Spectral and Spatial Methods for the Classification of Urban Remote Sensing Data

Doctoral Thesis

Mathieu FAUVEL

Thesis Supervisors: Jocelyn CHANUSSOT and Jon Atli BENEDIKTSSON

Grenoble Institute of Technology

Faculty of Engineering - University of Iceland

November 2007

Spectral and Spatial Methods for the Classification of Urban Remote Sensing Data

by

Mathieu Fauvel.

Thesis committee:

M.	Henri MAÎTRE	Chair
M.	Grégoire MERCIER	Referee
M.	Sebastiano B. SERPICO	Referee
M.	Albert BIJAOUI	Examiner
M.	Jordi INGLADA	Examiner
M.	Johannes R. SVEINSSON	Examiner
M.	Jocelyn CHANUSSOT	Supervisor
M.	Jon A. BENEDIKTSSON	Supervisor

Contents

Acknowledgements - Remerciements

I have done my Phd during the years 2004-2007, at both the GIPSA-lab and the University of Iceland, under the supervision of Jocelyn Chanussot and Jon Atli Benediktsson. During that period I received a lot of help and motive from family, friend and colleague (some of them become friend). Choosing who I would like to first thank is not easy. I should start by thanking Jocelyn Chanussot, that gives me the opportunity to do that thesis in a perfect environment for research, from Grenoble to Reykjavik through Denver and Barcelona. Similarly, I thank Jon Atli Benediktsson for his support and motivation during my stay at the University of Iceland and the last months of the thesis. I also thank Jocelyn and Jon for making them-self available whenever I needed. Working with them was a very pleasant and educational experience. I hope our collaboration will continue.

I thank Henri Maître for accepting to preside the committee and for its interest to my work. I would like to address my acknowledgements to Jordi Inglada and Johannes R. Sveinsson for their helpful comments and their criticisms. Thanks to Albert Bijaoui for its astrophysician analyze of my work. For reading deeply my thesis, I would to like thank Sebastiano B. Serpico and Grégoire Mercier, the reviewers of my thesis. I also thank Grégoire for all the discussions we had, and hope for another ones. Maybe at a Starbucks?

Now, the following will be in french, sorry. English returns for the introduction!

Mes remerciements vont ensuite vers les personnes que j'ai été contraint de fréquenter quotidiennement au GISPA-lab: en premier lieu, les habitants des BOXS, Matthieu et Seb, sans qui je n'aurai sûrement pas survécu aux chaleurs infernales des étés Grenoblois (vive la clim et le pastis). Je tiens à témoigner ma sincère amitié envers Caroline, Grégoire et encore Matthieu, j'espére que les chemins de randonnées ou de ski de fond ne nous éloignerons pas trop. Merci à Cédric, dit le Suricate des Sables, pour cette année passée trop vite. Merci aussi à Cédric, l'autre, pour ses précieux conseils et sa disponibilité et à Julien pour son calme, même après avoir quitté les BOXS. Merci à tous ceux qui font du GIPSA-lab un endroit agréable: Barbara, Pierre, Moussa et son iphone, Nicolas, Jérôme, Bidou et les autres que j'oublie sûrement.

En dehors du labo, ces trois dernières années, j'ai aussi patiné derrière une balle. Je voudrais en profiter pour remercier mes compagnons de galère, avec qui j'ai passé une grande parties de mes week end dans les trains, hôtels et salles polyvalentes des quatre coins de la France : Claude, Lionel, Tony, Pierre, Bertrand, Johan, Dédé, Bib, Constant, Lydie, les Gillet, les Rousset, les Lapicerella ainsi que tous les jeunes qui ont bien voulu me subir en tant qu'entraîneur.

Merci à mes parents, toujours là aux moments importants, à mes grand parents pour m'avoir nourri exclusivement à base de canard. Merci aussi à Tantine, Cousine et Hubert pour leur aide précieuse. A tous les Marmouyet, ils sont nombreux à force, merci pour vos encouragements.

Enfin, merci à celle qui m'a supporté, avec mon autisme, ma thèse et mon hockey et qui est restée malgré tout. A deux, ce fut plus facile.

Mathieu

Introduction

THE classification of optical urban remote-sensing data has become a challenging problem, due to recent advances in remote sensor technology. Spatial resolution is now as high as 0.75 meter for several satellites, *e.g.*, IKONOS, QUICKBIRD, and soon PLEIADES: For the same location, a panchromatic image with 0.75-meter spatial resolution and multispectral data with 3-meter spatial resolution are available. Hyperspectral sensors can simultaneously collect more than a hundred spectral bands of an area, with increasing spatial resolution, *e.g.* 1.5 meter for airborne sensors. The problem of detecting or classifying *urban areas* in remotely-sensed images with lower spatial resolution has now become the even bigger problem of analysing such urban areas. Figure 0.1 presents two data sets, with low and high spatial resolution respectively. In the left-hand image, the city of Reykjavik can be made out, but the structures of the city are not identifiable. The right-hand image represents one small section from the left-hand image, with ten times higher resolution.

Thanks to the finer resolution, urban structures are now accessible: Road, building, house, green space, bridge, car, and so on can be discerned *visually.* Furthermore, the high spectral resolution allows detailed physical analysis of the structures. An example of hyperspectral ROSIS data is given in Figure 0.2; for each set of data a very detailed description of the urban structures is possible, while using the three bands it is easier to distinguish the vegetation and man-made constructions.

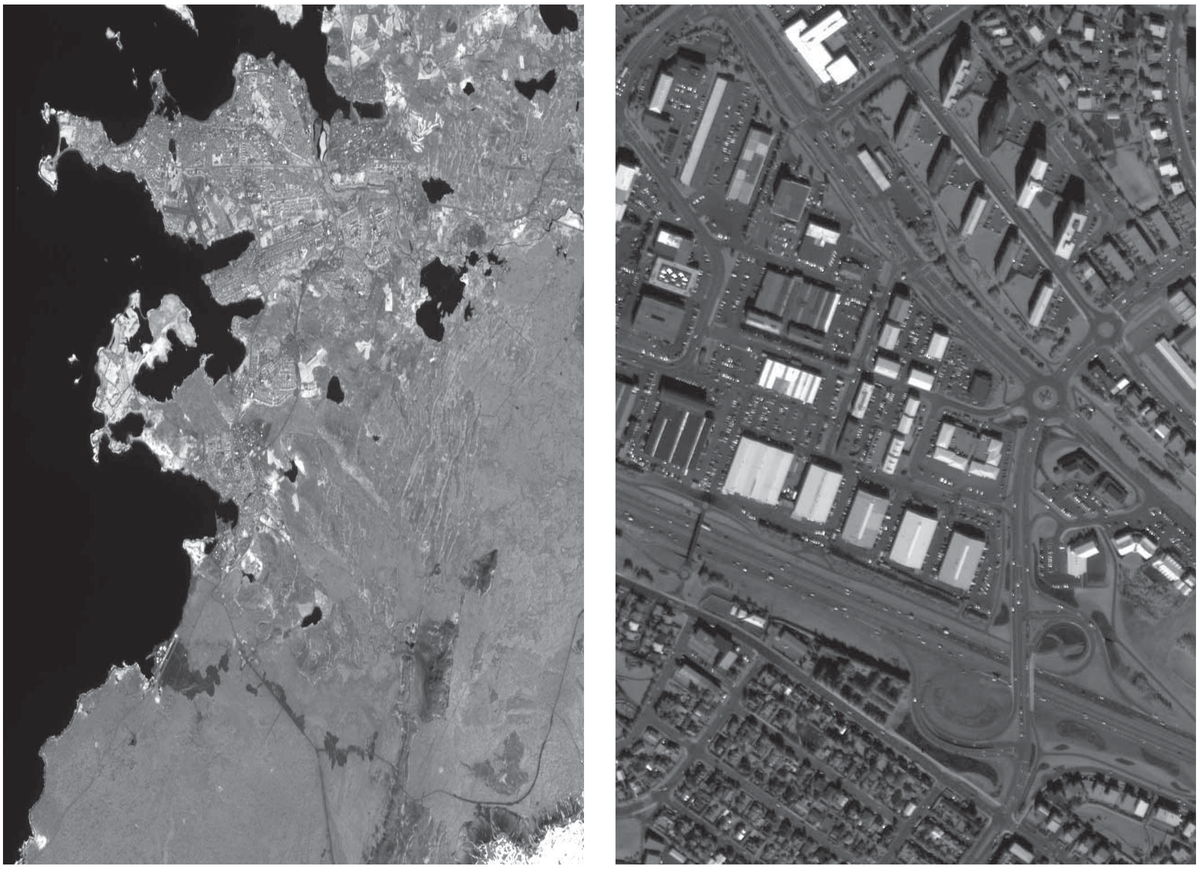
The detection and classification of such structures have many application [94]:

- Mapping and tracking: Classification algorithms can be used to create thematic maps, to follow the evolution of urban area growth. At a coarse level, it is possible to extract the *city network* to evaluate the connections between each zone of the city (suburbs, center, etc.). Change detection can be carried out over a number of years to analyze the way a city evolves.
- Risk management: By identifying residential, commercial, and industrial areas, it is possible to analyze the sensitivity of the different areas of a city to natural risks. A geophysical analysis makes it possible to obtain more accurate knowledge of where geological hazards might occur. After a natural (or other) disaster, remote sensing of urban areas can be used to guide the assistance effort.
- Social problems: By analyzing the spatial arrangement of the city: dense areas, open areas, etc. The distribution of critical services (*e.g.* hospitals, schools) can also be studied.
- Ecological problems: A major problem in urban area is preserving open spaces, which can be readily detected using, *e.g.*, multispectral data from IKONOS or QUICKBIRD.

Remote-sensing data are characterized by the dual nature of the information they provide: they can be viewed as a collection of spectra (the spectral domain) where each pixel is a vector and the components are the reflectance values at a certain wavelength; or they can be regarded as a collection of images (the spatial/image domain) acquired at different wavelengths.

First attempts to analyze urban area remote-sensing data used existing methodologies – techniques developed for land remote sensing, based on signal modeling [80]. Each pixel-vector is regarded as a signal, and signal-based processing algorithms are applied, mostly based on statistical modeling. The traditional approach for classifying remote-sensing data may be summed up as [81]: from the original data set, a feature reduction/selection step is performed according to the classes under consideration, then classification is carried out using these extracted features. When dealing with a small training set, an iterative procedure is usually applied, by adding semi-labelled samples to the training set according to certain criteria (un-labelled samples that are labelled after the first classification step). Another possible modification is the inclusion of contextual information: the classification algorithm uses not just the pixel itself, but also its neighbors. Markov Random Fields (MRF) are usually used within a statistical framework [86]. A survey of the current techniques for the analysis of remotely-sensed data can be found in [108].

Surprisingly, few classification algorithms exploit the spatial information contained in the remote-sensing data, the reason being the usually low resolution of the data. However, high spatial resolution data contain a lot of contextual information: for a given pixel we can extract the size, shape, and gray-level distribution of the structure to which it belongs. This information will not be the same if the pixel belongs to a roof or to a green area. This is also a way to discriminate various structures made of the same materials. If spectral information alone is used, the roofs of a private house and of a larger building will be



(a) (b)

Figure 0.1: *Satellite images: (a) Spot3 data of the Reykjavik area, Iceland – spatial resolution 10 m; (b) IKONOS data for part of central Reykjavik – spatial resolution 1 m.*

detected as the same type of structure. But using additional spatial information – the size of the roof, for instance – it is possible to classify these into two separate classes. Consequently, a joint spectral/spatial classifier is needed to classify urban remote-sensing data better. Landgrebe and co-workers were probably the first to propose a joint classifier, the well-know ECHO [80]. Later, Landgrebe and Jackson proposed an iterative statistical classifier based on MRF modelling [68]. However, MRF modelling suffers from the high spatial resolution: neighboring pixels are highly correlated and the standard neighbor system definition does not contain enough samples to be effective. Unfortunately, a larger neighbor system entails intractable computational problems, thereby limiting the benefits of conventional MRF modelling. Furthermore, algorithms involving MRF-based strategies traditionally require an iterative optimization step, such as simulated annealing, which is extremely time consuming. Yet the use of spatial information does result in improved classification accuracy and ought to be considered.

Benediktsson *et al.* have proposed using advance morphological filters as an alternative way of performing joint classification [8]. Rather than defining a crisp neighbor set for every pixel, morphological filters make it possible to analyze the neighborhood of a pixel according to the structures to which it belongs. This approach has given good results for various urban data sets [8, 28, 7]. But the construction of the features vector, the morphological profile (MP), can result in a high-dimensional vector where the spectral information is not fully exploited. The problems arising for classification of the MP are the same as those for hyperspectral data: the *curse of dimensionality*.

When the data is represented by a vector in a vector space of high dimensionality – say more than

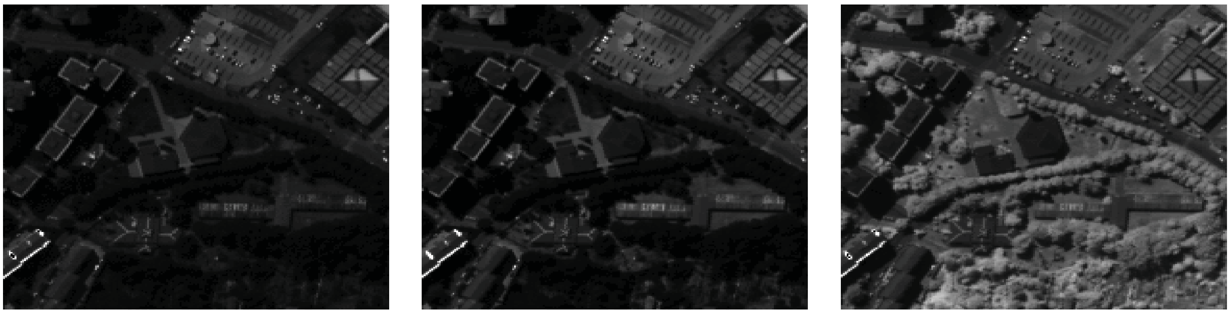


Figure 0.2: *ROSIS data: (a) Band 20 (b) Band 60 (c) Band 90.*

50 – theoretical and practical problems arise. Major instances are the difficulties of statistical estimation (aggravated by the relatively small size of the training set), the Hughes phenomenon, and redundancy in the vector component information. More details are given at the start of Chapter 1, but problems (which are not specific to urban areas) related to the dimensionality of the features vector have to be handled carefully because of *the requirement to add contextual information to the original spectral information*. However, traditional approaches such as statistical and neural methods may fail with high-dimensional data, and hence not be appropriate for use where spatial information is included.

The problem of dimensionality has traditionally been tackled by the use of Features Extraction (FE) algorithms [80, 49]. Standard unsupervised techniques are Principal Component Analysis (PCA), Independent Component Analysis (ICA); supervised ones include Discriminant Analysis Feature Extraction (DAFE), Decision Boundary Feature Extraction (DBFE), and Non-parametric Weighted Feature Extraction (NWFE) [80, 67]. It is preferable to use supervised transformations because the projection into the subspace minimizes a classification error criterion. However, the performance of such methods is closely related to the *quality* of the reference or training set. Consequently, unsupervised methods are of interest, but since the criterion to minimize is not related to the classification error, the projection is not optimal for the purpose of classification. However, the aim of feature-reduction algorithms is not necessarily classification, but also representation. Several unsupervised algorithms are used to find a subspace to represent hyperspectral data, for visualization or processing. The latter is very interesting, since many image-processing algorithms are only defined for mono-valued images. Using feature-reduction algorithms, it is possible to extract mono-valued images and, for instance, apply suitable algorithms to extract spatial information [7].

Analyzing remote-sensing data over urban areas presents multiple difficulties:

- The large size of the data is a problem,
- The ground truth is limited,
- Spatial/contextual information is needed,
- The extraction of the spatial information is difficult.

Another difficulty lies in the evolution in the properties of the data itself: spectral and spatial resolutions have increased a great deal since the ’70s, as well as the area covered by the sensor, and the magnitude of each spectral value – from the original 8-bits to 16-bits now. Hence algorithms well suited to a certain type of data will not necessarily be suitable for another type of data. For instance, one particular statistical model may hold good for a given sensor, but may be unsuitable for another sensor. The increasing complexity of remote-sensing data may explain why no significant improvement in classification accuracy for satellite image classification has been reported for quite some time [128]. This ought to encourage the use of a methodology that is based on sensor-invariant assumptions.

One very common assumption is the *Gaussian assumption* that assumes the data exhibit a Gaussian distribution [80]. This lies at the very foundation of the popular Gaussian Maximum Likelihood classifiers.

However, this assumption is not necessarily satisfied for all data sets – for instance, the Morphological Profile or multi-source data are examples of non-Gaussian data. Another sensor-based assumption is traditionally made when the user defines a fixed neighborhood for every pixel. A pixel's neighbors depend both on the structures to which it belongs and on the spatial resolution of the image.

One other consequence of the evolution in sensors is the availability of several data sets for the same area. In the '90s, Benediktsson and co-workers showed that the use of several sources increased classification accuracy [6, 10, 9, 13, 11]. There is a need for a general framework for incorporating the different sources into the classification process. Two approaches can be adopted, depending on whether the multi-source data is used before classification or after. The first deals with the sources themselves while the latter deals with the classifier outputs. If we consider the spatial information as another information source, spatial and spectral data fusion is possible.

The work presented in this thesis is an attempt to propose tools and methodologies that do not use either of the two previous assumptions and solve the problem of joint classification of remote-sensing data over urban areas. As a priority, classification should be achieved using both the spectral and spatial information available. Hence the spatial information extraction process should be sensor-independent, implying an adaptive methodology for analyzing the inter-pixel dependency of the image. Then the classification process should not involve sensor-dependent modeling. To assess the effectiveness of the method, we use various data sets: very high spatial resolution panchromatic data from two different sensors (IKONOS and PLEIADES) were used, as well as three different hyperspectral data sets, two from airborne sensors and one from a satellite sensor.

This work is divided into six chapters:

1. The first chapter of this thesis addresses the problem of feature extraction. A non-linear unsupervised feature-extraction algorithm is proposed to overcome the limitations of traditional PCA. The objective is to reduce the dimension of the data without any ground truth for classification using conventional classifiers that are adversely affected by dimensionality.
2. For analyzing data in the spatial domain, algorithms working on the structures of the image, such as Mathematical Morphology, seem very appropriate: they are able to analyze structures regardless of size. The morphological profile [101, 8] makes sensor-independent analysis possible in the spatial domain, though a degree of refinement is still possible taking the sensor's spatial resolution into account. In Chapter 2, the morphological profile is reviewed, together with its extension to the multi-spectral case. Because of inherent properties of the morphological filters, the morphological profile is unable to provide a full description of the scene. Moreover, morphological filters do not exist[1] for multi-dimensional images. An alternative approach based on self-complementary filters is proposed. The objective is to overcome the limitations of the morphological profile, without losing its advantages. The proposed filter makes it possible to analyze image structures independently of their local contrast.
3. The Support Vector Machine (SVM) is presented in Chapter 3 as a classifier suited to the problem of classification of remote-sensing data. It is robust to the dimensionality of the data and has good generalization performance, even in the situation of a limited training set. Moreover, classification does not involve any assumptions about data distribution. Its superiority over standard classifiers (statistical and neural) is studied using simulated and real data sets. Then several training procedures are investigated. In each case, SVM outperforms others classifiers in terms of classification accuracy.
4. Chapter 4 deals with the transferability of the decision function obtained by the SVM. As explained in the previous paragraph, the spectral characteristics of classes may change between two sensors, and also between two different locations – due to variations in illumination, for example. Thus the decision function obtained for a given data set may be not suitable for a new data set. In this work, we propose constructing the decision function using spatial features, which should be invariant for urban area, and constructing an invariant decision function.

[1] With exactly the same properties as for flat images.

Using the spatial features proposed in Chapter 2, a joint spectral/spatial classification based on SVM is proposed, the spectro-spatial SVM. A kernel formulation is suggested for such a purpose.

5. Decision fusion is addressed in Chapter 5. One way to improve the classification of urban area data is to take advantage of several existing classifiers. A framework is proposed that can handle several classifiers with various type of output. Fuzzy set and fuzzy logic are used to deal with uncertainty and the conflictual situations that may ensue. A specific application for an SVM-based classifier is then proposed.
6. Another way to use the spatial and spectral information by means of multi-source fusion is proposed in Chapter 6. Spectral and spatial information are seen as two separate sources. The fusion scheme proposed is in two steps: first, a supervised feature-reduction algorithm is used to remove possible redundancy, and then feature vectors are constructed by concatenation of the extracted spatial and spectral information.

These chapters are organized into three parts, each containing two chapters. Part 1 relates to feature extraction, whether in the spectral or spatial domain. Part 2 is devoted to the classification algorithms, especially the SVM, and to the inclusion of the spatial information. Part 3 deals with data fusion. Two levels of data fusion are investigated: at decision level and at data level. Finally, the work ends with some conclusions and prospects.

Kernel methods presented in Chapters 1, 3, and 4 use the so-called 'kernel trick'. An overview of kernel method theory is given in Appendix A. The equivalence between a positive semi-definite function and an inner product in Reproducing Kernel Hilbert Space is detailed. Then the 'Representer Theorem' is stated with its proof. We give a practical example of the 'kernel trick' to compute the smallest enclosing hypersphere. Appendix B explains how the accuracies are computed in this thesis. The confusion matrix is presented and accuracy estimators are summarized: overall accuracy (OA), average accuracy (AA), and the Kappa coefficient. In the final appendix, all of the data used in this work are presented, together with their training and testing sets.

Part I

Feature Extraction

Chapter 1

Spectral Feature Extraction

Abstract

The chapter deals with spectral feature extraction for hyperspectral data. Problems occurring with high-dimensional space are first briefly explained and the nature of feature extraction algorithms based on signal theory are presented. Starting from some limitations of the state-of-the-art algorithms, an unsupervised non-linear feature extraction algorithm, namely kernel principal component analysis, is detailed. Then, experiments on real hyperspectral data are conducted for the purpose of classification. Two different classifiers, a neural network and a linear support vectors machine, are used to classify the data using the extracted features. Experimental results show the effectiveness of the non-linear approach, which makes it possible to extract more informative features. Furthermore, it is found that the extracted features make the classification problem more linearly separable.

Contents

IN THE SPECTRAL DOMAIN, pixels are vectors where each component contains specific wavelength information provided by a particular channel [23]. The size of the vector is related to the number of bands that the sensor can collect. For hyperspectral data, 200 or more spectral bands of a same scene may be available, while for multispectral images about ten bands are accessible, and for panchromatic images, only one.

With increasing dimensionality of the data in the spectral domain, theoretical and practical problems arise. The idea of the *dimension* is intuitive, driven by experiments in one-, two- or three-dimensional space, and geometric concepts that are self-evident in these spaces do not necessarily apply in higher-dimensional space [80, 75]. For example, normally-distributed data have a tendency to concentrate in the tails, which seems to be contradictory with its bell-shaped density function. For the purpose of classification, these problems are related to the *curse of dimensionality*. In particular, Hughes showed that with a limited training set, beyond a certain limit, the classification accuracy decreases as the number of features increases [66]. 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 phenomenum, *feature selection / extraction* is usually performed as pre-processing to hyperspectral data analysis [80]. Such processing can also be performed for multispectral images, to enhance class separability or to remove a certain amount of noise.

In the following discussion, some well-known problems of hyperspectral data in the spectral domain are discussed (1.1), then the bases of feature selection are presented (1.2). Limitations of the state-of-the-art methods are highlighted and a non-linear version of Principal Component Analysis is presented (1.4). Experiments on the classification of real hyperspectral data are presented (1.5).

1.1 High-dimensional space

In this section, classic problems encountered in high-dimensional space are presented. Theoretical and experimental research sheds some light on these. A great deal of material can be found in [66, 70, 75] and a survey of high-dimensionality data analysis is given in [43]. The main results are expounded without the mathematical formulation, but an understanding of these concepts is essential in order to be able to analyze hyperspectral data:

1. The second-order statistic plays an important role in classification: it has been shown that when the dimensionality increases, considering only the variance of multivariate data led to significantly better classification results than considering only the mean [70, 82].
2. In high-dimensional space, normally-distributed data tend to concentrate in the tails, while uniformly-distributed data tend to the corners (the 'concentration of measure' phenomenon [43]).
3. *Hughes effect*: with a limited number of training samples, there is a classification accuracy penalty as the number of features increases beyond a certain point [66].
4. It has been proved that for good estimation of the parameters, the required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier [59].

From 2, the local neighborhood is very likely to be empty, making statistical estimation difficult and 1 is difficult to satisfy in a practical situation. Four suggests that the number of available training samples should increase with dimensionality, but in general this is not possible, especially with remote-sensing data, and this problem leads to 3. Hopefully, since high-dimensional spaces are mostly empty, the multivariate data can usually be represented in lower-dimensional space without losing significant information in terms of class separability. Furthermore, the projected data tend to be normally distributed [62, 41], thus enable the use of a conventional classifier based on the Gaussian assumption. Based on these considerations, several feature selection methods were proposed [80]. In the next section, we present some of the bases for dimensionality reduction.

1.2 Dimensionality reduction: feature selection – feature extraction

Dimensionality reduction is a technique aimed at reducing the dimensionality of data by mapping them onto another space of a lower dimensions, without discarding any meaningful information. Furthermore, *meaningful information* is defined according to the final processing: classification, detection, representation, and so on. Feature selection is the technique of selecting a subset of relevant features, while feature extraction is a method of combining features – both in order to obtain a relevant representation of the data in a lower-dimensional space. Such strategies were initially designed in accordance with both the specific characteristics of the remote sensors and the objectives (*e.g.* agricultural or geological context), such as the *Tasseled Cap* or the *NDVI* [80]. However, each transformation is sensor-dependent and the physical analysis associated with each transformation can be untractable for hyperspectral data.

Transformations based on statistical analysis have already proved to be useful for classification, detection, identification, or visualization of remote-sensing data [23, 109, 76, 135]. 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 is correlated with how well the training set represents the whole data set. Moreover, this transformation can be extremely time-consuming. 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 square sense. Note that both the unsupervised and supervised cases can be also divided into *linear* and *non-linear* algorithms.

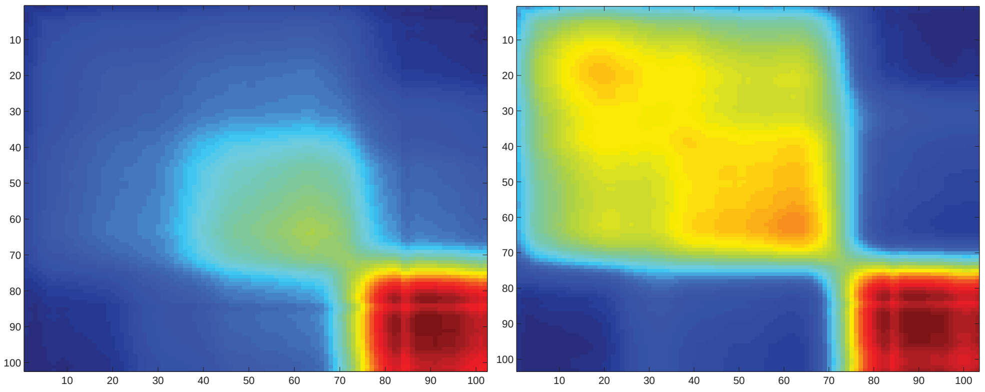
PCA plays an important rôle in the processing of remote-sensing images. Even though its theoretical limitations for hyperspectral data analysis have been pointed out [84, 80], in a practical situation the results obtained using PCA are still competitive for the purpose of classification [72, 85]. The 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 this method. Section 1.4 presents a non-linear version of PCA, namely *Kernel Principal Component Analysis* (KPCA), which considers higher-order statistics.

1.3 Principal Component Analysis

1.3.1 Principles

PCA is a classic technique in statistical data analysis. It aims to de-correlate the variables $x_1, \dots, x_n$ of a given random vector $\mathbf{x} \in \mathbb{R}^n$. The variables of the projected vector $\mathbf{y} = \mathbf{P}^t\mathbf{x}$ are uncorrelated with the other variables. This means that its covariance matrix $\Sigma_{\mathbf{y}} = \mathbb{E}\left[\mathbf{y}_c\mathbf{y}_c^t\right]$ is diagonal, where $\mathbf{y}_c$ is the centered vector $\mathbf{y}$. The computation of the covariance matrix can be written as:

$$\begin{aligned}
\Sigma_{\mathbf{y}} &= \mathbb{E}\left[\left(\mathbf{y}-\mathbf{m}_y\right)\left(\mathbf{y}-\mathbf{m}_y\right)^t\right] \\
&= \mathbb{E}\left[\left(\mathbf{P}^t\mathbf{x}-\mathbf{P}^t\mathbf{m}_x\right)\left(\mathbf{P}^t\mathbf{x}-\mathbf{P}^t\mathbf{m}_x\right)^t\right] \\
&= \mathbf{P}^t\mathbb{E}\left[\left(\mathbf{x}-\mathbf{m}_x\right)\left(\mathbf{x}-\mathbf{m}_x\right)^t\right]\mathbf{P} \\
&= \mathbf{P}^t\Sigma_{\mathbf{x}}\mathbf{P}.
\end{aligned} \tag{1.1}$$



(a) *Pavia Center* (b) *University Area*

Figure 1.1: *Covariance matrix for two hyperspectral data sets acquired using the same sensor.*

$\Sigma_{\mathbf{x}}$ is a real-valued symmetric matrix of finite dimension. By the *spectral theorem*, $\Sigma_{\mathbf{x}}$ can be diagonalized by an orthogonal matrix $\mathbf{M}$ ($\mathbf{M}^t = \mathbf{M}^{-1}$) : $\mathbf{M}^{-1}\Sigma_{\mathbf{x}}\mathbf{M} = \Sigma_{\mathbf{y}}$ (from (1.1)). By identification, $\mathbf{P}$ is an orthonormal matrix which is found by solving the eigenvalues (λ) problem with unitary norm condition on the eigenvectors ($\mathbf{v}$):

$$\begin{aligned} \lambda\mathbf{v} &= \Sigma_{\mathbf{x}}\mathbf{v} \\ \|\mathbf{v}\|_2 &= 1. \end{aligned} \tag{1.2}$$

It turns out that $\mathbf{P}$ consists of the set of all eigenvectors $\mathbf{v}$ of $\Sigma_{\mathbf{x}}$, with one eigenvector per column.

1.3.2 Reducing the dimensionality using PCA

The eigenvalues obtained represent the variance of the variable $\mathbf{y}$, *i.e.*, var($\mathbf{y}_i$)= λ_i. They are stored in decreasing order $\lambda_1 > \lambda_2 \cdots > \lambda_n$ and $\langle \mathbf{e}_{\lambda_i}, \mathbf{e}_{\lambda_j} \rangle_{\mathbb{R}^n} = \delta_{ij}$ [1]. Feature reduction is performed using the following postulate: *the greater the variance, the greater the contribution to the representation.* Thus, variables associated with high eigenvalues need to be considered and should remain after feature reduction. The problem lies in selecting sufficient principal components so that the reconstruction error is low. It can be shown [67] that the error in reconstruction, in the mean square sense, of $\mathbf{x}$ using only the k first principal components is

$$\text{MSE} = \sum_{i=k+1}^{n} \lambda_i. \tag{1.3}$$

Therefore, k is chosen in order to make the MSE fall below a given threshold t_{pca}, usually 5% or 10% of the total variance:

$$\frac{\sum_{i=k+1}^{n} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq t_{pca}. \tag{1.4}$$

Note this strategy is optimal for the purpose of representation [59]. As said in Section 1.2, PCA is an unsupervised algorithm which objective is to represent the data in a lower dimensional space withtout discarding meaningful information. It do not use a criterion which is related to the classification error.

[1] $\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$

1.3.3 Computing the PCA

The PCA is related to the diagonalization of the auto-correlation matrix of the initial random vector, which is estimated as:

$$\Sigma_{\mathbf{x}} = \mathbb{E}\left[(\mathbf{x}-\mathbf{m}_x)(\mathbf{x}-\mathbf{m}_x)^t\right] \approx \frac{1}{\ell-1}\sum_{i=1}^{\ell}\left(\mathbf{x}^i-\mathbf{m}_x\right)\left(\mathbf{x}^i-\mathbf{m}_x\right)^t \tag{1.5}$$

and the mean value is estimated:

$$\mathbf{m}_x = \mathbb{E}(\mathbf{x}) \approx \frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{x}^i \tag{1.6}$$

where ℓ is the number of observed data. Algorithm 1 presents a pseudo code for PCA.

This algorithm was applied in [135], where the authors derived the *NDVI* by means of PCA using the near-infrared and red IKONOS bands. In their experiments, the statistical processing made a physical interpretation possible. However, in general, such interpretation is difficult with hyperspectral data.

Figures 1.1(a) and 1.1(b) present the covariance matrices for two hyperspectral data sets. Solving the eigenvalues problem (1.2) yields the results reported in Table 1.1. Regarding the cumulative eigenvalues, in each case three principal components reach 95% of total variance. After PCA, the dimensionality of the new representation of *University Area* data set is 3 and for the *Pavia Center* data set, 2. This means that using second-order information, the hyperspectral data can be reduced to a two- or three-dimensional space. But, as experiments in Section 1.5 will show, hyperspectral richness is not fully handled using only the mean and variance/covariance of the data, and more advanced algorithms need to be used, bearing in mind the two advantages of PCA: low complexity and absence of parameters. In the next section, a non-linear version of PCA is presented. The non-linearity is introduced in order to take higher-order statistics into account.

Table 1.1: *PCA: Eigenvalues and cumulative variance in percentages for the two hyperspectral data sets.*

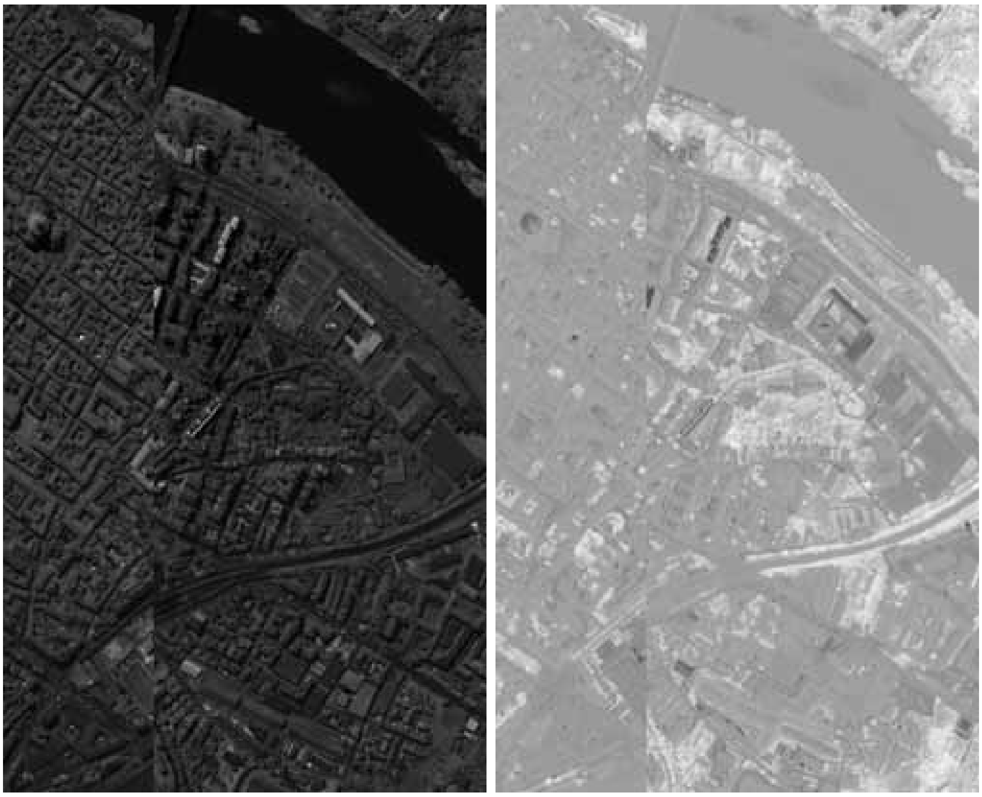
	Pavia Center		*University Area*	
Component	%	Cum. %	%	Cum. %
1	70.24	70.24	58.32	58.32
2	26.07	96.31	36.10	94.42
3	2.81	99.12	4.44	98.86
4	0.32	99.34	0.30	99.16

Algorithm 1 *Principal Component Analysis*

```
$\mathbf{m}_x = \frac{1}{\ell}\sum_{i=1}^{\ell}\mathbf{x}^i$
$\mathbf{x}_c = \mathbf{x} - \mathbf{m}_x$
$\Sigma_{\mathbf{x}} = \frac{1}{\ell-1}\sum_{i=1}^{\ell}\mathbf{x}_c^i(\mathbf{x}_c^i)^t$
Solve: $\lambda\mathbf{v} = \Sigma_{\mathbf{x}}\mathbf{v}$ subject to $\|\mathbf{v}\|_2 = 1$
Project on the first $k$ principal components: $\mathbf{x}_{pc} = \left[\mathbf{v}^1 \,\middle|\, \dots \,\middle|\, \mathbf{v}^k\right]^t \mathbf{x}$
```

1.4 Kernel PCA

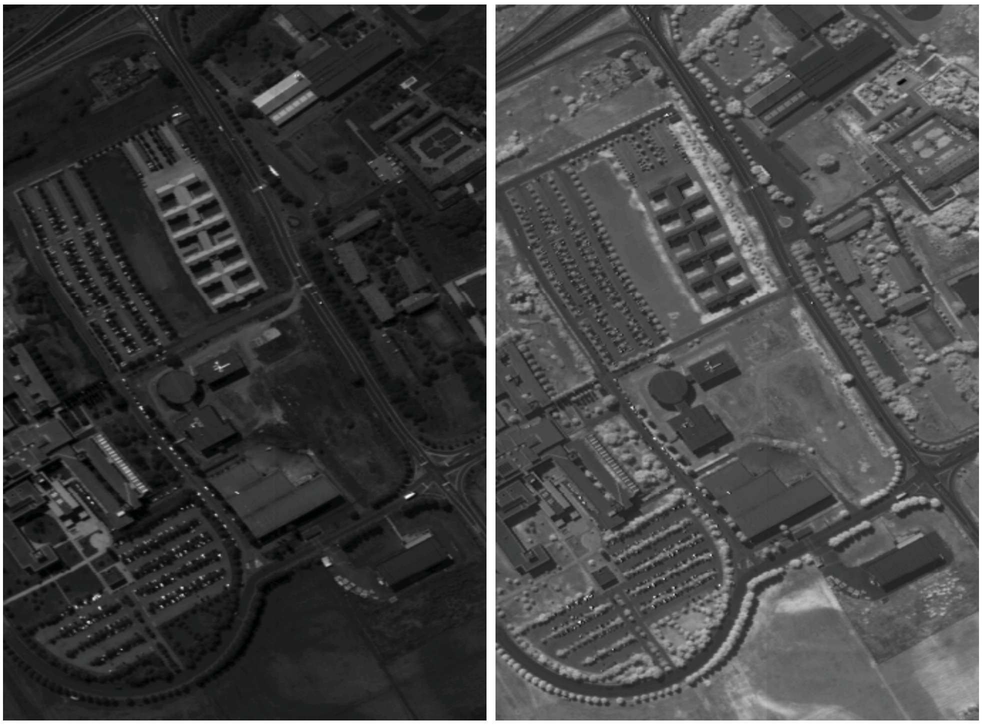
In recent years, several non-linear versions of PCA have been investigated [67, 42]. Traditionally, these have been based on a degree of non-linear processing/optimization in the *input space.* The one discussed



(a) First principal component

(b) Second principal component

Figure 1.2: *First two principal components,* Pavia Center *data set.*



(a) First principal component (b) Second principal component

Figure 1.3: *First two principal components,* University Area *data set.*

in this thesis is based on non-linear mapping to a *feature space* where the original PCA algorithm is then applied [113]. The complexity of the overall processing is reduced thanks to *kernel methods* that make it possible to compute the inner product in a feature space by means of a kernel function in the input space [111, 118, 93]. A brief presentation of kernel method theory can be found in Appendix A.

1.4.1 PCA in the feature space

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

$$\begin{aligned} \Phi : \mathbb{R}^n &\rightarrow \mathcal{H} \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}). \end{aligned} \tag{1.7}$$

Φ 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. In the following, the dot product in the RKHS is denoted by $\langle ., . \rangle_{\mathcal{H}}$ and its associated norm by $\|.\|_{\mathcal{H}}$.

The main idea behind the mathematical formulation is that the data are mapped onto a *subspace* of $\mathcal{H}$ spanned by $\Phi(\mathbf{x}^1), \dots, \Phi(\mathbf{x}^\ell)$. As a result, we are looking at solutions (eigenvectors) generated by the projected samples (see the *Representer Theorem* [110] in Appendix A).

To apply the kernel trick, we must first rewrite the PCA in the feature space in terms of inner product [113]. Considering that the data are centered in the feature space (we will show how in Section 1.4.2) the estimate of the covariance matrix $\Sigma_{\Phi(\mathbf{x})}$ is:

$$\Sigma_{\Phi(\mathbf{x})} = \frac{1}{\ell - 1} \sum_{i=1}^{\ell} \Phi(\mathbf{x}^i)\Phi(\mathbf{x}^i)^t \tag{1.8}$$

and the eigenvalues problem is:

$$\begin{aligned} \lambda \mathbf{v}_\Phi &= \Sigma_\mathbf{x} \mathbf{v}_\Phi \\ \|\mathbf{v}_\Phi\|_\mathcal{H} &= 1. \end{aligned} \tag{1.9}$$

Combining (1.8) and (1.9) leads to:

$$\begin{aligned} \lambda \mathbf{v}_\Phi &= \left(\frac{1}{\ell - 1} \sum_{i=1}^{\ell} \Phi(\mathbf{x}^i)\Phi(\mathbf{x}^i)^t \right) \mathbf{v}_\Phi \\ &= \frac{1}{\ell - 1} \sum_{i=1}^{\ell} \langle \Phi(\mathbf{x}^i), \mathbf{v}_\Phi \rangle_\mathcal{H} \Phi(\mathbf{x}^i). \end{aligned} \tag{1.10}$$

Clearly, $\mathbf{v}_\Phi$ now lies within the span of $\Phi(\mathbf{x}^1), \ldots, \Phi(\mathbf{x}^\ell)$:

$$\mathbf{v}_\Phi = \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}^i). \tag{1.11}$$

Substituting (1.11) into (1.10):

$$\begin{aligned} \lambda \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}^i) &= \frac{1}{\ell - 1} \sum_{i=1}^{\ell} \left\langle \Phi(\mathbf{x}^i), \sum_{j=1}^{\ell} \alpha_j \Phi(\mathbf{x}^j) \right\rangle_\mathcal{H} \Phi(\mathbf{x}^i) \\ \lambda \sum_{i=1}^{\ell} \alpha_i \Phi(\mathbf{x}_i) &= \frac{1}{\ell - 1} \sum_{\substack{i=1 \\ j=1}}^{\ell} \alpha_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j) \rangle_\mathcal{H} \Phi(\mathbf{x}^i). \end{aligned} \tag{1.12}$$

Putting (1.12) into $\sum_{m=1}^{\ell} \langle ., \Phi(\mathbf{x}^m) \rangle_\mathcal{H}$:

$$\lambda \sum_{\substack{i=1 \\ m=1}}^{\ell} \alpha_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^m) \rangle_\mathcal{H} = \frac{1}{\ell - 1} \sum_{\substack{i=1 \\ j=1 \\ m=1}}^{\ell} \alpha_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j) \rangle_\mathcal{H} \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^m) \rangle_\mathcal{H}. \tag{1.13}$$

Defining $\mathbf{K}$ as an $\ell \times \ell$ matrix by $K_{ij} := \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j) \rangle_\mathcal{H}$ and $\boldsymbol{\alpha}$ as the vector generated from α_i, (1.13) can be written as:

$$\lambda \mathbf{K} \boldsymbol{\alpha} = \frac{1}{\ell - 1} \mathbf{K}^2 \boldsymbol{\alpha}. \tag{1.14}$$

Thus to find the eigenvectors, one has to find $\boldsymbol{\alpha}$ by solving (1.14). However, it has been shown in [113, 112] that it is sufficient to solve the following eigenvalue problem for non-zero eigenvalues:

$$\lambda \boldsymbol{\alpha} = \frac{1}{\ell - 1} \mathbf{K} \boldsymbol{\alpha}. \tag{1.15}$$

To prove this [113], consider that eigenvectors lying in the subspace spanned by the projected sample are only of interest. Let us denote the set of solutions of (1.14) by $\mathcal{S}^1_{\boldsymbol{\alpha}}$ and the set of solutions of (1.15) by $\mathcal{S}^2_{\boldsymbol{\alpha}}$. Since $\mathbf{K}$ is real and symmetric, its eigenvectors $\boldsymbol{\psi}^i$, $i \in [1, \ell]$, form an orthonormal basis in $\mathcal{H}$ and we have $\nu_i \boldsymbol{\psi}^i = \mathbf{K} \boldsymbol{\psi}^i$ and $\boldsymbol{\alpha} = \sum_{i=1}^{\ell} a_i \boldsymbol{\psi}^i$. (1.14) can then be rewritten as:

$$\begin{aligned} \lambda \mathbf{K} \sum_{i=1}^{\ell} a_i \boldsymbol{\psi}^i &= \frac{1}{\ell - 1} \mathbf{K}^2 \sum_{i=1}^{\ell} a_i \boldsymbol{\psi}^i \\ \lambda \sum_{i=1}^{\ell} a_i \mathbf{K} \boldsymbol{\psi}^i &= \frac{1}{\ell - 1} \sum_{i=1}^{\ell} a_i \mathbf{K}^2 \boldsymbol{\psi}^i \\ (\ell - 1)\lambda \sum_{i=1}^{\ell} a_i \nu_i \boldsymbol{\psi}^i &= \sum_{i=1}^{\ell} a_i \nu_i^2 \boldsymbol{\psi}^i. \end{aligned} \tag{1.16}$$

So for $\mathcal{S}^1_{\boldsymbol{\alpha}}$:

$$\mathcal{S}^1_{\boldsymbol{\alpha}} := \left\{ \boldsymbol{\alpha} = \sum_{i=1}^{\ell} a_i \boldsymbol{\psi}^i \middle| \forall i : (\lambda(\ell-1) = \nu_i) \vee (\nu_i = 0) \vee (a_i = 0) \right\}. \tag{1.17}$$

Following the same reasoning, for $\mathcal{S}^2_{\boldsymbol{\alpha}}$:

$$\mathcal{S}^2_{\boldsymbol{\alpha}} := \left\{ \boldsymbol{\alpha} = \sum_{i=1}^{\ell} a_i \boldsymbol{\psi}^i \middle| \forall i : (\lambda(\ell-1) = \nu_i) \vee (a_i = 0) \right\}. \tag{1.18}$$

Clearly $\mathcal{S}^2_{\boldsymbol{\alpha}} \subset \mathcal{S}^1_{\boldsymbol{\alpha}}$ and $\mathcal{S}^{1/2}_{\boldsymbol{\alpha}}$ can be define as the set of solutions of (1.14) that are not solutions of (1.15):

$$\mathcal{S}^{1/2}_{\boldsymbol{\alpha}} := \left\{ \boldsymbol{\alpha} = \sum_{i=1}^{e} a_i \boldsymbol{\psi}^i \middle| \forall i : \nu_i = 0 \right\} \tag{1.19}$$

where e is the number of eigenvectors corresponding to zero eigenvalue ($e < \ell$). For all $\boldsymbol{\alpha} \in \mathcal{S}^{1/2}_{\boldsymbol{\alpha}}$

$$\begin{aligned}
\mathbf{K}\boldsymbol{\alpha} &= 0 \\
\forall i : \sum_{j=0}^{\ell} K_{ij}\alpha_j &= 0 \\
\forall i : \sum_{j=0}^{\ell} \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j) \rangle_{\mathcal{H}} \alpha_j &= 0 \\
\forall i : \langle \Phi(\mathbf{x}^i), \sum_{j=0}^{\ell} \alpha_j \Phi(\mathbf{x}^j) \rangle_{\mathcal{H}} &= 0 \\
\forall i : \langle \Phi(\mathbf{x}^i), \mathbf{v}_{\Phi} \rangle_{\mathcal{H}} &= 0.
\end{aligned} \tag{1.20}$$

This result shows that all eigenvectors of $\Sigma_{\Phi(\mathbf{x})}$ corresponding to eigenvectors with zero eigenvalues of $\mathbf{K}$ are irrelevant, because they are orthogonal to the subspace of $\mathcal{H}$ spanned by the projected sample. So the resolution of (1.15) gives the whole solution of (1.9). The normalization condition in (1.9) becomes:

$$\begin{aligned}
\langle \mathbf{v}^p_{\Phi}, \mathbf{v}^p_{\Phi} \rangle_{\mathcal{H}} &= 1 \\
\left\langle \sum_{i=1}^{\ell} \alpha^p_i \Phi(\mathbf{x}^i), \sum_{i=j}^{\ell} \alpha^p_j \Phi(\mathbf{x}^j) \right\rangle_{\mathcal{H}} &= 1 \\
\langle \boldsymbol{\alpha}^p, \mathbf{K}\boldsymbol{\alpha}^p \rangle_{\mathcal{H}} &= 1 \\
\lambda_p \langle \boldsymbol{\alpha}^p, \boldsymbol{\alpha}^p \rangle_{\mathcal{H}} &= 1 \\
\|\boldsymbol{\alpha}^p\|_{\mathcal{H}} &= \frac{1}{\lambda_p}.
\end{aligned} \tag{1.21}$$

Thus, the eigenvalue problem (1.9) is solved by:

$$\begin{aligned}
\lambda\boldsymbol{\alpha} &= \mathbf{K}\boldsymbol{\alpha} \\
\|\boldsymbol{\alpha}\|_{\mathcal{H}} &= \frac{1}{\lambda}
\end{aligned} \tag{1.22}$$

and the projection onto the k^{th} principal component:

$$\begin{aligned}
\Phi^k_{kpc}(\mathbf{x}) &= \langle \mathbf{v}_{\Phi(\mathbf{x})}, \Phi(\mathbf{x}) \rangle_{\mathcal{H}} \\
&= \langle \sum_{i=1}^{\ell} \alpha^k_i \Phi(\mathbf{x}^i), \Phi(\mathbf{x}) \rangle_{\mathcal{H}} \\
&= \sum_{i=1}^{\ell} \alpha^k_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}) \rangle_{\mathcal{H}}.
\end{aligned} \tag{1.23}$$

1.4.2 KPCA in the input space

As shown in the previous section, the PCA in the feature space consists of diagonalizing the square symmetric matrix $\mathbf{K}$ composed of all possible inner products of the set of mapped samples. This type of matrix is called a *Gram Matrix*:

$$\mathbf{K} = \begin{pmatrix} \langle\Phi(\mathbf{x}^1),\Phi(\mathbf{x}^1)\rangle_{\mathcal{H}} & \langle\Phi(\mathbf{x}^1),\Phi(\mathbf{x}^2)\rangle_{\mathcal{H}} & \dots & \langle\Phi(\mathbf{x}^1),\Phi(\mathbf{x}^\ell)\rangle_{\mathcal{H}} \\ \langle\Phi(\mathbf{x}^2),\Phi(\mathbf{x}^1)\rangle_{\mathcal{H}} & \langle\Phi(\mathbf{x}^2),\Phi(\mathbf{x}^2)\rangle_{\mathcal{H}} & \dots & \langle\Phi(\mathbf{x}^2),\Phi(\mathbf{x}^\ell)\rangle_{\mathcal{H}} \\ \vdots & \vdots & \ddots & \vdots \\ \langle\Phi(\mathbf{x}^\ell),\Phi(\mathbf{x}^1)\rangle_{\mathcal{H}} & \langle\Phi(\mathbf{x}^\ell),\Phi(\mathbf{x}^2)\rangle_{\mathcal{H}} & \dots & \langle\Phi(\mathbf{x}^\ell),\Phi(\mathbf{x}^\ell)\rangle_{\mathcal{H}} \end{pmatrix}. \quad (1.24)$$

Thanks to the kernel property, the inner product in the feature space can be derived in the input space using *kernel function*. Details can be found in appendix A. $\mathbf{K}$ is therefore called the *Kernel Matrix*:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}^1,\mathbf{x}^1) & k(\mathbf{x}^1,\mathbf{x}^2) & \dots & k(\mathbf{x}^1,\mathbf{x}^\ell) \\ k(\mathbf{x}^2,\mathbf{x}^1) & k(\mathbf{x}^2,\mathbf{x}^2) & \dots & k(\mathbf{x}^2,\mathbf{x}^\ell) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}^\ell,\mathbf{x}^1) & k(\mathbf{x}^\ell,\mathbf{x}^2) & \dots & k(\mathbf{x}^\ell,\mathbf{x}^\ell) \end{pmatrix}. \quad (1.25)$$

Using $\mathbf{K}$, the non-linearity is included in the kernel function and mapping is no longer necessary. All computations are done in the input space, while PCA is performed *implicitly* in the feature space. Projection on the first KPCs can also be done in the input space, using the kernel function in (1.23).

However, in the previous section we have assumed that the data are centered in $\mathcal{H}$. This can be done in the feature space:

$$\Phi_c(\mathbf{x}^i) = \Phi(\mathbf{x}^i) - \frac{1}{\ell}\sum_{k=1}^{\ell}\Phi(\mathbf{x}^k) \quad (1.26)$$

but to be applied directly in the input space, it needs to be done in terms of dot product [113]:

$$\begin{aligned} k_c(\mathbf{x}^i,\mathbf{x}^j) &= \langle\Phi_c(\mathbf{x}^i),\Phi_c(\mathbf{x}^j)\rangle_{\mathcal{H}} \\ &= \langle\Phi(\mathbf{x}^i) - \frac{1}{\ell}\sum_{k=1}^{\ell}\Phi(\mathbf{x}^k), \Phi(\mathbf{x}^j) - \frac{1}{\ell}\sum_{k=1}^{\ell}\Phi(\mathbf{x}^k)\rangle_{\mathcal{H}} \\ &= \langle\Phi(\mathbf{x}^i),\Phi(\mathbf{x}^j)\rangle_{\mathcal{H}} - \frac{1}{\ell}\sum_{k=1}^{\ell}\langle\Phi(\mathbf{x}^i),\Phi(\mathbf{x}^k)\rangle_{\mathcal{H}} - \frac{1}{\ell}\sum_{k=1}^{\ell}\langle\Phi(\mathbf{x}^k),\Phi(\mathbf{x}^j)\rangle_{\mathcal{H}} + \frac{1}{\ell^2}\sum_{\substack{k=1\\m=1}}^{\ell}\langle\Phi(\mathbf{x}^k),\Phi(\mathbf{x}^m)\rangle_{\mathcal{H}} \\ &= k(\mathbf{x}^i,\mathbf{x}^j) - \frac{1}{\ell}\sum_{k=1}^{\ell}k(\mathbf{x}^i,\mathbf{x}^k) - \frac{1}{\ell}\sum_{k=1}^{\ell}k(\mathbf{x}^k,\mathbf{x}^j) + \frac{1}{\ell^2}\sum_{\substack{k=1\\m=1}}^{\ell}k(\mathbf{x}^k,\mathbf{x}^m) \quad (1.27) \end{aligned}$$

The equivalent formula in *matrix form* is [112]:

$$\mathbf{K}_c = \mathbf{K} - \mathbf{1}_\ell\mathbf{K} - \mathbf{K}\mathbf{1}_\ell + \mathbf{1}_\ell\mathbf{K}\mathbf{1}_\ell \quad (1.28)$$

where $\mathbf{1}_\ell$ is a square matrix such as $(\mathbf{1}_\ell)_{ij} = \frac{1}{\ell}$. Finally, KPCA is performed in the input space following Algorithm 2.

Dimensionality reduction is performed the same way as in PCA. When the kernel matrix is diagonalized, the first k kernel principal components, corresponding to a set level of cumulative variance, are retained. In general, KPCA provides more kernel principal components than in PCA; furthermore, one may even obtain more kernel principal components than the initial dimension of the data (since the kernel matrix is generally of a higher dimension than the covariance matrix). But this property is of no interest when considering KPCA for feature extraction.

Algorithm 2 *Kernel Principal Component Analysis*

```
1: Compute $\mathbf{K}$ using (1.25)
2: Center $\mathbf{K}$ using (1.27)
3: Solve: $\lambda\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}$ subject to $\|\boldsymbol{\alpha}\|_2 = \frac{1}{\lambda}$
4: Project on the first $k$ kernel principal components: $\Phi_{kpc}(\mathbf{x}) = \left[\Phi^1_{kpc}(\mathbf{x}) \quad \dots \quad \Phi^k_{kpc}(\mathbf{x})\right]^t$
       $\Phi^k_{kpc}(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^k k(\mathbf{x}^i, \mathbf{x})$
```

1.4.3 Computing the KPCA

To compute the KPCA, it is first necessary to choose the kernel function to build the kernel matrix. For experimental purposes, the RBF kernel (A.13) was used:

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right). \tag{1.29}$$

As explained in Appendix A, the width σ of the exponential has to be tuned correctly to fit the data properly. Two strategies can be adopted:

1. Use a fixed σ and tune it according to some general characteristics, *e.g.*, the number of bands.
2. Perform a certain amount of processing to find an appropriate value, *e.g.*, fit σ equal to the radius of the minimal enclosing hypersphere (see Appendix A).

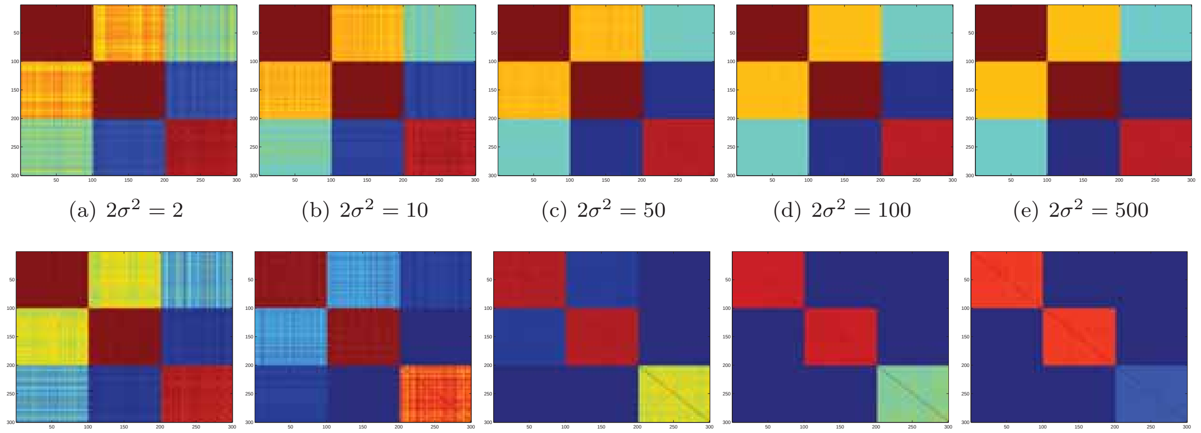
Both these strategies have been investigated in this thesis. The first one was motivated by the good performance of such a setting for the purpose of classification. The second strategy was based on the fact that the kernel matrix emphasizes the similarity between samples (the kernel matrix is sometimes called the *similarity matrix* [118]). Still from [118], a good kernel matrix is a matrix where clusters appear; the worst case is a diagonal matrix. To be sure that each sample is able to influence its neighbors in the spectral domain, the exponential kernel should be large enough for samples belonging to the same cluster to form blocks in the kernel matrix. In the introduction we stated that normally-distributed $\mathcal{N}(0, 1)$ data have a tendency to concentrate in the tails; it has been proved moreover that the expected norm of such variables is equal to the square root of the dimension [43]. This means that the dimension of the space grows faster than the mean distance between samples. Thus the strategy of setting σ equal to the dimension does not seem to be appropriate. It ought to be better to estimate the support of the samples. In this thesis, we have investigated the use of the minimal enclosing hypersphere. The main motivation is its relative computational simplicity and the possibility of using it in conjunction with kernel methods. An algorithm to compute the radius $\mathcal{R}_S$ and center $\mathcal{C}_S$ of the hypersphere is given in Appendix A. Note that we are only interested in the radius, as we tuned $2\sigma^2$ to be equal to the radius.

