DGK Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 843

Wolfgang J. Groß

Nonlinear Feature Normalization for Hyperspectral Feature Transfer

Ettlingen 2020

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5255-0

Diese Arbeit ist gleichzeitig veröffentlicht in:
Wissenschaftliche Arbeiten der Fachrichtung Geodäsie der Universität Stuttgart
http://dx.doi.org/10.18419/opus-10649, Stuttgart 2019

Veröffentlichungen der DGK

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften

Reihe C | Dissertationen | Heft Nr. 843

Nonlinear Feature Normalization for Hyperspectral Feature Transfer

Von der Fakultät für Luft- und Raumfahrttechnik und Geodäsie

der Universität Stuttgart

zur Erlangung des Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

Vorgelegt von

Dipl.-Math. techn. Wolfgang J. Groß

Geboren am 01.03.1985 in Karlsruhe, Deutschland

Ettlingen 2020

Verlag der Bayerischen Akademie der Wissenschaften

ISSN 0065-5325 ISBN 978-3-7696-5255-0

Diese Arbeit ist gleichzeitig veröffentlicht in:

Wissenschaftliche Arbeiten der Fachrichtung Geodäsie der Universität Stuttgart
http://dx.doi.org/10.18419/opus-10649, Stuttgart 2019

Adresse der DGK:

Ausschuss Geodäsie der Bayerischen Akademie der Wissenschaften (DGK)
Alfons-Goppel-Straße 11 ● D – 80 539 München
Telefon +49 – 331 – 288 1685 ● Telefax +49 – 331 – 288 1759
E-Mail post@dgk.badw.de ● http://www.dgk.badw.de

Prüfungskommission:
Vorsitzender: Prof. Dr.-Ing. Nico Sneeuw
Referent: Prof. Dr.-Ing. Uwe Sörgel
Korreferenten: Prof. Dr.-Ing. Uwe Stilla

Tag der mündlichen Prüfung: 12.07.2019

ISSN 0065-5325 ISBN 978-3-7696-5255-0

Contents

Acronyms

ATCOR *Atmospheric & Topographic Correction*. 18, 31, 34, 37, 39, 102

AVIRIS *Airborne Visible/Infrared Imaging Spectrometer*. 108

BRDF *Bidirectional Reflectance Distribution Function*. 38, 40, 60

BREFCOR *BRDF Effects Correction*. 39

CCA *Canonical Correlation Analysis*. 43

CIR *Colored Infrared*. 101, 103–107

DA *Domain Adaptation*. 41, 56, 60, 65, 66

DEM *Digital Elevation Model*. 36, 39, 91, 102

DSNU *Dark Signal Non-Uniformity*. 30, 31

ECC *Enhanced Correlation Coefficient*. 91, 92, 97, 98, 101, 102

ELC *Empirical Line Correction*. 34, 35

FLAASH *Fast Line-of-sight Atmospheric Analysis of Hypercubes*. 31, 34

FPN *Fixed-Pattern Noise*. 30

FT *Feature Transfer*. 11, 12, 17, 19, 22–24, 60, 63, 66, 86, 88, 99, 101, 103, 105, 111, 114, 128, 129, 131, 136, 138, 145, 147, 148

GFK *Geodesic Flow Kernel*. 59, 60, 138, 140, 141

GPS *Global Positioning System*. 90–92, 102

HM *Histogram Matching*. 138, 139

IMU *Inertial Measurement Unit*. 90

INS *Inertial Navigation System*. 90–92, 102

IPR *Individual Pixel Re-scaling*. 138–142

ISOMAP *Isometric Mapping*. 43, 44, 52–55, 65

KEMA *Kernel Manifold Alignment*. 44, 62, 63, 138, 141, 142

KSC *Kennedy Space Center*. 108, 118, 119, 121, 124–127

LiDAR *Light Detection And Ranging*. 25, 102

LLE *Locally Linear Embedding*. 43, 45, 55, 65

MA *Manifold Alignment*. 11, 17, 21–23, 33, 41, 42, 44, 46, 56–58, 64–66, 98, 99, 101, 103, 105, 138, 140–142, 145–148

ML *Manifold Learning*. 23, 41, 45, 46, 50, 51, 65, 145

MODTRAN *MODerate resolution atmospheric TRANsmission*. 31, 34

NFN *Nonlinear Feature Normalization*. 12, 14, 15, 20, 21, 23, 24, 33, 66, 67, 69–72, 74–78, 80–83, 85–87, 92, 97–101, 108, 109, 111, 113, 115, 117–125, 127–129, 136, 145–151

NFNalign *Nonlinear Feature Normalization for Data Alignment*. 12, 14, 15, 20, 22–24, 85–89, 91–95, 97, 99–101, 103, 105, 108, 109, 111, 113–116, 123, 128–143, 145, 147–150

NN *Nearest Neighbor*. 51, 55, 61, 69, 71, 72, 78, 83, 117, 119, 121–123, 126, 132, 133

PCA *Principal Component Analysis*. 37, 42, 50, 53, 60, 80, 81, 140, 149

PRNU *Photo Response Non-Uniformity*. 30

RBF *Radial Basis Function*. 61, 115

RICS *Rotation Invariant Cluster Sampling*. 110–112, 118–121, 123, 129

RMSE *Root Mean Squared Error*. 22, 116, 117, 128–131, 133–143, 147

SAM *Spectral Angle Mapper*. 21, 113, 115, 117–121, 123–127, 146, 147

SAR *Synthetic Aperture Radar*. 25

SGF *Sampling Geodesic Flow*. 58, 59

SNR *Signal-to-Noise Ratio*. 18, 26–28, 93, 101, 108, 131, 142

SSMA *Semi-Supervised Manifold Alignment*. 44, 60–62

SVM *Support Vector Machine*. 19, 22, 60, 63, 114, 115, 117, 128–132, 134, 137–142, 145, 147

UAS *Unmanned Aerial Systems*. 27

UISA *Unsupervised Image Sets Alignment*. 43

USGS *United States Geological Survey*. 82

WGS 84 *World Geodetic System from 1984*. 91

Abstract

Hyperspectral remote sensing is an important topic for deriving high-level information about the earth's surface. Applications include land cover mapping, precision farming, and the detection of environmental pollution. This is made possible by recording and evaluating narrow-band features that are characteristic of individual materials. External effects, however, lead to nonlinearities in the data and complicate data analysis. These effects include changes in illumination, hard and partial shadows, as well as transmission / multiple reflections by objects in the scene, and anisotropic effects for 3D objects.

Correcting these effects is required for robust data analysis. In particular, when comparing multiple data sets a unified representation is required. Physically motivated models for correcting atmospheric influences are generally used for the pre-processing of hyperspectral data. However, these models do not consider local variations, such as shadows and object geometry. Therefore, this thesis deals with data-driven approaches in the field of *Manifold Alignment* (MA) and *Feature Transfer* (FT) to transfer several data sets to a common system.

Previous research on these topics has focused primarily on learning the underlying geometry of high-dimensional data and aligning multiple datasets by determining the minimum discrepancy while preserving the individual data structure. Usually, a common domain with very high dimensionality is chosen to facilitate the alignment. The transformation into another domain, however, prevents physical interpretability. Also, inversion of one data set from the common domain to the domain of a target data set is difficult due to the pre-image problem.

The contributions of this thesis can be divided into two categories. The *Nonlinear Feature Normalization* (NFN) is a data-driven approach to mitigate nonlinear effects in hyperspectral data. NFN is a supervised method and requires training samples for each class in the scene. A new basis for data representation is defined, consisting of one spectral reference signature per class. The training data are then used to individually shift all samples towards the new basis. This significantly reduces the effects of nonlinearities, as shown by comparing classification results before and after the NFN transformation.

The NFN is then used to derive the *Nonlinear Feature Normalization for Data Alignment* (NFNalign). NFNalign transforms multiple data sets to the same basis in the common domain and then applies an inverse transformation to transfer data sets from the common domain to a domain of another data set. Since the dimensionality of the data is not changed during the transformation, it is possible to perform the inversion analytically. The functionality of NFNalign is demonstrated by transforming hyperspectral radiance data to reflection data. Thereby, the pre-processing step of the atmospheric correction can be replaced, shadows and other nonlinearities are corrected, and characteristic features of the spectral signatures are transferred. The quality of the alignment is demonstrated by applying an SVM model trained on a reference data set to the aligned data set. Additional alignment is assessed by applying a classification model trained on a reference data set to a test data set after it has been transformed to the domain of the reference with NFNalign. Further experiments investigate the robustness with regard to noise and errors in the training data as well as the alignment of data with different dimensions. Also, a comparison with common reference methods is performed.

Overall, NFN and NFNalign provide a complete framework for hyperspectral data alignment and FT.

Kurzfassung

Die hyperspektrale Fernerkundung ist ein leistungsfähiges Werkzeug, um komplexe Informationen über die Erdoberfläche abzuleiten. Anwendungen sind unter anderem Landbedeckungskartierung, Precision Farming und die Erkennung von Umweltverschmutzung. Ermöglicht wird dies durch die Aufzeichnung und Auswertung schmalbandiger Merkmale, die charakteristisch für einzelne Materialeien sind. Externe Effekte führen jedoch zu Nichtlinearitäten in den Daten und erschweren die Datenanalyse. Zu diesen Effekten gehören Änderungen in der Beleuchtung, Schlag- und Halbschatten und Transmission / Mehrfachreflexionen von Objekten in der Szene sowie anisotrope Effekte an 3D-Objekten.
Die Korrektur dieser Effekte ist für eine robuste Datenanalyse erforderlich. Insbesondere beim Vergleich mehrerer Datensätze ist eine einheitliche Darstellung erforderlich. Physikalische Modelle zur Korrektur atmosphärischer Einflüsse werden im Allgemeinen für die Vorverarbeitung von Hyperspektraldaten verwendet. Diese Modelle berücksichtigen jedoch keine lokalen Variationen, wie z. B. Schatten und Objektgeometrie. Daher behandelt diese Arbeit datengetriebene Ansätze aus dem Themengebiet Manifold Alignment und Merkmalsübertragung, um mehrere Datensätze in ein gemeinsames System zu überführen.
Bisherige Untersuchungen zu diesen Themen konzentrieren sich hauptsächlich auf das Erlernen der zugrunde liegenden Geometrie der hochdimensionalen Daten und das Alignment mehrerer Datensätze, indem die minimale Diskrepanz ermittelt wird und die individuelle Datenstruktur erhalten bleibt. Normalerweise wird eine einheitliche Domäne mit sehr hoher Dimension gewählt, um das Alignment zu erleichtern. Durch die Transformation in einen anderen Wertebereich wird jedoch die physikalische Interpretierbarkeit verhindert. Außer-

dem ist die Inversion eines Datensatzes aus der gemeinsamen Domäne in die Domäne eines Zieldatensatzes nur unter bestimmten Voraussetzungen möglich. Beispielsweise muss gezeigt werden, dass ein Urbild der Abbildung für alle Datenpunkte existiert.

Die Beiträge dieser Arbeit können in zwei Kategorien unterteilt werden. Die *Nonlinear Feature Normalization* (NFN) ist ein datengesteuerter Ansatz zur Abschwächung nichtlinearer Effekte in hyperspektralen Daten. NFN ist eine überwachte Methode und erfordert somit Trainingsdaten für jede Klasse in der Szene. Es wird eine neue Basis für die Datendarstellung definiert, die aus einer spektralen Referenzsignatur pro Klasse besteht. Die Trainingsdaten werden anschließend verwendet, um alle spektralen Signaturen individuell in Richtung der neuen Basis zu verschieben. Dies reduziert die Auswirkungen von Nichtlinearitäten signifikant, was durch den Vergleich von Klassifikationsergebnissen vor und nach der NFN-Transformation gezeigt wird.

Aus der NFN wird anschließend die *Nonlinear Feature Normalization for Data Alignment* (NFNalign) abgeleitet. NFNalign transformiert mehrere Datensätze zur gleichen Basis und wendet anschließend eine inverse Transformation an, um Datensätze aus der gemeinsamen Domäne in die Domäne eines anderen Datensatzes zu übertragen. Da die Dimensionalität der Daten während der Transformation nicht verändert wird, ist es möglich, die Invertierung analytisch durchzuführen. Die Funktionsweise von NFNalign wird demonstriert, indem hyperspektrale Radianzen in Reflexionsdaten umgewandelt werden. Hierdurch kann der Vorverarbeitungsschritt der Atmosphärenkorrektur ersetzt werden, Schatten und andere Nichtlinearitäten werden korrigiert und charakteristische Merkmale der spektralen Signaturen übertragen. Die Qualität des Alignment wird beurteilt, indem ein auf einem Referenzdatensatz trainiertes Klassifikationsmodell auf den Testdatensatz angewendet wird, nachdem dieser mit NFNalign in die Domäne der Referenz transformiert wurde. Weitere Experimente untersuchen die Robustheit im Bezug auf Rauschen und Fehler in den Trainingsdaten sowie das Alignment von Daten mit unterschiedlicher Dimension. Außerdem wird ein Vergleich mit gängigen Referenzmethoden durchgeführt.

Insgesamt bilden NFN und NFNalign ein vollständiges System für das Alignment hyperspektraler Daten und die Übertragung von Merkmalen.

Chapter 1

Introduction

This thesis investigates the task of data alignment and *Feature Transfer* (FT) for the joint analysis of hyperspectral data sets. Data pre-processing and *Manifold Alignment* (MA) techniques are discussed to mitigate especially local and nonlinear variations in the spectral signatures between corresponding samples. These are caused by different ambient conditions, sensor and object geometry, and reflective properties of different materials.
The contribution of this thesis can be divided into two main parts, namely the development of an algorithm to perform feature normalization that mitigates nonlinear effects in a single data set, and the combination of multiple such normalizations followed by an inversion to transfer a data set from its original domain to the domain of another data set.
This chapter provides an overview of the motivation and context of this dissertation. Section 1.1 formulates the problem statement and the challenges of aligning multiple hyperspectral data sets. Section 1.2 summarizes the contributions of this dissertation. The general structure of this thesis is provided in Section 1.3.

1.1 Problem Statement

The capability of hyperspectral sensors for sampling characteristic narrow-band features of different materials allows robust analysis for many applications, such as classification and

segmentation. The sensor developers constantly work to improve the sensor parameters, e.g., the *Signal-to-Noise Ratio* (SNR), the spatial and spectral resolution, and the robustness of the detectors, gratings, etc. The characterization of materials works extremely well in controlled environments like laboratory settings with calibrated light sources and reference targets for constant quality control of the sensor performance [65, 26]. While valuable information can be gained from these kinds of experiments, transferring this knowledge to hyperspectral data recorded in the field or with the help of air- and spaceborne platforms introduces new challenges.

Parameters that were known and controlled in the laboratory are then subject to changes and need to be measured, or their influence on the data has to be known. These effects include illumination changes, either by solar angle, clouds, or atmospheric disturbances [109]. The geometry of an observed object also has wavelength-dependent reflection properties [20, 111, 108]. Finally, the proximity of objects can lead to multiple reflections, which can be observed in the measured spectra [93].

The effects mentioned above influence the characteristic reflection, absorption and transmission properties of materials that can be exploited to deduce high-level information in the form of maps for land cover [18], vegetation stress [140] or chlorophyll content [137]. Generating these products with consistent quality from multitemporal, and possibly multisensor, hyperspectral data implies dealing with these environmental effects that introduce nonlinearities to the data [125, 119, 52]. To compare a hyperspectral remote sensing data set to a laboratory measurement or even other data sets from the same campaign requires a framework for correcting these effects. In general, the correction and mitigation of these nonlinear effects can happen on three different levels.

Data representation: The first step in data preprocessing is usually to perform radiometric and atmospheric correction. Radiometric correction of raw data to at-sensor radiance is accurate to a level of 2-3 % in absolute radiance units [108]. For comparing multitemporal remote sensing data, these values have to be further converted to bottom of atmosphere reflectance values. That is mostly done by inversion of a radiative transfer model, using

software like *Atmospheric & Topographic Correction* (ATCOR)-4 [103] or MODTRAN [8]. These codes produce reflectance values that are, on average, correct. However, they don't take per-pixel illumination changes [82] or anisotropy [135] into account. Furthermore, they usually require additional parameters for water vapor and aerosol content, visibility, etc. that need to be measured separately or estimated from the data. This process entails tuning and optimizing of many parameters and can be very time-consuming [125]. Besides approaches based on a physically motivated model, there also exist purely data-driven ways to find a common representation that allows FT for combined evaluation of multiple data sets [92, 119, 121, 31, 125].

Other approaches entail one or more of the following techniques.

Adapt the classifier: When the data contains nonlinear effects analysis becomes more difficult. One way deal with this is to adjust the classifier to be more robust towards these effects. One popular example is the introduction of a nonlinear kernel to improve the *Support Vector Machine* (SVM) classification. Other approaches include semi-supervised learning, using unlabeled samples to adapt weights [40, 12, 63], by spatial regularization [62] or adding few informative labeled samples [122, 80]. These approaches usually have several free parameters, require expertise in machine learning and are computationally expensive.

Encoding of invariances: Here, individual nonlinear effects are modeled, usually motivated by a specific physical characteristic. Examples are the orientation and geometry of an object [20], and signal attenuations for shadow correction [142, 11]. The correction requires accurate models and usually auxiliary information, e.g., time of data acquisition, a 3D elevation model or material specific parameters. It is also possible to use synthetic examples to make a classifier invariant towards nonlinearities, e.g., by adding simulated instances of rotated objects to the training data [60]. Comprehensive ground truth information including local material properties and variations is generally difficult and expensive to acquire in remote sensing. There exist some data-driven correction approaches for common materials, but no global models are available [111, 108]. Other algorithms perform correction of multiple scattering [93] and illumination equalization [107]. While all these procedures correct

certain aspects of nonlinearities in the data, a combination of all methods is time-consuming, and interference between individual methods cannot be ruled out.
Encoding of variances and adaptation of classifiers can be useful to develop new classifiers or to gain a better understanding of the underlying effects that cause nonlinearities, however, finding a data representation that is universally useful for data evaluation and comparison is a desirable goal. This thesis researches different approaches that allow comparison of hyperspectral data sets acquired under different conditions, including the transfer of characteristic features from the domain of the original data to a reference domain, where it can be evaluated with an established model. It also provides its own approach, which is an extension of the work from [50, 52, 49]. With this, it is possible to align multiple data sets in a domain with desirable properties where the features of the same materials are identified with each other. Additionally, the proposed data transformation method mitigates nonlinearities in the data as a byproduct during calculation. As the proposed method does not require additional information about physical properties of individual materials and instead uses labeled ground truth information, it can be easily used to correct whole measurement campaigns completely making large parts of the data preprocessing chain unnecessary.
A full description of all contributions of the proposed method is given in the next section.

1.2 Contributions

The contributions of this thesis can be divided into two main areas. The proposed *Nonlinear Feature Normalization* (NFN) algorithm, introduced in Chapter 4, transforms a data set to a new arbitrary basis and successfully mitigates nonlinearities in the process. The *Nonlinear Feature Normalization for Data Alignment* (NFNalign) algorithm, discussed in Chapter 5 combines multiple NFN calculations and its inversion to transform whole data sets from their original domain to a target domain.
The NFN is a data-driven approach for mitigating nonlinear effects in hyperspectral data that only requires some training samples per class in the data set. It allows, but is not

limited to, the mitigation of changes in illumination, hard and partial shadows, and transmission/multiple reflections by objects in the scene as well as anisotropic effects for 3D objects.

Usually, these effects are dealt with separately by using complex physical models that require additional information. Conversely, the proposed transformation is defined by a set of training samples per class. If the training data represents a good sampling of the high dimensional manifold for each class, the transformation is invariant to the underlying physical effects leading to the nonlinear behavior. The selection of basis vectors for the transformation has little effect for further analysis, which opens up some interesting opportunities. It allows for example dimensionality reduction by selecting the first p canonical unit vectors or to transform a radiance data set to reflectance using spectrally subsampled laboratory spectra. The successful mitigation of nonlinear effects can be evaluated by comparing a classification with the *Spectral Angle Mapper* (SAM) before and after the NFN transformation. The classification accuracy is drastically increased when applying SAM on the transformed data. A major advantage over the correction of individual effects is the ease of use. Manually selecting small training areas per class, including potential areas with nonlinearities, e.g., slanted roofs, objects in the shadow of a tree, etc., is enough to produce good results. Only two parameters are required in the present form, one for the strictness of the penalty function and one for the robustness towards errors in the training data. It is shown in Chapter 6 that good results are achieved almost regardless of parameter selection and optimization improves the outcome only by a few percents. The robustness of the NFN algorithm when the training data is contaminated with samples from other classes is also tested, and even errors of 20 % could be dealt with by increasing the neighborhood size. Additionally, NFN can be computed very fast, scales linearly with image size and can potentially be used even for real-time applications. Also, no global adjacency matrices calculation is required. This requirement is a major downside of other MA algorithms which are limited to smaller data sets or require specific memory management. While the NFN algorithm was able to improve linear classification for all tested data sets, it remains to be analyzed if it also performs well

when classes have very similar spectral signatures or classification relies on distinct features that only constitute to a couple of bands.

The arbitrary basis selection and the mitigation of nonlinearities are the two important features that triggered the development of the NFNalign algorithm. The former allows the transformation of multiple data sets to the same basis in a common domain while the latter guarantees that the data structure in the common domain is similar up to range for all data sets, i.e., radiance data in $[1, 16383]$ and reflectance data in $[0, 1]$. By calculating correspondences between samples of different data sets, it is possible to directly apply the inverse translation of a sample in one data set to its correspondence in another data set. The sample is transformed from the common domain to the original domain of its correspondence. Practical approaches for calculating the correspondences between co-registered and between spatially disconnected data sets are discussed and successfully applied. Since the dimension of the domains is not changed during the transformation, everything can be calculated analytically. In contrast to other MA approaches, this has the benefit that no ill-posed problem has to be solved for the inversion. This means that physical interpretability of the data aligned by NFNalign is possible by using the features from the reference domain. NFNalign can even be used to align radiance data to a reference reflectance data set, skipping the atmospheric correction step completely. The quality of the alignment is shown by classifying the aligned data with an SVM model learned on the reference data alone. Since the results are comparable to classifying the data in their original domains, FT with NFNalign can be considered successful. The *Root Mean Squared Error* (RMSE) before and after the transformation with NFNalign are compared and show a highly improved conformity between the aligned data sets. Comparing NFNalign to MA approaches revealed that data alignment in the reference domain was not possible without further consideration of the preimage problem. Thus, only FT to a common domain could be evaluated. However, physical interpretability is completely lost by the transformation to a high-dimensional common domain for multiple data sets. The only other approach that produced acceptable results was individual rescaling of the samples using the correspondences between samples. However,

the results of NFNalign are superior. Finally, linear interpolation of a low-dimensional data set to a high dimensional reference allows NFNalign to perform alignment of data sets with different dimensionalities.

1.3 Organisation

The objective of this thesis is to develop a practical approach for mitigation of nonlinear effects in high-dimensional data, the NFN. The method is then modified to allow data alignment and FT between multitemporal, multilocal and even multimodal data.
Chapter 2 introduces the background of hyperspectral imaging for remote sensing. Different scenarios are given that require data alignment techniques. Additionally, the challenges of data alignment and FT between data sets are discussed.
Chapter 3 provides an overview of significant tools for data alignment. It explains the process of radiometric and atmospheric pre-processing of hyperspectral data using physical models. Other techniques that are not necessarily restricted to hyperspectral data, i.e., *Manifold Learning* (ML) and MA, are introduced. The process of FT, exploiting already gained knowledge while analyzing one data set and applying it to another data set is also discussed.
Chapter 4 introduces the requirements and mathematical notation to describe the proposed NFN algorithm. The idea behind the method is emphasized by a set of simple examples. The NFN algorithm is then deduced in the current setting, and essential properties like fine-tuning, basis vector selection and computational complexity are discussed.
Chapter 5, then, expands on the properties of NFN to create the NFNalign. As NFNalign requires additional information in the form of pixel correspondences between data sets, practical methods to calculate these correspondences in different settings are presented. This chapter is concluded by a discussion of error propagation and alignment accuracy as well as computational complexity.

Chapter 6 is comprised of different experiments to evaluate NFN and NFNalign in different settings. First, multiple commonly used hyperspectral remote sensing data sets are introduced as well as a new data set prepared explicitly for this thesis. Then, suitable evaluation metrics are presented. These are applied to the task of using NFN for mitigation of nonlinear effects and then transferred to the evaluation of NFNalign for data alignment and FT.
Chapter 7 concludes this thesis by summarizing the contributions and discussing future research opportunities based on the proposed methods.

Chapter 2

Background

This chapter gives an overview of general hyperspectral imaging and related topics. Section 2.1 introduces the commonly used hyperspectral sensor design for remote sensing and gives information about planning and conducting a successful measurement campaign. Section 2.2 discusses the necessary steps of preprocessing to compare the raw measurements to spectral reflectance signatures. Since this is a complex process based on an incomplete physical model, Section 2.3 gives examples where the preprocessing fails and how specific effects influence the data.

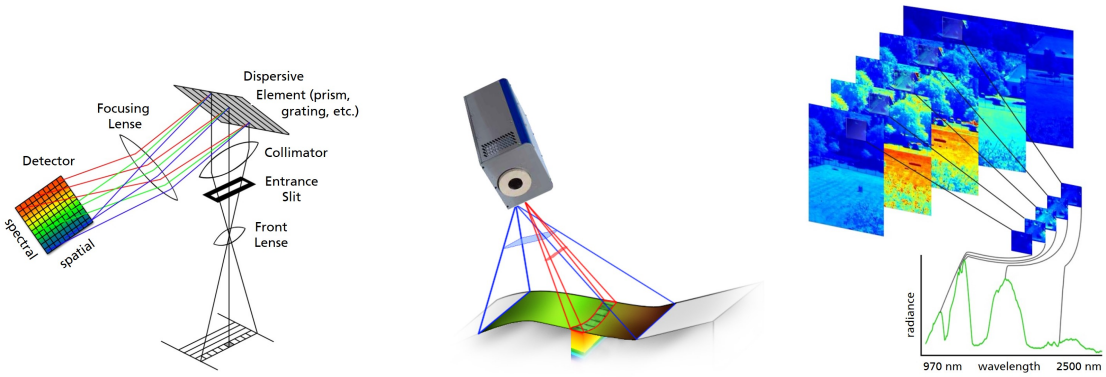
2.1 Hyperspectral Remote Sensing

Remote sensing is defined as the process of gathering information, in the form of electromagnetic radiation, about an object or phenomenon without making physical contact [104]. In this thesis, remote sensing refers to the collection of information about the earth surface with the use of air- or spaceborne systems. Sensors are generally categorized in active and passive systems. Active sensors like *Light Detection And Ranging* (LiDAR) (Light Detection And Ranging) or *Synthetic Aperture Radar* (SAR) emit a specific signal and record the reflection after the signal has interacted with an object. Passive sensors like photo cameras and multi-/hyperspectral sensors use ambient radiation, e.g., sunlight or artificial illumina-

tion sources. This requires an adjustment to signal source variations as the measured signal cannot be directly compared to a known calibrated source. In this section, an introduction to hyperspectral sensor design is given along with some guidelines for conducting measurement campaigns.

