

# Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas

Mathieu Fauvel<sup>\*†</sup>, Jocelyn Chanussot<sup>\*</sup> and Jón Atli Benediktsson<sup>†</sup>

<sup>\*</sup>GIPSA-lab, Departement Image Signal - Grenoble INP

BP 46 - 38402 Saint Martin d'Hères - FRANCE

E-mail: {mathieu.fauvel, jocelyn.chanussot}@gipsa-lab.inpg.fr

<sup>†</sup>Department of Electrical and Computer Engineering

University of Iceland, Hjardarhagi 2-6, 107 Reykjavik, ICELAND

E-mail: benedikt@hi.is

## Abstract

Kernel Principal Component Analysis (KPCA) is investigated for feature extraction from hyperspectral remote-sensing data. Features extracted using KPCA are classified using linear support vector machines. In one experiment it is shown that kernel principal component features are more linearly separable than features extracted with conventional principal component analysis. In a second experiment, kernel principal components are used to construct the Extended Morphological Profile (EMP). Classification results, in terms of accuracy, are improved in comparison to original approach which used conventional principal component analysis for constructing the EMP. Experimental results presented in this paper confirm the usefulness of the KPCA for the analysis of hyperspectral data. For the one data set, the overall classification accuracy increases from 79% to 96% with the proposed approach.

## Index Terms

Kernel principal component, hyperspectral remote sensing data, support vector machines, feature extraction, classification, morphological profile.

# Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas

## CONTENTS

<b>I</b>	<b>Introduction</b>	2
<b>II</b>	<b>The Extended Morphological Profile</b>	4
<b>III</b>	<b>Kernel Principal Component Analysis</b>	6
	III-A Kernel PCA problem . . . . .	6
	III-B PCA versus KPCA . . . . .	7
	III-C KPCA applied to remote sensing data . . . . .	8
<b>IV</b>	<b>Support Vector Machines</b>	9
<b>V</b>	<b>Experiments</b>	10
	V-A Data set . . . . .	11
	V-B Spectral feature extraction . . . . .	14
	V-C Classification of remote sensing data . . . . .	15
	V-C1 University Area . . . . .	17
	V-C2 Pavia Center . . . . .	17
	V-C3 Washington DC . . . . .	17
	V-D Discussion . . . . .	19
<b>VI</b>	<b>Conclusions</b>	22
<b>VII</b>	<b>Acknowledgment</b>	22
	<b>References</b>	22

# Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas

## I. INTRODUCTION

Classification of hyperspectral data from urban areas using kernel methods is investigated in this article. Thanks to recent advances in hyperspectral sensors, it is now possible to collect more than one hundred bands at a high spatial resolution [1]. Consequently, in the spectral domain, pixels are vectors where each component contains specific wavelength information provided by a particular channel [2]. The size of the vector is related to the number of bands the sensor can collect. With hyperspectral data, vectors belong to a high-dimensional vector space, *e.g.* the 100-dimensional vector space  $\mathbb{R}^{100}$ .

With increasing resolution of the data, in the spectral or spatial domain, theoretical and practical problems appear. For example, in a high-dimensional space, normally-distributed data have a tendency to concentrate in the tails, which seems contradictory with a bell-shaped density function [3], [4]. For the purpose of classification, these problems are related to the *curse of dimensionality*. In particular, Hughes showed that with a limited training set, classification accuracy decreases as the number of features increases beyond a certain limit [5]. This is paradoxical, since with a higher spectral resolution one can discriminate more classes and have a finer description of each class – but the data complexity leads to poorer classification.

To mitigate this phenomenon, *feature selection / extraction* is usually performed as pre-processing to hyperspectral data analysis [6]. Such processing can also be performed for multi-spectral images in order to enhance class separability or to remove a certain amount of noise.

Transformations based on statistical analysis have already proved to be useful for classification, detection, identification, or visualization of remote-sensing data [2], [7]–[10]. Two main approaches can be defined:

- 1) *Unsupervised feature extraction*: The algorithm works directly on the data without any ground-truth. Its goal is to find another space of lower dimension for representing the data.
- 2) *Supervised feature extraction*: Training set data are available and the transformation is performed according to the properties of the training set. Its goal is to improve class separability by projecting the data onto a lower dimensional space.

Supervised transformation is in general well suited to pre-processing for the task of classification, since the transformation improves class separation. However, its effectiveness correlates with how well the training set

represents the data set as a whole. Moreover, this transformation can be extremely time-consuming. Example of supervised features extraction algorithms are:

- Sequential forward/backward selection methods and the improved versions of them: these methods select some bands from the original data set [11]–[13].
- Band selection using information theory: A collection of bands are selected according to their mutual information [14].
- Discriminant Analysis, Decision Boundary and Non-Weighted feature extraction (DAFE, DBFE and NWFE) [6]. These methods are linear and use second order information for feature extraction. They are 'state-of-the-art' methods within the remote-sensing community.

The unsupervised case does not focus on class discrimination, but looks for another representation of the data in a lower-dimensional space, satisfying some given criterion. For Principal Component Analysis (PCA), the data are projected into a subspace that minimizes the reconstruction error in the mean squared sense. Note that both the unsupervised and supervised cases can also be divided into *linear* and *non-linear* algorithms [15].

PCA plays an important role in the processing of remote-sensing images. Even though its theoretical limitations for hyperspectral data analysis have been pointed out [6], [16], in a practical situation the results obtained using PCA are still competitive for the purpose of classification [17], [18]. The advantages of PCA are its low complexity and the absence of parameters. However, PCA only considers the second-order statistic, which can limit the effectiveness of the method.

A non-linear version of the PCA has been shown to be capable of capturing a part of higher order statistics, thus better representing the information from the original data set [19], [20]. The first objective of this article is the application of the non-linear PCA to high-dimensional spaces, such as hyperspectral images, and to assess influence of using non-linear PCA on classification accuracy. In particular, Kernel PCA (KPCA) [20] has attracted our attention. Its relation to a powerful classifier, support vector machines, and its low computational complexity make it suitable for the analysis of remote sensing data.

Despite the favorable performance of KPCA in many application, no investigation has been carried out in the field of remote sensing. In this paper, the first contribution concerns the comparison of extracting features using conventional PCA and using KPCA for the classification of hyperspectral remote sensing data. In our very first investigation in [21], we found that the use of kernel principal components as input to a neural network classifier lead to an improvement in classification accuracy. However, a neural network is a non-linear classifier, and the conclusions were difficult to generalize to other classifiers. In the present study, we make use of a linear classifier (Support Vector Machine) to draw more general conclusions.

The second objective of the paper concerns an important issue in the classification of remote sensing data: the use of spatial information. High-resolution hyperspectral data from urban areas provide both detailed spatial and spectral information. Any complete analysis of such data needs to include both types of information. However,

conventional methods use the spectral information only. An approach has been proposed for panchromatic data (one spectral band) using mathematical morphology [22], [23]. The idea was to construct a feature vector, the *morphological profile*, that includes spatial information. Despite good results in terms of classification accuracy, an extension to hyperspectral data was not straightforward. In fact, due to the multi-valued nature of pixels, standard image-processing tools which require a total ordering relation, such as mathematical morphology [24], cannot be applied. Plaza *et al.* have proposed an extension to the morphological transformation in order to integrate spectral and spatial information from the hyperspectral data [25]. In [26], Benediktsson *et al.* have proposed a simpler approach, *i.e.*, to use the PCA to extract representative images from the data and apply morphological processing on each first principal component independently. A stacked vector, the *extended morphological profile*, is constructed from all the morphological profiles. Good classification accuracies were achieved, but it was found that too much spectral information were lost during by the PCA transformation [27], [28].

Motivated by the favorable results obtained using the KPCA in comparison with conventional PCA, the second contribution of this paper is the analysis of the pertinence of the features extracted with the KPCA in the construction of the Extended Morphological Profile.

The article is organized as follows. The EMP is presented in Section II. The KPCA is detailed in Section III. The Support Vector Machines for the purpose of classification are briefly reviewed in Section IV. Experiments are presented on real data sets in Section V. Finally, conclusion are drawn in Section VI.

## II. THE EXTENDED MORPHOLOGICAL PROFILE

In this section, we briefly introduce the concept of the morphological profile for the classification of remote sensing images.