To verify this assumption empirically, we generated 100 samples from three different multi-dimensional Gaussian distributions, for several dimensionalities of the data. We have computed the kernel matrix using two different values of sigma, one equal to the number of variables and one equal to the radius of the minimal hypersphere. The results are reported in Figure 1.4. It is clear that when the dimensionality increased, tuning σ as a function of the radius yields the best results; in Figure 1.4(j) three clusters can be readily identified, each of them corresponding to one Gaussian distribution.

Figures 1.5 and 1.6 present the first Kernel PC (KPC) obtained with σ tuned according to the radius and Table 1.2 shows the variance and cumulative variance for the two data sets. The kernel matrix was constructed using 5000 randomly-selected samples. For the *Pavia Center* data set, the radius of the minimal enclosing hypersphere was 9.47 and 9.03 for the *University Area* data set. From the table, it can be seen that more kernel principal components are needed to achieve the same amount of variance than in the conventional PCA. For the *University* data set, the first 11 KPCs are needed to achieve 95% of the

Table 1.2: *KPCA: Eigenvalues and cumulative variance in percent for the two hyperspectral data sets.*

	Pavia Center		*University Area*	
Component	%	Cum. %	%	Cum. %
1	45.97	45.97	32.52	32.52
2	21.44	61.41	26.73	59.25
3	14.95	82.36	16.85	76.10
4	4.83	87.19	4.04	80.14



(a) $2\sigma^2 = 2$ (b) $2\sigma^2 = 10$ (c) $2\sigma^2 = 50$ (d) $2\sigma^2 = 100$ (e) $2\sigma^2 = 500$

(f) $2\sigma^2 = 1.29$ (g) $2\sigma^2 = 2.58$ (h) $2\sigma^2 = 5.36$ (i) $2\sigma^2 = 7.52$ (j) $2\sigma^2 = 16.48$

Figure 1.4: *Comparison of the kernel matrix for two σ tuning strategies. The dimensions of the samples were* 2, 10, 50, 100, *and* 500 *respectively. The first* 100 *were generated from the same distribution, then the next* 100 *samples, and so on.*

variance and only 8 for the *Center* data set. This may be an indication that more information is extracted with the KPCA. Experiments will show whether this information is useful or not for classification.

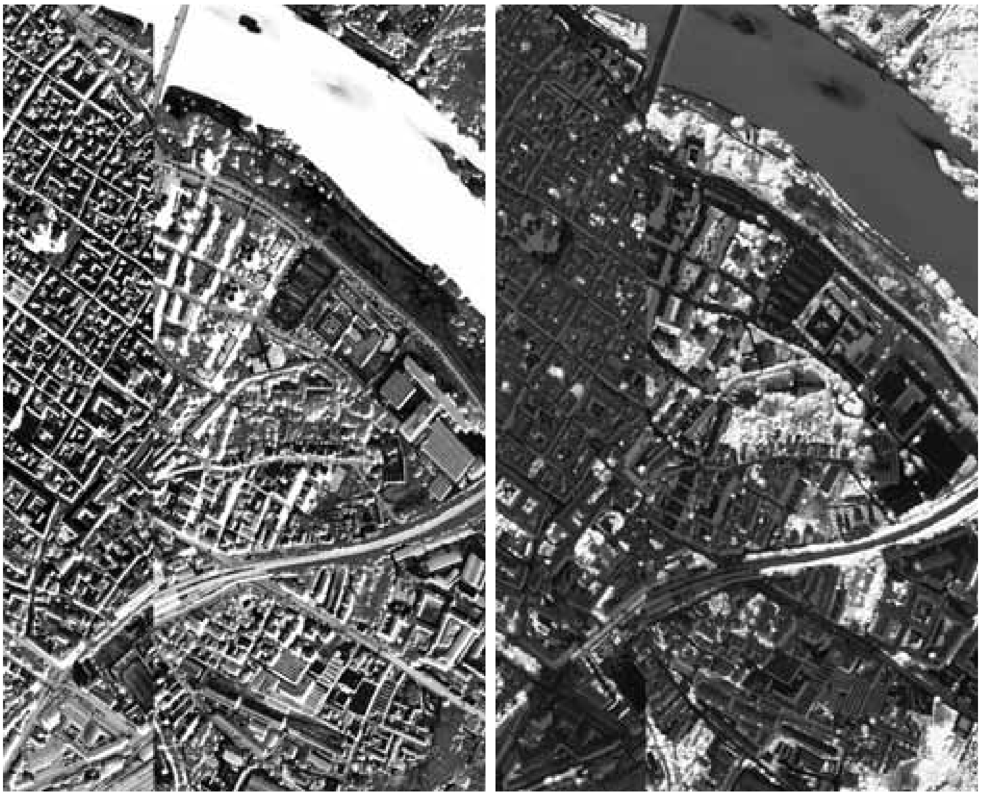
1.5 Experiments

In this section, experiments in feature reduction for the classification of hyperspectral data are presented. In the first experiment, KPCA is compared with PCA and a supervised feature-reduction algorithm, namely Decision Boundary Feature Extraction. In a second experiment, the influence of the σ parameter in kernel matrix construction is investigated in accordance with the scheme proposed in the previous section.

1.5.1 KPCA vs Traditional Methods

1.5.1.1 Objectives

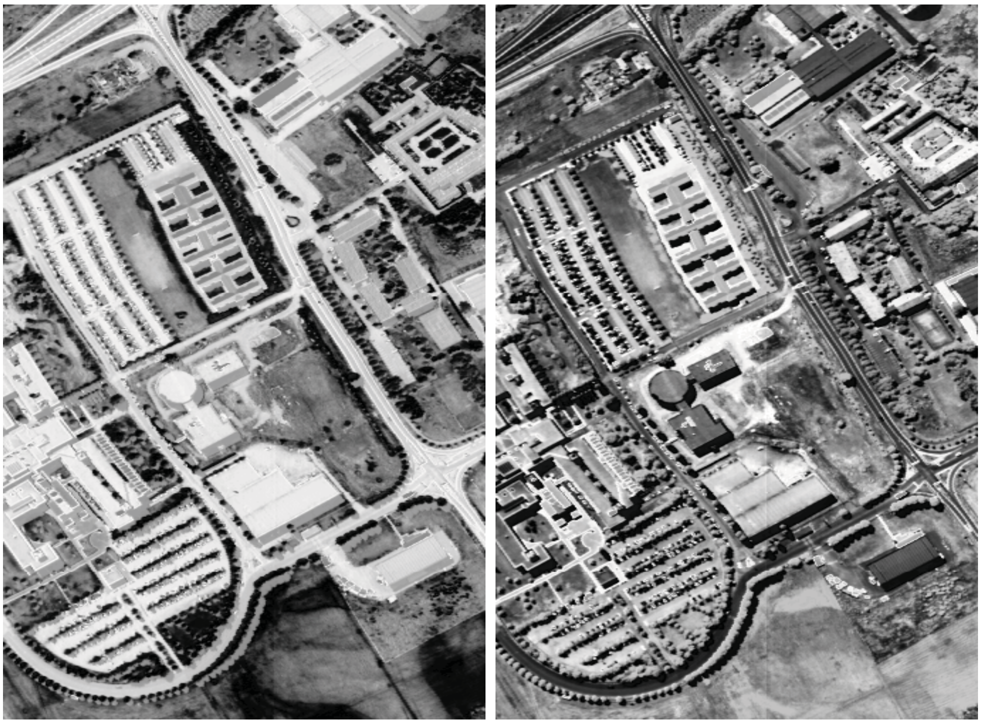
In this experiment [48], we have compared KPCA with PCA and a linear supervised feature extraction method, DBFE, as pre-processing for the classification of hyperspectral data using a neural network. We first recall briefly what DBFE is, and then present the experimental results.



(a) First kernel principal component

(b) Second kernel principal component

Figure 1.5: *First two kernel principal components,* Pavia Center *data set.*



(a) First kernel principal component

(b) Second kernel principal component

Figure 1.6: *First two kernel principal components,* University Area *data set.*

The principle of DBFE is to extract discriminantly informative features from the decision boundary between two classes [83, 80]. Under Bayes' theorem, the decision boundary is

$$\mathbf{x} : \{p(w_1)p(\mathbf{x}|w_1) = p(w_2)p(\mathbf{x}|w_2)\}\,. \tag{1.30}$$

where $p(w_i)$ is the appearance probability for the class i and $p(\mathbf{x}|w_i)$ is the conditional probability of $\mathbf{x}$ to belong to w_i. From equation (1.30), $\mathbf{x} \in w_1$ if $p(w_1)p(\mathbf{x}|w_1) > p(w_2)p(\mathbf{x}|w_2)$.

In [83, 80], the authors have defined *discriminant informative features* as features that change the position of $\mathbf{x}$ across the decision boundary. It can be proved that every discriminant informative feature is normal to the decision boundary at at least one point. By denoting as $\mathbf{n}(\mathbf{x})$ the unit normal vector to the decision boundary at point $\mathbf{x}$, they defined the Decision Boundary Feature Matrix (DBFM) [83]:

$$\Sigma_{DBFE} = \frac{1}{Q}\int_S \mathbf{n}(\mathbf{x})\mathbf{n}(\mathbf{x})^t p(\mathbf{x})d\mathbf{x}. \tag{1.31}$$

The rank of the DBFM of classification problem is equal to the intrinsic discriminant dimension. Hence the eigenvectors of the DBFM corresponding to non-zero eigenvalues are the necessary feature vectors to achieve the same classification accuracy as in the original space [80]. The authors of [83, 80] have generalized the DBFE to a multiclass case and propose an algorithm for a neural network classifier, which was used in their experiments.

Such an algorithm is specially designed for the classification problem. But the efficiency of the algorithm is very closely related to the training set. In the following experiments, we deliberately used a limited number of training samples in order to investigate this phenomenon [48].

Our test images are from the ROSIS 03 sensor, the number of bands is 103 with spectral coverage from 0.43 through $0.86\mu m$. The image area is 610 by 340 pixels. PCA and KPCA were applied on these data. The kernel function used was the RBF kernel, where the parameter γ was set to 100. The kernel matrix was computed with 5000 pixels, selected at random. For PCA, 95% of the total eigenvalue sum

Table 1.3: *Overall classification accuracy in percentage for the experiments in 1.5.1.2.*

	PCA			KPCA								DBFE		All
No. of feat.	1	2	3	1	2	3	4	5	6	7	28	8	26	103
Testing	37.7	69.9	73.1	37.3	66.0	72.4	73.1	74.4	74.9	74.5	55.1	64.1	43.4	27.3
Training	39.7	71.2	74.1	39.4	66.6	75.3	76.8	77.9	79.1	78.1	59.9	68.2	43.6	30.4

is achieved with the first three components, while with KPCA 28 components are needed. However, the total number of components with PCA is equal to the number of channels, while with KPCA it is equal to the size of the kernel matrix, *i.e.*, the number of samples used, which is significantly higher.

1.5.1.2 Experiments

In this experiment, the previously-extracted features were used as input to a back-propagation neural network classifier with one hidden layer. The number of neurons in the hidden layer is twice the number of outputs, *i.e.*, the number of classes. The training set consisted of 3921 pixels with labels. 9 classes were used in the classification: asphalt, meadow, gravel, tree, sheet metal, bare soil, bitumen, brick, and shadow. A quarter of the labeled samples were used for training, the others samples were used for testing. The results were compared with classification of the full spectrum and DBFE-transformed features.

The results are listed in Table 1.3. The classification of the full spectrum demonstrate the Hughes phenomenon: the number of training samples (980) was too small compared with the dimensionality of the data (103), leading to poor classification accuracy: 27.3%. Classification using 1 principal component yielded slightly better results for PCA and KPCA. For PCA, with 3 principal components (corresponding to 95% of the total variance) the overall classification accuracy on the test set is 73.1%; adding more bands does not significantly improve classification accuracy, and adding too many bands worsened classification, as predicted by the Hughes phenomenon. For KPCA, classification accuracy reached 74.5% with 7 features, corresponding to 81.02% of the variance. For 6 to 10 bands, the results remained nearly equal ($\approx$ 74.5%), and then decreased if the number of bands was increased further. For DBFE, with 8 features, corresponding to 62.2% of the total variance, classification is slightly worse than with two principal components, while with 26 features, corresponding to 95% of the variance criterion, classification accuracy is worse still. Nevertheless, with more training samples, DBFE ought to yield better results [8, 80]. The classification map obtained using 8 KPCs is shown in figure 1.7(a).

Using the first principal components from KPCA improved the classification accuracy slightly, but unlike PCA, 95% does not seem to be the optimum value of variance yielding best classification accuracy. In these experiments, 80% of the variance yielded the best results.

In this experiment, the parameter of the kernel was set at approximately the inverse of the number of bands. In the following experiment, we investigate automatic tuning of this value.

1.5.2 Selection of the kernel parameter

In the previous section, the kernel parameter was set empirically to the number of bands. In this section we propose estimating the distribution of the data in the input space by considering the radius of the minimal enclosing hypersphere, and tuning the kernel parameter accordingly. As seen in Section 1.4 for high-dimensional space this strategy performs better than using the number of dimensions. But that conclusion was subject to the Gaussianity of the data.

Now we are going to test this assumption on real hyperspectral data, where Gaussianity is not necessarily satisfied. For each data set, KPCA was performed with two σ values. Then the KPCs were

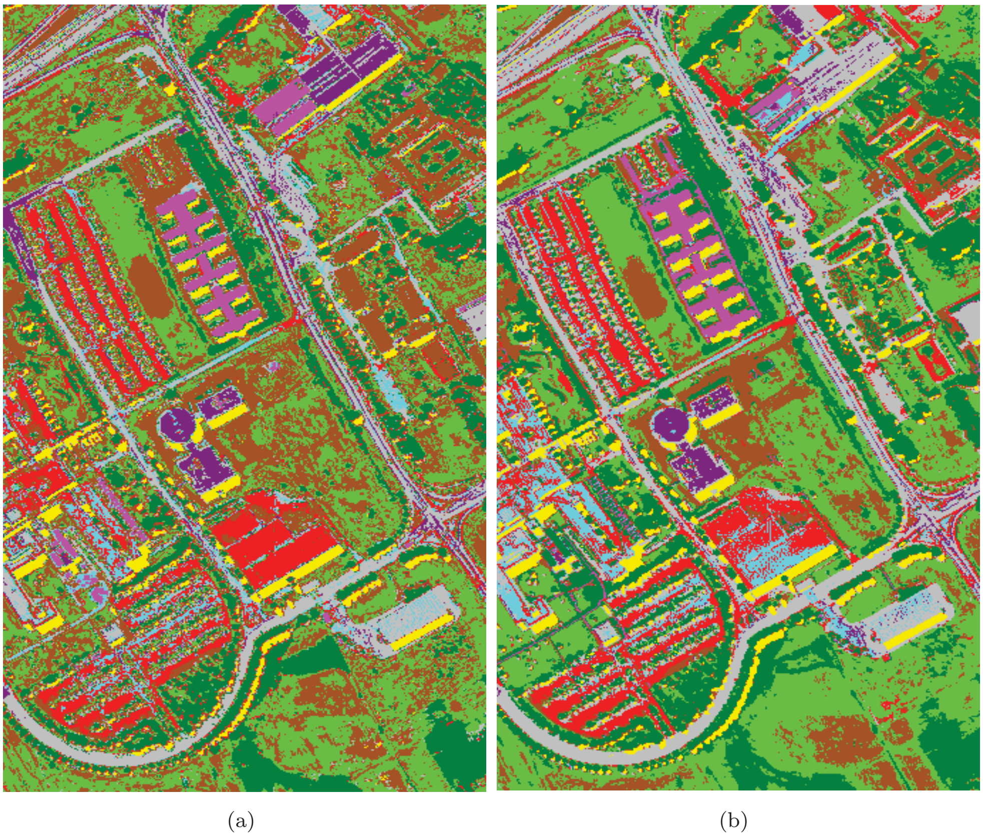


Figure 1.7: *Classification map obtained (a) using a neural network and the first 8 KPCs and (b) from the first 11 KPCs with a linear SVM. Class descriptions: asphalt, meadow, gravel, tree, sheet metal, bare soil, bitumen, brick, shadow.*

classified using a linear SVM. The classification results are still compared to those obtained with PCA. To speed up the processing, we only consider the first 103 (or 102) KPCs, *i.e.*, the cumulative variance is computed on 103 (or 102) features. For each KPCA we only use the 95% of the cumulative variance. Details of the linear SVM are given in Chapter 3. Classification results[2] are given in Table 1.4: κ is the kappa coefficient of agreement, which is the percentage of agreement corrected by the amount of agreement that might be expected due to chance alone, AA is the average of class classification accuracies and OA is the percentage of pixels correctly classified.

The radius found for the *University Area* data set was 9.03, and 9.48 for the *Pavia Center* data set. In terms of the results, it seems that classification accuracies are similar with both σ values. Only the number of bands corresponding to 95% of the total variance changes. The PC-based classification yields the worst results. When using a linear classifier, it is clear that the non-linear dimensionality reduction performed by the KPCA helps in classification. The classification map obtained with 11 KPCs for the *University Area* is shown in Figure 1.7(b).

1.6 Concluding remarks

In this chapter, the problem of the high dimensionality of the hyperspectral data has been addressed. A non-linear version of PCA is used to reduce the dimensionality for the purpose of classification. When a linear classifier is used, Kernel PCA provides more relevant features, as shown in the second experiment.

[2] Training and testing sets for this experiments were not the same as in the previous experiments.

Table 1.4: *Classification results for the* University Area *and* Pavia Center *data sets.*

	University Area				Pavia Center			
	No. of feat.	κ	AA	OA	No. of feat.	κ	AA	OA
$2\sigma^2 = \mathcal{R}_s$	11	62.04	79.36	69.91	8	94.63	90.71	96.20
$2\sigma^2 = 103$ (or 102)	4	62.73	75.01	71.43	4	94.02	90.32	95.76
PCA	3	56.42	73.51	65.52	2	82.08	78.83	86.93

By relevant, we mean that in this feature space, the classes considered are more linearly separable. However, when a non-linear classifier is used, the differences with conventional PCA are slight, and the resulting improvement in classification accuracy is due more to the dimensionality reduction than to a greater number of linearly-separable features.

The problem of tuning the kernel parameter has been also considered by looking at the minimal enclosing hypersphere. Even though this approach appears worthwhile with synthetic data sets, on real hyperspectral data no significant differences have been seen for the purpose of classification. Our conclusions are that, just as for classification using SVM, there is no one optimal value for the kernel parameter, but a wide range of values over which the algorithm yields simialr results.

In this section, feature reduction has been addressed for the purpose of classification only. But a large number of image processing algorithms can be only applied to a single-band image, and hence feature reduction can be helpful in extracting one band from multi- or hyperspectral data.

Chapter 2

Spatial Feature Extraction

Abstract

Spatial feature extraction from remote-sensing images is discussed in the following chapter. Spatial information in the context of remote sensing is defined. Then, Mathematical Morphology is presented as an important tool for the analysis of the spatial organization of an image. Basic and advanced morphological filters are discussed, such as opening and closing by reconstruction. Based on these filters, the Morphological Profile and its derivative are introduced as a methodology for extracting information about the shapes, sizes, and structures present in the image. Experimental results on a panchromatic image confirm the usefulness of the approach. However, due to the inherent properties of the Morphological Profile, some structures in the image are not processed. In this thesis, self-complementary filters are proposed to analyze all the structures in the image. Comparisons are made with the Morphological Profile. Finally, the problem of multi-valued images is addressed by mean of spectral feature reduction.

Contents

THE previous chapter presented algorithms to extract useful informative features from the spectral domain. This chapter discusses the extraction of spatial information, mainly using Mathematical Morphology. The primary processing is based on advanced morphological concepts which are detailed in the first section. We give the definition of Morphological Profile and then propose an alternative approach.

2.1 Introduction

With recent remote sensors, the acquired images now have very fine spatial resolution. Thus useful information could be extracted in the image domain. We usually talk about *contextual information* or *inter-pixel dependency*. This information can be modeled as the dependency between a pixel and its neighbors. According to this dependency, it is possible to define *structures*, which are connected sets of pixels with high dependency. For a remote sensing application, the structures are objects of high interest.

For image segmentation, two well-known approaches exist: find the boundary between structures (*i.e.* detect transitions), or find sets of pixels that share the same characteristics (*i.e.* detect similitude). When attempting to classify, the second approach is better suited, since we are not interested in the boundaries between regions, but in the regions themselves. Thus the contextual information needs to be extracted *into* structures and *between* structures. Figure 2.1 present some typical structures present in remote sensing data over a lightly built-up area.

Mathematical Morphology provides a well-established theory for analyzing the spatial relationship between sets of pixels [115, 116]. Readers will find a review of Mathematical Morphology applications in remote sensing in [122]. Advanced Mathematical Morphology operators can extract a great deal of structural information, such as:

- direction
- contrast
- size
- texture
- shape
- ...

In what follows, we recall the concept of the morphological profile and the derivative of it that we use to extract information about the size, shape, and contrast of the structures in the image [101, 8]. Then we present its extension to multi-valued images. The limitations of the morphological profile are also discussed. We then propose another approach, based on a self-complementary filter. The aim of this is to simplify the image to remove structures that are irrelevant for classification. In our approach, irrelevant structures are defined as structures that do not satisfy an area criterion, thus leading to the use of an area *self-complementary filter* [121].

2.2 Theoretical Notions

Mathematical Morphology is a theory for non-linear image processing. It aims to analyze spatial relationship between pixels. In this section, basics notions of Mathematical Morphology are presented. Readers interesting in Mathematical Morphology can find additional material in [120, 115, 116].

In image analysis, data are represented in discrete space $\mathbb{Z}^n$, and an image f is a mapping of a subset D_f of $\mathbb{Z}^n$.

$$f : \mathcal{D}_f \subset \mathbb{Z}^n \rightarrow \{0, \dots, f_{max}\} \tag{2.1}$$

where f_{max} is the maximum value of the image. In the following we only consider flat images, *i.e* $\mathcal{D}_f \subset \mathbb{Z}^2$. With Mathematical Morphology, objects of interest are viewed as a subsets of the image; then several sets of known size and shape (such as disk, square or line) can be used to characterize their morphology. These sets are called *Structuring Element* (SE). SE has always an origin, which is generally the symmetric

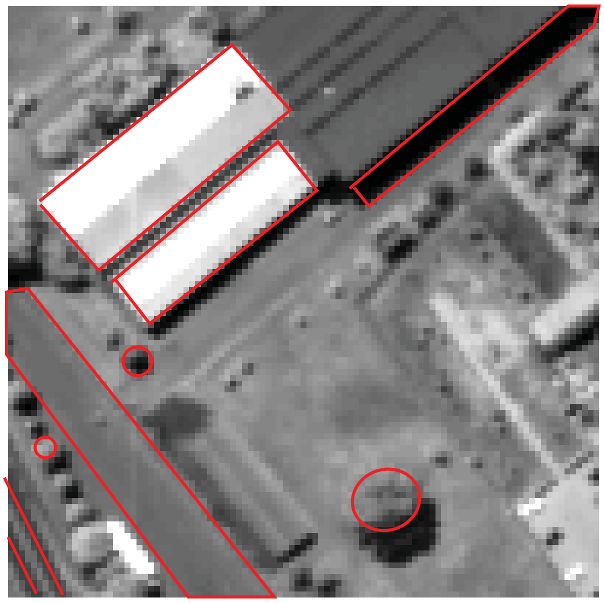
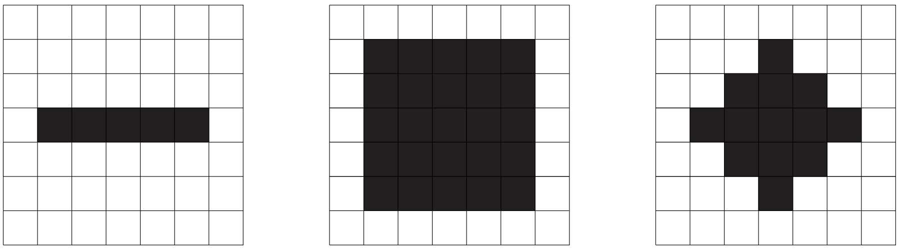


Figure 2.1: *Example of structural information: large, light, rectangular structure – small, circular, textured structure – long, thin, dark structure ...*

center. It allows the positioning of the SE at a given pixel $\mathbf{x}$ of f, *i.e*, its origin coincides with $\mathbf{x}$. Example of SE are given in figure 2.2.



(a) Line

(b) Square

(c) Discrete Disk

Figure 2.2: *Example of SE.*

For binary images ($f_{max} = 1$), Mathematical Morphology are mainly based on set operators such as union, intersection, complementation and translation: SE is positioning on each pixel $\mathbf{x}$ and a set operators is applied between the set which $\mathbf{x}$ belongs to and SE. For grey tone images, intersection $\cap$ of two sets becomes the *infimum* $\wedge$ and the union $\cup$ becomes *supremum* $\vee$. For two images f and g and a given pixel $\mathbf{x}$: $(f \wedge g)(\mathbf{x}) = \min[f(\mathbf{x}), g(\mathbf{x})]$ and $(f \vee g)(\mathbf{x}) = \max[f(\mathbf{x}), g(\mathbf{x})]$.

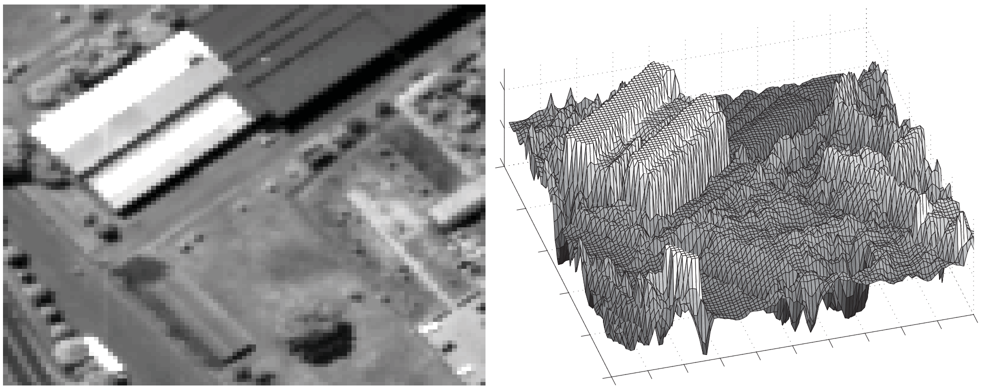
Definition 2.1 (Graph and Subgraph) *The* graph $\mathcal{G}$ *of image* f *is the set of points* $(\mathbf{x}, t)$ *such that* $\mathbf{x} \in \mathcal{D}_f$ *and* $t = f(\mathbf{x})$*:*

$$\mathcal{G}(f) = \left\{(\mathbf{x}, t) \in \mathbb{Z}^2 \times \mathbb{Z} | t = f(\mathbf{x})\right\}. \tag{2.2}$$

The subgraph $\mathcal{SG}$ *is the set of points* $(\mathbf{x}, t)$ *lying below the graph:*

$$\mathcal{SG}(f) = \left\{(\mathbf{x}, t) \in \mathbb{Z}^2 \times \mathbb{Z} | t \leq f(\mathbf{x})\right\}. \tag{2.3}$$

Figure 2.3 present an image and its graph. The two fundamental morphological operators are the *erosion* and the *dilation*.



(a) Original (b) Graph

Figure 2.3: *Example of graph.*

Definition 2.2 (Erosion) *The erosion $\epsilon_B(f)$ of an image f by a structuring element B is defined as:*

$$\epsilon_B(f) = \bigwedge_{b \in B} f_{-b}. \tag{2.4}$$

Where f_b is the translation by vector b of f, *i.e.*, $f_b(\mathbf{x}) = f(\mathbf{x} - b)$. The eroded value at a given pixel $\mathbf{x}$ is the minimum value of the image in the window defined by the SE when its origin is at $\mathbf{x}$. It shows where the SE fits the objects in the input image.

Definition 2.3 (Dilation) *The dilation $\delta_B(f)$ of an image f by a structuring element B is defined as:*

$$\delta_B(f) = \bigvee_{b \in B} f_{-b}. \tag{2.5}$$

The dilated value at a given pixel $\mathbf{x}$ is the maximum value of the image in the window defined by the SE when its origin is at $\mathbf{x}$. It shows where the SE hits the objects in the input image. The erosion and the dilation are *dual* transformations with respect to the complementation:

$$\epsilon_B(f) = [\delta_B([f]^c)]^c \tag{2.6}$$

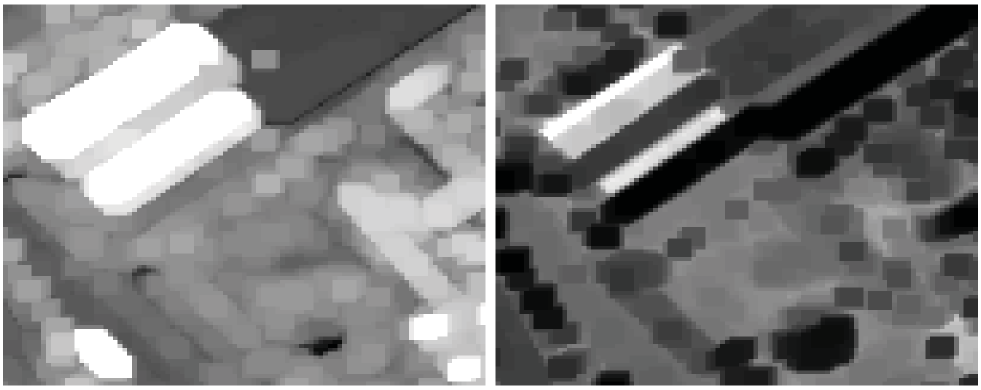
where $[\ ]^c$ is the complementation operator: $[f]^c(\mathbf{x}) = f_{max} - f(\mathbf{x})$. This property shows the dual effect of erosion and dilation. When erosion expands dark objects, dilation shrinks them (and vice-versa for clear objects). Moreover, clear (respectively dark) structures that cannot contain the SE are removed by erosion (dilation). Hence, both erosion and dilation are *non-invertible* transformation. Figure 2.4 shows examples of erosion and dilation. These two operators are the basic tools of Mathematical Morphology. The next operators, *opening* and *closing*, are a combination of erosion and dilation.

Definition 2.4 (Opening) *The opening $\gamma_B(f)$ of an image f by a SE B is defined as the erosion of f by B followed by the dilation with the SE B*[1].

$$\gamma_B(f) = \delta_B[\epsilon_B(f)]. \tag{2.7}$$

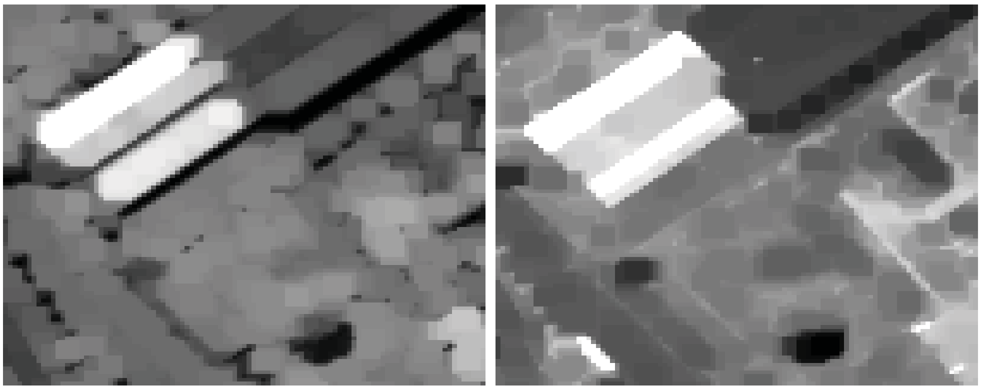
The idea to dilate the eroded image is to recover most structures of the original image, *i.e.*, structures that were not removed by the erosion.

[1] The true definition is with the transposed SE $\check{B}$. Transposition of B corresponds to its symmetric set with respect to its origin. For simplicity, we only consider SE whose origin is also the symmetric center, so $\check{B} = B$



(a) Dilation (b) Erosion

Figure 2.4: *Dilation and erosion of image 2.3(a) with a disk of radius 3 pixels.*



(a) Opening (b) Closing

Figure 2.5: *Opening and closing of image 2.3(a) with a disk of radius 3 pixels.*

Definition 2.5 (Closing) *The closing $\phi_B(f)$ of an image f by a SE B is defined as the dilation of f by B followed by the erosion with the SE B.*

$$\phi_B(f) = \epsilon_B[\delta_B(f)]. \tag{2.8}$$

Figure 2.5 shows result of closing and opening of an image by a disk with a radius of 3 pixels. It can be seen that structures of size less than the SE are totally removed. Even opening and closing are powerful operators, their major drawback is that their are not connected filters. It can be seen in figure 2.5 that many structures have merged, for example the two bright buildings have merged into one, see 2.5(b). To avoid that problem, *geodesic* morphology and *reconstruction* can be use. Reconstruction filters are connected filters and they have been proved to do not introduce discontinuities. Reconstruction filters [38] are based on geodesic morphology.

2.3 Geodesics transforms

Morphological geodesics transforms are morphological operators that use non-Euclidean geodesic distance.

Definition 2.6 (Geodesic dilation) *The geodesic dilation $\delta_g^{(1)}(f)$ of size 1 consists in dilating a marker f with respect to a mask g:*

$$\delta_g^{(1)}(f) = \delta^{(1)}(f) \wedge g. \tag{2.9}$$

The geodesic dilation of size n is obtained by performing n successive geodesic dilations of size 1.

Definition 2.7 (Geodesic erosion) *The geodesic erosion $\epsilon_g^{(1)}(f)$ is the dual transformation of the geodesic dilation.*

$$\epsilon_g^{(1)}(f) = \epsilon^{(1)}(f) \vee g. \tag{2.10}$$

Definition 2.8 (Reconstruction) *The reconstruction by dilation (erosion) of a marker f with respect to a mask g consists of repeating a geodesic dilation (erosion) of size one until stability,* i.e, *$\delta_g^{(n+1)}(f) = \delta_g^{(n)}(f)$ ($\epsilon_g^{(n+1)}(f) = \epsilon_g^{(n)}(f)$).*

$$Rec_g(f) = \delta_g^{(n)}(f), \tag{2.11}$$
$$Rec_g^*(f) = \epsilon_g^{(n)}(f). \tag{2.12}$$

With definition 2.8, it is possible to define connected transformation that satisfy the following assertion: *if the structure of the image cannot contain the SE then it is totally removed, else it is totally preserved.* These operators are called *opening/closing by reconstruction.*

Definition 2.9 (Opening-Closing by reconstruction) *The opening by reconstruction of an image f is defined as the reconstruction by dilation of f from the erosion of size n of f. Closing by reconstruction is defined by duality.*

$$\gamma_R^{(n)} = Rec_f(\epsilon^{(n)}(f)), \tag{2.13}$$
$$\phi_R^{(n)} = Rec_f^*(\delta^{(n)}(f)). \tag{2.14}$$

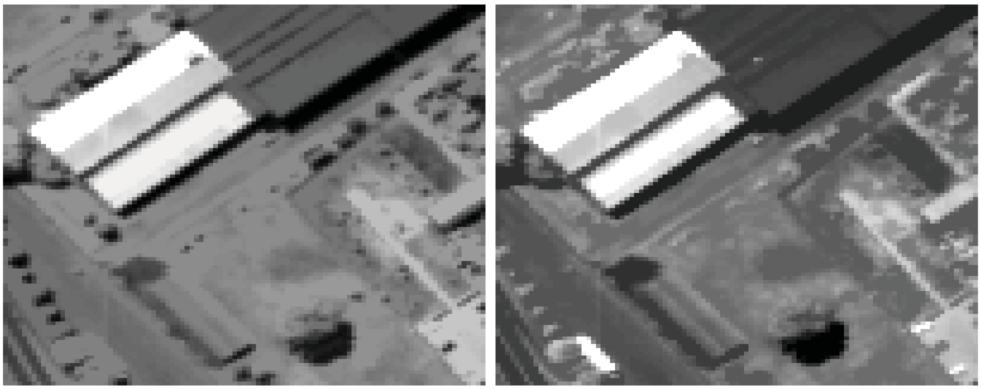
Figure 2.6 shows results of opening and closing by reconstruction. It can be clearly seen that these transformations introduce less noise than classical opening-closing. Shapes are preserved and structures still present after transformation are of a size greater than or equal to the SE. The use of opening and closing by reconstruction allowed to characterize morphological characteristics of structures present in the image. In addition, to determine size or shape of *all* objects present in an image, it is necessary to use a range of different SE size. This concept is called *Granulometry* [120].

2.4 Morphological tools

Based on the previously presented filters, many morphological tools can be developed. Formally, a morphological filter is an increasing and idempotent operation [120, 38]. The increasingness property ensures that the ordering relation between is preserved and idempotence property roughly means that the operation yields the same result whether it is done only once or several times. The *Granulometry* is based in the iteration of opening or closing, and the *Alternating Sequential Filter* (ASF) are based on the alternating use of opening and closing filters.

Definition 2.10 (Granulometry) *A Granulometry Φ_λ is defined by a transformation having a size parameter λ and satisfying the three following axioms:*

- Anti-extensivity*: the transformed image is less than or equal to the original image.*
- Increasingness*: the ordering relation between image is preserved.*



(a) Opening by reconstruction

(b) Closing by reconstruction

Figure 2.6: *Opening and closing by reconstruction with a disk of radius 3 pixels.*

- Absorption*: the composition of two transformations* Φ *of different size* λ *and* ν *will give always the result of transformation with the biggest size:*

$$\Phi_\lambda \Phi_\nu = \Phi_\nu \Phi_\lambda = \Phi_{\max(\lambda,\nu)}. \tag{2.15}$$

Granulometries are typically used for the analysis of the size distribution of structures in images. Classical granulometry by opening is build by successive opening operation of an increasing size. By doing so, image is progressively simplified. Using connected operators, like opening by reconstruction, no shape noise is introduced.

Anti-Granulometry is defined with the same axioms as granulometry and by replacing *anti-extensivity* axiom by *extensivity.*

Definition 2.11 (Alternating Sequential Filter (ASF)) *An ASF is the sequential combination of opening and closing of an increasing size.*

$$ASF_i = \gamma_1 \phi_1 \dots \gamma_i \phi_i. \tag{2.16}$$

As granulometry, ASF can be seen as a tool to analyze the size distribution of structure. But contrary to the granulometry, ASF analyze both the bright and dark structures. Obviously, an ASF beginning with an opening would not give the same result than an ASF beginning with a closing:

$$\gamma_1 \phi_1 \dots \gamma_i \phi_i \neq \phi_1 \gamma_1 \dots \phi_i \gamma_i. \tag{2.17}$$

2.5 Ordering relation

Precisely, Mathematical Morphology is defined on a *lattice* which is a *partially ordered set* in which every pair of elements has an unique supremum and an infimum. A partially ordered set $\mathcal{X}$ is a set satisfying, $\mathbf{x},\ \mathbf{y},\ \mathbf{z} \in \mathcal{X}$:

1. $\mathbf{x} \leq \mathbf{x}$
2. $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{y} \leq \mathbf{x}$ iif $\mathbf{y} = \mathbf{x}$
3. if $\mathbf{x} \leq \mathbf{y}$ and $\mathbf{y} \leq \mathbf{z}$ then $\mathbf{x} \leq \mathbf{z}$.

Totally ordered set is a partially ordered set where $\mathbf{x} \leq \mathbf{y}$ **or** $\mathbf{y} \leq \mathbf{x}$. For images with $\mathcal{D}_f \subset \mathbb{Z}^2$, scalar ordering relation is used, the supremum and infimum are respectively, f_{max} and 0. For non-flat images, there is no way to define unambiguous supremum and infimum values:

$$\begin{bmatrix}1\\5\\3\end{bmatrix} \overset{?}{\gtrless} \begin{bmatrix}0\\6\\1\end{bmatrix}. \tag{2.18}$$

When dealing with multi- or hyperspectral images, pixels are represented by vectors. Direct use of Mathematical Morphology on such data is not possible due to the loss of unambiguous ordering relation. Several solution can be used. One, based on dimension reduction using a bijective function, is proposed in Section 2.8.

2.6 Morphological Profile

2.6.1 Definition

The morphological profile (MP) was proposed by M. Pesaresi and J. A. Benediktsson, in [101], for the segmentation of high-resolution remotely-sensed images. It is based on two morphological tools:

1. Geodesic transform.
2. Granulometry and anti-granulometry.

Geodesic operators were used, owing to their connectedness property and their ability to preserve original shapes in the processed image. The use of different sizes for the structuring element (SE), leading to granulometry, was motivated by the necessity of exploring all sizes of structure within the image [101].

An MP is made up of an *opening profile* (OP) and a *closing profile* (CP). The OP at pixel $\mathbf{x}$ of image f is defined as an n-dimensional vector:

$$OP_i(\mathbf{x}) = \gamma_R^{(i)}(\mathbf{x}),\ \forall i \in [0, n] \tag{2.19}$$

where $\gamma_R^{(i)}$ is the opening by reconstruction using an SE of size i and n is the total number of openings. Further, the CP at pixel $\mathbf{x}$ of image f is defined as an n-dimensional vector:

$$CP_i(\mathbf{x}) = \phi_R^{(i)}(\mathbf{x}),\ \forall i \in [0, n] \tag{2.20}$$

where $\phi_R^{(i)}$ is the closing by reconstruction using 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 the $(2n+1)$-dimensional vector:

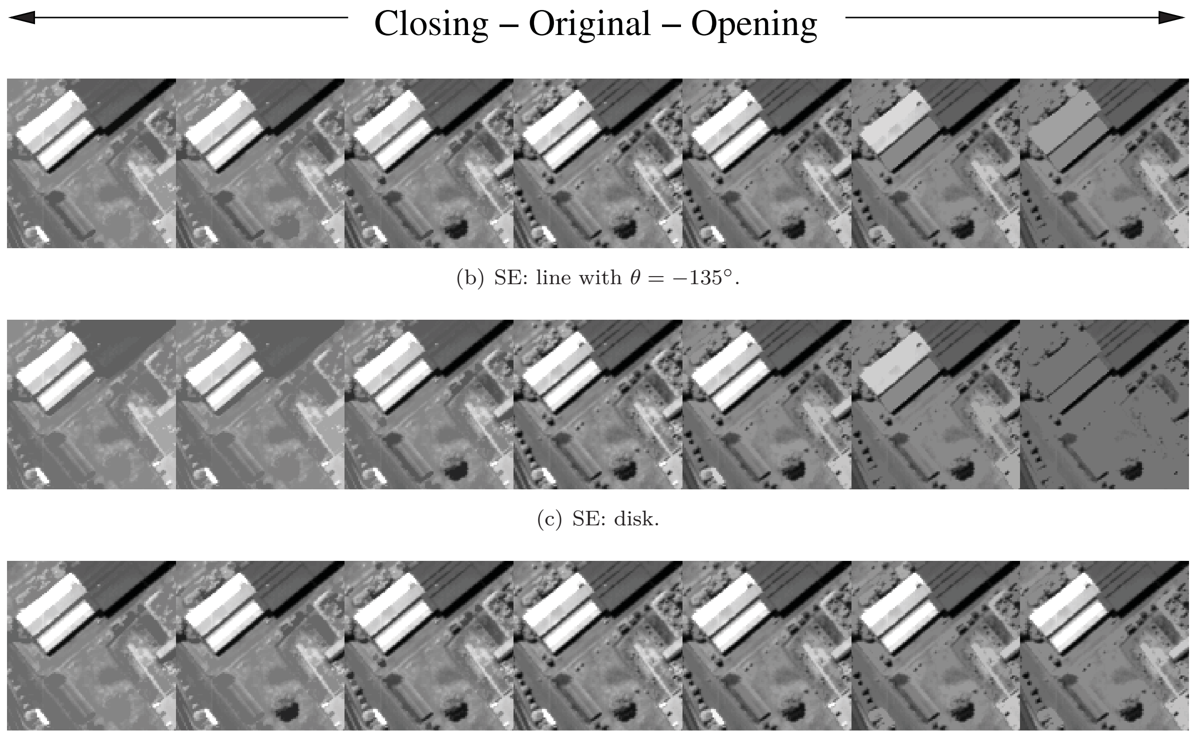
$$MP(\mathbf{x}) = \{CP_n(\mathbf{x}), \dots, f(\mathbf{x}), \dots, OP_n(\mathbf{x})\} \tag{2.21}$$

$$MP_i(\mathbf{x}) \begin{cases} = & CP_{n-i}(\mathbf{x}) & \text{if } 0 \leq i < n \\ = & f(\mathbf{x}) & \text{if } i = n \\ = & OP_i(\mathbf{x}) & \text{if } n < i \leq 2n. \end{cases} \tag{2.22}$$

The *Derivative of the Morphological Profile* (DMP) is defined as the $2n$-dimensional vector equal to the discrete derivative of MP.

$$DMP_i(\mathbf{x}) = MP_{i-1}(\mathbf{x}) - MP_i(\mathbf{x}) \tag{2.23}$$

Information provided by the MP-DMP is both spatial and radiometric. For a given pixel, the fact the DMP is centered should signify that the pixel belongs to a structure that is small compared to the SE used to derive the DMP. On the other hand, an unbalanced DMP (to the left or right) should indicate that the pixel belongs to a large structure. Further, the imbalance of the profile indicates whether the pixel belong to a structure that is darker (left side) or lighter (right side) than the surrounding ones. Finally, the amplitude of the DMP yields information about the local contrast of the structure.



(b) SE: line with $\theta = -135°$.

(c) SE: disk.

(d) SE: line with $\theta = 45°$.

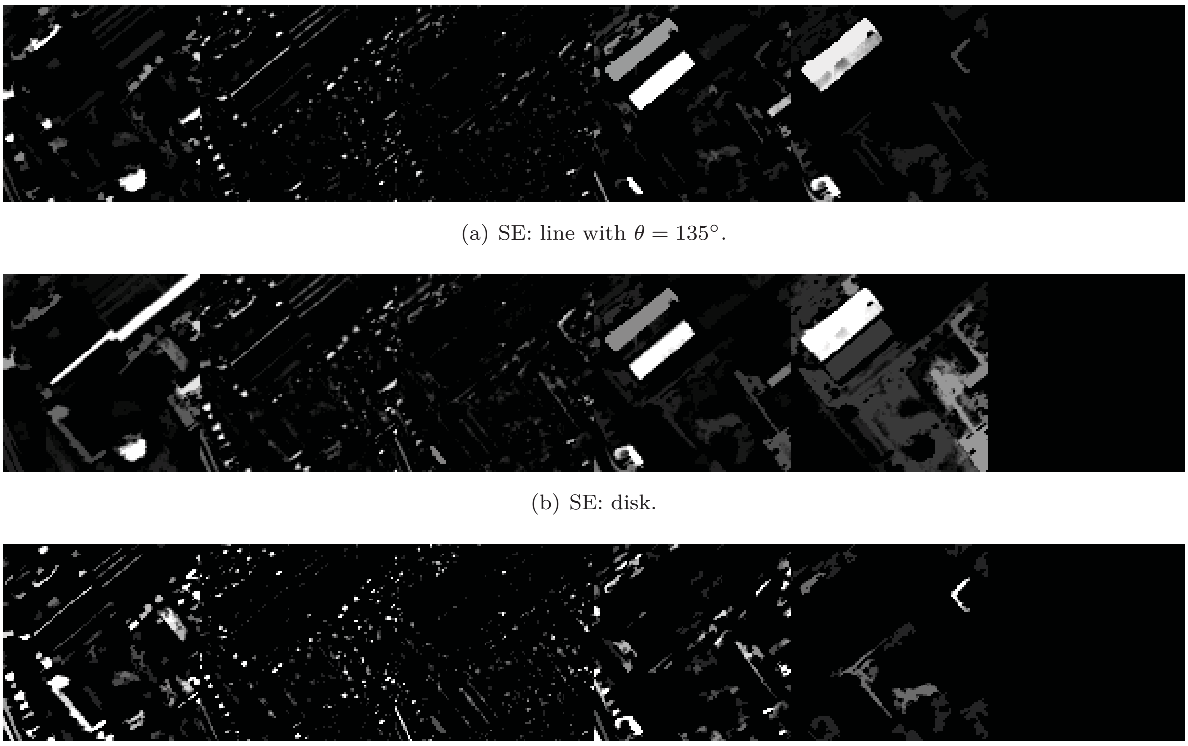
Figure 2.7: *MP with different SE. The MP were built with a line of length {3,12,21} for two orientation and a disk of radius {2,8,12}, respectively.*

2.6.2 Construction

To construct the MP-DMP, we first have to choose the structuring element (SE) for the opening/closing by reconstruction. The spatial characteristics of this SE will determine the information contained in the MP. For example, if a line of n pixels with a direction θ is used as the SE, linear structures of the same direction θ will be preserved and thus will be removed at the end of the MP-DMP, *i.e.*, for an opening with a large SE. From the point of view of classification point of view, it is better to use an isotropic SE to get more information – *e.g.*, considering roads, we would like to have the same MP whatever the orientation of the road. One possible isotropic SE is the discrete disk. Examples of MP-DMP using different SEs are shown in figures 2.7 and 2.8. In figure 2.7(b), the SE was a line with $\theta = 135°$, which is almost perpendicular to the direction of the two light buildings, while in figure 2.7(d), $\theta = 45°$ which is almost in the same direction. For the first MP we can clearly see that the buildings disappear in the opening part of the MP, while in the other profile they are preserved. When a disk is used, figure 2.7(c), we get a more homogeneous MP where small structures are removed first and then larger structures.

2.6.3 Classification using the MP-DMP

The MP and DMP can be used for the automatic detection of structures in the images [117, 103, 102]. For classification, the MP-DMP are regarded as feature vectors, where each class has a typical MP-DMP. Thus classification can be performed using any standard algorithms [8, 53]. Another possible interpretation is to regard the DMP as a possibility distribution and use fuzzy logic to classify pixels [27]. In the following we are only going to consider the MP-DMP as feature vectors, so if they were constructed



(a) SE: line with $\theta = 135°$.

(b) SE: disk.

(c) SE: line with $\theta = 45°$.

Figure 2.8: *DMP with different SE. The DMP is derivative of the MP. Clear structures indicate when they disappear in the MP.*

using n openings/closings by reconstruction, then:

$$MP(x) \in \mathbb{R}^{2n+1} \text{ and } DMP(x) \in \mathbb{R}^{2n}. \tag{2.24}$$

MP was applied first to panchromatic data to extract information about the size, shape, and local contrast of the structures [8]. Classification was performed using a neural network.

We present here some results using support vectors machines. The data used is an IKONOS panchromatic image from Reykjavik, Iceland. It is 975 × 639 with 1-m spatial resolution. We concentrated on six information classes, see Table 2.1, as in the original experiments [8, 53]. Full description of the data set is provided in Appendix C. The MP was constructed using 15 openings/closings with a disk as SE. Its radius was 2, 4, . . . , 30. Results of the experiment are listed in Table 2.2.

In terms of classification accuracy, using the original image alone, Figure 2.1, does not make it possible to distinguish between streets and residential lawns. Only the shadows are correctly classified. The MP and DMP provide information that makes it possible to discriminate between streets and residential lawns. The best overall and average accuracies were achieved using the MP. The DMP does not contain the original gray levels, but only the spatial information. Using the DMP improves classification accuracy slightly, whereas it is greatly improved when using both the spatial and the spectral information, *i.e.*, the MP.

This experiment confirms the usefulness of the MP over the use of the original image alone. However, the necessity of looking at a range of sizes of SE may result in redundancy in the MP. A greater problem, however, is the inherent nature of the opening/closing operators. This point is discussed in the next subsection and another approach is detailed.

Table 2.1: *IKONOS Reykjavik image: Training and testing sets.*

Class		Samples	
No	Name	Train	Test
1	Small Building	1526	34155
2	Open area	7536	25806
3	Shadows	1286	43867
4	Large Building	2797	39202
5	Street	3336	30916
6	Residential Lawns	5616	35147
Total		22.097	209.093

Table 2.2: *Classification accuracy for the IKONOS image.*

	OA	AA	K	1	2	3	4	5	6
Gray-level	42.87	44.59	31.02	40.79	58.37	85.50	25.82	0	57.04
MP	50.84	51.74	40.80	63.89	28.60	71.56	22.36	72.47	51.65
DMP	44.51	43.36	32.83	63.74	33.78	28.56	17.67	67.30	49.13
Self-com. Area	48.46	49.76	37.50	33.52	70.24	89.94	24.73	0.88	79.27

2.7 Area Filtering

Geodesic opening and closing filters are interesting because they preserve shapes. However, they cannot provide a complete analysis of urban areas because they only act on the *extrema* of the image. Moreover, some structures may be darker than their neighbors in some parts of the image, yet brighter than their neighbors in others. Although this problem can be partially addressed by using an alternate sequential filter [28], the MP, which conventionally processes extrema, thus provides an incomplete description of the inter-pixel dependency.

In [121], P. Soille has proposed using self-complementary filters[2] to analyze all the structures of an image, local *extrema*, be they *minima* or *maxima*, as well as regions with intermediate grey levels. This only assumes that any given structure of interest corresponds to one set of connected pixels. Based on an area criterion, a flat zone filter is proposed to remove small structures. This kind of filter is well suited to the analysis of panchromatic images: the very high spatial resolution results in excessively detailed data containing many irrelevant structures (*e.g.*, cars on the road).

As will be detailed in the following, the area self-complementary filter is not a morphological filter, since the increasingness property no longer holds. Thus the strategy used with the MP cannot be directly applied. In this work, we proposed another approach to extract the contextual information. The idea is to build an *adaptive neighbors systems for each pixel* [51].

Standard methods for the analysis of spatial information in panchromatic images usually define the neighborhood of each pixel by a fixed set such as 3×3 squares[21, 17]. This strategy fails when considering pixels that belong to the border of a structure: the fixed shape neighborhood then includes pixels from different features. For example, as shown in figure 2.9.(a), the classification of the marked pixel (roof) may be influenced by neighboring pixels actually belonging to the street.

It is clear that a pixel's neighbors depend on the structure to which it belongs. Thus a neighborhood

[2] The definition of self-complementary filter is given Section 2.7.1 page 43.

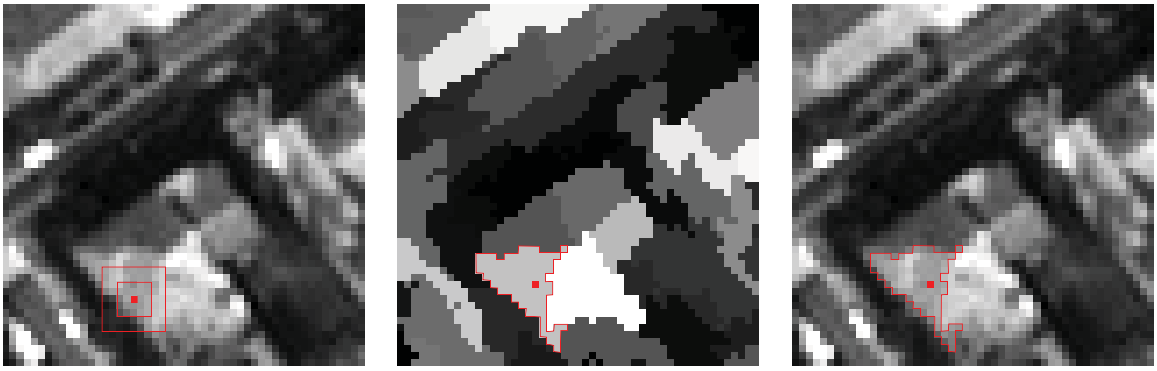


Figure 2.9: *Inter-pixel dependency estimation.*

system using a fixed shape or size is unable to define the neighborhood system correctly. In this work, we propose using advanced morphological tools to define the neighbors. The idea is to define neighbors in a *structure meaning*. We consider that is important to look at neighboring pixels that belong to the *same* structure. Furthermore, we think that the same inter-pixel dependency exists between pixels of the same structure, even if such structures are disjoint in the image. For example, roads of the same type have the same texture everywhere in the image. Hence for classification we need to look at the spectrum of a pixel, but also at the neighbors belonging to same structure as the considered pixel.

To solve this problem, an adaptive neighborhood has to be defined for each pixel. Furthermore, assuming that relevant structures have a sufficient area, this adaptive neighborhood should include a large number of pixels. Hence we propose defining the neighborhood of one given pixel as the resulting connected zone of the self-complementary area filter. This procedure is illustrated in figure 2.9. Figure 2.9.(b) is the area filtering of figure 2.9.(a). It is partitioned into flat zones and each flat zone belongs to a single structure in the original image. All the pixels belonging to one given flat zone are regarded as neighbors. Lastly, the inter-pixel dependency information is extracted from the original image by using the previously-defined set of neighbors (for every pixel, the neighborhood is defined as the flat zone to which it belongs). Figure 2.9.(c) highlights the neighborhood associated with the marked pixel. It is obviously more homogeneous and spectrally-consistent than the square shown in figure 2.9.(a).