Hyperspectral imaging is an important category in remote sensing. The sensors record a full spectrum in hundreds of bands for each pixel, which theoretically allows mining of all available information from the data without prior knowledge. The spatial relationship between neighboring spectra can also be exploited for more complex models in a spatial-spectral analysis, e.g., to improve accuracy and robustness for segmentation or classification tasks. While neighboring bands usually show high correlation, it is possible for specific effects to manifest in a feature only detectable in a small wavelength range. One example is the presence of hydrocarbon for oil spill detection, which only affects the wavelength around 1.73 μm [68]. Compared to commercial broadband sensors, hyperspectral sensors are still costly. As passive sensors, they require good illumination conditions to produce high SNR. For airborne hyperspectral remote sensing, the use of artificial light sources is not possible, which results in the planning of a data acquisition campaign at dates with the best possible conditions. The high spectral resolution usually comes at the cost of spatial resolution compared to broadband RGB or panchromatic imaging sensors. Figure 2.1(a) shows the schematic sensor design for a hyperspectral push broom sensor. These sensors record one line of spatial information per frame, while the second dimension of the detector chip is used for the spectral information. The sensor has to be moved relative to the scene to record a whole image, e.g., the sensor is installed into an airplane, and every frame records the spatial information perpendicular to the flight direction, while the forward movement of the plane ensures that subsequent frames can be combined to a full image. Usually, the sensor is installed with a nadir view.

The sensor works by collecting the reflected signal by the objects on the ground with a converging lens. An entrance slit and a collimator are used to eliminate stray light and aligning the signal onto a dispersive element, like a grating or a prism. This element spreads out



(a) Schematic diagram of a hyperspectral pushbroom sensor

(b) Comparing the image recording of an RGB camera (blue cone) to a single frame of a hyperspectral pushbroom sensor (red cone).

(c) Spectral signature for a single spectrum and example 2D images per band

Figure 2.1: Only one spatial line is recorded by hyperspectral pushbroom sensors each frame. The second dimension of the detector chip gathers the spectral information. An image can be constructed from consecutive frames by moving the sensor relative to the object. Each pixel then contains the whole spectral information in the form of a spectral signature.

the signal in different wavelength intervals, and each interval is subsequently focused onto a different line of the detector chip where the signal is recorded. Figure 2.1(b) visualizes the difference between a single frame of a 2D camera (blue cone) and the push broom field of view (red cone). Subsequent frames of push broom sensors on a moving platform can be stitched together to a full 2D coverage of an area. However, combining the frames is easier when the sensor is mounted on a stable platform or has at least some form of sensor stabilization. Strong vibrations and strong yaw, roll, and pitch of the platform can lead to gaps in the coverage.

Hyperspectral frame cameras that can record a full 2D image with complete hyperspectral information in a single frame are already available but severely limited in spatial and spectral resolution [6]. Since the sensor design is small and lightweight, they offer the great advantage to be used on *Unmanned Aerial Systems* (UAS) without the use of stabilization. However, push broom sensors are commonly used for remote sensing as they offer superior SNR and spatial/spectral resolution. This allows faster coverage of larger areas as it is not

limited to low altitudes or the battery life of UAVs.
Figure 2.1(c) gives an example of the spectral information at different wavelengths of the spectral signature, e.g., vegetation has wavelength intervals with very high reflectivity compared to inorganic materials.
Successfully acquiring hyperspectral remote sensing data of a defined area requires careful planning. Several parameters need to be taken into consideration. This includes

1. The laws for flight operations generally limit the altitude. Additionally, campaigns in no-flight zones, e.g., near airports or over military facilities require specific permits.

2. Weather and illumination conditions should be monitored in advance. Most campaigns in Europe are scheduled in spring or summer to ensure good conditions.

3. Flight planning based on the sensor parameters is a requirement to generate coverage of the area without gaps. To compensate for the movement of the platform due to sudden winds etc. an overlap between recorded data stripes is useful.

4. The width of a single flight line can be calculated from the field of view of the sensor and the altitude using the law of sines. The ground sampling distance is then determined by the number of pixels on the detector chip along the covered ground segment.

5. The number of necessary flight lines depends on the altitude, which is inversely proportional to the ground sampling distance.

6. The speed of the platform has to be synchronized with the frame rate and the exposure time of a single frame of the sensor. The speed above ground has to be taken into account to generate quadratic pixels in the recorded data.

For this thesis, data from a measurement campaign from 2014, conducted by the Fraunhofer IOSB was used. The data was recorded with an aisaEAGLE II hyperspectral sensor over the village of Greding, Germany. The sensor covers the wavelength range of $390 - 990$ nm with 127 bands. With an altitude of approximately 770 m above ground and a speed above

ground of 240 km/h, a frame rate of 60 Hz was necessary to create square pixels with a ground resolution of approximately 0.5×0.5 m. Due to good weather conditions, the exposure time was set to 5 ms for the whole campaign.
The following section schematically discusses the process of hyperspectral data preprocessing that was performed to convert the measured raw data to radiance and reflectance values.

2.2 Data Preprocessing

Hyperspectral data are recorded with the use of a frame grabber to extract the measured signal from the detector. The raw data is saved as digital numbers and has to be preprocessed to extract physical interpretability. While a lot of information can be extracted without radiometric correction, using a physical model has advantages when dealing with multitemporal data or when data from multiple sensors are jointly analyzed. The ultimate goal would be to transform all data sets to an absolute system where disturbance is eliminated. In the case of hyperspectral data representing each sample measurement in the form of reflectance brings many benefits. The reflectance of an object is defined as the effectiveness radiant energy is reflected by its surface. The reflectance of an object is a directional property and depends on multiple factors. This property is further discussed in Section 2.3.3, for now, it is assumed that reflectance is the effectiveness of an object reflecting radiant energy. If all data sets are corrected to reflectance, this enables direct comparison and allows the extraction of meaningful features in a controlled setting and transferring the knowledge to remote sensing. In other words, the transfer of features between data sets is possible.
Hyperspectral data preprocessing can be partitioned into sensor specific corrections and correction of environmental effects.

1. A sensor specific effect is caused by small differences in the sensitivity of individual detector elements. They are subject to an effect called *Fixed-Pattern Noise* (FPN) [84]. It consists of two parameters, the *Dark Signal Non-Uniformity* (DSNU) and the *Photo Response Non-Uniformity* (PRNU). DSNU describes the offset from the

average signal at a given setting and varies with temperature and integration time. It is independent of external illumination and can be estimated by calculating the mean signal per detector element for multiple frames, while the detector is covered. This is usually done by collecting multiple dark frames at the end of each recording, using a mechanical shutter. The PRNU describes the gain, the ratio between incoming optical power and a pixels electrical signal output. This can be measured in a controlled environment using certified reference materials and artificial illumination sources. If the detector has a linear photoresponse, calculating a single gain factor per pixel is sufficient for correction. Otherwise, a look-up table with multiple values for different levels of saturation is required. Other factors are the smile and keystone effect. The smile effect is a wavelength shift in the spectral domain which takes on the form of a smile or frown along the spatial position of the detector array [98]. The keystone effect is the spatial equivalent to the smile effect. Both can be caused by misalignment of the lens and gratings and are usually stronger along the edges of the detector chip [73]. Overall, these effects are generally corrected using a calibration file provided by the sensor manufacturer. These files are only updated every couple of years depending on the maintenance interval of the sensor. Since spaceborne sensors cannot be shipped in for re-calibration, different approaches to estimate changes of these parameters based on the measured data are employed [43]. Single dead pixels cause stripes in the data due to the push broom recording. These can be corrected by interpolation of their neighbors. Also, bad bands or bands where no useful information can be gathered, e.g., due to atmospheric absorption are often deleted.

2. Correction of environmental effects is necessary to adjust for ever-changing atmospheric and topographic conditions. In contrast to the sensor specific effects, environmental effects can change with every new flight line or based on spatial location. The most commonly addressed environmental variations are changes in the atmosphere, e.g., due to different aerosol layers and water vapor content. This is usually done using a physical model to correct the whole data set [103]. Variation in illumination and shadows

due to large buildings or clouds are local fluctuations of the spectral signatures and can vary over time, e.g., they can be different between flight stripes. Another external effect is the variation of reflectance as a function of viewing angle, solar illumination geometry, and texture of the observed material [20]. Finally, multiple reflections between adjacent objects also affect the measured signal. Since these effects are usually addressed individually, they are discussed in more detail in Section 2.3.

Figure 2.2 gives an overview of the preprocessing chain from the raw data to at-surface reflectance values. Initially, the frame grabber records raw digital numbers. Then, the data is first corrected for dead pixels and bad bands. Provided a dark current measurement was taken the noise can be reduced. The gain and offset can be corrected using a spectral calibration file usually acquired by a previous sensor calibration by the manufacturer. At this point, the data is already converted to at-sensor radiance values. The absorption between the sensor and the target has to be taken into account to convert these values to at-surface reflectance. Atmospheric models like ATCOR, *Fast Line-of-sight Atmospheric Analysis of Hypercubes* (FLAASH) or *MODerate resolution atmospheric TRANsmission* (MODTRAN) can be used to correct remote sensing data.

This thesis focuses on the correction of environmental-based nonlinearities. Sensor-specific nonlinear effects are less likely to be a consumer problem, as manufacturers usually deliver radiometric calibration files for conversion from digital numbers to radiance. Only the DSNU has to be recorded (often automatically with a mechanical shutter) and subtracted from the corresponding image lines. Additionally, it is advised to recalibrate hyperspectral systems at least every two years to ensure stability and update calibration files [7]. The correction of environmental effects instead is difficult due to a lot of possible variability for each parameter and the requirement for additional information or lengthy trial and error processes to estimate the best parameters.

The next section introduces the individual environmental-based effects that need to be corrected to compare multiple data sets. In the following chapters, the focus is shifted from

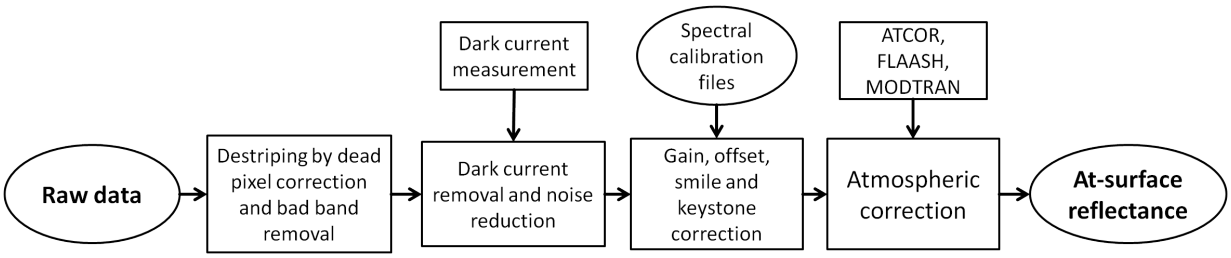
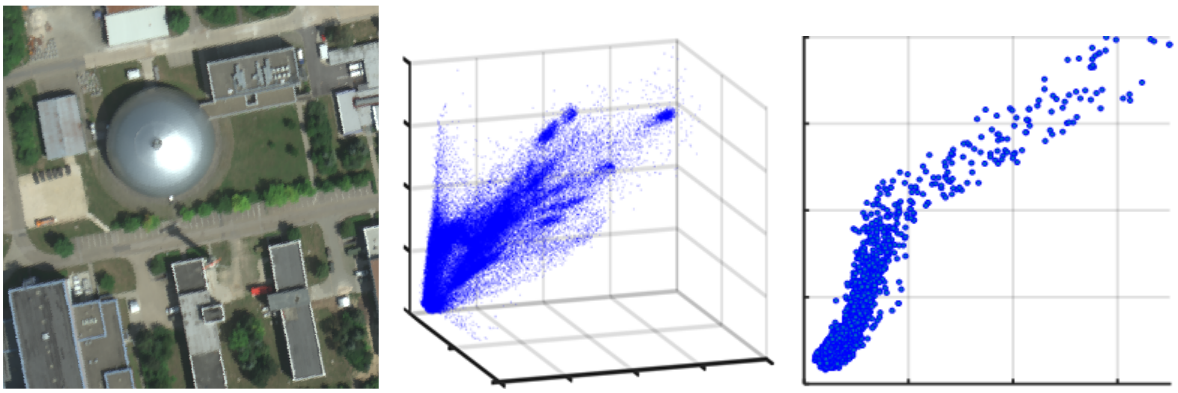


Figure 2.2: Layout of the general hyperspectral preprocessing chain starting with raw data in the form of digital numbers and progressing to at-surface reflectance.



(a) 3-band composite of a hyperspectral scene.

(b) The scatter plot of the whole scene for three bands shows one large clusters with multiple extensions.

(c) Scatter plot for the metal dome spectra showing the nonlinear behavior of the point cloud.

Figure 2.3: Hyperspectral data representation as a 3-band color image or scatter plots. Nonlinear structures become visible when single classes are singled out.

the individual physical foundation of each effect to their manifestation in the data, and from there to a unified approach to correct all effects simultaneously.

2.3 Challenges

To advance hyperspectral imaging from the scientific use to a consumer application, e.g., precision farming, change detection for land cover and cadaster updates, data preprocessing has to be automatized or at least made possible for non-scientists. As mentioned in Section 2.2, to accurately perform the correction of environmental effects additional measurements or at least the estimation of many external parameters is required. Generally, atmospheric

compensation is done to transform the data from at-sensor radiance to at-surface reflectance. This transformation is applied globally to the whole data set; local changes are not taken into account. These must be addressed to allow comparison of multiple data sets in a common setting.

The following sections give examples of common nonlinear effects in hyperspectral remote sensing data and some approaches for correction are introduced. These methods only work for a specific effect and are not suitable for correcting all effects simultaneously. The proposed NFN and the other algorithms from state of the art in MA in Chapter 3 are general better.

2.3.1 Atmosphere

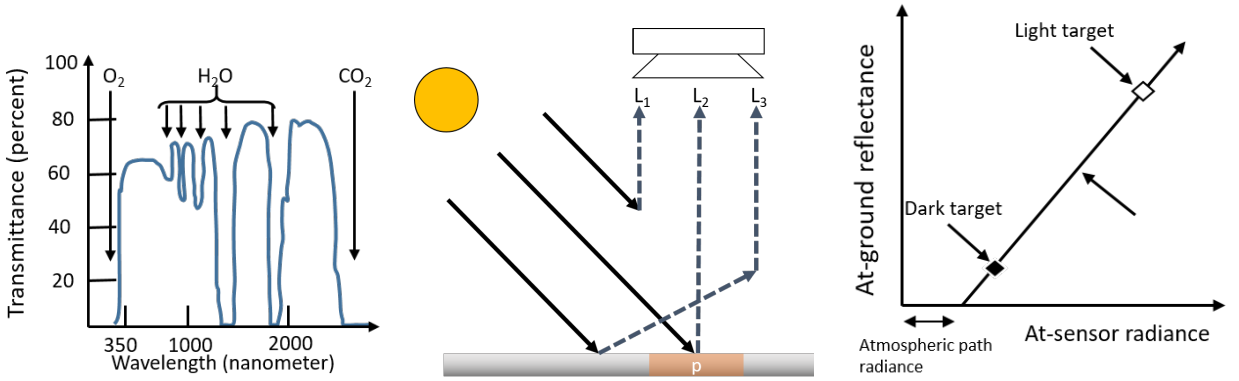
A significant challenge to transfering hyperspectral imaging from a laboratory setting to a remote sensing environment is the atmosphere between the illumination source and the sensor. This is often referred to as the sun-surface-sensor path [37]. The scattering and absorption of the radiation due to interaction with gases, aerosols, and clouds on its way from the sun to the sensor needs to be taken into account. Each form of interaction can be explained with a physical model, and most effects only affect certain wavelength ranges. The predominant effects are aerosol scattering and water vapor absorption. The general absorption properties of the atmosphere are characterized by the optical window. Figure 2.4(a) gives an overview of the wavelength ranges that pass through the atmosphere. For this thesis, only the wavelength range between 350-2500 nm plays a significant role. Here, mainly the O_2, H_2O, and CO_2 absorptions are relevant. A complete discussion of absorption properties of certain molecules for different wavelength ranges can be found in [45].

The model for the measured radiance in the sensor is a combination of three individual effects. Figure 2.4(b) shows the main components. The path radiance L_1 is caused by photons scattered towards the sensor by molecules in the atmosphere without interacting with the surface. L_2 is the reflected radiation, which contains the desired information about surface absorption features of the desired location. Finally, L_3 is called the neighborhood reflected radiation which is caused by scattering of radiation into the field of view of the sensor by

areas close to the desired location. This means atmospheric correction has to eliminate L_1 and L_3 while simultaneously retrieve the ground reflectance from L_2 [109]. These effects are wavelength-dependent, e.g., the path radiance decreases with wavelength and is very small for $\lambda > 800$ nm while the neighborhood radiation is negligible above $\lambda > 1500$ nm. A comprehensive guide on the properties and underlying physical models can be found in [104]. There exist numerous tools for atmospheric correction, e.g. ATCOR [103], MODTRAN [8], and FLAASH [83]. Generally, they require additional measurements like visibility, water vapor content and aerosol type for best possible performance.

Due to the complexity of properly using these commercially available solutions one approach for atmospheric correction is introduced here, that can easily be computed without any auxiliary information: the *Empirical Line Correction* (ELC). The ELC uses a set of calibration and validation targets directly from the acquired data set to transfer at-sensor radiance data to at-surface reflectance values. The idea is to calculate the ratio between the reflected radiation of a ground pixel and an ideal Lambertian standard surface under identical conditions of illumination, reflection geometry, and wavelength interval [30]. Ground reference measurements, e.g., with a field spectrometer to record accurate reflectance values for the calibration and validation targets can be used to calculate a correction model for each spectral band. In each band, the radiance and reflectance values of a dark and a bright object are used to determine a linear dependency in the form of gain and offset. The idea behind ELC is visualized in Figure 2.4(c) where the atmospheric path radiance equals the offset and the slope of the line between the dark and the light target equals the gain for the correction process. The accuracy of the correction can then be evaluated by how well the validation targets are modeled. Experiments have shown that the range between $0 - 70$ % of at-sensor radiance and at-surface reflectance has a linear dependency, essentially [116]. It is advised, however, to carefully check the results for artifacts as the initial condition of a Lambertian reflector in the scene is difficult to fulfill.

In practice, all of the above tools assume a constant atmosphere for the whole scene, which can lead to errors in the corrected data due to fluctuations during data acquisition [82].



(a) Atmospheric transmittance in percent for the wavelength range of $0 - 2500$ nm.

(b) Radiance components measured by the sensor. The area denoted by p stands for a single pixel recorded by the sensor.

(c) Schematics of ELC using dark and light reference targets.

Figure 2.4: Atmospheric transmittance, main components of measured radiance and ELC as a simple alternative to using an atmospheric model for the calculation of reflectance values.

While atmospheric correction is often cited as the required step to perform quantitative remote sensing, it is not the only external effect resulting in undesired spectral variability [37].

2.3.2 Shadows

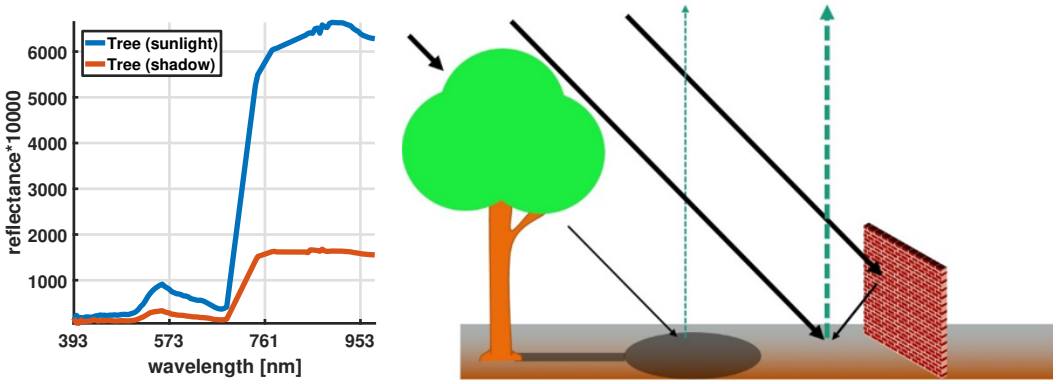
Shadows or more general variations of illumination, e.g., by clouds or tall buildings and trees, lead to an imbalance in the sun illumination [76]. Spectral signatures in shadow areas are dominated by diffuse illumination and show increased radiance levels in shorter wavelength ranges. This effect is caused by the Rayleigh scattering in the atmosphere and the higher aerosol scattering [110]. This results in spectra exhibiting less spectral variation in the blue spectral bands compared to a band of longer wavelengths. This phenomenon is depicted in Figure 2.5. The red spectral signature comes from a tree under a cloud shadow; the blue curve comes from a tree in a sunlit area. The tree blocks most of the incoming direct radiation, but diffuse scattering in the atmosphere allows to still get a signal in the shadow area. Since Rayleigh scattering is inversely proportional to the fourth power of the wavelength,

blue wavelengths are scattered more often than other colors. This leads to nonlinear effects when dealing with objects in and around shadows.

The first step in shadow correction is always to calculate a shadow mask, which gives the exact locations of areas to be corrected. This can be done with shadow invariant features, where the relationship between certain fixed wavelengths is used to transform the data to a color space that is independent of the color temperature [78]. In this case, the exact shadow borders can be manipulated with a threshold value. Another approach is based on a model for the solar position together with ray tracing and a *Digital Elevation Model* (DEM) [11]. Thus, the borders of shadow areas based on projective geometry can be extracted. This requires auxiliary information, which has to be recorded simultaneously, and high accuracy of the co-registration.

After the shadow mask is determined, the correction process can be started. Due to the nonlinear nature of the effect and the fact that shadows are not binary, i.e., either there is no shadow or a 100 % shadow, a scalar rescaling of shadow areas to the level of sunlit areas produces no accurate correction. A popular approach is to assume a linear mixing model, which interprets spectra as compounds of so-called endmembers, spectrally pure spectra characteristic for single materials. The decomposition of spectra into a matrix of known endmembers and their abundances is called spectral unmixing. When unmixing is calculated for both shadow and sunlit areas, it is possible to build a mapping between shadowed and unshadowed endmembers [94]. Other approaches use a decomposition based on the *Principal Component Analysis* (PCA) [86] or a kernel machine [119] to build a shadow-invariant transformation. One major drawback of these transformations is the difficulty in interpreting the results physically. This is due to the transformation to a feature space of different dimension [76].

Also, working with a binary shadow mask does not take the gradual transition between sunlight and shadow into account. A good example is the linear correction using the algorithm from [78] in Figure 2.6. Detecting the red and blue tarps after the correction is possible. However, the border areas are particularly difficult to correct [112]. Overall, shadows in the



(a) Comparison of two spectral signatures. One comes from a tree under cloud shadow, the other from a tree in bright sunlight.

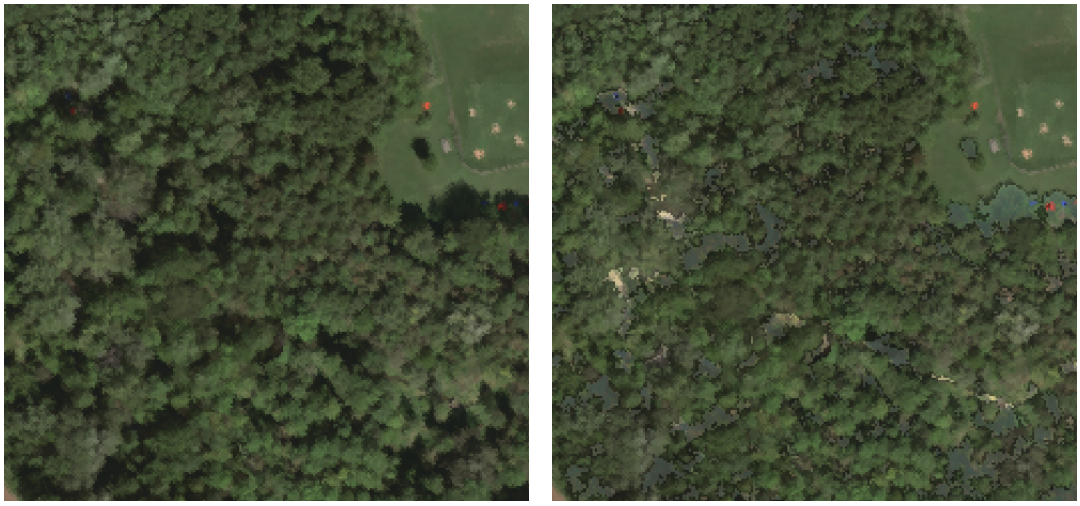
(b) Diagram for shadow and multiple reflection

Figure 2.5: Different physical effects influence the spectral signature. Occlusions mostly block longer wavelengths and the area bind an object is illuminated by diffuse light. Multiple reflections can be treated as a superposition of absorptions.

scene caused by clouds, buildings, and trees result in a relative shift towards the smaller wavelength ranges (blue shift) due to dispersion [112]. The newest iteration of ATCOR-4 has an additional tool to estimate a shadow mask from one band in the visible and one band in either near-infrared (0.8-1.0 μm) or ideally the shortwave infrared (around 1.6 and 2.2 μm). Unfortunately, this procedure requires multiple scene-dependent thresholds (core shadow, transition area) and is sensitive to shadow distribution in the scene.

2.3.3 Geometry and Surface Texture

The reflectance is a directional property of the surface materials and depends on the viewing and illumination geometry [103]. This effect can be described by the *Bidirectional Reflectance Distribution Function* (BRDF). It is commonly observed when the viewing and solar angle varies over a large angular range, e.g., due to object geometry or across-track brightness gradients [20]. One example is the different color of double-pitched roofs, where the side oriented towards the sun has higher reflectance compared to the surface facing away from



(a) Original data with shadows (b) Data after linear shadow correction

Figure 2.6: Example of linear shadow correction using the method from [78] on hyperspectral data. Targets (red and blue tarps) in the shadow can be detected after the correction. However, color distortions in the shadow area can be seen and the algorithm has no model to correctly transform the borders of the shadow areas.

the sun (and sensor).

For BRDF correction different physical models are required to balance specular and diffuse reflection terms [111]. Figure 2.7(b) gives an example of how multiple instances of diffuse and specular reflection add up. The requirement for correction of BRDF effects is especially strong in rugged undulating terrain with slopes facing different directions and scenes with geometric objects. An extreme example is depicted in Figure 2.3(a) with the metallic building resembling a half sphere. It shows dark areas on the backside but works essentially like a retro-reflector in the bright area, even going into the sensor saturation. A scatter plot of all pixels in the scene for three spectral bands is shown in Figure 2.3(b). Figure 2.3(c) shows only the scatter plot for the pixels of the metallic dome. It clearly illustrates the nonlinear behavior of reflectance for different viewing angles.