Mathematical morphology provides high level operators to analyze spatial inter-pixel dependency [29]. One widely used approach is the Morphological Profile (MP) [30] which is a strategy to extract spatial information from high spatial resolution images [22]. It has been successfully used for the classification of IKONOS data from urban areas using a neural network [23]. Based on the *granulometry* principle [24], the MP consists of the successive application of geodesic closing/opening transformations of increasing size. An MP is composed of the *opening profile* (OP) and the *closing profile* (CP). The OP at pixel  $\mathbf{x}$  of the image  $f$  is defined as a  $p$ -dimensional vector:

$$OP_i(\mathbf{x}) = \gamma_R^{(i)}(\mathbf{x}), \quad \forall i \in [0, p] \quad (1)$$

where  $\gamma_R^{(i)}$  is the opening by reconstruction with a structuring element (SE) of size  $i$  and  $p$  is the total number of openings. Also, the CP at pixel  $\mathbf{x}$  of image  $f$  is defined as a  $p$ -dimensional vector:

$$CP_i(\mathbf{x}) = \phi_R^{(i)}(\mathbf{x}), \quad \forall i \in [0, p] \quad (2)$$

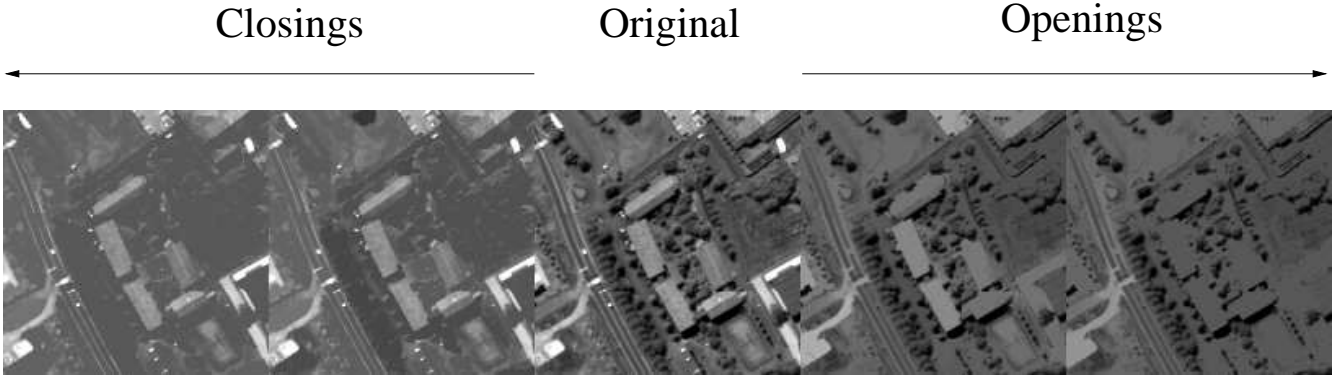


Fig. 1. Simple morphological profile with 2 openings and 2 closings. In the profile shown, circular structuring elements are used with radius increment 4 ( $r = 4, 8$  pixels). The image processed is part of Fig. 4.(a).

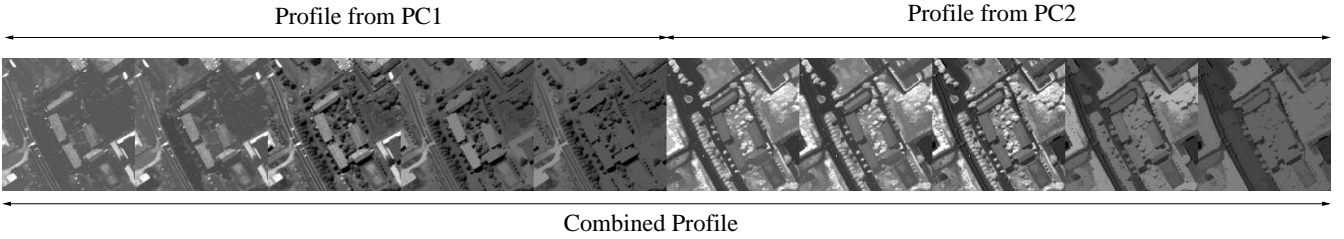


Fig. 2. Extended morphological profile of two images. Each of the original profiles has 2 opening and 2 closings. A circular structuring element with radius increment 4 was used ( $r = 4, 8$ ). The image processed is part of Fig. 4.(a).

where  $\phi_R^{(i)}$  is the closing by reconstruction with an SE of size  $i$ . Clearly, we have  $CP_0(\mathbf{x}) = OP_0(\mathbf{x}) = f(\mathbf{x})$ . By collating the OP and the CP, the MP of image  $f$  is defined as a  $2p + 1$ -dimensional vector:

$$MP(\mathbf{x}) = \{CP_p(\mathbf{x}), \dots, f(\mathbf{x}), \dots, OP_p(\mathbf{x})\}. \quad (3)$$

An example of MP is shown in Fig.1. Thus, from a single image a multivalued image results. The dimension of this image corresponds to the number of transformations. For application to hyperspectral data, characteristic images need to be extracted. In [26], it was suggested to use several principal components (PCs) of the hyperspectral data for such a purpose. Hence, the MP is applied on the first PCs, corresponding to a certain amount of the cumulative variance, and a stacked vector is built using the MP on each PC. This yields the *extended morphological profile* (EMP). Following the previous notation, the EMP is a  $q(2p + 1)$ -dimensional vector:

$$EMP(\mathbf{x}) = \{MP_{PC^1}(\mathbf{x}), \dots, MP_{PC^q}(\mathbf{x})\} \quad (4)$$

where  $q$  is the number of retaining PCs. An example of an EMP is shown in Fig.2.

As stated in the introduction, PCA does not fully handle the spectral information. Previous works using alternative feature reduction algorithms, such as Independent Component Analysis (ICA), have led to equivalent

results in terms of classification accuracy [31]. In this article, we propose the use of the KPCA rather than PCA for the construction of the EMP, *i.e.* the first kernel PCs (KPCs) are used to build the EMP. The assumption is that much more spectral information will be captured by the KPCA than with the PCA. The next section presents the KPCA and how the KPCA is applied to hyperspectral remote sensing images.

### III. KERNEL PRINCIPAL COMPONENT ANALYSIS

#### A. Kernel PCA problem

In this section, a brief description is given of kernel principal component analysis for feature reduction on remote-sensing data. The theoretical foundation may be found in [20], [32], [33].

The starting point is a set of pixel vectors  $\mathbf{x}^i \in \mathbb{R}^n$ ,  $i \in [1, \dots, \ell]$ . Conventional PCA solves the eigenvalue problem:

$$\lambda \mathbf{v} = \Sigma_{\mathbf{x}} \mathbf{v}, \text{ subject to } \|\mathbf{v}\|_2 = 1 \quad (5)$$

where  $\Sigma_{\mathbf{x}} = \mathbb{E} [\mathbf{x}_c \mathbf{x}_c^T] \approx \frac{1}{\ell-1} \sum_{i=1}^{\ell} (\mathbf{x}^i - \mathbf{m}_x) (\mathbf{x}^i - \mathbf{m}_x)^T$  and  $\mathbf{x}_c$  is the centered vector  $\mathbf{x}$ . A projection onto the first  $m$  principal components is performed as:  $\mathbf{x}_{pc} = [\mathbf{v}^1 \dots \mathbf{v}^m]^T \mathbf{x}$ .

To capture higher-order statistics, the data can be mapped onto another space  $\mathcal{H}^1$ :

$$\begin{aligned} \Phi : \mathbb{R}^n &\rightarrow \mathcal{H} \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}) \end{aligned} \quad (6)$$

where  $\Phi$  is a function that may be non-linear, and the only restriction on  $\mathcal{H}$  is that it must have the structure of a reproducing kernel Hilbert space (RKHS), not necessarily of finite dimension. PCA in  $\mathcal{H}$  can be performed as in the input space, but thanks to the kernel trick [34], it can be performed directly in the input space. The kernel PCA (KPCA) solves the following eigenvalue problem:

$$\lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha}, \text{ subject to } \|\boldsymbol{\alpha}\|_2 = \frac{1}{\lambda} \quad (7)$$

where  $\mathbf{K}$  is the kernel matrix constructed as follows:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}^1, \mathbf{x}^1) & \dots & k(\mathbf{x}^1, \mathbf{x}^\ell) \\ k(\mathbf{x}^2, \mathbf{x}^1) & \dots & k(\mathbf{x}^2, \mathbf{x}^\ell) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}^\ell, \mathbf{x}^1) & \dots & k(\mathbf{x}^\ell, \mathbf{x}^\ell) \end{pmatrix}. \quad (8)$$

The function  $k$  is the core of the KPCA. It is a positive semi-definite function on  $\mathbb{R}^n$  that introduces non-linearity into the processing. This is usually called a *kernel*. Classic kernels are the polynomial kernel,  $q \in \mathbb{R}^+$

<sup>1</sup>From now on,  $\mathbb{R}^n$  is called the input space and  $\mathcal{H}$  the feature space.

and  $p \in \mathbb{N}^+$ ,

$$k(\mathbf{x}, \mathbf{y}) = (\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{R}} + q)^p. \quad (9)$$

and the Gaussian kernel,  $\sigma \in \mathbb{R}^+$ ,

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right). \quad (10)$$

As with conventional PCA, once (7) has been solved, projection is then performed:

$$\Phi_{kpc}^m(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^m k(\mathbf{x}^i, \mathbf{x}). \quad (11)$$

Note it is assumed that  $\mathbf{K}$  is centered, otherwise it can be centered as [35]:

$$\mathbf{K}_c = \mathbf{K} - \mathbf{1}_\ell \mathbf{K} - \mathbf{K} \mathbf{1}_\ell + \mathbf{1}_\ell \mathbf{K} \mathbf{1}_\ell \quad (12)$$

where  $\mathbf{1}_\ell$  is a square matrix such as  $(\mathbf{1}_\ell)_{ij} = \frac{1}{\ell}$ .