2.7.1 Self-complementary area filters

Definition 2.12 (Complementariness) *Two operators Ψ and Θ are complementary if and only if:*

$$\Psi(f) = \Theta([f]^c). \tag{2.25}$$

A self-complementary operator is defined as an operator whose complementary operator is itself [120]:

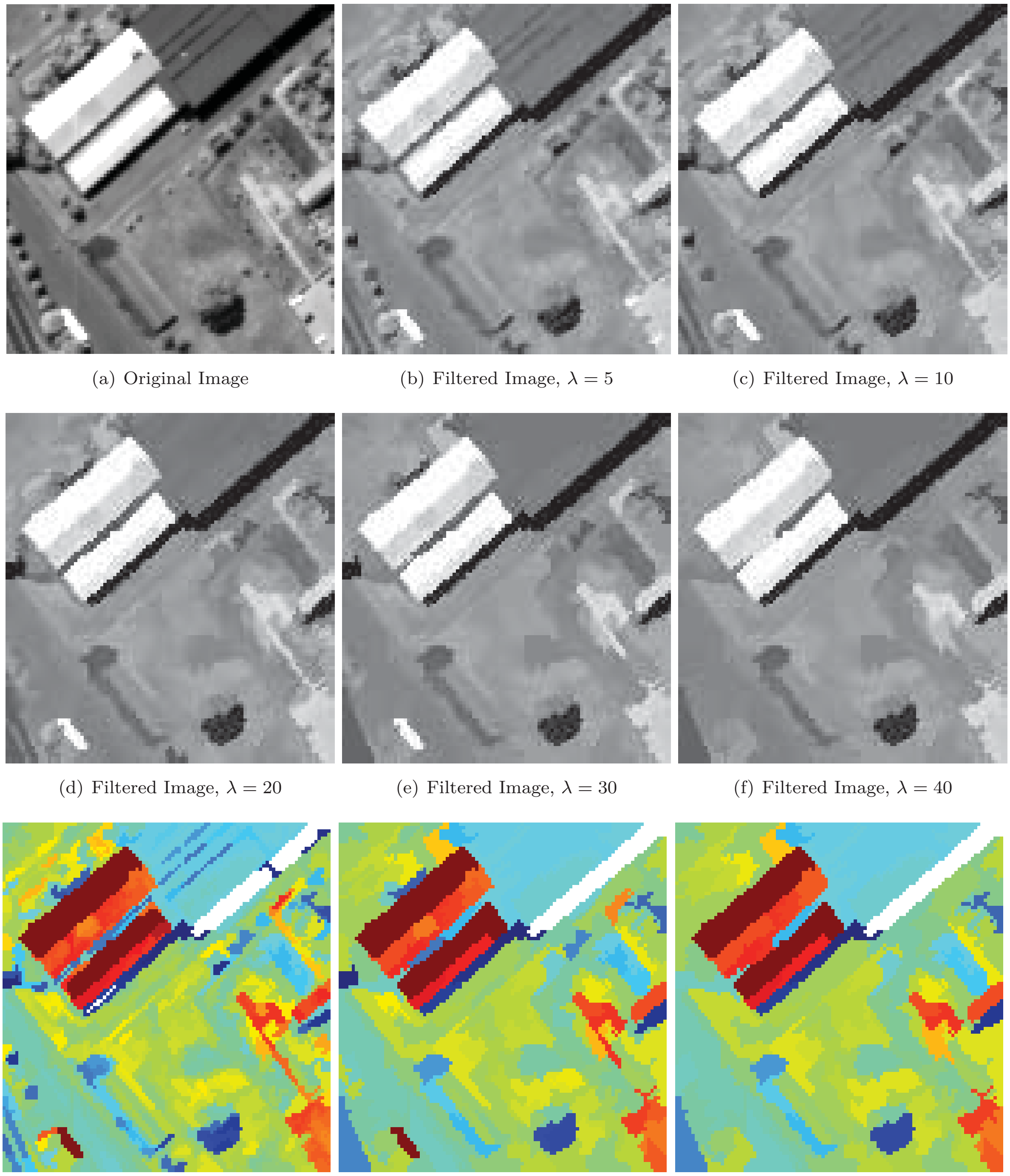
$$\Psi = \mathbf{C}\Psi. \tag{2.26}$$

Area self-complementary filters have been introduced to extend area opening and closing to all the structures of the image, not just its local *extrema* [121]. It is a two-step algorithm, involving:

1. Labelling all the flat zones that satisfy the area criterion λ [120],
2. Growing the labelled flat zones until an image partition is reached [2].

Better image structure preservation is achieved by iterating the algorithm until the desired size is obtained, *e.g.*, let f be the image to process, then $\Psi_\lambda(f) = \Psi_\lambda(\Psi_{\lambda-1}(\ldots(\Psi_2(f))))$, where λ is the minimum size of the remaining flat structures.

Figure 2.10 shows the results of area filtering using different value for the area parameter.



(a) Original Image

(b) Filtered Image, $\lambda = 5$

(c) Filtered Image, $\lambda = 10$

(d) Filtered Image, $\lambda = 20$

(e) Filtered Image, $\lambda = 30$

(f) Filtered Image, $\lambda = 40$

(g) Neighbors set for (c)

(h) Neighbors set for (e)

(i) Neighbors set for (f)

Figure 2.10: *Area self-complementary filtering and corresponding neighbors set. In images (g), (h), and (f), each color corresponds to a set of neighbors.*

2.7.2 Extracting the inter-pixel dependency

Once the neighborhoods have been defined for every pixel, spatial information can be estimated. In view of the small average size of the neighbors set (about 50 pixels, in order to preserve the smallest structures of interest), computing high-order statistics may not be appropriate. As can be seen from Fig. 2.9.(b), the neighbors sets do not have spatial orientation related to the actual structure to which they belong. As a result, computing some shape descriptors for the neighborhoods may not be useful either. Hence for each pixel $\mathbf{x}$, we propose simply computing the median value of the neighbors set $\Omega_{\mathbf{x}}$:

$$\Upsilon_{\mathbf{x}} = \text{med}(\Omega_{\mathbf{x}}). \tag{2.27}$$

The pseudo-code of the process is given in Algorithm 3.

Now, every pixel of the panchromatic image is characterized by its original gray-level (the spectral information) and by its inter-pixel dependency (the spatial information). The easiest way to use both pieces of information is to build a stack vector. This strategy was used for the morphological profile [8]. However, a strategy that makes it possible to control the amount of each type of information ought to be preferable [21, 91]. This can easily be achieved using an appropriate *kernel formulation*. This point is addressed in Chapter 4.

For comparison with MP-DMP, experiment were performed on the same IKONOS image. Results are reported in Table 2.2. In terms of the global accuracy values (OA, AA, and κ), the approach proposed performs slightly worse than the morphological one. The result are degraded by the 'street' class. This is consistent with the fact that we did not extract information about the size of the structures, but only about the grey-level distribution of pixels contained within the flat zone. More comments about this strategy are given in chapter 4.

Note that other spatial information can be extracted, such as a 'texture' feature. This feature should be as uncorrelated as possible to the spectral feature. The best features should be gray-level independent and isotropic.

All the methodologies presented – MP-DMP and area filter – act on a single-band image. Multi-valued images such as multi- or hyperspectral data also have meaningful spatial information. Nevertheless, extracting contextual information is more difficult, and we are faced in particular with the problem of an ordering relation for vector-valued pixels.

Algorithm 3 *Adaptive Neighbors Extraction*

```
Require: Original Image, $\lambda$
$\lambda_{tp} = 2$
while $\lambda_{tp} \leq \lambda$ do
    Label flat zones and keep only those which satisfy the area criterion $\lambda_{tp}$
    Grow the selected flat zone until the entire image has been processed
    $\lambda_{tp} = \lambda_{tp} + 1$
end while
for all the remaining flat zones do
    Compute the median value of the gray-level pixel, in the original image, that belong to the same flat zone
end for
```

2.8 Extension to multi-valued image

Extension of morphological tools to non-flat images is not straightforward. As presented in Section 2.5, the ordering relation no longer holds. Two strategies can be employed:

1. Attempt to extend filters to non-flat images,
2. Reduce the dimension of the data to obtain a one-dimensional image.

The first approach is certainly more promising in terms of image filtering results. However, to extend morphological filtering to multi-valued images, theoretical considerations would have to be taken into account that are beyond the scope of this thesis. Interested readers will find material about multi-valued images and morphological processing in [26, 29, 60, 79].

By confining our considerations to the problems of classification, a simpler approach based on the second strategy was used in [98, 7] where principal component analysis was used to reduce the dimensionality of hyperspectral data. The PCA was used to extract representative images since PCA is optimal for representation in mean square errors sense.

Finally the extension of the MP is straightforward: the first few principal components are used to build several MPs. Then, the MPs are stacked together to form the extended morphological profile [7]. Following the same scheme, the neighbors sets are defined on the first principal component, and are then generalized onto each data band. Experiments adopting these strategies are conducted in Chapter 4.

2.9 Conclusion

In this chapter, the data analysis was performed in the spatial space. Based on mathematical morphology, spatial features about shape, size, and local contrast were extracted for both panchromatic and hyperspectral data. Limitations of the conventional approach have been pointed out, and an area self-complementary filter was proposed to deal with intermediate regions. An adaptive definition of neighbors for each pixel is proposed.

Moreover, it is clear that correlation may exist between adjacent pixels and thus pixels should not be assumed to be independent when classifying pixels in the spectral domain. However, this assumption is still widely used in the classification of remote-sensing data. In the next part of this thesis, emphasis will be placed on the use of a joint classifier, using both spatial and spectral information. Support vector machines will also be addressed in particular.

Part II

Classification

Chapter 3

Support Vector Machines

Abstract

Support Vectors Machines for the classification of remote-sensing data are reviewed in this chapter. The SVMs have good capabilities for handling problems related to the classification of remote-sensing data: robustness to dimensionality, good generalization ability, and a non-linear decision function. These properties are discussed in the chapter and simulated experiments are given. Then experiments on real data are presented and several problems are addressed: the selection of the kernel parameters, the training of SVMs, and multi-class strategy. Based on our experiments, very good classification accuracies are possible using a standard formulation of the SVM, and an improvement in terms of processing time is possible using training strategy based on gradient descent.

Contents

THIS CHAPTER is devoted to the classification of remote-sensing data. In the first chapter, several problems related to the potential high dimensionality of the data were presented. The first strategy was to reduce the dimensionality so as to be able to apply existing classifiers. However, the problem of combining spatial and spectral information still remains. Some strategies do exist in the remote-sensing literature [68, 107], mostly based on *Markov random field* theory [86], but the problem is still not completely addressed: dimension reduction is still needed, and normal Markov modeling may suffer as a result of the very high spatial resolution.

Since the beginning of 21^{st} century, classifiers based on *statistical learning theory* have shown remarkable abilities to deal with both high-dimensional data and a limited training set [126, 127]. One of the best-known methods is undoubtedly the *Support Vector Machines* (SVM) [126, 127, 64, 112, 39]. SVM have found applications in many pattern recognition problems: text categorization [22, 71], hand-written character recognition [126, 127], image classification [33] and bio-informatics [114].

Naturally, SVMs have been investigated for the analysis of remote sensing data purposes. In [61], SVMs were used to classify species from AVIRIS hyperspectral data. Comparison with conventional classifiers confirmed the good performance of the SVMs. In [54, 90], the authors have compared SVMs favorably to more advanced classifiers like neural networks, discriminant analysis, and decision trees. In all cases, SVMs outperformed the other classifiers. Multisource classification, where statistical modeling is not usually possible, is also addressed in [63].

The main properties of an SVM classifier can be summarized as [127]:

1. Distribution-free classification approach,
2. Training step is reduced to a convex optimization problem,
3. Linear classifier in some *feature space*,
4. Non-linear classifier in the *input space*.

The remainder of this chapter is organized as follows. In the first section, the SVM is introduced, along with its theoretical background. Emphasis is placed on the interest of this notion for remote-sensing applications. Next, as for principal component analysis, the commonly-used non-linear SVM is presented using kernel methods. Then we review some existing approaches using the SVM for remote-sensing data analysis. Working from the basis of existing results, we then attempt to answer some of the usual questions: the choice of kernel, parameters selection, and training strategy. Experimental results are presented in the last section.

3.1 Linear Classifier

3.1.1 Introduction to the classification problem

The SVM belongs to the family of classification algorithms that solve a *supervised learning problem*: given a set of samples with their corresponding classes, find a function that assigns each sample to its corresponding class. The aim of statistical learning theory is to find a *satisfactory* function that will correctly classify training samples and unseen samples, *i.e.*, that has a low generalization error. The basic setting of such a classification problem is as follows. Given a training set $\mathcal{S}$:

$$\mathcal{S} = \big\{(\mathbf{x}^1, y_1), \dots, (\mathbf{x}^\ell, y_\ell)\big\} \in \mathbb{R}^n \times \{-1, 1\} \tag{3.1}$$

generated i.i.d. from an unknown probability law $\mathcal{P}(\mathbf{x}, y)$ and a loss function L, we want to find a function f from a set of functions $\mathcal{F}$ that minimizes its expected loss, or *risk*, $R(f)$:

$$R(f) = \int_{\mathcal{S}} L(f(\mathbf{x}), y) d\mathcal{P}(\mathbf{x}, y). \tag{3.2}$$

Unfortunately, since $\mathcal{P}(\mathbf{x}, y)$ is unknown, the above equation cannot be computed. However, given $\mathcal{S}$, we can still compute the *empirical risk*, $R_{emp}(f)$:

$$R_{emp}(f) = \frac{1}{\ell} \sum_{i=1}^{\ell} L(f(\mathbf{x}^i), y_i) \tag{3.3}$$

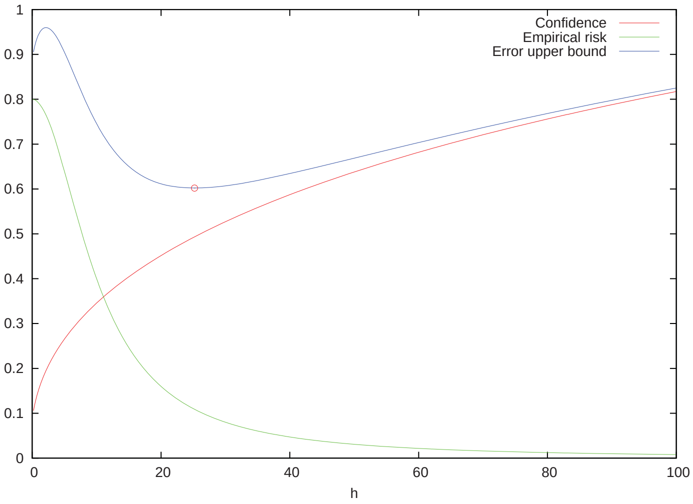


Figure 3.1: *Example of (3.5). The circle represents the minimum error upper bound.*

and try to minimize that. This principle is called *Empirical Risk Minimization* (ERM) and is employed in conventional learning algorithms, *e.g.*, neural networks. The law of large numbers ensures that if f_1 minimizes R_{emp}, we have

$$R_{emp}(f_1) \longrightarrow R(f_1) \tag{3.4}$$

as ℓ tends to infinity. But f_1 is not necessarily a minimizer of R. So the minimization of (3.3) could yield an unsatisfactory solution to the classification problem. An example, arising from the *No free lunch* theorem [130], is that given one training set it is always possible to find a function that fits the data with no error but which is unable to classify a single sample from the testing set correctly.

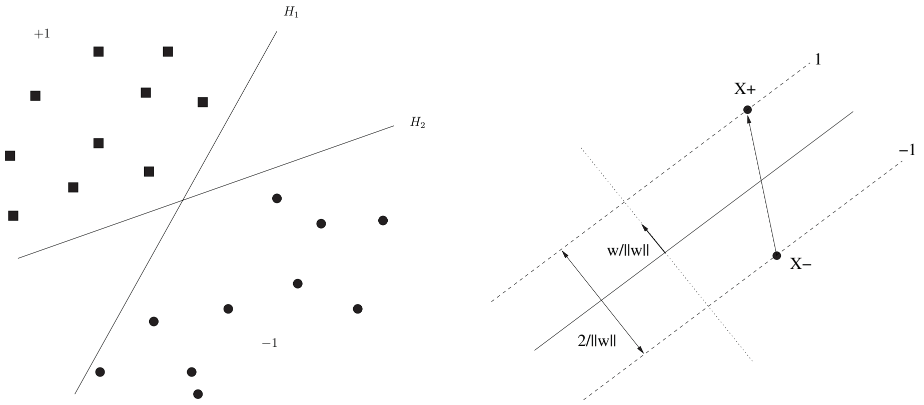
To solve this problem, the classic *Bayesian* approach consists of selecting a distribution *a priori* for $\mathcal{P}(\mathbf{x}, y)$ and then minimizing (3.2). In statistical learning, no assumption is made as to the distribution, but only about the *complexity* of the class of functions $\mathcal{F}$. The main idea is to favor *simple* functions, to discard over-fitting problems, and to achieve a good generalization ability [93]. One way of modeling the complexity is given by the VC[1] theory [126]: the complexity of $\mathcal{F}$ is measured by the VC dimension h, and the *structural risk minimization* (SRM) principle allows to select the function $f \in \mathcal{F}$ that minimizes an upper bound error [126, 127]. Hence, the upper bound is defined as a function depending on R_{emp} and h. For example, given a set of functions $\mathcal{F}$ with VC dimension h and a classification problem with a loss function $L(\mathbf{x}, y) = \frac{1}{2}|y - f(\mathbf{x})|$, then for all $1 > \eta > 0$ and $f \in \mathcal{F}$ we have [126, 127]:

$$R(f) \leq R_{emp}(f) + \sqrt{\frac{h\left(\ln(\frac{2\ell}{h}) + 1\right) - \ln(\frac{\eta}{4})}{\ell}} \tag{3.5}$$

with probability of at least $1 - \eta$ and for $\ell > h$. Following VC theory, the training step of this classifier should minimize the right terms of the inegality 3.5. Figure 3.1 shows an toy example of (3.5). Other bounds can be found for different loss functions and measures of complexity in [126].

In the following, we are going to present one particular class of function that leads to a linear decision function and the definition of the SVM classifier.

[1] Vapnik-Chervonenkis



(a) (b)

Figure 3.2: *(a) Linear classification with separating hyperplane, (b) Margin of the canonical separating hyperplane.*

3.1.2 Linear support vector machines

Definition 3.1 (Linear decision function) *A linear decision function f is a function from a vector space $\mathcal{X}$ to the real space $\mathbb{R}$ with the following form, with $\mathbf{w} \in \mathcal{X}$ and $b \in \mathbb{R}$:*

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{X}} + b. \tag{3.6}$$

The decision boundary $\ker(f) = \{\mathbf{x} | f(\mathbf{x}) = 0\}$ leads to the definition of the *separating hyperplane.* Linear classifiers g are derived by considering the sign of such a function to solve problem (3.1): $y = g(\mathbf{x}) = \operatorname{sgn}(f(\mathbf{x}))$.

Definition 3.2 (Separating hyperplane) *Given a supervised classification problem (3.1), a separating hyperplane $H(\mathbf{w}, b)$ is a linear decision function that separate the space into two half-spaces, each half-space corresponding to the given class,* i.e.*, $sgn\left(\langle \mathbf{w}, \mathbf{x}_i \rangle_{\mathcal{X}} \ + \ b\right) \ = \ y_i$ for all samples from $\mathcal{S}$.*

We have the equivalence between the linear classifier (functional viewpoint) and the separating hyperplane (geometrical viewpoint).

Definition 3.3 (Canonical separating hyperplane) *A canonical separating hyperplane verifying:*

$$\min_{\mathbf{x} \in \mathcal{X}} \left(|\langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{X}} + b| = 1 \right). \tag{3.7}$$

Every separating hyperplane can be transformed into a canonical one, since the decision boundary of a hyperplane is defined up to a multiplicative constant:

$$\langle \mathbf{w}, \mathbf{x} \rangle_{\mathcal{X}} + b = 0 \iff \langle \frac{\mathbf{w}}{q}, \mathbf{x} \rangle_{\mathcal{X}} + \frac{b}{q} = 0, \ q \neq 0. \tag{3.8}$$

In the following, we will only consider real n-dimensional vector space $\mathcal{X} = \mathbb{R}^n$. For remote-sensing applications, n usually represents the number of spectral bands that the sensor can collect, *e.g.*, for panchromatic data $n = 1$, for multispectral data $n \in [4, 10]$ and for hyperspectral data $n \gg 10$.

On the assumption that the data are linearly separable, the learning problem is to find the parameters $(\mathbf{w}, b)$. As can be seen from Figure 3.2.(a), several separating hyperplanes can be acceptable. Following the VC theory, we want to find the hyperplane that minimizes the upper bound of the risk (3.5). For the class of separating hyperplanes, the VC dimension can be estimated by the *margin* [119], where the margin is defined as the minimum distance of a training sample from the decision boundary, be correctly classified or not.

The margin can be computed as follows (see Figure 3.2.(b)): consider the canonical hyperplane and the samples that are the closest to the hyperplane; the margin is given by the distance between these two samples $(\mathbf{x}_+ - \mathbf{x}_-)$ projected onto the unitary vector normal to the hyperplane: $\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}_+ - \mathbf{x}_- \rangle = \frac{2}{\|\mathbf{w}\|}$. So canonical hyperplanes have their margin equal to twice the inverse of $\|\mathbf{w}\|$. And so finally the link between the margin and the VC dimension h is [126, 93]:

$$h \leq \min\left(\|\mathbf{w}\|^2 \mathcal{R}^2, n\right) + 1 \tag{3.9}$$

where $\mathcal{R}$ is the radius of the smallest hypersphere around the data (see A.4).

Lastly, the learning strategy is to minimize the right terms of (3.5):

- The empirical risk: by constraining the hyperplane parameters to give perfect classification of the training set, *i.e.*, $y_i\left(\langle \mathbf{w}, \mathbf{x}^i\rangle_{\mathbb{R}^n} + b\right) \geq 1,\ \forall i \in 1, \ldots, \ell$.
- The complexity term: this is a monotonically increasing function of the variable h. So it can be minimized by considering h and (3.9), where $\mathcal{R}$ is fixed by the training samples. Thus the complexity term is minimized by minimizing $\|\mathbf{w}\|^2$ [93].

This is a constraint quadratic optimization problem we need to solve:

$$\begin{array}{ll} \text{minimize} & \dfrac{\langle \mathbf{w}, \mathbf{w}\rangle_{\mathbb{R}^n}}{2} \\ \text{subject to} & y_i\left(\langle \mathbf{w}, \mathbf{x}^i\rangle_{\mathbb{R}^n} + b\right) \geq 1,\ \forall i \in 1, \ldots, \ell. \end{array} \tag{3.10}$$

It is usually solved using Lagrange multipliers [18]. The Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{\langle \mathbf{w}, \mathbf{w}\rangle_{\mathbb{R}^n}}{2} + \sum_{i=1}^{\ell} \alpha_i \left(1 - y_i\left(\langle \mathbf{w}, \mathbf{x}^i\rangle_{\mathbb{R}^n} + b\right)\right) \tag{3.11}$$

have to be minimized with respect to the variables $\mathbf{w}, b$ and maximized with respect to α_i. At the optimal point, the gradient vanishes:

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{w} - \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}^i \quad = \quad 0, \tag{3.12}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{\ell} \alpha_i y_i \quad = \quad 0. \tag{3.13}$$

From (3.12), we can see that $\mathbf{w}$ lives in the subspace spanned by the training samples: $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}^i$. By substituting (3.12) and (3.13) into (3.11), we get the dual quadratic problem, with only one variable α_i:

$$\begin{array}{lcl} \max\limits_{\alpha}\ g(\alpha) & = & \displaystyle\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^i, \mathbf{x}^j\rangle_{\mathbb{R}^n} \\ \text{subject to} & & 0 \leq \alpha_i \\ & & \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{array} \tag{3.14}$$

When this dual problem is optimized, we obtain α_i and hence $\mathbf{w}$. This leads to the decision rule:

$$g(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i \langle \mathbf{x}, \mathbf{x}^i\rangle_{\mathbb{R}^n} + b\right). \tag{3.15}$$

The constraints assume that the data are linearly separable. For real applications, this might be too restrictive, and this problem is traditionally solved by considering *soft margin* constraints: $y_i\left(\langle \mathbf{w}, \mathbf{x}^i\rangle_{\mathbb{R}^n} + b\right) \geq$

$1+\xi_i$ which allow some training errors during the training process and use an upper bound of the empirical risk $\sum_{i=1}^{\ell} \xi_i$. The optimization problem changes slightly to:

$$\begin{aligned} &\text{minimize} && \frac{\langle \mathbf{w}, \mathbf{w} \rangle_{\mathbb{R}^n}}{2} + C \sum_{i=1}^{\ell} \xi_i \\ &\text{subject to} && y_i \left(\langle \mathbf{w}, \mathbf{x}^i \rangle_{\mathbb{R}^n} + b \right) \geq 1 - \xi_i, \ \forall i \in 1, \dots, \ell \\ & && \xi_i \geq 0, \ \forall i \in 1, \dots, \ell. \end{aligned} \tag{3.16}$$

C is a constant controlling the number of training errors. The Lagrangian becomes

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\langle \mathbf{w}, \mathbf{w} \rangle_{\mathbb{R}^n}}{2} + \sum_{i=1}^{\ell} \alpha_i \left(1 - \xi_i - y_i \left(\langle \mathbf{w}, \mathbf{x}^i \rangle + b \right)\right) - \sum_{i=1}^{\ell} \beta_i \xi_i + C \sum_{i=1}^{\ell} \xi_i \tag{3.17}$$

and the dual problem:

$$\begin{aligned} \max_{\alpha} g(\alpha) \quad &= \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^i, \mathbf{x}^j \rangle_{\mathbb{R}^n} \\ \text{subject to} \quad & \quad 0 \leq \alpha_i \leq C \\ & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{aligned} \tag{3.18}$$

Ultimately, the only change from (3.14) is the upper bound values of α_i. Considering the Karush-Kuhn-Tucker conditions at optimality [18]

$$\left\{ \begin{array}{rcl} 1 - \xi_i - y_i \left(\langle \mathbf{w}, \mathbf{x}^i \rangle + b \right) & \leq & 0 \\ \alpha_i & \geq & 0 \\ \alpha_i \left(1 - \xi_i - y_i \left(\langle \mathbf{w}, \mathbf{x}^i \rangle + b \right)\right) & = & 0 \\ \xi_i & \geq & 0 \\ \beta_i & \geq & 0 \\ \beta_i \xi_i & = & 0 \\ \dfrac{\partial L}{\partial \mathbf{x}} = \mathbf{w} - \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i \mathbf{x}^i & = & 0 \\ \dfrac{\partial L}{\partial b} = \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i & = & 0 \\ \dfrac{\partial L}{\partial \xi_i} = -\alpha_i - \beta_i + C & = & 0 \end{array} \right. \tag{3.19}$$

it will be seen that the third condition requires that $\alpha_i = 0$ or $\left(1 - \xi_i - y_i \left(\langle \mathbf{w}, \mathbf{x}^i \rangle + b \right)\right) = 0$. This means the solution $\boldsymbol{\alpha}$ is sparse and only some of the α_i are non zero. Thus $\mathbf{w}$ is supported by some training samples – those with non-zero optimal α_i. These are called the *support vectors*.

3.1.3 Application to synthetic remote-sensing data analysis

In the first chapter, problems related to the dimensionality of the representation space were addressed by means of feature reduction. The principal problem was the statistical estimation, owing to the reduced training set. The SVM involves margin maximization, which can be regarded as a geometric rather than a probabilist approach[2]. Therefore, no statistical estimate is needed. To aid understanding, we can reformulate the SVM principles from a geometric viewpoint:

[2] Note that the bound presented in the previous subsection holds in probability.

- The optimal hyperplane that separates the classes is found by considering the local configuration of the data in the dimensional space. This exploits the emptiness property of high-dimensional space.
- Through the optimality condition, we can expect the SVM to have good generalization ability, even in the situation of a limited training set.

To illustrate these properties, we have performed toy experiments. First, data with a Gaussian distribution were generated in $\mathbb{R}^2$. Then we constructed decision functions for three classifiers, where $\boldsymbol{\mu}$ is a mean vector and $\boldsymbol{\Sigma}$ a covariance matrix:

1. *Minimum distance to the mean classifier*: Samples are assigned to whichever class has the smallest Euclidean distance to its mean. The decision boundary is

$$\left\{\mathbf{x}|2\mathbf{x}^T(\boldsymbol{\mu}_- - \boldsymbol{\mu}_+) + (\boldsymbol{\mu}_+^T\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-^T\boldsymbol{\mu}_-) = 0\right\}. \tag{3.20}$$

2. *The SVM classifier*: We use the soft margin formulation. The decision boundary is

$$\left\{\mathbf{x}|\sum_{i=1}^{\ell}\alpha_i y_i\langle\mathbf{x}^i,\mathbf{x}\rangle + b = 0\right\}. \tag{3.21}$$

3. *Quadratic Bayesian classifier*: Under the Gaussian assumption and with the same covariance matrix, the decision boundary is

$$\left\{\mathbf{x}|\left((\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)^T\boldsymbol{\Sigma}^{-1}\right)\mathbf{x} - \frac{1}{2}(\boldsymbol{\mu}_+^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_-) = 0\right\}. \tag{3.22}$$

All these classifiers are linear and thus lead to a separating hyperplane. Two of them were based on statistical modeling, and one is the SVM. Two clusters were considered, linearly separable, and 100 samples per class were used for training, then 1,000 samples per class for testing. The different decision boundaries can be seen in Figure 3.3. The classification results are illustrated in Figure 3.4. Clearly, the last two classifiers perform better, which is not surprising since the minimum distance classifier is really a naive classifier. The SVM performs as well as the Bayesian classifier (which was implemented with the true *a priori* for the probability distribution). This first experiment shows that even if no *a priori* about the distribution was used, SVM performs very well.

The use of Gaussian data was not arbitrary since remote-sensing data are often assumed to have Gaussian distribution [80]. But in the case of hyperspectral data, they are many more than 2 dimensions. To test the behavior of SVM in the ”high dimensional space / small training set” situation, we conducted a second experiment. Multivariate Gaussian data were generated with increasing dimension, and classification was performed using the same three classifiers, but with a fixed training set of 40 samples per class. For each dimension, 100 experiments with 1,000 test samples per class were performed and the mean results are plotted in Figure 3.5. Again, the minimum distance classifier performs the worst, and as in the previous experiment, the SVM and Gaussian classifiers perform equally well for low and moderate dimensions. But, above a certain dimension (≈ 28), classification accuracy decreases for the Gaussian classifier. The problem is related to the estimated covariance matrix, which becomes badly conditioned and hence not invertible. Unlike the other classifiers, SVM does not suffer from the dimensionality and performs perfect classification.

These two experiments reveal some properties of SVM that render it suitable for remote-sensing applications. However, it is well known that classes of interest in remote sensing are partially overlapped [109]. Hence the choice of a linear function may not be optimal. Hopefully, the use of kernel methods will make it possible *to have both* the effective linear training model *and* the powerful discriminant ability of a non-linear model.

3.2 Non-linear SVM

Through the use of kernel methods, it is possible to build a non-linear SVM in a very elegant way. It should be noted that all the SVM training algorithms and decision rules are expressed in terms of inner

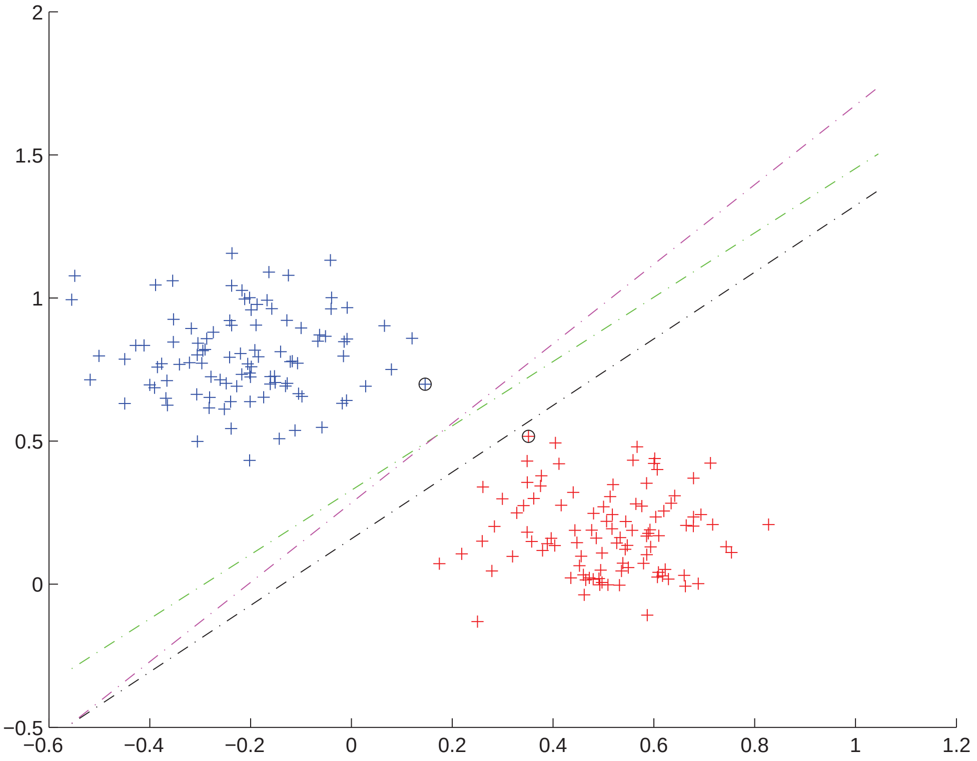
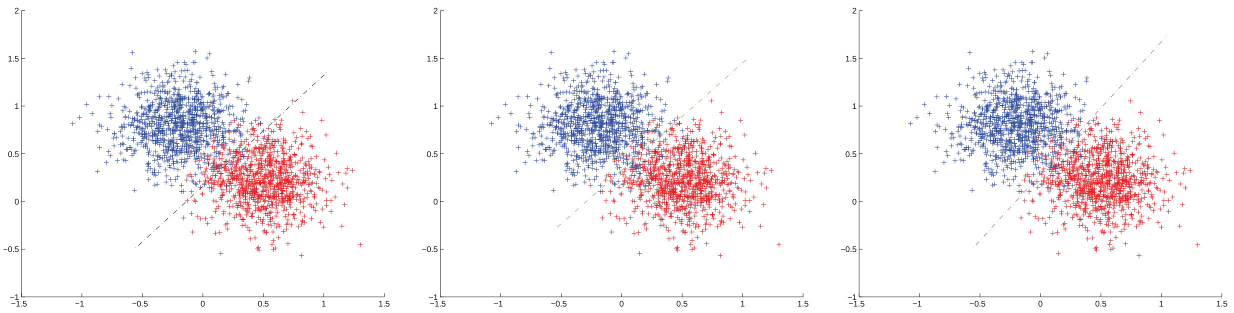


Figure 3.3: *Toy example: Two Gaussian clusters. Black circles: support vectors. Black line: decision boundary minimum distance classifier, green line: SVM, magenta line: Bayesian classifier.*



Minimum distance classifier

88.20%

SVM classifier

93,20%

Gaussian Bayesian classifier

92,60%

Figure 3.4: *Toy example: Two Gaussian clusters. Classification accuracies for the three classifiers. The percentages below the graphs are the overall classification accuracy.*

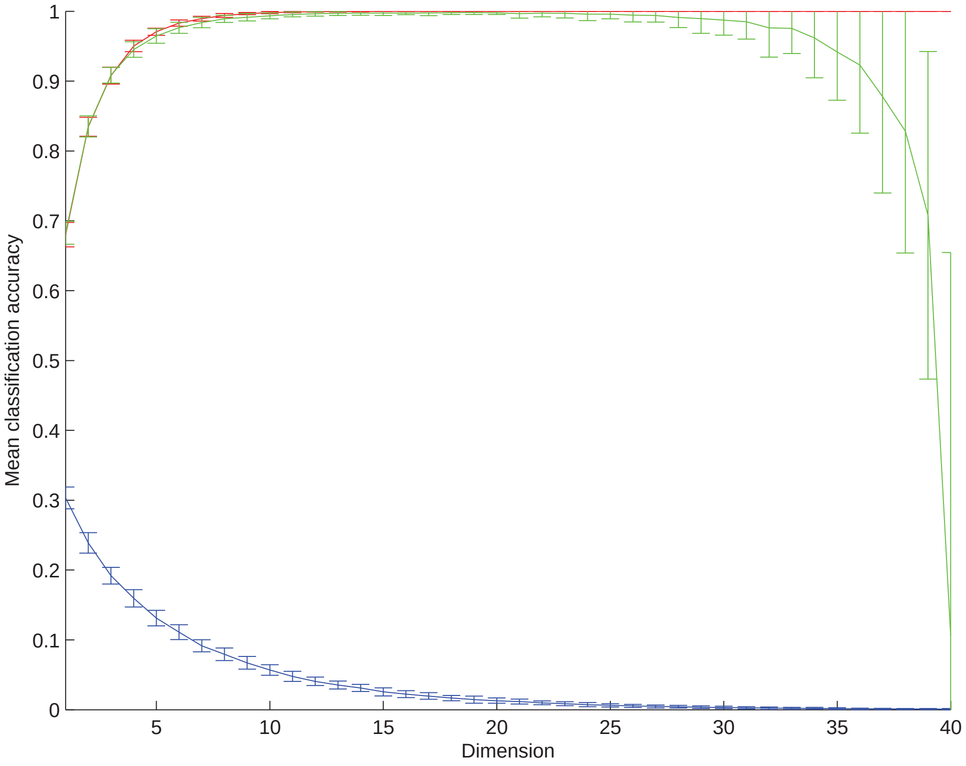


Figure 3.5: *Toy example: Two Gaussian clusters in high-dimensional space. Blue line: minimum distance classifier, Green line: Gaussian classifier, Red line: SVM. The line shows the mean classification accuracy and the bar is the standard deviation over 100 experiments.*

product. Just as with PCA, the *kernel trick* can be applied here too – see appendix A for details. The inner product in (3.18) is replaced by a kernel function $(\langle \mathbf{x}^i, \mathbf{x}^j \rangle \Rightarrow k(\mathbf{x}^i, \mathbf{x}^j))$:

$$\begin{aligned} \max_{\alpha} g(\alpha) \quad &= \quad \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j k(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} \quad & \quad 0 \leq \alpha_i \leq C \\ & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{aligned} \tag{3.23}$$

Now, the SVM is a non-linear classifier in the input space $\mathbb{R}^n$, but is still linear in the feature space – the space induced by the kernel function. The decision function is simply:

$$g(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{\ell} \alpha_i y_i k(\mathbf{x}, \mathbf{x}^i) + b \right) \tag{3.24}$$

Classic kernels were those used in the SVM literature; see appendix A for a description of kernels:

- Polynomial kernel

$$k(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle_{\mathcal{X}} + q)^p \tag{3.25}$$

- RBF kernel

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2} \right). \tag{3.26}$$

To illustrate the effectiveness of non-linear SVM, we performed another experiment. We generated data with two clusters, one Gaussian cluster with zero mean and one cluster with *ring* distribution with zero mean, see Figure 3.6. Linear classifiers cannot handle this sort of data set. For the experiments, we use a polynomial kernel with $p = 2$ and $q = 0$ with the SVM. The minimum distance classifier can also be *kernelized*, since it can be expressed in terms of inner product. For the Bayesian classifier, we made the assumption of two Gaussian clusters, with identical means but different covariance. This leads to the following decision rules:

1. *Minimum distance to the mean classifier*:

$$\left\{ \mathbf{x} | \frac{2}{\ell} \sum_{i,k=1}^{\ell/2} \langle \mathbf{x}^i, \mathbf{x}^k \rangle^2 - \frac{2}{\ell} \sum_{j,l=1}^{\ell/2} \langle \mathbf{x}^j, \mathbf{x}^l \rangle^2 - \frac{2}{\ell} \sum_{m=1}^{\ell} y_m \langle \mathbf{x}^m, \mathbf{x} \rangle^2 = 0 \right\}. \tag{3.27}$$

2. *The SVM classifier*:

$$\left\{ \mathbf{x} | \sum_{i=1}^{\ell} \alpha_i y_i \langle \mathbf{x}^i, \mathbf{x} \rangle^2 + b = 0 \right\}. \tag{3.28}$$

3. *Quadratic Bayesian classifier*:

$$\left\{ \mathbf{x} | \mathbf{x}^T \left(\Sigma_+^{-1} - \Sigma_-^{-1} \right) \mathbf{x} + \log\left(\frac{\det(\Sigma_+)}{\det(\Sigma_-)} \right) = 0 \right\}. \tag{3.29}$$

The decision functions are plotted in Figure 3.6. The Bayesian classifier is unable to classify the data correctly. In this situation, the Gaussian assumption is not verified and thus the classification is not optimal. The SVM classifier fits the ellipsoidal geometry of the data well, which is not true for the minimum distance classifier. It is important to note that if the SVM performs better than the Bayesian classifier, it is because the Gaussian assumption does not hold and the data are linearly separable in the feature space induced by the polynomial kernel.

This poses the problem of *kernel selection.* Which kernel should we use for the classification of remote-sensing data? In the following, we see that classic kernels already perform well but, by including some *a priori*, adapted kernels could provide better results.

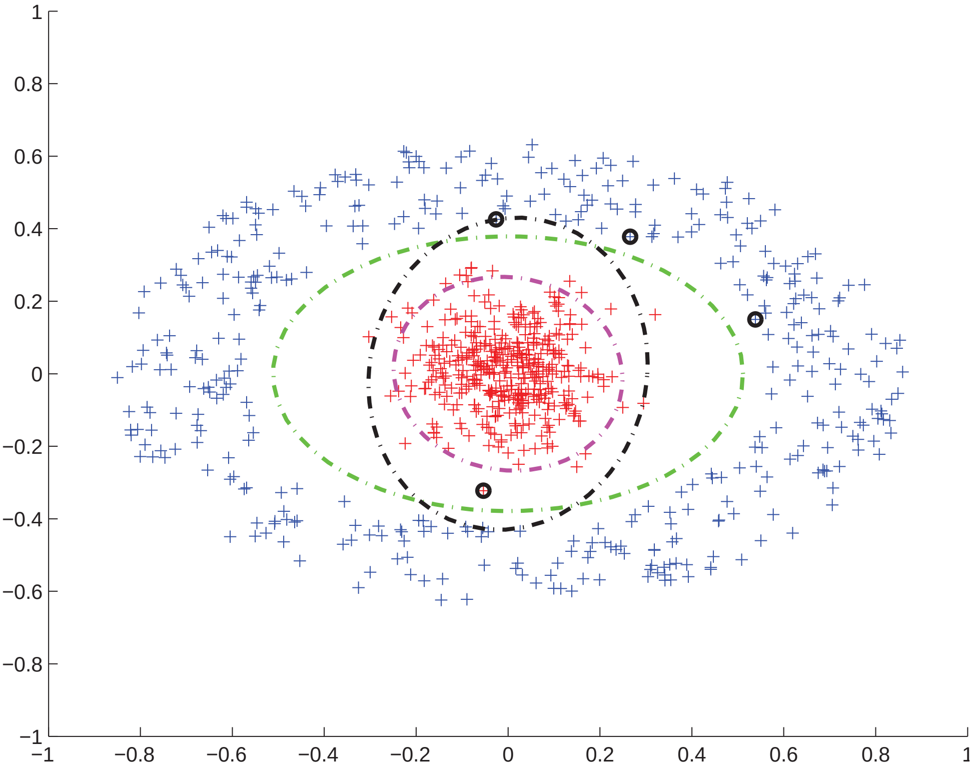


Figure 3.6: *Fictional example: Two non-linearly separable clusters. Black circles: vectors. Black line: decision boundary minimum distance classifier, green line: SVM, magenta line: Bayesian classifier.*

3.3 Multi-class SVM

SVMs are designed to solve binary problems where the class labels can only take two values, *e.g.*, ±1. For a remote-sensing application, several 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 an m-class problem [112].

- **One Versus the Rest:** m binary classifiers are applied on each class against the others. Each sample is assigned to the class with the *maximum* output.
- **Pairwise Classification**: $\frac{m(m-1)}{2}$ 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 sample to that class.

Pairwise classification has proven to be more suitable for large problems [65]. 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.

Others strategies have been proposed within the remote-sensing community, such as hierarchical trees or global training [90, 54]. However, classification accuracies were similar, or worse, and the complexity of the training process is increased. Therefore, we have used the two classic approaches in our experiments.

3.4 Review of application of SVM for remote-sensing data analysis

Before presenting the results of using SVM on real data sets, we will first review the principal articles dealing with the same problem. We focused our attention on the training problem and the dimensionality of the data.

The training of an SVM in a remote-sensing context has been addressed mainly by Foody and co-workers [54, 57, 58]. They first investigated the behavior of SVMs with small training sets [54] and compared the results with other classic classifiers. However, their experiments were carried out using multispectral images where the three most informative feature bands were extracted and used as the input to the classifiers. Hence the suitability of SVM for solving the 'High dimensional space / small training set' problem was not totally analyzed[3]. In [57, 58], particular attention was paid to the design of the training set. Foody has shown that a training set might be well suited for a given classifier, while unsuitable for another classifier [55]. For a statistical classifier, training pixels should be as pure as possible, while with SVMs, training pixels should be at the boundary between classes in the features space [57]. Experiments using SVM classifiers have validated this idea. Pursuing the same idea, Foody has shown that the training set can be reduced when considering several one-class classifiers instead of a multi-class classifier [58].

In this thesis, we assume that the training set is provided with the data, and thus we cannot influence its design. The problem to be considered therefore is how well or badly the classifier works with a reduced training set? Readers especially interested in the design of training set are invited to refer to Foody's work.

In [20], several neural networks were compared to the SVM for the classification of hyperspectral data. The robustness of SVMs was demonstrated and the best results were obtained using a non-linear SVM. Thus the superiority of the SRM over the ERM was confirmed for remote-sensing applications. Melgani and Bruzzone have compared several approaches for the classification of multi-class remote-sensing problems [90]. As we concluded in the previous section, the simpler the approach, the better the results. Still in [90], SVMs were also positively compared to K-nn and RBF classifiers, where the data have higher dimensionality than the data used in [20] (200 bands as against 128 bands). Note that RBF classifiers and SVM were studied in [126, 127], where the authors noted the favorable behavior of the SVM, from both a theoretical and a practical point of view.

However, although these experiments do highlight the favorable behavior of the SVM with high-dimensional data, they used a full training set, and we do not really know what the results would have been if a smaller training set had been used. Furthermore, we do not know which kernel was used. In [20], the polynomial kernel performs better, while in [90] it is the Gaussian kernel. This motivated the experiments below (3.5.1): several kernels were used to classify hyperspectral data with full dimensionality and with a very small training set. Details are given in the next section.

One other point has not been addressed: the selection of the model or the kernel parameters. Cross-validation is often used, which can result in very lengthy processing. Several strategies have been developed, mostly based on an upper error bound [31]. One particularly interesting approach used gradient descent to fit the model. The main problem is that the error bound holds in limit or in probability and sufficient training samples may be needed to obtain a satisfactory estimate of the error bound. Thus in a second experiment 3.5.2 we tested a strategy developed by Chapelle *et al.* [36] for fitting the model when the training set contains very few parameters. This approach was also compared to cross validation.

3.5 Test Kernels

The preceding section's short review discussed existing experiments conducted with SVMs. In our opinion, some aspects of classic SVMs for the classification of remote-sensing data have not been addressed yet. We propose to investigate some of these next. First, training on high-dimensional data with a reduced training set was assessed. We compared three kernels: the polynomial kernel, the Gaussian kernel and the SAM kernel. The last kernel has been used to handle multispectral remote-sensing data characteristics [92]; we will introduce it briefly at the beginning of section 3.5.1. In the second experiment, we proposed using another training approach to fit the kernel parameter based on Chapelle's work [31].

[3] Note that in this configuration, the other classifiers also performed well with limited training sets.

3.5.1 Small training sets

The Spectral Angle Mapper (SAM) was introduced to measure similarity between spectral signatures for identification of spectra [76]. SAM is a scale-invariant metric that has been used in many remote-sensing problems and it has been shown to be robust against variations in spectral energy [76]. This metric α focuses on the angle between two vectors:

$$\alpha(\mathbf{x}^i, \mathbf{x}^j) = \arccos\left(\frac{\langle \mathbf{x}^i, \mathbf{x}^j \rangle}{\|\mathbf{x}^i\| \,.\, \|\mathbf{x}^j\|}\right). \tag{3.30}$$

As mentioned in [76], Euclidean distance is not scale invariant; however, due to atmospheric attenuation or variation in illumination, spectral energy can be different for two samples even if they belong to the same class and distance, suggesting that SAM would be preferable. To introduce the scale-invariance assumption into the kernel, we first have to write the RBF kernel as [112] $k(\mathbf{x}^i, \mathbf{x}^j) = f(d(\mathbf{x}^i, \mathbf{x}^j))$ where d is a metric on $\mathbb{R}^n$ and f is a function on $\mathbb{R}_0^+$. For the Gaussian RBF, $f(t) = \exp(-\gamma t^2)$, $t \in \mathbb{R}_0^+$, and $d(\mathbf{x}^i, \mathbf{x}^j)) = \left\|\mathbf{x}^i - \mathbf{x}^j\right\|$, *i.e.*, the Euclidean distance. Changing the Euclidean distance to the SAM distance leads to the spectral kernel

$$k(\mathbf{x}^i, \mathbf{x}^j) = \exp\left(-\gamma\alpha(\mathbf{x}^i, \mathbf{x}^j)^2\right). \tag{3.31}$$

This kernel was originally introduced by Mercier *et al.* in [92] for the classification of multispectral images. As far as we are aware, no analysis has been carried out using this type of kernel on hyperspectral data.

In the following experiment, kernels are compared for the classification of hyperspectral data with limited training samples [47]. The data used in the experiments are very high resolution hyperspectral data. The image used is the right-hand part of Pavia Center, Italy. It is 492 × 1096 pixels and contains 102 spectral bands. Training and test sets are listed in Table 3.1, see Appendix C for a complete description of the data. Small training sets were randomly extracted from the training set and were composed of 10, 20, 40, 60, 80, and 100 pixels per class respectively. The pairwise multi-class strategy classifiers were trained with each of these training subsets and then evaluated using the entire test set. These experiments were repeated five times (with five independent training subsets) and the mean accuracy values were reported. Three kernels were tested: the polynomial kernel, the Gaussian, and the SAM-based kernel. During each training process, the kernel parameter σ, p and the penalty term C were adjusted to maximize the estimated overall accuracy, which was computed using fivefold cross-validation [112]. The SVM training was computed using the LIBSVM library [24] and the program was modified to include SAM kernel.

Table 3.2 summarizes the results obtained using the polynomial, Gaussian, and SAM RBF kernels. These values were extracted from the *confusion matrix* [109]. SVM generalizes very well: with only 10 training pixels per class over 90% accuracy is achieved by all kernels. It is also clear that the classification accuracy correlates with training set size. But the difference in terms of accuracy is fairly small: for instance, with the Gaussian RBF kernel, the OA obtained with only 10 training pixels per class is only 2.7% lower than the OA obtained with the complete training set. However, the computation time (including optimum parameter selection, training, and classification) takes only about 10 minutes with 10 training pixels, compared to over 12 hours with the full training set.

Regarding the polynomial and the Gaussian RBF kernel, the results are very similar. However, the Gaussian RBF performs better with very small training sets. The use of the SAM kernel gives slightly degraded classification results in terms of the AA, OA, and the Kappa coefficient. One explanation lies in the fact that urban scenes are less sensitive to spectral variations than agricultural areas, with weeds at various stages of development and different layers casting shadows. Another reason is because the SAM kernel does not use the energy of each pixel-spectrum (its norm). Differences between classes are in the shape of the pixel-spectrum (may be regarded as the angle) and the energy of the pixel-spectrum (its norm). Using the polynomial or Gaussian kernels, both kinds of information are used:

$$\begin{aligned}
(\langle \mathbf{x}, \mathbf{y} \rangle + 1)^p &= (\|\mathbf{x}\| \ \|\mathbf{y}\| \ \cos(\theta) + 1)^p \\
\exp\left(-\gamma\|\mathbf{x} - \mathbf{y}\|^2\right) &= \exp\left(-\gamma\left(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\langle \mathbf{x}, \mathbf{y} \rangle\right)\right) \\
&= \exp\left(-\gamma\left(\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\|\mathbf{x}\| \ \|\mathbf{y}\| \ \cos(\theta)\right)\right)
\end{aligned} \tag{3.32}$$

Table 3.1: *Information classes and training/test samples for the right-hand part of* Pavia Center *data set.*

Class		Samples	
No	Name	Train	Test
1	Water	745	65278
2	Trees	785	6508
3	Meadow	797	2900
4	Brick	485	2140
5	Soil	820	6549
6	Asphalt	816	7555
7	Bitumen	808	6479
8	Tile	223	3122
9	Shadow	195	2165
Total		5536	103504

Figure 3.7 shows the classification map obtained for all three kernels with only 10 pixels per class.

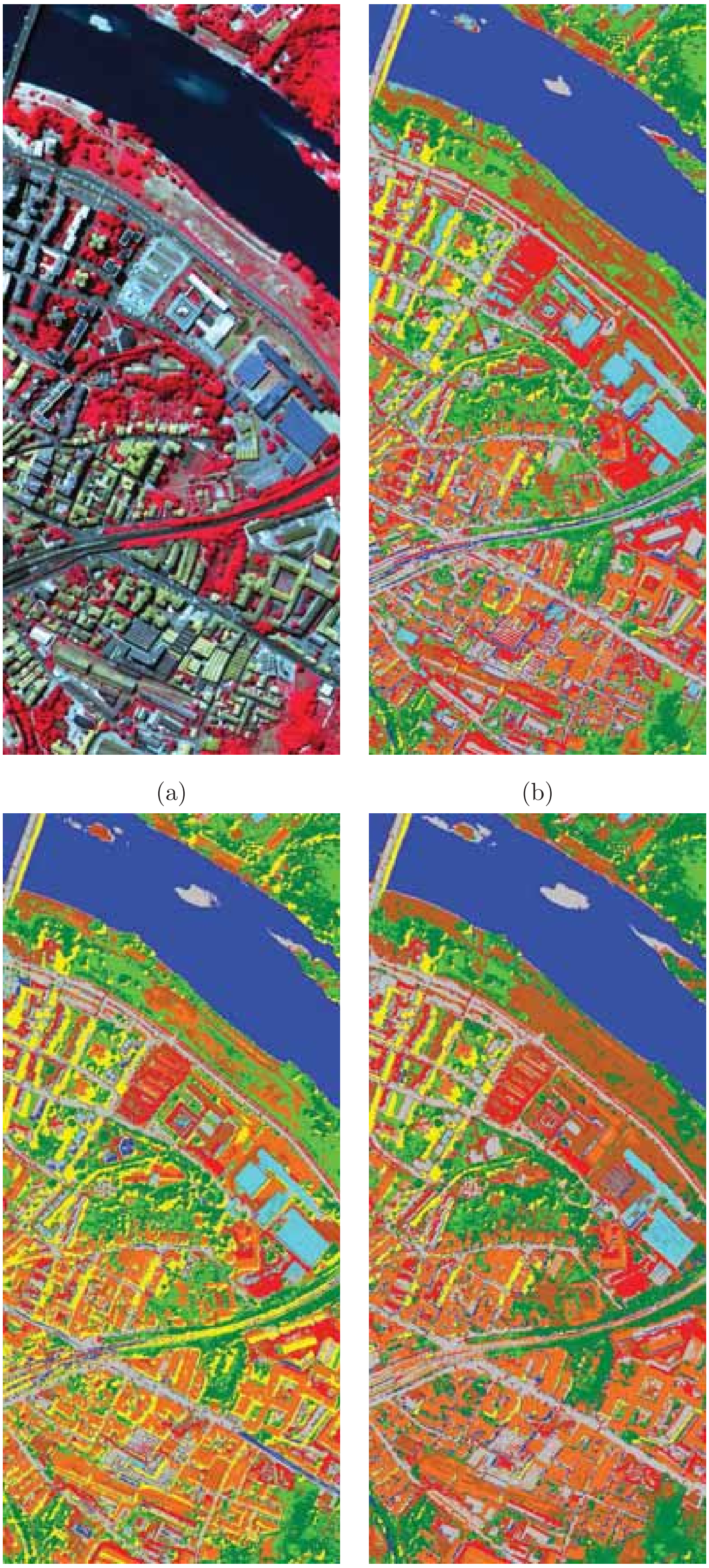
3.5.2 Fitting the parameters

Find the optimum parameters for the SVM is not a straightforward task. The values of the kernel parameters can have a considerable influence on the learning capacity, an example is given in Appendix A. Fortunately, even though their influence is significant, their optimum values are not critical, *i.e.*, there is a range of values for which the SVM performs equally well. Figure 3.8 shows cross-validation results for the Gaussian and polynomial kernels. They both clearly show a plateau of values that gives the same cross-validation accuracies.

One drawback of cross-validation parameter selection is that the SVM needs to be trained several times, *e.g.*, for two parameters where 10 values are tested, 500 training cycles have to be performed with 5-fold cross validation for each binary classifier. Where more than two parameters are considered, it becomes intractable, even for small-scale problems.

One possible approach is to consider some *a-priori*: if we consider that the data are linearly separable in the feature space, we can set C to a high value to impose no training errors, or at least very few. Some pre-processing on the data range could also be done: stretch the data between -1 and 1. This can be useful for the Gaussian kernel because it limits the range of values to be tested for the σ parameters. This strategy was successfully applied in our experiments: we stretched the data between -1 and 1, set $C = 200$, and tested this set of values for σ^2 =$\{0.125,0.25,0.5,1,2,4\}$. We compared the classification accuracies with those obtained with a larger range of values; 21 values were tested for each parameter, see Figure 3.8. No significant differences were found in term of classification accuracy, while the training time was clearly much shorter when fewer parameters were tested.

Another approach has been developed by Chapelle [36, 31] and Keerthi[74]. They consider the case



(a)

(b)

(c)

(d)

Figure 3.7: *(a) Original hyperspectral image, three-channel color composite. (b) Thematic map produced using Gaussian RBF kernel (c) Spectral RBF kernel and (d) polynomial kernel with 10 training pixels per class.*

Table 3.2: *Classification accuracies for several training set size [47].*

Training Set Size		10	20	40	60	80	100	All
Gaussian	OA	93.85	94.51	94.51	94.71	95.36	95.29	96.45
	AA	88.76	91.00	92.66	92.04	93.24	93.39	95.08
	κ	0.90	0.91	0.92	0.91	0.92	0.92	0.94
Poly	OA	92.34	92.77	94.20	94.07	94.29	94.81	96.03
	AA	87.87	88.91	91.74	92.41	92.31	93.35	94.91
	κ	0.87	0.88	0.90	0.90	0.90	0.91	0.93
SAM	OA	93.32	93.87	93.79	94.23	94.40	94.54	95.56
	AA	86.36	88.64	91.26	91.67	91.89	92.61	94.26
	κ	0.89	0.90	0.90	0.90	0.90	0.91	0.93

of a soft-margin non-linear SVM with quadratic penalization (ξ^2 in (3.16))

$$\begin{aligned} \max_{\alpha} g(\alpha) &= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \tilde{k}(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} & \quad 0 \leq \alpha_i \\ & \quad \sum_{i=1}^{\ell} \alpha_i y_i = 0 \end{aligned} \tag{3.33}$$

where $\tilde{k}(\mathbf{x}^i, \mathbf{x}^j) = k(\mathbf{x}^i, \mathbf{x}^j) + \frac{\delta_{ij}}{C}$, for details see [31] page 65. An error bound related to (3.9) is derived, the *radius margin* bound, which is a function of the kernel parameters. The minimization of this function yields the optimum parameters for the SVM. If we denote the parameter vector by $\mathbf{p}$ and the upper error bound by T, we have

$$T(\mathbf{p}) := \|\mathbf{w}\|^2 \mathcal{R}^2. \tag{3.34}$$

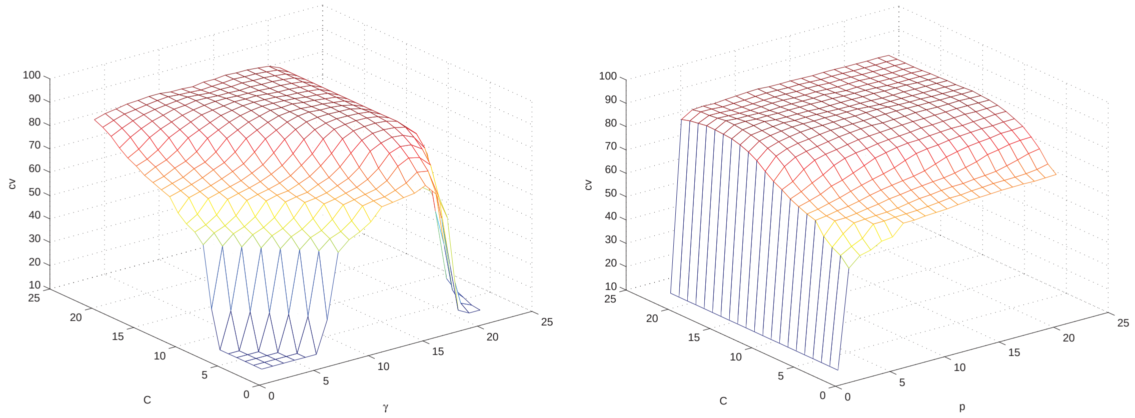
where

$$\begin{aligned} \|\mathbf{w}\|^2 &= 2\sum_{i=1}^{\ell} \alpha_i - \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j \tilde{k}(\mathbf{x}^i, \mathbf{x}^j) \\ \mathcal{R}^2 &= \sum_{i=1}^{\ell} \beta_i \tilde{k}(\mathbf{x}^i, \mathbf{x}^i) - \sum_{i,j=1}^{\ell} \beta_i \beta_j \tilde{k}(\mathbf{x}^i, \mathbf{x}^j) \end{aligned} \tag{3.35}$$

In the case of the Gaussian kernel, $\mathbf{p} = [C, \sigma^2]$ and $\tilde{k}(\mathbf{x}^i, \mathbf{x}^i) = 1 + 1/C$.