It is possible to eliminate some sources of this effect individually. The common across-track brightness gradients are caused by the changing viewing angle. A method called nadir normalization calculates the brightness as a function of scan angle to calculate an adjustment

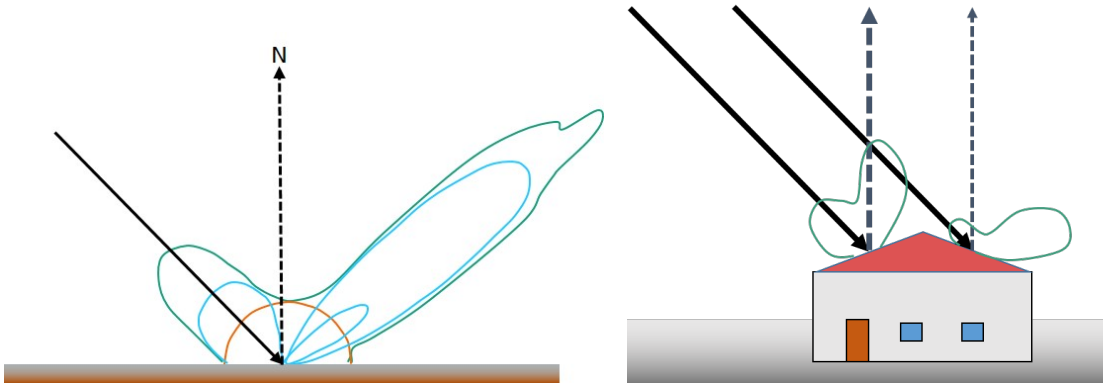
factor [107].

Correction is also required for areas with strong geometric variations. This leads to a variation in the solar zenith angle, the angle between the sun and the surface normal (see Figure 2.7(b), [104]). This can lead from areas of maximum solar irradiance to zero direct irradiance, i.e., shadows. Correction of this effect is difficult as the usual assumption of diffuse (Lambertian) reflection behavior can cause overcorrection in dimly lit areas [111] and are not applicable to objects with more specular reflections like the dome in Figure 2.3(a). This can lead to misclassification in bright areas. A simple way to address this problem is to use the local solar zenith angle and a threshold parameter to apply a variable correction factor depending on the incident angle. However, this approach does not take material dependent reflection as a composite of diffuse and specular components into account and requires a DEM to calculate the solar zenith angle.

A more sophisticated method is the *BRDF Effects Correction* (BREFCOR) [111, 108], which comes as an addon to ATCOR-4. However, it requires a DEM, estimation of surface cover classes and multiple training scenes with the same classes to calculate an anisotropy index per pixel.

2.3.4 Multiple Reflections

The presence of multiple reflections is difficult to quantify in hyperspectral remote sensing images. The effect is mostly considered together with nonlinear spectral unmixing [10]. Spectral unmixing derives mixture coefficients of different materials from measured spectra, when the spatial resolution is insufficient or multiple materials are intimately mixed. In this case, the model is based on bilinear interaction between two different components, e.g., the reflection from a red brick wall onto the ground in Figure 2.5(b). The reasoning is that a light ray is reflected multiple times while undergoing absorption at every interaction with another material [58]. The change is defined by the reflectance of each material the light ray interacts with.



(a) Sketch of BRDF (green) as the sum of a diffuse (orange) and multiple specular (blue) reflections.

(b) Sketch of BRDF depending on object geometry and viewing angle, a slope oriented away from the incident radiation has significantly lower reflectance towards the sensor. The BRDF has to be adapted to the wavelength of the measured radiation to account for effects such as iridescence and luminescence.

Figure 2.7: The measured reflection per band depends on the BRDF of an object. Different geometries between incident angle and sensor also affect the measured radiance.

The effect is also present inside the canopy of vegetation [127]. Here, it is mixed with self-shadowing and can lead to strong nonlinearities in the measured spectra. Only bilinear interactions are considered here. Higher-order multi-linear interactions are not expected to have a significant contribution to warrant their inclusion due to added complexity in the model [58]. These effects are also only considered while modeling a classifier or unmixing algorithm with a highly accurate physical model. Due to its complexity correction is only done for two-class problems [21], for measurements in a controlled environment [90], and within close range of the target [101].

Overall this effect only occurs on rare occasions it is mostly ignored due to its insignificance and the complex correction approaches. Using a data driven approach that can correct any of the effects mentioned in this chapter will also reduce the impact of multiple reflections.

Chapter 3

Related Work

After the discussion of data preprocessing in Section 2.2 and methods for correcting individual nonlinear effects in Section 2.3, this chapter focuses on data-driven approaches that simultaneously address all nonlinear effects regardless of their source. These are generally detached from physical models, but rather work based on a geometric framework to align multiple data sets in a common domain that allows further analysis.
This chapter first introduces different approaches and terms to the general problem of data alignment in Section 3.1. Section 3.2 discusses individual techniques for ML that can be used to describe the data sets in a new system. Section 3.3 expands this approach to multiple data sets by presenting methods for *Domain Adaptation* (DA) and MA. These approaches generally use high-dimensional domains for the alignment process. The inversion to a dedicated target domain for physical interpretation, however, requires solving of the pre-image problem, which is discussed in Section 3.4. Finally, Section 3.5 summarizes the examined data alignment techniques and sets the proposed algorithm of this thesis in contrast.

3.1 Introduction to Data Alignment Techniques

Joint evaluation of multiple similar data sets is a common problem in machine learning. High-dimensional data, such as hyperspectral data, is often much simpler than the dimen-

sionality would indicate [106]. In particular, a given high-dimensional data set may contain multiple features that are caused by the same underlying effect. The stereotypical example for this are pictures of the same object with a single moving light source. Treating each image as a vector in high-dimensional space appears very complex at first. A way to simplify the data representation is desirable. When the behavior of an effect, in this case the moving light source, is known the corresponding components can be parametrized and the underlying feature space is embedded in a two-dimensional manifold [89]. Modeling individual effects is often not possible due to data dimensionality, the number of different effects or unknown sources. Thus, machine learning in the form of MA techniques is used in this thesis to find suitable data representations.

Through MA, it is possible to formulate a framework for discovering a unified representation of multi-temporal and multi-source data sets [119]. The goal is to utilize the inherent relationship between samples of each data set, and to derive a way to map all data sets to a common space.

This section is dedicated to the different techniques and approaches to extract the necessary information from the data sets.

Depending on the available information data alignment can be addressed as an unsupervised, a semi-supervised, or a supervised problem. Each category exploits different information to describe the inherent structure of the data [34].

Unsupervised learning tries to describe the inherent structure of an unlabeled input, generally without a set way to evaluate the results. The most iconic unsupervised approach is the PCA [136], where a new representative orthogonal basis from the eigenvectors of the data is calculated to reveal the essential data patterns. As the eigenvectors corresponding to the largest eigenvalues encode the parts of the data with the highest variance, data reduction by discarding those corresponding to smaller eigenvalues is a popular approach. PCA is a linear transformation that requires no parameters. Thus, it cannot deal with strong nonlinear deformations. This problem is alleviated in [31] by introducing the kernel PCA, where the data is transformed to a high-dimensional reproducing kernel Hilbert space, which facilitates

a description of nonlinear manifolds with the use of linear operations. During recent years, some unsupervised methods with interesting properties were introduced. A famous method, originally introduced in [117], is the *Isometric Mapping* (ISOMAP). ISOMAP computes a quasi-isometric, low-dimensional embedding for a set of high-dimensional data points by approximating the manifold with a neighborhood graph, calculating the shortest paths in the graph, and performing low-dimensional embedding. Contrary to the global approach of ISOMAP, *Locally Linear Embedding* (LLE) aims at recovering a low-dimensional structure of the data by analyzing its local interaction between samples. Starting with a neighborhood graph, LLE determines a low-dimensional embedding that preserves the representation of individual samples by linear combinations of its neighbors [2]. Both approaches are discussed in more detail in Section 3.2.

The *Canonical Correlation Analysis* (CCA) [92] and its kernelized approach [71] draw independent and identically distributed samples from two data sets to form input pairs with the goal to maximize the correlation. In [129] local neighborhoods are characterized by weight matrices to find underlying structures. Since the computational cost is very high even for small neighborhoods, another approach is introduced in [97] that tries to locally fit a B-spline curve to the data to calculate matching scores between manifolds. Another approach is *Unsupervised Image Sets Alignment* (UISA) [25], where a reference image that is pre-structured into a set of locally linear models is used as a reference. Multiple linear transformations are then learned to align new images with the reference. Other methods exploit the local geometry of the data representation and optimize the geodesic flow between subspaces to find a common representation in a high-dimensional space [44, 42].

In general, unsupervised approaches adapt to new data sets and can help to find underlying structures, while introducing prior knowledge about the inherent structure is very complex and often limits versatility.

Semi-supervised learning exploits labeled as well as unlabeled samples to model the inherent structure. This approach is motivated by the fact that the labeling process is often costly, while unlabeled information is available in abundance. Labeled samples can serve as

a structure with high reliability while unlabeled samples are used to refine the model [16]. In MA, the common problem is to find a mapping between a data set of labeled samples and a new unlabeled data set. Popular examples are the *Semi-Supervised Manifold Alignment* (SSMA) [125], from which the *Kernel Manifold Alignment* (KEMA) [119] was derived. They operate by calculating adjacency matrices to set up an undirected graph and optimize the graph Laplacian for (dis-)similarities. This approach causes samples of the same class labels to become closer, while simultaneously pushing those of different class labels apart. Other approaches can be found in [63] and [139], where kernel machines are used to adapt a classifier to possible variances in subsequent data. Several unsupervised approaches can also be adapted to work with existing labels in the target domain to guide the calculation [44, 42]. The semi-supervised approaches often suffer from high computational complexity and memory requirements, as most methods require some form of adjacency matrix that can get large, considering the abundance of unlabeled data available.

Supervised learning calculates a model for input labels in all available domains. Labels in the target domain guide the learning process [34]. In this category, the goal is to find projections to a common latent space, where all data sets have similar statistical characteristics [119]. A lot of unsupervised and semi-supervised learning approaches can be adjusted to incorporate additional labeled samples in all domains [125, 119, 44, 42].

The underlying framework becomes less adaptive to new input data but is better tailored to solving specific problems. For practical experiments, almost all of the former approaches use labeled data in all domains and can, thus, be considered supervised.

However, each mentioned MA method has its caveats. ISOMAP computes a globally optimal representation but suffers from high computational cost and memory requirements. In [2] the authors propose a partitioning of the data to compute multiple solutions on a smaller scale and fuse the resulting manifold representations with a series of linear transformations. There exist alterations of ISOMAP that tackle the computational complexity and make it applicable for MA of large data sets [4]. However, depending on the data structure and the chosen neighborhood size, ISOMAP is prone to short-circuit errors when the folds of the

manifold are small, or the data is noisy. It can also require careful preprocessing of the data to guarantee topological stability [5]. LLE produces a locally optimal mapping, but since the feature space has a much lower dimension multiple sources, e.g., different classes can become mixed. As a consequence, the necessary features in the data can be lost. LLE works best on data sets with uniform sampling density, but can produce distorted embeddings when the manifold dimension is larger than one [106, 144].

Using kernel methods and transforming the data to a potentially infinite dimensional feature space can circumvent the mixture of classes [81]. However, results obtained in such a feature space do not permit interpretation in meaningful physical units, they can only be exploited for decisional problems like classification [119].

If the physical interpretation of the aligned data is the goal, the data needs to be inverted back to the source domain. The main problem is that methods based on kernel machines have to deal with the pre-image problem [145, 69]. The pre-image problem is concerned with finding corresponding patterns in the original domain for points in the potentially infinite dimensional feature space. Existing methods require specific kernel methods together with a particular kernel function to compute an approximate solution [69, 132, 77]. These are often problem-specific, depend on fine-tuning of hyperparameters and are computationally challenging [119].

The remainder of this chapter is divided into two parts.

Section 3.2 introduces commonly used approaches for finding the low-dimensional manifold structure through ML. For hyperspectral data, the basic assumption is that the true dimensionality of the data is much lower than the number of bands due to the high correlation between neighboring bands. ML is an approach to (nonlinear) dimensionality reduction. Visualization of high-dimensional data is difficult and a simple approach is to project the data on a lower-dimensional subspace. This allows better visualization of the data structure, but it comes at the loss of information, depending on the projection. To mitigate this, a number of approaches exist that help preserve the information while simultaneously finding a projection that reduces the dimensionality. Section 3.3 then deals with finding projections between

data sets or their low-dimensional projections in the form of MA. This is based on the idea in [3], where a manifold constraint to the problem of correlating sets of high-dimensional vectors was first introduced. MA operates on the idea that disparate data generated by a similar process has common features in terms of manifold representation [119].

3.2 Manifold Learning

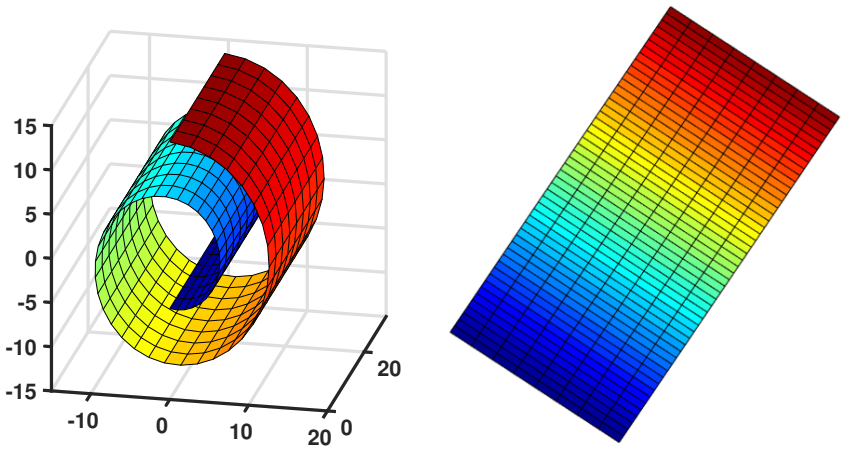
ML is an approach to nonlinear dimensionality reduction. This is an interesting topic for analysis and visualization of hyperspectral data with up to hundreds of bands. Generally, hyperspectral data is visualized by selecting three bands and combining them to an RGB image. Those bands are commonly selected to resemble broadband sensors like RGB cameras. This is only a small part of the recorded information, but visual interpretation of more than three bands simultaneously can be difficult. A common assumption is that the high-dimensional data is comprised of structures with lower dimensionality [118]. This is reasonable as neighboring bands are usually highly correlated.

By representing measurements in high-dimensional spaces in the form of low-dimensional manifolds it is possible to generalize vector calculus to curved spaces [118]. An example for a low-dimensional manifold is given in Figure 3.1. It shows the swiss roll data set in 3D, its curvature, and its efficient 2D representation. The parameterization for the swiss roll can be acquired by calculating the principal curvature, i.e. the eigenvalues of a shape operator at different points. This allows mapping the data to a 2D Cartesian coordinate system.

In this section, the concept of ML is introduced by defining a mathematical framework to describe these structures. After this, prominent examples are given.

3.2.1 Mathematical Framework

For the following chapters, a way to mathematically describe the procedures is required. In this section, the important vocabulary is introduced.



(a) The swiss roll data is a 2D manifold in a 3D space; depicted with color-coded curvature along the manifold starting at $(0, 0, z)$.

(b) Parameterization along principal curvature allows unrolling the data set onto a plane, essentially describing the data set with 2D coordinates.

Figure 3.1: Example of parameterizing the swiss roll data set to map it to a lower-dimensional space.

The first step is to define a suitable space where the calculation is carried out. For this a topological space is chosen. The following definition uses open sets to define a topological space [1].

Definition 3.2.1 (Topological space). A topological space is an ordered pair $(\mathcal{X}, \tau)$, where $\mathcal{X}$ is a set and τ is a collection of subsets of $\mathcal{X}$, satisfying the following axioms:

1. The empty set and $\mathcal{X}$ itself belong to τ.
2. Any (finite or infinite) union of members of τ still belongs to τ.
3. The intersection of any finite number of members of τ still belongs to τ.

Definition 3.2.2 (Hausdorff space). A topological space $\mathcal{X}$ is called a Hausdorff space, if it satisfies the third separation axiom, i.e., for any two distinct points $x, y \in \mathcal{X}$ there exists a neighborhood U of x and a neighborhood V of y such that $U \cap V = \emptyset$.

Additionally, the second axiom of countability is required.

Definition 3.2.3 (Second axiom of countability). If $\mathcal{X}$ is a second-countable space there exists some collection $\mathcal{U} = \{U\}_{i=1}^{\infty}$ of open subsets of $\mathcal{X}$ such that any open subset of $\mathcal{X}$ can be written as a union of elements of some subfamily of $\mathcal{U}$.

This means that a second-countable space is a topological space whose topology has a countable base. For example, the Euclidean space $\mathbb{R}^d$ with its usual topology is second-countable. A base of open balls comprised of all balls with rational radii and rational center coordinates is countable and still forms a basis of $\mathbb{R}^d$.
A manifold $\mathcal{M}$ can then be defined as follows.

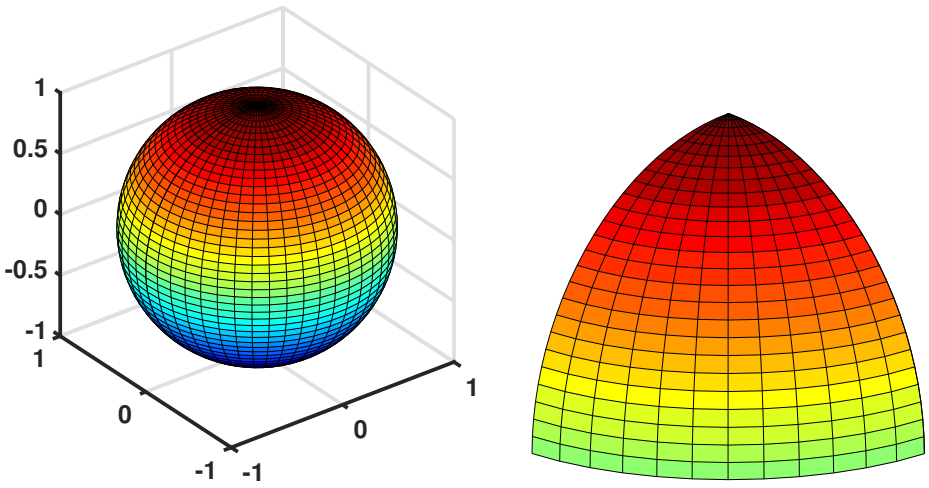
Definition 3.2.4 (Manifold). Let $\mathcal{M}$ be a topological space. $\mathcal{M}$ is called d-dimensional manifold or d-manifold, if

1. $\mathcal{M}$ is a Hausdorff space
2. $\mathcal{M}$ satisfies the second axiom of countability, i.e., $\mathcal{M}$ is a second-countable space
3. $\mathcal{M}$ is locally Euclidean, i.e., for any point $x \in \mathcal{M}$ there exists a neighborhood U that is homeomorphic to the Euclidean space $\mathbb{R}^n$.

From this definition, it follows that $\mathbb{R}^d$ is a manifold since it is trivially homeomorphic to itself. Subsequently, this is also the space commonly used for hyperspectral and many other data sets.
Calculating with manifolds requires a way to define a coordinate system on the manifold. This is done in the form of combining individual charts to build an atlas. They are defined as follows.

Definition 3.2.5 (Chart). A chart (U, φ) for a topological space $\mathcal{M}$ is a homeomorphism φ from an open subset U of $\mathcal{M}$ to an open subset of a Euclidean space.

With charts, it is possible to locally define a metric on a manifold $\mathcal{M}$ to use, e.g., for distance calculation without taking the curvature into account. A fitting analogy is the measurement of interior angles in a triangle. In Euclidean geometry, the sum of all interior



(a) A sphere in 3D space

(b) One eighth of the sphere such that the sum of interior angles of the triangle are equal to $3\pi/2$.

Figure 3.2: Example of parameterizing the segment of a sphere to map it to a lower-dimensional space.

angles in a triangle is π. This works as long as the triangle is drawn on a level surface. When curvature is introduced to the surface the total sum of interior angles changes. E.g., if $\mathcal{M}$ is a spherical manifold the sum of interior angles is $> \pi$ in spherical geometry. A demonstrative example is a triangle where one side is on the equator of a sphere with a length of $\pi r/2$, where r is the radius of the sphere. If the remaining two sides are going north at an angle of $\pi/2$, they will intersect in the north pole at an angle of $\pi/2$. Thus, a triangle with a sum of interior angles of $3\pi/2 > \pi/2$ was constructed. Figure 3.2 shows a sphere and the corresponding triangle to illustrate that example. With this local definition it makes sense to combine multiple charts to form an atlas such that the whole manifold can be described.

Definition 3.2.6 (Atlas). An atlas for a topological space $\mathcal{M}$ is a collection $\{(U_\alpha, \varphi_\alpha) \,|\alpha \in A\}$ of charts on $\mathcal{M}$, indexed by a set A, such that $\bigcup_{\alpha \in A} U_\alpha = \mathcal{M}$. If the codomain of each chart is the d-dimensional Euclidean space, then $\mathcal{M}$ is a d-manifold.

In this thesis and for the example algorithms given hereafter and in Section 3.3, the codomain or target set of the individual charts will always be a Euclidean space. They come equipped with all required metric conditions and generalize well into higher dimensions [14].

This makes them intuitive to work with. In practice, manifolds are used to generalize vector calculus of $\mathbb{R}^d$ to curved spaces via local charts homeomorphic to an open subset of $\mathbb{R}^d$.
Overall, ML is an approach to nonlinear dimensionality reduction. These are often derived from linear methods, e.g., the extension of PCA with a kernel to deal with nonlinear structures.
Some prominent examples are introduced in the following sections.

3.2.2 Principal Component Analysis

The PCA is maybe the most prominent approach to data representation and dimensionality reduction [136]. The PCA calculates a new orthogonal coordinate system which maximizes the variance in the direction of each component in descending order. In the context of ML, the principal components define a global (trivial) manifold as the new coordinate system. It is a rotation of the original coordinate system. For dimensionality reduction, the components corresponding to the largest eigenvalues encode the maximal variation. This means that a truncated PCA using the k components corresponding to the k largest eigenvalues encodes equal or more variance than any other combination of components. A truncated PCA with $k < d$ projects each data point onto a lower dimensional subspace which best approximates the data X in the least squares sense.
The PCA is arguably the most popular ML algorithm due to its simplicity. The relationship between data samples is not changed during the calculation as only the coordinate system is adjusted. The data is centered at the origin of a new coordinate system. The rotated basis given by the calculated principal component vectors can be considered individual 1-manifolds. However, their union is neither a 1-manifold nor is it a d-manifold [31]. Reducing the dimensionality by eliminating dimensions corresponding to small eigenvalues simplifies the data at the cost of possible loss of information, e.g., features that are only present in a few samples and are, thus, not significant in terms of total data variation [136].

3.2.3 Graph-based Manifold Learning

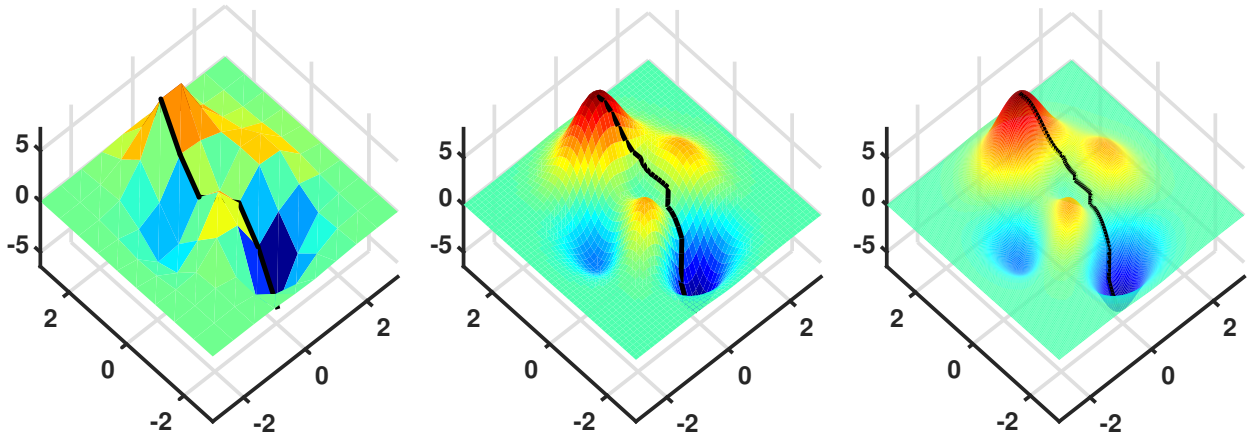
When transitioning from a global model to a data-driven approach, adjacency between samples can be used to infer the data structure. Enforcing certain rules a *k-Nearest Neighbor* (NN) graph or an ϵ-ball neighborhood graph can be used to approximate the distances between samples along the curvature of the manifold [121, 9]. Building approximate geodesics on a graph is influenced by how well the graph samples the underlying manifold. This is demonstrated on the peaks plot from MATLAB using different sampling sizes in Figure 3.3. Only the immediate neighbors are used to build the graph.

With graph-based ML, the goal is to describe the distance between samples along connected paths on the surface of the manifold — these paths approximate geodesics, which are the generalization of a straight line to curved spaces in differential geometry, i.e., the shortest path between two points on a curved surface [9]. Thus, an alternative description of the data along its surface is possible. The Dijkstra's algorithm can then be used to compute the shortest paths in the calculated graph [64].

The construction of such graphs is straightforward. Each sample $x \in \mathcal{X}$ is connected by edges to the k nearest (by some metric) neighbors or all neighbors within an ϵ-ball of a specified radius ϵ, respectively. A weakness of the k-NN approach is the fact that the distance between samples can be large, e.g., for outliers with few ($< k$) neighboring samples. When the goal is to find the latent data structure, those edges are not contributing useful information. One way to circumvent this is the symmetric k-NN graph. Consider two vertices $x_i, x_j \in \mathcal{X}$ with $i \neq j$. x_i and x_j are connected, if x_i is in the neighborhood of x_j and vice versa.

For the ϵ-ball neighborhood graph two vertices are connected by an edge if the distance between them is smaller or equal to ϵ. A completely connected graph can be obtained by setting ϵ to the maximal distance between any two vertices in the data set. Problems with this approach can occur with small ϵ or sparsely sampled data resulting in multiple disjointed components. Then, the ϵ-ball neighborhood graph may not be able to properly recover the geometry of highly curved manifolds.

The selection of suitable parameters k or ϵ can be crucial to solving a specific problem. A



(a) Approximate geodesic on peaks plot with 100 nodes. The shortest path from the global minimum to the global maximum has a length of 14.5.

(b) Approximate geodesic on peaks plot with 2500 nodes. The shortest path from the global minimum to the global maximum has a length of 15.7.

(c) Approximate geodesic on peaks plot with 4000 nodes. The shortest path from the global minimum to the global maximum has a length of 15.9.

Figure 3.3: Building approximate geodesics on a graph is influenced by how well the graph samples the underlying manifold. Constructing the paths was done using only the eight immediate neighbors per sample.

study of the influence of parameter selection on clustering results can be found in [79].

3.2.4 ISOMAP

ISOMAP stands for Isometric mapping and was first introduced in [117]. It computes a quasi-isometric, low-dimensional embedding for a set of high-dimensional data points. The pairwise adjacency matrix is exploited through approximating the geodesic distances along the edges of a graph as introduced in Section 3.2.3. ISOMAP is performed in three steps.