### B. PCA versus KPCA

Let us start by recalling that the PCA relies on a simple generative model: The  $n$  observed variables result from a linear transformation of  $m$  Gaussianly distributed latent variables and thus it is possible to recover the latent variable from the observed one by solving (5).

To better understand the link and the difference between PCA and KPCA, one must note that the eigenvectors of  $\Sigma_{\mathbf{x}}$  can be obtained from those of  $\mathbf{X}\mathbf{X}^T$ , where  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_\ell]^T$  [36]. Consider the eigenvalue problem:

$$\gamma \mathbf{u} = \mathbf{X}\mathbf{X}^T \mathbf{u}, \text{ subject to } \|\mathbf{u}\|_2 = 1. \quad (13)$$

The left part is multiplied by  $\mathbf{X}^T$  giving

$$\begin{aligned} \gamma \mathbf{X}^T \mathbf{u} &= \mathbf{X}^T \mathbf{X}\mathbf{X}^T \mathbf{u} \\ \gamma \mathbf{X}^T \mathbf{u} &= (\ell - 1) \Sigma_{\mathbf{x}} \mathbf{X}^T \mathbf{u} \\ \gamma' \mathbf{X}^T \mathbf{u} &= \Sigma_{\mathbf{x}} \mathbf{X}^T \mathbf{u} \end{aligned} \quad (14)$$

which is the eigenvalue problem (5):  $\mathbf{v} = \mathbf{X}^T \mathbf{u}$ . But  $\|\mathbf{v}\|^2 = \mathbf{u}^T \mathbf{X}\mathbf{X}^T \mathbf{u} = \gamma \mathbf{u}^T \mathbf{u} = \gamma \neq 1$ . Therefore, the eigenvectors of  $\Sigma_{\mathbf{x}}$  can be computed from eigenvectors of  $\mathbf{X}\mathbf{X}^T$  as  $\mathbf{v} = \gamma^{-0.5} \mathbf{X}^T \mathbf{u}$ .

The matrix  $\mathbf{X}\mathbf{X}^T$  is equal to:

$$\begin{pmatrix} \langle \mathbf{x}^1, \mathbf{x}^1 \rangle & \dots & \langle \mathbf{x}^1, \mathbf{x}^\ell \rangle \\ \langle \mathbf{x}^2, \mathbf{x}^1 \rangle & \dots & \langle \mathbf{x}^2, \mathbf{x}^\ell \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{x}^\ell, \mathbf{x}^1 \rangle & \dots & \langle \mathbf{x}^\ell, \mathbf{x}^\ell \rangle \end{pmatrix} \quad (15)$$

which is the kernel matrix with a linear kernel:  $k(\mathbf{x}^i, \mathbf{x}^j) = \langle \mathbf{x}^i, \mathbf{x}^j \rangle_{\mathbb{R}^n}$ . Using the kernel trick  $k(\mathbf{x}^i, \mathbf{x}^j) = \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j) \rangle_{\mathcal{H}}$ ,  $\mathbf{K}$  can be rewritten in a similar form as (15):

$$\begin{pmatrix} \langle \Phi(\mathbf{x}^1), \Phi(\mathbf{x}^1) \rangle_{\mathcal{H}} & \dots & \langle \Phi(\mathbf{x}^1), \Phi(\mathbf{x}^\ell) \rangle_{\mathcal{H}} \\ \langle \Phi(\mathbf{x}^2), \Phi(\mathbf{x}^1) \rangle_{\mathcal{H}} & \dots & \langle \Phi(\mathbf{x}^2), \Phi(\mathbf{x}^\ell) \rangle_{\mathcal{H}} \\ \vdots & \ddots & \vdots \\ \langle \Phi(\mathbf{x}^\ell), \Phi(\mathbf{x}^1) \rangle_{\mathcal{H}} & \dots & \langle \Phi(\mathbf{x}^\ell), \Phi(\mathbf{x}^\ell) \rangle_{\mathcal{H}} \end{pmatrix}. \quad (16)$$

From (15) and (16), the advantage of using KPCA comes from an appropriate projection  $\Phi$  of  $\mathbb{R}^n$  onto  $\mathcal{H}$ : In this space, the data should better match the PCA model. It is clear that the KPCA shares the same properties as the PCA, but in different space.

To illustrate how the KPCA works, a short example is given here. Fig. 3.(a) represents three Gaussian clusters. The conventional PCA would result in a rotation of the space, *i.e.*, the three clusters would not be identified. Fig. 3.(b) and (c) represent the projection onto the first two kernel principal components (KPCs). Using a Gaussian kernel the structure of the data is better captured than with PCA: A cluster can be clearly identified on the first KPC (see Fig. 3.(e)). However, the obtained results are different with a polynomial kernel. In that case, the clusters are not as well identified as with the Gaussian kernel. Finally, from the contour plots, Fig. 3.(e) and (f), the non linear projection of the KPCA can be seen while linear projection with the PCA can be seen in Fig. 3.(d): The contour plots are straight lines with PCA while curved lines with KPCA.

This synthetic experiment reveals the importance of the choice of kernels. In the next section the selection of a kernel adapted to hyperspectral remote sensing data is discussed.

### C. KPCA applied to remote sensing data

To compute the KPCA, it is first necessary to choose the kernel function to build the kernel matrix. This is a difficult task which is still under consideration in the 'kernel method' community [37]. However, when considering the two classical kernels in (9) and (10), one can choose between them using some prior information: If it is known that higher-order statistic are relevant to discriminate samples, a polynomial kernel should be used. But under the Gaussian cluster assumption, the Gaussian kernel should be used. Hyperspectral remote sensing data are known to be well approximated by a Gaussian distribution [7], and thus in this work a Gaussian kernel is used.

With the Gaussian kernel, one hyperparameter needs to be tuned, *i.e.*,  $\sigma$ . The  $\sigma$  controls the width of the exponential function. A too small value of  $\sigma$  causes  $k(\mathbf{x}^i, \mathbf{x}^j) = 0$ ,  $i \neq j$ , *i.e.*, each sample is considered as an individual cluster. While a too high value causes  $k(\mathbf{x}^i, \mathbf{x}^j) = 1$ , *i.e.*, all samples are considered neighbors. Thus, only one cluster can be identified. Several strategies can be used, from cross-validation to density estimation [38]. The choice of  $\sigma$  should reflect the range of the variables, to be able to detect samples that belong to the

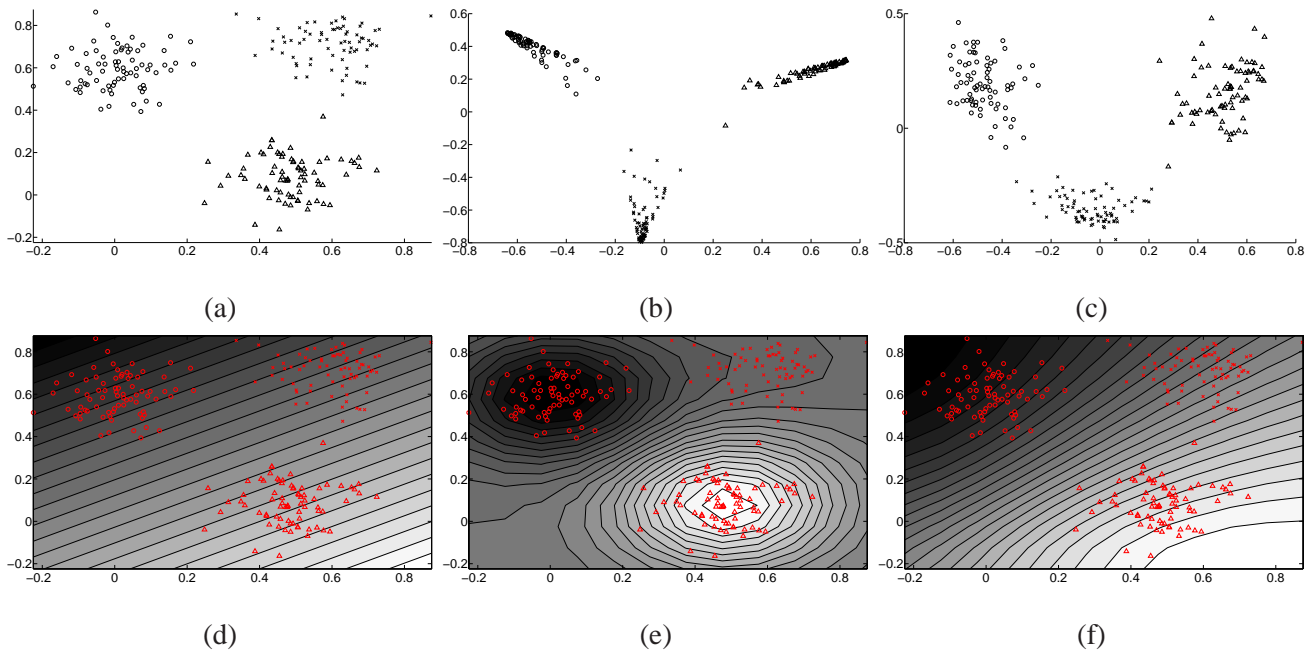


Fig. 3. PCA versus KPCA: (a) Three Gaussian clusters, and their projection onto the first two kernel principal components with (b) a Gaussian kernel and (c) a polynomial kernel. (d), (e) and (f) represent, respectively, the contour plot of the projection onto the first component for the PCA, the KPCA with Gaussian kernel and the KPCA with a polynomial kernel. Note how with the Gaussian kernel the first component ‘picks out’ the individual clusters [20]. The intensity of the contour plot is proportional to the value of the projection, *i.e.*, light gray indicates that  $\Phi_{kpc}^1(\mathbf{x})$  has a high value.

same cluster from those that belong to others clusters. A simple, yet effective, strategy was employed in this experiment. It consists of stretching the variables between 0 and 1, and fixing  $\sigma$  to a value that provides good results according to some criterion. For a remote sensing application, the number of extracted KPCs should be of same order than the number of species/classes in the image. From our experiments,  $\sigma$  was fixed at 4 for all data sets.