The kernel parameters $\mathbf{p}$ have to minimize T. The computation of the gradient of T requires the gradients of $\|\mathbf{w}\|^2$ and $\mathcal{R}^2$. These depend on optimum α_i, which is also dependent on $\mathbf{p}$. Chapelle has proven that since $\|\mathbf{w}\|^2$ and $\mathcal{R}^2$ are computed via an optimization problem, the gradient of α_i does not enter into the computation of their gradients [35]:

$$\mathbf{grad}\,(T(\mathbf{p})) = \begin{bmatrix} \frac{\partial T}{\partial C} \\ \frac{\partial T}{\partial \sigma^2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \|\mathbf{w}\|^2}{\partial C} \mathcal{R}^2 + \frac{\partial \mathcal{R}^2}{\partial C} \|\mathbf{w}\|^2 \\ \frac{\partial \|\mathbf{w}\|^2}{\partial \sigma^2} \mathcal{R}^2 + \frac{\partial \mathcal{R}^2}{\partial \sigma^2} \|\mathbf{w}\|^2 \end{bmatrix}. \tag{3.36}$$



(a)

(b)

Figure 3.8: *Cross validation grid for (a) Gaussian kernel and (b) polynomial kernel. C is tuned for both kernel, σ for the Gaussian kernel and p for the polynomial kernel (q=1).*

where

$$\begin{aligned}
\frac{\partial\|\mathbf{w}\|^2}{\partial C} &= \sum_{i=1}^{\ell} \frac{\alpha_i}{C^2} \\
\frac{\partial\|\mathbf{w}\|^2}{\partial \sigma^2} &= -\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j \frac{\|\mathbf{x}^i-\mathbf{x}^j\|^2}{2\sigma^4}\tilde{k}(\mathbf{x}^i,\mathbf{x}^j) \\
\frac{\partial\mathcal{R}^2}{\partial C} &= -\sum_{i=1}^{\ell} \frac{\beta_i(1-\beta_i)}{C^2} \\
\frac{\partial\mathcal{R}^2}{\partial \sigma^2} &= -\sum_{i,j=1}^{\ell} \beta_i\beta_j \frac{\|\mathbf{x}^i-\mathbf{x}^j\|^2}{2\sigma^4}\tilde{k}(\mathbf{x}^i,\mathbf{x}^j)
\end{aligned} \tag{3.37}$$

Details of the derivatives can be found in [36, 74, 31]. Using a conventional gradient descent algorithm, it is possible to find kernel parameters with fewer SVM training cycles than with cross-validation training. Moreover, it is possible to evaluate many kernel parameters, for example with one σ^2 per dimension:

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\boldsymbol{\sigma}^2}\right) \tag{3.38}$$

$$= \exp\left(-\sum_{m=1}^{n}\frac{(x_m-y_m)^2}{2\sigma_m^2}\right) \tag{3.39}$$

To evaluate the benefit of this type of bound, we performed two experiments. First, we compared the classification accuracies for 4 different training strategies, all with a Gaussian kernel:

1. Cross validation of a L_1 SVM, ξ_i,
2. Cross validation of a L_2 SVM, ξ_i^2,
3. Gradient descent with two parameters $\mathbf{p} = [C, \sigma^2]$,
4. Gradient descent with $n+1$ parameters $\mathbf{p} = [C, \sigma_1^2, \ldots, \sigma_n^2]$.

Then the influence of the size of the training set was investigated. The results of the first experiment are reported in Table 3.4. We used the One Versus All strategy for solving the multi-class problem. To better see the influence on each individual training process, we did not apply the *winner takes all* rule at the end of each individual classification, hence the results can be regarded as 9 binary classification problems. To assess the effectiveness of the gradient approach, we compared the number of optimizations during the training rather than training processing time, because of the differences in the algorithms (the standard SVM program is highly optimized). The data used in the experiments are very high-resolution hyperspectral data. The image used is 'University Area', Italy. It is 340 × 610 pixels and contains 103 spectral bands. Training and test sets are listed in Table 3.3, see appendix C for a complete description of the data.

Regarding the classification accuracies in Table 3.4, the best results were achieved using cross-validation training. Both L_1 and L_2 versions gave similar results. However, when considering the number of optimizations performed, the best results were achieved with the gradient strategies. In particular, when only two parameters need to be set, the number of optimizations for the gradient is half the number for cross- validation. Surprisingly, the use of multivalued σ does not provide better results. However, it would not be possible to fit parameters for this type of kernel using cross-validation.

To investigate the influence of training set size, we performed experiments on the same data set, but with the training set reduced to 10 pixels per class, randomly selected. For each class, the experiment was repeated 20 times, and the mean and standard deviation values are plotted in Figure 3.9. As with the entire training set, the different training strategies lead to roughly comparable results. The mean classification accuracies are 90.34%, 90.65%, and 91.12% for the multivalued gradient, the classic gradient, and the cross validation (with L_1 SVM) respectively. Since L_1 and L_2 SVM performed similarly, we have only reported results for the most standard SVM. It is interesting to note that with the multivalued gradient, there are 104 parameters to tune $(C, \sigma_1, \ldots, \sigma_{103})$, and only 90 samples for the training. There is a grave risk of over-fitting! However, in terms of classification accuracy, the SVM still performs well. Considering training time, with such a small training set, optimization takes less than 1 second with any of the strategies.

To conclude, the gradient-based training produced remarkable results in terms of classification accuracies compare to cross-validation. It made it possible to fit many kernel parameters, with no over-fitting. Complex kernels are now conceivable in practical situations. However, as can be seen from the two previous experiments, a more specific kernel does not always yield more accurate results, and the kernel still needs to be chosen carefully.

Table 3.3: *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

Table 3.4: *Classification accuracies in percenatage and number of optimizations for the* University Area *data set, for several training strategies.*

Training Method	**Cross-validation L_1**		**Cross-validation L_2**		**Gradient**		**Gradient Multi.**	
Class	%	No. of Optim.	%	No. of Optim.	%	No. of Optim.	%	No. of Optim.
1	96.12	30	96.13	30	94.68	19	94.88	56
2	87.23	30	86.49	30	79.05	16	78.31	35
3	97.80	30	97.82	30	97.08	20	97.33	22
4	96.85	30	96.44	30	94.56	14	94.40	9
5	99.74	30	99.87	30	99.21	2	99.21	2
6	87.24	30	87.41	30	87.72	18	85.39	28
7	98.75	30	98.80	30	98.76	17	98.54	20
8	96.54	30	96.41	30	96.68	19	96.81	30
9	99.96	30	99.96	30	99.93	11	99.99	12
Mean value	95.58	30	95.48	30	94.19	15	93.87	24

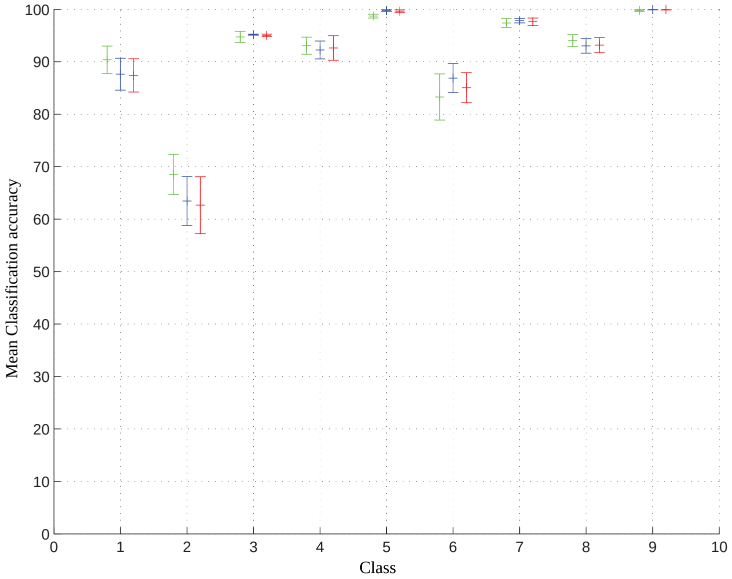


Figure 3.9: *Classification accuracy for each class with a small training set: 10 samples per class. Red lines are the results for the multivalued gradient, blue lines are for the single gradient, and the green lines are for cross-validation. The bars indicate the standard deviations over the 20 experiments. Horizontal axis is the corresponding class and the vertical axis is the overall accuracies.*

3.6 Comparison to standard classifier

In this section, we compare SVM classification to others standard classifiers: a Maximum Likelihood (ML) classifier with a Gaussian assumption and a back-propagation neural network (NN) classifier with one hidden layer. DBFE was applied for the maximum likelihood, while for the neural network PCA and independent component analysis (ICA) were used [67]. Due to the high dimensionality of the data, without feature reduction the neural network training was intractable. For the statistical classification and feature extraction, the MultiSpec program was used [80], and the neural network was programmed using Matlab via the Neural Network toolbox [97].

Experiments on hyperspectral data are reported, since the dimensionality of such data is probably the main difficulty for existing classifiers. The problem of panchromatic data is more related to the extraction of informative spatial features, and this issue will be addressed in the next chapter.

Experiments were performed on two different data sets. A complete description of the data set is given in Appendix C. In the following discussion, readers are invited to refer to the appendix for the description of the data.

Classification accuracy was assessed using overall accuracy (OA), which is the number of correctly-classified samples divided by the number of test samples, and the average accuracy (AA), which represents the average of class classification accuracy. For all experiments, an SVM with a Gaussian kernel was used and the parameters were fitted using cross-validation. From previous considerations 3.5.2, the parameter C was set at 200 and σ^2 was chosen from $\{0.5, 1, 2, 4\}$ by five-fold cross-validation.

Two sets of moderate hyperspectral data were analyzed. The first one contains 103 bands and the second one 102 bands. The spatial resolution is 1.3 meter per pixel. Both data sets are described in appendix C, in the ROSIS data section. Testing and training sets are also detailed.

3.6.1 University Area

Maximum Likelihood classifier based on the Gaussian assumption was applied on the entire data set and on the features extracted using DBFE based on 99% of the variance. The neural classifier was applied on the first three principal components, corresponding to 95% of the cumulative variance. From these three principal components, three independent components were extracted and classified by the neural classifier. Finally the SVM with a Gaussian kernel was used to classify the entire data set. Results are reported in Table 3.5. For the first experiment, two multi-class strategies were used: the One vs. All and the One vs. One approach.

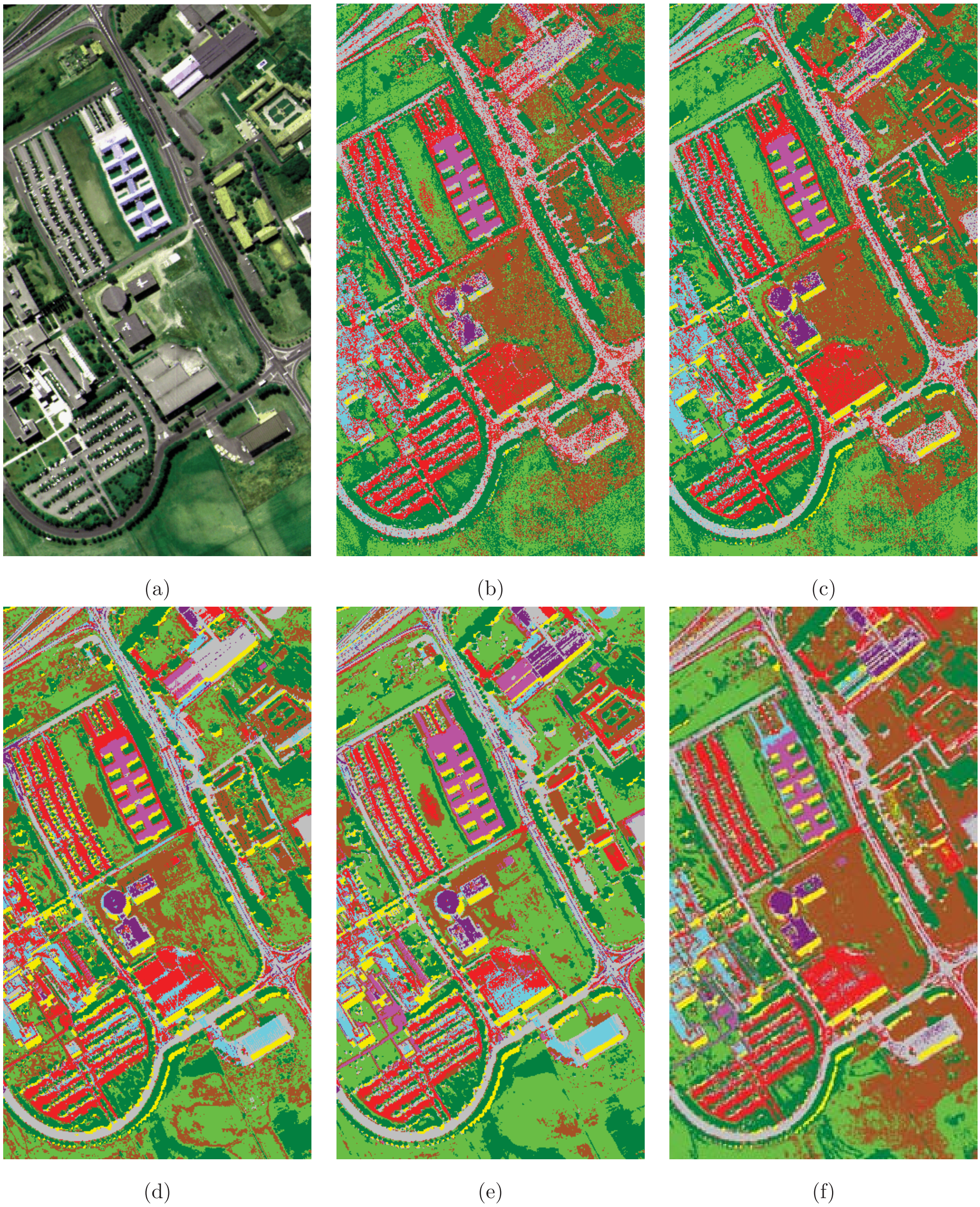
First of all, regarding the results from SVM classifiers, no significant differences between multiclass classification strategies are found. This confirms the conclusions drawn in [65]. For the next experiment, we shall only be referring to results obtained using the One vs. All approach.

Clearly, the SVM classifier outperforms all the other classifier in terms of classification accuracy. The Maximum Likelihood needs DBFE to perform correctly and achieves an overall accuracy of 77.9%. The neural network performs similarly with the features extracted using PCA or ICA; note that adding more features does not improve the results significantly, and adding too many features produces a reduction in the OA. The overall accuracy for the PCA+NN is 66.7% against 71.7% using the ICA+NN, while the average accuracy is 77.6% for the PCA+NN against 76.7% for the ICA+NN. The SVM classifier achieves an OA of 80.1% and AA of 88.3%.

Classification maps are shown in Figure 3.10. Visual inspection of the thematic map reveals that the results provided by the ML are subject to severe noise. The neural network yields less noisy results. The thematic map provided by the SVM is less noisy than the others.

3.6.2 Pavia Center

Maximum likelihood was applied on the entire data set and on the features extracted using DBFE based on 99% of the variance. As with the University Area, three principal and independent components were classified with the neural network. The SVM with a Gaussian kernel was used to classify the entire



(a) (b) (c)

(d) (e) (f)

Figure 3.10: *(a) False color original image of* University Area, *(b) Classification map using maximum likelihood, (c) Classification map using DBFE + ML, (d) Classification map using PCA+NN, (e) Classification map using ICA+NN, and (f) Classification map using SVM.*

Table 3.5: *Classification accuracies in percentage for the* University Area *for several classifiers.*

Classifier	ML	ML	NN	NN	SVM 1 vs. All	SVM 1 vs. 1
FE	-	DBFE	PCA	ICA	-	-
Features	103	31	3	3	103	103
1	66.8	72.9	71.4	70.1	80.6	83.7
2	61.8	71.4	56.7	73.4	68.5	70.3
3	47.8	65.5	53.1	61.7	73.1	70.3
4	97.7	97.6	98.0	96.1	97.5	97.8
5	100.0	100.0	99.6	99.5	99.5	99.4
6	82.1	91.2	57.5	35.8	94.8	92.3
7	51.3	82.0	80.9	72.5	91.5	91.6
8	79.1	82.2	81.2	82.0	91.9	92.6
9	26.5	90.8	99.7	99.4	97.0	96.6
OA	68.5	77.9	66.7	71.7	80.1	81.0
AA	68.1	83.7	77.6	76.7	88.3	88.3

data set. Results are reported in Table 3.6. In accordance with our previous conclusions, we have only reported the results for the One vs. All multi-class strategy.

From the table, the SVM performs better than the other classifiers in terms of classification accuracy. The overall accuracy is 98.1% and the average accuracy is 95.8%. The maximum likelihood classifier achieved an overall test accuracy of 93.8% with the entire data set and 94.5% with the extracted features. The neural classifier performs similarly with the principal and independent components.

Classification maps are shown in Figure 3.11. As with the University Area data set, the ML classifier yields a noisy thematic map. The neural and SVM classifiers yield more homogeneous thematic maps.

3.7 Beyond the SVM

In the first part of this chapter, we reviewed the basics of SVM. Based on statistical learning theory, SVM implements a strategy to *learn from the data* and to prevent over-fitting. Based on a geometric approach rather than on probabilistic modeling, SVMs possess several properties that make them well suited for the analysis of remote-sensing data. They are not impaired by the dimensionality of the data and they have good generalization ability, even in the situation of a reduced training set. In Section 3.5, experiments have been carried out to first demonstrate the effectiveness of the SVM, and then propose an alternative model selection for the classification of remote-sensing data. Experiments on real hyperspectral data sets confirm the superiority of SRM (SVM) over ERM (neural network) and conventional statistical classifiers.

Many theoretical advances have been made, on the optimization problem [16] and on the problem formulation [32]. Semi-supervised learning has been investigated successfully and represents a very promising methodology for remote-sensing applications: unlabelled samples are used to fit the separating hyperplane [34] better. Direct application of such techniques to remote-sensing data can be found in [37, 19]. Several strategies for semi-supervised classification of remote-sensing data, in particular using SVM, are detailed in [89]. Very promising results are being achieved, in particular for knowledge transfer.

All the methods presented work with only the spectral information. But as we have seen in the first part of the thesis, the spatial information is highly informative. Not much research has been done on

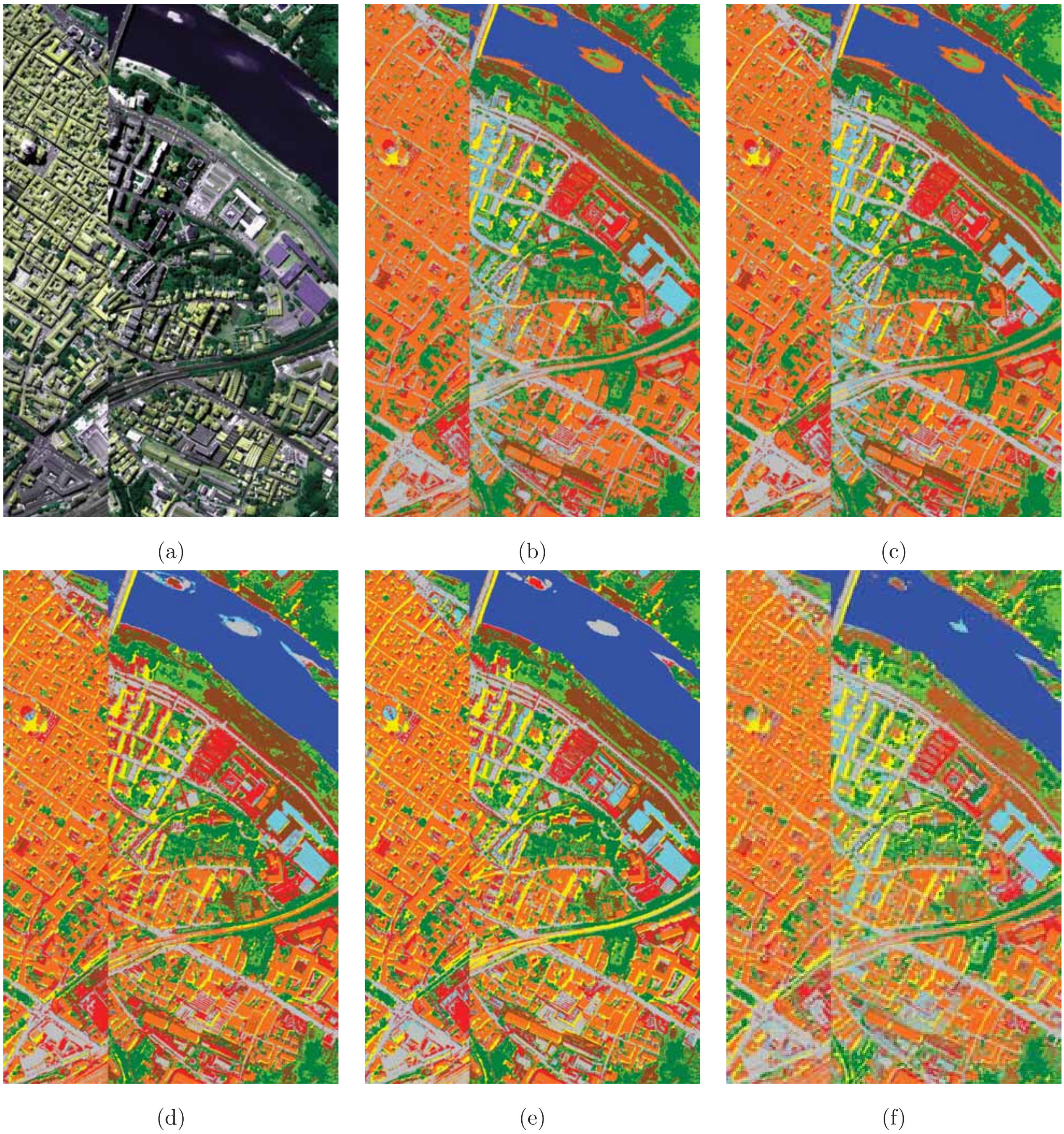


Figure 3.11: *(a) False color original image of* Pavia Center*, (b) Classification map using maximum likelihood, (c) Classification map using DBFE + ML, (d) Classification map using PCA+NN, (e) Classification map using ICA+NN, and (f) Classification map using SVM.*

Table 3.6: *Classification accuracies in percentage for the* Pavia Center *for several classifiers.*

Classifier	ML	ML	NN	NN	SVM
FE	-	DBFE	PCA	ICA	-
Features	102	28	3	3	102
1	92.0	91.5	95.7	97.1	99.1
2	91.3	92.0	81.8	90.2	90.8
3	96.6	97.7	92.9	84.1	97.4
4	81.8	86.9	72.7	80.6	87.5
5	95.2	95.6	91.0	85.1	94.6
6	85.9	94.4	93.6	94.1	96.4
7	95.6	96.4	80.4	84.4	96.5
8	99.4	99.3	98.3	98.4	99.5
9	79.6	92.3	99.6	99.3	100.0
OA	93.8	94.5	94.5	95.5	98.1
AA	90.8	94.0	89.6	90.4	95.8

inlcuding such information in SVM formulation in contrast to statistical approaches. In the next chapter, several approaches dealing with the use of both type of information will be presented. In particular, a novel approach to knowledge-transfer is proposed, based on the spatial information rather than the spectral information, and an SVM-based strategy is proposed to include both spatial and spectral information throughout the classification process.

Chapter 4

Advanced SVM for remote-sensing data classification

Abstract

Advanced SVMs for the classification of remote sensing data are investigated in this chapter. First we study the transferability of the decision function for the classification of remote-sensing images from different locations acquired with the same sensor. Rather than using the spectral information as in a traditional knowledge transfer system, we proposed using morphological information to construct the SVM's decision function. Experimental results show the approach appropriate for urban area remote-sensing data classification. Then, joint classification using both spatial and spectral information is addressed in the second the part of this chapter. Based on the adaptive neighborhood proposed in Chapter 2, both types of information are merge using kernel formulation and classification is performed using the SVM. Comparison is made of the proposed approach with the Extended Morphological Profile in the experimental section for both hyper-spectral and panchromatic data.

Contents

4.1 Introduction

In Chapter 3 we discussed the suitability of SVM to analyze remote-sensing data. A great deal of work in the remote-sensing community concerns the standard uses of SVM: for the classification of certain types of data, spectra identification or regression, event detection, and so on. The theoretical evolution of the SVM into semi-supervised learning theory [34] has been investigated by Bruzzone and co-workers [19, 37, 89]. This was applied to the knowledge transfer problem: using unlabeled samples from the new data set, the semi-supervised SVM modifies the position of the separating hyperplane to fit the new data set better. The feature used for classification was the original spectral information. In the next section, we propose solving the problem in a different way: we suggest using features that are mainly invariant and hence are different from spectral features. For urban area data, we propose using spatial features. Using invariant features, no additional training step are needed and good classification accuracies are achieved [50].

In section 4.3, we address the problem of using spatial information for the classification of remote-sensing data. This problem has been studied using a statistical approach with the Markov Random Field (MRF) theory [86, 68, 107]. The contextual inter-pixel dependency definition from MRF was extended to SVM in [21] and [17] where spatial information was modeled as the gray-level distribution in a fixed-size window around a given pixel. The results confirm the usefulness of such modeling, but it suffers from the same problem as MRF: border effects and spatial resolution. Following on from this work, we propose to use the adaptive neighborhood system defined in Chapter 2 to solve both these problems. Experimental results confirm the benefits of such an approach.

4.2 Transferability of the hyperplane

For some applications, such as emergency response or Earth survey, the speed and accuracy of the algorithm are a critical issue. The main shortcoming of conventional methods is the need for a training step for each new data set, involving potentially tedious manual labeling. Recent works have discussed the possibility of *knowledge transfer* for classification algorithms. In [105, 106], the authors exploited an existing classifier to construct a new classifier for a novel problem. This task, very difficult because of the variation of the class characteristics in the spectral domain, is tackled by using a Binary Hierarchical Classifier where an *update step* is added when using a new data set. Another similar approach using transductive SVMs is found in [19].

For the analysis of urban areas with VHR data, the features used for classification are extracted from the structures within the image. In order to apply knowledge transfer for the classification of this type of data, it is necessary to extract features that *do not change* over time or space. For man-made constructions, shape, size, texture, or orientation are possible features that ought to be almost invariant.

Using the *Morphological Profile* (MP) and its derivative (DMP) [8], information about the shape, size, and local contrast of the structures present in the image can be extracted. We have proposed using the MP as an invariant features vector for the classification of VHR data for dense urban areas [50]. For classification, a support vector machines (SVM) classifier with Gaussian kernel has been used. This classifier has exhibited good performance for urban data analysis [51].

Even with the same remote sensor, differences in illumination result in images with different radiometric information. This point is also discussed, and several histogram-based algorithms are used and their influence on the knowledge transfer is addressed.

4.2.1 Morphological Profiles as space-invariant feature vectors

As detailed in Chapter 2, MPs contains local spatial information. To achieve a good estimate of the spatial information, the MP has to be correctly constructed. Hence the shape and size of the SE have to be properly chosen. Traditionally, the *disk* is used. It has the property of being isotropic, *i.e.*, the property of being independent of direction. The number of openings/closings and the step size of the

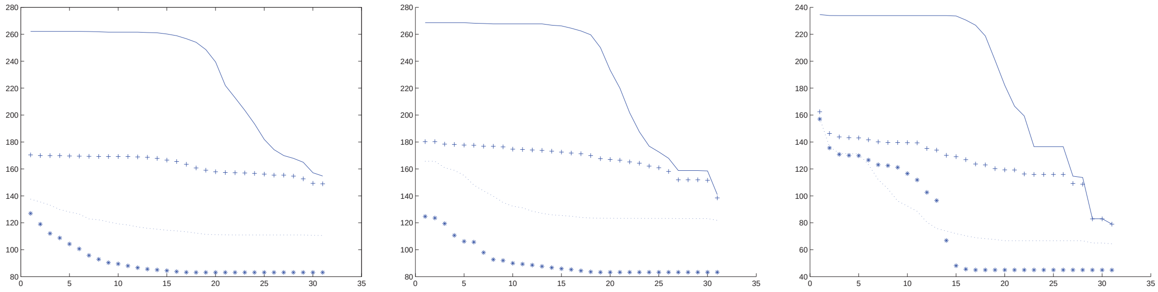


Image 1 Image 2 Image 3

Figure 4.1: *Mean Morphological Profile for panchromatic images. The horizontal axis represent the morphological profile component MP_i and the vertical axis represent the value of $MP_i(\mathbf{x})$ in gray level. ’$+$‘ correspond to the class* road, *’$*$‘ correspond to the class* shadow, *’$-$‘ correspond to the class* building, *and ’.‘ correspond to the class* open area.

SE have to be chosen to cover all the structure in the image. In our applications, we chose the following parameters:

- SE: disk,
- Initial size: R=2 pixels,
- Number of openings/closings: 15,
- Step: 2.

This results in an MP of size 31 for each pixel. This seems to be sufficiently accurate for VHR data.

Then we estimated a *typical* profile for each class. The MP has been computed for our test images. Using some labeled pixels we can have several profiles. A typical profile is estimated by taking the mean profile of all the referenced profiles. In view of the high number of samples, this seems to be a good estimate. We have tried other statistical value estimates, like *median*, but the mean value gave good results. In this thesis, we are only going to report the results obtain using the mean profile.

Figure 4.1 shows the mean MP for three images and for four classes, namely: road, building, shadow, and open area. Images 1 and 2 were extracted from the same panchromatic image, for two different locations. Image 3 is extracted from another panchromatic image, acquired using the same sensor. The data sets are described briefly in Section 4.2.4 and a complete description can be found in C.

4.2.2 Scaling

Prior to construction of the MP, three different scalings were used, to analyze their influence on the transferability. Considering an image I, $I(x)$ a pixel, I_{min} / I_{max} the lowest / highest values, respectively of I, μ_I and σ_I^2 the mean and variance of I, and H_I the histogram of I, the scaling algorithms were:

1. *Histogram stretching*: The pixel values were stretched into $[0, 1]$,

$$I'(x) = \frac{I(x) - I_{min}}{I_{max} - I_{min}}. \tag{4.1}$$

2. *Standardization of the histogram*: remove the mean and fix unitary variance,

$$I''(x) = \frac{I(x) - \mu_I}{\sigma_I}. \tag{4.2}$$

3. *Histogram equalization*: linearization of the histogram:

$$I'''(x) = I(x) \int_0^x H_I(z)dz. \tag{4.3}$$

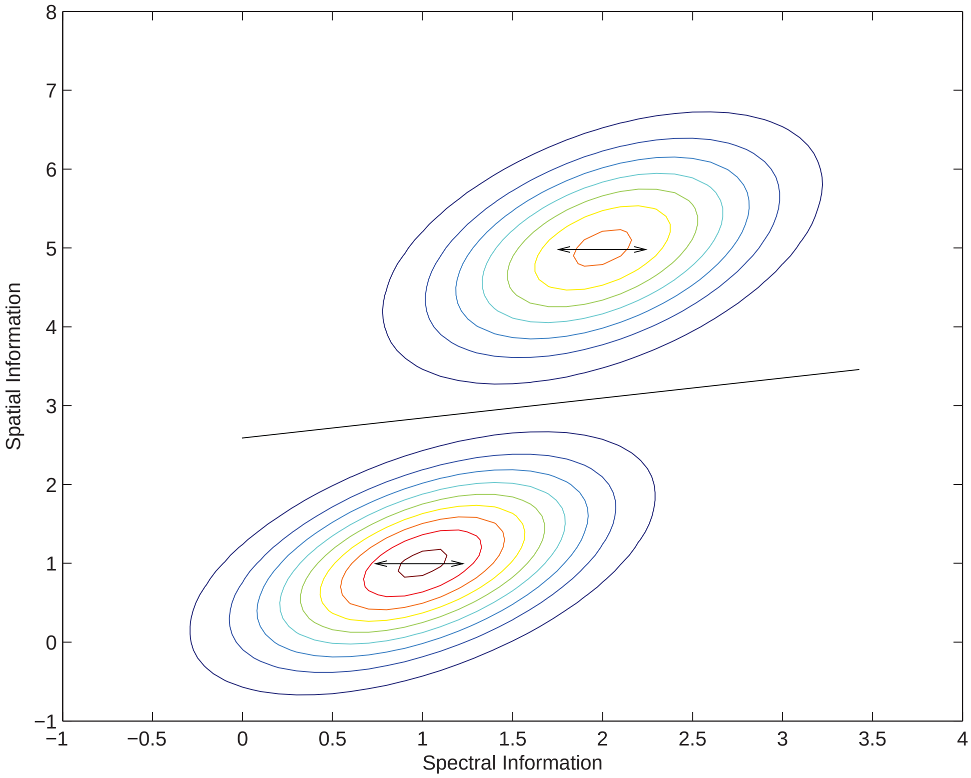


Figure 4.2: *Spatial knowledge transfer principle. For the two clusters, some variation in the spectral information may occur for different images. Owing to some invariant spatial information, the optimal separating hyperplane is still able to classify the clusters correctly.*

4.2.3 Knowledge transfer

A feature vector of size 31 is associated with each pixel. Every pixel now lies in a vector space of dimension 31. The localization of pixels within the space is defined by both its gray level and spatial information. As presented in the introduction, the pixel spectral information can change between different images. But the spatial characteristics of the class should be invariant: a building is still bigger than a road, or a road is always longer than an open area. The localization of a pixel in the vector space for a same class but for different images should be the same since most of the features are invariant. Thus the separating hyperplane found using these features can be used for discrimination of the same classes for different images. This is illustrated in Figure 4.2. Even if some spectral variation occurs, the decision boundary remains the same.

In the next experiment, we implemented the following strategy, empirical but nonetheless effective. For a given data set, a hyperplane that separates the data in the feature space was found using a classic SVM training process. This decision function was then used to classify data from another data set without any training process.

4.2.4 Experiments

Experiments were conducted on various panchromatic images. For each image, the MP is computed and classification is performed using the SVM with a Gaussian SVM and the parameters are tuned using by a 5-fold cross-validation. Two series of experiments were conducted, with and without scaling, respectively. Results are presented in the next two sections.

The data set consists of three panchromatic images extracted from simulated PLEIADES images provided by CNES (satellite to be launched in 2008). The spatial resolution is 0.75 meter per pixel. All

Table 4.1: *Information classes and training/test samples for the three images.*

	Image 1		Image 2		Image 3	
	train	test	train	test	train	test
Road	780	2450	393	905	864	1665
Building	845	2293	355	1005	172	1327
Shadow	798	2588	518	1104	136	446
Open Area	1738	3886	96	583	343	1776
Total	4161	11217	1362	3597	1515	5212

Table 4.2: *Classification accuracy for Image 1.*

	Grayscale			MP		
Training set	Orig.	Im. 2	Im. 3	Orig.	Im. 2	Im. 3
Class 1	85.63	82.67	34.73	94.86	78.13	30.77
Class 2	91.71	89.43	79.50	91.41	22.26	79.55
Class 3	93.00	85.30	76.92	88.25	76.92	76.92
Class 4	86.43	77.74	55.84	97.39	65.35	65.52
Average	89.20	83.79	62.50	92.98	60.66	63.03

images are urban areas. Uncorrelated training and test set were built for each image, see Table 4.1. A complete description of the data is given in the Appendix C. Image 1 corresponds to Toulouse 1, Image 2 corresponds to Toulouse 2 and Image 3 corresponds to Perpignan. For simplicity, we shall refer to these henceforth as Image 1, 2 and 3.

4.2.4.1 MP versus grayscale information

In the first experiments, we compared the *knowledge transfer* with and without MP. The optimal separating hyperplane was constructed using the original gray values and features vector induced by the MP. Results are reported in Tables 4.2, 4.3 and 4.4. The row 'Training set' indicates with which hyperplane the data were classified, *e.g.*, 'Orig.' indicates classification was done using the hyperplane found with the original training set, while 'Im. 2' indicates classification was done using the hyperplane found with the training set from the second image.

From the Tables 4.2, 4.3 and 4.4, we can see that for two images extracted from the same data set (Im. 1 and 2), the grayscale information leads to better classification results. Moreover, the results are not that different, *e.g.* 88.83% → 87.75% in Table 4.3, whereas they are clearly worse when the MP is used.

When considering images extracted from two different data sets (Im. 1 and 3, Im. 2 and 3), the results are better with the MP. In these cases, the radiometric information has changed and the spatial information included in the MP contributes to the knowledge transfer.

4.2.4.2 Influence of scaling

As explained in Section 4.2.2, different histogram-based transformations were tested, prior to construction of the MP. For the sake of clarity, we only report results of knowledge transfer for images extracted from two different data sets (as described in the previous section, radiometric information is

Table 4.3: *Classification accuracy for Image 2.*

	Grayscale			MP		
Training set	Orig.	Im. 1	Im. 3	Orig.	Im. 1	Im. 3
Class 1	79.32	81.04	38.92	90.18	77.98	25.16
Class 2	96.50	88.76	71.06	83.81	70.89	72.06
Class 3	87.76	95.07	69.30	93.80	69.30	69.30
Class 4	88.54	86.40	69.11	93.63	83.79	83.79
Average	88.83	87.75	54.59	90.33	75.49	62.57

Table 4.4: *Classification accuracy for Image 3.*

	Grayscale			MP		
Training set	Orig.	Im. 1	Im. 2	Orig.	Im. 1	Im. 2
Class 1	71.04	58.32	55.45	79.21	67.72	68.05
Class 2	78.41	81.63	56.98	81.58	61.11	25.46
Class 3	91.65	71.66	71.52	94.95	91.44	91.44
Class 4	85.74	54.40	55.66	84.63	64.31	65.95
Average	81.71	66.65	59.90	85.09	71.15	64.98

sufficient for images extracted from a single data set). Results are presented in Tables 4.5, 4.6 and 4.7. $Im.\ a \leftarrow Im.\ b$ means that image a is classified by the optimal hyperplane found with the image b training set. Results in brackets are the classification accuracies obtained with the original training set.

From the tables, all the radiometric corrections led to improved classification accuracies, linear scaling and standardization providing the best results, while equalization seems to perform worse. Equalization stretches the data artificially, according to the cumulative histogram. But the cumulative histogram is too image-dependent, and hence may not be appropriate for knowledge transfer.

The best results were obtained with linear scaling: when image 3 is classified with the optimal hyperplane found with image 1 (2), the final classification accuracy is 86.47% (74.74%) as against 71.15% (64.98%) without scaling.

It is interesting to note that the best knowledge transfer occurs with the images that have the largest training sets. The SVM training algorithm found the samples that support the separating hyperplane, and consequently, with a larger training set, more discriminative samples can be extracted. An algorithm has been proposed by Bruzzone *et al.* [19] based on this principle. An optimal hyperplane is found and is updated by adding/removing a number of support vectors found using unlabeled samples.

4.2.4.3 Discussion

Transferability of spatial features for the classification of urban area has been discussed. Morphological processing was used to extract invariant features and support vector machines were used to classify the data. For two images extracted from the same data set, radiometric information performs well, leading to good classification performance. However, the classes were defined at a coarse level: *building*, *road*... If finer definition is desired, spatial definition should contribute to knowledge transfer. For two images extracted from two different data sets, MP with linear scaling of the data gave promising results. More advanced SVM algorithms should help classification. In particular, semi-supervised SVM needs to be

Table 4.5: *Classification accuracy with linear-scaled data.*

	Im. 3←Im. 1	Im. 3←Im. 2	Im. 1←Im. 3
Class 1	80.14 (80.23)	79.98 (80.23)	65.49 (94.49)
Class 2	87.20 (85.25)	88.16 (85.25)	75.92 (96.38)
Class 3	91.44 (96.33)	76.88 (96.33)	81.97 (93.72)
Class 4	87.13 (81.19)	53.94 (81.19)	59.71 (88.85)
Average	86.47 (85.75)	74.74 (85.75)	70.77 (93.36)

Table 4.6: *Classification accuracy with standardized data.*

	Im. 3←Im. 1	Im. 3←Im. 2	Im. 1←Im. 3
Class 1	76.38 (76.17)	68.91 (76.17)	60.88 (92.99)
Class 2	91.15 (82.73)	83.78 (82.73)	80.89 (95.35)
Class 3	81.81 (66.72)	69.45 (86.72)	78.87 (93.93)
Class 4	74.00 (81.60)	62.20 (81.60)	64.94 (86.13)
Average	80.83 (76.80)	71.08 (76.80)	71.40 (92.1)

Table 4.7: *Classification accuracy with equalized data.*

	Im. 3←Im. 1	Im. 3←Im. 2	Im. 1←Im. 3
Class 1	81.61 (80.33)	63.54 (80.33)	35.17 (92.95)
Class 2	87.77 (85.26)	63.16 (85.26)	79.10 (95.23)
Class 3	86.78 (95.58)	72.17 (95.58)	84.54 (93.73)
Class 4	74.65 (83.55)	53.01 (83.55)	61.79 (92.51)
Average	85.20 (86.18)	62.97 (86.18)	65.15 (93.60)

investigated [34]. Current research is now focusing on normalization of the MP rather than the initial image, and comparing its influence on classification results.

4.3 Merging spatial and spectral information through a kernel formulation

In Chapter 2, a novel approach was proposed for extracting spatial information for each pixel. It is based on a self-complementary area filter. If we consider a one-dimensional image, for each pixel we have its radiometric information $\mathbf{x}$ and its inter-pixel dependency $\Upsilon_{\mathbf{x}}$. In the following, we detail our strategy of using both types of information, then we extend this approach to hyperspectral data. Comparisons are made with the Extended Morphological Profile.

4.3.1 Kernel formulation

Camps-Valls *et al.* [21] studied different composite kernels for hyperspectral image classification. They extracted spatial information from a square window centered on the pixel. From their experiments, the weighted kernel that allows a trade-off between the spatial and spectral information seems to provide the best results. The authors proposed using a statistical estimate of inter-pixel dependency, the mean and standard deviation, as the spatial information. The use of that type of information in addition provides equivalent results, but using the weighted kernel, classification accuracy was increased by about 10%. But in terms of the final classification map, many pixels at structure boundaries were misclassified, though the structures were more homogeneous. This phenomenon is due to the square window: the *beneficial* effect is the homogeneous regions and the *negative* effect is the mis-classified pixels at boundaries. A weighted kernel is also successfully applied in [91]: The authors have performed wavelet-based multi-scale decomposition, to model local texture. Then this textural information was included in the processing by means of weighted kernels.

Another approach was proposed by Bovolo *et al.* in [17]. Rather than including spatial information with the kernel definition, inter-pixel dependency was modeled as the mean of pixel gray values in a pixel's neighborhood system. Then this information was included directly in the training process as new constraints for the optimization problem. However, only the inter-pixel dependency of the support vectors was used for the final classification. This approach was designed to cope with noisy training sets. When considering clean training sets, it is of limited interest, owing to the high correlation between adjancy pixels.

The approach proposed attempts to exploit the weighted kernel and the adaptive neighbors systems proposed previously [51]. The weighted kernel can be constructed thanks to kernel properties. It is possible to define kernels that use both kinds of information without running into intractable computational problems. Rules for kernel construction can be found in [112] and in Appendix A. We used the linearity property to construct the new kernel:

if k_1 and k_2 *are kernels, and* $\mu_1, \mu_2 \geq 0$*, then* $\mu_1 k_1 + \mu_2 k_2$ *is a kernel.*

Using the previous property, we defined the spectro-spatial kernel $\mathcal{K}$ as:

$$\begin{array}{rcl} \mathcal{K}^{\lambda}_{\gamma,\mu} \ : \ \mathbb{R}^n \times \mathbb{R}^n & \to & [0,1] \\ (\mathbf{x},\mathbf{z}) & \mapsto & \mu k^{spect}_{\gamma}(\mathbf{x},\mathbf{z}) + (1-\mu) k^{spat}_{\gamma}(\mathbf{x},\mathbf{z}) \\ & & 0 \leq \mu \leq 1, \ 0 \leq \lambda, \ 0 \leq \gamma. \end{array} \tag{4.4}$$

From our experiments [47], we defined the spectral kernel as:

$$\begin{array}{rcl} k^{spect}_{\gamma} \ : \ \mathbb{R}^n \times \mathbb{R}^n & \to & [0,1] \\ (\mathbf{x},\mathbf{z}) & \mapsto & \exp\left(-\dfrac{\|\mathbf{x}-\mathbf{z}\|^2}{2\gamma^2}\right). \end{array} \tag{4.5}$$

The spatial kernel is defined as follows:

$$\begin{array}{rcl} k_{\gamma}^{spat} \ : \ \mathbb{R}^n \times \mathbb{R}^n & \to & [0,1] \\ (\mathbf{x}, \mathbf{z}) & \mapsto & \exp\left(-\frac{\|\Upsilon_{\mathbf{x}} - \Upsilon_{\mathbf{z}}\|^2}{2\gamma^2}\right). \end{array} \tag{4.6}$$

For pixels belonging to the same set, the spatial information is the same. By proceeding in this way, we hope to obtain more homogeneous labeled zones in the final classification. The weighting factor μ controls the amount of the spatial and spectral information in the final kernel. It is tuned during the training process, as the parameter γ.

4.3.2 Extension to hyperspectral

As proposed in Section 2.8, the neighbors system is defined on the first principal component and generalized to each band. Thus Υ is a vector of same dimension as the original pixel-vector. Using a kernel function, the extension is immediate.

4.3.3 Experimental results

In this section, we compare the proposed approach to the MP and the EMP. For the first experiment, we provide a detailed analysis of the method and of the results obtained. Then a comparison is made. A complete description of the data set is given in the appendix C. In the following, readers are invited to refer to the appendix for the description of the data.

Classification accuracy was assessed using overall accuracy (OA), which is the number of correctly-classified samples divided by the number of test samples, average accuracy (AA), which represents the average of class classification accuracy, and the kappa coefficient of agreement (κ), which is the percentage agreement corrected by the level of agreement that could be expected due to chance alone. These criteria were used to compare classification results and were computed using the confusion matrix. Furthermore, the statistical significance of differences was computed using McNemar's test, which is based upon the standardized normal test statistic [56]:

$$Z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{4.7}$$

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 of Z indicates whether classifier 1 is more accurate than classifier 2 ($Z > 0$) or vice-versa ($Z < 0$).

SVM was used for all the experiments, and the parameters were fitted using cross-validation. Three SVM parameters need to be tuned: C the penalty term, γ the width of the Gaussian kernel, and μ the weighting factor in $\mathcal{K}$. From previous considerations in Section 3.5.2, C did not have a strong influence on the classification results when set higher than 10. For all the experiments, it was set at 200. The two other parameters were set using five-fold cross-validation, $\sigma^2 \in \{0.5, 1, 2, 4\}$ and $\mu \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. The SVM algorithm was implemented using a modified version of LIBSVM [24]. When using the Gaussian kernel, only the σ^2 value was tuned. Each original data set was scaled between $[-1, 1]$ using a band-wise range-stretching algorithm. The multi-class approach used was 'One vs. All', since this allows tuning of μ independently for each class.

4.3.3.1 Hyperspectral data set

University Area The first data set is the 'University Area'. This is composed of several man-made structures: buildings, roads, etc. and many classes of vegetation: grass, trees, bare soil. The original false color image can be seen in Figure 4.3 and details about the training and testing sets in Appendix C. For the classification, nine classes were defined, namely: trees, asphalt, bitumen, gravel, metal sheet, shadow, bricks, meadow, and soil.

Comparison to original SVM For this experiment, we first investigated the influence of the parameter λ on the definition of the neighborhood set. We tried several values for λ ranging from 2 to 40. Results of the classification are given in Table 4.8. Regarding the variation in the classification results, it seems that the parameter λ has a variable influence on the classification accuracies for each class. To explain this, three situations can be identified:

1. The spectral information is sufficient to discriminate the sample, and this spectral information is not noisy, so additional information is not needed (classes 9 and 5).
2. The size of the structure to which the sample belongs is:
    - large; when area filtering is performed, the sample is directly merged into a structure of a size larger than λ. This leads to better discrimination of the class concerned (classes 1, 7, and 8).
    - small; when area filtering is performed, the sample may be merged into another structure. For example, when considering the class *Tree*, this may be merged with class *Meadow*. This leads to poorer discrimination of the class concerned (class 4).
3. The class is highly textured and area filtering smoothes the structure. This leads to better discrimination of the class concerned (classes 2, 3, and 6).

According to the results in Table 4.8, the proposed approach outperforms the classic SVM. The statistical significance of differences in classification accuracy is reported in Table 4.9. The proposed classifier yields better results than those from the classic SVM. Kernel parameters found after the training step are given in Table 4.10. The value of μ confirms that a spatial kernel is useful for discrimination, since small values of μ are selected during the training process (corresponding to the inclusion of more spatial than spectral information). It is worth noting that too high a value of λ does not help classification; it makes area filtering too strong, thereby removing too many relevant structures.

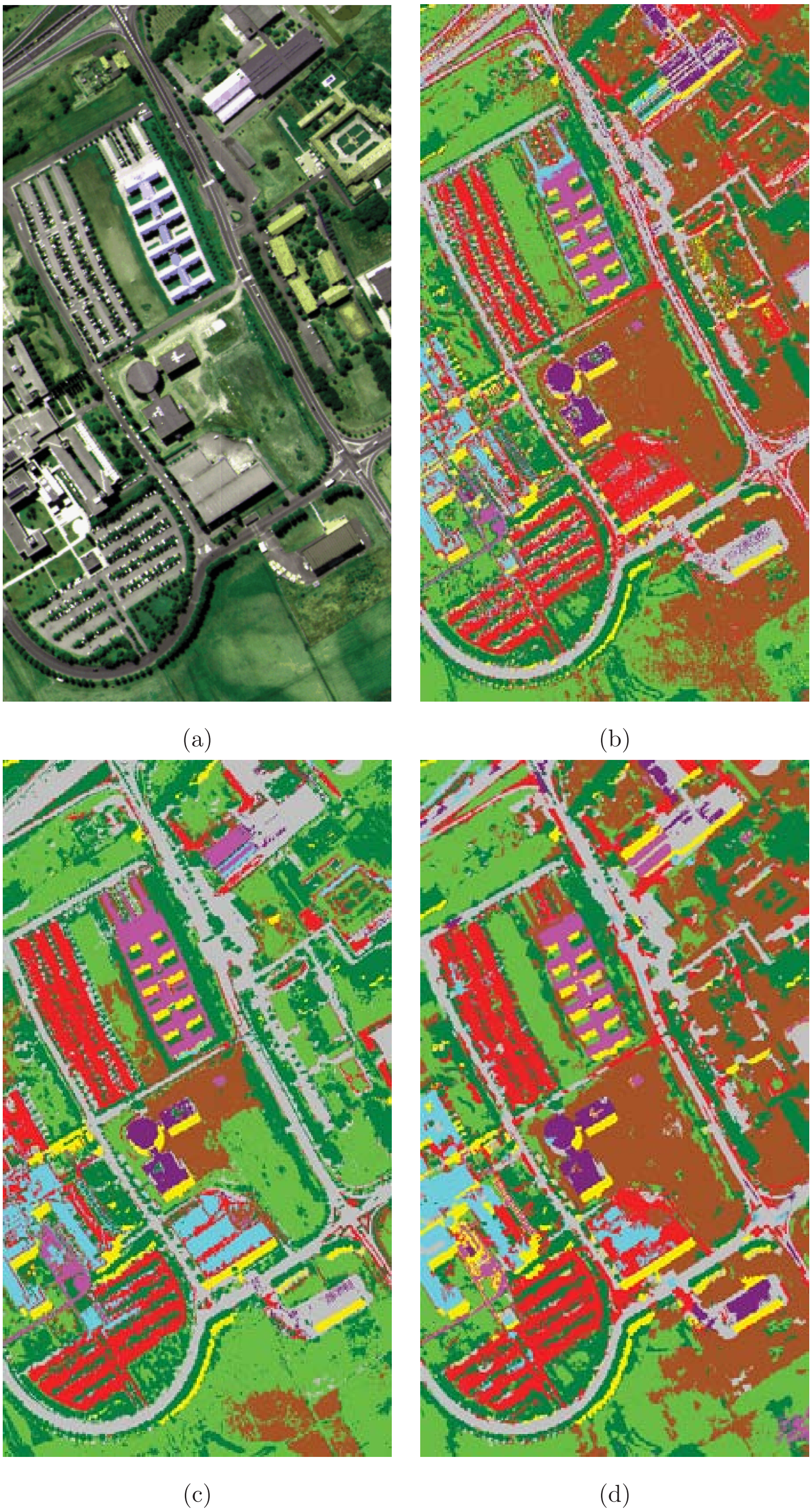
From this first experiment, it emerges that adding neighbors for classification improved the final results for some classes but not for all. Regarding the nature of the classes, the optimum neighborhood system and the ratio between spatial and spectral information seem to be different for each class and need to be tuned during the training process.

Figure 4.3 shows a false color of the original image and classification maps using the original kernels and the proposed kernel.

Comparison to EMP Principal component analysis was applied to the data. The first three components were retained and morphological processing was applied. For each component, 4 opening/closings with a disk as SE and an increment of 2 were computed. Thus the EMP was a vector with 27 components. For the SVM, a Gaussian kernel was used. The parameters were fitted in the same manner as the classic kernel in the previous experiment.

Results are listed in Table 4.8. The statistical difference is Z = 27.69. From the table, the proposed approach performs better in terms of classification accuracy than the EMP. However, for the *Asphalt* class the EMP produces better classification. This class correspond to the roads in the image, which are clearly fine, linear structures. Examining the thematic map Figure 4.3.(c), the roads seem to be better identified and more ‘continuous’ than in Figure 4.3.(d). The morphological profile extracts information about the shape and size of the structure, while the median value of the adaptive neighborhood is more an indication of the gray level distribution of the structure, hence it is not surprising that the MP performs best for that class. Note that both spectro-spatial approaches perform better for that class than the use of spectral information alone. In terms of classes with no typical shape such as meadow, gravel, or bare soil, the proposed approach outperforms the EMP in terms of classification accuracy. The adaptive neighborhood fits such structures better, and the value extracted from these structures helps in the discrimination (the smoothing effect describe above).

In the next experiment, an image of a dense urban area is classified. According to the considerations mentioned above, the EMP ought to be the best at dealing with this type of data.



(a) (b) (c) (d)

Figure 4.3: *(a) False color original image of University Area. (b) Classification map using the RBF kernel. (c) Classification map using the EMP. (d) Classification map using the proposed kernel where* $\lambda = 30$.

Table 4.8: *Classification accuracies for University Area data set. The best results for each class are reported in bold face.* Δ *is the difference between the best* $\mathcal{K}^n$ *and the original kernel.* $\mathcal{K}^n$ *means that classification was performed using the proposed kernel and area filtering of size* n.

Classifier	RBF	EMP	$\mathcal{K}^2$	$\mathcal{K}^5$	$\mathcal{K}^{10}$	$\mathcal{K}^{15}$	$\mathcal{K}^{20}$	$\mathcal{K}^{25}$	$\mathcal{K}^{30}$	$\mathcal{K}^{35}$	$\mathcal{K}^{40}$	Δ
1	80.64	**93.33**	80.41	80.08	82.39	83.25	84.57	86.32	84.36	84.57	86.13	5.68
2	68.47	73.40	72.28	72.27	75.17	72.28	76.38	76.32	**78.52**	75.18	75.16	10.05
3	73.80	52.45	76.51	77.13	81.71	87.04	**90.61**	89.71	84.80	85.52	85.90	16.81
4	97.49	**99.31**	96.44	96.64	94.19	94.94	95.07	94.61	96.87	94.84	97.65	0.16
5	99.49	99.48	99.11	99.26	99.41	96.51	99.48	98.29	**99.88**	99.63	99.63	0.39
6	94.83	61.90	95.29	94.33	97.26	95.67	96.88	94.09	95.61	**98.31**	97.87	3.48
7	91.50	**97.67**	93.38	94.51	89.70	91.50	94.14	93.91	95.56	96.17	93.83	4.67
8	91.88	95.17	92.07	92.67	**96.14**	94.46	94.70	95.84	95.44	95.30	95.79	4.26
9	97.04	92.29	95.04	94.83	95.56	92.29	93.56	93.24	**97.78**	96.73	97.25	0.74
OA	80.13	79.83	81.89	81.85	84.04	82.79	85.33	85.22	**86.11**	84.90	85.28	4.26
AA	88.33	85.00	88.95	89.09	90.17	89.77	91.71	91.37	91.98	91.80	**92.14**	3.19
κ	75.19	74.15	77.27	77.23	79.86	78.36	81.45	81.29	**82.35**	80.93	81.42	5.12

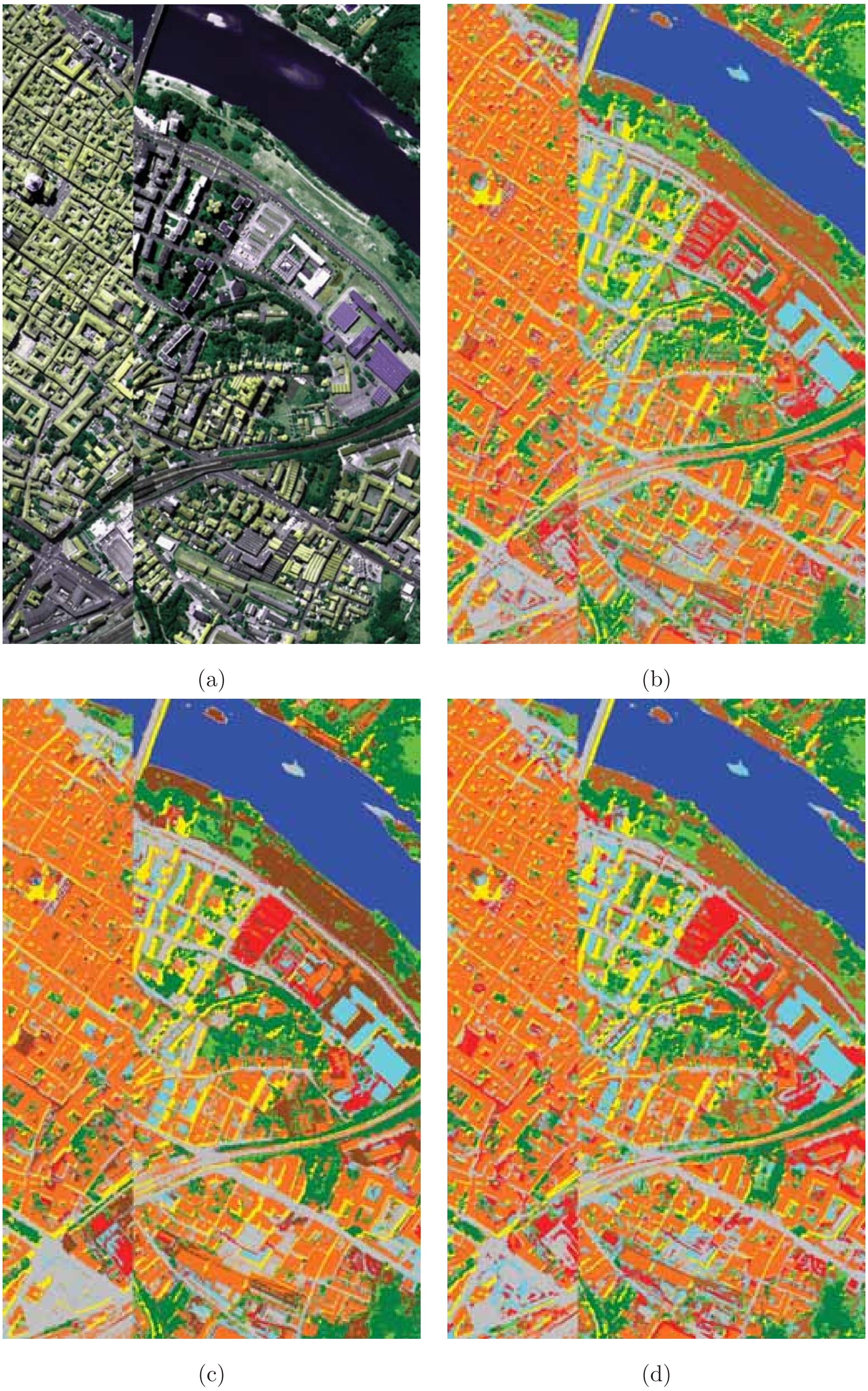
Pavia Center The second data set is an image of a dense urban area, the ’Pavia Center’. We want to investigate the usefulness of the spectro-spatial approach when a large number of structured objects are present in the image. The original false color image can be seen in Figure 4.4 and details about the training and testing sets are given in appendix C. For the classification, nine classes were defined, namely: water, tree, meadow, brick, soil, asphalt, bitumen, tile, and shadow.