1. Approximate the manifold with a neighborhood graph on the data set,
2. Compute pairwise shortest paths with the Dijkstra algorithm, and
3. Perform multidimensional scaling on the matrix of the shortest geodesic distances

ISOMAP has many desirable properties. It works well on developable (i.e., Gaussian cur-

vature $K = 0$ everywhere, see [66]) and convex manifolds. Then, the geodesic distances approximated by the graph are similar to the Euclidean distances in the reduced space after multidimensional scaling is performed. Although ISOMAP has some theoretical convergence guarantee, it requires a certain data quality for geodesic approximation. The following drawbacks are discussed in [118, 5] in more detail:

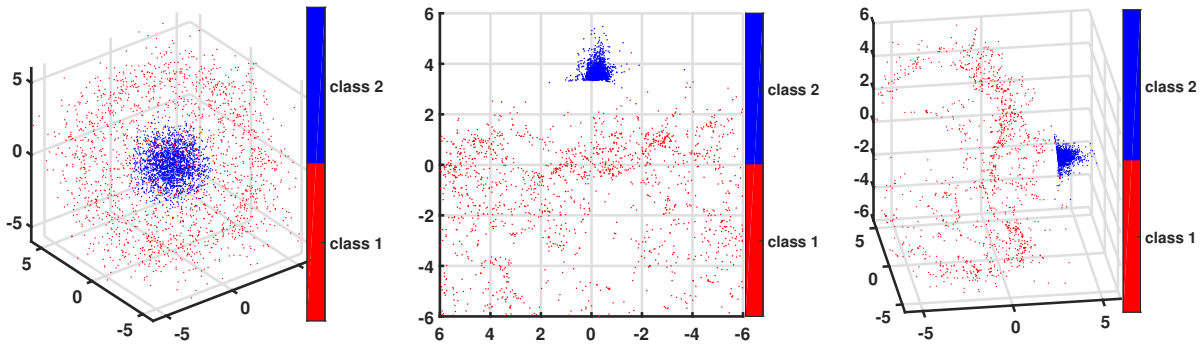
1. Manifolds without boundary cannot be unfolded into the reduced space, e.g., a donut-shaped manifold

2. The algorithm is topologically unstable and requires careful preprocessing of the data

3. Depending on the data structure and the chosen neighborhood size, it is prone to short-circuit errors when the folds of the manifold are small, or the data is noisy. This can be seen in Figure 3.4, where the mapping of different classes fails due to noise.

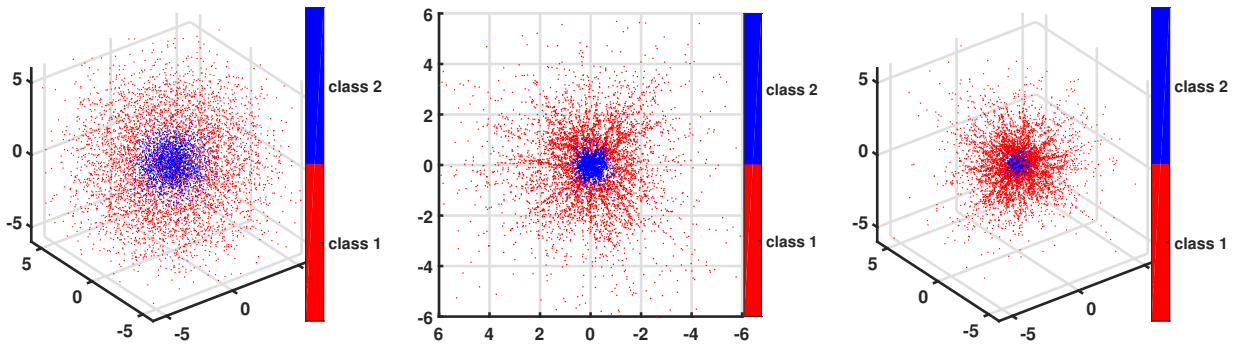
4. As a global method, ISOMAP scales badly for large data sets, as it requires the decomposition of a dense matrix. A possible solution is the use of the Landmark ISOMAP algorithm, where a set of $n \ll N$ samples from the original data X are used as representatives [115]. This reduces the computational complexity from $\mathcal{O}(kN^2log(N))$ for shortest path estimation and $\mathcal{O}(N^3)$ for the $N \times N$ matrix factorization to $\mathcal{O}(knNlog(N))$ and $\mathcal{O}(nN^2)$, respectively. With this approach, only the distances from a sample to all landmark points is calculated. Similar to a truncated PCA the transformation can be expressed by computing a weighted linear combination of eigenvectors of the adjacency matrix. The weights are determined by the squared distance of a sample to the landmark points.

If the neighborhood for graph calculation is too large, connectivity can change dramatically even with a single short-circuit error [5]. Lowering the neighborhood, on the other hand, can lead to many disconnected graphs and a fragmented manifold, which makes an accurate estimation of geodesics impossible. A priori information about the global geometry of the data is required to select the proper k or ϵ. According to [117], using an algorithm to



(a) Original data (b) ISOMAP (2D) ISOMAP (3D)

Figure 3.4: Example of well performing ISOMAP algorithm. Classes are separated in the final projection.



(a) Original data (b) ISOMAP (2D) ISOMAP (3D)

Figure 3.5: Example of short circuiting ISOMAP algorithm. If neighboring samples of different classes are too close together, ISOMAP representation leads to intimate mixtures of classes.

find "meaningful low-dimensional structures hidden in high-dimensional observations" are exactly the cases where this a priori information is unavailable.

An example of a proper ISOMAP calculation is shown in Figure 3.4. Figure 3.5 shows an example where the ISOMAP calculation for two classes fails. In both cases, the data is shaped like a ball, and one class is inside of the other, but with disjoint manifolds. When the noise level increases in Figure 3.5 the disjoint manifolds grow closer, and the neighborhood search is no longer able to separate the different classes. This leads to a mixture in the final mapping as the result of a short-circuit error.

3.2.5 Locally Linear Embedding

The LLE was first introduced in [106]. Contrary to the ISOMAP approach, this algorithm aims at recovering the global low-dimensional structure of a data set by analyzing its local interaction between samples. The calculation is done along the following pattern.

1. Calculate a neighborhood graph on $\mathcal{X}$

2. For each sample $x \in \mathcal{X}$ calculate weights that characterize the sample as linear combinations from its neighbors

3. Find a low-dimensional representation of each sample based on the eigenvectors while simultaneously preserving the linear combination in the embedded space for all samples and their neighborhood

The reconstruction error is given as the quadratic function of the squared errors of the linear combination. The weights are subject to a sum-to-one constraint. Together this makes the transformation invariant to rotations, rescalings and translation [106]. A neighborhood preserving map is then calculated by keeping the weights fixed and optimizing the coordinates of a low-dimensional projection by solving a sparse $n \times n$ eigenvalue problem, where n is the number of samples of the data set.
According to [106], LLE produces better results with many problems compared to ISOMAP, and it is generally faster to compute when taking advantage of sparse matrix algorithms. However, LLE shows poor performance when the sampling density of the data set varies a lot. Another problem is the ambiguity of linear combinations when a neighborhood has more than N points (only applies to ϵ-ball neighborhoods). For a data set with sufficiently uniform sampling, it makes sense to determine the number of NNs per sample $k \leq N$ using cross-validation [2].

3.3 Manifold Alignment Algorithms

After reviewing different approaches to learn the latent manifold structure in Section 3.2, an interesting concept is to align multiple manifolds with a certain correspondence from different data sets. MA is also often interchanged or at least closely related to the terms DA, covariate shift, and transfer learning. It exploits the assumption that different data sets produced by similar procedures share a common underlying manifold representation. The goal is to define a common space $\mathcal{C}$ and suitable projections to transform the data sets to this space while preserving the corresponding structures or topology. If the alignment in the common space is successful, it is possible to transfer information, e.g., labels for ground truth, training data, etc. from one domain to another. The main objective according to [42] is to adapt classifiers trained on one data set to another while maintaining a good performance. While the concept of MA can be extended to arbitrarily many initial data sets, it makes sense to restrict discussion to the alignment of only two data sets to simplify the notation. In this section, the mathematical framework from Section 3.2.1 is extended. Instead of aligning a single data set to a low-dimensional space homeomorphic to $\mathcal{R}^d$, the goal is now to align multiple data sets in a common space.

3.3.1 Mathematical Framework Extended

This section introduces the necessary terms and definitions for accurately describing the MA framework. The idea of aligning two data sets is usually based on the following assumptions and input data:

1. The source domain $\mathcal{X}^*$ with desirable properties, e.g., good pre-processing, illumination conditions, and enough labeled samples to build a robust classifier.

2. The target domain $\mathcal{X}$ usually refers to a data set that is acquired by similar mechanics but with different characteristics compared to the source domain.

3. The common domain $\mathcal{C}$ where both data sets are aligned. $\mathcal{C}$ can have a different dimensionality than $\mathcal{X}$ and $\mathcal{X}^*$.

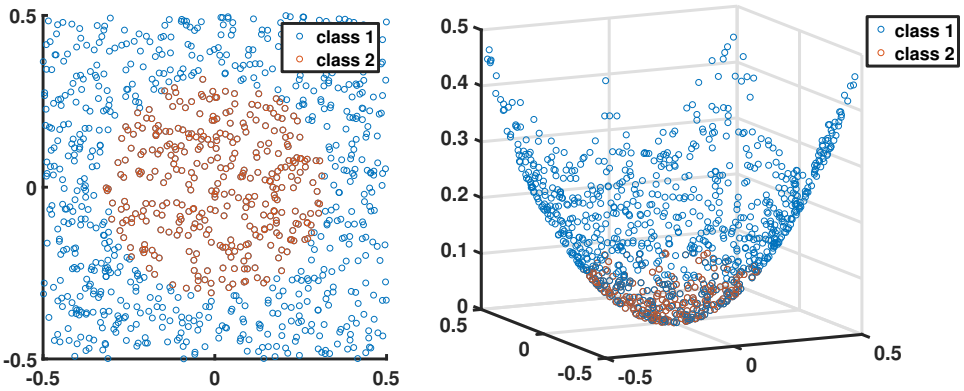
Since the alignment of multiple data sets and the description of their inherent nonlinear structure can be difficult in the same dimension, the kernel trick is often used to perform the alignment in a possibly infinite dimensional space. This allows the separation of data sets and the calculation of nonlinear decision planes by transforming the data into a higher dimensional feature space, where a linear separation is possible.

Definition 3.3.1 (The Kernel Trick). Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$ be samples in a d-dimensional domain. The function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel. The explicit calculation of a possibly infinite dimensional representation of individual samples with a feature map $\varphi : \mathcal{X} \to \mathcal{C}$ is not necessary, if K satisfies

$$K(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle_{\mathcal{C}}, \tag{3.1}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{C}}$ is an inner product.

With the Kernel Trick, the explicit mapping of samples to a possibly infinite dimensional feature space can be circumvented. Instead only the relationship of the samples in the form of inner products needs to be computed. This is often computationally cheaper than calculating the coordinates of the whole data set in the feature space explicitly [42].
Many algorithms apply this in the form of a kernel matrix $\boldsymbol{K}$ where $\boldsymbol{K}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \varphi(\boldsymbol{x}_i), \varphi(\boldsymbol{x}_j) \rangle_{\mathcal{C}}$.
An example is given in Figure 3.6. Here, the two classes of the data set are not linearly separable in the original space. Using the feature map $\varphi(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$, where x_1, x_2 are the scalar entries of the sample vector $\boldsymbol{x} \in \mathbb{R}^2$, the kernel becomes $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}' + \boldsymbol{x}^2 \boldsymbol{x}'^2$. With this the samples are mapped to a 3-dimensional space where a linear decision boundary can easily be calculated.
For MA, this property is used to perform the necessary transformations linearly in the high-dimensional feature space. The same transformation would be nonlinear with respect to the



(a) Scatter plot of original data set

(b) Scatter plot of data after the feature map $\varphi(x_1, x_2) = (x_1, x_2, x_1^2 + x_2^2)$ was applied

Figure 3.6: Scatter plot of a data set with two classes before and after the feature map was applied. After the transformation to a higher dimensional space a linear decision plane can be used to separate the classes.

original space [119]. With this, the basic idea behind most MA algorithms is introduced. In the following, a few selected algorithms are shortly described.

3.3.2 Sampling Geodesic Flow

The idea of *Sampling Geodesic Flow* (SGF) was first introduced in [44]. This approach was developed to align samples that are far apart on the underlying manifold due to drastic changes, e.g. geometric variations. The idea is to use intermediate subspaces to learn domain-invariant features, specifically in the form of interpolated features between the source and the target domain. The goal of a subsequent learning algorithm is then to select the most invariant features.

The algorithm performs the following steps:

1. Constructing a geodesic flow curve which connects the source and the target domain along the Grassmann manifold. A Grassmann manifold $G(d, D)$ is a space that parametrizes all d-dimensional subspaces of the D-dimensional vector space ($d \leq D$).

2. Sampling a fixed number of subspaces from this curve

3. Projecting original feature vectors onto these subspaces and concatenating the projected vectors into "feature super-vectors"

4. Reducing the dimensionality of the "super-vectors"

5. Using the resulting representation as new feature vectors to construct classifiers

While the original results are promising, [42] mainly criticizes the impracticality of parameter optimization via cross-validation for the number of sampled subspaces and their dimensionality. Instead, the basic idea of SGF is extended using a kernel approach in the next section.

3.3.3 Geodesic Flow Kernel

The idea of the *Geodesic Flow Kernel* (GFK) approach is the explicit construction of an infinite-dimensional feature space that assembles information for both the source domain $\mathcal{X}$ and the target domain $\mathcal{X}^*$, which serves as a reference [42]. The geodesic flow parametrizes the smooth change from the source to the target domain. Let $\Phi(t)$ be the subspace for a $t \in (0, 1)$. $\Phi(t)^T x$ defines the projection of a feature vector x onto this subspace. The parameter t controls the appearance of the projection. A t close to 0 corresponds to a projection more likely from the source domain. Conversely, a t close to 1 corresponds to a projection closely related to the target domain. The goal is now to find a projection satisfying this for a set of features that are characteristic for both domains. A classifier built on this projection is likely to perform well on both data sets.
The selection of t (or a sampling of t) is made simple using the kernel trick (see Section 3.3.1). For two feature vectors x_i, x_j the projections $\Phi(t)$ for a continuous t from 0 to 1 are calculated. The results are concatenated into the infinite-dimensional feature vectors z_i^∞ and z_j^∞. The geodesic flow kernel can then be written as

$$\langle z_i^\infty, z_j^\infty \rangle = \int_0^1 \left(\Phi(t)^T x_i\right)^T \left(\Phi(t)^T x_j\right) dt = x_i^T G x_j \tag{3.2}$$

where $G \in \mathbb{R}^{D \times D}$ is a positive semi-definite matrix. [42] shows how G can be calculated in closed-form.

The only free parameter of this approach is the dimension d of the subspaces. To extend this approach to an unsupervised technique, the authors propose an automatic way to estimate the optimal dimension by using a subspace disagreement measure. The idea is to compute the PCA of the source and target data individually, and the PCA of both data sets combined. All three PCAs are similar when the two data sets are close to each other on the Grassmannian manifold. The dimension d should be chosen high enough to encode the variances in the source domain, but not so high that the two subspaces have orthogonal directions. The reason for the last statement is that DA fails when variances captured in one subspace cannot be transferred to the other subspace. Further information is available in [42, 41].

The supervised GFK approach is used in Section 6.5.3 to compare results to the proposed method of this thesis. It is shown that GFK, FT of an SVM classifier trained on one data set produces good results on another data set. However, inversion to a reference domain where an original model could be used for interpretation was not successful. Additionally, the alignment with GFK shows problems with strong nonlinearities like shadows and BRDF effects that have no continuous representation, e.g., two-sided roofs where only the samples of two different viewing angles are measured.

3.3.4 Semi-Supervised Manifold Alignment

The SSMA uses labeled information of different classes to encode knowledge about data structure and geometry of the data [126]. Unlabeled data can be added to refine the structures. The relationship between labeled and unlabeled samples can be encoded with an undirected graph in the form of the graph Laplacian matrix $\boldsymbol{L}$. Let $\boldsymbol{X} \in \mathbb{R}^{d \times n}$, where d is the dimensionality and n the number of samples. $\boldsymbol{L}$ can then be defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \tag{3.3}$$

where $\boldsymbol{D} \in \mathbb{R}^{n \times n}$ is the degree matrix, a diagonal matrix counting the number of connections for each sample. $\boldsymbol{W} \in \mathbb{R}^{n \times n}$ is the adjacency matrix which only contains non-zero entries between neighboring samples which are determined by k-NN or an ϵ neighborhood. $\boldsymbol{L}$ encodes the norm of derivatives for the decision function along the graph built upon the samples [119].

The goal is then to transform the data sets to a latent domain $\mathcal{C}$ where samples of the same class become closer while simultaneously increasing the inter-class distance. This is done by using three different adjacency matrices:

1. $\boldsymbol{W}_s$ is the similarity matrix with entries $\boldsymbol{W}_s^{ij} = 1$ if the corresponding samples belong to the same class and 0 otherwise, including the unlabeled samples.

2. $\boldsymbol{W}_d$ is the dissimilarity matrix with entries $\boldsymbol{W}_d^{ij} = 1$ if the corresponding samples belong to different classes and 0 otherwise, including the unlabeled samples.

3. $\boldsymbol{W}$ represents the topology of the domains, e.g., by calculating adjacency matrices with a *Radial Basis Function* (RBF) or a k-NN graph for each domain separately and joined in a block-diagonal matrix.

This leads to three different graph Laplacian matrices $\boldsymbol{L}_s, \boldsymbol{L}_d$, and $\boldsymbol{L}$, respectively. The SSMA then minimizes the joint cost function by finding the smallest non-zero eigenvalues of the following generalized eigenvalue problem

$$\boldsymbol{Z}(\boldsymbol{L} + \mu \boldsymbol{L}_s)\boldsymbol{Z}^T\boldsymbol{V} = \lambda \boldsymbol{Z}\boldsymbol{L}_d\boldsymbol{Z}^T\boldsymbol{V}, \tag{3.4}$$

where $\boldsymbol{Z}$ contains the individual data matrices in block-diagonal form and $\boldsymbol{V}$ contains in the columns the eigenvectors organized in rows per domain [119, 126].

The pseudo-inverse of the eigenvector of the target domain can be used to transfer any data set from the latent domain $\mathcal{C}$ to the target domain according to [130]. It was tested experimentally in [124].

The following section expands on SSMA to mitigate some of its caveats. The problem

of SSMA to align data sets with strong nonlinearities is mitigated by introducing kernel functions. SSMA only allows the use of the same kernel for all data sets. This limits the use for the alignment of heterogeneous data.

3.3.5 Kernel Manifold Alignment

The KEMA approach is the extension of the SSMA [125] and uses the kernel trick (see Section 3.3.1) to align multimodal data sets [119]. One benefit of KEMA over SSMA is that strong nonlinear deformations can now be accurately modeled.
The kernelization of SSMA to generate the KEMA algorithm is done by exploiting the theorem about the direct sum of Hilbert spaces [102].

Theorem 3.3.1 (Direct sum of Hilbert spaces:). Given two Hilbert spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, the set of pairs $\{\boldsymbol{x}, \boldsymbol{y}\}$ with $\boldsymbol{x} \in \mathcal{H}_1$ and $\boldsymbol{y} \in \mathcal{H}_2$ is a Hilbert space $\mathcal{H}$ with the inner product

$$\langle \{x_1, y_1\}, \{x_2, y_2\} \rangle = \langle x_1, x_2 \rangle_{\mathcal{H}_1} + \langle y_1, y_2 \rangle_{\mathcal{H}_2} \tag{3.5}$$

This is called the direct sum of the spaces and is denoted as $\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2$. This property extends to a finite summation of Hilbert spaces.

This allows the mapping of all data sets to possibly different Hilbert spaces using, again, possibly different feature maps φ_i. This leads to the generalized eigenproblem

$$\boldsymbol{\Phi} \left(\boldsymbol{L} + \mu \boldsymbol{L}_s \right) \boldsymbol{\Phi}^T \boldsymbol{U} = \lambda \boldsymbol{\Phi} \boldsymbol{L}_d \boldsymbol{\Phi}^T \boldsymbol{U}, \tag{3.6}$$

where $\boldsymbol{\Phi}$ is the block diagonal matrix containing the data matrices and $\boldsymbol{U}$ contains the eigenvectors organized in rows for the particular Hilbert space. Since the eigenvectors can be of infinite dimension the Riesz representation theorem [105] is used to express the eigenvectors in the form of linear combinations of mapped samples. This leads to $\boldsymbol{U} = \boldsymbol{\Phi} \boldsymbol{\Lambda}$.

By multiplying both sides of Equation (3.6) by $\boldsymbol{\Phi}^T$ and replacing the dot products with the corresponding kernel matrices $\boldsymbol{K}_i = \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i$, the final problem becomes

$$\boldsymbol{K} \left(\boldsymbol{L} + \mu \boldsymbol{L}_s \right) \boldsymbol{K} \boldsymbol{\Lambda} = \lambda \boldsymbol{K} \boldsymbol{L}_d \boldsymbol{K} \boldsymbol{\Lambda}, \tag{3.7}$$

where $\boldsymbol{K}$ is the block diagonal matrix containing the individual kernel matrices $\boldsymbol{K}_i$.
Generally, KEMA has many desirable properties. It can align data sets with different dimensionalities by using a dedicated kernel for each domain, it performs well for manifolds of a diverse structure while maintaining the individual topology, and it is robust to strong nonlinearities and errors in graph estimation.
The corresponding publication (see [119]) also discusses the inversion of KEMA from the latent domain $\mathcal{C}$ to a target domain. This is generally desirable to measure the accuracy of the alignment in physical units. The exact pre-image does usually not exist when working with kernel functions (see Section 3.4). This requires the use of approximate methods that have ways to deal with discontinuities in the form of regularization [69, 132].
KEMA proposes to use a linear kernel for the inverse mapping. However, the inversion procedure is not part of the published Matlab code and implementation to compare KEMA to the proposed method of this thesis (see Section 6.5.4) did not produce results useful results, i.e., its use for mitigation of nonlinearities, physical interpretability, and data preprocessing could not be verified.[1].
However, alignment in a latent domain with KEMA and FT could successfully be tested. Learning an SVM model on the transformed data from one domain and applying it to the aligned data of another domain yielded good classification results.

3.4 The Pre-Image Problem

The main reason to give up kernel methods for the alignment of multiple data sets, is the pre-image problem.

[1]The Matlab code can be found on `https://github.com/dtuia/KEMA`

The pre-image problem is defined as follows [132]:
Given a sample $\widetilde{\boldsymbol{x}}$ in the possibly infinite dimensional common domain $\mathcal{C}$, find a corresponding sample $\boldsymbol{x}$ in the target domain $\mathcal{X}$ such that $\widetilde{\boldsymbol{x}} = \varphi(\boldsymbol{x})$.
When kernel functions are used for the alignment $\mathcal{C}$ is usually of higher dimensionality than $\mathcal{X}$. While each sample has an exact pre-image in its domain of origin, transforming one sample to another domain is difficult.
Let $\boldsymbol{x}^* \in \mathcal{X}^*$ be a sample in the target domain and $\boldsymbol{x} \in \mathcal{X}$ be a sample in another domain to be aligned. Their projections to the common domain $\mathcal{C}$ are denoted by $\widetilde{\boldsymbol{x}}^*$ and $\widetilde{\boldsymbol{x}}$, respectively. The inversion of $\widetilde{\boldsymbol{x}} \in \mathcal{C} \rightarrow \boldsymbol{x}' \in \mathcal{X}^*$ is generally ill-posed, as the relationship $\|\widetilde{\boldsymbol{x}}^* - \widetilde{\boldsymbol{x}}\|_2 \leq \delta \|\boldsymbol{x}^* - \boldsymbol{x}'\|_2$ for some $\delta \in \mathbb{R}$. This means that a small distance between samples in the common domain does not necessarily translate to a small distance in the target domain after the inversion.
An example would be the inversion close to a singularity of the kernel function. The inverse solution is unbounded resulting in a projection far from the expected/desired solution.
Approaches to solving these problems are usually tied to specific assumptions about the data structure and the domains. When the pre-image does not exist it is possible to calculate approximate solutions. A particular approach treats the inversion as a nonlinear optimization problem and proposes a fixed-point iteration scheme to calculate a solution [113]. However, this approach suffers from several issues. The nonlinearity of the optimization problem can cause numerical instability, and the iterative solver can get stuck in local minima, requiring multiple restarts with different initialization. The solution is also calculated individually for each sample, which makes the whole process very time-consuming. Finally, simply using a distance measure in the common domain might not be the best parameter to optimize, when the goal is to minimize the classification error. This approach also only works for a specific set of kernel functions, restricting the methods to particular, e.g., Gaussian, kernels. Other approaches exploit a distance constraint to find the location of the pre-image [69] or learn the inverse feature map from training samples [132].
Overall, the inversion of MA techniques based on kernel functions is a complex topic. Lit-

erature research and the experiments in this thesis do not hint at a solution that is both accessible and produces adequate results at the same time. Thus, the proposed method in this thesis refrains from using kernel functions and instead focuses on a same-space data representation for the alignment process without the requirement for higher dimensional domains.

3.5 Summary

The recent demand in high resolution, large-scale multitemporal and multimodal remote sensing data justifies the high interest in ML [18, 17, 117, 4] and DA [130, 119, 52] techniques. They strive to combine all available data for analysis. ML techniques, such as ISOMAP [117] or LLE [2] are essentially nonlinear dimensionality reduction methods to simplify the description of high-dimensional data sets. ML procedures are usually unsupervised approaches to describe the latent data structure. The goal of MA is then to define projections between data sets with similar underlying manifold structures [56, 119, 143]. These procedures usually require training data in both domains, but also unsupervised methods exist [129].
However, the ML and alignment methods have their caveats. ISOMAP and LLE produce a locally optimal mapping. However, since the feature space has a much lower dimension multiple sources, e.g., different classes, can become mixed. Without further constraints, interesting structures in the data may be lost. this can be circumvented by using kernel methods and transforming the data to a potentially infinite dimensional feature space [81]. However, results obtained in such a feature space do not permit interpretation in meaningful physical units, they can only be exploited for decisional problems like classification [119]. If a physical interpretation of the aligned data is the goal, the data needs to be inverted back to the source domain. The main problem is that methods based on kernel machines have to deal with the pre-image problem [145]. The pre-image problem deals with finding corresponding patterns in the original domain for points in the potentially infinite dimensional feature space. Existing methods require specific kernel methods together with a particular kernel

function to compute an approximate solution [69, 132]. These are often problem-specific, depend on fine-tuning of hyperparameters and are computationally challenging [119].

MA is generally defined as an approach to match corresponding data sets while simultaneously preserving the topology of each data set. DA, while encompassing the MA techniques also includes methods that allow data transformation with the goal to transfer useful features to a different but related problem. This also includes the correction of individual effects from Section 2.3. On one side, the preservation of the original topology is not always a necessary constraint. On the other side, individual correction of undesired effects can be time-consuming, and interference between subsequent preprocessing steps is a common problem.

This thesis proposes an approach without the requirement for kernel methods or physical models. The idea is to find a simplified, physically meaningful data representation that does not have the topology constraint, but still allows data alignment and FT from one data set to another. Following, Chapter 4 introduces the NFN as a method for simplified data representation and Chapter 5 expands on this idea to align multiple data sets in a chosen reference domain.