Section V presents experimental results using the KPCA on real hyperspectral images. As stated in the introduction, the aim of using the KPCA is to extract relevant features for the construction of the EMP. The classification of such features with the Support Vector Machines is described in the next section.

#### IV. SUPPORT VECTOR MACHINES

The Support Vector Machines (SVM) are surely one of the most used kernel learning algorithms. They perform robust non-linear classification of samples using the kernel trick. The idea is to find a separating hyperplane in some feature space induced by the kernel function while all the computations are done in the original space [39]. A good introduction to SVM for pattern recognition may be found in [40]. Given a training set  $\mathcal{S} = \{(\mathbf{x}^1, y_1), \dots, (\mathbf{x}^\ell, y_\ell)\} \in \mathbb{R}^n \times \{-1; 1\}$ , the decision function is found by solving the convex optimization

problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} g(\boldsymbol{\alpha}) &= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} & \quad 0 \leq \alpha_i \leq C \text{ and } \sum_{i=1}^{\ell} \alpha_i y_i = 0 \end{aligned} \quad (17)$$

where  $\boldsymbol{\alpha}$  are the Lagrange coefficients,  $C$  a constant that is used to penalize the training errors, and  $k$  the kernel function. Same than KPCA, classic effective kernels are (9) and (10). A short comparison of kernels for remotely sensed image classification may be found in [41]. Advanced kernel functions can be constructed using some prior [42].

When the optimal solution of (17) is found, *i.e.*,  $\alpha_i$ , the classification of a sample  $\mathbf{x}$  is achieved by observing to which side of the hyperplane it belongs:

$$y = \text{sgn} \left( \sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}^i, \mathbf{x}) + b \right). \quad (18)$$

SVM are designed to solve binary problems where the class labels can only take two values:  $\pm 1$ . For a remote-sensing application, several species/classes are usually of interest. Various approaches have been proposed to address this problem. They usually combine a set of binary classifiers. Two main approaches were originally proposed for  $C$ -class problems [35]:

- *One-versus-the-rest*:  $C$  binary classifiers are applied on each class against all the others. Each sample is assigned to the class with the maximum output.
- *Pairwise Classification*:  $\frac{C(C-1)}{2}$  binary classifiers are applied on each pair of classes. Each sample is assigned to the class getting the highest number of votes. A vote for a given class is defined as a classifier assigning the pattern to that class.

*Pairwise classification* has proved more suitable for large problems [43]. Even though the number of classifiers used is larger than for the *one-versus-the-rest* approach, the whole classification problem is decomposed into much simpler ones. Therefore, the pairwise approach was used in our experiments. More advanced approaches applied to remote-sensing data can be found in [44].

SVM are primarily a *non-parametric* method, yet some hyperparameters do need to be tuned before optimization. In the Gaussian kernel case, there are two hyperparameters:  $C$  the penalty term and  $\sigma$  the width of the exponential. This is usually done by a cross-validation step, where several values are tested. In our experiments,  $C$  is fixed to 200 and  $\sigma^2 \in \{0.5, 1, 2, 4\}$  is selected using 5-fold cross validation. The SVM optimization problem was solved using the LIBSVM [45]. The range of each feature was stretched between 0 and 1.

## V. EXPERIMENTS

Three real data sets were used in the experiments. They are detailed in the following. The original hyperspectral data are termed 'Raw' in the rest of the paper.

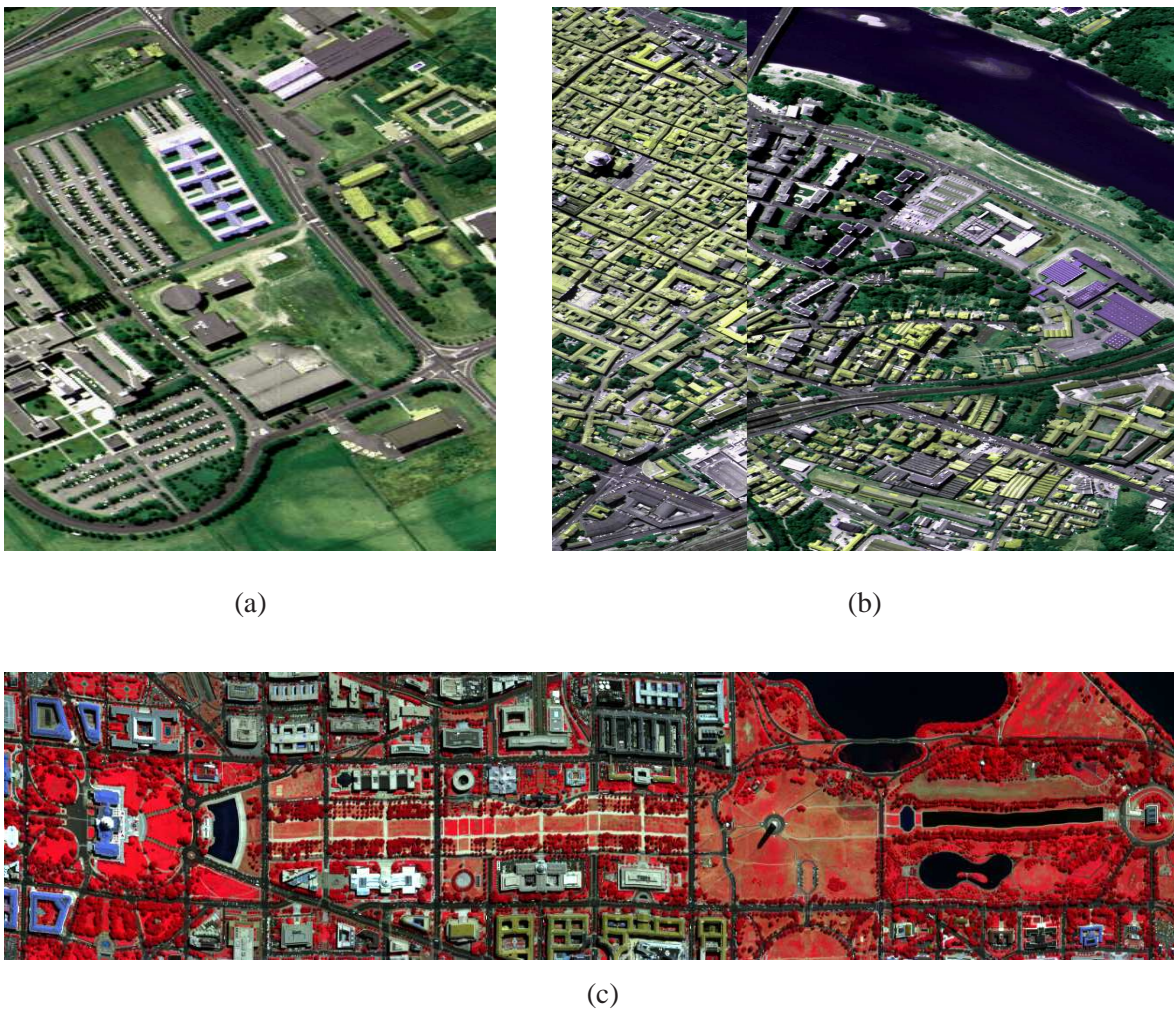


Fig. 4. ROSIS data: (a) University Area, (b) Pavia Center. HYDICE data: (c) Washington DC.