Comparison to original SVM For the second data set, classification accuracies were already very good. For convenience, only the best results have been reported, with comparison to the original results, see Table 4.11. Following analysis of the parameter λ, we can say that classes 1, 2, 8, and 9 are separable using only the spectral information and adding spatial information does not help significantly in discrimination. Structures belonging to class 4 are merged together during the area filtering, and this leads to better discrimination. Textured classes (3, 5, and 7) are also better separated. However, in this image, class 6 corresponds to narrow streets, and the area filtering impairs classification of this type of structure. Nevertheless, in the final analysis OA, AA and κ are improved with the proposed kernel, and the results are statistically different Z=8.82.

Figure 4.4 presents the false color original image and classification maps using the original kernels and the proposed kernel.

Comparison to EMP The EMP was constructed following the same scheme as before. It comprised 27 features, and classification was performed using an SVM with a Gaussian kernel. Classification accuracies are reported in Table 4.11. The statistical difference is Z = -15.81, which means that the EMP performs better in this case. In terms of the global accuracy from the table, the EMP gave a slightly better result.

Regarding the class-specific accuracies, the same conclusions can be drawn as in the previous experiments. The class *Asphalt* is better classified with the EMP than with the approach proposed, while the class *Meadow* is discriminated better with the adaptive neighborhood. This confirms our conclusion above. The class *Tile* was already well classified using spectral information, hence no significant difference



(a) (b) (c) (d)

Figure 4.4: *(a) False color original image of Pavia Center. (b) Classification map using the RBF kernel. (c) Classification map using the EMP. (d) Classification map using the proposed kernel where* $\lambda = 20$.

Table 4.9: *Statistical Significance of Differences in Classification (Z) for the University Area data set.*

	rbf	$\mathcal{K}^{02}$	$\mathcal{K}^{05}$	$\mathcal{K}^{10}$	$\mathcal{K}^{15}$	$\mathcal{K}^{20}$	$\mathcal{K}^{25}$	$\mathcal{K}^{30}$	$\mathcal{K}^{35}$
$\mathcal{K}^{02}$	10.26								
$\mathcal{K}^{05}$	10.14	-0.38							
$\mathcal{K}^{10}$	22.66	15.51	15.86						
$\mathcal{K}^{15}$	15.29	5.17	5.43	-8.75					
$\mathcal{K}^{20}$	28.97	22.71	23.84	9.30	16.17				
$\mathcal{K}^{25}$	29.98	21.40	22.41	8.20	15.05	-1.06			
$\mathcal{K}^{30}$	33.30	26.84	27.66	14.37	20.11	5.85	7.35		
$\mathcal{K}^{35}$	27.33	18.55	18.89	5.98	13.6	-2.96	-2.26	-10.96	
$\mathcal{K}^{40}$	29.38	21.03	21.58	8.54	15.17	0.34	0.46	-7.12	3.78

Table 4.10: *Kernel parameter found by 5-fold cross-validation for University Area data set ($\lambda = 30$).*

Class	1	2	3	4	5	6	7	8	9
μ	0.3	0.1	0.1	0.1	0.9	0.5	0.1	0.2	0.4
γ	0.5	1	0.5	0.5	1	1	1	0.5	4

is found between EMP and the approach proposed, even this class corresponds to most of the roofs in the image.

Figure 4.4 shows the thematic map obtained using the EMP.

Washington DC Mall The last hyperspectral data set is 'Washington DC Mall'. Some parts of the image represent a dense urban area and some parts of the image represent vegetation area. The spatial resolution was lower, but it contains more spectral bands, see appendix C for more details, in particular for the training and testing set. Since the spatial resolution is lower, the structures are not as well defined as for the previous data set. This could disrupt the filtering. For the classification seven classes were defined, namely: roof, road, grass, tree, trail, water, and shadow.

Comparison to original SVM As the spatial resolution is lower, the parameter λ needs to be tuned to a lower value than in the previous experiments. We tested a range of values from 3 to 15. For all values, the OA was over 98.40% and not statistically different; the best κ value enabled us to choose 4 as the optimal value. The labeled data are relatively small and so it is difficult to produce statistically different results. However, regarding the results from the Table 4.12, the spectro-spatial approach leads to a improvement in classification accuracy. The statistical significance of the difference Z is 2.66.

In terms of the class-specific accuracies, the biggest improvement with the spectro-spatial approach is obtained for class 5, *Tree*, while class 2, corresponding to roads, is classified slightly worse.

Comparison to EMP The EMP comprised 27 features, resulting from the MP of the first three principal components. Classification was performed using the Gaussian SVM. Classification accuracies are reported in Table 4.12. The difference in classification is not statistically significant (Z=1.75).

Using the EMP, the shadow class is clearly classified worse than with the other two approaches. For this type of spatial resolution data, the square SE of radius 2 fails to fit many of the shadow structures, which are thus deleted at the start of the morphological processing: hence the spatial characteristic is

Table 4.11: *Classification accuracies for Pavia Center data set for the standard SVM, the EMP and the proposed approach.*

	OA	AA	κ	1	2	3	4	5	6	7	8	9
RBF	98.06	95.76	97.25	99.08	90.81	97.44	87.49	94.56	96.43	96.54	99.48	100
$\mathcal{K}^{20}$	98.43	97.13	97.79	99.15	90.04	98.12	94.00	99.45	95.82	98.15	99.47	99.93
EMP	98.95	97.72	98.51	99.82	90.94	95.50	99.07	99.06	97.99	97.49	99.62	99.97

Table 4.12: *Classification accuracies for Washington DC data set for the standard SVM, the EMP and the proposed approach.*

	OA	AA	κ	1	2	3	4	5	6	7
RBF	98.32	98.14	97.58	96.95	99.28	100	100	92.02	99.92	92.78
$\mathcal{K}^{4}$	98.96	98.48	98.50	98.28	97.60	100	100	99.75	99.92	93.81
EMP	98.33	96.79	97.59	97.29	99.28	99.43	99.84	98.27	99.92	83.41

estimated poorly. Considering the *Road* classification, the EMP performs better than the spectro-spatial approaches.

Figure 4.5 shows the thematic map obtained using the EMP.

4.3.3.2 Panchromatic data

In these experiments, we sought to confirm the results obtained for the hyperspectral data set and test the spectro-spatial approach on very high spatial resolution.

IKONOS data The first data set is the IKONOS data 'Reykjavik 1'. Experimental results were given in Chapter 2, page 42. The very high spatial resolution requires the λ value to be higher than in the previous experiments. In this experiment λ=40.

Comparison to original SVM Globally, accuracy has improved using the spectro-spatial SVM. The OA was 42.87% using gray level information alone, while it is 48.46% using the approach proposed. The classification accuracy for the classes *Open area* and *Shadow* has improved, from 58.37% to 70.24% and 85.50% to 89.94 % respectively. However, *Residential lawns* and *Streets* cannot be differentiated using the spectro-spatial approach, since no information about the size of the structure is added into the classification process.

Comparison to MP The MP leads to better classification accuracy: the Kappa coefficient is 40.80, as against 37.50 for the spectro-spatial SVM. In terms of the class-specific accuracy, the MP allows differentiation between *Street* and *Residential lawn*, while *Open area* and *Shadow* are classified better with the spectro-spatial SVM.

PLEIADES data [51] The second panchromatic data set was the PLEIADES data 'Toulouse 1 and 2'. A full description of the data and the testing and training sets can be found in appendix C. For both data sets, the MP was constructed using 15 geodesic openings/closings, the structuring element being a disk of size 2, 4 . . . 30.

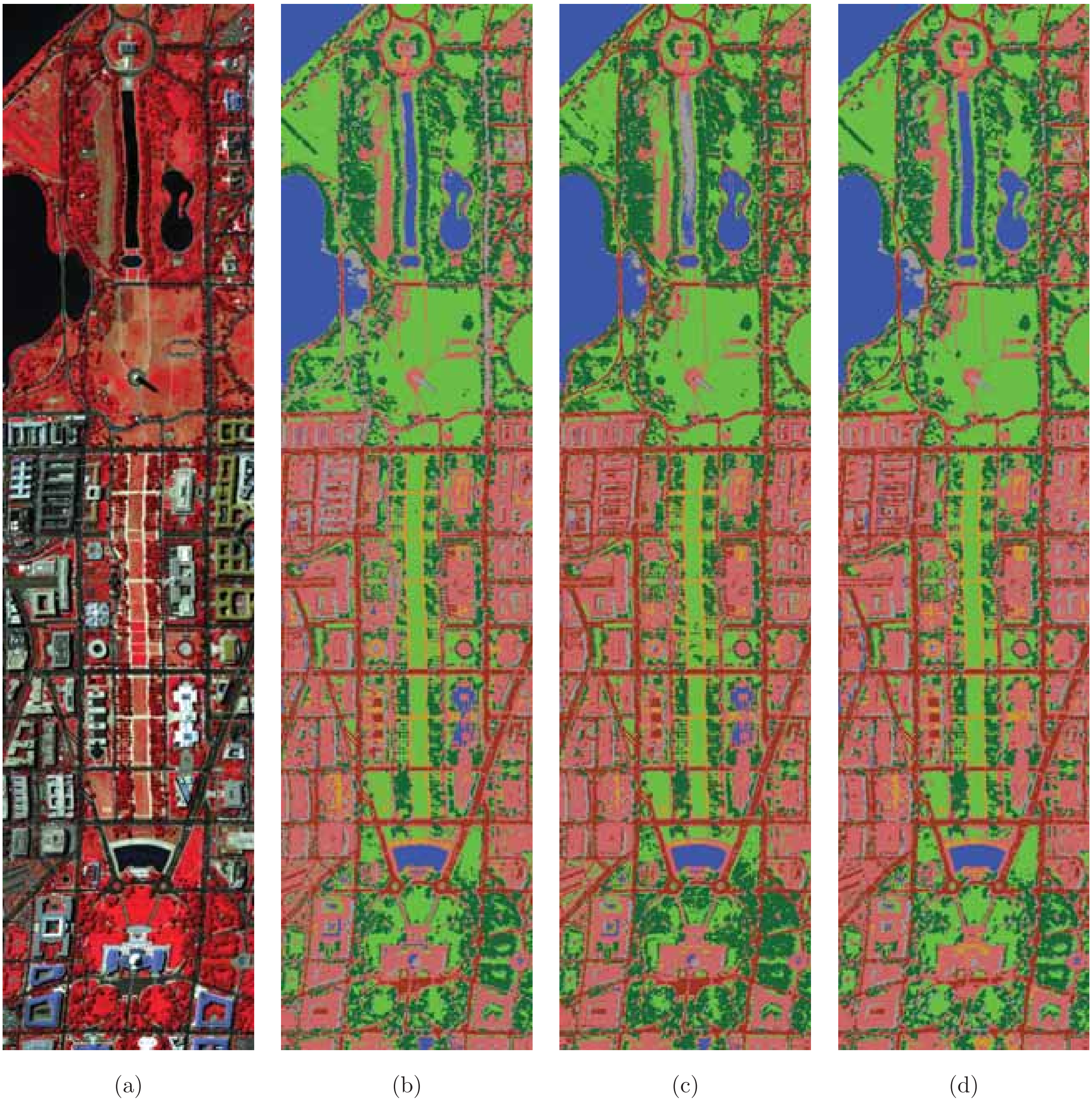


Figure 4.5: *(a) False color original image of Center. (b) Classification map using the RBF kernel. (c) Classification map using the EMP. (d) Classification map using the proposed kernel where* $\lambda = 20$.

Table 4.13: *Parameter μ found by cross-validation for* Toulouse 1

Class	Road	Building	Shadow	Open Area
Toulouse 1	0.3	0.6	0.1	0.4
Toulouse 2	0.6	0.6	0.4	0.9

Table 4.14: *Classification accuracies in percentage for the PLEIADES data* Toulouse 1 *and* 2*.* $\lambda = 45$ *for the standard SVM, the EMP and the proposed approach.*

	Toulouse 1			Toulouse 2		
	SVM	SS SVM	MP + SVM	SVM	SS SVM	MP + SVM
OA	76.99	82.25	80.58	71.34	77.54	83.74
AA	75.53	80.91	78.55	63.75	73.01	81.49
κ	68.33	75.80	73.14	60.39	69.22	77.90
Road	70.69	77.31	82.42	95.47	93.15	85.64
Building	86.09	93.86	48.03	99.64	100.00	93.91
Shadow	60.97	64.15	85.52	59.90	62.89	74.73
Open area	84.35	88.32	98.15	0	36.02	65.69

Comparison to original SVM Classification accuracy is reported in Table 4.14. The optimum value for $\lambda = 45$ was found heuristically during the training step.

In terms of the three global classification accuracy estimates (OA, AA, and κ), the approach proposed led to an improvement in classification. Except for one class for the second data set, the simultaneous use of spatial and spectral information invariably leads to better discrimination of the different classes.

In Table 4.13, the values of μ for each binary sub-problem are reported. For the first data set, the spatial information is weighted heavier than the spectral information, while for the second data set the usage seems to be more balanced. This tends to prove that the amount of each kind of information in fact needs to be carefully tuned during the training process and should not be set to the same value for all classes.

Figure 4.6 shows the original image and the thematic maps obtained using the original kernel and the proposed kernel.

Comparison with the MP As with the spectro-spatial SVM, the MP leads to an improvement of the classification in terms of accuracy. For the first data set, the spectro-spatial kernel performs better, while the MP performs better for the second data set.

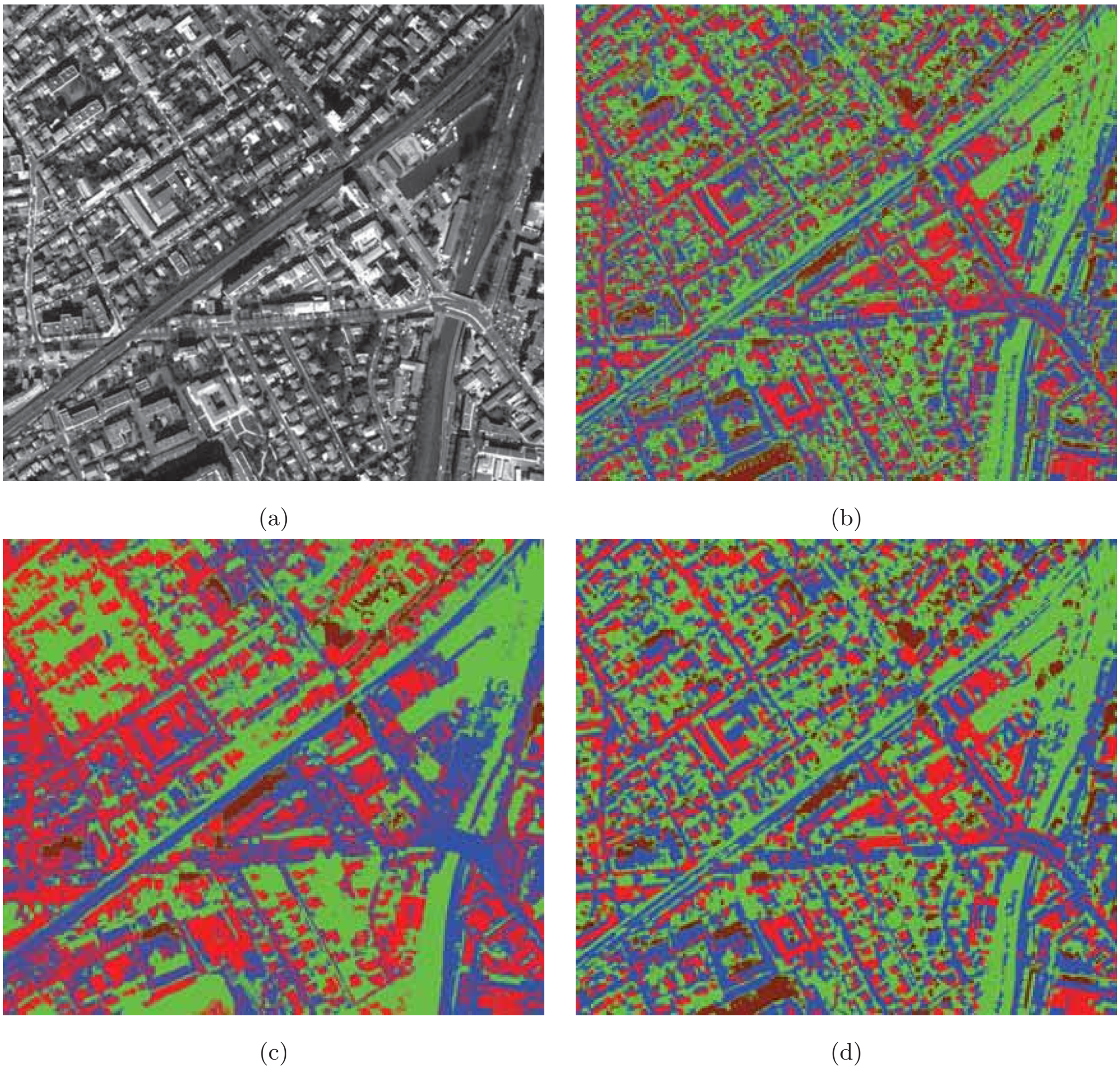
As with the hyperspectral data sets, the MP performs better for the structured classes: *Road* and *Building*, while the spectro-spatial approach performs better for *Shadow* and *Open area*.

Figure 4.6 presents the thematic map obtained using the MP.

4.3.4 General comments

The proposed approach outperforms the classic Gaussian radial basis kernel. Some remarks are discussed below:

- **Area filtering**: Used as a pre-processing step, this filter provides a simplified image where relevant structures are still present and details are removed. However, too high a value for λ may eliminate small structures, such as trees in the 'University Area' data experiment or narrow streets in the



(a)

(b)

(c)

(d)

Figure 4.6: *(a) False color original image of Toulouse 1, (b) Classification map using the RBF kernel, (c) Classification map using the MP, (d) Classification map using the proposed kernel where* $\lambda = 45$.

'Pavia Center' data experiment. This parameter needs to be chosen carefully. Despite the fact the filter is not a connected filter, since the the values of λ are always low, the output is consistent with the initial image. In future works, a method for definig a connected filter needs to be addressed.

- **Multiband extension**: Due to the problem that there is no *ordering relation* between vector-valued pixels, direct extension of the area filter is not possible. The proposed approach was extended to hyperspectral data by considering the first principal component. This methodology had been successfully applied when using morphological filtering for the purpose of classification. In our experiments, the results confirm the interest of this scheme.
- **Spectro-spatial kernel**: This formulation allows a compact definition of the classification algorithm. Thus very few parameters need to be tuned during the training stage. From these experiments, the values of γ does not seem to have a strong influence on the overall results ($\approx 1-2\%$) when the data are scaled between $[-1, 1]$. The relative proportions of spatial and spectral information has a stronger influence in the final classification.
- **$\boldsymbol{\lambda}$ value**: In the approach proposed, this parameter was selected globally, *i.e.*, all the classes share the same value. But in terms of class classification accuracy, see Table 4.8, it can clearly be seen that the optimum value of λ is class-dependent. Hence in future work λ ought to be included in the training process and selected for each class independently.
- **Spectro-spatial SVM vs. EMP + SVM**: Regarding the experimental results, both approaches lead to improved classification in terms of accuracy. The spectro-spatial approach performs better for peri-urban areas, while the EMP leads to better results for very dense urban areas. However, the results are highly correlated to class definition: when considering classes according to geometrical characteristics (size or shape) the EMP performs better, but when considering classes according to textural or spectral characteristics, the spectro-spatial approach leads to better classification. Hence the method needs to be chosen in accordance with the data and the classes defined.

4.3.5 Discussion

A novel approach using spatial and spectral information has been presented. Defining a weighted kernel allows it to be applied with low complexity. A key point is the definition of the spatial neighborhood and spatial information. In this thesis, we use the median value as an estimation of the inter-pixel dependency within a structure. Experiments have yielded good results in terms of classification accuracy. Comparison was made with the EMP. The proposed approach appears to perform better when the image area is not a dense urban area (University Area and Washington DC), while the EMP performs better for dense urban areas (Pavia Center). This is due to the morphological processing, which extracts geometrical information about the structure, while the proposed approach extracts only information about its gray-level distribution.

This result is not however surprising, as explained in the experiment. One possible extension lies in the definition of new spatial information. The median value does not provide information about the shape, size, or homogeneity of the neighborhood set. Other parameters could be extracted, thus leading to another definition of the spatial kernel. For instance, textural information could be extracted. In [88], a method for the estimation of the characteristic scale at each pixel was proposed. Such type of information needs also to be included in the classification process.

Part III

Data Fusion

Chapter 5

Decision Fusion

Abstract

The following chapter deals with the problem of the fusion of several classifier outputs. The aim of decision fusion is to obtain a more reliable classification map using several classifiers, each one having its own advantages. The following work is based on fuzzy logic, which is presented at the beginning of the chapter. Then information fusion theory is reviewed and the fusion framework is detailed. It is based on the estimation of the local confidence of each classifier by modeling the classifier outputs using a fuzzy set. Global confidence is defined consistent with the per-class performance of each classifier. The fusion is performed according to the local and global confidence. Experimental results are given for two panchromatic images. A dedicated fusion approach is then proposed for the SVM where the distance to the hyperplane is used a measure of the reliability of sources.

Contents

IN THE second part of the thesis, we have dealt with many classification algorithms: maximum likelihood, neural network, and particular attention was paid to support vector machines. These algorithms were applied to different inputs: spectral features, spatial features, morphological profile, etc. All these different methods have their own characteristics and advantages. Neural network and SVM have the advantage that no prior information about the distribution of the input data is needed. However, if an accurate multivariate statistical model can be determined, statistical methods should provide better classification accuracies than neural networks. Classifiers based on possibilistic models do not need any training and class definitions can be achieved using linguistic variables [27]. Furthermore, computation time is usually shorter with statistical approaches than neural methods or SVM.

For a given data set, performance in terms of *overall* and *by class* classification accuracies usually depends on the classes considered, *i.e.* on their spectral and spatial characteristics. For instance, methods based on morphological filtering are well suited to classifying structures with a typical spatial shape, like man-made constructions. On the other hand, algorithms based on spectral information perform better for the classification of vegetation and soils. As a consequence, we advocate using several approaches and trying to capitalize on the strengths of each algorithm. This concept is called *decision fusion*[6]. Decision fusion can be defined as the process of fusing information from several individual data sources after each data source has undergone a preliminary classification. For instance, Benediktsson and Kanellopoulos [6] proposed a multi-source classifier based on a combination of several neural/statistical classifiers. The samples are first classified using two classifiers (a neural network and a multi-source classifier); every sample with results that agree is assigned to the corresponding class. Where there is a conflict between the classifiers, a second neural network is used to classify the remaining samples. The main limitation of this method is the need for large training sets to train the different classifiers. In [69], Jeon and Landgrebe used two decision fusion rules to classify multi-temporal Thematic Mapper data. Recently, Lisini *et al.* [87] proposed combining sources according to their class accuracies. In the present study, the decision fusion rule is modeled using fuzzy data fusion rules. Fuzzy-based fusion techniques have already been applied in various decision fusion schemes. For instance, Tupin *et al.* [125] combined several structure detectors to classify SAR images using the Dempster-Shafer theory. Chanussot *et al.* [30] proposed several strategies to combine the output of a line detector applied to multi-temporal images. Also dealing with multi-temporal SAR images, Amici *et al.* [3] investigate the usefulness of fuzzy and neuro-fuzzy techniques to fuse the multi-temporal information for monitoring flooded areas.

For data fusion-based classification methods, two main categories can be defined: The fusion of features or information prior to classification, and the fusion of decisions post-classification. Both these approaches are addressed in the thesis. Decision fusion is investigated below; feature fusion will be addressed in the following chapter.

In this chapter, we propose a general framework for aggregating the results from different classifiers. Conflicting situations, where the different classifiers disagree, are resolved by estimating the point-wise accuracy and modeling the global reliability for each algorithm [132]. This leads to the definition of an adaptive fusion scheme ruled by these reliability measures. The proposed algorithm is based on fuzzy sets and possibility theory. Then this framework is used as a basis for the fusion of SVM classifiers with different inputs.

The framework of the problem addressed is modeled as follows: For a given data set, n classes are considered, and m classifiers are assumed to be available. For an individual pixel, each algorithm provides an output of a membership degree for each of the considered classes. The set of these membership values is then modeled as a fuzzy set, and the corresponding degree of fuzziness determines the point-wise reliability of the algorithm. The overall accuracy is defined manually for each class after a statistical study of the results obtained with each classifier used separately. Hence the fusion is performed by aggregating the different fuzzy sets provided by the different classifiers. It is adaptively ruled by the reliability information and does not require any further training. The decision is postponed until the end of the fusion process in order to take best advantage of each algorithm and enable more accurate results in conflicting situations.

This chapter is organized as follows. Fuzzy set theory and measures of fuzziness are briefly presented in Section 5.1. Section 5.1.2 presents the model for the output of each classifier in terms of a fuzzy set. The problem of information fusion is next discussed in Section 5.2. The proposed fusion scheme is detailed in Section 5.3 and experimental results are presented in Section 5.4. Then the framework is modified for application to SVM classifiers, and experiments on hyperspectral data are presented. Finally, conclusions are drawn.

5.1 Fuzzy set theory

Traditional mathematics assigns a membership value of 1 to elements that are members of a set, and 0 to those that are not, thus defining *crisp sets*. However, *fuzzy set* theory deals with the concept of partial membership of a set, using real-valued membership degrees ranging from 0 to 1. Fuzzy set theory was introduced in 1965 by Zadeh as a means of modeling the vagueness and ambiguity in complex systems [133]. It is now widely used to process imprecise or uncertain data [77, 124]. In particular, it offers an appropriate framework for handling the output of any given classifier for further processing, since this does not usually come in binary form and includes a degree of ambiguity. In this section, we first recall general definitions and properties of fuzzy sets. Then, we detail the model used for representing the classifier output.

5.1.1 Fuzzy set theory

5.1.1.1 Definitions

Definition 5.1 (Fuzzy subset) *A fuzzy subset*[1] *F of a reference set U is a set of ordered pairs $F = \{(x, \mu_F(x)) \mid x \in U\}$, where $\mu_F : U \to [0, 1]$ is the membership function of F in U.*

Definition 5.2 (Normality) *A fuzzy set is said to be* normal *if and only if:* $\max \mu_F(x) = 1$.

Definition 5.3 (Support) *The support of a fuzzy set F is defined as:*

$$Supp(F) = \{x \in U \mid \mu_F(x) > 0\}.$$

Definition 5.4 (Core) *The core of a fuzzy set is the (crisp) set containing the points with the largest membership value (1). It is empty if the set is non-normal.*

5.1.1.2 Logical operations

Classical Boolean operations extend to fuzzy sets [133]. With F and G two fuzzy sets, classic extensions are defined as follows:

Equality The equality between two fuzzy sets is defined as the equality of their membership functions:

$$\mu_F = \mu_G \Leftrightarrow \forall x \in U, \mu_F(x) = \mu_G(x). \tag{5.1}$$

Inclusion The inclusion of one set within another is defined by the inequality of their membership functions:

$$\mu_F \subset \mu_G \Leftrightarrow \forall x \in U, \mu_F(x) \leq \mu_G(x). \tag{5.2}$$

Union The union of two fuzzy sets is defined by the maximum of their membership functions:

$$\forall x \in U, (\mu_F \cup \mu_G)(x) = \max\{\mu_F(x), \mu_G(x)\}. \tag{5.3}$$

[1] For convenience, we will use the term *fuzzy set* instead of *fuzzy subset* in the following, where a fuzzy set F is described by its membership function μ_F.

Intersection The intersection of two fuzzy sets is defined by the minimum of their membership functions:

$$\forall x \in U, (\mu_F \cap \mu_G)(x) = \min\{\mu_F(x), \mu_G(x)\}. \tag{5.4}$$

Complement The complement of a fuzzy set F is defined by:

$$\forall x \in U, \mu_{\overline{F}}(x) = 1 - \mu_F(x). \tag{5.5}$$

5.1.1.3 Measures of fuzziness

Fuzziness is an intrinsic property of fuzzy sets. To measure to what degree a set is fuzzy, and thus estimate the corresponding ambiguity, several definitions have been proposed [40] [134]. Ebanks [45] proposed defining the degree of fuzziness as a function f with the following properties:

1. $\forall F \subset U$, if $f(\mu_F) = 0$ then F is a crisp set
2. $f(\mu_F)$ is maximum if and only if $\forall x \in U, \mu_F(x) = 0.5$
3. $\forall (\mu_F, \mu_G) \in U^2$, $f(\mu_F) \geq f(\mu_G)$ if $\forall x \in U \begin{cases} \mu_G(x) \geq \mu_F(x) & if \quad \mu_F(x) \geq 0.5 \\ \mu_G(x) \leq \mu_F(x) & if \quad \mu_F(x) \leq 0.5 \end{cases}$
4. $\forall F \in U$, $f(\mu_F) = f(\mu_{\overline{F}})$. A set and its complement have the same degree of fuzziness
5. $\forall (\mu_F, \mu_G) \in U^2$,$f(\mu_F \cup \mu_G) + f(\mu_F \cap \mu_G) = f(\mu_F) + f(\mu_G)$

Derived from probability theory and classic Shannon entropy, De Luca and Termini [40] defined a fuzzy entropy satisfying the above five properties:

$$H_{DTE}(\mu_F) = -K \sum_{i=1}^{n} \Big(\mu_F(x_i)\log_2(\mu_F(x_i)) + (1-\mu_F(x_i))\log_2(1-\mu_F(x_i))\Big). \tag{5.6}$$

Bezdeck [96] proposed an alternative measure of fuzziness based on a multiplicative class.

Definition 5.5 (Multiplicative Class) *A multiplicative class is defined as:*

$$H_*(\mu_F) = K \sum_{i=1}^{n} g(\mu_F(x_i)), \; K \in R^+ \tag{5.7}$$

where $g(\mu_F)$ is defined as:

$$\begin{cases} g(t) &= \widetilde{g}(t) - \min\limits_{0 \leq t \leq 1} \widetilde{g}(t) \\ \widetilde{g}(t) &= h(t)h(1-t) \end{cases} \tag{5.8}$$

and h is a concave increasing function on $[0,1]$:

$$h : [0,1] \rightarrow R^2, \forall x \in [0,1] \; h'(x) > 0 \; and \; h''(x) < 0. \tag{5.9}$$

The multiplicative class makes it possible to define various measures of fuzziness, where different choices of g lead to different behaviors. For instance, let $h : [0,1] \rightarrow R^+$ be $h(t) = t^\alpha, \; 0 < \alpha < 1$. The function h satisfies the required conditions for the multiplicative class, and the function:

$$H_{\alpha QE}(\mu_F) = \frac{1}{n2^{-2\alpha}} \sum_{i=1}^{n} \mu_F(x_i)^\alpha (1-\mu_F(x_i))^\alpha \tag{5.10}$$

is a measure of fuzziness, $\alpha - Quadratic\ entropy$. Rewriting (5.10) as:

$$\begin{cases} H_{\alpha QE}(\mu_F) &= \dfrac{1}{n}\sum\limits_{i=1}^{n} S_{\alpha QE}(\mu_F(x_i)) \\ S_{\alpha QE}(\mu_F(x_i)) &= \dfrac{\mu_F(x_i)^\alpha (1-\mu_F(x_i))^\alpha}{2^{-2\alpha}} \end{cases} \tag{5.11}$$

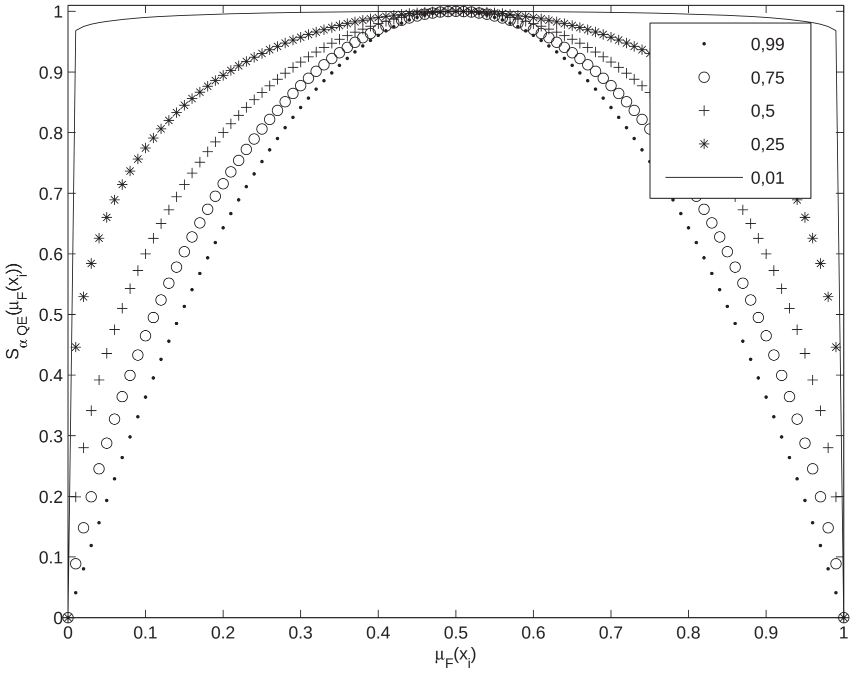


Figure 5.1: *Influence of parameter α on $S_{\alpha QE}$*

Table 5.1: *Degree of fuzziness for different fuzzy sets computed using α − Quadratic entropy*

$\alpha =$	0.01	0.25	0.5	0.75	0.99
$H_{\alpha QE}(\mu_F(x)) = atan(x)$	0.959	0.600	0.394	0.271	0.196
$H_{\alpha QE}(\mu_F(x)) = x$	0.967	0.725	0.549	0.423	0.333
$H_{\alpha QE}(\mu_F(x)) = U(x)$	0	0	0	0	0
$H_{\alpha QE}(\mu_F(x)) = 0.5$	0.993	0.840	0.707	0.594	0.503

we can analyze the influence of parameter α (see Figure 5.1): the measure becomes more and more selective as α increases from 0 to 1. With α close to 0, all the fuzzy sets have approximately the same degree of fuzziness and the measure is not sensitive to changes in μ_F, whereas with α close to 1, the measure is highly selective, with the degree of fuzziness decreasing rapidly when the fuzzy set differs from $\mu_F = 0.5$. Consequently, an intermediate value such as $\alpha = 0.5$ usually provides a good trade-off [96].

Examples of fuzzy sets and their fuzziness values are given in Figure 5.2 and Table 5.1, respectively. For the binary set, fuzziness is null with respect to condition 1 above. Condition 2 is fulfilled, since the fuzziness is maximum for $\mu_F(x) = 0.5$. In respect of condition 3, the fuzzy set with the arctan membership function has a lower fuzziness than the fuzzy set with the linear membership function.

5.1.2 Class representation

An n-class classification problem is considered, for which m different classifiers are available. For a given pixel x, the output of classifier i is the set of numerical values:

$$\{\mu_i^1(x), \mu_i^2(x), \dots, \mu_i^j(x), \dots, \mu_i^n(x)\} \tag{5.12}$$

where $\mu_i^j(x) \in [0, 1]$ (after a normalization, if required) is the membership degree of pixel x to class j according to classifier i. The higher this value, the more likely it is that the pixel belongs to class j (if a

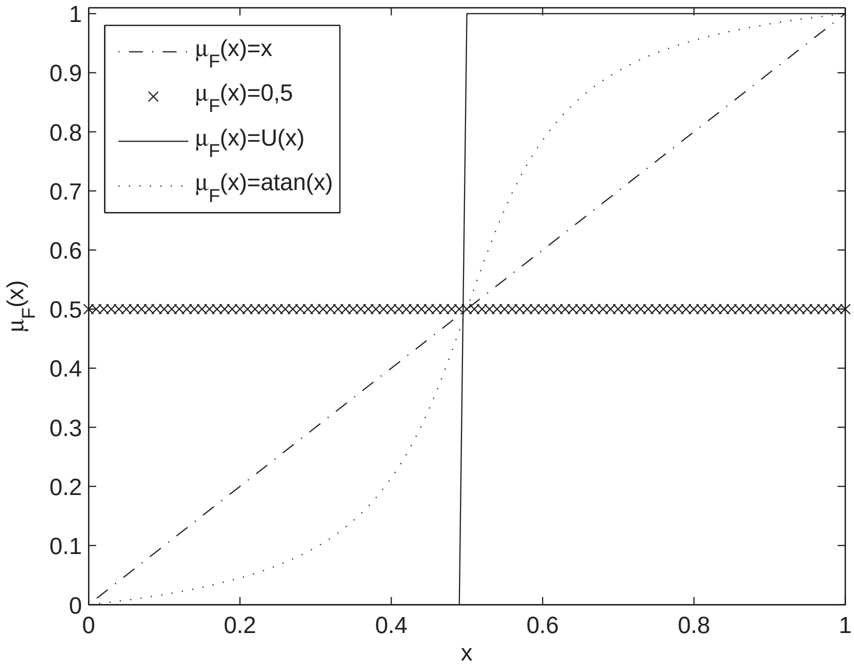


Figure 5.2: *Example of four fuzzy sets with different degrees of fuzziness.*

single classifier is used, the decision is taken by selecting the class j maximizing $\mu_i^j(x)$: $class_{selected}(x) = argmax_j(\mu_i^j(x)))$. Depending on the classifier, $\mu_i^j(x)$ can be of a different nature: probability, posterior probability at the output of a neural network, membership degree at the output of a fuzzy classifier, *etc.* In all cases, the set $\pi_i(x) = \{\mu_i^j(x), j = 1, ..., n\}$ provided by each classifier i may be considered as a fuzzy set.

To sum up: For every pixel x, m fuzzy sets are computed, one for each classifier. This set of fuzzy sets constitutes the input for the fusion process:

$$\{\pi_1(x), \pi_2(x), \ldots, \pi_i(x), \ldots, \pi_m(x)\}. \tag{5.13}$$

Two conflicting sets are represented in Figure 5.3: for this pixel, trusting the first classifier (on the left), class number 4 would be selected, whereas if we trust the second classifier (on the right), it would be class number 5. The handling of such conflicting situations is the central issue that needs to be addressed by the fusion system. In fact, the fusion of non-conflicting results is of little interest in our case: though it might increase our confidence in the corresponding result, it certainly won't change the final decision, and hence won't improve classification performance. On the contrary, in the case of conflicting results, at least one classifier is wrong, and the fusion gives a chance to correct this and improve classification performance. Fuzzy set theory provides various combination operators for aggregating these fuzzy sets, which are discussed in the next section.

5.2 Information Fusion

After briefly recalling the basics of data fusion, in this section we discuss the problem of measuring the confidence of individual classifiers. We end by proposing an adaptive fusion operator. In the following, we denote the fuzzy set i by π_i and the number of sources by m.

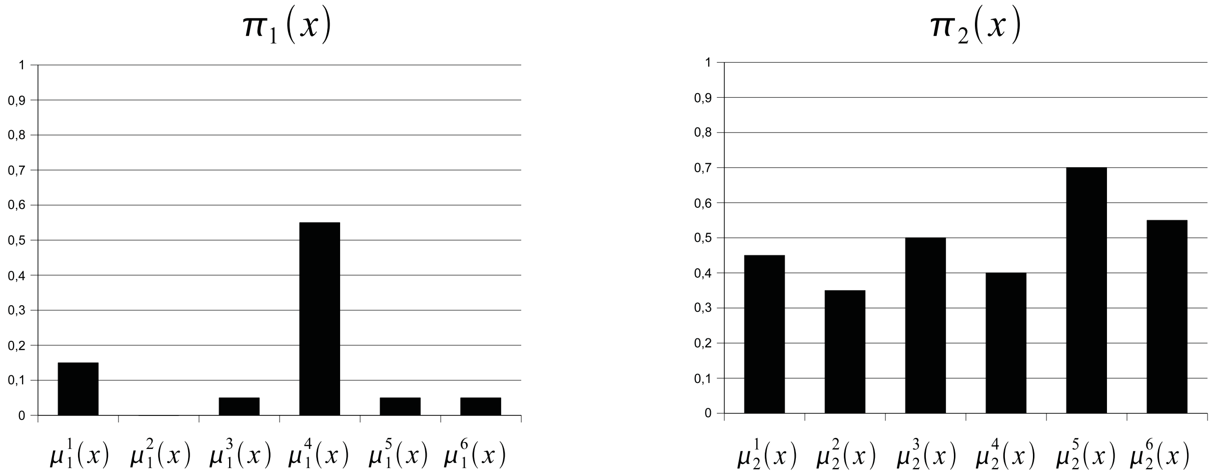


Figure 5.3: *Example of two conflicting sets π for a given pixel x*

5.2.1 Introduction

Data fusion consists of combining information from several sources in order to improve the decision [15]. As previously mentioned, the most challenging issue is to solve conflicting situations where some of the sources disagree. Numerous combination operators have been proposed in the literature. They can be classified into three different kinds, depending on their behavior [95]:

- *Conjunctive combination*: This corresponds to *severe* behavior. The resulting fuzzy set is necessarily smaller than the initial sets and the core is included in the initial cores (it can only decrease). The largest conjunctive operator is the fuzzy intersection (5.4) leading to the following fuzzy set: $\pi_\wedge(x) = \bigcap_{i=1}^{N} \pi_i(x)$. *T-norms* are conjunctive operators. They are commutative, associative, increasing, and have $\pi_i(x) = 1$ as a neutral element (*i.e.* if $\pi_2(x) = 1$ then $\pi_\wedge(x) = \pi_1(x) \cap \pi_2(x) = \pi_1(x)$). They satisfy the following property:

$$\pi_\wedge(x) \leq \min_{i \in [1,m]} \pi_i(x). \tag{5.14}$$

- *Disjunctive combination*: This corresponds to *indulgent* behavior. The resulting fuzzy set is necessarily larger than the initial sets and the core contains the initial cores (it can only increase). The smallest disjunctive operator is the fuzzy union (5.3), leading to the following fuzzy set: $\pi_\vee(x) = \bigcup_{i=1}^{N} \pi_i(x)$. *T-conorms* are disjunctive operators. They are commutative, associative, increasing, and have $\pi_i(x) = 0$ as neutral element. They satisfy the following property:

$$\pi_\vee(x) \geq \max_{i \in [1,m]} \pi_i(x). \tag{5.15}$$

- *Compromise combination*: This corresponds to intermediate *cautious* behaviors. $T(a, b)$ is a compromise combination if it satisfies:

$$\min(a, b) < T(a, b) < \max(a, b). \tag{5.16}$$

For the purpose of illustration, we can consider the following imaginary problem. To estimate how old a person is, two estimates are available, each modeled by a fuzzy set. These fuzzy sets are represented in Figure 5.4.a - note that they are highly conflicting. From these two sources of information, we want to classify a person into one of the three following classes: young (under 30), middle aged (between 30 and

65), or old (over 65). To illustrate the three possible modes of combination, we aggregate the information using the min operator (T-norm), the max operator (T-conorm), and the three different compromise operators. The results are shown in Figure 5.4. The decision is taken by selecting the class corresponding to the maximum membership.

Conjunctive combination Figure 5.4.b shows the result obtained with the min operator, *i.e.* the less severe conjunctive operator. It is a unimodal fuzzy set. This fuzzy set is sub-normalized, but this problem could be solved using $\pi'_{\wedge}(x) = \frac{\pi_{\wedge}(x)}{\sup_x(\pi_{\wedge}(x))}$ but this would not change the shape of the result. In this case, the decision would be *middle aged*, which is not compatible with any of the initial sources. The sources here disagree strongly and the conjunctive fusion does not help classification. To sum up: conjunctive operators are not suited for conflicting situations.

Disjunctive combination Figure 5.4.c shows the result obtained with the max operator, *i.e.*, the less indulgent disjunctive operator. The resulting membership function is multi-modal and all the maxima are of equal amplitude. Again, no satisfactory decision can be made.

Compromise combination Three different operators of this type are discussed, all based on measuring the conflict between sources, defined as $1 - C$ with:

$$C(\pi_1, \pi_2) = \sup_x \min(\pi_1(x), \pi_2(x)). \tag{5.17}$$

These three compromise combination operators have been proposed by D. Dubois and H. Prade in [104]. Bloch has classified these operators as *Contextual Dependant (CD) Operators* [14] - where context may be, e.g., a conflict between the sources, knowledge about reliability of a source, or a degree of spatial information. These operators have been proposed in possibility theory [44] but they can also be used in fuzzy set theory for combining membership functions [14]. Being able to adapt to the context, these operators are more flexible and hence yield interesting results. The first operator considered (5.18):

$$\pi(x) = \begin{cases} \max\left(\frac{\min(\pi_1(x),\pi_2(x))}{C(\pi_1,\pi_2)}, \min(\max(\pi_1(x), \pi_2(x)), 1 - C(\pi_1, \pi_2))\right) & \text{if} \quad C(\pi_1, \pi_2) \neq 0 \\ \max(\pi_1(x), \pi_2(x)) & \text{if} \quad C(\pi_1, \pi_2) = 0 \end{cases} \tag{5.18}$$

adapts its behavior as a function of the conflict between the sources [14]:

- it is conjunctive if the sources have low conflict
- it is disjunctive if the sources have high conflict
- it behaves in a compromise way in the event of partial conflict

Figure 5.4.d shows the result obtained using operator (5.18). The corresponding decision (middle aged) is still not satisfactory.

In this case, some information on source reliability needs to be included, and the most reliable source(s) should be favored in the fusion process. Different situations may be considered [14]:

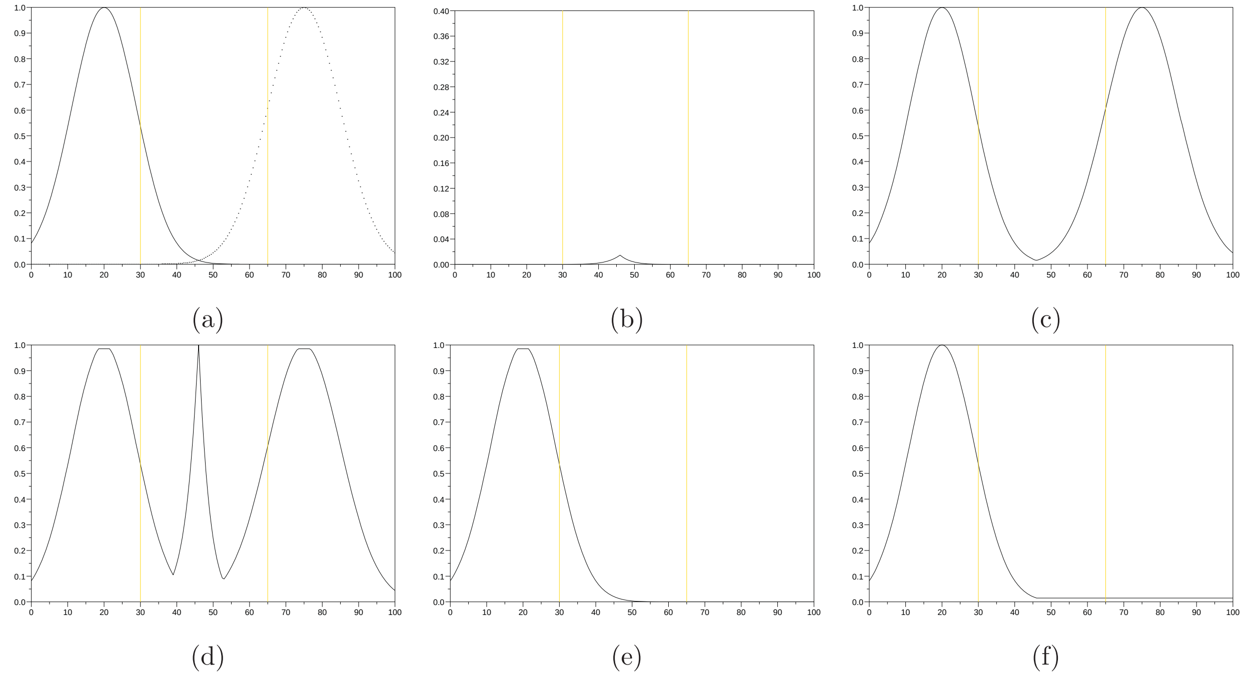
- It is possible to assign each source a numerical degree of reliability.
- A subset of sources is reliable, but we do not know which one(s).
- The relative reliability of the sources are known, but with no quantitative values. However, priorities can be defined between the sources.

The following two adaptive operators are examples of *prioritized fusion operator*[104].

$$\pi(x) = \min(\pi_1(x), \max(\pi_2(x), 1 - C(\pi_1, \pi_2))) \tag{5.19}$$

$$\pi(x) = \max(\pi_1(x), \min(\pi_2(x), C(\pi_1, \pi_2))). \tag{5.20}$$

For both operators, when $C(\pi_1, \pi_2) = 0$, π_2 contradicts π_1 and only the information provided by π_1 is retained. In this case, π_2 is considered as a specific piece of information while π_1 is viewed as a fuzzy



(a) (b) (c)

(d) (e) (f)

Figure 5.4: *Examples of combination operators: (a) shows the distribution of the two possibilities, (b) and (c) show the result of the* min *and the* max *operators, respectively. (d), (e) and (f) show the results of the three compromise operators presented in (5.18), (5.19) and (5.20), respectively.*

default value. Assuming π_1 is more accurate than π_2, we get the result shown in Figure 5.4.e and f, permitting a satisfactory decision.

To sum up: conjunctive and disjunctive combination operators are ill-suited to handling conflicting situations. These situations need to be solved using CD operators incorporating reliability information.

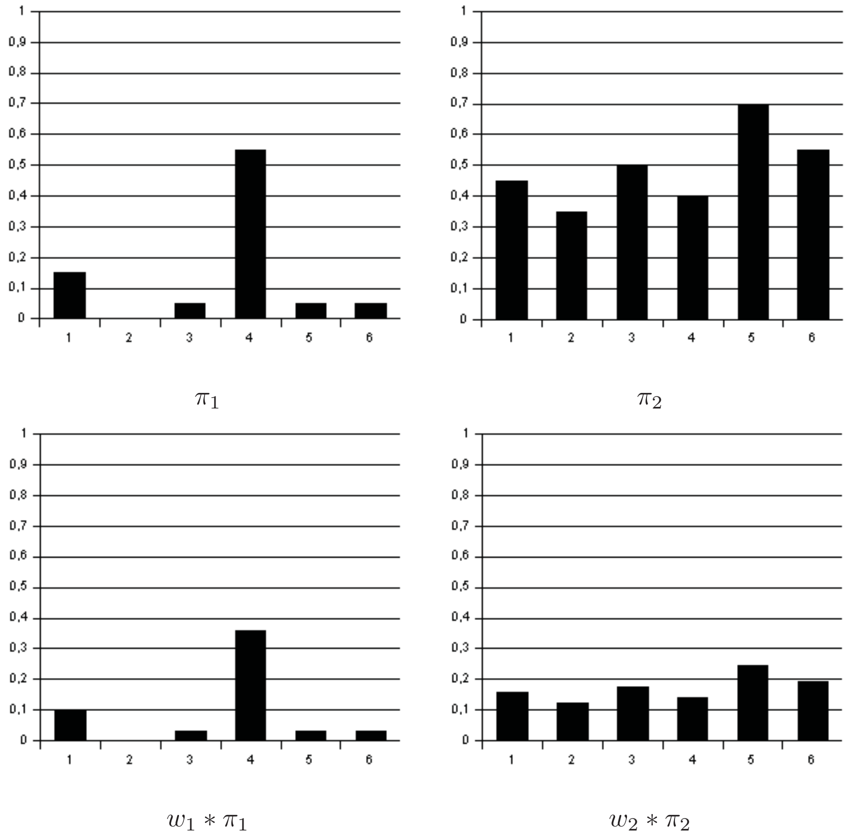
5.2.2 Confidence measurement

5.2.2.1 Point-wise accuracy

For a given pixel and a given classifier, we propose interpreting the degree of fuzziness of the fuzzy set $\pi_i(x)$ defined in (5.12) as a point-wise measure of the accuracy of the method. We intuitively consider that the classifier is *reliable* if one class has a high membership value while all the others have a membership value close to zero. Conversely, when no one membership value is significantly higher than the others, the classifier is *unreliable* and too much account should not be take of the results it provides in the final decision. In other words, uncertain results are obtained when the fuzzy set $\pi_i(x)$ has a high degree of fuzziness - the highest degree being reached for uniformly distributed membership values.

To reduce the influence of unreliable information and thus improve the relative weight of reliable information, we weight each fuzzy set by:

$$\begin{cases} w_i = \dfrac{\displaystyle\sum_{k=0,k\neq i}^{m} H_{\alpha QE}(\pi_k)}{\displaystyle(m-1)\sum_{k=0}^{m} H_{\alpha QE}(\pi_k)} \\ \displaystyle\sum_{i=0}^{m} w_i = 1 \end{cases} \tag{5.21}$$



π_1

π_2

$w_1 * \pi_1$

$w_2 * \pi_2$

Figure 5.5: *Normalization effects. This figure shows two fuzzy sets (π_1 and π_2) with different fuzziness ($H_{\alpha QE}(\pi_1) = 0.51$, $H_{\alpha QE}(\pi_2) = 0.97$, $w_1 = 0.65$ and $w_2 = 0.35$). The normalization effect is shown on the right. The influence of classifier 2 is reduced more by w_2 than classifier 1 is reduced by w_1.*

where $\alpha = 0.5$, $H_{\alpha QE}(\pi_k)$ is the fuzziness value of source k, and m is the number of sources. When a source has a low fuzziness value, w_i is close to 1 and it affects corresponding fuzzy set only slightly. Figure 5.5 illustrates the effects of this normalization.

5.2.2.2 Overall accuracy

Over and above the adaptation to the local context described in the previous paragraph, we can also use prior knowledge regarding the performance of each classifier. This knowledge is modeled for each classifier i and for each class j by a parameter f_i^j. Such overall accuracy can be determined by a separate statistical study on each of the classifiers used. If, for a given class j, the user considers that the results provided by classifier i are satisfactory, parameter f_i^j is set to one. Otherwise, it is set to zero. Since this decision is binary, we assume that for each class there is at least one method ensuring satisfactory global reliability.

5.2.3 Combination operator

Numerous combination rules have been proposed in the literature, from simple conjunctive or disjunctive rules, such as min or max operators, to more elaborate CD operators, such as those defined by (5.19) and (5.20) where the relative reliability of each source is used. However, using these operators sources always have the same hierarchy and the fusion scheme does not adapt to the local context. In [46], we propose the following extension:

$$\mu_f^j(x) = \max\Big(\min\big(w_i\mu_i^j(x), f_i^j(x)\big),\ i \in [1,m]\Big) \tag{5.22}$$

where f_i^j is the overall confidence of source i for class j, w_i is the normalization factor defined in (5.21), and μ_i^j is an element of the fuzzy set π_i defined in (5.12). This combination rule ensures that only reliable sources are taken into account for each class (pre-defined coefficients f_i^j), and that the fusion also automatically adapts to the local context by favoring the source that is locally the most reliable (weighting coefficients w_i).

5.3 The Fusion scheme

We present here the complete proposed fusion scheme. In the first step, each classifier is applied separately (but no decision is taken). In the second step, the results provided by the different algorithms are aggregated. The final decision is taken by selecting the class with the largest resulting membership value.

The fusion step is organized as follows:
For each pixel :

1. Separately build the fuzzy set $\pi_i(x) = \{\mu_i^1(x), \mu_i^2(x), \ldots, \mu_i^j(x), \ldots, \mu_i^n(x)\}$ for each classifier i, with n classes.
2. Compute the fuzziness value $H_{\alpha QE}(\pi_i)$ for each fuzzy set $\pi_i(x)$.
3. Normalize data using w_i defined in (5.21).
4. Apply operator (5.22).
5. Select the class corresponding to the highest resulting membership value.

The block diagram of the fusion process is given in Figure 5.6. Note from this that the range of the fuzzy sets is rescaled before the fusion step in order to combine data with the same range. This is achieved using the following range stretching algorithm:

- for all $\pi_i(x) = \{\mu_i^1(x), \ldots, \mu_i^j(x), \ldots, \mu_i^n(x)\}$, compute :
  - $M = \max\limits_{j,x}\left[\mu_i^j(x)\right]$,
  - $m = \min\limits_{j,x}\left[\mu_i^j(x)\right]$,
  - for all $\mu_i^j(x)$, compute :
    - $\mu_i^j(x) = \dfrac{\mu_i^j(x) - m}{M - m}$.

5.4 Experimental Results

In this section, we present the application of the proposed general fusion scheme to improving classification results using remote-sensing images from urban areas. The proposed approach was applied to two very high-resolution IKONOS panchromatic images from Reykjavik, Iceland. Six classes were considered in each case, namely: large buildings, houses, large roads, streets, open areas, and shadows. Each image consists of a single channel with 1m resolution.

Two classification algorithms were used: a conjugate gradient neural network [8] and a fuzzy classifier [27]. Both consist of two steps. The first step is feature extraction by morphological filters and the second step is the actual classification, using either a neural network or a fuzzy possibilistic model. The classification accuracies for the different classifiers were compared to determine the global confidence in the fusion process. The inputs to the fusion process were the posterior probabilities from the outputs of the neural network and the membership values for the fuzzy classifier. These inputs are displayed as images in Figure 5.7.

