the right most target. The covariance matrix and kernel matrix were calculated from the randomly chosen background samples obtained from the given test image.

Figure 2 shows the detection results for the DR-II using KMFD with the three different kernels and MFD. The corresponding ROC curves for the detection results in Fig. 2 is shown in Fig. 3. KMFD with any choice of the three kernels could detect all the targets at a very low false alarm rate ($N_f \approx 3 \times 10^{-4}$), while conventional MFD detected all the targets at a much higher false alarm rate ($N_f \approx 3 \times 10^{-3}$).

6. CONCLUSIONS

We have extended the conventional matched filter detector to a nonlinear version by implicitly mapping the input data into a much higher dimensional feature space in order to make use of highorder nonlinear correlations between the spectral bands of a hyperspectral image. KMFD, the kernel counterpart of MFD, was implemented with several different kernels, each with different characteristics. In general, KMFD with all the kernels that were used showed a superior detection performance when compared to the conventional MFD for the HYDICE images tested in this paper.

7. APPENDIX I. (KERNEL PCA)

In this Appendix we present derivation of Kernel PCA and its properties providing the relationship between the covariance matrix and the corresponding Gram matrix. Our goal is to prove (21). The estimated covariance matrix for the centered input data in the feature space is given by $\hat{\mathbf{C}}_{\Phi} = \mathbf{X}_{\Phi} \mathbf{X}_{\Phi}^T$. The PCA eigenvectors are computed by solving the eigenvalue problem

$$\lambda \mathbf{v}_{\Phi} = \hat{\mathbf{C}}_{\Phi} \mathbf{v}_{\Phi} = \frac{1}{N} \sum_{i=1}^{N} \Phi(\mathbf{x}_{i}) \Phi(\mathbf{x}_{i})^{T} \mathbf{v}_{\Phi}$$
(24)
$$= \frac{1}{N} \sum_{i=1}^{N} \langle \Phi(\mathbf{x}_{i}), \mathbf{v}_{\Phi} \rangle \Phi(\mathbf{x}_{i}),$$

where \mathbf{v}_{Φ} is an eigenvector in \mathcal{F} with a corresponding nonzero eigenvalue λ . Eq. (24) indicates that each eigenvector \mathbf{v}_{Φ} with corresponding $\lambda \neq 0$ are spanned by $\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_N)$ – i.e.

$$\mathbf{v}_{\Phi} = \sum_{i=1}^{N} \beta_i \Phi(\mathbf{x}_i) = \mathbf{X}_{\Phi} \boldsymbol{\beta}, \qquad (25)$$

where $\mathbf{X}_{\Phi} = \begin{bmatrix} \Phi(\mathbf{x}_1) & \Phi(\mathbf{x}_2) & \dots & \Phi(\mathbf{x}_N) \end{bmatrix}$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_N)^T$. Substituting (25) into (24) and multiplying with $\Phi(\mathbf{x}_n)^T$, $n = 1, \dots, N$, yields

$$\lambda \sum_{i=1}^{N} \beta_{i} < \Phi(\mathbf{x}_{n}), \Phi(\mathbf{x}_{i}) >$$

$$= \frac{1}{N} \sum_{i=1}^{N} \beta_{i} \Phi(\mathbf{x}_{n}) \Phi(\mathbf{x}_{i}) \Phi(\mathbf{x}_{i})^{T} \sum_{i=1}^{N} \Phi(\mathbf{x}_{i})$$

$$= \frac{1}{N} \sum_{i=1}^{N} \beta_{i} < \Phi(\mathbf{x}_{n}), \sum_{j=1}^{N} \Phi(\mathbf{x}_{j}) < \Phi(\mathbf{x}_{j}), \Phi(\mathbf{x}_{i}) >>,$$
for all $n = 1, ..., N$.
$$(26)$$

We denote by $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X}) = (\mathbf{K})_{ij}$ the $N \times N$ kernel matrix whose entries are the dot products $\langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Eq. (26) can now be rewritten as

$$N\lambda\beta = \mathbf{K}\beta,\tag{27}$$

where β turn out to be the eigenvectors with nonzero eigenvalues of the kernel matrix **K**. Note that each β need to be normalized by the square root of its corresponding eigenvalue.

The kernel matrix eigen decomposition is given by

$$\mathbf{K} = \mathcal{B}\Omega \mathcal{B}^T, \tag{28}$$

where $\mathcal{B} = [\beta^1 \ \beta^2 \ \dots \ \beta^N]$ are the eigenvectors of the kernel matrix and Ω is a diagonal matrix with diagonal values equal to the eigenvalues of the kernel matrix **K**. The pseudoinverse of the estimated background covariance matrix $\hat{\mathbf{C}}^{\#}$ and inverse of the Gram matrix \mathbf{K}^{-1} can also be written as

$$\hat{\mathbf{C}}_{\Phi}^{\#} = \mathbf{V}_{\Phi} \Lambda^{-1} \mathbf{V}_{\Phi}^{T}$$
⁽²⁹⁾

and

$$\mathbf{K}^{-1} = \mathcal{B}\Omega^{-1}\mathcal{B}^T, \qquad (30)$$

respectively. The eigenvalues of the covariance matrix in the feature space and the eigenvalues of the kernel matrix are related by

$$\Lambda = \frac{1}{N}\Omega.$$
 (31)

Substituting (31) into (30) we obtain the relationship

$$\mathbf{K}^{-1} = \frac{1}{N} \mathcal{B} \Lambda^{-1} \mathcal{B}^T, \qquad (32)$$

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

8. REFERENCES

- D. Manolakis, G. Shaw, and N. Keshava, "Comparative analysis of hyperspectral adaptive matched filter detector," in *Proc. SPIE*, April 2000, vol. 4049, pp. 2–17.
- [2] A. Ruiz and E. Lopez-de Teruel, "Nonlinear kernel-based statistical patten analysis," *IEEE Trans. Neural Networks.*, vol. 12, pp. 16–32, 2001.
- [3] F. Abdallah, C. Richard, and R. Lengellé, "An improved training algorithm for nonlinear kernel discriminants," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2798–2806, Oct. 2004.
- [4] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm : A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. 43, no. 2, Feb. 2005.
- [5] D. H. John and D. E. Dudgeon, Array Signal Processing-Concepts and Techniques, Prentice Hall, 1993.
- [6] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, pp. 4–24, Apr. 1988.
- [7] J. C Harsanyi, Detection and Classification of Subpixel Spectral Signatures in Hyperspectral Image Sequences, Ph.D. dissertation, Dept. Elect. Eng., Univ. of Maryland, Baltimore County, 1993.
- [8] B Schökopf and A. J. Smola, *Learning with Kernels*, The MIT Press, 2002.