#### A. Data set

Airborne data from the ROSIS-03 (Reflective Optics System Imaging Spectrometer) optical sensor are used for the first two experiments. The flight over the city of Pavia, Italy, was operated by the Deutschen Zentrum für Luft- und Raumfahrt (DLR, the German Aerospace Agency) within the context of the HySens project, managed and sponsored by the European Union. According to specifications, the ROSIS-03 sensor provides 115 bands with a spectral coverage ranging from  $0.43$  to  $0.86\mu\text{m}$ . The spatial resolution is  $1.3$  m per pixel. The two data sets are:

- 1) *University Area*: The first test set is around the Engineering School at the University of Pavia. It is  $610 \times 340$  pixels. Twelve channels have been removed due to noise. The remaining 103 spectral channels are processed. Nine classes of interest are considered: Tree, asphalt, bitumen, gravel, metal sheet, shadow, bricks, meadow, and soil.
- 2) *Pavia Center*: The second test set is the center of Pavia. The Pavia center image was originally  $1096 \times$

TABLE I  
INFORMATION CLASSES AND TRAINING/TEST SAMPLES FOR THE UNIVERSITY AREA DATA SET.

Class		Samples	
No	Name	Train	Test
1	Asphalt	548	6641
2	Meadow	540	18649
3	Gravel	392	2099
4	Tree	524	3064
5	Metal Sheet	265	1345
6	Bare Soil	532	5029
7	Bitumen	375	1330
8	Brick	514	3682
9	Shadow	231	947
Total		3921	42776

1096 pixels. A 381 pixel wide black band in the left-hand part of image was removed, resulting in a ‘two part’ image of  $1096 \times 715$  pixels. Thirteen channels have been removed due to noise. The remaining 102 spectral channels are processed. Nine classes of interest are considered: Water, tree, meadow, brick, soil, asphalt, bitumen, tile, and shadow.

Airborne data from the HYDICE sensor (Hyperspectral Digital Imagery Collection Experiment) was used for the third experiments. The HYDICE was used to collect data from flightline over the Washington DC Mall. Hyperspectral HYDICE data originally contained 210 bands in the  $0.4\text{-}2.4\mu\text{m}$  region. Channels from near-infrared and infrared wavelengths are known to contained more noise than channel from visible wavelengths. Noisy channels due to water absorption have been removed and the set consists of 191 spectral channels. The data were collected in August 1995 and each channel has 1280 lines with 307 pixels each. Seven information class were defined, namely: Roof, road, grass, tree, trail, water and shadow. Fig. 4 shows false color images for all the data sets.

Available training and test sets for each data set are given in Tables I, II and III. These are selected pixels from the data by an expert, corresponding to a predefined species/classes. Pixels from the training set are excluded from the test set in each case and vice-versa.

The classification accuracy was assessed with:

- An overall accuracy (OA) which is the number of well classified samples divided by the number of test samples.
- An average accuracy (AA) which represents the average of class classification accuracy.
- A kappa coefficient of agreement ( $\kappa$ ) which is the percentage of agreement corrected by the amount of

TABLE II  
INFORMATION CLASSES AND TRAINING/TEST SAMPLES FOR THE PAVIA CENTER DATA SET.

Class		Samples	
No	Name	Train	Test
1	Water	824	65971
2	Tree	820	7598
3	Meadow	824	3090
4	Brick	808	2685
5	Bare Soil	820	6584
6	Asphalt	816	9248
7	Bitumen	808	7287
8	Tile	1260	42826
9	Shadow	476	2863
Total		7456	148152

TABLE III  
INFORMATION CLASSES AND TRAINING/TEST SAMPLES FOR THE *Washington DC Mall* DATA SET.

Class		Samples	
No.	Name	Train	Test
1	Roof	40	3794
2	Road	40	376
3	Trail	40	135
4	Grass	40	1888
5	Tree	40	365
6	Water	40	1184
7	Shadow	40	57
Total		280	6929

agreement that could be expected due to chance alone [7].

- A class accuracy which is the percentage of correctly classified samples for a given class.

These criteria were used to compare classification results and were computed using a confusion matrix. Furthermore, the statistical significance of differences was computed using McNemar's test, which is based upon the standardized normal test statistic [46]:

$$Z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \quad (19)$$

where  $f_{12}$  indicates the number of samples classified correctly by classifier 1 and incorrectly by classifier 2. The difference in accuracy between classifiers 1 and 2 is said to be statistically significant if  $|Z| > 1.96$ . The sign

TABLE IV

PCA: EIGENVALUES AND CUMULATIVE VARIANCE IN PERCENTAGES FOR THE THREE HYPERSPECTRAL DATA SETS.

Component	Pavia Center		University Area		Washington DC	
	%	Cum. %	%	Cum. %	%	Cum. %
1	72.85	72.85	64.85	64.85	53.38	53.38
2	21.03	93.88	28.41	93.26	18.65	72.03
3	04.23	98.11	05.14	98.40	03.83	75.87
4	00.89	99.00	00.51	98.91	02.00	77.87
5	00.30	99.30	00.25	99.20	00.66	78.00

of  $Z$  indicates whether classifier 1 is more accurate than classifier 2 ( $Z > 0$ ) or vice-versa ( $Z < 0$ ). This test assumes that the training and the test samples are related and is thus adapted to the analysis since the training and test sets were the same for each experiment for a given data set.

### B. Spectral feature extraction

Solving the eigenvalues problem (5) for each data set yields the results reported in Table IV. Looking at the cumulative eigenvalues, in each ROSIS case, three principal components (PCs) reach 95% of total variance. After the PCA transformation, the dimensionality of the new representation of the *University Area* data set and the *Pavia Center* is 3, if the threshold is set to 95% of the cumulative variance. The results for the third data set are somewhat different. Acquired from a higher range of wavelengths, more noise is contained in the data and more bands were removed by comparison to the ROSIS data. That explains why more PCs are needed, *i.e.*, 40 PCs, to reach 95% of the cumulative variance. But from the table, it can be clearly seen that the first two PCs contains most of the information. This means that using second-order information, the hyperspectral data can be reduced to a two- or three-dimensional space. But, as experiments will show, hyperspectral richness is not fully handled using only the mean and variance/covariance of the data.

Table V shows the variance and the cumulative variance for the three data sets when KPCA is applied. The kernel matrix in each case was constructed using 5000 randomly-selected samples. From the table, it can be seen that more kernel principal components (KPCs) are needed to achieve the same amount of variance as for the conventional PCA. For the *University* data set, the first 12 KPCs are needed to achieve 95% of the cumulative variance, 11 for the *Washington DC* data set and only 10 for the *Pavia Center* data set. That may be an indication that more information is extracted and the KPCA is more robust to the noise, since a reasonable number of features are extracted from the *Washington DC* data set.

To test this assumption, the Mutual Information (MI) between each (K)PC has been computed. The classical correlation coefficient was not used since the PCA is optimal for that criterion. For comparison, the normalized

TABLE V

KPCA: EIGENVALUES AND CUMULATIVE VARIANCE IN PERCENT FOR THE TWO HYPERSPECTRAL DATA SETS (KPCA).

Component	Pavia Center		University Area		Washington DC	
	%	Cum. %	%	Cum. %	%	Cum. %
1	43.94	43.94	31.72	31.72	40.99	40.99
2	21.00	64.94	26.04	57.76	20.18	61.17
3	15.47	80.41	19.36	75.12	13.77	74.95
4	05.23	85.64	06.76	81.88	05.99	80.94
5	03.88	89.52	04.31	86.19	05.22	86.16

MI was computed<sup>2</sup>. The MI is used to test independence between two variables and intuitively the MI measures the information that the two variables share. A MI close to 0 indicates independence, while a high MI indicates dependence and consequently similar information. Fig. 5 presents the MI matrices, which represents the MI for each pair of extracted features with both PCA and KPCA, for the *Washington DC* data set. From Fig. 5(a), PCs number 4 to 40 contain more or less the same information since they correspond to a high MI. Although uncorrelated, these features are still dependent. This phenomenon is due to the noise contained in the data which is not Gaussian [6] and is distributed over several PCs. From Fig. 5(a), KPCA is less sensitive to the noise, *i.e.*, in the feature space the data match better the PCA model and the noise tends to be Gaussian. Note that with KPCA, only the first 11 KPCs are retained against 40 with conventional PCA.

To visually assess what is contained in each different (K)PC, Fig.6 represents the first, second and thirtieth PC for both the PCA and the KPCA. It can be seen that:

- 1) The extracted PCs are different (all the images have been linearly stretched between 0 and 255 for the purpose of visualization),
- 2) The thirtieth PCs contains only noise, while the thirtieth KPCs still contain some information and spatial structure can be detected with the EMP.

In conclusion of this section, the KPCA can extract more information from the hyperspectral data than the conventional PCA, and is robust to the noise that can affect remote sensing data. The next question is: Is this information useful for the purpose of classification? In the next section, experiments are conducted using features extracted by the PCA and the KPCA, for the classification or for the construction of the EMP.