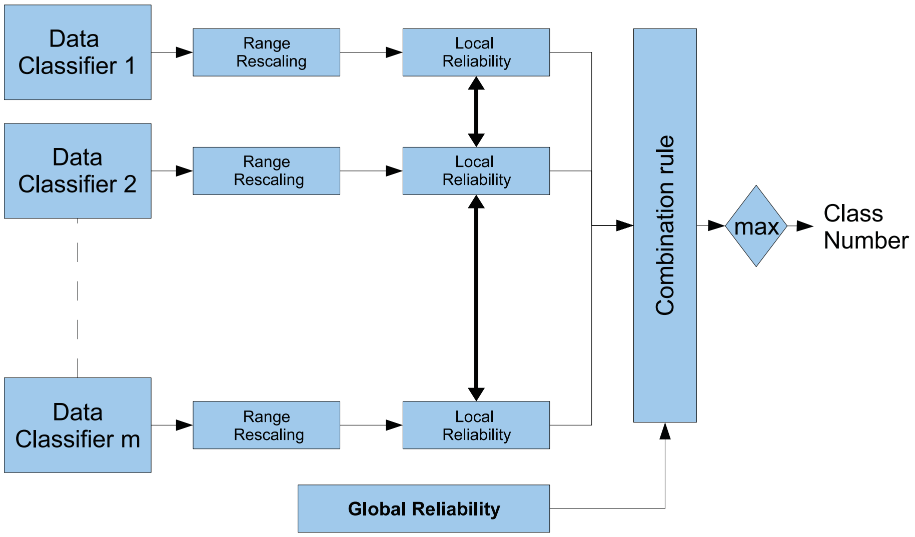


Figure 5.6: *Block diagram of the fusion method*

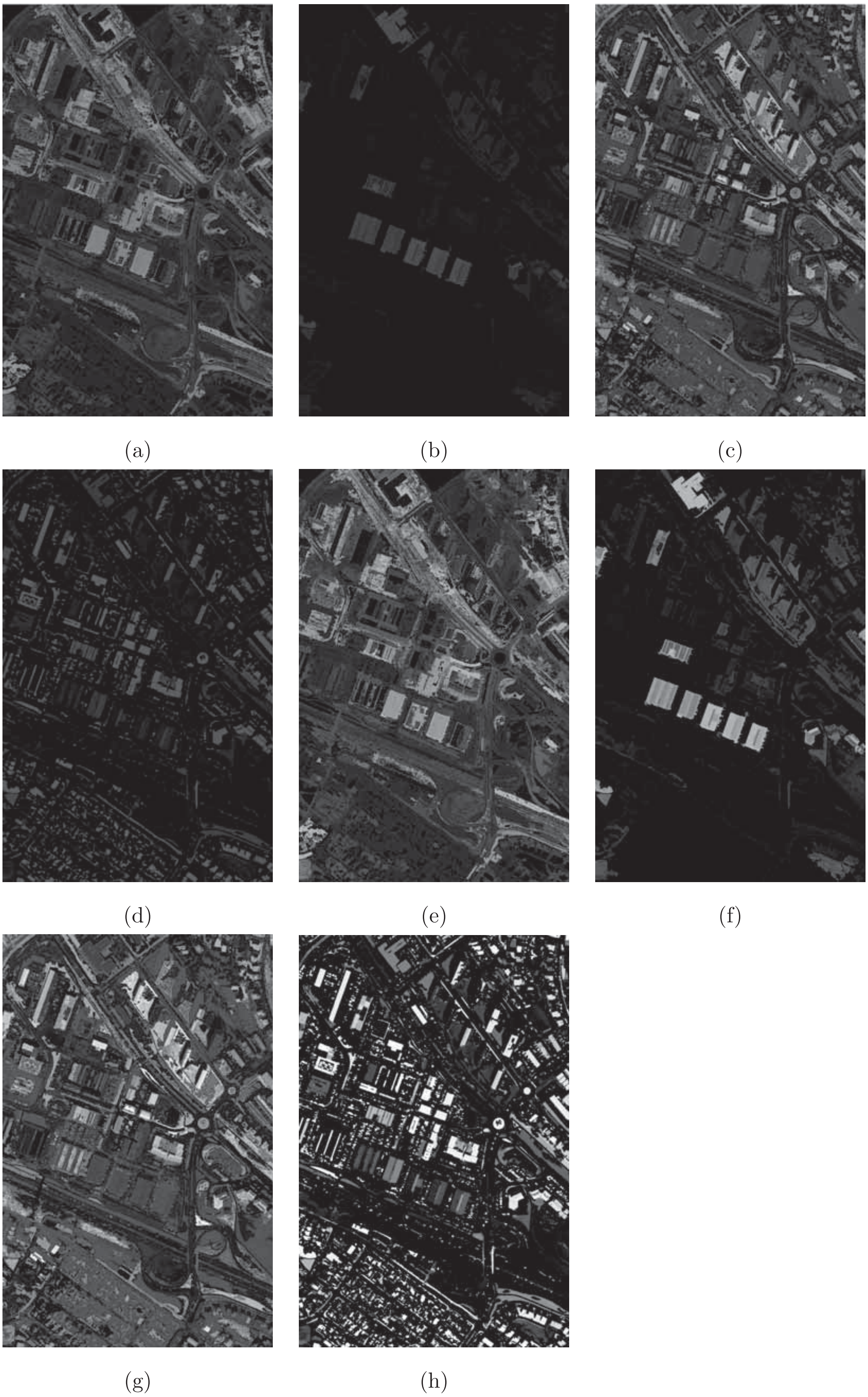
Table 5.2: *Confidence indices for image 1*

	Neural Network	Fuzzy Logic
Large Buildings	0	1
Houses	0	1
Large Roads	1	1
Streets	1	0
Open Areas	0	1
Shadows	0	1

5.4.1 First test image

The first test image (976× 640 pixels) is shown in Figure 5.8.a. Table 5.3 shows the test accuracies for the two classifiers. In order to test the generalization ability of the classifiers, independent samples were used for training and testing. See Appendix C for a full description of the training and testing sets. Starting from the class classification accuracies, the global reliabilities were set as follows: the neural network classifier gave higher accuracies than the fuzzy classifier for the classes 'streets' and 'large roads'. However, for the other four classes, the fuzzy classifier outperformed the neural network in terms of accuracies. In the fusion, we defined the indices of confidence in a binary manner according to the accuracies. For a given class, full confidence was given to the best classifier, *i.e.* the one with the highest classification accuracy. Then, if the accuracy of the other classifier was close to the highest (by 5%), full confidence was granted to that classifier too. Otherwise, the confidence index was set to zero. The confidence values are listed in Table 5.2.

The accuracy obtained for the final classification is given Table 5.3. The overall accuracy increased from 40.3% for the neural network and 52.1% for the fuzzy classifier to 59.1% using fusion. Small houses and larger buildings were classified similarly with the fuzzy classifier but the 'streets' classification accuracy improved from 9.8% to 55.7% with the use of the neural network information. The classification accuracies for shadows and open area also improved, from 83.3% and 52.2% respectively to 86.6% and 60.9% respectively. On the other hand, the classification accuracy for large roads reduced from 59.1% to 43.7%. Both the original and classified images are shown in Figure 5.8.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 5.7: *Possibility maps. (a) and (c) represent the membership maps given by the neural network respectively for the classes* buildings *and* houses*. (b) and (d) represent the membership maps given by the fuzzy classifier for the classes* buildings *and* houses*. (e), (f), (g), and (h) are the stretched versions of the four images above using the algorithm given in Section 5.3.*

Table 5.3: *Test accuracies in percentages for Image 1*

%	Neural Network	Fuzzy Logic	Fusion
Large Buildings	26.2	47.6	47.4
Houses	33.4	67.8	67.4
Large Roads	59.1	58.8	43.7
Streets	55.6	9.8	55.7
Open Areas	30.9	52.2	60.9
Shadows	32.7	83.3	86.6
OA	40.3	52.1	59.1
AA	39.7	53.3	60.3

Table 5.4: *Confidence indices for Image 2*

	Neural Network	Fuzzy Logic
Large Buildings	1	0
Houses	0	1
Large Roads	1	1
Streets	1	0
Open Areas	1	1
Shadows	0	1

The results of the first experiment illustrate the complementary behaviors of the fuzzy and neural network classifiers. Even though overall accuracy is higher with the fuzzy classifier, the neural classifier performs better in terms of accuracy for the classes 'large roads' and 'streets'. Note that these accuracy figures were obtained using manual ground truth where each pixel in the original image was labeled. Since no pre- or post-processing was done, the accuracies should be interpreted in a relative rather than an absolute way.

5.4.2 Second test image

The second test image is 700× 630 pixels. Table 5.5 shows the test accuracies for the two classifiers used in the second experiment. The global reliability was defined in the same way as in the first experiment. The confidence indices are listed in Table 5.4.

The test accuracies for the final classification are given in Table 5.5. As this shows, the overall accuracy increased from 57.0% for the neural network and 43.1% for fuzzy classifier to 75.7% after fusion. With fusion, classification accuracy for open areas increased from 46.5% to 73.7%. 'Streets' and 'Open Areas' classification accuracies were similar for the fuzzy classifier and the neural network. The biggest improvement after fusion was achieved in the classification of large roads, where the classification accuracy improved from 0.0% to 94.2%. Furthermore, the overall road ('Large roads' + 'Streets') classification accuracy increased from 41.5% to 58.6%. But at the same time, the classification accuracy for streets reduced from 83.6% to 22.7%. Both the original and classified images are shown in Figure 5.8.

Table 5.5: *Test accuracies in percentage for Image 2*

%	Neural Network	Fuzzy Logic	Fusion
Large Buildings	89.6	26.3	94.8
Houses	29.9	42.8	33.8
Large Roads	0	0	94.2
Streets	83.6	77.4	22.7
Open Areas	46.5	44.9	73.7
Shadows	43.7	98.7	90.4
OA	57.0	43.1	75.7
AA	48.9	48.4	68.3

5.4.3 Comparison with other combination rules

In this subsection we compare the results provided by the proposed operators with other combination rules. Where possible, we use the accuracy measurement previously defined in Section 5.2.2. For the min and max operators we compute experiments with and without point-wise accuracy information. We do the same for the operator (5.18). Conflict was computed for both cases. For operators (5.19) and (5.20), the less accurate classifier was chosen as the less accurate classifier based on the overall test accuracy.

The results obtained are given in Table 5.6 and Table 5.7. As can be seen from the tables, our proposed method outperformed the other combination rules in terms of accuracy. It can be seen that the classification accuracy for streets is still not satisfactory. None of the combination rules were able to use the information provided by the neural network.

For the max operator, the point-wise accuracy information improved the classification accuracy as compared to fusion using the max operator without point-wise accuracy information. This was due to the *normalization effect*: the unreliable information was reduced thanks to operator (5.21). Conversely, point-wise accuracy information impaired the classification using the min operator. Here, unreliable information was reduced by operator (5.21) and was unfortunately able to be selected. Adaptive operators (5.18) and (5.20) seemed to perform better with point-wise accuracy information, the overall accuracy for operators (5.18) and (5.20) increased from 36.7% and 39.5% to 42.9% and 42.5%, respectively. No significant changes were noted for the operator 5.19.

From these experiments, it can be concluded that if no information is available on source reliability, point-wise accuracy can be used to significantly improve the fusion. However, knowledge about the global reliability of each classifier seems to be more useful. Finally, to investigate the influence of the contextual information, two additional experiments were conducted. In each experiment, we removed one type of contextual information and compared the results in terms of classification accuracy to those obtained using both types of contextual information. For the global information, if we set its values to 1 for both classifiers and all classes, operator (5.22) becomes the simple max operator using point-wise accuracy information; this experiment was already performed in the previous paragraph. For the point-wise accuracy information, all w_i were set to 1 and only the global information was retained. Results are listed in Table 5.8. From these experiments, it is clear that both types of contextual information are needed to achieve good classification in terms of accuracy.

The results of these further experiments demonstrate the need to control the fusion process. Without information about conflict, accuracy, and confidence, accuracies are generally worse than before fusion. While the point-wise accuracy is easy to compute and is independent of the classifiers, global accuracy is a critical problem with this method. More development is needed to allow them to be defined automatically.

Table 5.6: *Test accuracies in percentages for different combination rules without point-wise accuracy measurement of Image 1*

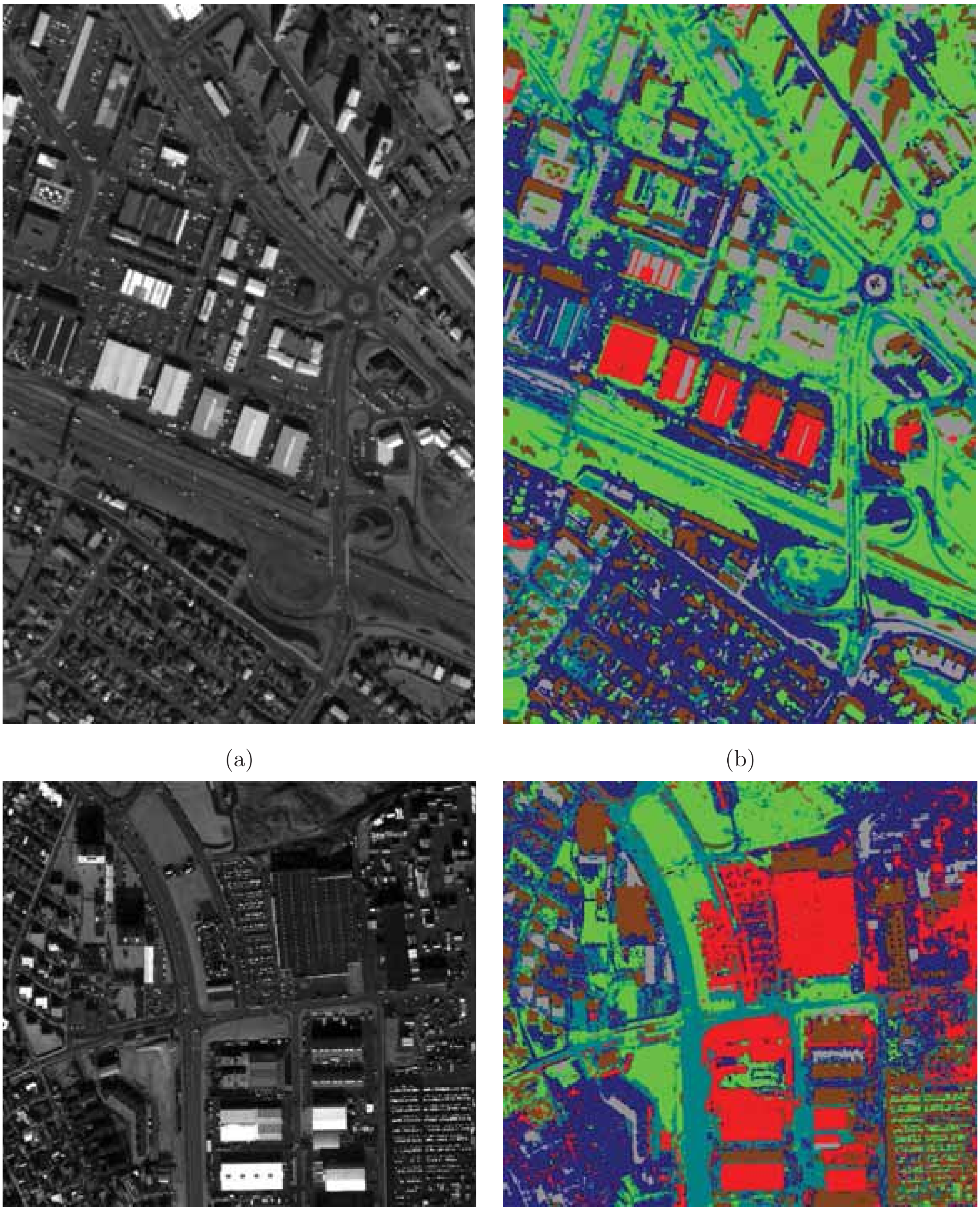
%	Max	Min	Operator (5.18)	Operator (5.19)	Operator (5.20)
Large Buildings	31.6	42.8	32.7	47.8	39.6
Houses	68.2	67.2	65.1	67.6	64.3
Large Roads	66.4	68.0	66.4	59.4	69.6
Streets	2.1	5.9	2.1	7.2	4.2
Open Areas	9.1	9.1	9.1	8.3	13.1
Shadows	52.8	81.1	52.8	84.4	53.5
OA	37.0	43.0	36.7	42.6	39.5
AA	38.4	45.7	38.0	46.1	40.7

Table 5.7: *Classification accuracies in percentages for different combination rules with point-wise accuracy measurement of Image 1*

%	Max	Min	Operator (5.18)	Operator (5.19)	Operator (5.20)
Large Buildings	48.4	40.8	48.4	47.8	47.6
Houses	70.2	55.6	70.2	67.8	67.3
Large Roads	59.7	71.6	59.7	59.4	59.7
Streets	6.1	7.2	6.1	7.2	6.8
Open Areas	8.1	9.7	8.1	8.3	8.4
Shadows	84.2	70.9	84.2	84.3	84.1
OA	42.9	40.5	42.9	42.7	42.5
AA	46.1	42.6	46.1	46.2	45.7

Table 5.8: *Classification accuracies in percentages for operator (5.22) with different types of contextual information*

%	Point-wise accuracy	Global accuracy	Both accuracies
Large Buildings	48.4	42.9	47.7
Houses	70.2	67.2	67.4
Large Roads	59.7	64.5	43.7
Streets	6.1	4.9	55.7
Open Areas	8.1	37.0	60.9
Shadows	84.2	92.8	86.6
OA	42.9	49.5	59.1
AA	46.1	51.5	60.3



(a) (b)

(c) (d)

Figure 5.8: *Test images and results; (a) Original IKONOS image 1, (b) image 1 classification results, (c) Original IKONOS image 2, (d) image 2 classification results.*

5.5 Application to the fusion of SVM classifiers

The previous framework was applied to a fuzzy logic classifier and a neural classifier. Both were using the same features vector, the morphological profile. When dealing with panchromatic data, MPs yield better results, whatever the specific classes considered, than the single raw band. However, this is not true when the original data set is hyperspectral data. In that case, the original spectral data was able to yield better results for some classes, while the EMP was able to perform better for other classes.

For data fusion, the best situation is where the classifiers have complementary results. In the following we proposed fusing the results obtained by separate use of the spectral data and the Extended Morphological Profile. Each data set was processed by SVM classifiers. SVMs were chosen because they are good at handling remote-sensing data, see Chapter 3 and 4. The results from each classifier are aggregated according to the inherent characteristics of the SVM outputs:

- outputs are not bounded,
- outputs are signed numbers.

Conventional fuzzy fusion operators such as T-norm, T-conorm, or symmetrical sum [14] are unable to handle signed data. For the fusion, we need to define operators that use the sign information. Note that algorithms do exist to provide probability output with SVMs [119]. But this involves adding an optimization step to the SVM training, and can be highly time-consuming. In this thesis, we suggest three different operators that can handle the standard SVM output. First, a modified version of the max operator, namely the *absolute maximum* decision rule, is applied. Secondly, classifier agreement is suggested. Agreement is seen as the probability of the outputs of each classifier. And thirdly, a rule based on majority voting, initially used for multi-class SVM, is investigated.

5.5.1 Decision Fusion

As explained in the previous chapter, the SVM decision function returns the sign of the distance to the hyperplane. For the fusion scheme, it is more useful to have access to the confidence of the classifier rather than the final decision [46], as detailed in the previous section. With SVMs, it is possible to get the distance from the hyperplane by a simple change in the decision functions. For a given sample, the greater the distance from the hyperplane, the more reliable the label. This is the basis of the *one-against-all* strategy [112]. For the combination process, we choose this distance to fuse. We consider that the most reliable source is the one that gives the greatest *absolute* distance.

In this approach, we first used the *absolute maximum* decision rule. For an m-source problem $\{S^1, S^2, \dots, S^m\}$ and for a pairwise classification mutli-class strategy, where $S^1_{ij} = d^1_{ij}$ is the distance provided by the first SVM classifier which separates class i from j, this decision rule is defined as follows:

$$S_f = AbsMax(S^1, \dots, S^m) \tag{5.23}$$

where $AbsMax$ is the set of logical rules:

$$\begin{aligned} \textbf{if}(\left|S^1\right| > \left|S^2\right|, \dots, |S^m|) \quad &\textbf{then} \quad S^1 \\ \textbf{else if}(\left|S^2\right| > \left|S^1\right|, \dots, |S^m|) \quad &\textbf{then} \quad S^2 \\ &\vdots \\ \textbf{else if}(|S^m| > \left|S^1\right|, \dots, \left|S^{m-1}\right|) \quad &\textbf{then} \quad S^m. \end{aligned} \tag{5.24}$$

The second operator considered takes classifier agreement into account. Each distance is multiplied by the maximum membership probability[2] associated with the two classes considered. Then the absolute maximum is used to fuse the results. The probabilities are simply computed by [131]:

$$p_i = \frac{2}{m(m-1)} \sum_{j=0, j \neq i}^{m} I(d_{ij}) \tag{5.25}$$

[2] It is the probability for a class to be selected at the end of the process.

where I is the indicator function $I(x) = 1$ if $x \geq 0$ else $I(x) = 0$. For the fusion, the absolute distance is used as in (5.23), where source S^k is weighted by the corresponding p^k:

$$S^f = AbsMax\left(\max(p_i^1, p_j^1)S^1, \ldots, \max(p_i^m, p_j^m)S^m\right). \tag{5.26}$$

The third operator is the one used to combine binary classifiers in the *one-versus-one* strategy. If we have two SVM classifiers, and apply each of them on a dataset with the same number p of classes, each classifier builds $p(p-1)/2$ binary classifiers, see Section 3.3 page 60, and uses majority voting. Thus, we propose constructing a new set of classifiers containing $p(p-1)$ classifiers. Then, we apply a conventional majority voting scheme.

Finally, fusion is performed as follows. First, for each classifier, we extract the distance from the hyperplane for each sample. Then the data are fused using one of the three operators. For the operators based on absolute maximum, majority voting is performed. For all the operators, the class having the highest number of votes is selected as the winner.

5.5.2 Experiment

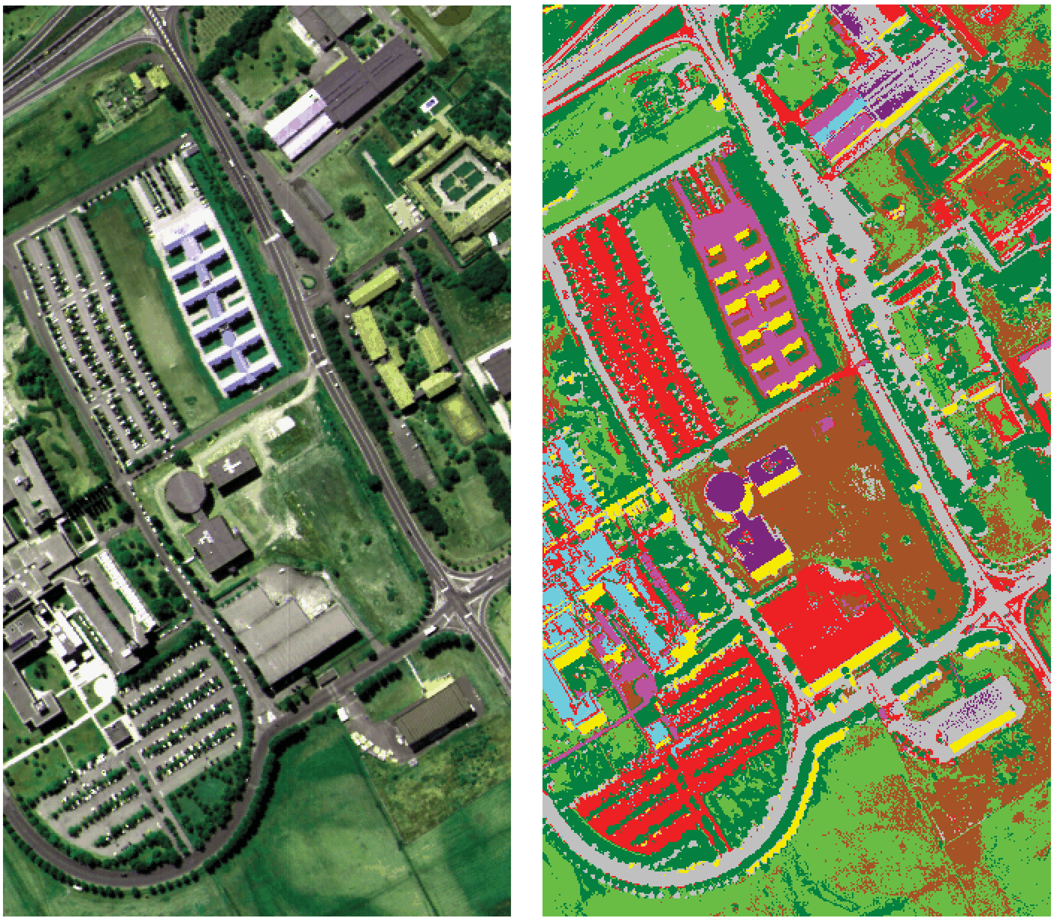
The proposed approach has been tested on real hyperspectral data. The University data set was used, see appendix C. The original image is shown in Fig. 5.9.(a). Three principal components were selected and the morphological profile was built using 10 openings/closings by reconstruction. The image was first classified using the spectral data (103 bands) and then using the EMP (63 bands). Gaussian kernels were used for each experiment. The parameters (C, γ) of the SVM were tuned using five-fold cross validation. The results were combined according to the classification scheme previously defined. The accuracies in terms of classification are listed in Table 5.9. The overall accuracy (OA) is the percentage of correctly classified pixels, whereas the average accuracy (AA) represents the average of the individual class accuracies. The coefficient Kappa is another criterion traditionally used in remote-sensing classification to measure the degree of agreement, and takes into account the correct classification that might have been obtained 'by chance' by weighting the measured accuracies. *Per Class* classification accuracy has also been reported. The classification map for the absolute maximum fusion operator is presented in Fig. 5.9.(b).

As can be seen from the table, the fusion step using absolute maximum improves classification accuracies. The highest overall accuracy, as well as the highest average accuracy and the highest Kappa value, were achieved when the absolute maximum and probability were used conjointly. By comparing the global accuracies (OO, OA, Kappa), it is clear that the use of probabilities does not help a great deal in the fusion process in these experiments. The use of the majority voting rule does not improve the results compared to those obtained with the EMP. Regarding the per class accuracies, it is interesting to note that the normalized absolute maximum rule provided the best per class results in *only* three cases. However, all the accuracies are over 82%, and the accuracy is close to the highest obtained accuracy for all classes. In terms of computing time, majority voting is the combination rule that leads to the shortest processing, while the absolute maximum approach requires slightly more time. Assessing the probabilities increases the computing time.

It is important to highlight the complementarity of the two initial results. From the table, the classifiers have complementary ability for class 1, 2, 3 and 7. After the fusion, the results have improved significantly for these classes: during the fusion step, the reliable information has been selected.

Table 5.9: *Classification accuracies in percentages for SVM classification using the spectral data, the EMP, and for the three fusion operators.*

	Spect.	PCA+EMP	Abs. Max.	A.M.+Prob.	Maj. Vot.
OA	80.99	85.22	89.56	**89.65**	86.07
AA	88.28	90.76	93.61	**93.70**	88.49
Kappa	76.16	80.86	86.57	**86.68**	81.77
Class 1	83.71	**95.36**	93.18	93.02	93.98
Class 2	70.25	80.33	83.89	83.96	**85.34**
Class 3	70.32	**87.61**	82.13	82.23	64.94
Class 4	97.81	98.37	**99.67**	**99.67**	**99.67**
Class 5	99.41	**99.48**	**99.48**	99.41	**99.48**
Class 6	**92.25**	63.72	91.21	91.83	61.55
Class 7	91.58	**98.87**	96.99	97.22	93.01
Class 8	92.59	95.41	96.39	96.41	**98.83**
Class 9	96.62	97.68	**99.58**	**99.58**	**99.58**



(a) (b)

Figure 5.9: *Rosis University Area. (a) false color original image, (b) classification map after fusion*

5.6 Conclusion

The fusion of several classifiers has been considered in the classification of panchromatic remote-sensing data from urban areas. Considering first the general situation where two classifiers are classifying the same data set, we suggest a complementary use of different classifiers. The proposed method is based on a fuzzy combination rule. Two measures of accuracy are used in the combination rule: the first, based on prior knowledge, defines global reliabilities, both for each classifier and for each class. The second automatically estimates the point-wise reliability of the results provided by each classifier, and thus makes it possible for the fusion rule to be adapted to the local context. The proposed approach does not require any training and takes only about 1 minute of computation time for each image using a Pentium 4 PC. Furthermore, no prior assumptions are needed in terms of data modeling (e.g. Bayes theory, possibility theory, *etc.*) prior to data fusion. The experimental results obtained show that the complementary use of different classifiers leads to a significant improvement in global classification accuracies. The overall accuracy was improved by about 7% in the first experiment and 18% in the second experiment. Another application of such methodology can be found in [49]. A key feature of the framework presented is its generality for *decision level fusion*. Though only two classifiers were used in the chapter, additional algorithms could easily be added to the process. For instance, specialized algorithms such as street detectors could be employed, without increasing errors in building detection. This generalization also holds good for the inclusion of multi-source data such as multi-spectral or multi-temporal images. One algorithm could be used on each image and fusion could then be performed using the results computed on each image. In this approach, α-Quadratic entropy was chosen for the fuzziness evaluation because the sensitivity of that measurement can be modified with the value of α. Several other measurements could be used, e.g. *fuzzy entropy* [40]. One limitation of the proposed approach is the use of binary values for the global confidence. With fuzzy confidence, the combination rule could be rewritten with *T-conorm* and *T-norm*, which are less indulgent and less severe than *max* and *min* respectively. Moreover, the use of the *T-conorm* and *T-norm* would make a finer definition of global accuracy possible.

A second methodology has then been presented: Decision fusion for an SVM classifier. In this case, the input features were complementary and were classified using the same algorithm. Three operators based on the main characteristics of the SVM outputs were proposed. The operators were based on the assumption that the absolute distance from a hyperplane gives meaningful information about classifier agreement. In experiments, the proposed approach outperformed each of the individual classifiers in terms of overall accuracies. The use of the absolute maximum operator led to a significant improvement in terms of classification accuracy. It is noteworthy that other operators are able to use sign as an informative feature. The classic *mean* or MYCIN rules [14] are examples of possible operators. Unfortunately, for a two-source problem, such operators have the same influence on the sign of the fused data as the absolute maximum. Thus in our case majority voting led to the same results. In this experiment, only one type of kernel was used. One possible extension of the proposed method would be to include other sources using different kernels. Polynomial kernels, which are known to perform well on complex data, could be investigated. The good performance of the proposed combination scheme is interesting because it does not use information about the global reliability of the source. One topic for future research is the use of a more advanced fusion scheme that takes into account the performance of the classifiers. Another field for investigation is the use of upper error bound, such as the radius margin bound, to estimate the global reliability of each binary SVM classifier.

Chapter 6

Multisource data

Abstract

Multi-source data for the classification of remote-sensing images is addressed in this chapter. Multi-source data are seen as the aggregation of several information sources from the same location, e.g. *spectral data and spatial/textural data. This approach is proposed to overcome traditional approaches that are mainly based on either spectral data or spatial data alone. In this chapter, spectral features are extracted from the hyper-spectral data by an appropriate Feature Extraction algorithm and spatial data are extracted using the Extended Morphological Profile, still with a Feature Extraction algorithm. Then a stacked vector is built. Classification is performed using an SVM classifier and the stacked vector is used as an input vector. Experiments are conducted on two hyper-spectral data sets and the results obtained confirm the usefulness of such an approach.*

Contents

THE EXTENDED morphological profile EMP has been primarily designed for classification of urban structures, and it does not fully utilize the spectral information in the data. As has been concluded in the preceding chapters, the use of spectral information can be critical for classification of non-structured information in urban areas, *e.g.* vegetation or soil classes. In Chapter 4 we have proposed another methodology for improved extraction of spatial information from non-structured classes. However, its performances for structured classes was slightly worse than those obtained using the EMP.

In this chapter, we propose adopting a data fusion strategy to overcome the shortcoming of the EMP. It is based on *multi-source data* classification [6, 63]: The combination of the spectral data and the EMP by means of a stack vector.

Multi-source data may be regarded as part of data fusion theory, since data with different characteristics are combined. With the development of remote sensors, such data are readily accessible: radar data, optical data, and elevation data of a given area can be used together to improve classification accuracy. A great deal of work has been done within the remote-sensing community and has offered enhanced capabilities for classifying target surfaces. In [12, 13], Benediktsson *et al.* have successfully used multi-source data comprising Landsat MSS and ancillary topographic data such as elevation, slope and aspect for classification. For the particular situation of *forest classification*, it was demonstrated that significant improvements can be made by using multi-source approaches [73, 129]. For the above papers, classification was performed using a neural network, since no accurate statistical model was available. A statistical multi-source classifier was developed by Solberg *et al.* to classify optical data from Landsat TM and multi-temporal SAR data from ERS-1 [123]. The fusion algorithm was based on the model described by Benediktsson and Swain [11]. More recently, SPOT and elevation data were used as an input to a Generalized Positive Boolean Function classifier for landslide classification. Experimental results have shown improvement when using fused input [25]. The problem under consideration here concerns the fusion of information extracted from the original data, *i.e.* operations are performed to extract useful information from a single data set. For our application, we are only considering one data set, whether it is hyperspectral or not, and the additional sources are obtained by some form of spatial processing, *e.g.* morphological profile.

When classifying multi-source data, it was preferable to use neural classifiers, since statistical approaches lack an accurate model of the data. In the previous chapter, the superiority of SVM, implementing Structural Risk Minimization, over the neural classifier, implementing Empirical Risk Minimization, has been detailed. Hence in the following experiments the SVM with a Gaussian kernel was used, see Chapter 3 for details of the algorithm. Note that SVM has already been applied for multi-source classification in [63], where several different formats for coding the output were investigated. Here once again, the best results were obtained using the one-against-all and one-against-one multi-class strategies.

The remainder of this paper is organized as follows. The features used are described and the proposed fusion scheme is detailed. The studies of their characteristics suggested the use of some feature-reduction algorithms. Some standard algorithms are briefly recalled. Then experiments are presented.

6.1 Spatial and spectral feature extraction

6.1.1 Spectral and spatial data

Multi-band data, especially hyperspectral data, contain a great deal of information about the spectral properties and the land cover of the data. Tighter definition of the classes is possible for a greater number of classes. Based on the spectral signatures of the classes, many advanced pixel-based classifiers have been proposed: SVM, Neural Network, and so on. However, these did not use the spatial content of the image, and the resulting thematic maps sometimes look *noisy* (salt and pepper classification noise). Approaches involving Markov Random Field (MRF) and Monte Carlo optimization have been proposed. These used contextual information. The main drawback of this type of algorithm is the *computing time*, which can be extended even for small data sets. In view of the high dimensionality of recent acquired data, in both the spectral and spatial domains, computationally light algorithms are of interest. The Morphological Profile

Table 6.1: *Properties of spectral and spatial data*

Hyperspectral data		EMP	
$\nearrow$	Fine physical description	$\nearrow$	Geometrical information
$\nearrow$	Directly accessible	$\sim$	Needs to be extracted
$\sim$	Redundancy	$\sim$	Redundancy
$\searrow$	No spatial information	$\searrow$	Reduced spectral information

(see Chapter 2) has been proposed as an alternative way of exploiting spatial information. Compared to the MRF-based classifiers, the MP and its extension into multi-valued image EMP offers the possibility of making use of geometrical information (shape, size, etc.) and performs well on many types of data (panchromatic, multi-spectral, and hyper-spectral data). However, as stated in the introduction, one shortcoming of this approach is that it fails to make full use of the spectral information in the data, and consequently several approaches based on the MP/EMP have been proposed in order to fully exploit the spatial and spectral information [1, 52, 100].

Table 6.1 sums up the properties of these types of data. The first main consideration is the complementary nature of the data. As will be shown in the experiments, this has an incidence on the discrimination abilities of such data. The fusion of two types of information should clearly result in an improvement in classification accuracy.

The second consideration is the potential redundancy in each feature set - see [80] for spectral features and [7] for spatial features. Hence feature extraction (FE) algorithms may be of interest. Note that we are not looking for improved classification accuracy, but rather increased processing speed, since SVM performs well with high-dimensional data. In this work, we investigate two commonly used FE algorithms: the Decision Boundary Feature Extraction (DBFE) and the Non-parametric Weighted Feature Extraction (NWFE) [80]. These algorithms will be described briefly in the following subsection.

6.1.2 Fusion scheme

The method proposed is based on data fusion of the morphological information and the original data: first, an extended morphological profile is created based on the PCs from the hyperspectral data. Secondly, feature extraction is applied on the morphological data and the original hyperspectral data. Finally, the extended morphological profile after feature extraction and the feature-extracted vector from the original data are concatenated into one stacked vector and then classified by the SVM.

Figure 6.1 illustrates the data fusion scheme when using DBFE as the feature-reduction algorithm. Note that in this work we have only been extracting morphological information, but it is equally possible to use other processing to extract other types of spatial information and include this into the stacked vector.

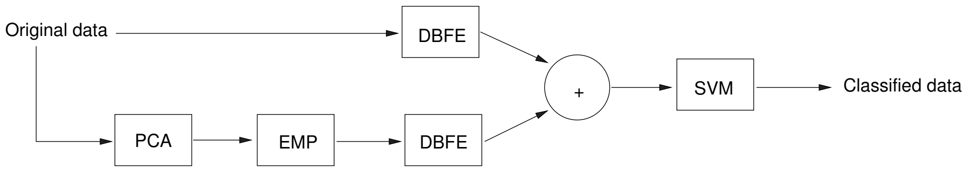


Figure 6.1: *Proposed data fusion scheme.*

6.1.3 Feature reduction

In this section, we briefly recapitulate the feature-extraction algorithms used in our experiments. DBFE has already been described in greater detail in Chapter 1, page 25.

6.1.3.1 DBFE

It was shown in [83] that both discriminantly informative features and redundant features can be extracted from the decision boundary between two classes. The features are extracted from the decision boundary feature matrix (DBFM). The eigenvectors of the DBFM corresponding to non-zero eigenvalues are the feature vectors necessary to achieve the same classification accuracy as in the original space. The efficiency of the DBFE is related to the training set and it can be computation-intensive.

6.1.3.2 NWFE

To overcome the limitations of the DBFE, Kuo and Landgrebe [78] proposed non-parametric weighted feature extraction. NWFE is based on Discriminant Analysis Feature Extraction, by focusing on samples near the eventual decision boundary. The main ideas of NWFE are assigning different weightings to each sample in order to compute local means, and defining non-parametric 'between' and 'within' class scatter matrix [80].

Many experiments have shown the effectiveness of these approaches in the classification of hyperspectral data [80]. They are usually applied to the spectral data, but Benediktsson and co-workers have successfully applied them to the EMP [7].

6.2 Experiments

6.2.1 Data set

For these experiments, we used the two ROSIS data sets (see Appendix C). For each data set, we compare the classification accuracies obtained using either spectral features, morphological features, or the fused features. We also investigate the influence of feature reduction.

The classification accuracy was assessed using overall accuracy (OA), which is the number of correctly-classified samples divided by the number of test samples; average accuracy (AA), which represents the average of the class classification accuracies; and the coefficient of agreement (κ), which is the percentage agreement corrected by the degree of agreement that might be expected due to chance alone. These criteria were computed using the confusion matrix and were used to compare classification results.

Feature extraction was performed using MultiSpec [80] while the morphological operations were performed using Matlab and Image Toolbox. Classification was performed using the LIBSVM through its Matlab interface [24]. The one-vs.-one multi-class approach was used in these experiments.

6.2.2 University Area data set

Classification was performed independently using the spectral information or the extended morphological profile. The confusion matrices are reported in Tables 6.2 and 6.3. In terms of global accuracy, both approaches perform equally well, with the spectral approach showing a slight advantage. Note that this is consistent with the characteristics of the scene: the University Area is a mixture of some man-made structures and natural materials. So morphological information is not as useful as it might be in a very dense urban area. Examining the class-specific accuracy closely, we can see from the tables that the classes for which each approach performs well are complementary, *e.g.* the spectral approach performs better for classes 3, 6, 9 while the EMP approach performs better for classes 1, 2, 7, 8. So we need to look at these classes after fusion to see if the best information was used, *i.e.* if classification accuracy improves.

Table 6.2: *Confusion Matrix for classification of the* ***University Area*** *data set using the* ***original hyperspectral data*** *(103 bands). Global accuracies:* ***OA*** $= 79.48\%$***,*** ***AA*** $= 88.14\%$***, and*** $\boldsymbol{\kappa} = 74.47\%$*.*

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	5594	26	110	17	14	13	359	498	0	84.36
2	0	12346	0	2088	0	4181	0	34	0	66.20
3	27	7	1511	0	0	3	2	549	0	71.99
4	0	24	0	3003	10	27	0	0	0	98.01
5	0	0	3	1	1338	0	0	0	3	99.48
6	13	163	1	48	104	4683	0	17	0	93.12
7	103	0	0	1	0	0	1213	13	0	91.20
8	40	10	205	4	0	19	7	3397	0	92.25
9	21	0	11	0	0	0	0	0	915	96.62

Table 6.3: *Confusion Matrix for classification of the* ***University Area*** *data set using the* ***EMP*** *(27 bands). Global accuracies:* ***OA*** $= 79.14\%$***,*** ***AA*** $= 84.30\%$***, and*** $\boldsymbol{\kappa} = 73.25\%$*.*

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	6266	8	8	146	0	1	33	169	0	94.49
2	55	13581	0	1869	0	3143	0	1	0	72.82
3	3	3	1117	8	0	0	4	964	0	53.21
4	5	17	0	3030	0	11	0	0	1	98.89
5	0	0	0	1	1339	0	0	0	5	99.55
6	41	1920	1	99	82	2872	0	14	0	58.10
7	44	0	5	1	0	0	1278	2	0	96.09
8	8	9	148	9	0	0	0	3508	0	95.27
9	0	0	83	0	0	0	0	0	864	91.23

We performed the experiment using the concatenated vector. The vector was constructed from the 103 spectral bands and the 27 features from the EMP. The vector was used directly as an input to the SVM. The classification results are reported in Table 6.4. The global accuracy has improved. The κ is 79.13%, as against 74.47% for the spectral approach and 73.25% for the EMP. In terms of the class-specific accuracy, the classification accuracies for classes 1, 7, 8 have improved compared to both individual approaches, and all of the classes are better classified than the worst case of the individual approaches.

Feature reduction was applied to the morphological data and original data before concatenation. Then the stacked vector was classified by the SVM. The κ is plotted on Fig. 6.2 using several values for the DBFE and NWFE variance criterion. Best results are obtained with 95% and 80% of the variance criteria for DBFE and NWFE respectively. Using 95% of the variance criteria with DBFE, the hyperspectral data is reduced to 27 features and the EMP to 10 features. With NWFE and 80%, 7 features were extracted from the hyperspectral data and 6 from the EMP. The confusion matrices are reported in

Table 6.4: *Confusion Matrix for classification of the* ***University Area*** *data set using the* ***original hyperspectral data*** *and the* ***EMP*** *(130 bands). Global accuracy:* $\boldsymbol{OA} = 83.53\%$***,*** $\boldsymbol{AA} = 89.39\%$***, and*** $\boldsymbol{\kappa} = 79,13\%$*.*

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	6321	0	30	61	0	1	36	181	1	95.32
2	6	13699	0	528	0	4416	0	0	0	73.45
3	2	3	1383	6	0	0	3	702	0	65.88
4	1	18	0	3039	0	0	0	0	6	99.18
5	0	0	0	1	1338	0	0	1	5	99.47
6	397	304	4	30	56	4232	0	6	0	84.15
7	34	0	1	0	0	0	1293	2	0	97.21
8	9	7	122	1	0	4	0	3539	0	96.11
9	1	0	59	0	0	0	0	0	887	93.66

Table 6.5: *Confusion Matrix for classification of the* ***University Area*** *data set using* ***DBFE features*** *corresponding to* 95% *of the variance (37 bands). Global accuracy:* $\boldsymbol{OA} = 87.97\%$***,*** $\boldsymbol{AA} = 89.43\%$***, and*** $\boldsymbol{\kappa} = 84.40\%$*.*

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	6029	8	50	96	0	4	224	220	0	90.92
2	1	16021	0	830	0	1797	0	0	0	85.90
3	34	0	1215	7	0	0	3	840	0	57.88
4	4	8	0	3040	0	12	0	0	0	99.21
5	0	0	0	0	1338	0	0	1	6	99.47
6	3	595	0	123	13	4291	0	4	0	85.32
7	57	0	3	2	0	0	1266	2	0	95.18
8	12	5	124	8	0	4	0	3529	0	95.84
9	1	0	45	0	0	0	0	0	901	95.14

Tables 6.5 and 6.6. Classification maps for the different approaches are shown in Fig. 6.3.

In terms of the class-specific accuracy, the DBFE approach improves the classification of class 2, while class 3 is classified worse than with full hyperspectral data and EMP. However, DBFE outperforms the independent classification using the spatial or spectral information. In this test, it yields the best classification results. Similar comments may be made for the results obtained with NWFE. However, the number of features needed to achieve the same accuracy is significantly lower for the NWFE approach. Since the SVM is linearly related to the dimensionality of the data, lower-dimensional data increased the speed of the training process. Furthermore, if we were using a statistical classifier, *e.g.* a Gaussian maximum likelihood classifier, the benefit of such a reduction in the dimensionality could be more significant.

The different results for the University Area data are summarized in Table 6.7.

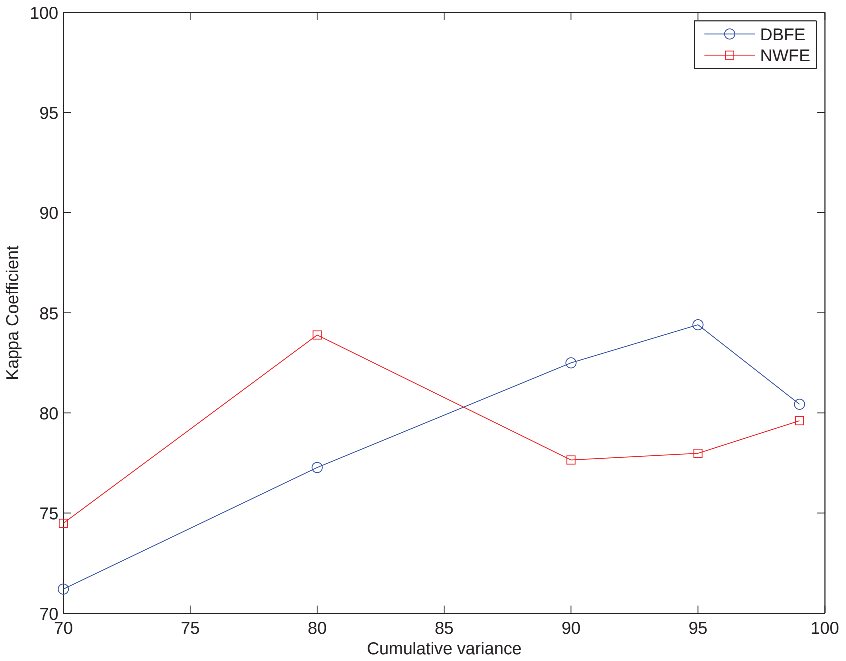


Figure 6.2: *κ values for several cumulative variance values for the DBFE and the NWFE applied to the University Area data set.*

Table 6.6: *Confusion Matrix for classification of the **University Area** data set using **NWFE features** corresponding to 80% of the variance (13 bands). Global accuracies: **OA** = 87.59%, **AA** = 88.93%, **and κ** = 83.89%.*

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	5756	6	214	76	1	4	229	345	0	86.80
2	20	16217	0	880	0	1529	0	3	0	86.95
3	11	7	1328	8	0	0	6	739	0	63.26
4	0	28	0	3019	1	16	0	0	0	98.53
5	0	0	0	0	1339	0	0	1	5	99.88
6	88	614	1	78	80	4155	0	13	0	82.62
7	23	0	12	0	0	0	1285	10	0	96.61
8	3	7	140	15	0	4	1	3512	0	95.38
9	6	0	83	0	0	0	0	0	858	90.60

6.2.3 Pavia Center data set

For the second test, the scene is a very dense urban area. Morphological information should be useful for discrimination here. SVM classification was applied to the original hyperspectral data and the EMP. The confusion matrices are given in Tables 6.8 and 6.9. It can be seen that the SVM classifier achieved excellent global accuracy. The morphological-based classification shows a slight advantage, as class 4 is classified better. The other classes are equally well classified. Thus the data fusion needs to improve the classification of that class, while maintaining very good results for the others classes.

Table 6.7: ***University Area***. *Summary of the class-specific test accuracies in percentages for SVM classification.*

	Spectral	DMP	Spec. DMP	DBFE 95%	NWFE 80%
OA	79.48	79.14	83.53	87.97	87.59
AA	88.14	84.30	89.39	89.43	88.93
κ	74.47	73.25	79.13	84.40	83.89
Class 1	84.36	94.50	95.33	90.92	86.80
Class 2	66.20	72.82	73.46	85.91	86.95
Class 3	71.99	53.22	65.89	57.88	63.26
Class 4	98.01	98.89	99.18	99.22	98.53
Class 5	99.48	99.55	99.48	99.48	99.88
Class 6	93.12	58.11	84.15	85.32	82.62
Class 7	91.20	96.09	97.22	95.19	96.61
Class 8	92.26	95.27	96.12	95.84	95.38
Class 9	96.62	91.24	93.66	95.14	90.60

Table 6.8: *Confusion Matrix for classification of the* ***Pavia Center*** *data set using the* ***original hyperspectral*** *data (102 bands). Global accuracy:* ***OA*** $= 97.67\%$***, AA*** $= 95.60\%$***, and*** $\boldsymbol{\kappa} = 96.71\%$.

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	64880	0	0	0	0	481	587	23	0	98.30
2	0	6932	665	0	0	0	0	1	0	91.23
3	0	65	2990	0	35	0	0	0	0	96.76
4	0	0	0	2375	267	7	31	5	0	88.45
5	0	2	19	282	6187	0	90	4	0	96.97
6	0	0	0	53	2	8909	231	54	0	96.32
7	0	0	0	119	7	140	6996	25	0	96.00
8	1	0	0	143	19	60	32	42569	2	99.39
9	1	1	0	0	0	0	0	0	2861	99.93

We performed the experiment using the concatenated vector. The vector was constructed from the 102 spectral bands and the 27 features from the EMP. The vector was used directly as an input to the SVM. The classification results are reported in Table 6.10. Almost every individual class-specific accuracy has improved, especially class 4. This results in a slight improvement in the global accuracy. However, the results are already very high, and so the differences in terms of classification accuracy are less statistically significant than with the University Area data set.

Feature reduction was applied on the morphological data and original data before concatenation. Then the stacked vector was classified by the SVM. The κ is plotted on Fig. 6.4 using several values for the DBFE and NWFE variance criterion. Best results are obtained with 99% variance criterion for both DBFE and NWFE. Using 99% of the variance with DBFE, the hyperspectral data is reduced to

Table 6.9: *Confusion Matrix for classification of the* ***Pavia Center*** *data set using the* ***EMP*** *(27 bands). Global accuracy:* ***OA*** = 98.69%*,* ***AA*** = 97.69%*,* ***and*** $\boldsymbol{\kappa}$ = 98.15%.

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	65366	0	0	0	0	494	111	0	0	99.08
2	0	6961	635	0	0	0	0	2	0	91.61
3	0	82	2972	0	36	0	0	0	0	96.18
4	0	0	0	2642	34	0	6	3	0	98.39
5	0	0	6	11	6511	0	18	38	0	98.89
6	0	0	0	5	3	9061	149	30	0	97.97
7	0	0	0	12	6	135	7133	1	0	97.88
8	0	0	0	33	31	28	18	42715	1	99.74
9	0	0	0	0	0	0	0	16	2847	99.44

Table 6.10: *Confusion Matrix for classification of the* ***Pavia Center*** *data set using the* ***original hyperspectral data*** *and the* ***EMP*** *(129 bands). Global accuracy:* ***OA*** = 99.69%*,* ***AA*** = 98.07%*,* ***and*** $\boldsymbol{\kappa}$ = 98.15%.

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	65089	0	0	0	0	881	1	0	0	98.66
2	0	7106	479	0	1	0	0	12	0	93.52
3	0	8	2965	0	36	0	0	0	0	95.95
4	0	0	0	2652	25	0	5	3	0	98.77
5	0	1	6	9	6546	0	18	4	0	99.42
6	0	0	0	8	1	9096	129	14	0	98.35
7	0	0	0	7	1	119	7157	3	0	98.21
8	0	0	0	12	28	33	15	42738	0	99.79
9	2	0	0	0	0	0	0	0	2861	99.93

51 features and the EMP to 15 features. With NWFE and 99% of the variance criterion, 44 features were extracted from the hyperspectral data and 20 from the EMP. The confusion matrices are reported in Tables 6.11 and 6.12.

Neither of the feature extraction algorithm (DBFE or NWFE) leads to better results. But the same classification accuracy is achieved with many less features - nearly half - thus reducing the total training and classification time. In this experiment, NWFE provides the best results when almost all the cumulative variance is used, the reverse of the situation in the previous experiment. Classification maps for the different approaches are shown in Fig. 6.5. Visually, the thematic map produced with the NWFE features seems less noisy than with the DBFE features, especially in the top left-hand corner, which represents a particularly dense urban area.

The different results for the Pavia Center data are summarized in Table 6.13.

Table 6.11: *Confusion Matrix for classification of the* ***Pavia Center*** *data set using* ***DBFE features*** *corresponding to* 99% *of the variance (66 bands). Global accuracy:* $\boldsymbol{OA} = 98.65\%$***,*** $\boldsymbol{AA} = 97.30\%$***, and*** $\boldsymbol{\kappa} = 98.19\%$.

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	65424	24	10	0	0	48	370	95	0	99.17
2	0	6838	750	0	0	0	0	10	0	88.99
3	0	84	2983	0	23	0	0	0	0	96.53
4	0	1	0	2656	19	0	7	2	0	98.91
5	0	5	1	8	6536	0	12	22	0	99.27
6	0	0	1	4	0	9105	131	7	0	98.45
7	0	0	0	13	0	138	7135	1	0	97.91
8	0	2	0	10	39	16	13	42744	2	99.80
9	0	100	0	0	0	0	0	26	2737	98.59

Table 6.12: *Confusion Matrix for classification of the* ***Pavia Center*** *data set using* ***NWFE features*** *corresponding to* 99% *of the variance (64 bands). Global accuracy:* $\boldsymbol{OA} = 98.87\%$***,*** $\boldsymbol{AA} = 97.95\%$ ***and*** $\boldsymbol{\kappa} = 98.41\%$.

Ref.	Classification Data									
Classes	1	2	3	4	5	6	7	8	9	Prod. acc.
1	65449	0	0	0	0	522	0	0	0	99.20
2	0	7027	570	0	0	0	0	1	0	92.48
3	0	93	2990	0	7	0	0	0	0	96.76
4	0	0	0	2673	3	0	8	1	0	99.55
5	0	0	1	9	6567	1	3	3	0	99.74
6	0	0	1	10	2	9128	107	0	0	98.70
7	0	0	0	8	0	108	7171	0	0	98.40
8	0	0	0	14	44	63	0	42705	0	99.71
9	67	0	0	0	0	0	0	21	2775	96.92

6.2.4 Comparison with spectro-spatial SVM

In this section, we draw comparisons with the results obtained using the spectro-spatial SVM approach. The classification accuracies are summarized in Table 6.14. The multi-source approach performs better in terms of classification accuracy with both data sets. For the University Area data set, best class-specific accuracies were obtained with the multi-source data, expect for two classes (3 and 6). In these cases, it seems that the morphological information perturbs the classification process; as a result, the AA is higher for the SS SVM. This phenomenon does not occur with the Pavia Center data set.

Regarding the thematic maps obtained with both approaches, the one produced using the SS SVM is less noisy than the one using the multi-source approach.

Note that the slight difference between the accuracies obtained in this chapter and in Chapter 4 are

Table 6.13: ***Pavia Center***. *Summary of the class-specific test accuracies in percentages for SVM classification.*

	Spectral	DMP	Spec. DMP	DBFE 99%	NWFE 99%
OA	97.67	98.69	99.69	98.65	98.87
AA	95.60	97.69	98.07	97.30	97.95
κ	96.71	98.15	98.15	98.10	98.41
Class 1	98.35	99.08	98.66	99.17	99.21
Class 2	91.23	91.62	93.52	90.00	92.49
Class 3	96.76	96.18	95.95	96.54	96.76
Class 4	88.45	98.40	98.77	98.92	99.55
Class 5	96.97	98.81	99.42	99.27	99.74
Class 6	96.32	97.98	98.36	98.45	98.70
Class 7	96.01	97.89	98.22	97.91	98.41
Class 8	99.40	99.74	99.79	99.81	99.72
Class 9	99.93	99.44	99.93	98.60	96.93

due to the multi-class strategies.

6.3 Conclusion

From the previous experiments, it is not clear which of the feature extraction methods should be used for the fusion of morphological and spectral features. From a theoretical point of view, NWFE originated because of some inherent problems with DBFE [80]. Hence, it is preferable to use NWFE, especially when only a small training set is available.

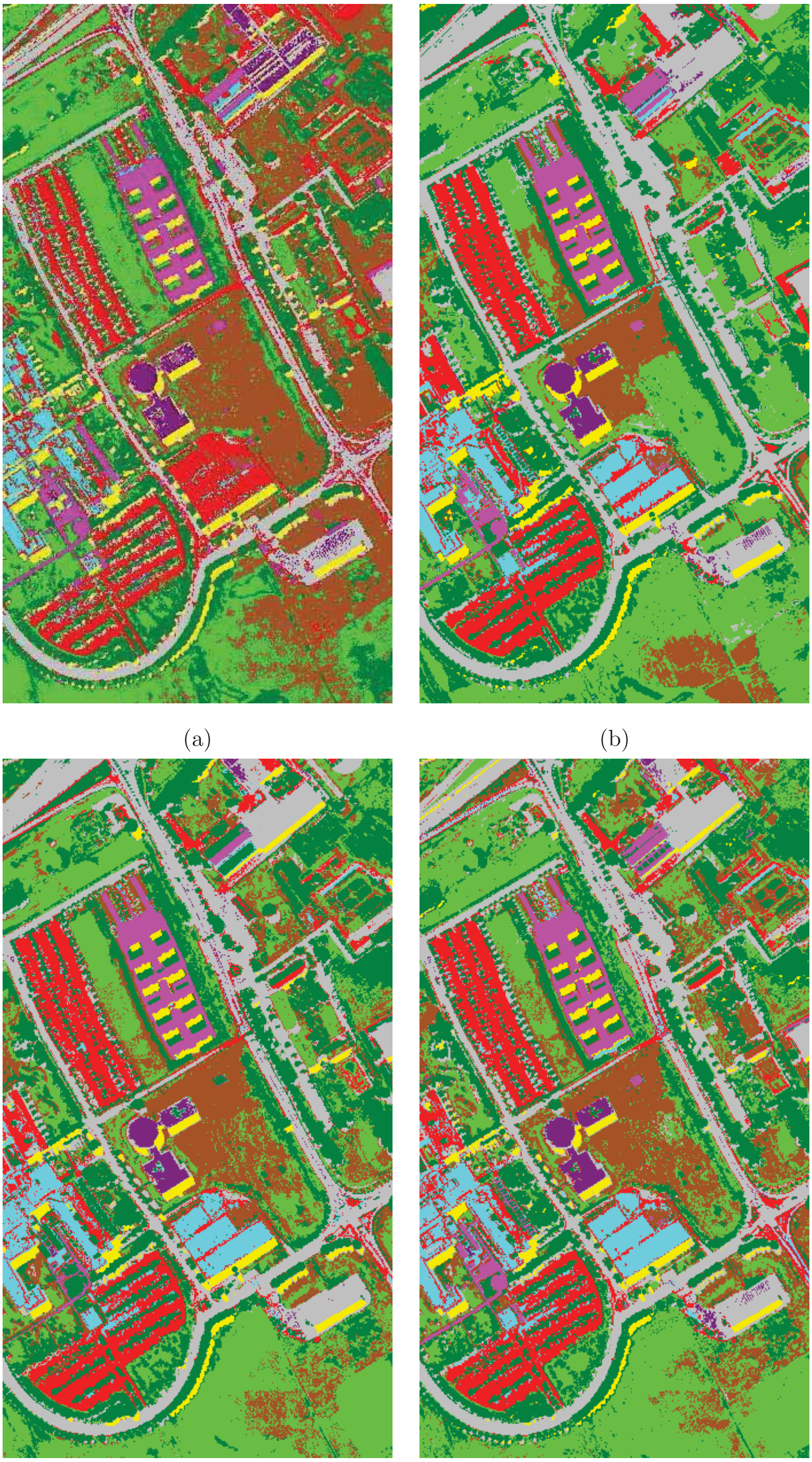
The full stacked vector contains a great deal of redundancy, as it is well known there is redundancy in the hyperspectral data [80] as well as in the EMP [7]. This is confirmed by the experiments. SVM is known to be robust to dimensionality, so the need for feature reduction might be called into question. However, lower-dimensional data reduced processing time, which can be crucial for some applications. More importantly, it has been proven that SVM can be affected by dimensionality in cases where many of the features are irrelevant [31]. By construction, the stacked vector may contain the same information many times over, and in the end feature extraction step is needed to ensure correct classification for all data sets (the usefulness of feature reduction in the classification of remote-sensing data using SVM was assessed in [5]).