Chapter 4

Nonlinear Feature Normalization

This chapter, introduces the NFN. This algorithm aims at *normalizing* the spectral signatures by reducing unwanted nonlinear effects and aligning the results to a pre-defined basis for easy interpretation. The normalization is done by individual translations for each spectrum based on training data for each class in the scene [51].
This chapter is divided into four parts.
In Section 4.1 the general idea and the assumptions for training data selection, as well as the necessary notation for a proper mathematical description, are introduced.
Section 4.2 describes the NFN algorithm.
Section 4.3 gives a numerical example to demonstrate the geometric process during transformation. Additionally, the transformation is visualized for a problem with three bands.
At last, Section 4.4 discusses the selection of basis vectors, the penalty function, the preservation of spectral features, and the computational complexity of NFN.

4.1 Assumptions and Notation

Let $X \in \mathbb{R}^{d\times n}$ be a data set with n samples and d bands. Each sample $\boldsymbol{x}_i, i = 1\ldots, n$ belongs to one of p classes. Let $L_j \in \mathbb{R}^{d\times l_j}$ be the training data for class $j = 1, \ldots, p$, with $L_{j_1} \cap L_{j_2} = \emptyset$ for $j_1 \neq j_2$ and $L = \bigcup_{j=1}^{p} L_j$ the complete set of training data. Furthermore,

let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_p \in \mathbb{R}^d$ be the class reference spectra defining the new basis, represented by the matrix $B \in \mathbb{R}^{d \times p}$. The manifolds containing all possible variations of samples per class (e.g., due to external effects) are the open subspaces $\mathcal{M}_j \subseteq \mathbb{R}^d$.

Training data and reference spectra can be part of the original data set X or come from external sources, e.g., spectral libraries or a supervised selection process. The reference spectra in B can theoretically be arbitrary vectors, e.g., Euclidean unit vectors or material spectra from a spectral library. The training data need to represent the inherent data structure to model the exact conditions of the data set. Overall, the following assumptions about data structure and training data are used [128]:

1. **Assumption 1: Class uniqueness** For classes j_1, j_2 with $j_1 \neq j_2 \Rightarrow \mathcal{M}_{j_1} \cap \mathcal{M}_{j_2} = \emptyset$. This means that two identical samples belong to the same class, which also implies that even for data containing noise doesn't lead to an overlap of different class manifolds. This assumption also means that training data are purely chosen to distinguish between iconic features, not semantic properties, i.e., discriminating between a hospital, and the town hall with the same roof material is not possible.

2. **Assumption 2: Training data selection** Training samples represent the underlying shape of class manifolds $\mathcal{M}_j$. This means all possible variations of each class should be included during training data selection, i.e., if samples of a class lie in shadow, some need to be included in the training data to reduce the risk of projecting them to another manifold.

3. **Assumption 3: Closed world** Class labels represent all occurring classes of interest in the data set X. This means the data set contains no additional or hidden classes for which nonlinear effects should be reduced.

These assumptions are commonly made for any classifier that tries to separate multiple classes based on their characteristic features.

Proving that Assumption 1 is fulfilled is generally not possible, unless all the class labels are already known in advance. Violations can occur between very similar classes, e.g., different

vegetation species, and in dark areas where the signal is low compared to the noise level. Contrary to the example of short-circuiting of ISOMAP in Section 3.2.4, where a violation of this assumption results in a global change of manifold representation, NFN transformation becomes only erroneous in the immediate area of the violation.

Casually speaking, Assumption 1 states that robust discrimination between different classes is only possible if the classes have mutually distinct features. Geometrically, each of the p different class manifolds $\mathcal{M}_1, \ldots, \mathcal{M}_p \subseteq \mathbb{R}^p$ occupies a unique space in $\mathbb{R}^p$, and for two different manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, $\mathcal{M}_i \cap \mathcal{M}_j = \emptyset$ for $i \neq j$ holds. If there exist two samples $\boldsymbol{x}_1, \boldsymbol{x}_2$ with $\boldsymbol{x}_1 = \boldsymbol{x}_2$ and $\boldsymbol{x}_1 \in \mathcal{M}_i$, then it automatically follows that $\boldsymbol{x}_2 \in \mathcal{M}_i$.

Assumption 2 states that a representative sampling of each class manifold structure is necessary to perform NFN accurately. Contrary to unsupervised or semi-supervised approaches NFN uses only labeled samples to learn the data structure and exploit it to refine the training data. This assumption can be used as a guide for human analysts to select training data in a new data set.

Assumption 3 states that all occurring classes of a data set are known a priori. This assumption is required only for classes that are required for further analysis. Additional classes that are not included in the labels will be drawn closer to other classes during the subsequent transformation and separation from the other classes becomes more difficult.

This concludes the necessary assumptions and notation to describe the NFN transformation.

4.2 NFN Algorithm

The NFN algorithm computes a translation vector for each spectrum $\boldsymbol{x}_i$ in the data set. To simplify the notation the index i is dropped. NFN transformation can then be computed as follows.

Starting with a sample $\boldsymbol{x} \in \mathbb{R}^d$ of the data set X, the first step is to calculate the NN distance towards the training data per class as

$$\delta_j = \min_{\boldsymbol{x}_t \in L_j} \|\boldsymbol{x} - \boldsymbol{x}_t\|_2 \,. \tag{4.1}$$

In this thesis the Euclidean norm denoted by $\|\cdot\|_2$ is used as the standard metric to measure distances. The use of other metrics is permissible, but additional test are necessary to balance the other parameters of the NFN transformation.

Additionally, the distance vector $\widetilde{\boldsymbol{d}}_j \in \mathbb{R}^n$ between the sample $\boldsymbol{x}$ and the class references $\boldsymbol{b}_j$ is calculated according to

$$\widetilde{\boldsymbol{d}}_j = \boldsymbol{b}_j - \boldsymbol{x}. \tag{4.2}$$

At this point p NN distances and p vectors identifying the sample $\boldsymbol{x}$ with the references $\boldsymbol{b}_j$ were calculated.

The final step is a vector addition of $\boldsymbol{x}$ with a convex combination of $\widetilde{\boldsymbol{d}}_j$, weighted by coefficients $\widetilde{c}_j$.

To calculate the coefficients, the NN distances δ_j from (4.1) are modified using a penalty function $g : \mathbb{R}^p \to \mathbb{R}^p$. The goal of g is to favor translation in directions where the distance towards training data is small and simultaneously prevent translation when the distance to the nearest training samples is big. Further discussion on penalty functions can be found in 4.4.1. In this thesis, the nonlinear penalty function

$$g(\boldsymbol{\delta}) = \boldsymbol{\delta}^{-t}, \text{with } t \in \mathbb{N} \tag{4.3}$$

is used. $\boldsymbol{\delta}^{-t}$ stands for element by element operation of δ_j^{-t} for all j.

The coefficients $\widetilde{\boldsymbol{c}}$ for the convex combination of $\widetilde{\boldsymbol{d}}_j$ are given by

$$\widetilde{c}_j = g(\delta_j) / \sum_{j=1}^{p} g(\delta_j). \tag{4.4}$$

The final translation is then calculated by

$$\widetilde{\boldsymbol{x}} = \boldsymbol{x} + \sum_{l=1}^{p} \widetilde{c}_j \widetilde{\boldsymbol{d}}_j \tag{4.5}$$

Algorithm 1 NFN algorithm

procedure NFN($\boldsymbol{x}, L, B, g$) ▷ Normalize $\boldsymbol{x}$ with respect to B
  for $j = 1, \ldots, p$ **do**
    $\delta_j \leftarrow \min_{\boldsymbol{x}_t \in L_j} \|\boldsymbol{x} - \boldsymbol{x}_t\|_2$ ▷ NN distance
    $\widetilde{\boldsymbol{d}}_j \leftarrow \boldsymbol{b}_j - \boldsymbol{x}$ ▷ Translation vector to reference
  end for
  if $\delta_j = 0$ for one j **then**
    $\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{b}_j$ ▷ Prevent division by 0 if $\boldsymbol{x} \in L$
  else
    for $j = 1, \ldots, p$ **do**
      $c_j \leftarrow g(\delta_j)$ ▷ Penalty function on NN distance
    end for
    $\widetilde{c}_j \leftarrow c_j / \sum_{l=1}^{p} c_l$ ▷ Normalization for convex combination
    $\widetilde{\boldsymbol{x}} \leftarrow \boldsymbol{x} + \sum_j \widetilde{c}_j \widetilde{\boldsymbol{d}}_j$
  end if
Optional $\widetilde{\boldsymbol{x}} \leftarrow \widetilde{\boldsymbol{x}} \cdot \|\boldsymbol{x}\|_2 / \|\widetilde{\boldsymbol{x}}\|_2$ ▷ Re-normalization to improve visualization
Output $\widetilde{\boldsymbol{x}}, \widetilde{c}_j, \widetilde{\boldsymbol{d}}_j$
end procedure

If $\boldsymbol{x}$ is among the training spectra, the distance $\delta_j = 0$ for one $j = 1, \ldots, p$ and consequently $g(\delta_j) \to \infty$ for this j. In this case, $\widetilde{\boldsymbol{c}} = \boldsymbol{e}_j$, where $\boldsymbol{e}_j$ are the canonical unit vectors of $\mathbb{R}^p$. Equation (4.5) then simplifies to $\widetilde{\boldsymbol{x}} = \boldsymbol{b}_j$, if $\boldsymbol{x} \in L_j$. This means every sample $\boldsymbol{x} \in X$ identical to one of the training spectra is automatically identified with the reference spectrum $\boldsymbol{b}_j$ of the corresponding class j.

An optional step is to scale the transformed samples to their original norm by

$$\widetilde{\boldsymbol{x}}_N = \widetilde{\boldsymbol{x}} \cdot \|\boldsymbol{x}\|_2 / \|\widetilde{\boldsymbol{x}}\|_2 . \tag{4.6}$$

This step preserves distinguishable illumination differences in the transformed data and improves visualization. However, it is not necessary for mitigating nonlinear effects.

Pseudo-code for the NFN algorithm is given in Algorithm 1. It is often beneficial to change the distance calculation from Equation (4.1) to a k-NN (k-NN) search by

$$\delta_j = \frac{1}{k} min_{t_1,\ldots,t_k} \sum_{i=1}^{k} \|\boldsymbol{x} - \boldsymbol{x}_{t_i}\|_2 . \tag{4.7}$$

This reduces the effect of outliers in training data by taking the mean distance between $\boldsymbol{x}$ and the k nearest elements. This helps when lots of training data per class are available, but the labels are not completely reliable. This way, the impact of single mislabeled training spectra becomes less severe. The effect is analyzed in Section 6.3.3.
The effects of different parameters t for the penalty function and k for the number of NNs are analyzed in Section 6.3.2.

4.3 Simple Examples

4.3.1 Individual Sample in two bands

This section shows the step-by-step progression of the NFN transformation performed on a single sample $\boldsymbol{x}$ with only two classes. Simple numeric values are used to illustrate the effects of each step on the intermediate result.
The basis $B = \left\{ \begin{pmatrix} 5 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 5 \end{pmatrix} \right\}$ is chosen for the NFN algorithm and the transformation for the sample $\boldsymbol{x} = \begin{pmatrix} 2 \\ 2 \end{pmatrix}$ is calculated. Assuming a set of training data $L_1 = \left\{ \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\}, L_2 = \left\{ \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right\}$, the distances δ_j from $\boldsymbol{x}$ to the closest point in L_j are calculated as

$$\delta_1 = min_{x_k \in L_1} \|\boldsymbol{x} - \boldsymbol{x}_k\|_2 = \left\| \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 3 \\ 2 \end{pmatrix} \right\|_2 = 1, \tag{4.8}$$

and

$$\delta_2 = min_{x_k \in L_2} \|\boldsymbol{x} - \boldsymbol{x}_k\|_2 = \left\| \begin{pmatrix} 2 \\ 2 \end{pmatrix} - \begin{pmatrix} 2 \\ 4 \end{pmatrix} \right\|_2 = 2. \tag{4.9}$$

The corresponding translation vectors for $\boldsymbol{x}$ towards the basis B are

$$\widetilde{\boldsymbol{d}}_1 = \boldsymbol{b}_1 - \boldsymbol{x} = \begin{pmatrix} 5 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 3 \\ -2 \end{pmatrix}, \tag{4.10}$$

and

$$\widetilde{\boldsymbol{d}}_2 = \boldsymbol{b}_2 - \boldsymbol{x} = \begin{pmatrix} 0 \\ 5 \end{pmatrix} - \begin{pmatrix} 2 \\ 2 \end{pmatrix} = \begin{pmatrix} -2 \\ 3 \end{pmatrix}. \tag{4.11}$$

The penalty function is calculated element-wise as

$$g(\boldsymbol{\delta}) = \boldsymbol{\delta}^{-t}. \tag{4.12}$$

For $t = 2$, this results in

$$g(\boldsymbol{\delta}) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}^{-2} = \begin{pmatrix} 1 \\ \frac{1}{4} \end{pmatrix} \tag{4.13}$$

Normalization of the coefficients to generate a convex combination is done by

$$\widetilde{c}_j = g(\delta_j) / \sum_{l=1}^{2} g(\delta_j) \tag{4.14}$$

leads to

$$\widetilde{c}_1 = 1 / \frac{5}{4} = \frac{4}{5} \tag{4.15}$$

and

$$\widetilde{c}_2 = \frac{1}{4} / \frac{5}{4} = \frac{1}{5}, \tag{4.16}$$

respectively. This results in the transformed sample

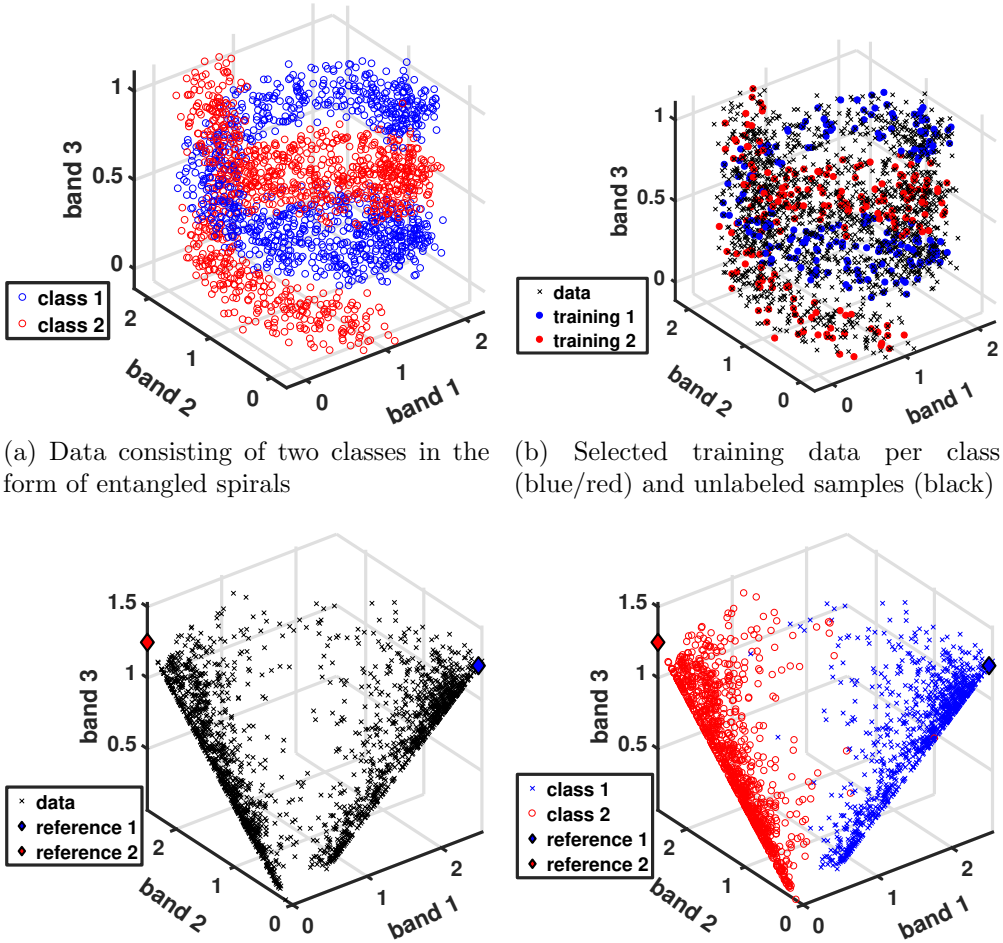
$$\begin{aligned}
\widetilde{\boldsymbol{x}} &= \boldsymbol{x} + \sum_{j=1}^{2} \widetilde{c}_j \widetilde{d}_j \\
&= \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \left(\frac{4}{5} \cdot \begin{pmatrix} 3 \\ -2 \end{pmatrix} + \frac{1}{5} \begin{pmatrix} -2 \\ 3 \end{pmatrix} \right) \\
&= \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 12/5 \\ -8/5 \end{pmatrix} + \begin{pmatrix} -2/5 \\ 3/5 \end{pmatrix} \\
&= \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 2 \\ -1 \end{pmatrix} \\
&= \begin{pmatrix} 4 \\ 1 \end{pmatrix}.
\end{aligned} \tag{4.17}$$

The resulting sample is considerably closer to the class 1 reference due to $\delta_1 < \delta_2$. Re-normalizing $\widetilde{\boldsymbol{x}}$ to the norm of $\boldsymbol{x}$ yields

$$\begin{aligned}
\widetilde{\boldsymbol{x}} &= \widetilde{\boldsymbol{x}} \cdot \|\boldsymbol{x}\|_2 \,/\, \|\widetilde{\boldsymbol{x}}\|_2 \\
&= \begin{pmatrix} 4 \\ 1 \end{pmatrix} \cdot \left\| \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\|_2 \Big/ \left\| \begin{pmatrix} 4 \\ 1 \end{pmatrix} \right\|_2 \\
&= \begin{pmatrix} 4 \\ 1 \end{pmatrix} \cdot \left(\sqrt{8}/\sqrt{17} \right) \\
&\approx \begin{pmatrix} 2.74 \\ 0.69 \end{pmatrix}.
\end{aligned} \tag{4.18}$$

4.3.2 Simulated Data in three bands

To illustrate NFN computation Fig. 4.1 shows a step-by-step transformation performed on a 3D toy example. The data consists of two different classes in the form of entangled spirals. NFN transforms the data by aligning it to a new two-dimensional basis, each spiral



(a) Data consisting of two classes in the form of entangled spirals

(b) Selected training data per class (blue/red) and unlabeled samples (black)

(c) Data after NFN transformation towards class references, depicted as blue and red diamonds

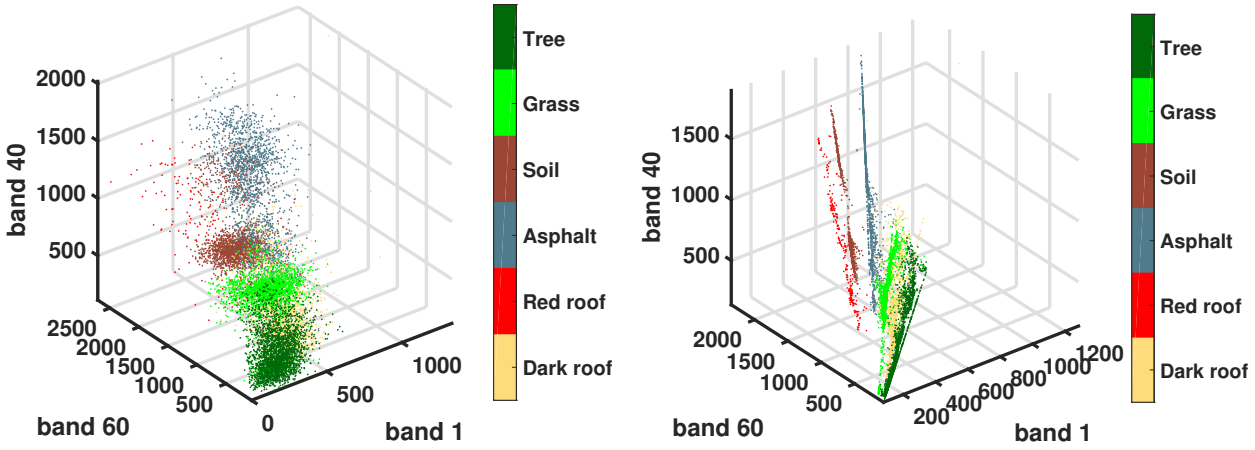
(d) NFN transformed data with colored samples according to the original class labels shows improved linear separability.

Figure 4.1: Progression of NFN transformation from entangled spirals to linearly separable data in the same space.

corresponding to one of the basis vectors. Fig. 4.1a shows the intended sample labeling for the red and blue source — a set of training data, depicted in Fig. 4.1b. The reference spectra, given by the red and blue diamond in Figure 4.1c, are used to define a linear basis, and all training spectra are identified with their respective reference. The unlabeled samples individually gravitate towards the nearest training samples per source and are translated to the new basis according to Equation (4.5). The resulting linearization is depicted in Fig. 4.1c. Finally, the linearized data with original class colors are shown in Fig. 4.1d. Most data points are drawn close to the new basis vectors; those in-between are points with similar shortest distance to the training data of both classes in the original data set. The inherent data structure after NFN transformation shows a linearized structure and can easily be classified using a linear decision plane. The nonlinear effects were successfully mitigated by the NFN algorithm.

4.3.3 Real Hyperspectral Data

An example of the performance of NFN on real hyperspectral data is shown in Fig. 4.2. The images show scatter plots for three bands of a real hyperspectral data set. Each point in the image corresponds to a spectral signature. The samples are color-coded according to the corresponding class labels. Before the NFN transformation, the different classes are clustered close together and finding separation planes appears difficult. After the NFN transformation, the individual class structure is linearized according to the reference spectrum for each class. For example, the gray and red class appeared to overlap in the original data but can be easily separated after the transformation. This is also true for the other classes. The yellow, green and dark green class seem to still overlap in the image, however, changing the visualization by selecting a different band combination shows that linear separation is also possible here.



(a) Hyperspectral data with colored class labels.

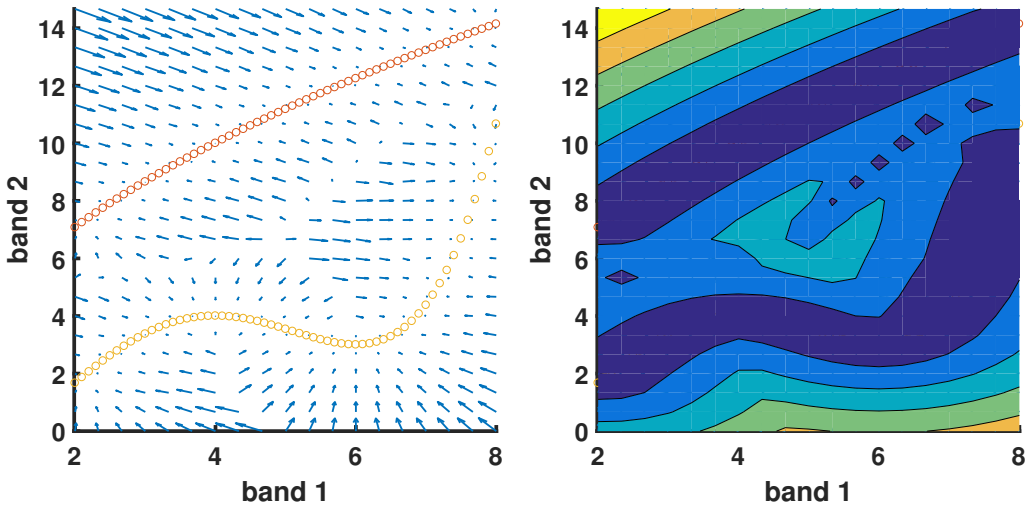
(b) Data after NFN-transformation

Figure 4.2: Applying NFN to real hyperspectral data results in a linearized data structure. Depicted are scatter plots for the original data and the NFN-transformed data. The samples are color-coded according to their class labels.

4.4 Discussion and Properties of NFN

4.4.1 Penalty Function

The NFN algorithm aims at transforming individual samples towards a more linear data representation, concerning the matrix of class reference spectra B. The NFN transformation draws samples of the same class closer to the corresponding reference, while samples of different classes are simultaneously pulled apart. This can be done by weighting the translation depending on the distance between the test and training samples. At this time only uniform weighting for all classes and bands is used. Figure 4.3(a) shows the NFN transformation for uniformly distributed training samples and two classes in the form of a vector field. The arrows indicate the length and direction of the translation. Since the coefficient vectors $\boldsymbol{c}$ are normalized to 1, the length of the translation is determined by the distance to the nearest training samples. There are areas in the plot where the distance to both classes is almost equal. This results in almost equal coefficients for each class. In the case of this 2D example, the translations towards each class cancel each other out. When one distance is larger than the other, the penalty function shifts the coefficients in favor of the closest class. If the



(a) Vector field to visualize direction of attraction on uniformly distributed samples between two classes.

(b) Contour plot showing magnitude of attraction for the same example.

Figure 4.3: Progression of NFN transformation from entangled spirals to linearly separable data in the same space.

distance to that class is still relatively large, a great translation is beneficial to bring the sample in line with the general structure. If the distance is already relatively small, a short translation to prevent identifying the sample with the class reference is performed. If this was permissible, the risk of losing important features during the NFN transformation increases. Figure 4.3(b) shows the same experiment using contour lines to visualize the magnitude of translation concerning the distance from the training samples. The large translation on the outside of the two sets of training samples is due to collinearity of the classes. E.g., for a sample in the upper left corner of the graph, the NNs per class are roughly in the same direction. This means both individual translations are performed in the same direction. The penalty function together with the normalization of coefficients has to make sure that a sample is not able to 'overshoot' the associated nearest training samples. This is an important property as it could potentially invert characteristic features. In hyperspectral data, the number of classes p is usually much smaller than the dimensionality d, and collinearity is not a problem.

The penalty function should be positive and monotonically decreasing with increasing dis-

tance from the associated training samples per class considering the requirements above. The coefficient c is then calculated according to Equation (4.4) by normalizing $g(\delta_j)$ for all δ_j of one sample. This leads to large coefficients when one δ_j is significantly larger than the rest.

In this thesis the penalty function

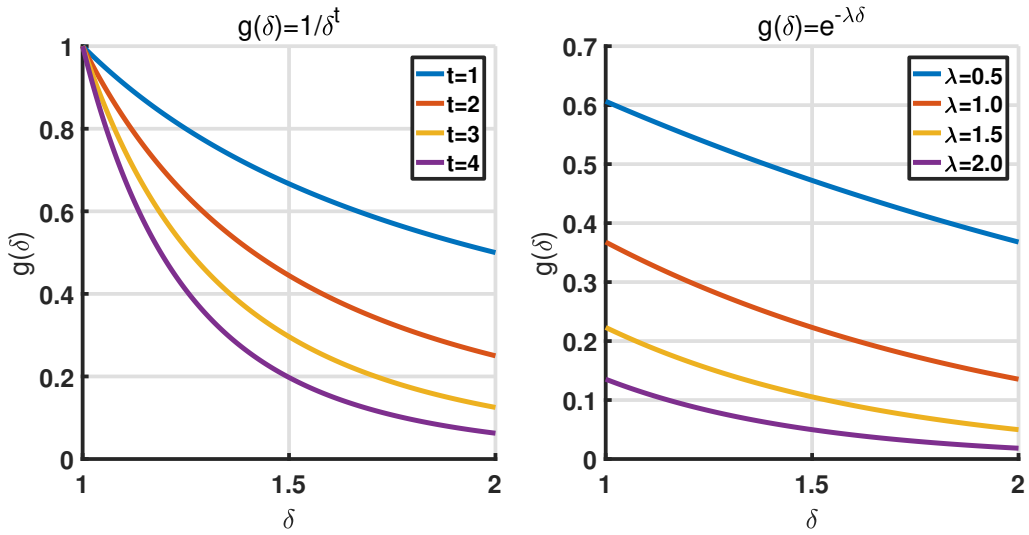
$$g(\boldsymbol{\delta}) = \delta^{-t}, \tag{4.19}$$

with $t \in \mathbb{N}$ is used. This function was specifically chosen to make it impossible for a sample to 'overshoot' the training samples of the nearest class for $t > 1$, even in the case of collinearity. Alternatively, using the penalty function

$$g(\boldsymbol{\delta}) = e^{-\lambda\boldsymbol{\delta}}, \tag{4.20}$$

with $\lambda \in \mathbb{R}_+$, produces similar results. A comparison of the two functions with different parameter values is depicted in Figure 4.4. In practice, it was easier to fine-tune (4.19). Alternatives based on sigmoid functions are also possible, but have not been tested extensively. For this thesis, each class is considered to be equally important. It is theoretically possible to apply individual penalty functions based on class membership of the training data. However, careful parameter selection, prior knowledge of class distribution, etc. are required to produce optimal results. The risk of miscalculation increases with the number of classes in the scene.