### C. Classification of remote sensing data

Several experiments were conducted to evaluate KPCs as a suitable feature for 1) the classification of remote sensing images and 2) the construction of the EMP. For the first item, linear SVM are used to perform the

$${}^2I_n(\mathbf{x}, \mathbf{y}) = \frac{I(\mathbf{x}, \mathbf{y})}{\sqrt{I(\mathbf{x}, \mathbf{x})} \sqrt{I(\mathbf{y}, \mathbf{y})}}$$

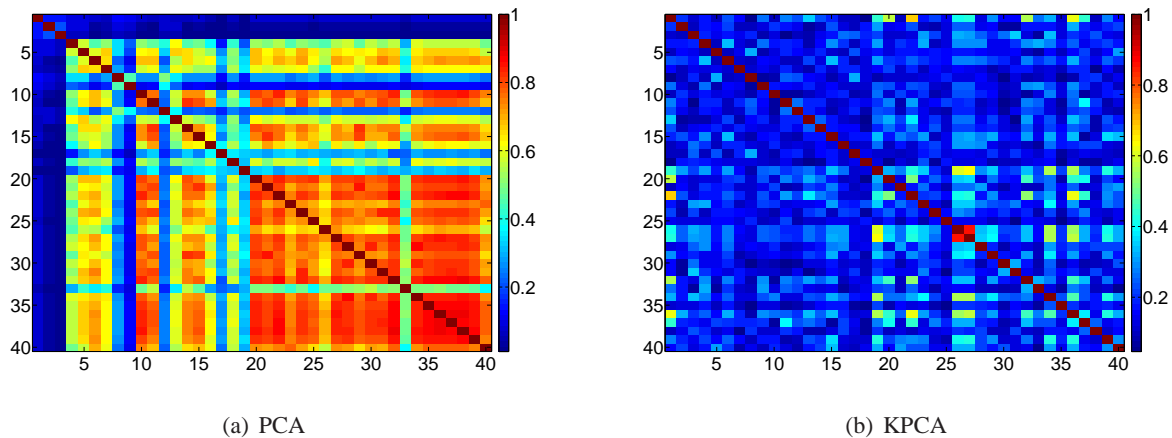


Fig. 5. Mutual Information matrices for the *Washington DC* data set.

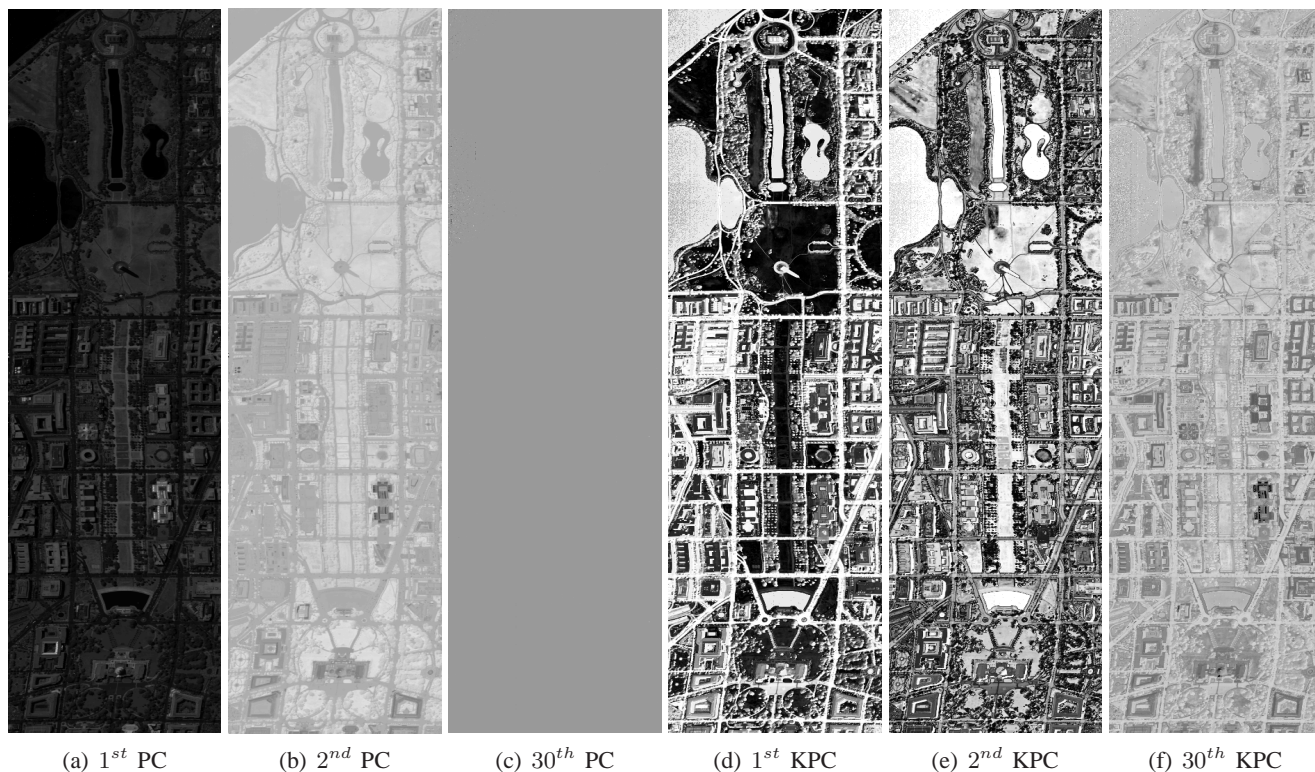


Fig. 6. (Kernel) Principal Component for the *Washington DC* data set.

classification. The aim is to investigate whether the data are easily classified after the PCA or the KPCA. Therefore a linear classifier is used to limit its influence on the results. For the EMP, as state in the introduction, too much information are lost during the PCA, and experiments should confirm that the KPCA extracts more information. In the following, an analysis of the results for each data sets is provided.

In each case, the EMP was constructed using (K)PCs corresponding to 95% of the cumulative variance. A circular SE with a step size increment of 2 was used. Four openings and closings were computed for each

(k)PC, resulting in an EMP of dimension  $9 \times m$  ( $m$  being the number of retained (K)PCs).

1) *University Area*: The results are reported in Table VI and the  $Z$  tests in Table VII. Regarding the global accuracies, the linear classification of PCA and KPCA features is significantly better than what is obtained by directly classifying the spectral data. Although feature extraction helps for the classification whatever the algorithm, the difference between PCA- and KPCA-based results is not statistically significant, *i.e.*  $|Z| \leq 1.96$ .

The non-linear SVM yield to a significant improvement in terms of accuracy when compared to linear SVM. The KPCA features are the more accurately classified, with an OA equal to 79.81%. The raw data are classified using the non-linear SVM and a significant improvement of the accuracy is achieved. However, the PCA features lose a lot of spectral information as compared to the KPCA and the classification of the PCA feature is less accurate than the one obtained using the all spectral channel or KPCs.

EMP constructed with either PCs or KPCs outperformed all others approaches in classification. The  $\kappa$  is increased by 15 % with  $EMP_{PCA}$  and by 20% with  $EMP_{KPCA}$ . The statistical difference of accuracy  $Z = -35.33$  clearly demonstrates the benefit of using the KPCA rather than the PCA.

Regarding the class accuracy, the highest improvements were obtained for class 1 (*Asphalt*), class 2 (*Meadow*) and class 3 (*Gravel*). For these classes, the original spectral information was not sufficient and the morphological processing provided additional useful information.

Thematic maps obtained with the non-linear SVM applied to the Raw data,  $EMP_{PCA}$  and  $EMP_{KPCA}$  are reported in Fig. 7. For instance, it can be seen that building in the top right corner (made of bitumen) is detected with  $EMP_{KPCA}$  while totally missed with  $EMP_{PCA}$ . The region corresponding to class 2, *meadow*, are more homogeneous in the image Fig.7.(c) than in the two others images.

2) *Pavia Center*: The results are reported in Table VIII and the  $Z$  tests in Table IX. The Pavia Center data set was easier to classify since even the linear SVM provide very high classification accuracy. Regarding the global accuracies, feature extraction does not improve the accuracies, for both linear and non-linear SVM. Yet, the KPCA performs significantly better than the PCA in terms of accuracies; even more, the KPCA + linear SVM outperform the PCA + non-linear SVM. Even high accuracy for linear SVM, the use of non-linear SVM is still justified since significantly higher accuracies are obtained with  $Z = 2.07$ .

Again, the very best results are obtained with EMP for both the PCA and the KPCA. However, the statistical significance of difference is lower than with the University Area data set although it is still significant:  $Z = -2.90$ .

For the class accuracy, most of the improvement is done on class 4 (*Brick*) which is almost perfectly classified with the  $EMP_{KPCA}$  and the non-linear SVM.

3) *Washington DC*: The results are reported in Table V-C3 and the  $Z$  tests in Table XI. The ground truth of the Washington DC data sets is limited, resulting in a very small training and test sets. As mentioned in Section V-B, the data contain non-Gaussian noise, and the number of PCs needed to reach 95% of the cumulative

TABLE VI  
CLASSIFICATION RESULTS FOR THE UNIVERSITY AREA DATA SET.