Favorable performance has been obtained using the multi-source approach compared to the SS SVM in terms of classification accuracy. However, the thematic map obtained with the multi-source data is more noisy than with the SS SVM. Some elementary post-processing could overcome the problem.

In conclusion, in terms of the other results obtained using fusion of morphological and spectral information, the results presented outperformed those from our previous experiments [99, 100]. Furthermore, for the data sets under consideration, the multi-source approach yields the best results obtained in this thesis.

Table 6.14: *Summarized classification accuracies for SS SVM and the Data fusion approach for the* ***University Area*** *and the* ***Pavia Center****.*

	University Area		Center Center	
	SS SVM	Data Fusion	SS SVM	Data Fusion
OA	86.11	87.97	98.43	99.69
AA	91.98	89.43	97.13	98.07
κ	82.35	84.40	97.79	98.15
1	84.36	90.92	99.15	98.66
2	78.52	85.91	90.04	93.52
3	84.30	57.88	98.12	95.95
4	96.87	99.22	94.00	98.77
5	99.88	99.48	99.45	99.42
6	95.61	85.32	95.82	98.36
7	95.56	95.19	98.15	98.22
8	95.44	95.84	99.47	99.79
9	97.78	94.14	99.93	99.93



(a)

(b)

(c)

(d)

Figure 6.3: *ROSIS University Area: Classification map obtained using SVM from: (a) original hyperspectral data, (b) EMP, (c) 37 DBFE features, and (d) 13 NWFE features. Classification accuracies are reported in Tables 6.2, 6.3, 6.5, and 6.6.*

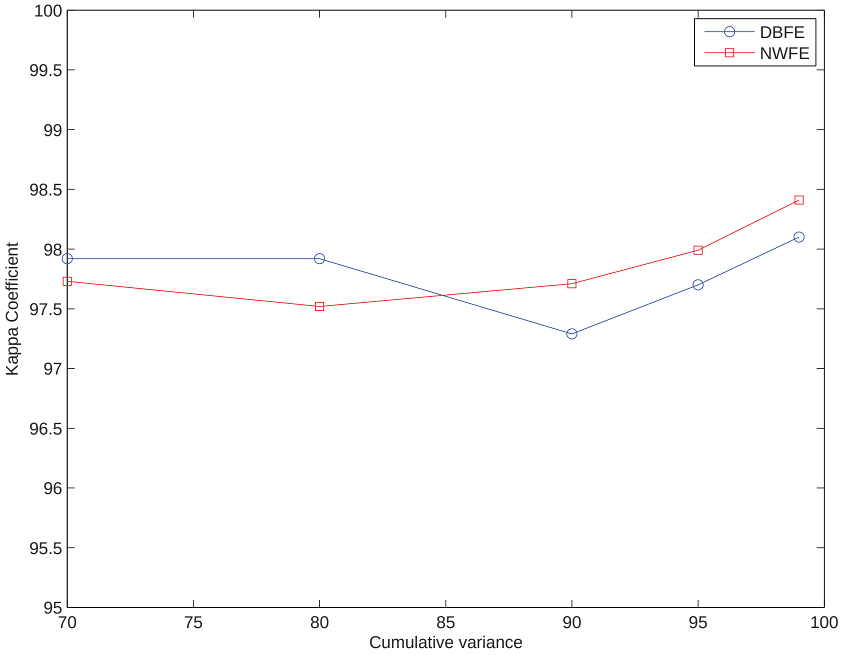
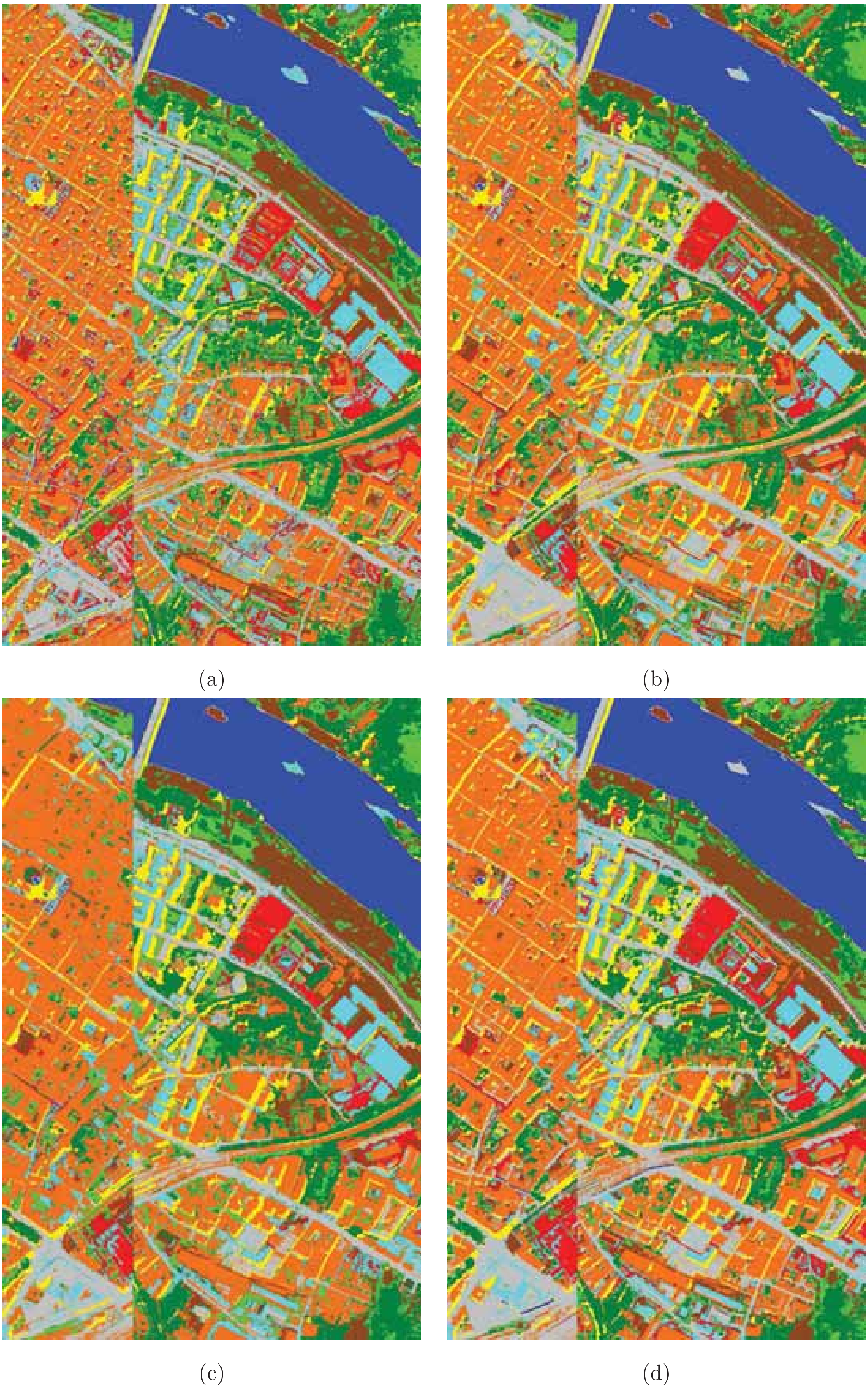


Figure 6.4: *κ values for several values of the cumulative variance for DBFE and NWFE applied to the Pavia Center data set.*


(a)

(b)

(c)

(d)

Figure 6.5: *ROSIS Pavia Center area: Classification map obtained with SVM from: (a) original hyperspectral data, (b) EMP, (c) 66 DBFE features, and (d) 64 NWFE features. Classification accuracies are reported in Tables 6.8, 6.9, 6.11 and 6.12.*

Conclusions

THE CLASSIFICATION of remote-sensing data over urban areas was addressed in this thesis. The objective was to propose a methodology to include the spatial information into the classification process in an appropriate way. Two general strategies have emerged from the present work, both yielding satisfactory results in terms of classification accuracy.

For the first strategy, we propose a two step approaches:

1. The first step consists of extracting the spatial information in the remotely-sensed data. We have assumed that the spatial organization of the image is required to be viewed in a structural sense. Hence, Mathematical Morphology was a natural choice as an image processing tool. The concept of the Morphological Profile has demonstrated a good capability for extracting useful spatial information. However, it does have certain theoretical limitations. An alternative concept, based on self-area filtering was proposed for extracting radiometric information about the structures in the image. The problem of multi-dimensional data has been tackled by the use of first principal components. We use the fact that PCA minimizes reconstruction error under the L2 norm to extract representative images from the data and perform the spatial analysis on these extracted feature. For the purpose of classification, this approach represents a good trade-off between complexity and efficiency.
2. The second step consists of classifying the data using the extracted spatial and spectral features. The use of support vector machines provides a solution to the problem of dimensionality related to hyperspectral data and the small training set size. For the Morphological Profile, a stacked vector is created using the extracted spatial feature. For the spectro-spatial SVM, we define a kernel that uses both the spectral and spatial information. In addition, a weighting parameter that controls the influence of each type of information is included in the proposed kernel. Experimental results exhibit excellent accuracies and complementary behavior of the two classification approaches. The morphological approach extract more informative geometrical features than the one based on self-complementary filter. Thus the morphological approach is better suited for dense urban area. On the other hand, the approach based on a self-complementary filter is better suited for peri-urban areas.

The second strategy is based on data fusion. We first investigated the fusion of the output from several classifiers and proposed a framework based on fuzzy logic. In this scenario, fusion is performed after a separate classification. Good results in terms of classification accuracy are obtained compared to the results obtained with each individual classifier. A special attention has been focused on the fusion of several SVM classifiers, for which a specific fusion scheme has been proposed.

The final chapter dealt with multi-source data. Spatial and spectral features from the same area are considered as two separate data sets. Data fusion is then performed prior to classification. The objective here was to include more spectral information in the Extended Morphological Profile.

In terms of classification accuracy, both strategies lead to improved results. It is interesting to note that the best results for each strategy are almost equal. Some conclusions may thus be drawn:

- For the Morphological Profile:
  - Experimental results with data fusion confirm that spectral information needs to be included in the feature vector (for multi-valued data).
  - Morphological Profile is better suited to very dense urban areas, due to the connectedness property.
- For the spectro-spatial SVM:
  - The adaptive neighbor definition is suitable to remote-sensing images.
  - Information other than the median value needs to be extracted from the neighbors set.
- For the data fusion:
  - Better results are obtained when fusing classifier outputs with complementary inputs.
  - Data fusion of classifier outputs seems the most promising.

There are many possible perspectives for pursuing this work. First of all, the PCA used to extract representative images could be changed to a more advanced algorithm. As seen in Chapter 1, Kernel PCA may be one possibility. The definition of image processing algorithms that can be applied to multi-valued data is certainly one point that needs to be addressed in the future.

Possible evolutions for the spectro-spatial SVM are the definition of new spatial information, such as textural features. Training time could be reduced using the radius margin bound.

The transferability of the hyperplane should be improved by the use of semi-supervised learning. A closer look at the feature space induced by the kernel function may provide an elegant solution to this particular problem.

Data fusion at a decision level has been validated as a useful method of aggregating several decision results. For reasons of practicality, the experiments presented in this thesis used a limited number of sources, and there is a need to evaluate the approach using many more classifiers. In particular, class specific classifiers (road or building detector) and global classifiers should be aggregated together.

As a final conclusion to this thesis, the conjoint use of spectral and spatial information with an appropriate classification scheme, whether that is the SVM or data fusion, provides good analysis of the urban area that is still not perfect, but is exploitable and helpful for analysts. Key points are that analyst intervention is minimized during processing, and that the methodology can be applied to many types of optical data.

Part IV

Appendix

Appendix A

Kernel Methods

Contents

KERNEL METHODS have been successfully applied to various algorithm, ranging from classification, regression, estimation [93, 111]... They are based on *reproducing kernel Hilbert spaces* (RKHS) theory where RKHS are special case of Hilbert space. From this theory, it is possible to analyze various data structures in their original representation (vectors, matrix, strings, probabilities...) with the theoretical framework of Hilbert spaces. Many algorithms can take benefit of kernel methods and the most famous one is surely the Support Vector Machines (SVM). Others well-know methods have used RKHS with success: principal and independent component analysis, Fisher discriminant, maximum likelyhood...In this thesis, we have presented the Kernel Principal Component Analysis and the SVM. Both take advantage of the *kernel trick*.

All this methods share the same property: they can be formulated with inner product. Their *kernelized version* is just found by substituting the inner product by a *kernel function*, the kernel trick is related to this substitution.

The remainder of this appendix is organized as follow. Some mathematical background are review in the first section. Then the main theorem is presented in the second section. Generalized representer theorem is then discussed in the third section. A synthetic example is given as a conclusion.

A.1 Mathematical background

In this section, basics of functional analysis will be presented. For the following we note $\mathcal{X}$ any given set made of *vector* and $\mathcal{K}$ any given field (*e.g.* $\mathbb{R}$ or $\mathbb{C}$) made of *scalar*. Vectors should be considered as elements of $\mathcal{X}$, which can be sample, set, function ...

A.1.1 Vector space

Definition A.1 (Vector space) *A vector space over the field $\mathcal{K}$ is a set $\mathcal{X}$ together with two laws: the vector addition and the scalar multiplication, which satisfy the following properties:*

- Vector addition: $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ belong to $\mathcal{X}$.
  1. Commutativity: $\mathbf{x}+\mathbf{y}=\mathbf{y}+\mathbf{x}$
  2. Associativity: $(\mathbf{x}+\mathbf{y})+\mathbf{z}=\mathbf{x}+(\mathbf{y}+\mathbf{z})$
  3. Identity element: $\mathbf{x}+\mathbf{0}=\mathbf{x}$ ($\mathbf{0}$ is called the zero vector)
  4. Inverse element: $\mathbf{x}+(-\mathbf{x})=\mathbf{0}$
- Scalar multiplication: a and b belongs to $\mathcal{K}$.
  1. Distributivity: $(a+b)(\mathbf{x}+\mathbf{y})=a\mathbf{x}+b\mathbf{x}+a\mathbf{y}+b\mathbf{y}$
  2. Associativity: $a(b\mathbf{x})=(ab)\mathbf{x}$
  3. Identity element: $1\mathbf{x}=\mathbf{x}$

Definition A.2 (Inner product) *An* inner product *is a map $\langle .,. \rangle_{\mathcal{X}} : \mathcal{X}\times\mathcal{X}\rightarrow\mathcal{K}$ satisfying the following axioms:*

1. *Conjugate symmetry:* $\langle \mathbf{x},\mathbf{y}\rangle_{\mathcal{X}}=\overline{\langle \mathbf{y},\mathbf{x}\rangle_{\mathcal{X}}}$
2. *Sesquilinearity:* $\langle a\mathbf{x}+b\mathbf{y},c\mathbf{z}+d\mathbf{w}\rangle_{\mathcal{X}}=a\bar{c}\langle \mathbf{x},\mathbf{z}\rangle_{\mathcal{X}}+a\bar{d}\langle \mathbf{x},\mathbf{w}\rangle_{\mathcal{X}}+b\bar{c}\langle \mathbf{y},\mathbf{z}\rangle_{\mathcal{X}}+b\bar{d}\langle \mathbf{y},\mathbf{w}\rangle_{\mathcal{X}}$
3. *Non-negativity:* $\langle \mathbf{x},\mathbf{x}\rangle_{\mathcal{X}}\geq 0$
4. *Positive definiteness:* $\langle \mathbf{x},\mathbf{x}\rangle_{\mathcal{X}}=0\Leftrightarrow \mathbf{x}=\mathbf{0}$

The standard inner product in the Euclidean space, $\mathbf{x}\in\mathbb{R}^n$ and $n\in\mathbb{N}$, is called the dot product: $\langle \mathbf{x},\mathbf{y}\rangle_{\mathbb{R}^n}=\sum_{i=1}^n x_iy_i$.

Definition A.3 (Pre-Hilbert Space) *A* pre-Hilbert space *is a vector space $\mathcal{X}$ over the field $\mathcal{K}$ (either real $\mathbb{R}$ or complex $\mathbb{C}$ number) endowed with a inner product. It can be of any dimension, even non-finite.*

Definition A.4 (Norm) *A* norm *is a real-valued function $\|.\|_{\mathcal{X}}$ over $\mathcal{X}$ which satisfy the following properties, for $\mathbf{x}$, $\mathbf{y}\in\mathcal{X}$:*

1. $\|\mathbf{x}\|_{\mathcal{X}} \geq 0$
2. $\|\mathbf{x}\|_{\mathcal{X}} = 0 \Leftrightarrow \mathbf{x} = \mathbf{0}$
3. $\|a\mathbf{x}\|_{\mathcal{X}} = |a|.\|\mathbf{x}\|_{\mathcal{X}}$
4. $\|\mathbf{x} + \mathbf{y}\|_{\mathcal{X}} \leq \|\mathbf{x}\|_{\mathcal{X}} + \|\mathbf{y}\|_{\mathcal{X}}$

Definition A.5 (Normed vectors space) *A normed vector space $(\mathcal{X}, \|.\|_{\mathcal{X}})$ is a vector space $\mathcal{X}$ together with a norm $\|.\|_{\mathcal{X}}$.*

To each inner product, it is possible to associate a norm: $\|\mathbf{x}\|_{\mathcal{X}} = \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle_{\mathcal{X}}}$. The norm induces a notion of distance between two vectors $\mathbf{x}$ and $\mathbf{y}$: $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_{\mathcal{X}}$. Therefore, a vector space endows with a distance is called a *metric vector space* $(\mathcal{X}, d)$. Every normed vector spaces are metric vector spaces. However, the converse is not true:

$$\text{pre-Hilbert vector space} \Rightarrow \text{ normed vector space} \Rightarrow \text{ metric vector space.} \tag{A.1}$$

Definition A.6 (Cauchy sequence) *A sequence $(\mathbf{x}_i)_{i\in\mathbb{N}}$ in a metric vector space $(\mathcal{X}, d)$ is said to be a* Cauchy sequence *if for all $\epsilon \in \mathbb{R}_*^+$ it exits $n \in \mathbb{N}$ such as for all $p, q \geq n$ we have $d(\mathbf{x}_p, \mathbf{x}_q) \leq \epsilon$. Note that all convergent sequences are* Cauchy sequences *but the converse is not true in general.*

Definition A.7 (Complete vector space) *A metric vector space $(\mathcal{X}, d)$ is said to be* complete *if any Cauchy sequences converge to an element* in $\mathcal{X}$*. This property is linked to the distance d. $\mathcal{X}$ can be complete for a given d but non complete for another d'.*

For any metric vector space, it is possible to construct a complete metric vector space. Especially, every pre-Hilbert vector space can be completed to a Hilbert vector space.

Definition A.8 (Hilbert vector space) *An* Hilbert vector space $(\mathcal{H}, \langle ., .\rangle_{\mathcal{H}})$ *is a normed vector space complete for the distance stemming from its inner product. Or straightforwardly, an* Hilbert space *is a complete pre-Hilbert space.*

Hilbert space are the generalization of Euclidean and Hermitian space to infinite dimension. Theory of linear functions over an Hilbert space is very well developed. Fourier transform is defined over an Hilbert space, as the Quantic Theory. In the following section A.1.2, we need two classical properties of Hilbert space. The first is the *Cauchy-Schwarz* inequality:

$$\forall \mathbf{x}, \mathbf{y} \in (\mathcal{H}, \langle ., .\rangle_{\mathcal{H}}),\ |\langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{H}}| \leq \|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}} \tag{A.2}$$

where$\|\mathbf{x}\|_{\mathcal{H}} = \sqrt{\langle \mathbf{x}, \mathbf{x}\rangle_{\mathcal{H}}}$. The second one is the *Riesz representation theorem*:

Theorem A.1 (Riesz representation theorem) *For all linear continuous form f on $(\mathcal{H}, \langle ., .\rangle_{\mathcal{H}})$, there is an unique $\mathbf{y}$ in $\mathcal{H}$ such as:*

$$\forall \mathbf{x} \in \mathcal{H}, f(\mathbf{x}) = \langle \mathbf{y}, \mathbf{x}\rangle_{\mathcal{H}}. \tag{A.3}$$

Suppose $\Phi_{\mathbf{x}}$ is linear form over an functional Hilbert space such as:

$$\begin{aligned} \Phi_{\mathbf{x}} : \mathcal{H} \quad &\rightarrow \quad \mathcal{K} \\ f \quad &\rightarrow \quad f(\mathbf{x}) \end{aligned} \tag{A.4}$$

then according to the previous theorem states, an unique functional $g_{\mathbf{x}}$ exists such as :

$$\Phi_{\mathbf{x}}(f) = \langle g_{\mathbf{x}}, f\rangle_{\mathcal{H}}. \tag{A.5}$$

Some remarks should be made about the completion of pre-Hilbert space forming by a class of functions [4]. The *completion theorem* says that the completion of metric vector space is always possible by adding the limits of Cauchy sequences, if necessary. However, the completed space will not necessarily form a class of functions and when considering the special case of functional vector space one has to look for the *functional completion* theorem.

Theorem A.2 (Functional Completion) *Let $\mathcal{H}$ be the class of $\mathcal{K}$-valued functions defined in $\mathcal{X}$, and $\mathcal{H}$ forms a pre-Hilbert vector space. There exist a functional completion if and only if:*

1. $\forall \mathbf{x} \in \mathcal{X},\ f \in \mathcal{H} :\ |f(\mathbf{x})| \leq M_{\mathbf{x}} \|f\|$
2. $(f_i)_{i \in \mathbb{N}}$ *a Cauchy sequence of* $\mathcal{H}$*:* $\forall \mathbf{x},\ \lim\limits_{m \to \infty} f_m(\mathbf{x}) = 0 \Rightarrow \lim\limits_{m \to \infty} \|f_m\| = 0$

A.1.2 Kernels

In the following we restrict the field $\mathcal{K}$ to $\mathbb{R}$ or $\mathbb{C}$.

Definition A.9 (Positive semi-definiteness) *A $\mathcal{K}$-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathcal{K}$ on a vector space is said to be* positive semi-definite *if, and only if:*

1. $\forall (\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{X},\ k(\mathbf{x}, \mathbf{y}) = \overline{k(\mathbf{y}, \mathbf{x})}$
2. $\forall n \in \mathbb{N}, \forall \xi_1, \ldots, \xi_n \in \mathcal{K}, \forall \mathbf{x}^1, \ldots, \mathbf{x}^n \in \mathcal{X}, \sum_{i,j=1}^{n} \bar{\xi}_i \xi_j k(\mathbf{x}^i, \mathbf{x}^j) \geq 0$

Properties of positive semi-definite functions (PSDF):

1. If $(\mathcal{X}, \langle ., . \rangle_{\mathcal{X}})$ is an Hilbert space, then $\langle ., . \rangle_{\mathcal{X}}$ is PSDF.
2. If $g : \mathcal{X} \to \mathcal{K}$ then $k(\mathbf{x}, \mathbf{y}) = \overline{g(\mathbf{x})} g(\mathbf{y})$ is PSDF.
3. If k_1 and k_2 are PSDF, and with $\lambda_1, \lambda_2 \in \mathbb{R}^+$ then $\lambda_1 k_1 + \lambda_2 k_2$ is also PSDF.
4. If k_1 and k_2 are PSDF then $k_1.k_2$ is also PSDF.
5. if k_1 is PSDF then $k = \exp(k_1)$ is also PSDF.

The PSDF can be seen a generalization of inner product. Every inner product is clearly a PSDF. In general, PSDF does not have the linearity property of inner product but the Cauchy-Schwarz inequality is somewhat preserved, since we have [110]:

$$|k(\mathbf{x}, \mathbf{y})|^2 \leq k(\mathbf{x}, \mathbf{x}).k(\mathbf{y}, \mathbf{y}). \tag{A.6}$$

Definition A.10 (Kernel) *A $\mathcal{K}$-valued function $k : \mathcal{X} \times \mathcal{X} \to \mathcal{K}$ is called a* kernel *on $\mathcal{X}$ if there exits a $\mathcal{K}$-Hilbert vector space ($\mathcal{H}$, $\langle ., . \rangle_{\mathcal{H}}$) and a map $\Phi : \mathcal{X} \to \mathcal{H}$ such as:*

$$\forall \mathbf{x}, \mathbf{y} \in \mathcal{X} : k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{y}), \Phi(\mathbf{x}) \rangle_{\mathcal{H}} \tag{A.7}$$

Usually, $\mathcal{X}$ is called the input space, $\mathcal{H}$ the feature space and Φ the feature map.

Definition A.11 (Reproducing kernel) *$\mathcal{H}$ is vector space of $\mathcal{K}$-valued functions over $\mathcal{X}$ and $(\mathcal{H}, \langle ., . \rangle_{\mathcal{H}})$ its associated Hilbert vector space ($f \in \mathcal{H}, \|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$). The kernel k is a* reproducing kernel *of $\mathcal{H}$ if :*

1. $\forall \mathbf{x}$ *and* $\mathbf{y} \in \mathcal{X},\ k(\cdot, \mathbf{y}) \in \mathcal{H}$*:*

$$\begin{aligned} k(\cdot, \mathbf{y}) : \mathcal{X} &\to \mathcal{K} \\ \mathbf{x} &\mapsto k(\mathbf{x}, \mathbf{y}) \end{aligned} \tag{A.8}$$

2. $\forall \mathbf{x} \in \mathcal{X}$ *and* $\forall f \in \mathcal{H} : f(\mathbf{x}) = \langle f, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$

Especially, we have the reproducing property: $k(\mathbf{x}, \mathbf{y}) = \langle k(\cdot, \mathbf{y}), k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}}$ ($f = k(\cdot, \mathbf{y})$).

An Hilbert vector space where a reproducing kernel exists is called a *reproducing kernel Hilbert space* (RKHS). Roughly speaking, a RKHS is *smaller* than a general Hilbert space.

Properties of reproducing kernel [4] :

1. If k exists, it is unique.
2. A reproducing kernel exists if, and only if, for all $\mathbf{x}$ the linear form

$$\begin{aligned} \Phi_{\mathbf{x}} : \mathcal{H} &\to \mathcal{K} \\ f &\mapsto f(\mathbf{x}) \end{aligned} \tag{A.9}$$

is a continuous linear form.

3. A reproducing kernel is a PSDF
4. Every sequence $(f_i)_{i\in\mathbb{N}}$ which converges strongly to a function f, converges also at every point (point-wise convergence):

$$\lim_{i\to\infty} \|f_i\|_{\mathcal{H}} = \|f\|_{\mathcal{H}} \Rightarrow \lim_{i\to\infty} f_i(\mathbf{x}) = f(\mathbf{x}),\ \forall \mathbf{x} \in \mathcal{X} \tag{A.10}$$

Proof :

1. Let k and k' be two reproducing kernels on $\mathcal{H}$, for all $\mathbf{x} \in \mathcal{X}$:

$$\begin{aligned}\|k(\cdot,\mathbf{x}) - k'(.,\mathbf{x})\|_{\mathcal{H}} &= \langle k(\cdot,\mathbf{x}) - k'(.,\mathbf{x}), k(\cdot,\mathbf{x}) - k'(.,\mathbf{x})\rangle_{\mathcal{H}} \\ &= \langle k(\cdot,\mathbf{x}), k(\cdot,\mathbf{x})\rangle_{\mathcal{H}} - \langle k(\cdot,\mathbf{x}), k'(.,\mathbf{x})\rangle_{\mathcal{H}} - \langle k'(.,\mathbf{x}), k(\cdot,\mathbf{x})\rangle_{\mathcal{H}} + \langle k'(.,\mathbf{x}), k'(.,\mathbf{x})\rangle_{\mathcal{H}} \\ &= k(\mathbf{x},\mathbf{x}) - k'(\mathbf{x},\mathbf{x}) - k(\mathbf{x},\mathbf{x}) + k'(\mathbf{x},\mathbf{x}) \\ &= 0.\end{aligned} \tag{A.11}$$

So $k = k'$ for all $\mathbf{x} \in \mathcal{X}$.

2. On one hand, if k exists :

$$\begin{aligned}|\Phi_{\mathbf{x}}(f)| = |f(\mathbf{x})| &= |\langle f, k(\cdot,\mathbf{x})\rangle_{\mathcal{H}}| \\ |f(\mathbf{x})| &\leq \|f\|_{\mathcal{H}}\|k(\cdot,\mathbf{x})\|_{\mathcal{H}} \\ |f(\mathbf{x})| &\leq \sqrt{k(\mathbf{x},\mathbf{x})}\|f\|_{\mathcal{H}}.\end{aligned} \tag{A.12}$$

the linear form $\Phi_{\mathbf{x}}$ is continuous.
On the other hand, if $\Phi_{\mathbf{x}}$ is continuous, by the representation theorem of Riesz it exists $g_{\mathbf{x}} \in \mathcal{H}$ such as $\Phi_{\mathbf{x}}(f) = \langle g_{\mathbf{x}}, f\rangle_{\mathcal{H}} = f(\mathbf{x})$. Then $g_{\mathbf{x}} = k(\cdot,\mathbf{x})$ is a reproducing kernel since $g_{\mathbf{x}}$ belongs to $\mathcal{H}$ and has the reproducing property.

3. By definition we have $k(\mathbf{x},\mathbf{y}) = \langle k(\cdot,\mathbf{y}), k(\cdot,\mathbf{x})\rangle_{\mathcal{H}} = \overline{\langle k(\cdot,\mathbf{x}), k(\cdot,\mathbf{y})\rangle_{\mathcal{H}}} = \overline{k(\mathbf{y},\mathbf{x})}$. Thanks again to the reproducing property, the second condition of positive semi-definiteness can be rewritten as

$$\begin{aligned}&\sum_{i,j=1}^{n} \bar{\xi}_i\xi_j \langle k(\cdot,\mathbf{x}^j), k(\cdot,\mathbf{x}^i)\rangle_{\mathcal{H}} \\ &= \left\langle \sum_{j=1}^{n} \xi_j k(\cdot,\mathbf{x}^j), \sum_{i=1}^{n} \xi_i k(\cdot,\mathbf{x}^i) \right\rangle_{\mathcal{H}} \\ &= \| \sum_{i=1}^{n} \xi_i k(\cdot,\mathbf{x}^i)\|_{\mathcal{H}}^2 \\ &\geq 0.\end{aligned} \tag{A.13}$$

The converse of this property is at the very basis of *kernel methods*. It leads to theorem (A.3) that will be stated latter.

4. $\forall \mathbf{x} \in \mathcal{X},\ |f_i(\mathbf{x}) - f(\mathbf{x})| = |\langle k(\cdot,\mathbf{x}), f_i - f\rangle_{\mathcal{H}}| \leq \|f_i - f\|\sqrt{k(\mathbf{x},\mathbf{x})}$. So if strong convergence in norm holds ($\lim_{i\to\infty} \|f_i - f\|_{\mathcal{H}} = 0$) then point-wise convergence also ($\lim_{i\to\infty} |f_i(\mathbf{x}) - f(\mathbf{x})| = 0$, for all $\mathbf{x} \in \mathcal{X}$).

A.2 From feature space to Kernel feature space

In the previous section, we have shown that input space and feature space could be linked with a kernel function, see definition A.10. Furthermore, particular kernels have been presented leading to special Hilbert space. In that configuration, we have seen that every reproducing kernels are also positive semi-definite functions. Moore and Aronszajn have proved that the converse is true [4].

Theorem A.3 (Moore and Aronszajn) *To every positive semi-definite function k there corresponds one and only one RKHS admitting k as a reproducing kernel.*

The proof is based on the construction of such an Hilbert space with a given positive semi-definite function k. For this, suppose we have a given non-empty set $\mathcal{X}$ and $k : \mathcal{X} \times \mathcal{X} \to \mathcal{K}$ a positive semi-definite function. We have to show 1) k is a kernel and 2) the Hilbert space associated to k is an RKHS, *i.e.*, k is a reproducing kernel (the unicity is enclosed in the existence, see property 1 of reproducing kernel).

We can consider the set $\mathcal{H}_0$ of linear $\mathcal{K}$-valued functions such as :

$$f := \sum_{i=1}^{p} \alpha_i k(\cdot, \mathbf{x}^i) | \mathbf{x}^i \in \mathcal{X},\ \alpha_i \in \mathcal{K} \text{ and } p \in \mathbb{N}. \tag{A.14}$$

It is easy to show that $\mathcal{H}_0$ is a vector space. The next point is to construct pre-Hilbert space and then try to apply the completion theorem. We endow $\mathcal{H}_0$ with the functional $\left(g := \sum_{j=1}^{q} \beta_j k(\cdot, \mathbf{y}^j)\right)$:

$$\langle f, g\rangle_{\mathcal{H}_0} := \sum_{i,j=1}^{p,q} \alpha_i \bar{\beta}_j k(\mathbf{y}^j, \mathbf{x}^i). \tag{A.15}$$

First we can see that for $q = 1$ and $\beta = 1$ ($g = k(\cdot, \mathbf{y})$) we have:

$$\langle f, g\rangle_{\mathcal{H}_0} = \sum_{i=1}^{p} \alpha_i k(\mathbf{y}, \mathbf{x}^i) = f(\mathbf{y}) = \langle f, k(\cdot, \mathbf{y})\rangle_{\mathcal{H}_0} \tag{A.16}$$

and the reproducing property ($p = 1$ and $\alpha = 1$, $f = k(\cdot, \mathbf{x})$):

$$\langle k(\cdot, \mathbf{x}), k(\cdot, \mathbf{y})\rangle_{\mathcal{H}_0} = k(\mathbf{y}, \mathbf{x}). \tag{A.17}$$

Lets show $\langle \cdot, \cdot\rangle_{\mathcal{H}_0}$ defines a inner product in $\mathcal{H}_0$ (the number in the list are related to those of definition A.2):

1. $\overline{\langle g, f\rangle_{\mathcal{H}_0}} = \sum_{i,j=1}^{p,q} \overline{\bar{\alpha}_i \beta_j k(\mathbf{x}^i, \mathbf{y}^j)} = \sum_{i,j=1}^{p,q} \alpha_i \bar{\beta}_j \overline{k(\mathbf{x}^i, \mathbf{y}^j)} = \sum_{i,j=1}^{p,q} \alpha_i \bar{\beta}_j k(\mathbf{y}^j, \mathbf{x}^i) = \langle f, g\rangle_{\mathcal{H}_0}$
2. $h = \sum_{k}^{r} \gamma_k k(\cdot, \mathbf{z}^k)$. To prove the linearity for the first variable, consider
   - $\langle f, g\rangle_{\mathcal{H}_0} + \langle h, g\rangle_{\mathcal{H}_0} = \sum_{i,j=1}^{p,q} \alpha_i \bar{\beta}_j k(\mathbf{y}^j, \mathbf{x}^i) + \sum_{k,j=1}^{r,q} \gamma_k \bar{\beta}_j k(\mathbf{y}^j, \mathbf{z}^k) = \sum_{i,j,k=1}^{p,q,r} (\alpha_i + \gamma_k)\bar{\beta}_j (k(\mathbf{y}^j, \mathbf{x}^i) + k(\mathbf{y}^j, \mathbf{z}^k))$ $= \langle f + h, g\rangle_{\mathcal{H}_0}$
   - $\langle \lambda f, g\rangle_{\mathcal{H}_0} = \sum_{i,j=1}^{p,q} \lambda \alpha_i \bar{\beta}_j k(\mathbf{y}^j, \mathbf{x}^i) = \lambda \langle f, g\rangle_{\mathcal{H}_0}$

   With the conjugate symetry, we have the sesquilinearity.
3. $\langle f, f\rangle_{\mathcal{H}_0} = \sum_{i,j=1}^{p} \alpha_i \bar{\alpha}_j k(\mathbf{x}^j, \mathbf{x}^i) \geq 0$ since k is PSDF.
4. Using the previous property, for $\lambda \in \mathcal{K}$ we have : $\langle f - \lambda g, f - \lambda g\rangle_{\mathcal{H}_0} \geq 0$. Following the same flow as the classical Cauchy-Schwarz proof, we get: $|\langle f, g\rangle_{\mathcal{H}_0}| \leq \sqrt{\langle f, f\rangle_{\mathcal{H}_0}} \sqrt{\langle g, g\rangle_{\mathcal{H}_0}}$. Using equation (A.16) we can see: $|f(\mathbf{x})| = |\langle f, k(\cdot, \mathbf{x})\rangle_{\mathcal{H}_0}| \leq \sqrt{\langle f, f\rangle_{\mathcal{H}_0}} \sqrt{k(\mathbf{x}, \mathbf{x})}$. If $\langle f, f\rangle_{\mathcal{H}_0} = 0$ then $f(\mathbf{x}) = 0$ for all $\mathbf{x}$, so $f = \mathbf{0}$.

We have shown that $\mathcal{H}_0$ with the inner product (A.15) is a pre-Hilbert space. We note $\|\cdot\|_{\mathcal{H}_0}$ the norm associated to the inner product $\langle \cdot, \cdot\rangle_{\mathcal{H}_0}$. We now use the completion theorem to complete $\mathcal{H}_0$ to an Hilbert space. Since we are dealing we reproducing kernel, every functions in $\mathcal{H}_0$ can be written in terms of inner product and kernel function. So we have for every functions $\mathbf{x}$:

$$|f(\mathbf{x})| \leq \sqrt{k(\mathbf{x}, \mathbf{x})} \|f\|_{\mathcal{H}_0}. \tag{A.18}$$

Now we have to verify the second condition by considering Cauchy sequence $(f_m)_{m\in\mathbb{N}}$ in $\mathcal{H}_0$. The previous equation (A.18) does not help us and we need to rewrite the norm of f_m in inner product formulation, to use the explicit expression of f_m, see equation (A.14):

$$\begin{aligned} \|f_m\|^2_{\mathcal{H}_0} &= \langle f_m, f_m\rangle_{\mathcal{H}_0} \\ &= \sum_{i,j=1}^{p} \alpha_i \bar{\alpha}_j k(\mathbf{x}^j, \mathbf{x}^i) \\ &= \sum_{j=1}^{p} \bar{\alpha}_j \sum_{i=1}^{p} \alpha_i k(\mathbf{x}^j, \mathbf{x}^i) \\ &= \sum_{j=1}^{p} \bar{\alpha}_j f_m(\mathbf{x}^j) \end{aligned} \tag{A.19}$$

so if for all $\mathbf{x}$ in $\mathcal{X}$, $f_m(\mathbf{x})$ converges to $f(\mathbf{x}) = 0$ then $\|f_m\|$ converges to zero. This satisfy the second condition of the completion theorem, *i.e.*, $\mathcal{H}_0$ can be completed to obtain a complete Hilbert space $\mathcal{H}$.

Finally, the PSDF is a reproducing kernel since:

1. It satisfies the definition A.11 in the above constructed Hilbert space $\mathcal{H}$
2. It maps $\mathcal{X}$ to $\mathcal{H}$:

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{H} \\ \mathbf{x} &\mapsto k(\cdot, \mathbf{x}) \end{aligned} \tag{A.20}$$

and we have the equality: $k(\mathbf{x}, \mathbf{y}) = \langle k(\cdot, \mathbf{y}), k(\cdot, \mathbf{x})\rangle_{\mathcal{H}}$.

The major consequence of this theorem is that it allows the use of functional theory over Hilbert spaces for a various classes of problems. The main issue is to define the kernel function. This is classically done by combining classical kernels with above given rules. However, optimal solution should be found using specific kernel associated to a specific problem. Another crucial point is that the mapping is implicitly done while all the operations are done in the input space. In the following we present two classical kernels widely used in kernel based algorithm.

Definition A.12 (Polynomial Kernel) *If $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $q \in \mathbb{R}^+$ and $p \in \mathbb{N}^+$, the polynomial kernel is defined as*

$$k(\mathbf{x}, \mathbf{y}) = \left(\langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{X}} + q\right)^p . \tag{A.21}$$

It corresponds to the inner product in a $\binom{n+p}{p}$ dimensional vector space, where n is the dimension of $\mathcal{X}$, and each component is made of monomial up to degree p. For example, consider the following polynomial kernel k:

$$\begin{aligned} k : \mathbb{R}^2 \times \mathbb{R}^2 &\to \mathbb{R} \\ (\mathbf{x}, \mathbf{y}) &\mapsto \left(\langle \mathbf{x}, \mathbf{y}\rangle_{\mathbb{R}^2} + 1\right)^2 \\ &\mapsto (x_1y_1 + x_2y_2 + 1)^2 \\ &\mapsto (x_1y_1)^2 + (x_2y_2)^2 + 2x_1y_1 + 2x_2y_2 + 2x_1x_2y_1y_2 + 1 \\ &\mapsto \langle \Phi(\mathbf{x}), \Phi(\mathbf{y})\rangle_{\mathbb{R}^6} \end{aligned} \tag{A.22}$$

where $\Phi(\mathbf{x}) = \left[x_1^2 \ x_2^2 \ \sqrt{2}x_1 \ \sqrt{2}x_2 \ \sqrt{2}x_1x_2 \ 1\right]$ and $\dim(\Phi(\mathbf{x})) = \binom{2+2}{2} = 6$. Its components are monomials made of the original vector's component.

Definition A.13 (Radial Basis Functions Kernel (RBF)) *For $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $\sigma \in \mathbb{R}^+$, the RBF kernel is defined as:*

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \tag{A.23}$$

The parameter σ controls the smoothness of the function. When σ is set to large value, the kernel function tends to a very flat functions. The value controls how the neighbors in the input space are taking into account. It is illustrated in the figure A.2. We have considered the functions f define in the feature space with RBF:

$$f(\mathbf{x}) = \sum_{i=1}^{p} \alpha_i \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}^i\|^2}{2\sigma^2}\right) \tag{A.24}$$

This function represent an density estimator in the input space. For the figure A.2, we have generated data from an additive mixture of two Gaussian. It can be seen from the figure that with a too small or too big value of σ, the estimation is not accurate. This is relatively important since when using the RBF kernel one has to choose a proper σ value. This issue is investigated in section 1.4 for the KPCA.

To conclude this section, some discussion about the feature space induce by the kernel function should be done. If we consider a polynomial kernel function of degree 2 with $q = 0$. We can easily see that the corresponding mapping is :

$$\begin{array}{rcl} \Phi : \mathbb{R}^2 & \rightarrow & \mathbb{R}^3 \\ \mathbf{x} = (x_1, x_2) & \mapsto & \Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2) \end{array} \tag{A.25}$$

Then if we look at the associated feature space we see that the $\mathbb{R}^3$ is not totally span by the $\Phi(\mathbf{x})$, since the third dimension is linked with the first ones. The figure A.1 represent the square $[-3; 3] \times [-3; 3]$ in $\mathbb{R}^2$ mapped onto $\mathbb{R}^3$ by Φ. The remaining question is *does the solution of our problem live in the spanned space*? If not, we cannot access to it or only partially. Hopefully, the *representer theorem* gives us the condition to apply kernel methods.

A.3 The Representer Theorem

Theorem A.4 (The Representer Theorem [64, 110]) *Let k be a positive semi-definite function on $\mathcal{X}$, a training set $(\mathbf{x}^i, y_i) \in \mathcal{X} \times \mathcal{Y}$ made of ℓ elements, $g_{emp} : (\mathcal{X} \times \mathcal{Y} \times \mathcal{K})^\ell \rightarrow \mathbb{R} \cup \infty$ be any arbitrary cost function and $g_{reg} : \mathcal{K} \rightarrow [0, \infty[$ a strictly monotonically increasing function. Define $\mathcal{H}$ the RKHS induced by k; then any $f \in \mathcal{H}$ minimizing the regularized risk :*

$$R_{reg} = g_{emp}\left((\mathbf{x}^i, y_i, f(\mathbf{x}^i))_{i\in\{1,\dots,\ell\}}\right) + g_{reg}\left(\|f\|_{\mathcal{H}}\right) \tag{A.26}$$

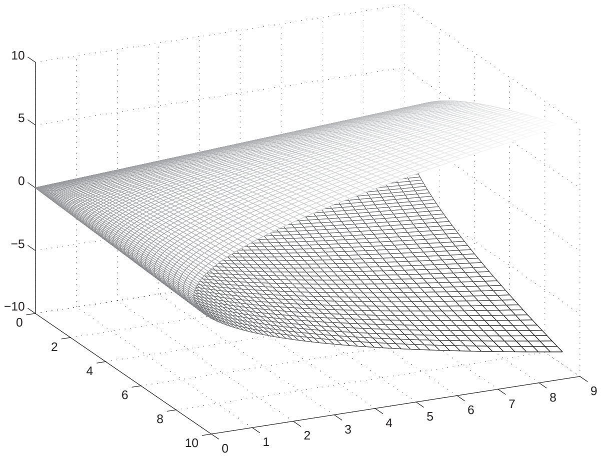
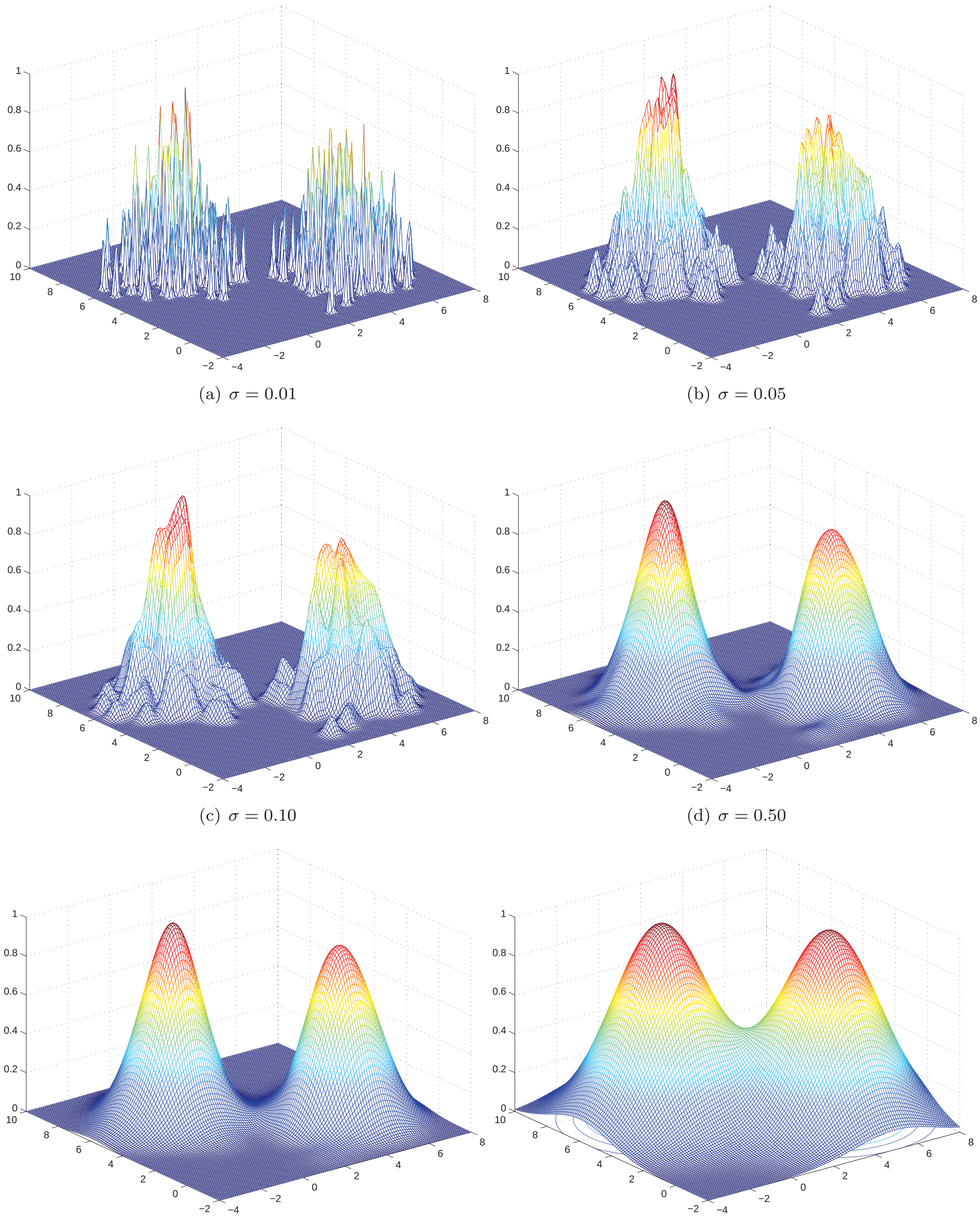


Figure A.1: *A view of the kernel feature space in the feature space.*



(a) $\sigma = 0.01$

(b) $\sigma = 0.05$

(c) $\sigma = 0.10$

(d) $\sigma = 0.50$

(e) $\sigma = 1.00$

(f) $\sigma = 5.00$

Figure A.2: *Influence of σ on density estimation for a mixture of two Gaussian, from [64].*

admits a representation of the form :

$$f = \sum_{i=1}^{\ell} \alpha_i k(\cdot, \mathbf{x}^i) \qquad\qquad \alpha \in \mathcal{K} \tag{A.27}$$

Proof: Let first write the kernel map:

$$\begin{array}{rcl} \Phi_{\mathbf{x}} : \mathcal{X} & \rightarrow & \mathcal{H} \\ \mathbf{x} & \mapsto & \Phi_{\mathbf{x}} = k(\cdot, \mathbf{x}). \end{array} \tag{A.28}$$

Using the reproducing property, the evaluation of $\Phi_{\mathbf{x}}$ is done using k: $\Phi_{\mathbf{x}}(\mathbf{y}) = k(\mathbf{y}, \mathbf{x}) = \langle \Phi_{\mathbf{x}}, \Phi_{\mathbf{y}} \rangle_{\mathcal{H}}$. Given the training set, any $f \in \mathcal{H}$ can be decomposed into a part that lives in the spanned space and a part $\mathbf{u}$ which is orthogonal to it ($\langle \Phi_{\mathbf{x}^i}, \mathbf{u} \rangle_{\mathcal{H}} = 0,\ \forall i \in \{1, \ldots, \ell\}$) :

$$f = \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} + \mathbf{u}. \tag{A.29}$$

Still using the reproducing property, the evaluation of f is:

$$\begin{aligned} f(\mathbf{x}^j) &= \langle \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} + \mathbf{u}, k(\cdot, \mathbf{x}^j) \rangle_{\mathcal{H}} \\ &= \langle \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} + \mathbf{u}, \Phi_{\mathbf{x}^j} \rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{\ell} \alpha_i \langle \Phi_{\mathbf{x}^i}, \Phi_{\mathbf{x}^j} \rangle_{\mathcal{H}} \end{aligned} \tag{A.30}$$

which does not depend on $\mathbf{u}$. So the functional g_{emp} is independent of $\mathbf{u}$. For the second term $g_{reg}\left(\| \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} + \mathbf{u} \|_{\mathcal{H}}\right)$, using the orthogonality it can be rewritten $g_{reg}\left(\sqrt{\| \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} \|_{\mathcal{H}}^2 + \|\mathbf{u}\|_{\mathcal{H}}^2}\right)$, finally using the strict monotonic of g_{reg}, we have:

$$\sqrt{\| \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} \|_{\mathcal{H}}^2 + \|\mathbf{u}\|_{\mathcal{H}}^2} \geq \left\| \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} \right\|_{\mathcal{H}} \tag{A.31}$$

$$g_{reg}\left(\|f\|_{\mathcal{H}}\right) \geq g_{reg}\left(\left\| \sum_{i=1}^{\ell} \alpha_i \Phi_{\mathbf{x}^i} \right\|_{\mathcal{H}} \right). \tag{A.32}$$

The equality holds for $\mathbf{u} = \mathbf{0}$. So, setting $\mathbf{u} = \mathbf{0}$ does not change g_{emp} while strictly reducing g_{reg}, thus any minimizer must have $\mathbf{u} = \mathbf{0}$. Combining (A.29) and (A.28) we finally have:

$$f = \sum_{i=1}^{\ell} \alpha_i k(\cdot, \mathbf{x}^i) \tag{A.33}$$

which proves the Representer Theorem.

We will give two machine learning algorithms that satisfy the Representer Theorem, the Kernel Principal Component Analysis and the Support Vector Machines, both were used in this thesis.

KPCA: The PCA in the feature space is an unsupervised linear feature extraction by a functional f that produce unit empirical variance's outputs, g_{reg} is a strictly monotonically increasing function:

$$g_{emp}\left((\mathbf{x}^i, y_i, f(\mathbf{x}^i))_{i \in \{1, \ldots, \ell\}}\right) = \left\{ \begin{array}{ll} 0 & \text{if } \frac{1}{\ell} \sum_{i=1}^{\ell} \left(f(\mathbf{x}^i) - \frac{1}{\ell} \sum_{i=j}^{\ell} f(\mathbf{x}^j) \right)^2 = 1 \\ \infty & \text{otherwise.} \end{array} \right. \tag{A.34}$$

SVM: The SVM is two-class $\{-1; 1\}$ classification problem, which try to maximize the margin $\|f\|$. Here

$$g_{emp}\left((\mathbf{x}^i, y_i, f(\mathbf{x}^i))_{i\in\{1,\ldots,\ell\}}\right) = \sum_{i=1}^{\ell} \max\left(0, 1 - y_i f(\mathbf{x}^i)\right) \tag{A.35}$$

and the regularizer

$$g_{reg} = \lambda\|f\|^2. \tag{A.36}$$

A.4 The minimal enclosing hypersphere

In this section, we present an example taking benefit of the kernel methods. Suppose we have samples in some Euclidean space $\mathbf{x}^i \in \mathbb{R}^n, \forall i \in [1,\ell]$, and a positive semi-definite real-valued kernel k that maps $\mathbb{R}^n \to \mathcal{H}$ such as $\mathbf{x} \mapsto \Phi(\mathbf{x})$. We want to find the smallest hypersphere $S(\mathcal{C}_S, \mathcal{R}_S)$ that contains $\Phi(\mathbf{x})$, *i.e.*, we want to find the center $\mathcal{C}_S$ such as the radius is minimal:

$$\min_{\mathcal{C}_s}(\mathcal{R}_S) = \min_{\mathcal{C}_s}\left(\max_i\left(\|\Phi(\mathbf{x}^i) - \mathcal{C}_S\|_{\mathcal{H}}\right)\right) \tag{A.37}$$

The previous equation (A.37) can be rewritten as:

$$\begin{aligned}
\min_{\mathcal{C}_s}\left(\mathcal{R}_S^2\right) &= \min_{\mathcal{C}_s}\left(\max_i\left(\langle\Phi(\mathbf{x}^i) - \mathcal{C}_S, \Phi(\mathbf{x}^i) - \mathcal{C}_S\rangle_{\mathcal{H}}\right)\right)\\
&= \min_{\mathcal{C}_s}\left(\max_i\left(\langle\Phi(\mathbf{x}^i), \Phi(\mathbf{x}^i)\rangle_{\mathcal{H}} + \langle\mathcal{C}_S, \mathcal{C}_S\rangle_{\mathcal{H}} - 2\langle\Phi(\mathbf{x}^i), \mathcal{C}_S\rangle_{\mathcal{H}}\right)\right)\\
&= \min_{\mathcal{C}_s}\left(\max_i\left(\|\Phi(\mathbf{x}^i)\|_{\mathcal{H}}^2 - 2\langle\Phi(\mathbf{x}^i), \mathcal{C}_S\rangle_{\mathcal{H}}\right) + \|\mathcal{C}_S\|_{\mathcal{H}}^2\right).
\end{aligned} \tag{A.38}$$

We can identify with equation (A.26):

$$g_{reg}(\|\mathcal{C}_S\|_{\mathcal{H}}) = \|\mathcal{C}_S\|_{\mathcal{H}}^2 \tag{A.39}$$

and

$$g_{emp}\left((\mathbf{x}^i, y_i, \mathcal{C}_S(\mathbf{x}^i))_{i\in\{1,\ldots,\ell\}}\right) = \max_i\left(\|\Phi(\mathbf{x}^i)\|_{\mathcal{H}}^2 - 2\langle\Phi(\mathbf{x}^i), \mathcal{C}_S\rangle_{\mathcal{H}}\right). \tag{A.40}$$

From the Representer Theorem we can say that $\mathcal{C}_s$ admits a representation of the form:

$$\mathcal{C}_s = \sum_{i=1}^{\ell} \beta_i k(\cdot, \mathbf{x}^i). \tag{A.41}$$

The above problem can be considered as a constraint optimization problem (note now the Representer Theorem does not applied anymore):

$$\begin{aligned}
&\min \mathcal{R}_S^2\\
\text{subject to}\quad &\|\Phi(\mathbf{x}^i) - \mathcal{C}_S\|_{\mathcal{H}}^2 - \mathcal{R}_S^2 \leq 0, \forall i \in [1,\ell]
\end{aligned} \tag{A.42}$$

Its associated *Lagrangian* is ($\beta_i \in \mathbb{R}^+$)

$$L(\mathcal{R}_S, \mathcal{C}_S, \boldsymbol{\beta}) = \mathcal{R}_S^2 + \sum_{i=1}^{\ell} \beta_i\left(\|\Phi(\mathbf{x}^i) - \mathcal{C}_S\|_{\mathcal{H}}^2 - \mathcal{R}_S^2\right). \tag{A.43}$$

According to the *Lagrange* theory, L has to be minimized with ratio to the primal variable $\mathcal{R}_S$, $\mathcal{C}_S$ and to be maximized with ratio to the dual variable $\boldsymbol{\beta}$ (the Lagrange multipliers). The objective and the constraint functions are convex and differentiable. We can write the *Karush-Kuhn-Tucker* (KKT) optimality conditions [18]:

$$\left\{\begin{array}{rcl} \dfrac{\partial L}{\partial \mathcal{R}_s} = 2\mathcal{R}_s + \displaystyle\sum_{i=1}^{\ell} \beta_i(-2\mathcal{R}_s) & = & 0 \\ \dfrac{\partial L}{\partial \mathcal{C}_s} = -2\displaystyle\sum_{i=1}^{\ell} \beta_i(\Phi(\mathbf{x}^i) - \mathcal{C}_S) & = & 0 \\ \|\Phi(\mathbf{x}^i) - \mathcal{C}_S\|^2_{\mathcal{H}} - \mathcal{R}^2_S & \leq & 0,\ \forall i \in [1, \ell] \\ \beta_i & \geq & 0,\ \forall i \in [1, \ell] \\ \beta_i \left(\|\Phi(\mathbf{x}^i) - \mathcal{C}_S\|^2_{\mathcal{H}} - \mathcal{R}^2_S\right) & = & 0,\ \forall i \in [1, \ell] \end{array}\right. \tag{A.44}$$

From the first equalities, we have $\sum_{i=1}^{\ell} \beta_i = 1$ and $\mathcal{C}_s = \sum_{i=1}^{\ell} \beta_i \Phi(\mathbf{x}^i)$. The Lagrangian can be rewritten as:

$$\begin{aligned} L(\mathcal{R}_S, \mathcal{C}_S, \boldsymbol{\beta}) &= \sum_{i=1}^{\ell} \beta_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^i)\rangle_{\mathcal{H}} - 2\sum_{i=1}^{\ell} \beta_i \langle \Phi(\mathbf{x}^i), \mathcal{C}_S\rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \beta_i \langle \mathcal{C}_S, \mathcal{C}_S\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{\ell} \beta_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^i)\rangle_{\mathcal{H}} - 2\sum_{i,j=1}^{\ell} \beta_i\beta_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j)\rangle_{\mathcal{H}} + \sum_{i,j=1}^{\ell} \beta_i\beta_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j)\rangle_{\mathcal{H}} \\ &= \sum_{i=1}^{\ell} \beta_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^i)\rangle_{\mathcal{H}} - \sum_{i,j=1}^{\ell} \beta_i\beta_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j)\rangle_{\mathcal{H}}. \end{aligned} \tag{A.45}$$

We finally have the dual formulation of the problem. Neither of the primal variables appear in the Lagrangian. The min-max problem is reduced to a maximization problem:

$$\begin{aligned} \max_{\beta} G(\boldsymbol{\beta}) = \ & \sum_{i=1}^{\ell} \beta_i \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^i)\rangle_{\mathcal{H}} - \sum_{i,j=1}^{\ell} \beta_i\beta_j \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j)\rangle_{\mathcal{H}} \\ \text{subject to} \ & \beta_i \geq 0 \\ & \sum_{i=1}^{\ell} \beta_i = 1. \end{aligned} \tag{A.46}$$

This can be directly done in the input space using the kernel: $k(\mathbf{x}^i, \mathbf{x}^j) = \langle \Phi(\mathbf{x}^i), \Phi(\mathbf{x}^j)\rangle_{\mathcal{H}}$:

$$\begin{aligned} \max_{\beta} G(\boldsymbol{\beta}) = \ & \sum_{i=1}^{\ell} \beta_i k(\mathbf{x}^i, \mathbf{x}^i) - \sum_{i,j=1}^{\ell} \beta_i\beta_j k(\mathbf{x}^i, \mathbf{x}^j) \\ \text{subject to} \ & \beta_i \geq 0 \\ & \sum_{i=1}^{\ell} \beta_i = 1. \end{aligned} \tag{A.47}$$

The KKT conditions say also that the duality gap is null, *i.e.*, optimal $\tilde{L}$ is equal to optimal $\tilde{G}$, and the last equality of (A.44) implies: $\tilde{L} = \mathcal{R}^2_S$. Finally, solving in the input space the convex constraint optimization problem (A.47) provides the hypersphere parameters : $(\mathcal{R}_S, \mathcal{C}_S)$.