The topic of weighting can even be extended to the level of individual spectral bands. A possible reason can be that hyperspectral data have different statistical properties per band. Using Euclidean distance on $\mathbb{R}^d$ for the calculation of k-NN does not take individual variances per band into account. This topic is discussed further in Section 7.2 as it can be vital to distinguish between wanted and unwanted features. However, only the general concept is discussed as the number of possible approaches exceeds the scope of this thesis.



(a) Plots for the penalty function (4.19) with different t.

(b) Plots for the penalty function (4.20) with different λ.

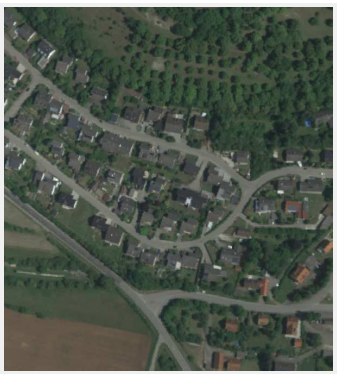
Figure 4.4: Comparison of two possible penalty functions and their progression using different parameters.

4.4.2 Basis Selection

In this section, the selection of reference vectors per class to build the basis B is discussed. Several theoretical examples are covered, and the most practical approach when NFN is used for hyperspectral data is highlighted.

Every class must be represented by a single reference vector for NFN transformation. They can be anything with the same dimension as the data set X, so $b_j \in \mathbb{R}^d$ for $j = 1, \ldots, p$.

When thinking about a new basis to represent the data, the first thing that comes to mind is an orthonormal basis describing the standard basis with canonical unit vectors. As only p vectors are required, assuming without loss of generality that the basis B consists of the unit vectors $e_1, \ldots, e_p \in \mathbb{R}^d$ is viable. This representation has the advantage that the distance between classes is normalized. It has similarities with a nonlinear PCA in the form that the most information is encoded in the first p components. This means it can also be used for dimensionality reduction. The difference is that the classes are not sorted with regard to their variance, but the first p components of a nonlinear PCA (e.g., kernel-PCA [31])



(a) RGB color composite of the Greding1_ref data set (see Section 6.1).

(b) Band 1 of NFN transformation for the class 'Red roof'.

(c) Band 2 of NFN transformation for the class 'Asphalt'.

(d) Band 3 of NFN transformation for the class 'Soil'.

(e) Band 4 of NFN transformation for the class 'Grass'.

(f) Band 5 of NFN transformation for the class 'Tree'.

Figure 4.5: The first six bands after NFN transformation of a real hyperspectral data set using the standard basis $e_1, \ldots, e_6 \in \mathbb{R}^d$.

would encode roughly the same amount of variance. However, this comes with the same disadvantages as PCA. After the transformation, a physical interpretation of the samples is no longer possible as the representation is shifted and rotated favoring an easy description. Also, since the distance between all classes is now equal, there is no longer a context in the form of similarity between classes. An RGB color composite of a hyperspectral data set and the the first five bands after NFN transformation with the standard basis $e_1, \ldots, e_6 \in \mathbb{R}^d$ are depicted in Figure 4.5. It is apparent that each band encodes the associated class. Consider the two different vegetation classes *Tree* and *Grass* and the class *Asphalt*. A classifier would usually be able to discriminate between vegetation and non-vegetation in the original

data. However, confusion between the two vegetation classes is a common occurrence due to spectral similarity. The distance to the *Asphalt* class would be identical to the distance between the two vegetation classes when using the standard basis for this example. During classification, this can result in a higher amount of vegetation samples classified as *Asphalt*. Whether this is a problem depends on the desired output. For generating landcover maps, it makes more sense to accept mistakes during classification of similar classes. If a single sample in a meadow is classified as *Tree*, for example, it is reasonable to assume that this error is simply due to similar spectral characteristics of the vegetation classes. If instead that sample is classified as *Asphalt*, it looks suspicious.

The goal of the NFN transformation is to identify all instances (including the nonlinear variations) of a class with only one signature. This is similar to spectral libraries, e.g., the spectral library of the *United States Geological Survey* (USGS) [65], where one signature per material is deposited. To also maintain the physical interpretability after NFN transformation it is reasonable to compose the basis B of appropriate material spectra. These can be drawn from spectral libraries, using measurements of laboratory or field spectrometers, or simply computed as the mean spectrum per class from the training data. Using spectra from a library or another instrument than the recording sensor usually requires interpolation to match the spectral dimension d. It is shown in the following chapter that the data set X and the basis B are not required to be from the same domain. This means using a basis composed of reflectance vectors to transform a radiance data set with NFN is possible as in both cases the relative distance between classes is similar.

For the experiments in this thesis, generally, the mean vector of all training samples per class is used to build B. Using an artificially constructed basis B can be beneficial when the data representation and a dimensionality reduction is desired (see Figure 4.5). However, it makes sense to use realistic spectra as reference vectors to preserve the physical interpretability. Also, most benchmark data sets have no assigned library or spectrometer measurements per class. This is mostly attributed to cost and time efficiency, as data acquisition can take place in rough terrain and over a large area.

4.4.3 Computational Complexity of NFN

Considering the run time of the NFN transformation, calculating the NN distance in Step 3 is the most time-consuming operation [35]. For a data set with d dimensions and training samples $l \ll n$, a k-NNs approach to calculating the translation vectors $\boldsymbol{d}_j$ can be computed in $\mathcal{O}(dlk)$ for one sample in our data set. This has to be performed for all n samples resulting in a complexity of $\mathcal{O}(ndlk)$. Since the computation of the translation is independent for each sample, this single for-loop can be parallelized without any changes to the procedure.
Currently, the NFN algorithm is implemented in MATLAB and the inherent k-NN function is used to perform the transformation. In the case of $d > 10$, which is almost guaranteed for hyperspectral data, the k-NN search of MATLAB defaults to an exhaustive search.
The practical runtime test on an office computer with Intel Core i7-6700, 32 GB RAM, and NVIDIA Quadro K1200 performed NFN on a hyperspectral data set with 670×606 pixel with 127 bands and a mean 2130 training samples for each of six classes in ~ 140 seconds.
It is possible to further reduce the computational complexity by using the median of medians approach [22] for the k-NN calculation. It operates by calculating an approximate median of the distances per sample, which can be used to reduce the problem size by a factor of 0.7. The approximate solution can deviate from the globally optimal solution, but the computational complexity reduces $\mathcal{O}(nl)$. Further improvement of the runtime of NFN, including optimization of the k-NN calculation, are possible with an implementation on GPU [74]. This approach is promising for real-time calculation of NFN.

Chapter 5

Nonlinear Feature Normalization for Data Alignment

In this chapter, the NFN algorithm is modified first to align multiple data sets to the same basis and then invert the transformation [51]. This enhancement effectively allows transformation of one data set to the domain of another. For this, additional requirements and notations are introduced in Section 5.1. In Section 5.2 multiple NFN transformations and their inverse translations are concatenated to construct the NFNalign algorithm. Section 5.3 presents two practical approaches to calculate pixel correspondences between the data sets, which are required for NFNalign to perform properly. A discussion of the approximate data alignment error and the computational complexity of NFNalign can be found in Section 5.4.

5.1 Requirements and Notation continued

For this section the same notation as in chapter 4 is used. Additionally, new terms to accurately describe the alignment of two or more data sets and the inversion to the original domain are supplemented.

Mathematically, a domain is any connected open subset of a finite-dimensional vector space. In this thesis, the term is mainly used to indicate the space occupied by the data including its

possible variations due to nonlinear effects. For the following operations, it is not necessary to further specify additional properties like the smoothness of the boundary, etc.

Let $X \in \mathbb{R}^{d \times n}$ be a data set in the domain $\mathcal{X} \subseteq \mathbb{R}^d$ with corresponding training data L, and $X^* \in \mathbb{R}^{d \times n^*}$ be a data set in the domain $\mathcal{X}^* \subseteq \mathbb{R}^d$ with training data L^*. The domains stand for feature spaces, e.g., different steps in the preprocessing chain or simply data from different sensors.

At this stage, the dimensionality d of the two data sets are assumed to be equal. An approach on how to handle different dimensionalities is discussed in Section 6.5.5. The number of samples n and n^* are not required to be equal for both data sets.

Let the common domain $\mathcal{C} \subseteq \mathbb{R}^d$ be the space both data sets are transformed to using the NFN transformation. In the following, $\mathcal{X}$ is referred to as the test domain and $\mathcal{X}^*$ as the reference domain. Reasons to choose a specific data set can be ideal illumination conditions, an optimal viewing angle, and good atmospheric correction.

One important point is the calculation of correspondence between individual samples of different domains. This problem can be practically solved by either spatial or spectral similarity and is discussed in Section 5.3, as it can be detached from the NFNalign algorithm. For the description of the NFNalign algorithm in the next section, the correspondence between the samples x and x^* is simply inferred.

5.2 NFNalign Algorithm

With the notation from Sections 4.1 and 5.1 it is now possible to describe the process of FT between domains based on the NFN transformation.

The first step is to perform NFN transformation on the reference sample $\boldsymbol{x}^*$ and the test sample $\boldsymbol{x}$ and align them with the same basis B. Further information about the selection of suitable basis vectors can be found in Section 4.4.2. Let $\widetilde{\boldsymbol{x}}^*, \widetilde{c}_j^*, \widetilde{\boldsymbol{d}}_j^*$ be the NFN output of the reference data X^* and $\widetilde{\boldsymbol{x}}, \widetilde{c}_j, \widetilde{\boldsymbol{d}}_j$ the NFN output of the test data X.

At this stage, it makes sense to perform a re-scaling of $\widetilde{\boldsymbol{x}}$ to eliminate unwanted effects due

to different illumination conditions between the data sets. The optimal re-scaling factor λ^* minimizes satisfies

$$\widetilde{\delta}^* = \min_{\lambda>0} \|\lambda\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}^*\|_2 \,. \tag{5.1}$$

This means the distance $\widetilde{\delta}^*$ between the reference sample $\widetilde{\boldsymbol{x}}^*$ and the re-scaled test sample $\lambda\widetilde{\boldsymbol{x}}$ is minimal, which in turn transfers to the final alignment error of the data alignment in the reference domain $\mathcal{X}$. This is further discussed in Section 5.4.1. To calculate the optimal re-scaling factor λ^* the angle α between $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{x}}^*$ can be used. α is calculated by

$$\alpha = \arccos\left(\frac{\widetilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{x}}^*}{\|\widetilde{\boldsymbol{x}}\|_2 \,, \|\widetilde{\boldsymbol{x}}^*\|_2}\right). \tag{5.2}$$

Together this results in

$$\lambda^* = \alpha\frac{\|\widetilde{\boldsymbol{x}}^*\|_2}{\|\widetilde{\boldsymbol{x}}\|_2}. \tag{5.3}$$

With

$$\widetilde{\boldsymbol{x}}_{temp} = \lambda^*\widetilde{\boldsymbol{x}} \tag{5.4}$$

the final alignment step is performed by applying the inverse translation from $\boldsymbol{x}^* \rightarrow \widetilde{\boldsymbol{x}}^*$ to the rescaled $\boldsymbol{x}_{temp}$ by

$$\widetilde{\boldsymbol{x}}_{align} = \widetilde{\boldsymbol{x}}_{temp} - \sum_j \widetilde{c}_j^*\widetilde{\boldsymbol{d}}_j^*, \tag{5.5}$$

where $\widetilde{c}_j^*$ are the coefficients and $\widetilde{\boldsymbol{d}}_j^*$ the translation vectors of NFN$(\boldsymbol{x}^*, L^*, B, g)$.

Overall, the sample $\boldsymbol{x} \in \mathcal{X}$ was first transformed to $\widetilde{\boldsymbol{x}} \in \mathcal{C}$ by the NFN transformation, re-scaled to have minimal distance to $\widetilde{\boldsymbol{x}}^*$, and subsequently transferred to $\mathcal{X}^*$ by applying the inverse of the original NFN transformation applied to $\boldsymbol{x}^*$.

The pseudo-code for NFNalign is given in Algorithm 2. Step 4 in Algorithm 2 is equivalent to the combination of Equations (5.2)-(5.4).

An alternative re-scaling approach is the calculation of $\widetilde{\boldsymbol{x}}_{temp}$ as

$$\widetilde{\boldsymbol{x}}_{temp} = \widetilde{\boldsymbol{x}}\frac{\|\widetilde{\boldsymbol{x}}^*\|_2}{\|\widetilde{\boldsymbol{x}}\|_2}. \tag{5.6}$$

Algorithm 2 NFNalign algorithm

1: **procedure** NFNALIGN($\boldsymbol{x}, \boldsymbol{x}^*, L, L^*, B$) ▷ Align $\boldsymbol{x}$ and $\boldsymbol{x}^*$
2: $\widetilde{\boldsymbol{x}}, \widetilde{c}_j, \widetilde{\boldsymbol{d}}_j \leftarrow NFN(\boldsymbol{x}, L, B)$
3: $\widetilde{\boldsymbol{x}}^*, \widetilde{c}_j^*, \widetilde{\boldsymbol{d}}_j^* \leftarrow NFN(\boldsymbol{x}^*, L^*, B)$
4: $\widetilde{\boldsymbol{x}}_{temp} \leftarrow \|\widetilde{\boldsymbol{x}}^*\|_2 \, cos(\alpha) \frac{\widetilde{\boldsymbol{x}}}{\|\widetilde{\boldsymbol{x}}\|_2}$ ▷ Rescaling, α is the angle between $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{x}}^*$
5: $\boldsymbol{x}_{align} \leftarrow \widetilde{\boldsymbol{x}}_{temp} - \sum_j \widetilde{c}_j^* \widetilde{\boldsymbol{d}}_j^*$ ▷ Perform inverse translation
6: **Output** $\boldsymbol{x}_{align}$
7: **end procedure**

Thus, $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{x}}^*$ have the same magnitude, which translates over to $\boldsymbol{x}_{align}$ and $\boldsymbol{x}^*$. However, this leads to big alignment errors $\widetilde{\delta}^*$ (and δ^* in the domain $\S^*$, respectively) when the pixel correspondences are not reliable, e.g. on the border between two materials. It is shown in Section 5.4.1 that Step 4 in Algorithm 2 produces the minimal alignment error.

An important fact is the existence of the inverse transformation. As discussed in Chapter 3 many FT methods have do deal with an ill-posed problem if they want to calculate an alignment in one of the original domains. Otherwise, physical interpretation of the results is generally not possible as the data sets are aligned in a high-dimensional space. There exist methods for regression or to check for the existence of the pre-image [69, 132]. However, most approaches are problem and data dependent and can not be generalized. Due to the constant dimensionality of domains and the performed vector calculus, this is not an issue with NFNalign. This directly translates to the property of error propagation, which is discussed in Section 5.4.1.

The NFNalign algorithm requires some form of correspondence between $\boldsymbol{x}$ and $\boldsymbol{x}^*$. A pixel-by-pixel mapping, e.g., concerning co-registration, would be ideal. However, perfect geo-referencing or co-registration cannot always be guaranteed. Also, the aspect angle of data acquisition can lead to samples in previously occluded areas for which no match exists. Section 5.3 is dedicated to give practical tools for establishing these correspondences.

5.3 Pixel Correspondence Calculation

The goal of NFNalign is to transform two or more data sets to a common domain $\mathcal{C}$ and invert the transformation of a chosen reference data set $X^* \subseteq \mathcal{X}^*$ to transform all other data sets from $\mathcal{C}$ to $\mathcal{X}^*$. Since each sample of every data set is transformed individually, the inverse transformation for each sample needs to be properly assigned individually. This procedure is done using pixel correspondences that establish a relationship for each sample of the test data to one or more samples of the reference data X^*. This requirement can be treated as an extra step in the data alignment process and can be identified with the process of minimizing the distances between point clouds via optimization.

In this section, two practical approaches to calculating pixel correspondences are discussed. The first approach relies on geographic information and can be used to align multiple data sets of the same area. The second approach is designed to also work on spatially disjoint data sets by exploiting the spectral similarities in the common domain $\mathcal{C}$. Even for data sets of the same region, this approach can be better, e.g., scenes recorded months apart showing substantial changes in construction or the phenological cycle.

5.3.1 Geographic Pixel Correspondence

Using the geographic information to construct the pixel correspondences is best applicable to data sets recorded throughout several days in the same location. Geographic correspondences exploit the fact that usually, large areas of a scene remain static throughout the day. Only cars and other moving objects contribute to changes, which constitute only a small fraction of the pixels. Ideally, the goal for geographic pixel correspondences is to use the pixel coordinates in each data set directly. In the following, this is referred to as pixel-by-pixel correspondence as corresponding samples have the same pixel location in both data sets.

Geographic information is generally recorded simultaneously to other sensor data for air-

and spaceborne imaging as it is required for georeferencing. For hyperspectral push-broom sensors, a commercial *Global Positioning System* (GPS) / *Inertial Navigation System* (INS) solution is installed next to the sensors. It consists of a GPS antenna, a data logging computer, and an *Inertial Measurement Unit* (IMU) to compensate for the roll, pitch, and yaw movement of the sensor during the flight. This additional hardware is required to attribute the correct GPS coordinates to every pixel. The accuracy of this so-called georeferencing process is important, especially when multiple data sets are evaluated together. Several factors can influence the accuracy.

1. GPS accuracy depends on the number of satellites, atmospheric conditions, and receiver design [85, 38]. It can be improved by using dual-frequency receivers [91], a second antenna or nearby GPS base stations for differential GPS measurements [88], and during post-processing [87].

2. It is important to accurately measure the alignment of the IMU regarding boresight parameters and lever arms [75] during data acquisition. These parameters can be optimized in post-processing. Boresight correction denotes the correction of angular misalignments and displacement between the GPS antenna, the IMU, and the principal points of each sensor. Especially the angular alignment is critical to the accuracy. Even misalignments ≥ 0.7 mrad can result in huge displacements on the ground as the error scales linearly with the altitude.

3. Also, the stability, drift, and noise of the IMU influence the accuracy.

In practice, when post-processing of GPS / INS data and boresight correction are optimized, it is possible to perform direct georeferencing with an absolute accuracy of approximately one pixel [75]. Small deviations from the optimal parameters can lead to big errors in georeferencing [53]. This effect is not immediately apparent when only analyzing a single image. With multiple images of the same region, these errors can be observed by superposition and comparison of linear structures like roads or buildings.
It is also important to note, that georeferencing uses the GPS / INS information together

with a sensor model to assign geographical coordinates to each pixel in an image. The image is projected onto the reference ellipsoid defined by the *World Geodetic System from 1984* (WGS 84). For regions with significant height differences, the use of a DEM is required to minimize projection errors.

One example of errors in the initial georeferencing is given in Figure 5.1(a) where a mosaic of RGB color composites from hyperspectral data is shown. The geographical alignment errors between the data sets are visible across tile borders.

If the georeferencing accuracy is insufficient for directly generating a pixel-by-pixel correspondence, it is possible to use image warping techniques to improve the alignment. The underlying problem can be treated as a mapping between two coordinate systems. Generally, a suitable geometric transformation is calculated to correctly modeling this mapping. Depending on the application and the alignment strategy, the commonly used approaches use affine [54, 36] or projective (homography, [39, 114]) transformations. In the next step, an objective function that measures the displacement concerning pixel- or feature-based techniques is constructed [96]. This objective function is then optimized for the optimal transformation parameters.

The practical approach that generated the best results for the experiments in this thesis was the optimization of an objective function based on the *Enhanced Correlation Coefficient* (ECC) maximization [29]. It is important to note, that this approach is invariant to contrast and brightness changes in the scene. This means ECC is robust to the effects NFNalign aims to correct. If this were not the case, the results of ECC would vary with the different conditions during data acquisition and ultimately interfere with the spectral alignment as well. Also, while the function for parameter optimization of the ECC algorithm is nonlinear, the iterative scheme to solve the optimization problem is linear and can be calculated quickly. [1]

Figure 5.1 shows the improvement in co-registration accuracy between direct georeferencing (using only the raw GPS / INS data) and georeferencing with post-processed GPS / INS, and boresight correction followed by ECC maximization.

[1]The MATLAB code for ECC maximization is available on `https://de.mathworks.com/matlabcentral/fileexchange/27253-ecc-image-alignment-algorithm--image-registration-`.

(a) Mosaic of georeferencing from raw GPS / INS data

(b) Mosaic of georeferencing after GPS / INS post-processing, borsight correction and ECC maximization

Figure 5.1: Mosaic with RGB composites of three different data sets from the Greding measurement campaign. Comparison between georeferencing without optimization of GPS / INS data and corrected data after ECC maximization warping. The success of warping can be seen by the perfect alignment of objects across tile borders.

The accuracy is sufficient to directly use the spatial pixel locations as correspondences for a pixel-by-pixel mapping. It has to be noted, however, that it is not possible to correct effects that result from different viewing angles like occlusions. One example is the visibility of building facades in one image due to an oblique viewing angle, while they are not visible in nadir view in the other images.

5.3.2 Pixel Correspondence by Spectral Similarity

Spatial correspondences may not always be possible. Either the time between data acquisition was so long that significant phenological or structural changes would negatively impact the result or the data sets were recorded in entirely different regions. However, it is important for NFNalign that all data sets share the same classes. Otherwise, it is not possible to create a common basis in $\mathcal{C}$ that both data sets can be aligned to with NFN.

During NFNalign both data sets are aligned to the same basis B in the common domain $\mathcal{C}$. At this point in the alignment process, the two data sets are already similar regarding spectral similarity, but the range may be different. Radiance data, for example, is normally saved as 14-bit integers, which results in values $\in [1, 16383]$ per band, while reflectance data $\in [0, 1]$ is usually multiplied by 10000 and converted to integer values to reduce storage space. Thus, Step 4 of Algorithm 2 is executed to adjust the individual sample norm according to the pixel correspondences.

The spectral alignment in $\mathcal{C}$ is exploited to calculate a suitable match between samples of different data sets when a pixel-by-pixel mapping is not possible, e.g., geographical coordinates are not available, or the depicted regions of both data sets do not overlap. The angle between two samples (see Equation (5.8)) of different data sets is a good indicator of spectral similarity. This measure is invariant to the magnitude of the individual samples.

To establish pixel correspondences for the whole data set $\widetilde{X} \subseteq \mathcal{C}$ the first step is to calculate the angle α between all individual samples of $\widetilde{X}$ and all samples in $\widetilde{X}^*$. Subsequently, the minimal angle α^* (i.e., the maximum of $cos(\alpha)$) between a sample $\widetilde{\boldsymbol{x}} \in \widetilde{X}$ and the samples in $\widetilde{X}^*$ determines the correspondence.

It can be shown that NFNalign with this correspondence produces the smallest alignment error. For this, let $\widetilde{\delta}^*$ be the distance between $\widetilde{\boldsymbol{x}}_{temp}$ and $\widetilde{\boldsymbol{x}}^*$ and α^* the corresponding smallest angle. Assuming there exists a sample $\widetilde{\boldsymbol{x}}^\dagger$ for which $\alpha^\dagger > \alpha$ while simultaneously $\widetilde{\delta}^\dagger < \widetilde{\delta}^*$ is a contradiction to the laws of sine, provided that $\alpha^* \in [0, \pi/2]$. The last statement is usually not problematic as the goal is to find the smallest angle and hyperspectral data are usually represented by positive values, resulting in all spectra being located in the first quadrant $X \in \mathcal{R}^d_+$.

However, there are some things to consider with this approach. One is the re-scaling in Step 4 from Algorithm 2. Since the SNR changes with illumination, it is possible that the maximal spectral similarity occurs between a $\widetilde{\boldsymbol{x}}$ with a small magnitude and $\widetilde{\boldsymbol{x}}^*$ with a high magnitude. The re-scaling then results in a large alignment error. This predominantly happens when data from different domains are aligned, e.g., alignment of radiance to reflectance data,

as they operate on different ranges. This is also the reason why other similarity metrics, e.g., based on Euclidean distance are not viable. Another problem is related to neighboring samples corresponding to pixels with different illumination resulting in an unnatural inhomogeneous brightness distribution after NFNalign. This can be rectified using spatial filters on the aligned data X_{align} to correct abrupt jumps in magnitude between similar signatures spatially close to each other.

Also, computing the spectral angle between all samples of $\widetilde{X}$ and $\widetilde{X}^*$ requires nn^* individual operations and can in practice be too time-consuming for average to large data sets. In this case, a reduction of the samples in $\widetilde{X}^*$ will speed up the process. Possible approaches are a systematic or random selection of a set percentage of samples. More complicated techniques based on clustering or segmentation can also reduce the samples to a smaller but still representative subset [61, 138].

5.4 Properties of NFNalign

This section shows the relationship between the sample differences in the common and the reference domain. The propagation of the alignment error before and after Step 5 of Algorithm 2 is discussed in Section 5.4.1. Section 5.4.2 discusses the computational complexity of NFNalign together with the different approaches to establish pixel correspondences.

5.4.1 Calculating the alignment error

In this section, the performance of Step 4 in Algorithm 2 to minimize the alignment error via re-scaling of the transformed sample $\widetilde{\boldsymbol{x}}$ is discussed.

It is shown that the alignment error is non-increasing under this algorithm and that the calculated scaling factor λ^* is optimal with respect to the length of $\widetilde{\boldsymbol{x}}$. Additionally, it is shown that the final alignment error $\widetilde{\delta}^* = \|\lambda^*\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}^*\|_2$ in the common domain $\mathcal{C}$ is identical to the alignment error δ^* in the original reference domain $\mathcal{X}^*$.