Feature Nb of features	SVM & Linear kernel			SVM & Gaussian kernel				
	Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
	103	3	12	103	3	12	27	108
OA	76.40	78.32	78.22	79.48	78.38	79.81	92.04	<b>96.55</b>
AA	85.04	81.77	87.58	88.14	85.16	87.60	93.21	<b>96.23</b>
$\kappa$	68.67	71.95	72.96	74.47	72.73	74.79	89.65	<b>95.43</b>
1	81.44	72.63	85.44	84.35	78.83	82.63	94.60	<b>96.23</b>
2	59.61	80.61	63.89	66.20	71.31	68.81	88.79	<b>97.58</b>
3	75.94	59.31	71.18	71.99	67.84	67.98	73.13	<b>83.66</b>
4	81.09	97.55	96.83	98.01	98.17	98.14	99.22	<b>99.35</b>
5	<b>99.55</b>	<b>99.55</b>	99.48	99.48	<b>99.55</b>	99.41	<b>99.55</b>	99.48
6	93.94	58.82	90.61	93.12	78.62	92.34	<b>95.23</b>	92.88
7	89.62	84.74	90.90	91.20	88.12	90.23	98.87	<b>99.10</b>
8	84.79	82.84	91.99	92.26	86.28	91.88	99.10	<b>99.46</b>
9	<b>99.47</b>	99.89	97.89	96.62	97.68	97.47	90.07	98.31

TABLE VII

STATISTICAL SIGNIFICANCE OF DIFFERENCES IN CLASSIFICATION ( $Z$ ) FOR THE UNIVERSITY AREA DATA SET. EACH CASE OF THE TABLE REPRESENTS  $Z_{rc}$  WHERE  $r$  IS THE ROW AND  $c$  IS THE COLUMN.

$Z_{rc}$		SVM & Linear kernel			SVM & Gaussian kernel				
		Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
Linear	Raw		-13.68	-18.91	-23.76	-13.88	-23.28	-73.77	-89.61
	PCA	13.68		0.41	-4.81	-0.27	-6.41	-57.49	-83.49
	KPCA	18.91	-0.41		-8.14	-0.69	-10.15	-64.42	-82.07
Gaussian	Raw	23.76	4.81	8.14		5.14	-2.49	-60.28	-78.69
	PCA	13.88	0.27	0.69	-5.14		-7.19	-59.90	-82.43
	KPCA	23.28	6.41	10.15	2.49	7.19		-59.45	-78.34
	EMP <sub>PCA</sub>	73.77	57.49	64.42	60.28	59.90	59.45		-35.33
	EMP <sub>KPCA</sub>	89.61	83.49	82.07	78.69	82.43	78.34	35.33	

variance is high.

From the global accuracies, all the different approaches perform similarly. It is confirmed with the  $Z$  test. Linear and non-linear SVM applied on the raw data sets provide the same results, and it is the same for the KPCA features. Despite high number of feature, PCA and linear SVM provide poor results. But surprisingly, one of the best results are obtained with PCA features and non-linear SVM. It means that non-linear can properly deal with the noise contains in the PCs.



Fig. 7. Thematic map obtained with the University Area: (a) Raw data, (b)  $EMP_{PCA}$ , (c)  $EMP_{KPCA}$ . The classification was done by SVM with a Gaussian kernel. The color-map is as follows asphalt, meadow, gravel, tree, metal sheet, bare soil, bitumen, brick and shadow.

As with previous experiments, best accuracies are achieved with the EMP, but also with the PCA, and non-linear SVM. The difference in the three classification are not statistically significant, as can be seen from Table XI.

Regarding the class accuracies, the class 7, *shadow*, is perfectly classifier only by  $EMP_{KPCA}$ .

#### D. Discussion

As in stated the introduction, the first objective of this paper was to assess the relevance of the KPCA as a feature reduction tool for hyperspectral remote sensing imagery. From the experiments with the linear SVM, the classification accuracies are at least similar (one case) or better (two cases) with the features extracted with KPCA thus legitimizing KPCA as a suitable alternative to PCA. The same conclusion can be drawn when the classification is done with non-linear SVM.

The second objective was to use the KPCs for the construction of an EMP. Comparison with an EMP constructed with PCs is significantly favorable to KPCA for two cases. For the most difficult data, the *University Area*, the OA reaches 96.55% with  $EMP_{KPCA}$  which is 4.5% more than with  $EMP_{PCA}$ . This results strengthen the use of KPCA against PCA.

TABLE VIII  
CLASSIFICATION RESULTS FOR THE PAVIA CENTER DATA SET.

Feature Nb of features	SVM & Linear kernel			SVM & Gaussian kernel				
	Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
	102	3	10	102	3	10	27	90
OA	97.60	96.54	97.39	97.67	96.99	97.32	98.81	<b>98.87</b>
AA	95.42	92.34	94.38	95.60	93.56	94.40	98.14	<b>98.25</b>
$\kappa$	96.62	95.14	96.32	96.71	95.76	96.23	98.32	<b>98.41</b>
1	98.41	98.82	98.57	98.35	98.80	98.49	<b>99.07</b>	98.91
2	<b>93.43</b>	85.56	90.94	91.23	87.33	89.06	92.67	92.01
3	96.57	94.82	95.15	<b>96.76</b>	94.98	95.40	96.38	96.31
4	88.27	81.15	83.87	88.45	82.94	82.50	<b>99.70</b>	99.59
5	94.41	88.97	94.99	93.97	95.23	94.55	99.39	<b>99.77</b>
6	95.17	94.82	95.36	96.32	94.72	96.06	98.48	<b>99.24</b>
7	93.18	88.14	91.12	96.01	89.24	94.50	97.98	<b>98.58</b>
8	99.38	98.30	99.43	99.40	98.83	99.07	99.68	<b>99.89</b>
9	99.93	99.93	<b>99.97</b>	99.93	99.93	99.93	99.93	99.55

TABLE IX  
STATISTICAL SIGNIFICANCE OF DIFFERENCES IN CLASSIFICATION ( $Z$ ) FOR THE PAVIA CENTER DATA SET. EACH CASE OF THE TABLE REPRESENTS  $Z_{rc}$  WHERE  $r$  IS THE ROW AND  $c$  IS THE COLUMN.

$Z_{rc}$		SVM & Linear kernel			SVM & Gaussian kernel				
		Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
Linear	Raw		26.74	6.13	-2.07	15.15	7.87	-32.71	-34.26
	PCA	-26.74		-21.01	-27.71	-13.00	-19.70	-52.92	-54.27
	KPCA	-6.13	21.01		-9.45	12.19	2.38	-37.30	-40.23
Gaussian	Raw	2.07	27.71	9.45		18.9	13.30	-32.25	-35.36
	PCA	-15.15	13.00	-12.19	-18.91		-10.13	-45.78	-47.67
	KPCA	-7.78	19.70	-2.38	-13.30	10.13		-39.03	-42.41
	EMP <sub>PCA</sub>	32.71	52.92	37.30	32.25	45.78	39.03		-2.90
	EMP <sub>KPCA</sub>	35.26	54.27	40.23	35.36	47.67	42.41	2.90	

For the third data set, which contains non-Gaussian noise, the KPCA clearly deal better with the noise than PCA. Furthermore, a reasonable number of KPCs was extracted, *i.e.*, 10 compared to 40 extracted with PCA.

In this article, the Gaussian kernel was used for both the KPCA and the non-linear SVM. For the KPCA, the statistical behavior of the data has justified this choice and for the SVM previous experiments have shown that the Gaussian kernel produce the best accuracies. However, when no or little prior information is available from the data, the choice of the kernel for the KPCA is not straightforward. A Gaussian kernel is in general a

TABLE X  
CLASSIFICATION RESULTS FOR THE WASHINGTON DC DATA SET.

Feature Nb of features	SVM & Linear kernel			SVM & Gaussian kernel				
	Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
	103	40	11	103	40	11	360	99
OA	98.16	97.85	98.18	98.16	<b>98.84</b>	98.18	98.64	98.73
AA	98.89	95.95	97.20	96.89	97.65	97.20	98.02	<b>99.39</b>
$\kappa$	97.35	96.90	97.38	97.35	<b>98.32</b>	97.38	98.04	98.16
1	97.05	96.50	97.08	97.05	98.10	97.08	<b>97.52</b>	<b>97.52</b>
2	98.08	99.28	98.32	98.08	98.28	98.32	<b>99.52</b>	98.80
3	<b>100</b>	99.43	<b>100</b>	<b>100</b>	99.43	<b>100</b>	<b>100</b>	<b>100</b>
4	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
5	98.02	98.77	98.02	98.02	99.26	98.02	<b>99.51</b>	<b>99.51</b>
6	99.51	97.88	99.35	99.59	99.84	99.35	<b>99.92</b>	<b>99.92</b>
7	85.57	79.38	87.65	85.57	87.63	87.63	89.69	<b>100</b>

TABLE XI

STATISTICAL SIGNIFICANCE OF DIFFERENCES IN CLASSIFICATION ( $Z$ ) FOR THE WASHINGTON DC DATA SET. EACH CASE OF THE TABLE REPRESENTS  $Z_{rc}$  WHERE  $r$  IS THE ROW AND  $c$  IS THE COLUMN.