As numerical experiments, we generated 200 samples from a two dimensional Gaussian $\mathcal{N}(\boldsymbol{\mu}, \sigma\mathbf{I})$ with $\boldsymbol{\mu} = \Big[0.65, 0.65\Big]^t$ and $\sigma = 0.1$. The minimal enclosing hypersphere were computed in the feature space induces by a polynomial kernel of degree 2, see (A.25). The intersection of the hypersphere and the kernel feature space was computed. Results are presented in figure A.3. The radius found in the input space was 0.29 while in the feature space it was 0.50. Note that the radius is the Euclidean distance between the center of the hypersphere and the more distant projected element, it is not the geodesic distance along the surface defined by the kernel feature space.

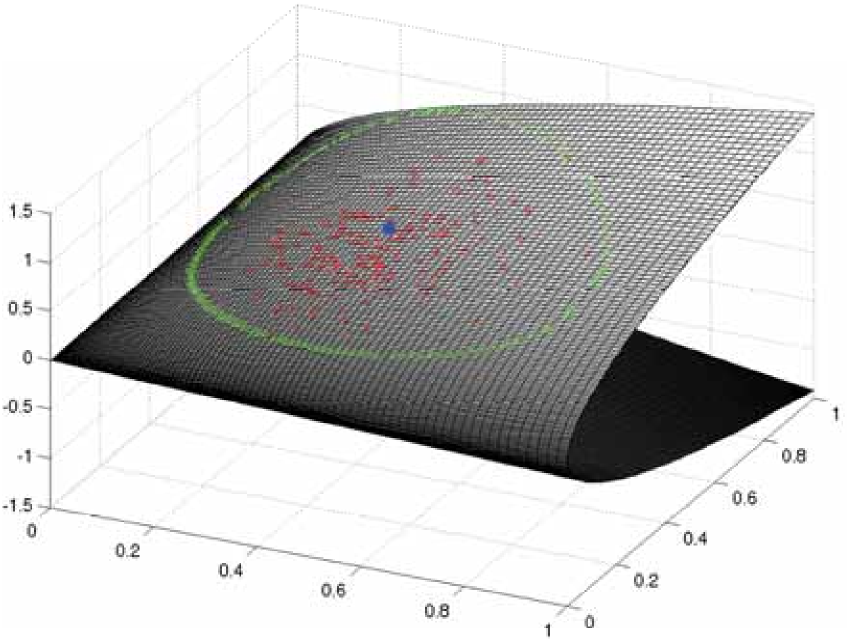


Figure A.3: *Intersection of the kernel feature space and the minimal enclosing hypersphere: The blue point is the center $\mathcal{C}_x$, the green stars are the intersection of the kernel feature space with the hypersphere and the red cross are the initial sample* $\mathbf{x}^i$.

A.5 Conclusions

Basics of kernel methods have been reviewed. The equivalence between a positive semi-definite function and reproducing Hilbert space has been stated with the Moore and Aronszajn's theorem. The problem of feature space and kernel feature space has been addressed with the Representer Theorem.

These theorems have very important practical consequences: if on a given set we can define a positive semi-definite function, we can use all algorithms which can be stated in an inner product form. More, the definition of a suitable kernel for our problem can improve the performance of many algorithm.

The example given in the previous section presents the classical work-flow to use kernel method: first express the problem in inner product form and then define the kernel functions. We chose to present the minimal enclosing hypersphere because of its applications to many kernel based algorithms. In this thesis, we have used it to estimate kernel's parameter in the KPCA, and to estimate errors bounds in the SVM. Furthermore, the optimization problem is very similar to the SVMs one.

Appendix B

Assessing the accuracy

THE ESTIMATION of the classification accuracy is based on the *confusion matrix*. From that matrix it is possible to evaluate the exactitude of a given classification map by comparison to the reference map. Several estimates, from global estimation to specific estimation are extracted from the confusion matrix. They are detailed in the following as well as the confusion matrix.

Definition B.1 (Confusion matrix) *In the field of artificial intelligence, a confusion matrix is a visualization tool typically used in supervised learning . Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. One benefit of a confusion matrix is that it is easy to see where the system is confusing (*i.e., *commonly mis-labelling one class as another).*

An example of confusion matrix is given Table B.1 for a 3-classes problem. C_i represents the class i and C_{ij} is the number of pixels assign to the class j by the classifier which are referenced as class i.

Definition B.2 (Overall Accuracy) *The* overall accuracy *(OA) is the percentage of correctly classified pixels:*

$$OA = \frac{\sum_i^{N_c} C_{ii}}{\sum_{ij}^{N_c} C_{ij}} \times 100. \tag{B.1}$$

Definition B.3 (Class Accuracy) *The* Class Accuracy *(or producer's accuracy) (CA) is the percentage of correctly classified pixels for a given class.*

$$CA_i = \frac{C_{ii}}{\sum_j^{N_c} C_{ij}} \times 100. \tag{B.2}$$

Definition B.4 (Average Accuracy) *The* average accuracy *(AA) is the mean of class accuracy for all the classes.*

$$AA = \frac{\sum_i^{N_c} CA_i}{N_c} \times 100. \tag{B.3}$$

An OA or an AA is closed to 100% (0%) means that the classification accuracy is almost perfect (wrong). When a referenced set is unbalanced, the OA may not be representative of the true performance of the classifier. For instance, if a class has very few number of referenced pixels, its influence will be very low in the computation of the OA, while it will be more influent in the AA since the mean is done the number of classes rather than the whole number of pixels. Strong difference between OA and AA may indicate that a specific class is wrongly classified with a high proportion.

Definition B.5 (Kappa Coefficient) *The* Kappa Coefficient *(κ) is a statistical measure of agreement. It is the percentage agreement corrected by the level of agreement that could be expected due to chance*

Table B.1: *Confusion Matrix, N is the number of referenced pixel and N_c is the number of classes.*

Percentage	Classification Data				
Reference Data	C_1	C_2	C_3	Row Total	Producer's Accuracy
C_1	C_{11}	C_{12}	C_{13}	$\sum_i^{N_c} C_{1i}$	$\frac{C_{11}}{\sum_i^{N_c} C_{1i}}$
C_2	C_{21}	C_{22}	C_{23}	$\sum_i^{N_c} C_{2i}$	$\frac{C_{22}}{\sum_i^{N_c} C_{2i}}$
C_3	C_{31}	C_{22}	C_{33}	$\sum_i^{N_c} C_{3i}$	$\frac{C_{33}}{\sum_i^{N_c} C_{3i}}$
Column Total	$\sum_i^{N_c} C_{i1}$	$\sum_i^{N_c} C_{i2}$	$\sum_i^{N_c} C_{i3}$	N	
User's Accuracy	$\frac{C_{11}}{\sum_i^{N_c} C_{i1}}$	$\frac{C_{11}}{\sum_i^{N_c} C_{i2}}$	$\frac{C_{33}}{\sum_i^{N_c} C_{i3}}$		

alone. It is generally thought to be a more robust measure than simple percent agreement calculation since κ takes into account the agreement occurring by chance.

$$\begin{aligned}
\kappa &= \frac{P_o - P_e}{1 - P_e} \\
P_o &= OA \\
P_e &= \frac{1}{N^2}\sum_i^{N_c} C_i.C._i \\
C_{i\cdot} &= \sum_j^{N_c} C_{ij} \\
C_{\cdot i} &= \sum_j^{N_c} C_{ji}
\end{aligned} \tag{B.4}$$

Total agreement is achieved if $\kappa = 1$ while there are no agreement when $\kappa \leq 0$.

Appendix C

Data Set

Contents

THIS chapter is dedicated to the description of the data used in this thesis. To avoid repetition, all the information are sum up in the following. Two types of remote sensing data were available: one very high resolution panchromatic real IKONOS image, two very high resolution panchromatic simulated PLEIADES images and three hyperspectral real data set. Each data set was acquired over urban area. Next, spectral coverage, spatial resolution, training and testing set are given.

C.1 Hyperspectral data

C.1.1 ROSIS data

Airborne data from the ROSIS-03 (Reflective Optics System Imaging Spectrometer) optical sensor are used for the experiments. The flight over the city of Pavia, Italy, was operated by the Deutschen Zentrum fur Luft- und Raumfahrt (DLR, the German Aerospace Agency) in the framework of the HySens project, managed and sponsored by the European Union. According to specifications the number of bands of the ROSIS-03 sensor is 115 with a spectral coverage ranging from 0.43 to 0.86μm. The spatial resolution is 1.3m per pixel. Two data set were available: the *University area* and the *Pavia Center.*

C.1.1.1 University Area

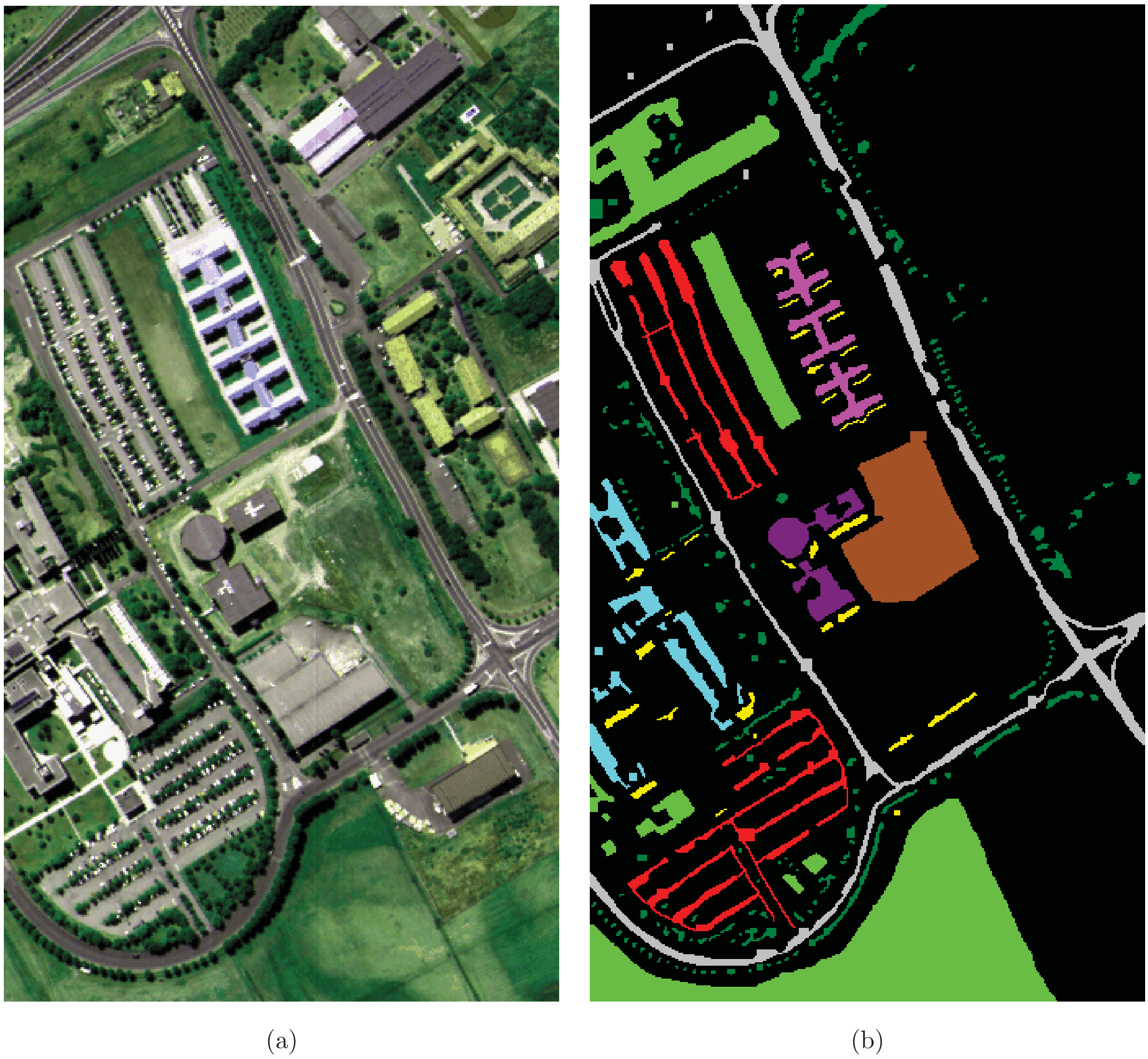
The original data set is 610 by 340 pixels. Some channels (12) have been removed due to noise. The remaining 103 spectral dimensions are processed. Nine classes of interest are considered, namely: trees, asphalt, bitumen, gravel, metal sheet, shadow, bricks, meadow and soil. The image was acquired around the Engineering School at the University of Pavia. False color image is presented in Figure C.1.(a) and the available testing set in Figure C.1.(b). Testing and training set are detailed in Table C.1.

C.1.1.2 Pavia Center

The second ROSIS data set is the center of Pavia. The Pavia center image was originally 1096 by 1096 pixels. A 381 pixel wide black in the left part of image was removed, resulting in a “two part” image. This “two part” image is 1096 by 715 pixels. Some channels (13) have been removed due to noise. The remaining 102 spectral dimensions are processed. Nine classes of interest are considered, namely: water, tree, meadow, brick, soil, asphalt, bitumen, tile and shadow. False color image is presented in

Table C.1: *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



(a)

(b)

Figure C.1: *ROSIS University Area: (a) three-channel color composite and (b) available reference data: asphalt, meadow, gravel, tree, metal sheet, bare soil, bitumen, brick and shadow.*

Table C.2: *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 C.3: *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

Figure C.2.(a) and the available testing set in Figure C.2.(b). Testing and training set are detailed in Table C.2.

C.1.2 HYDICE data

Airborne data from the HYDICE sensor (Hyperspectral Digital Imagery Collection Experiment) was used for the experiments. The HYDICE was used to collect data from flightline over the Washington DC Mall. Hyperspectral HYDICE data originally contained two hundred and ten bands in the 0.4-2.4μm region. Noisy channels have been removed and the set consists of 191 spectral channels. It was 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. False color images is presented in Figure C.3.(a) and the available testing set in Figure C.3.(b). Testing and training set are detailed in Table C.3.

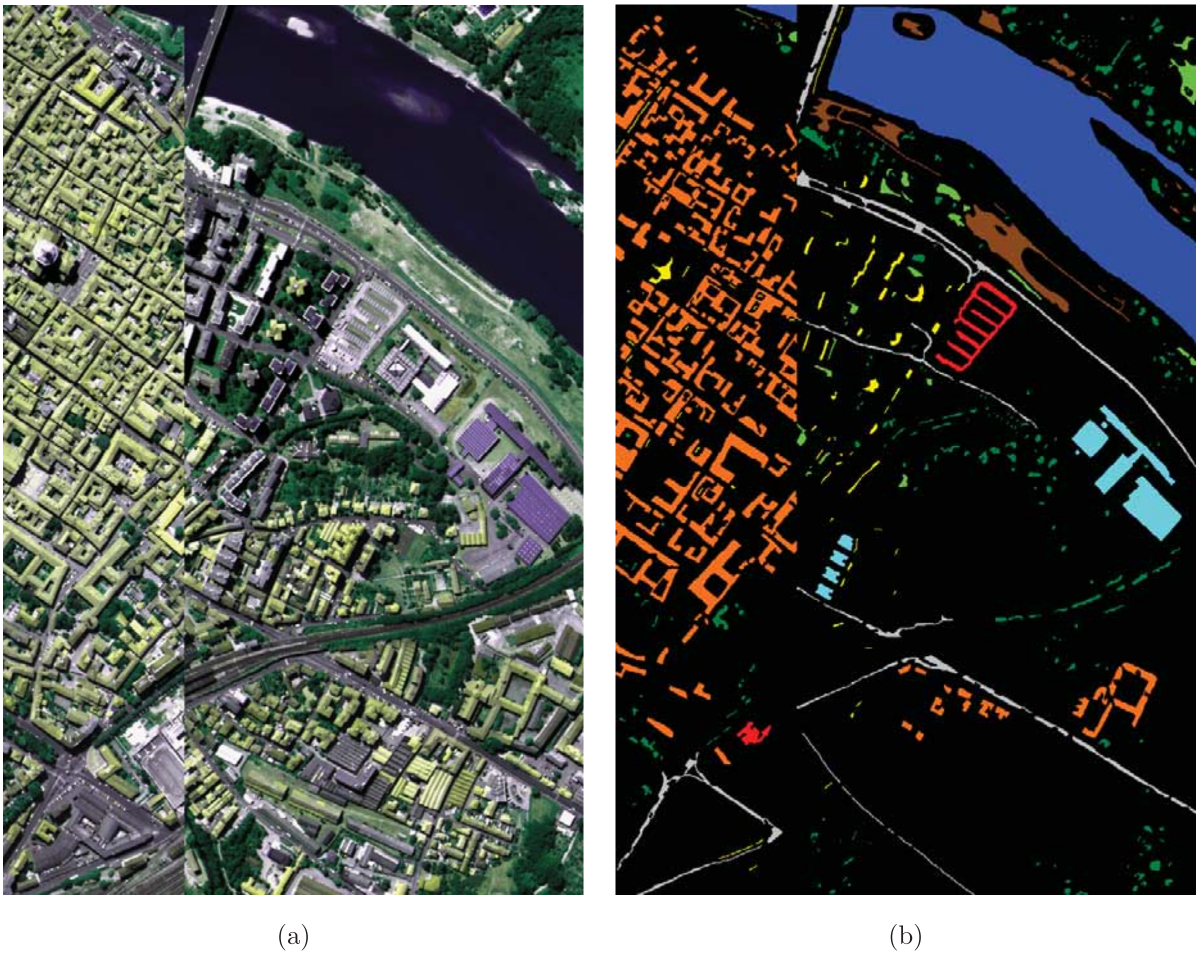
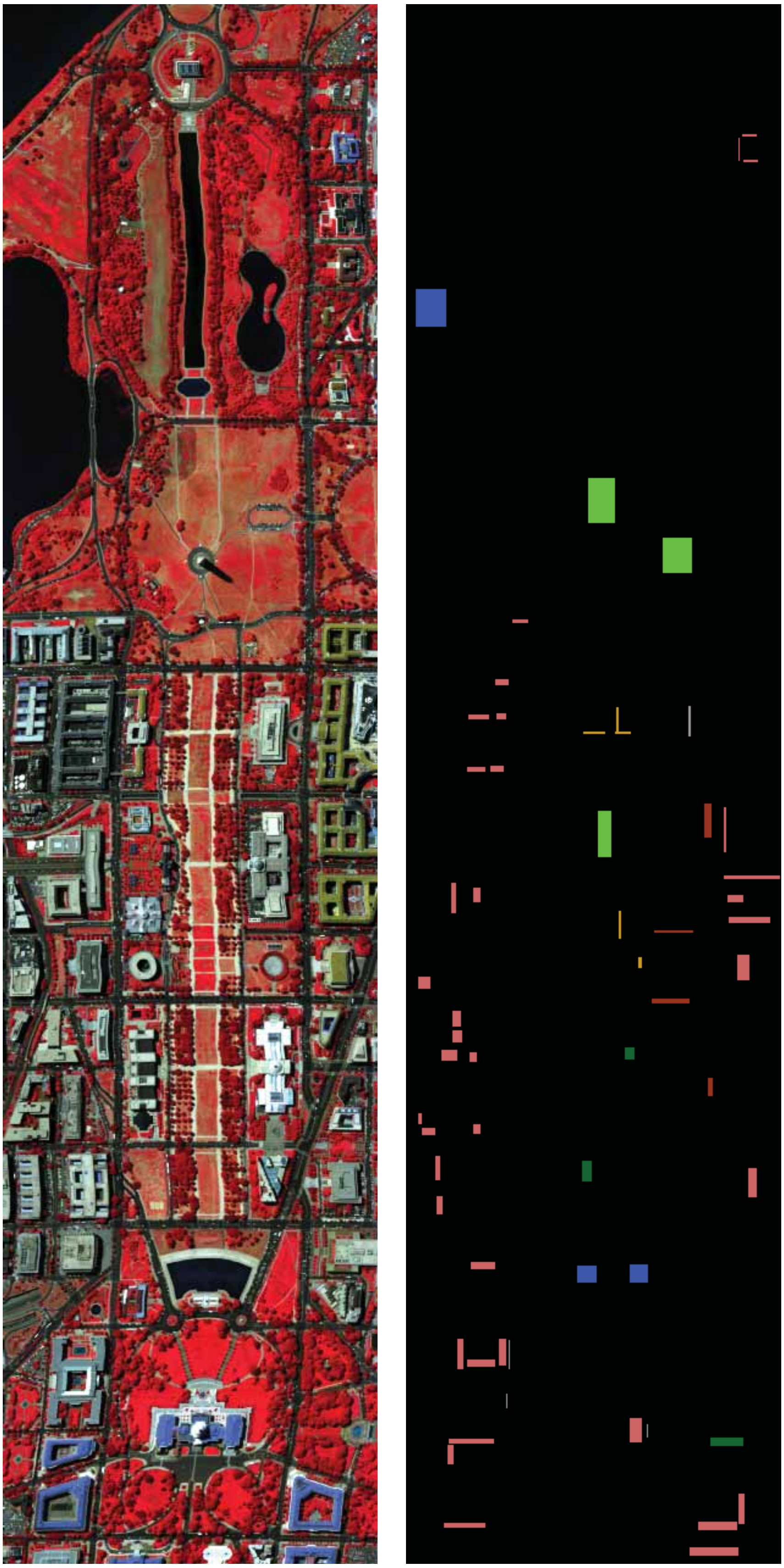
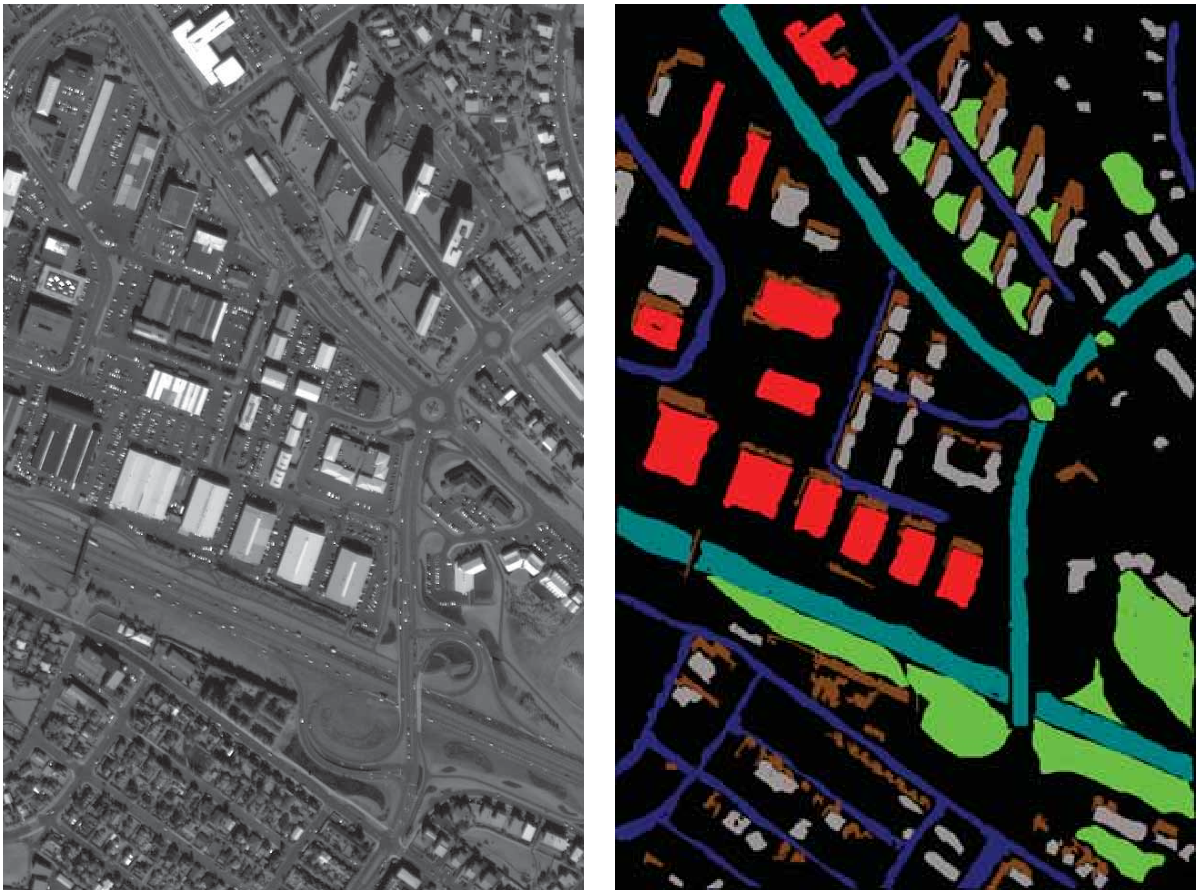


Figure C.2: *ROSIS Pavia Center: (a) three-channel color composite and (b) available reference data: water, trees, meadow, brick, soil, asphalt, bitumen, tile and shadow.*



(a) (b)

Figure C.3: *HYDICE Washington DC Mall: (a) three-channel color composite and (b) available reference data: roof, road, trail, grass, tree, water and shadow.*



(a) (b)

Figure C.4: *IKONOS Reykjavik 1 (a): grayscale panchromatic image and (b) available reference data: large building, small building, street, residential lawn, open area* and *shadow.*

C.2 Panchromatic data

C.2.1 IKONOS data

Two IKONOS data from Reykjavik, Iceland set were used. They are very high resolution panchromatic images of 1m resolution with a spectral coverage from 0.45 to 0.90 μm . Six classes were considered in each case, namely: large building, small building, residential lawn, street, open area and shadow.

C.2.1.1 Reykjavik 1

The first IKONOS images is 975 by 639 pixels. Original data and testing set are in Figure C.4. Testing and training set are detailed in Table C.4.

C.2.1.2 Reykjavik 2

The first IKONOS images is 700 by 630 pixels. Original data and testing set are in Figure C.5. Testing and training set are detailed in Table C.5.

Table C.4: *Information classes and training-test samples for the* IKONOS Reykjavik 1 *data set.*

Class		Samples	
No	Name	Train	Test
1	Small Building	1526	34155
2	Open area	7536	25806
3	Shadows	1286	43867
4	Large Building	2797	39202
5	Street	3336	30916
6	Residential Lawns	5616	35147
Total		22.097	209.093

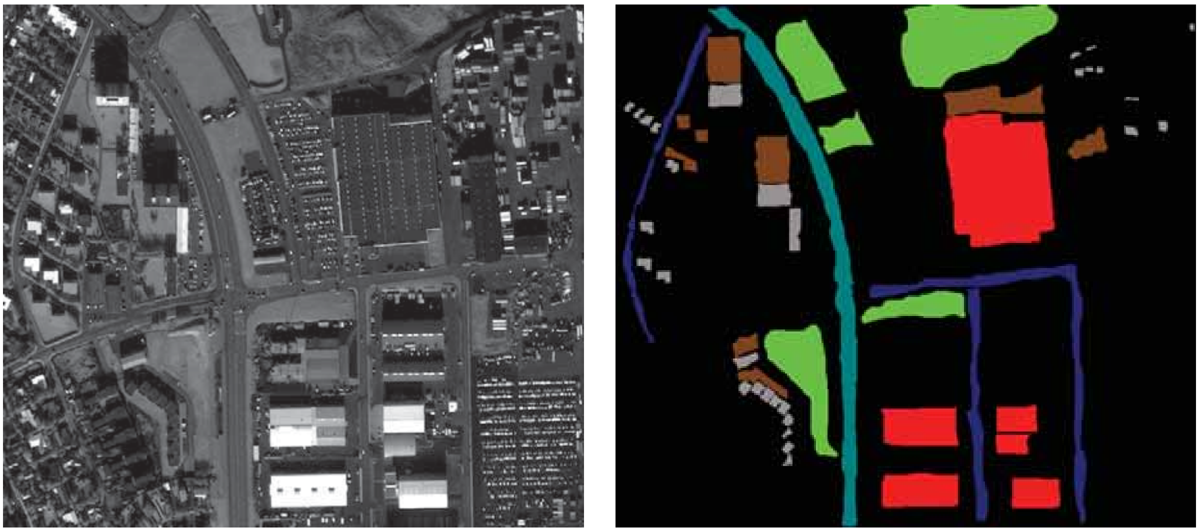

(a) (b)

Figure C.5: *IKONOS Reykjavik 2 (a): grayscale panchromatic image and (b) available reference data: large building, small building, street, residential lawn, open area* and *shadow.*

Table C.5: *Information classes and training-test samples for the* IKONOS Reykjavik 2 *data set.*

Class		Samples	
No	Name	Train	Test
1	Small Building	1963	6213
2	Open area	6068	28144
3	Shadows	2619	10610
4	Large Building	5599	29768
5	Street	2489	11940
6	Residential Lawns	4103	12066
Total		22.741	98.701

Table C.6: *Information classes and training-test samples for the* PLEIADES Toulouse 1, Toulouse 2 *and* Perpignan *data set.*

	Toulouse 1		Toulouse 2		Perpignan	
	train	test	train	test	train	test
1	780	2450	393	905	864	1665
2	845	2293	355	1005	172	1327
3	798	2588	518	1104	136	446
4	1738	3886	96	583	343	1776
Total	4161	11217	1362	3597	1515	5212

C.2.2 PLEIADES data

The data set consists of three panchromatic images extracted from simulated PLIEADES images provided by CNES (satellite to be launched in 2008). The spatial resolution resolution is 0.75 meter by pixel. All images are urban areas. Four classes were considered in each case, namely: building, street, open area and shadow.

C.2.2.1 Toulouse 1

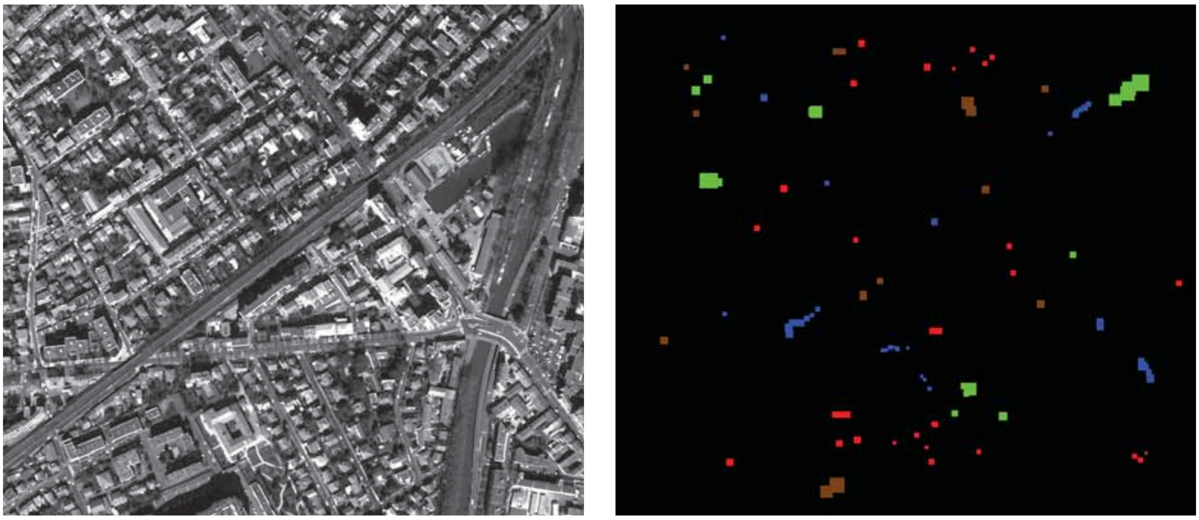
The image consists in 886 by 780 pixels. It has been acquired over the city of Toulouse, France. Original data and testing set are in Figure C.6. Testing and training set are detailed in Table C.6.

C.2.2.2 Toulouse 2

The image consists in 602 by 540 pixels. It has been acquired over the city of Toulouse, France. Original data and testing set are in Figure C.7. Testing and training set are detailed in Table C.6.

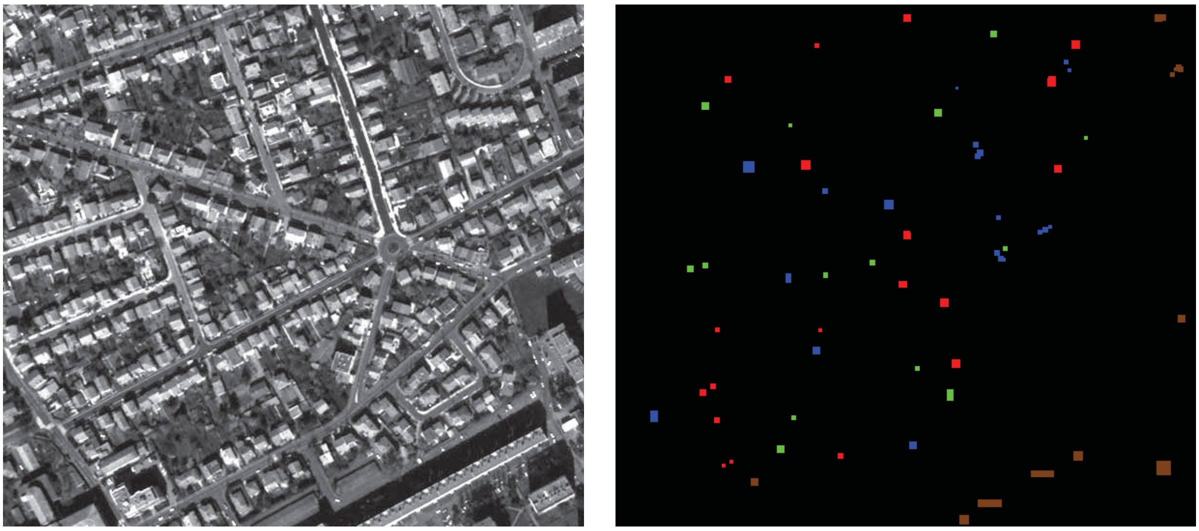
C.2.2.3 Perpignan

The image consists in 732 by 746 pixels. It has been acquired over the city of Perpignan, France. Original data and testing set are in Figure C.8. Testing and training set are detailed in Table C.6.



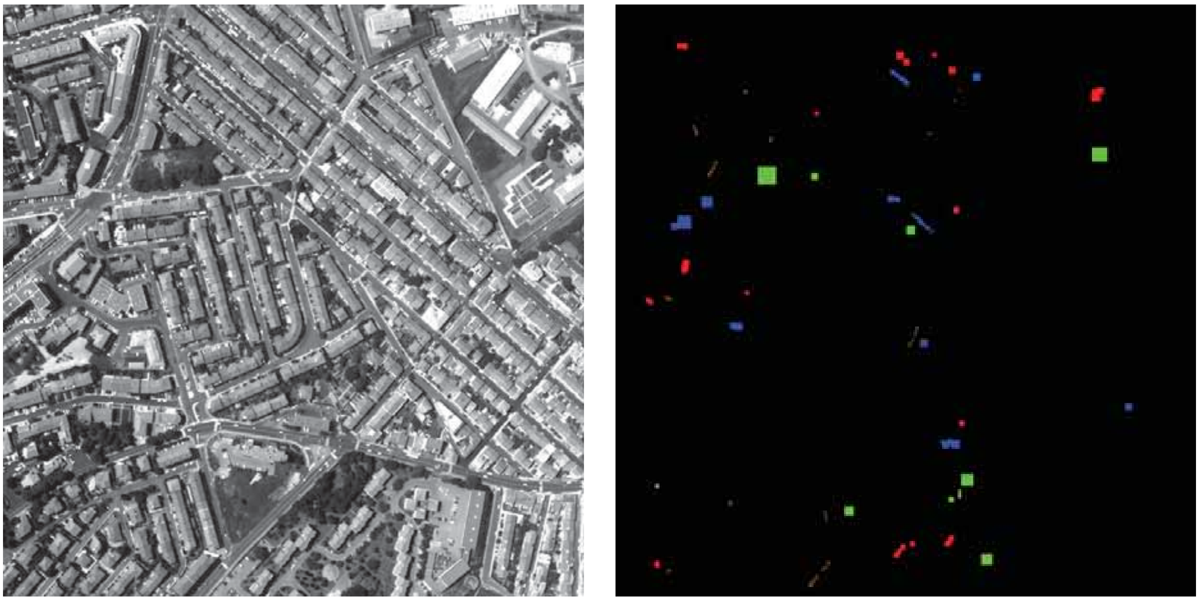
(a) (b)

Figure C.6: *PLEIADES Toulouse 1 (a): grayscale panchromatic image and (b) available reference data: buildings, Road, Open Area* and *shadow*.



(a) (b)

Figure C.7: *PLEIADES Toulouse 2 (a): grayscale panchromatic image and (b) available reference data: buildings, Road, Open Area* and *shadow*.



(a) (b)

Figure C.8: *PLEIADES Perpignan (a): grayscale panchromatic image and (b) available reference data: buildings, Road, Open Area* and *shadow.*

Index

Bibliography

[1] F. D. Acqua, P. Gamba, A. Ferrari, J. A. Palmason, and J. A. Benediktsson. Exploiting spectral and spatial information in hyperspectral urban data with high resolution. *IEEE Geoscience and Remote Sensing Letters*, 1(4):322–326, October 2004.

[2] R. Adams and L. Bischof. Seede region growing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[3] G. Amici, F. Dell'Acqua, P. Gamba, and G. Pulina. A comparison of fuzzy and neuro-fuzzy data fusion for flooded area mapping using SAR images. *International Journal of Remote Sensing*, 25(20):4425–4430, October 2004.

[4] N. Aronszajn. Theory of reprodusing kernel. Technical Report 11, Harvard University, Division of engineering sciences, 1950.

[5] Y. Bazi and F. Melgani. Toward an optmal svm classification system for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3374–3385, November 2006.

[6] J. A. Benediktsson and I. Kanellopoulos. Classification of multisource and hyperspectral data based on decision fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 37:1367–1377, May 1999.

[7] J. A. Benediktsson, J. A. Palmason, and J.R. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, March 2005.

[8] J. A. Benediktsson, M. Pesaresi, and K. Arnason. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, 41(9):1940–1949, September 2003.

[9] J. A. Benediktsson, J. R. Sveinsson, O. K. Ersoy, and P. H. Swain. Parallel consensual neural network. *IEEE Transactions on Neural Networks*, 8:54–65, January 1997.

[10] J. A. Benediktsson, J. R. Sveinsson, and P. H. Swain. Hybrid consensus theoric classification. *IEEE Transactions on Geoscience and Remote Sensing*, 35:833–843, July 1997.

[11] J. A. Benediktsson and P. H. Swain. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 28:540–552, July 1990.

[12] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy. Neural network approaches versus statistical methods in classification of multisource remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 28:540–552, July 1990.

[13] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy. Conjugate-gradiant neural networks in classification of multisource remote sensing data. *International Journal of Remote Sensing*, 14(15):2883–2903, 1993.

[14] I. Bloch. Information combination operators for data fusion: A comparative review with classification. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 26(1):52–67, January 1996.

[15] I. Bloch. *Fusion d'informations en traitement du signal et des images.* Germes LAVOISIER, 11 rue Lavoisier, 75008 Paris, 2003.

[16] L. Bottou and C.-J. Lin. Support vector machine solvers. Technical report, 2006. Software available at http://leon.bottou.org/papers/bottou-lin-2006.

[17] F. Bovolo, L. Bruzzone, and M. Marconcini. A novel context-sensitive SVM for classification of remote sensing images. In *Geoscience and Remote Sensing Symposium.* IGARSS '06. Proceedings, July 2006.

[18] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University press, 2006.

[19] L. Bruzzone, M. Chi, and M. Marconcini. A novel transductive SVM for semisupervised classification remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11):3363–3373, November 2006.

[20] G. Camps-Valls, L. Gomez-Chova, J. Calpe-Maravilla, J. D. Martin-Guerrero, E. Soria-Olivas, L. Alonso-Chorda, and J. Moreno. Robust support vector method for hyperspectral data classification and knowledge discovery. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7), July 2004.

[21] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Francés, and J. Calpe-Maravilla. Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, 3(1):93–97, January 2006.

[22] Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. Word sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082, Febrary 2003.

[23] C. Chang. *Hyperspectral imaging. Techniques for spectral detection and classification.* Kluwer Academic, 2003.

[24] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at *http://www.csie.ntu.edu.tw/ cjlin/libsvm.*

[25] Y.-L. Chang, L.-S. Liang, C.-C. Han, J.-P. Fang, W.-Y. Liang, and K.-S. Chen. Multisource data fusion for landslide classification using genralized positive boolean functions. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1697–1708, June 2007.

[26] J. Chanussot. *Approches vectorielles ou marginales pour le traitement d'images multi-composantes.* PhD thesis, Universite de Savoie, France, 1998.

[27] J. Chanussot, J. A. Benediktsson, and M. Fauvel. Classification of remote sensing images from urban areas using a fuzzy possibilistic model. *IEEE Geoscience and Remote Sensing Letters*, 3(1):40–44, January 2006.

[28] J. Chanussot, J. A. Benediktsson, and M. Pesaresi. On the use of morphological alternated sequential filters for the classification of remote sensing images from urban areas. In *Proc. IEEE Geoscience and Remote Sensing Symposium.* IGARSS '03. Proceedings, July 2003.

[29] J. Chanussot and P. Lambert. Total ordering based on space filling curves for multivalued morphology. In *4th International Symposium on Mathematical Morphology and its Applications*, pages 51–58, June 1998.

[30] J. Chanussot, G. Mauris, and P. Lambert. Fuzzy fusion techniques for linear features detection in multitemporal SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3):1292–1305, May 1999.

[31] O. Chapelle. *Support Vector Machines: Induction Principle, Adaptive Tuning and Prior Knowledge.* PhD thesis, Université Pierre et Marie Curie, 2002.

[32] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–1178, March 2007.

[33] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines of histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, September 1999.

[34] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning.* Adaptive computation and machine learning. MIT Press, Cambridge, Mass., USA, September 2006.

[35] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, 2002.

[36] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vectors machines. *Machine Learning*, 46(1), 2002.

[37] M. Chi and L. Bruzzone. Semisupervised classification of hyperspectral images by svms optimized in the primal. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1870–1880, June 2007.

[38] J. Crespo, J. Serra, and R. W. Schafer. Theoretical aspects of morphological filters by reconstruction. *Signal Processing*, 47(2):201–225, November 1995.

[39] N. Cristianni and J. Shawe-taylor. *An introduction to support vector machines.* Cambridge, University press, 2000.

[40] A. DeLuca and S. Termini. A definition of nonprobabilistic entropy in the setting of fuzzy set theory. *Information and Control*, 20:301–312, 1972.

[41] P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The annals of statistics*, 12(3):793–815, 1984.

[42] K. I. Dimantaras and J. T. Kung. *Principal component neural networks.* New York: Wiley, 1996.

[43] D. L. Donoho. High-dimensional data analysis: the curses and blessing of dimensionality. In *AMS Mathematical challenges of the 21st century*, 2000.

[44] D. Dubois and H. Prade. *Combination of information in the framework of possibility theory.* New York: Academic, January 1992.

[45] B. R. Ebanks. On measures of fuzziness and their representations. *J.Math. Analysis and Application*, 94:24–37, 1983.

[46] M. Fauvel, J. Chanussot, and J. A. Benediktsson. Decision fusion for the classification of urban remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10), October 2006.

[47] M. Fauvel, J. Chanussot, and J. A. Benediktsson. Evaluation of kernels for multiclass classification of hyperspectral remote sensing data. In *IEEE International Conference on Acoustics, Speech and Signal Processing.* ICASSP'06. Proceedings, May 2006.

[48] M. Fauvel, J. Chanussot, and J. A. Benediktsson. Kernel principal component analysis for feature reduction in hyperspectral images analysis. In *IEEE* 7^{th} *Nordic Signal Processing Symposium*, 2006.

[49] M. Fauvel, J. Chanussot, and J. A. Benediktsson. *Decision fusion for hyperspectral classification, in Hyperspectral Data Exploitation.* John Wiley and Sons, New York, 2007.

[50] M. Fauvel, J. Chanussot, and J. A. Benediktsson. How transferable are spatial features for the classification of very high resolution remote sensing data. In *Urban remote sensing joint event 2007*. URBAN 2007, April 2007.

[51] M. Fauvel, J. Chanussot, and J. A. Benediktsson. A joint spatial and spectral SVM's classification of panchromatic images. In *Proc. IEEE Geoscience and Remote Sensing Symposium*. IGARSS '07. Proceedings, July 2007.

[52] 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, July 2007.

[53] M. Fauvel, J. A. Palmason, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson. Classification of remote sensing imagery with high spatial resolution. In *SPIE*, 2005.

[54] G. F. Foody and Ajay Mathur. A relative evaluation of multiclass image classification by support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42:1335–1343, June 2004.

[55] G. M. Foody. The significance of border training patterns in classification by a feedforward neural network using back propagation learning. *International Journal of Remote Sensing*, 20(18), December 1999.

[56] G. M. Foody. Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy. *Photogrammetric Engineering & Remote Sensing*, 70(5):627–633, May 2004.

[57] G. M. Foody and A. Mathur. The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM. *Remote Sensing of Environment*, 103(2), July 2006.

[58] G. M. Foody, A. Mathur, C. Sanchez-Hernandez, and D. S. Boyd. Training set size requirements for the classification of a specific class. *Remote Sensing of Environment*, 104(1), September 2006.

[59] K. Fukunaga. *Introduction to statistical pattern recognition.* CA: Academic Press, San Diego, 1990.

[60] J. Goutsias, H. Heijmans, and K. Sivakumar. Morphological operators for image sequences. *Computer vision and image understanding*, 62(3):326–346, 1995.

[61] J. A. Gualtieri and S. Chettri. Support vector machines for classification of hyperspectral data. In *Proc. IEEE International Geoscience and Remote Sensing Symposium*, volume 2. IGARSS '00. Proceedings, July 2000.

[62] P. Hall and K. Li. On almost linearity of low dimensional projections from high dimensional data. *The annals of statistics*, 21(2):867–889, 1993.

[63] G. H. Halldorsson, J. A. Benediktsson, and J. R. Sveinsson. Support vector machines in multisource classification. In *Geoscience and Remote Sensing Symposium*, volume 3. IGARSS '03. Proceedings, July 2003.

[64] R. Herbrich. *Learning kernel classifiers: Theory and algorithms.* MIT Press, 2002.

[65] C. W. Hsu and C. J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13:415–425, March 2002.

[66] G. F. Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, IT-14:55–63, January 1968.

[67] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent component analysis.* Wiley Series, 2001.

[68] Q. Jackson and D. A. Landgrebe. Adaptive bayesian contextual classification based on markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, 40(11):11, November 2002.

[69] B. Jeon and D. A. Landgrebe. Decision fusion approach for multitemporal classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37:1227–1233, May 1999.

[70] L. Jimenez and D. A. Landgrebe. Supervised classification in high dimensional space: geometrical, statistical and asymptotical properties of multivariate data. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 28(1):39–54, February 1993.

[71] T. Joachims. Text categorization with support vector mahcines. In *Proceeding of European conference on machine learning*, 1998.

[72] 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^{th} Nordic Signal Processing Symposium*, 2006.

[73] I. Kanellopoulos, G. G. Wilkinson, and A. Chiuderi. Lan cover mapping using combined Landsat TM imagery and textural features from ers-1 synthetic aperture radar imagery. In *Proc. SPIE Image and Signal processing for remote sensing*, September 1994.

[74] S. Sathiya Keerthi. Efficient tuning of svm hyperparameters unsing radius/margin bound and iterative algorithms. *IEEE Transactions on Neural Networks*, 13(5), 2002.

[75] M. G. Kendall. *A course in the geometry of n-dimensions.* Dover Publication, New York, 1961.

[76] N. Keshava. Distance metrics and band selection in hyperspectral processing with application to material identification and spectral libraries. *IEEE Transactions on Geoscience and Remote Sensing*, 42:1552–1565, July 2004.

[77] G. J. Klir and B. Yuan. *Fuzzy set and Fuzzy Logic: Theory and Applications.* Prentice Hall PTR, 1995.

[78] B.-C. Kuo and D. A. Landgrebe. A robust classification procedure based on mixture classifiers and nonparametric weighted feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40(11):2486–2494, November 2002.

[79] P. Lambert and J. Chanussot. Extending mathematical morphology to color image processing. In *1st International Conference on Color in Graphics and Image Processing*, pages 158–163, October 2000.

[80] D. A. Landgrebe. *Signal Theory Methods in Multispectral Remote Sensing.* John Wiley and Sons, New Jersey, 2003.

[81] D. A. Landgrebe. Multispectral land sensing: where from, where to? *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):414–421, March 2005.

[82] C. Lee and D. A. Landgrebe. Analyzing high dimensional multispectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 31(4):792–800, July 1993.

[83] C. Lee and D. A. Landgrebe. Feature extraction based on decision boundaries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):388–400, April 1993.

[84] M. Lennon. *Méthodes d'analyse d'images hyperspectrales. Exploitation du capteur aéroporté CASI pour des applications de cartographie agro-environnementale en Bretagne.* PhD thesis, Université de Rennes 1, 2002.

[85] M. Lennon, G. Mercier, M.C. Mouchot, and L. Hubert-Moy. Curvilinear component analysis for nonlinear dimensionality reduction of hyperspectral images. In *SPIE*, 2001.

[86] Stan Z. Li. *Markov Random Field Modeling in Image Analysis, second edition.* Springer, 2001.

[87] G. Lisini, F. Dell'Acqua, G. Triani, and P. Gamba. Comparison and combination of multiband classifiers for landsat urban land cover mapping. In *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS'05)*, volume CD ROM, August 2005.

[88] B. Luo, J-F. Aujol, Y. Gousseau, and H. Maître. Cartographie des échelles d'une image satellitaire. In *GRETSI-07*, September 2007.

[89] M.Chi. *Advanced semi-supervised techniques for the classification of remote sensing data.* PhD thesis, DIT - University of Trento,, 2006.

[90] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, August 2004.

[91] G. Mercier and F. Girard-Ardhuin. Partially supervised oil-slick detection by SAR imagery using kernel expansion. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2839–2846, October 2006.

[92] G. Mercier and M. Lennon. Support vector machines for hyperspectral image classification with spectral-based kernels. In *Geoscience and Remote Sensing Symposium*, volume 1. IGARSS '03. Proceedings, July 2003.

[93] K-R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–202, March 2001.

[94] M. Netzband, C. L. Redman, and W. L. Stefanov. Challenges for applied remote sensing science in the ubran environment. In *Urban remote sensing joint event 2007.* URBAN 2005, March 2005.

[95] M. Oussalah. Study of some algebraical properties of adaptive combination rules. *Fuzzy set and systems*, 114:391–409, 2000.

[96] N. R. Pal and J. C. Bezdek. Measuring fuzzy uncertainty. *IEEE Transactions on Fuzzy Systems*, pages 107–118, May 1994.

[97] J. A. Palmason. Classification of hyperspectral data from urban areas. Master's thesis, Faculty of Engineering - University of Iceland, 2005.

[98] J. A Palmason, J. A. Benediktsson, and K. Arnason. Morphological transformations and feature extraction for urban data with high spectral and spatial resolution. In *Proc. IEEE Geoscience and Remote Sensing Symposium.* IGARSS '03. Proceedings, July 2003.

[99] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson, and J. Chanussot. Classification of hyperspectral data from urban areas using morpholgical preprocessing and independent component analysis. In *Proc. IEEE International Geoscience and Remote Sensing Symposium.* IGARSS'05. Proceedings, July 2005.

[100] J. A. Palmason, J. A. Benediktsson, J. R. Sveinsson, and J. Chanussot. Fusion of morphological and spectral information for classification of hyperspectral urban remote senisng data. In *Proc. IEEE International Geoscience and Remote Sensing Symposium.* IGARSS'06. Proceedings, July 2006.

[101] M. Pesaresi and J. A. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(2):309–320, February 2001.

[102] M. Pesaresi, A. Gerhardinger, and F. Kayitakire. Monitoring settlement dynamics by anisotropic textural analysis by panchromatic vhr data. In *Proc. of Urban remote sensing joint event*, 2007.

[103] M. Pesaresi and E. Pagot. Post-conflict reconstruction assessment using image morphological profile and fuzzy multicriteria approach on 1-m-resolution satellite data. In *Proc. of Urban remote sensing joint event*, 2007.

[104] H. Prade and D. Dubois. Possibility theory in information fusion. *Proceedings of the Third International Conference on Information Fusion*, 2000.

[105] S. Rajan and J. Ghosh. Exploiting class hierarchies for knowledge transfer in hyperspectral data. In *Multiple Classifier Systems*, 2005.

[106] S. Rajan, J. Ghosh, and M. M. Crawford. Exploiting class hierarchies for knowledge transfer in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 44(11), 2006.

[107] G. Rellier, X. Descombes, F. Falzon, and J. Zerubia. Texture feature analysis using a gauss-markov model in hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1543–1551, July 2004.

[108] J. A. Richards. Analysis of remotely sensed data: The formative decades and the future. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):422–432, March 2005.

[109] J. A. Richards and X. Jia. *Remote Sensing Digital Image Analysis: An Introduction.* Springer, 1999.

[110] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Proceeding of the Annual Conference on Computational Learning Theory*, 2001.

[111] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsh, K-R Müller, G. Rätsch, and A. J. Smola. Input space versue feature sapce in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, May 1999.

[112] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support vector machines, regularization, optimization, and beyond.* MIT Press, 2002.

[113] B. Schölkopf, A.J. Smola, and K.-R Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[114] B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel methods in computational biology.* MIT press, 2004.

[115] J. Serra. *Image Analysis and Mathematical Morphology.* U.K. Academic, 1982.

[116] J. Serra. *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances.* U.K. Academic, 1988.

[117] A. K. Shackelford, C. H. Davis, and X. Wang. Automated 2-d building footprint extraction from high-resolution satellite multispectral imagery. In *Proc. of IEEE Geoscience and remote sensing symposium*, 2004.

[118] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, 2004.

[119] A. Smola, P. L. Barlett, B. Schölkopf, and D. Schuurmans. *Advances in large margin classifiers.* MIT press, 200O.

[120] P. Soille. *Morphological Image Analysis, Principles and Applications- 2nd edition*. Springer, 2003.

[121] P. Soille. Beyond self-duality in morphological image analysis. *Image and Vision Computing*, 23(2):249–257, 2005.

[122] P. Soille and M. Pesaresi. Advances in mathematical morphology applied to geoscience and remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9):2042–2055, September 2002.

[123] A. H. S. Solberg, A. K. Jain, and T. Taxt. Multisource classification of remotely sensed data: Fusion of landsat TM and SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 32:768–778, July 1994.

[124] K. Tanaka. *An introduction to Fuzzy Logic for practical application*. Springer, 1996.

[125] F. Tupin, I. BLoch, and H. Maitre. A first step toward automatic interpretation of SAR images using evidential fusion of several structure detectors. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3):1327–1343, May 1999.

[126] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

[127] V. Vapnik. *The Nature of Statistical Learning Theory, Second Edition*. Springer, New York, 1999.

[128] G. G. Wilkinson. Resutls and implication of a study of fifteen years of satellite image classification experiments. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):433–440, March 2005.

[129] G. G. Wilkinson, S. Folving, I. Kanellopoulos, N. McCormick, K. Fullerton, and J. Megier. Forest mapping from multi source satellite data using neural network classifiers - an experiement in portugal. *Remote sensing of environement*, 12:84–106, 1995.

[130] D. H. Wolpert. The lack of a priori distinction between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996.

[131] T. Wu, C. Lin, and R. Weng. Probability estimates for multiclass classification by pairwise coupling. *Journal of Machine Learning*, 5:975–1005, August 2004.

[132] R. R. Yager. A general approach to the fusion of imprecise information. *International Journal of Intelligent Systems*, 12:1–29, 1997.

[133] L. A. Zadeh. Fuzzy sets. *Information and Control*, pages 338–353, 1965.

[134] L. A. Zadeh. Probability measures of fuzzy events. *J.Math. Analysis and Application*, 23:421–427, 1968.

[135] C. Ünsalan and K. L. Boyer. Linearized vegetation indices based on a formal statistical framework. *IEEE Transactions on Geoscience and Remote Sensing*, 42(7):1575–1585, July 2004.