The scaling factor λ^* minimizes the distance between the sample $\widetilde{\boldsymbol{x}}$ and the reference $\widetilde{\boldsymbol{x}}^*$ in

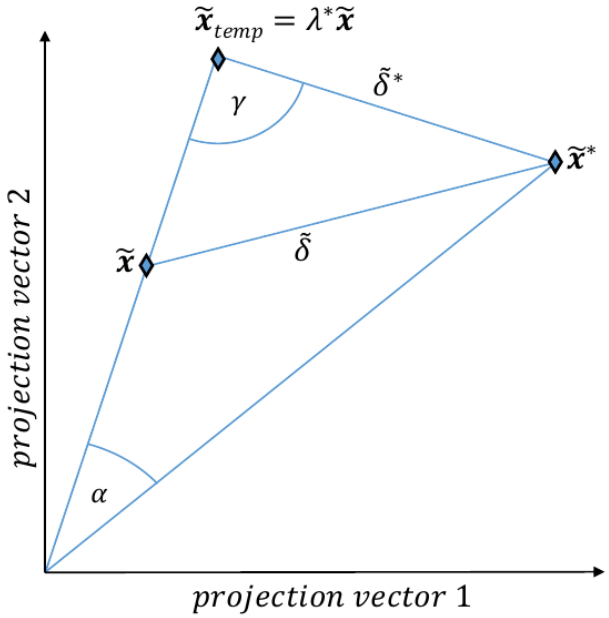


Figure 5.2: Vector presentation of two corresponding samples in their 2D projective plane. This graph illustrates the minimization of the optimal alignment error δ^* through re-scaling of $\widetilde{\boldsymbol{x}}$.

the least squares sense. The possible transformations are limited to a single scalar re-scaling of the vector $\widetilde{\boldsymbol{x}}$. This is done to preserve the spectral features of $\widetilde{\boldsymbol{x}}$.

To calculate the error progression during NFNalign, consider two corresponding samples $\boldsymbol{x} \in X$, $\boldsymbol{x}^* \in X^*$. The transformed samples $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{x}}^*$ in the common space $\mathcal{C}$ are calculated to the same basis B with Algorithm 1. Without loss of generality, the problem can be reduced to a 2D projection of the high-dimensional space. Thus, the following calculations are carried out in the 2-dimensional plane spanned by the vectors $\widetilde{\boldsymbol{x}}$ and $\widetilde{\boldsymbol{x}}^*$. This does not affect the angle between the samples. The re-scaling only adjusts the length of the sample vector $\widetilde{\boldsymbol{x}}$, but not the direction.

To find the scaling $\lambda^* \in \mathbb{R}$ of $\widetilde{\boldsymbol{x}}$ that minimizes the distance $\widetilde{\delta}^*$ in the least squares sense, the following optimization problem is considered.

$$\arg\min_{\lambda>0} \|\lambda\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}^*\|_2 \tag{5.7}$$

For the optimal scaling factor λ^* the angle γ in Figure 5.2 is $\pi/2$.
With this, the first step is to calculate the angle α by

$$\alpha = arccos\left(\frac{\widetilde{\boldsymbol{x}}^T\widetilde{\boldsymbol{x}}^*}{\|\widetilde{\boldsymbol{x}}\|_2\|\widetilde{\boldsymbol{x}}^*\|_2}\right). \tag{5.8}$$

This results in

$$\lambda^* = \alpha\frac{\|\widetilde{\boldsymbol{x}}^*\|_2}{\|\widetilde{\boldsymbol{x}}\|_2}. \tag{5.9}$$

The minimal distance $\widetilde{\delta}^*$ between $\widetilde{\boldsymbol{x}}^*$ and $\widetilde{\boldsymbol{x}}_{temp} = \lambda^*\widetilde{\boldsymbol{x}}$ is then

$$\widetilde{\delta}^* = \|\lambda^*\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}^*\|_2. \tag{5.10}$$

The propagation of the alignment error during the inverse transformation can then be calculated by applying the inverse transformation originally performed on $\boldsymbol{x}^*$ to $\widetilde{\boldsymbol{x}}_{temp}$. Then, $\boldsymbol{x}_{align}$ is calculated by

$$\boldsymbol{x}_{align} = \widetilde{\boldsymbol{x}}_{temp} - \sum_{j=1}^{p}\widetilde{c}_j^*\widetilde{\boldsymbol{d}}_j^*, \tag{5.11}$$

where $\widetilde{c}_j^*$ and $\widetilde{\boldsymbol{d}}_j^*$ are the translation vectors and coefficients for transforming $\boldsymbol{x}^*$ to $\widetilde{\boldsymbol{x}}^*$ (see Step 5 of Algorithm 2).
To calculate the alignment error in the reference domain $\mathcal{X}^*$ consider the following

$$\begin{aligned}
\delta^* &= \|\widetilde{\boldsymbol{x}}_{temp} - \widetilde{\boldsymbol{x}}^*\|_2 && (5.12)\\
&= \left\|\widetilde{\boldsymbol{x}}_{temp} - \left(\boldsymbol{x}^* + \sum_{j=1}^{p}\widetilde{c}_j^*\widetilde{\boldsymbol{d}}_j^*\right)\right\|_2 && (5.13)\\
&= \left\|\left(\widetilde{\boldsymbol{x}}_{temp} - \sum_{j=1}^{p}\widetilde{c}_j^*\widetilde{\boldsymbol{d}}_j^*\right) - \boldsymbol{x}^*\right\|_2 && (5.14)\\
&= \|\boldsymbol{x}_{align} - \boldsymbol{x}^*\|_2 && (5.15)
\end{aligned}$$

The inversion from $\widetilde{\boldsymbol{x}}_{temp} \to \mathcal{X}^*$ is a concatenation of vector additions. It employs the same

translation vectors that were used to calculate $\widetilde{\boldsymbol{x}}^*$ from $\boldsymbol{x}^*$. Together with the changed sign of the translation this simply results in a parallel shift with inverted direction, thus, the alignment error $\widetilde{\delta}^*$ is invariant under this transformation and is carried over to $\mathcal{X}^*$.
The relationship between the original error $\widetilde{\delta} = \|\widetilde{\boldsymbol{x}} - \widetilde{\boldsymbol{x}}^*\|_2$ and the minimized error $\widetilde{\delta}^*$ is given by

$$cos(\beta) = \frac{\widetilde{\delta}^*}{\widetilde{\delta}} \Rightarrow \widetilde{\delta}^* = cos(\beta)\widetilde{\delta} \tag{5.16}$$

Since $\gamma = \pi/2$, it automatically follows that $\beta \in [0, \pi/2]$. This translates to $\cos(\beta_2) \in [0, 1]$, and thus $\widetilde{\delta}^* \leq \widetilde{\delta}$.
This means in conclusion that the alignment error is non-increasing with the proposed re-scaling and is carried over from the common domain $\mathcal{C}$ to the reference domain $\mathcal{X}^*$.

5.4.2 Computational Complexity of NFNalign

NFNalign is mainly composed of two or more individual NFN calculations. Each individual transformation can be calculated in $\mathcal{O}(ndlk)$ (see 4.4.3). When assuming only two data sets with the same number of samples, the number of operations for the initial transformation can simply be doubled. The re-scaling of the test sample $\boldsymbol{x}^*$ with Step 4 of Algorithm 2 is required. This can be done in $\mathcal{O}(d)$ and does not change the asymptotic complexity of $\mathcal{O}(ndlk)$. This also holds for the inverse translation, as it is only in $\mathcal{O}(Nd)$.
Additional operations are required to calculate the pixel correspondences. Depending on the kind of correspondence, spatial or spectral, the computational complexity changes.
Using the ECC algorithm to spatially align two data sets of the same location with the same number of pixels n additionally depends on the number of parameters q for the desired transformation, e.g., $q = 2$ for translation, $q = 6$ for an affine transformation and $q = 8$ parameters for a homography. The MATLAB code from Section 5.3.1 can then be computed in $\mathcal{O}(qn^2)$ per iteration. Restricting the procedure to specific classes of parametric models can reduce the computational complexity to $\mathcal{O}(qn)$ per iteration.

The practical runtime test for ECC alignment of two data sets with $n = 670*606$ samples was performed on an office computer with Intel Core i7-6700, 32 GB RAM, and NVIDIA Quadro K1200. After 20 iterations the results were good enough for pixel-by-pixel correspondence. The calculation was completed in ~ 50 seconds.

Calculating the pixel correspondences via spectral similarity is performed by computing the spectral angle between one sample and an image with n samples. This requires $\mathcal{O}(nd)$ operations according to Equation (5.8). To find all the spectral correspondences between two images of n samples each increases this to $\mathcal{O}(n^2 d)$.

In practice, this is very time-consuming. An approach to reduce this calculation is to only compute the spectral similarities between the test data $\widetilde{X}$ and a subset of the reference data $\widetilde{X}^*$ in the common domain $\mathcal{C}$. A systematic or random selection of a set percentage of samples, segmentation [61] or active learning [138] can reduce the reference samples to a suitable subset.

Per definition, the transformed training samples $\widetilde{L}^*$ can be ignored, as they are identical in $\mathcal{C}$ up to the norm.

Overall, NFN can be implemented as a very fast and computationally efficient way for MA. ECC maximization for co-registration is only required once per data set (reference excluded) making subsequent optimization of different parameter combinations for t and k even faster. This is not true for correspondences based on spectral similarity, as parameter variations can affect spectral similarities.

Chapter 6

Experiments

In this chapter, the performance of the proposed NFN algorithm (see Chapter 4) and NFNalign algorithm (see Chapter 5) is evaluated. This is done by applying the NFN to commonly used hyperspectral benchmark data and comparing classification results with and without the transformation. Additionally, specific data sets were prepared to show the capability of NFNalign for MA and transfer learning. NFNalign is also tested for transfer learning in a multimodal setting. The algorithms robustness and their sensitivity towards the amount of training data and parameter variations are tested.
The chapter is structured as follows. In Section 6.1 the data sets used for the evaluation are introduced. Section 6.2 explains the training data selection and evaluation metrics for the following sections. Section 6.3 contains the experiments analyzing the mitigation of nonlinear effects by a single application of the NFN algorithm. Consequently, section 6.4 deals with the evaluation of NFNalign for data alignment and FT, including multimodal alignment.

6.1 Data Sets

In this section, the data sets used for evaluation are introduced. The focus lies on the new benchmark for data alignment and FT in Subsection 6.1.1. It was specifically created for

Table 6.1: Quick reference guide with important parameters of all used data sets and their use in different experiments.

	Greding1-3 _rad/_refl	Zurich'02 Zurich'06	Montelly Prilly	Pavia Pavia University	KSC
Size in pixel	670 × 606 (each)	329 × 347 (each)	1064 × 1248 1040 × 1032	1096 × 715 610 × 340	512 × 614
# bands	127	4	8	102 103	176
Labeled pixel	127688 104273 105655	13240 (each)	727534 596975	148142 42776	927
# classes	6	8	6	9	13
Alignment method	spatial	spatial	spectral	no	no
Multi-modal alignment	yes	no	no	no	no

this thesis and was approved for sharing with the scientific community[1]. The data sets used for the evaluation of the NFN algorithm are common benchmark data sets for classification. They are introduced in Subsection 6.1.3. Additional multitemporal and multilocal data, which are used in the data alignment experiment with NFNalign are also introduced here. These data sets are currently not openly available. They were acquired through Professor Devis Tuia of the University of the Wageningen University, Netherlands.

Table 6.1 gives a summary of the most important parameters of all used data sets as well as their applicability for each algorithm. All data sets are at least analyzed for mitigation of nonlinear effects with NFN. A full list of parameters including images, ground truth masks, and label information for each data set can be found in Appendix A.

[1]To get a copy of the data set; please contact the author of this thesis at wolfgang.gross@iosb.fraunhofer.de

6.1.1 New Feature Transfer Benchmark Data Set

Evaluation of MA and FT is difficult due to a lack of suitable benchmark data. Ideally, such a benchmark data set consists of multiple images recorded at different times over the same region. It should contain the same classes, but with natural variations, e.g., caused by varying illumination, viewing angles and object geometry. Another interesting aspect that results in an enormous challenge to MA procedures is data sets from different sensors or at different levels in the data pre-processing chain. Unfortunately, commonly used hyperspectral benchmark data sets usually consist of a single image with a corresponding ground truth mask of labeled areas. As these data sets can still be useful to evaluate the mitigation of nonlinear effects of the proposed NFN transformation, they are introduced in subsection 6.1.3.

Here, the focus is on creating a suitable benchmark data set to evaluate MA and FT with NFNalign. For this, three images from a measurement campaign conducted in 2014 over the village of Greding, Germany, were selected [46, 47]. The Greding data was recorded with an aisaEAGLE II hyperspectral sensor. The sensor covers the wavelength range of $390 - 990$ nm with up to 488 bands. With an altitude of approximately 770 m above ground, a frame rate of 60 Hz was necessary to create square pixels with a ground resolution of 0.5×0.5 m. It is possible to perform hardware-based spectral binning to allow high frame rates. Thus, the data was only recorded with 127 bands, where each band is an average of up to 4 neighboring bands. The binning has the added benefit of effectively increasing the SNR by a factor of 2 for bands comprised of 4 neighboring detector elements in the spectral direction.

The selected images all depict approximately the same rural area, consisting mostly of single-family homes, single-lane roads, meadows, trees, and some agricultural fields. A comparison of the respective *Colored Infrared* (CIR) composites calculated from the reflectance data can be seen in Figure 6.1. All images were cut to 670×606 pixels to match the Greding1 data set. Small black areas on the borders of Greding2 and Greding3 data are a result of ECC image warping, which was done to establish the spatial pixel correspondence (see Section 5.3.1,[29]). The areas without spectral signatures are ignored in the evaluation process.

The data sets were extracted from longer flight lines and radiometrically and geometrically preprocessed. Radiometric preprocessing included dark current removal and sensor-specific correction using calibration data from the manufacturer and their proprietary software CaliGeo. This procedure translates the raw digital numbers in the data to at-sensor radiance values. Subsequently, the atmospheric correction was performed using ATCOR [103] to calculate ground reflectance values for each sample. This process was performed multiple times with varying parameters until a satisfactory match between corrected samples and ground measurements of multiple references, e.g., asphalt, bitumen, spectralon, was achieved. The evaluation was done by comparing single sample spectra in the atmospherically corrected data to reflectance spectra from an ASD FieldSpec4 [26], that were recorded during this campaign. Since atmospheric correction is performed for the whole data set, local effects like cloud shadows and nonlinearities due to object geometry are not corrected.
Spatial co-registration between all data sets was achieved through georeferencing. For this, GPS / INS information was used to project the data to a simultaneously recorded LiDAR DEM. Boresight correction [75] followed by image-based warping using the ECC algorithm (see 5.3.1, [29]) lead to almost pixel-perfect co-registration. However, this does not take occlusions due to different viewing angles into account. One example is visible walls of houses in the Greding1 and Greding3 data, while the Greding2 data was recorded with an almost perfect nadir view. A mosaic of the co-registered data is depicted in Figure 6.3.
For each image, the ground truth for the six dominant land cover classes, Dark roof (ocher), Red roof (red), Asphalt (gray), Soil (brown), Grass (green), and Tree (dark green) was manually selected. A comparison of the individual ground truth masks can be seen in Figure 6.2. The ground truth was selected with the following rationale:

1. The same area should be select in all three images, if possible. It can dramatically reduce the amount of manual work if the co-registration by GPS / INS measurement and image warping is good enough for pixel-by-pixel correspondence.
2. Selected areas of a single class should encompass all featured spectral variations in the scene, e.g., grass in direct sunlight and the shade of a tree/building.

(a) CIR composite of Greding1 reflectance data

(b) CIR composite of Greding2 reflectance data

(c) CIR composite of Greding3 reflectance data

Figure 6.1: Comparison of the Greding reflectance data set scaled individually for best visualization. The differences between Greding1 and Greding2 is a small change in viewing angle and solar position resulting in an alteration in the direction shadows are cast. Greding3 data was recorded with areas affected by cloud shadows.

3. Selected areas should be as large as possible without selecting samples from another class or background clutter. This point is mentioned to enforce an exhaustive ground truth map allowing for variable partitioning into training and test areas.

Each image is now available in at-sensor radiance and ground reflectance values, allowing the evaluation of FT across domains. Reflectance data of Greding1 was chosen as the reference X^* for all experiments involving MA due to optimal illumination conditions. Since no spatial distortion happens during the atmospheric correction, ground truth masks can be used for either domain. A complete list of data set parameters, including label information for each class, can be found in Appendix A.

6.1.2 Additional Benchmark for NFNalign

Additional data sets were acquired to supplement the specifically prepared Greding data and to further demonstrate the usefulness of NFNalign.The following data sets are used in addition to the Greding data for the evaluation of NFNalign.

1. Zurich'02 and Zurich'06 data [81]: The Zurich'02/'06 data consists of two images

(a) Ground truth mask for Greding1

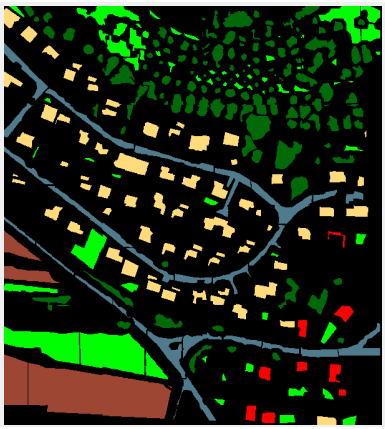
(b) Ground truth mask for Greding2

(c) Ground truth mask for Greding3

Figure 6.2: Comparison of Greding ground truth masks. The same areas were selected in all images.

(a) Mosaic of RGB composites

(b) Mosaic of CIR composites



(c) Mosaic of ground truth

Figure 6.3: Greding mosaics; ground truth masks. The same areas were selected in all images.

recorded by the QuickBird satellite. The first image was recorded in August 2002, the second in October 2006 over the residential neighborhood of Brüttisellen, Zurich. The sensor provides only four bands, three in the visible and one in the near infrared. The data sets are well co-registered so that pixel-by-pixel correspondence can be used for NFNalign. The spatial resolution of both data sets is 2.6 m. Different acquisition conditions, e.g., variations in off-nadir angles and seasonal effects of the vegetation cycle complicate data synthesis and make adaptation necessary according to [121]. A large number of pixels were labeled into nine classes by accurate photo-interpretation and manual selection.

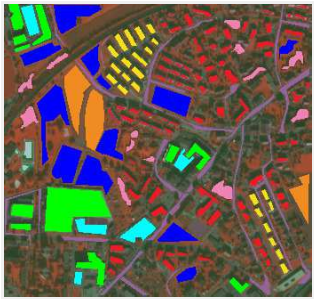
2. Prilly/Montelly data [120]: The data set was recorded over the city of Lausanne, Switzerland, in 2011 and consists of two spatially separate scenes from the neighborhoods of Prilly and Montelly, respectively. The WorldView-2 data was pansharpened with the Gram-Schmidt method [70] to reach a spatial resolution of 0.6 m. The sensor records eight bands with bandwidths between $40 - 125$ nm. Since the original ground truth information of the Montelly data set consists of 10 classes, only the six classes common to both data sets are used for evaluation. The spatial offset between the depicted areas makes the use of pixel-by-pixel correspondences impossible. In this case, spectral similarities in the common domain $\mathcal{C}$ are used to perform NFNalign and evaluate the success of FT (see Section 5.3.2).

The Zurich and Montelly/Prilly data were courtesy of Devis Tuia from the Wageningen University, Netherlands. The Zurich and Montelly/Prilly data sets are not commonly used as classification benchmarks but are a great asset for MA and FT in this thesis. The CIR images and ground truth are depicted in Figure 6.4 and Figure 6.5.

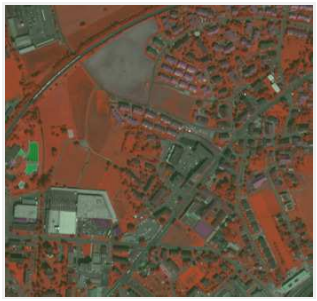
6.1.3 Common Benchmark Data

There also exist multiple openly available hyperspectral benchmark data sets. They are generally georeferenced and radiometrically and atmospherically corrected. The chosen data

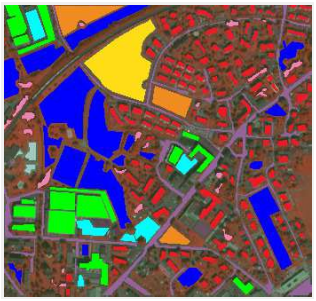
(a) CIR composite of Zurich'02 data set



(b) CIR composite of Zurich'02 data set with superimposed mask

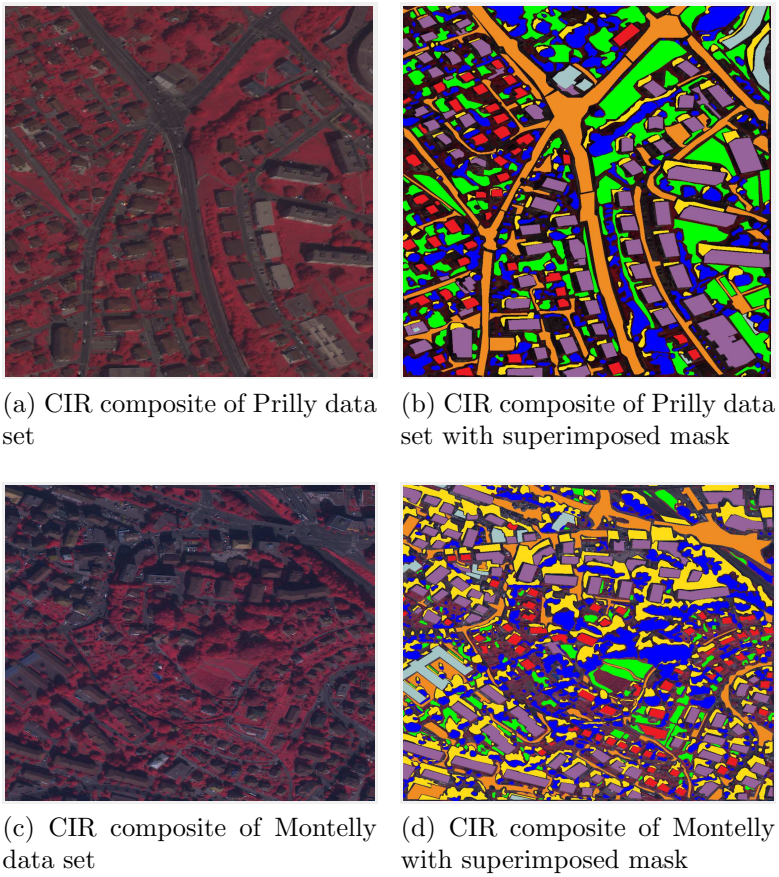


(c) CIR composite of Zurich'06 data set



(d) CIR composite of Zurich'06 with superimposed mask

Figure 6.4: CIR composites of Zurich data from 2002 and 2006, and overlay with respective ground truth masks.



(a) CIR composite of Prilly data set

(b) CIR composite of Prilly data set with superimposed mask

(c) CIR composite of Montelly data set

(d) CIR composite of Montelly with superimposed mask

Figure 6.5: CIR composites of Prilly/Montelly data and overlay with respective ground truth masks.

sets come with ground truth masks. Those data sets only consist of a single image with corresponding ground truth. This means only the performance of the NFN algorithm for mitigating nonlinear effects can be evaluated. The following data sets are used.

1. Pavia and Pavia University data [100]: The Pavia data set consists of two different scenes acquired by the ROSIS sensor over Pavia, Italy. The sensor covers the wavelength range of 430-860 nm with a ground resolution of 1.3 m. The Pavia Center data consists of 1097×715 samples with 102 bands each. The Pavia University data has 610×340 samples at 103 bands. Both data sets were previously corrected for pixels that contained erroneous information. Ground truth of both data sets is partitioned into nine different classes. Unfortunately, only 4 of the classes are present in both scenes, which prevents direct data alignment with NFNalign using a pixel correspondence map built from spectral similarity (see 5.3). Also, a shadow class is present in both images. For the proposed method, these areas would ideally be labeled as one of the other classes as shadow is one effect NFN can mitigate.
2. *Kennedy Space Center* (KSC) data [55]: The KSC data was acquired by the NASA *Airborne Visible/Infrared Imaging Spectrometer* (AVIRIS) in 1996. The AVIRIS sensor collects data from $400 - 2500$ nm in 224 bands of 10 nm width. The ground resolution is 18 m due to the high altitude of approximately 20 km. Bands with water absorption features or low SNR were removed, and 176 bands remained for analysis. The resulting image is a scene with 512×614 pixel. Ground truth was partitioned into 13 classes by using landcover maps from color infrared photography from the Landsat Thematic Mapper imagery. [55] states the difficulty of discriminating between the classes due to spectral similarity for certain vegetation types. Due to the low spatial resolution mixtures of multiple class spectra among the labeled pixels are possible.

The Pavia and KSC data can be downloaded from `http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes`. Together with the Greding, Zurich, and Montelly/Prilly data, this list of data sets allows comprehensive evaluation of NFN and

NFNalign.

6.2 Preparation and Evaluation Metrics

This section is dedicated to the data set and ground truth data preparation. The goal is to evaluate NFN and NFNalign in a realistic setting. Different strategies for training data selection are discussed. All commonly used benchmark data sets from Section 6.1.3 are already corrected for bad bands. All available bands for the Zurich, Montelly/Prilly and Greding data are used to demonstrate the easy accessibility of NFN.

6.2.1 Training Data Selection

All data sets come with labeled masks containing the ground truth information per image pixel. In this thesis, two different approaches for dividing the ground truth into disjoint training and test data sets are compared.

Systematic Sampling: For Systematic Sampling a fixed fraction α of spectra from each class are selected by searching for the indices of a given class j in the ground truth map. Assuming N_j pixel are labeled as class j every $\lfloor \alpha^{-1} \rfloor$-th entry, beginning with the first, are selected as training data.

This approach aims to give a uniform distribution of training data over the scene and was chosen in favor of random training sample selection for its reproducibility. One drawback of this approach is that test and training samples can be spatially next to each other. Because similar objects are usually close to each other, this can artificially improve the evaluation, as the likelihood of neighboring pixels being classified as the same label as the training sample is very high. This approach is used nonetheless to evaluate the effects of different parameter variations for k and t. This approach also has the highest likelihood of satisfying Assumption 2, the assumption that the training data contains all possible variations per class, without manual interaction.

Cluster Sampling: This approach aims to keep training data spatially separate from the test data to eliminate correlation and neighborhood effects. However, it is important to note that without manual interaction Assumption 2 cannot be guaranteed. The algorithm to divide the ground truth according to Cluster Sampling can be found in [57]. The basic idea is to perform a k-NN clustering on the ground truth mask while using the spatial pixel coordinates as additional features. This process ensures that the likelihood of two neighboring pixels ending up in the same set (either training or test) is higher compared to two pixels on opposite sides of the scene. The set percentage of training data is enforced by setting the maximum number of training samples per class. The test samples are simply the ground truth samples that were not selected to be training samples.

Using Cluster Sampling as it was introduced in [57] produces bad results due to the non-representative training data selection. It generally violates Assumption 2 from section 4.1 as it only selects training data from a small area. Without modification, this algorithm covers only a small amount of the nonlinear effects as hyperspectral data generally suffers from across-track illumination variation, which interferes with classification and spectral matching [95]. Thus, selection of training areas from different sides of the data set is mandatory. For this, the Cluster Sampling algorithm is modified to *Rotation Invariant Cluster Sampling* (RICS). The goal of RICS is to select training data near the corners and along the edges of the data set.