$Z_{rc}$		SVM & Linear kernel			SVM & Gaussian kernel				
		Raw	PCA	KPCA	Raw	PCA	KPCA	EMP <sub>PCA</sub>	EMP <sub>KPCA</sub>
Linear	Raw		2.81	-0.57	0	-6.03	-0.57	-5.68	-6.78
	PCA	-2.81		-3.00	-2.81	-7.84	-3.00	-8.00	-7.98
	KPCA	0.57	3.00		0.57	-5.81	0	-5.39	-6.63
Gaussian	Raw	0	2.81	-0.57		-6.03	-0.57	-5.68	-6.78
	PCA	6.03	7.84	5.81	6.03		5.81	2.06	1.02
	KPCA	0.57	3.00	0	0.57	5.81		-5.39	-6.63
	EMP <sub>PCA</sub>	5.68	8.00	5.39	5.68	-2.06	5.39		-1.40
	EMP <sub>KPCA</sub>	6.78	7.98	6.63	6.78	-1.02	6.63	1.40	

good initial choice. However, the best results are surely obtained with a more appropriate kernel.

The computational load for the KPCA is increased by comparison to the PCA. Both involve matrix inversions which are  $o(d^3)$ , where  $d$  is the number of variable for the PCA and the number of samples for the KPCA; clearly  $d_{PCA} \ll d_{KPCA}$ , e.g., for the *Washington DC* data set  $d_{PCA} = 191$  and  $d_{KPCA} = 5000$ . Thus, even if the KPCA involves a well known matrix algorithm, the computational load (both in terms of CPU and memory) is higher than with the PCA.

## VI. CONCLUSIONS

This article presents KPCA-based methods with application to the analysis of hyperspectral remote-sensing data. Two important issues have been considered: 1) Unsupervised feature extraction by means of the KPCA, 2) Construction of the EMP with KPCs. Comparisons were done with the conventional PCA. Comparisons in terms of classification accuracies with a linear SVM demonstrate that KPCA extracts more informative features and is more robust to the noise contained in the hyperspectral data. Classification results of the EMP built with the KPCA significantly outperforms those obtained with the EMP with the PCA.

Practical conclusions are that, where possible, the KPCA should be used in preference to the PCA because the KPCA extracts more useful features for the purpose of classification. However, one limitation of the KPCA is its computational complexity, related to the size of the kernel matrix, which can limit the number of samples used. In our experiments, 5000 random samples were used leading to satisfactory results.

Our current investigations are oriented to non-linear Independent Component Analysis, such as Kernel ICA [47], for the construction of the EMP and to a sparse KPCA in order to reduce the complexity [48].

## VII. ACKNOWLEDGMENT

The authors thank the reviewers for their many helpful comments. This research was supported in part by the Research Fund of the University of Iceland and the Jules Verne Program of the French and Icelandic governments (PAI EGIDE).

## REFERENCES

- [1] M. Fauvel, J. Chanussot, and J. A. Benediktsson, *Decision fusion for hyperspectral classification*, in *Hyperspectral Data Exploitation*, C. Chang, Ed. John Wiley and Sons, New York, 2007.
- [2] C. Chang, *Hyperspectral imaging. Techniques for spectral detection and classification*. Kluwer Academic, 2003.
- [3] C. Lee and D. A. Landgrebe, "Analyzing high dimensional multispectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 4, pp. 792–800, July 1993.
- [4] L. Jimenez and D. A. Landgrebe, "Supervised classification in high dimensional space: geometrical, statistical and asymptotical properties of multivariate data," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, no. 1, pp. 39–54, February 1993.
- [5] G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Inf. Theory*, vol. IT-14, pp. 55–63, January 1968.
- [6] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. New Jersey: John Wiley and Sons, 2003.
- [7] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*. Springer, 1999.
- [8] N. Keshava, "Distance metrics and band selection in hyperspectral processing with application to material identification and spectral libraries," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, pp. 1552–1565, July 2004.
- [9] C. U. Unsalan and K. L. Boyer, "Linearized vegetation indices based on a formal statistical framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1575–1585, July 2004.
- [10] K.-S. Park, S. Hong, P. Park, and W.-D. Cho, "Spectral content characterization for efficient image detection algorithm design," *EURASIP Journal on Advances Signal Processing*, vol. 2007, p. 14 pages, 2007.

- [11] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, pp. 153–189, 1997.
- [12] P. Somol, P. Pudil, J. Novovicova, and P. Paclik, "Adaptive floating search methods in feature selection," *Patterns recognition letters*, vol. 20, pp. 1157–1163, 1999.
- [13] S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1360–1367, 1994.
- [14] B. Guo, S. R. Gunn, R. I. Damper, and J. D. B. Nelson, "Band selection for hyperspectral image classification using mutual information," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 4, pp. 522–526, 2006.
- [15] H. Kwon and N. M. Nasrabadi, "A comparative analysis of kernel subspace target detectors for hyperspectral imagery," *EURASIP Journal on Applied Signal Processing*, vol. 2007, p. 13 pages, 2007.
- [16] M. Lennon, "Méthodes d'analyse d'images hyperspectrales. Exploitation du capteur aéroporté CASI pour des applications de cartographie agro-environnementale en bretagne," Ph.D. dissertation, Université de Rennes 1, 2002.
- [17] L. Journaux, X. Xavier, I. Foucherot, and P. Gouton, "Dimensionality reduction techniques: an operational comparison on multispectral satellite images using unsupervised clustering," in *IEEE 7<sup>th</sup> Nordic Signal Processing Symposium*, 2006.
- [18] M. Lennon, G. Mercier, M. Mouchot, and L. Hubert-Moy, "Curvilinear component analysis for nonlinear dimensionality reduction of hyperspectral images," in *SPIE*, 2001.
- [19] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent component analysis*. Wiley Series, 2001.
- [20] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
- [21] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Kernel principal component analysis for feature reduction in hyperspectral images analysis," in *IEEE 7<sup>th</sup> Nordic Signal Processing Symposium*, 2006.
- [22] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 2, pp. 309–320, February 2001.
- [23] J. A. Benediktsson, M. Pesaresi, and K. Arnason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1940–1949, September 2003.
- [24] P. Soille, *Morphological Image Analysis, Principles and Applications- 2nd edition*. Springer, 2003.
- [25] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 466–479, 2005.
- [26] J. A. Benediktsson, J. A. Palmason, and J. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 480–491, Mar. 2005.
- [27] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson, and J. Chanussot, "Fusion of morphological and spectral information for classification of hyperspectral urban remote sensing data," in *Proc. IEEE International Geoscience and Remote Sensing Symposium. IGARSS'06. Proceedings*, Jul. 2006.
- [28] M. Fauvel, J. Chanussot, J. A. Benediktsson, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," in *Proc. IEEE International Geoscience and Remote Sensing Symposium. IGARSS'07. Proceedings*, Jul. 2007.
- [29] T. Géraud and J.-B. Mouret, "Fast road network extraction in satellite images using mathematical morphology and Markov random fields," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 16, pp. 2503–2514, 2004.
- [30] X. Jin and C. H. Davis, "Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 14, pp. 2196–2206, 2005.
- [31] J. A. Palmason, "Classification of hyperspectral data from urban areas," Master's thesis, Faculty of Engineering - University of Iceland, 2005.

- [32] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsh, K.-R. Müller, G. Rätsch, and A. J. Smola, “Input space versus feature space in kernel-based methods,” *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1000–1017, May 1999.
- [33] K.-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, “An introduction to kernel-based learning algorithms,” *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–202, March 2001.
- [34] N. Aronszajn, “Theory of reproducing kernel,” Harvard University, Division of engineering sciences, Tech. Rep. 11, 1950.
- [35] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2002.
- [36] J. A. Lee and M. Verleysen, *Nonlinear dimensionality reduction*. Springer, 2007.
- [37] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.
- [38] M. Fauvel, “Spectral and spatial methods for the classification of urban remote sensing data,” Ph.D. dissertation, Institut National Polytechnique de Grenoble, <http://tel.archives-ouvertes.fr/tel-00258717/fr/>, 2007.
- [39] V. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.
- [40] C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
- [41] M. Fauvel, J. Chanussot, and J. A. Benediktsson, “Evaluation of kernels for multiclass classification of hyperspectral remote sensing data,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP’06. Proceedings, May 2006.
- [42] Q. Yong and Y. Jie, “Modified kernel functions by geodesic distance,” *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 16, pp. 2515–2521, 2004.
- [43] C. W. Hsu and C. J. Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Netw.*, vol. 13, pp. 415–425, Mar. 2002.
- [44] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” *IEEE Trans. Geosci. Remote Sens.*, vol. 42, pp. 1778–1790, Aug. 2004.
- [45] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [46] G. M. Foody, “Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy,” *Photogrammetric Engineering & Remote Sensing*, vol. 70, no. 5, pp. 627–633, May 2004.
- [47] F. R. Bach and M. I. Jordan, “Kernel independent component analysis,” *Journal of Machine Learning Research*, pp. 1–48, 2002.
- [48] T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*. MIT Press, 2001.