The following adjusted algorithm has to be performed for every class $j = 1, \ldots, p$. The class index is dropped here to simplify the notation. Let $I \in \mathbb{N}^{n_x \times n_y}$ be the ground truth mask of a scene and $\tau \in \mathbb{N}$ the number of samples of class j. Let $\boldsymbol{s} = (\boldsymbol{s}_x, \boldsymbol{s}_y)$ be the tuple of spatial coordinates for all samples with the same class label j. Assuming without loss of generality that $S = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\tau)$ is ordered according to

$$\|\boldsymbol{s}\|_2 = \left(\sqrt{\boldsymbol{s}_x^2 + \boldsymbol{s}_y^2}\right) \tag{6.1}$$

in ascending order, it is possible to select the first half of the training data directly. If τ is the number of samples in S, the training data are the first $\alpha\tau/4$ entries of S as well as the

last $\alpha\tau/4$ entries. With this, the selected training data clusters are located in the upper left and the lower right corner of I. The samples in the other two corners can be selected by simply rotating I by $\pi/2$ and repeating the process.

Depending on the spatial class distribution and α the same location may be selected twice, once before and once after rotation. This effect reduces the effective amount of training samples as the inclusion of multiple instances of the same sample would skew the k-NN step of the proposed transformation.

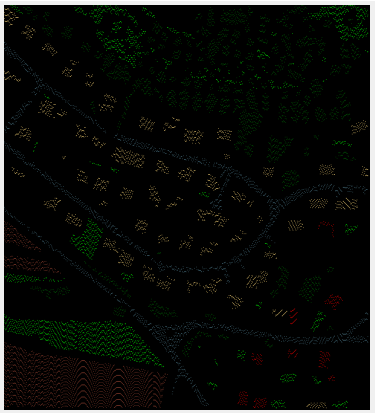
If S is not sorted from the start, resorting and index transfer is required to perform the same operations.

The corresponding results using the different selection methods are marked by S for Systematic Sampling and C for Cluster Sampling, respectively. A comparison of training and test masks for Systematic Sampling and RICS is depicted in Figure 6.6 for an amount of 10 % training spectra per class.

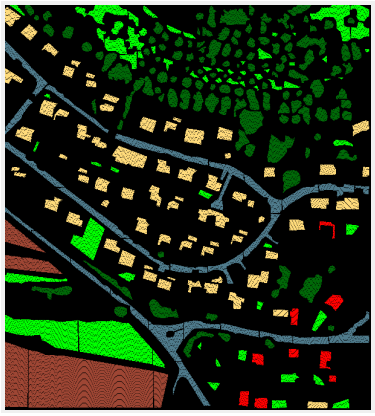
Additional training sets are used to evaluate the robustness of the proposed algorithms. These are generated by randomly assigning wrong class labels to a specified fraction of training data. The goal is to evaluate the properties of NFN and NFNalign when the training data contains outliers. This setup was chosen to simulate errors in the ground truth selection process. The corresponding results are marked by, e.g., $0.1swap$ for a 10 % swap of training data per class. For NFNalign flawless training data is assumed for the reference data, and only a certain amount of training data in the test images is swapped. This experiment was designed to simulate a well-curated reference data, where labels and training data were carefully selected, compared to the test data where the labels were quickly produced to perform FT as quickly as possible for a new data set.

6.2.2 Spectral Angle Mapper

The goal of NFN transformation is to reduce the effect of nonlinearities in the data. This is done by individually calculating a translation for each sample, that aligns the data set with a new basis. Each of the basis vectors is related with one class of the data set. The training

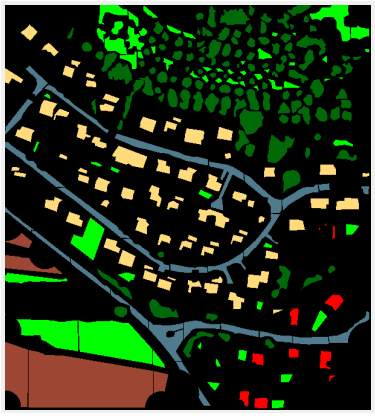


(a) Systematic Sampling of Greding1 ground truth with 10 % training data



(b) Complementary test data sampling for Systematic Sampling

(c) RICS of Greding1 ground truth with 10 % training data



(d) Complementary test data sampling for RICS

Figure 6.6: Comparison of training and test locations for Systematic Sampling and RICS.

samples, when chosen directly from the image, are identified with the corresponding basis vector up to norm. Assuming mostly complete training data with regards to encountered nonlinear effects, NFN performs data transformation such that individual samples are drawn more closely towards the basis vector with the closest training data. This means that radial distance between samples of different classes increases, while simultaneously radial intra-class distance is reduced. To measure the improvement of radial separability, the SAM classifier is used [67]. Treating each sample as a vector from the origin to the coordinates of the sample, the angle α between two samples $x, y \in \mathbb{R}^d$ is calculated as

$$\alpha(\boldsymbol{x}, \boldsymbol{y}) = arccos\left(\frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2}\right). \tag{6.2}$$

The SAM classifier computes the index of minimal spectral angle between a test sample $\boldsymbol{x}$ and the class reference spectra $\boldsymbol{b}_j$ for $j = 1, \ldots, p$. This means a sample $\boldsymbol{x}$ belongs to class j^*, if $\alpha(\boldsymbol{x}, \boldsymbol{b}_j^*) < \alpha(\boldsymbol{x}, \boldsymbol{b}_j) \forall j \neq j^*$. A smaller angle α represents a closer match between the sample $\boldsymbol{x}$ and the reference $\boldsymbol{b}_j$.

The denominator of Equation (6.2) makes the result invariant to the length of the vectors, thus, making the SAM relatively robust against illumination changes [134]. However, it was shown in [72] that SAM is sensitive to nonlinearities like shadows, changes in viewing angle and object geometry. This property makes the SAM classifier a perfect fit to evaluate the mitigation of nonlinearities by NFN. If the theory behind the NFN algorithm is correct, SAM classification should improve after NFN transformation compared to its application on the original data.

6.2.3 Support Vector Machine

The SAM classifier can be used to evaluate the property of the NFN algorithm to mitigate nonlinearities in the data. To evaluate NFNalign, however, a nonlinear classifier is required, as the algorithm transforms a test data set X into the domain of the reference data X^*. The transformation from the common domain $\mathcal{C}$ to $\mathcal{X}^*$ introduces the characteristic nonlinearities

of X^* to X.

To evaluate the performance of NFNalign regarding data alignment and FT a nonlinear classifier trained on the training data L^* is required. When the data alignment by NFNalign is successful, classification of the aligned data X_{align} should be comparable to the classification of X with the same classifier trained on the original training data L.

For this task, the SVM classification is selected. The SVM is a binary classifier defined by a separating hyperplane, also called decision plane [131]. For a given set of training samples, the algorithm calculates a decision function represented by an optimal hyperplane to categorize new samples [24]. The decision function is defined by a relatively small set of training samples; the so-called support vectors, that maximize the margin around the hyperplane. This problem can be formulated as a quadratic optimization problem for which efficient solvers exist [23]. For the modeling of nonlinear class structures, SVM can be adapted to calculate the decision plane in a high-dimensional space using the kernel trick (see Section 3.3.1). The transformation improves classification results even for strong nonlinearities in the data [141, 119].

The decision whether a sample belongs to class 1 or class 2 is determined by calculating the dot product between the sample and all support vectors. To adapt SVM for nonlinear decision boundaries, the dot product can be replaced with an inner product in the form

$$K(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle_{\mathcal{V}}, \tag{6.3}$$

where $\boldsymbol{x}$ is the test sample, $\boldsymbol{x}'$ is a support vector, $\varphi : \mathcal{X} \to \mathcal{V}$ the feature map, and $\mathcal{V}$ the high-dimensional space.

The kernel trick can now be used to replace explicit computation of $varphi(\boldsymbol{x})$ and $varphi(\boldsymbol{x}')$ by simply calculating $K(\boldsymbol{x}, \boldsymbol{x}')$ instead. In this case, explicitly calculating or even knowing the feature map φ is not necessary [28].

For the experiments in this thesis, the RBF kernel is used, which is based on a Gaussian function. It is defined as

$$K((\boldsymbol{x}, \boldsymbol{x}')) = exp\left(-\gamma \left\|\boldsymbol{x} - \boldsymbol{x}'\right\|_2^2\right), \tag{6.4}$$

where $\gamma = 1/(2\sigma^2)$ is a free parameter to control the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$. γ is the inverse of the standard deviation of the RBF kernel. A small γ defines a Gaussian function with a large variance. In this case two points $\boldsymbol{x}$ and $\boldsymbol{x}'$ can be considered similar even if they are far apart from each other. Conversely, for large γ the points $\boldsymbol{x}$ and $\boldsymbol{x}'$ have to be close to each other to be considered similar. Finding the optimal parameter γ is usually done with a grid search.

Since SVM is a binary classifier (p=2) and the data sets in this thesis have more than two classes, the general approach is to decompose the classification problem with $p > 2$ into multiple binary problems [133, 131]. The practical calculation for the experiments uses the libSVM MATLAB package as a multi-class SVM where each class is in turn separated from all others (one-against-all classification) with optimized step size selection[15].[2]

6.2.4 Evaluation metrics

The proposed NFN and NFNalign algorithms are applied to different scenarios (see Sections 6.3 and 6.4). The quality of NFN for mitigating nonlinear effects is measured using the linear SAM classifier (see Section 6.2.2 and comparing the results before and after NFN transformation. If the NFN successfully mitigates nonlinear effects, the classification results are expected to improve on the transformed data.

NFNalign is evaluated by comparing classification results of a nonlinear SVM using an RBF kernel (see Section 6.2.3) on the aligned data, using only the training data L^* from the reference X^*.

To measure the quality of the classification in a single value the Cohen's Kappa coefficient

[2]The libSVM package can be downloaded here: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

κ is utilized [19]. This coefficient is a statistic to measure inter-rater reliability. It can be written as

$$\kappa = \frac{p_0 - p_e}{1 - p_e}, \tag{6.5}$$

where p_0 is the relative observed agreement among raters and p_e is the probability of accidental agreement. This measure is more reliable than overall accuracy due to the fact that class imbalances are incorporated. **Example**: Consider the following problem with two classes and a ground truth of

$$GT = \begin{pmatrix} 98 & 0 \\ 0 & 2 \end{pmatrix} \tag{6.6}$$

and the two different classification results

$$C_1 = \begin{pmatrix} 98 & 1 \\ 0 & 1 \end{pmatrix} \text{and} \, C_2 = \begin{pmatrix} 97 & 0 \\ 1 & 2 \end{pmatrix}. \tag{6.7}$$

The overall accuracy for C_1 is 0.99 %, while $\kappa(C_1) \approx 0.66$. For C_2 the accuracy is 0.99 %, while $\kappa(C_2) \approx 0.80$. This means that a single mislabeled sample has no influence on the overall accuracy regardless of representation. For κ, however, it is a big difference if half of the samples in one class are categorized differently, regardless of the total number of samples in that class. Some data sets have dominant classes, e.g. Pavia where almost half of the labeled samples are water. Not differentiating between individual class performances would skew the results.

Additionally, measuring the spectral match between corresponding samples can also be used as a lead to determine the quality of NFNalign since it aims at transforming one data set to the domain of another. For two corresponding samples $\boldsymbol{x}$ and $\boldsymbol{x}*$ the RMSE can be calculated as

$$RMSE(\boldsymbol{x}, \boldsymbol{x}^*) = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - x_i^*)^2}. \tag{6.8}$$

Here, d is again the dimension of the data sets and the x_i, x_i^* are the scalar entries of $\boldsymbol{x}$ and $\boldsymbol{x}^*$, respectively.

For an image $X \in \mathbb{R}^{d \times n}$ there exist n correspondences with $X^* \in \mathbb{R}^{d \times n^*}$. To produce a single value from the n RMSEs the mean value of all corresponding RMSEs is calculated. For simplicity this value is denoted as τ.
A small τ is a good indicator that an SVM classification trained on the reference training data L^* will produce good results, as the space the aligned data X_{align} occupies in $\mathcal{X}^*$ is close the X^*.

6.3 Evaluation of NFN

This section analyzes the performance of NFN for reducing nonlinear effects. This is done by comparing the classification results of the linear SAM classifier (see Section 6.2.2) before and after NFN transformation. Section 6.3.1 shows the capabilities of NFN for mitigating nonlinearities. Section 6.3.2 discusses the performance for various amounts of training data, the power t of the penalty function, and the number of NNs k. Additionally, the robustness of NFN is analyzed in Section 6.3.3 when the training data is contaminated by spectra from other classes.

6.3.1 NFN for Mitigation of Nonlinear Effects

In this section, NFN transformation is performed on the data sets from 6.1. The Cohen's Kappa[32] statistic for SAM classification (see Section 6.2.2) on the original and the NFN transformed data are compared. This is done to illustrate the classification improvement of linear SAM on NFN transformed data over the original data. The SAM is a classifier that measures the similarity between a reference and a test spectrum by calculating the cosine of the angle between the two d-dimensional vectors. Small angles correspond to high similarity. The SAM is a linear classifier which makes it vulnerable to nonlinear data structures. Together with the NFN transformation, SAM can be considered a nonlinear classifier, and direct comparison appears improper at first. However, the goal to show the successful mitigation of nonlinearities is best evaluated by comparing the performance of the same linear

classifier before and after the transformation.

Table 6.2 contains the Cohen's kappa results for the different data sets. For the Greding data, only the mean result over all six data sets (three radiance and three reflectance images) is listed. The reference vector for each class was calculated as the mean value per band of the corresponding training spectra. 10 % of the labeled spectra per class were selected as training data; the remaining spectra were used for the evaluation. The parameters were set to $t = 4$ and $k = 5$. Other parameter combinations are discussed in Section 6.3.2 to evaluate the sensitivity of NFN concerning parameter tuning.

The first column of Table 6.2 shows the performance of SAM on the original data. The second column lists the SAM results after NFN transformation for Systematic Sampling, and the third for RICS (see Section 6.2.1).

The classification performed on NFN transformed data is better in all test cases, often with a difference of more than $\kappa = 0.1$. Results derived from training data selected with Systematic Sampling are the best overall. However, those results have a purely theoretical value. Neighborhood effects and an even distribution of training samples over the whole image artificially enhance the completeness of the training set. There is no practical way to acquire that form of ground truth for large areas or even whole campaigns. The results in the third column of Table 6.2 were calculated using training samples selected with RICS. While the results are much worse compared to NFN with Systematic Sampling, i.e., KSC, PaviaU, Montelly, Prilly, and Zurich data, this shows that RICS is not ideal for training data selection. There are possibly more nonlinear effects in the image, and an automatic selection of training data along the image borders may not necessarily satisfy Assumption 2 of Section 4.1. Human interaction, active or guided learning [138] etc. can help to improve the results.

The performance difference in the form of classification images for all three cases are depicted in Figure 6.7 for the PaviaU data and in Figure 6.8 for the Zurich'02 data. A detailed description of the colormap and the corresponding classes for all data sets can be found in Appendix A.

Overall, these results show that NFN successfully mitigates nonlinearities in the data. The

Table 6.2: κ statistics of SAM classification on original and NFN transformed data. SAM results of NFN transformed data are separated into Systematic Sampling and RICS for training data selection.

	κ_{SAM}	$\kappa_{SAM-NFN}$ Systematic Sampling	$\kappa_{SAM-NFN}$ RICS
KSC	0.340	**0.787**	0.645
Pavia	0.868	**0.970**	0.922
PaviaU	0.479	**0.824**	0.608
Greding	0.871	**0.984**	0.935
Montelly	0.420	**0.664**	0.566
Prilly	0.519	**0.741**	0.657
Zurich'02	0.396	**0.775**	0.683
Zurich'06	0.360	**0.707**	0.592

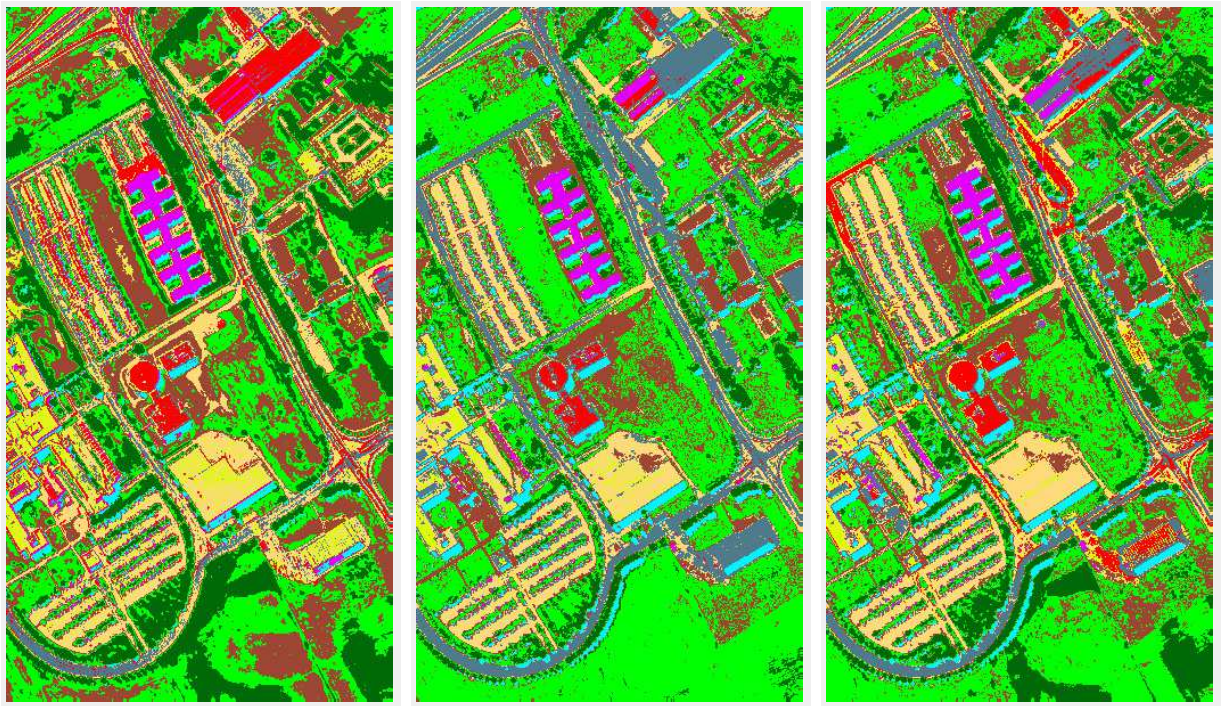
next sections analyze the impact of parameter variations and the robustness when dealing with flawed training data.

6.3.2 Parameter Optimization

This section is dedicated to analyzing different parameter combinations for the power t of the penalty function and the number of NNs k. The sensitivity of an algorithm when it comes to parameter selection and optimization is important for its practical applicability. NFN was developed for quick mitigation of nonlinearities mostly relying on user-generated ground truth. The reason for this is that a human analyst can easily decide from context whether a shadow area is the same class as one of the adjacent classes. The brain is generally much better at understanding contextual information compared to a computer.

The values $t = 1, \ldots, 5$ and $k = [1251020]$ are tested in all possible combinations for all data sets from Section 6.1. Since results are stable across all available data, only the results on the PaviaU and Zurich'02 data are shown by way of example.

Figure 6.9 shows the dependency of the SAM results after NFN transformation based on different parameters. The x-axis gives the number of NNs k. Each curve stands for a different power t of the penalty function $g(\boldsymbol{\delta}) = \boldsymbol{\delta}^{-t}$. Additionally, the SAM result on the original

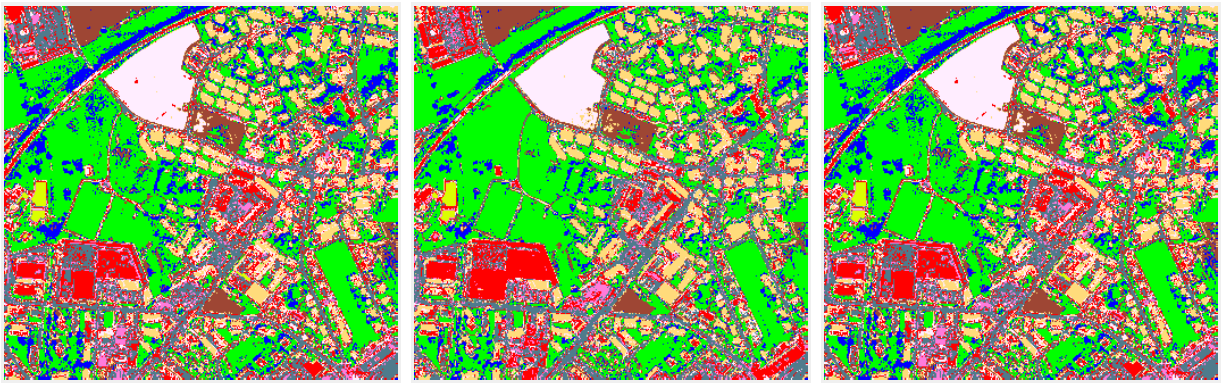


(a) SAM classification map of the original PaviaU data, $\kappa = 0.479$

(b) SAM-NFN classification map of the PaviaU data using Systematic Sampling for training data selection, $\kappa = 0.824$

(c) SAM-NFN classification map of the PaviaU data using RICS for training data selection, $\kappa = 0.608$

Figure 6.7: Classification maps for SAM, SAM-NFN with Systematic Sampling and SAM-NFN with RICS. Comparison shows that mitigation of nonlinear effects by NFN transformation followed by SAM classification is successful.



(a) SAM classification map of the original Zurich'02 data, $\kappa = 0.396$

(b) SAM-NFN classification map of the Zurich'02 data using Systematic Sampling for training data selection, $\kappa = 0.775$

(c) SAM-NFN classification map of the Zurich'02 data using RICS for training data selection, $\kappa = 0.683$

Figure 6.8: Classification maps for SAM, SAM-NFN with Systematic Sampling and SAM-NFN with RICS. Comparison shows that mitigation of nonlinear effects by NFN transformation followed by SAM classification is successful.

data is given as a baseline in green. All individual images show similar behavior. κ increases with increasing t, however, the improvement for consecutive values of t slowly decreases and eventually converges. Also, each plot shows similar behavior.

SAM on the original data is only better for the Pavia, Greding and Montelly data sets, and only for $t = 2$. For the Pavia data, this can be explained with a large number of water samples. The samples have a low magnitude and relatively high noise, which causes other samples with a similar magnitude to be drawn closer. An explanation for the Greding data are the strong nonlinearities, especially of the 'Tree' class due to the high spatial resolution. In the Montelly data set, the number of classes exceeds the number of bands, which causes false classification among all classes. These effects are all alleviated by larger t as the samples are drawn comparatively closer to their real classes during NFN transformation. In all other cases, SAM on NFN transformed data is superior for all parameter combinations.

The optimal number of NNs k, however, depends on the data set and the available ground truth for training. The variations of κ for a fixed t are especially noticeable for the KSC data. Here, using more neighbors results in a worse result. Since KSC has only 927 labeled

samples divided among 13 classes, it is evident that for an amount of 10 % training samples almost all the selected training data per class is used for every single test sample. This prevents correct sampling of nonlinearities and results in worse results compared to only using a few neighbors. The Pavia data shows the best result for $k = 2$. It is a small improvement over $k = 1$ due to the added robustness. The classification results decrease for larger k as the confusion with the dominant water class starts to affect more test samples in the neighborhood selection process. All other data sets have much more labeled samples which results in better results for larger k. This can be explained by a large number of training samples per class and their good representation of the underlying nonlinearities. Also, using more than one training samples for each individual transformation of a sample mitigates the effect of single mislabeled samples. This effect is analyzed in the next section by using manipulated ground truth with a certain amount of wrong labels per class.

Figure 6.10 shows the progression of the κ statistic for larger values of t. Geometrically, the samples are translated to be identical to the nearest reference sample $\boldsymbol{b}_j$ up to norm. The progression indicates that the classification results improve with increasing t. However, physical interpretability of the resulting spectra after NFN transformation becomes trivial as all spectra are identified with their closest class while simultaneously all features of the original spectra are lost in the process. The NFN transformation aims to serve as a nonlinear data transformation that mitigates unwanted effects. The tuning of t to mitigate nonlinearities while simultaneously preserving desirable spectral characteristics is an important topic that requires further research. This includes individual band weights for each sample depending on its neighborhood. An idea how to mitigate unwanted effects while simultaneously preserving other features is discussed in 7.2.1. The drop off at the end of the Pavia data can again be explained by the dominant water class. The same effect with a smaller impact is also observed in the PaviaU and the Greding data due to strong nonlinearities. For all other results, the monotonous behavior indicates a good selection of training samples that obey all necessary assumptions. The number of NNs k has the same effect already discussed for Figure 6.9. When a certain value of k was superior, it also transfers to the test of large t.

The constant green line stands again for the SAM classification result on the original data. For $k = 1$ letting $t \to \infty$ makes NFN default to a NN clustering as the coefficient vector $\boldsymbol{c}$ (see Equation 4.4) normalizes the largest entry to one while all others approach zero. For other values of k, the result consequently defaults to a k-NN classification.

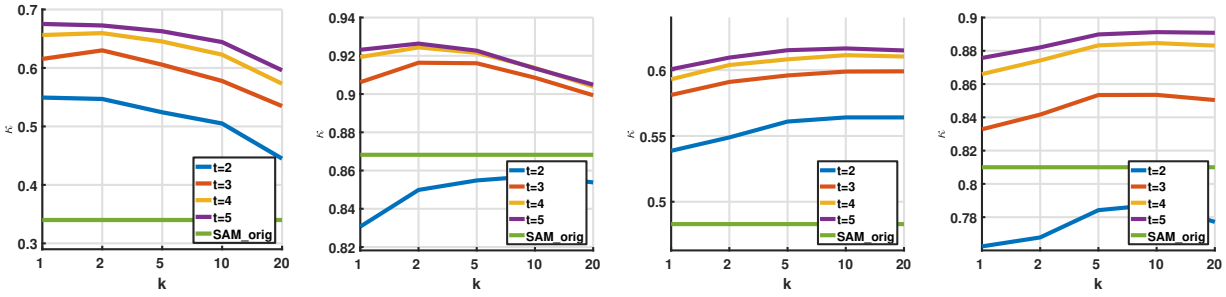
Overall, this section shows that the parameter selection process is simple, e.g., an NFN with $t = 4$ and $k = 5$ followed by a SAM classification produced results that are within 95 % of the optimal result for all data sets. Larger k should only be chosen when the training samples per class are large as well. For simple classification purposes setting $t = 50$ makes sense. However, this eliminates all features other than those of the nearest class in the transformed spectrum. Since this completely eliminates the purpose of NFNalign t is set to values between $[1, 5]$ for the data alignment experiments in Section 6.4.

When directly comparing the variation in the results for each data set, we can conclude that small values for k are better for classification, while higher values can help to mitigate effects of mislabeled ground truth. Overall, the effects of t for $t > 2$ are insignificant and accidentally choosing k bigger than necessary resulted in a decrease of κ by approximately 0.02, compared to the best test result.

6.3.3 Robustness

Until now the ground truth was considered flawless. Since ground truth masks are usually generated by human operators errors and wrong labels cannot be ruled out. In this section, varying percentages of the training samples are artificially assigned a false class label to simulate a rushed ground truth selection. This is done to analyze the impact of errors in the training data on the NFN transformation.

To calculate the flawed training samples per class the first step is to extract a set amount of samples per class with the RICS method from the ground truth mask. Then, a set amount of samples, e.g., 10 % of the training samples per class are selected and assigned a different class label at random. The other class labels are assumed to be uniformly distributed to not favor a specific class. After that, the training sets $L_1, \ldots, L_p$ are adjusted accordingly. All

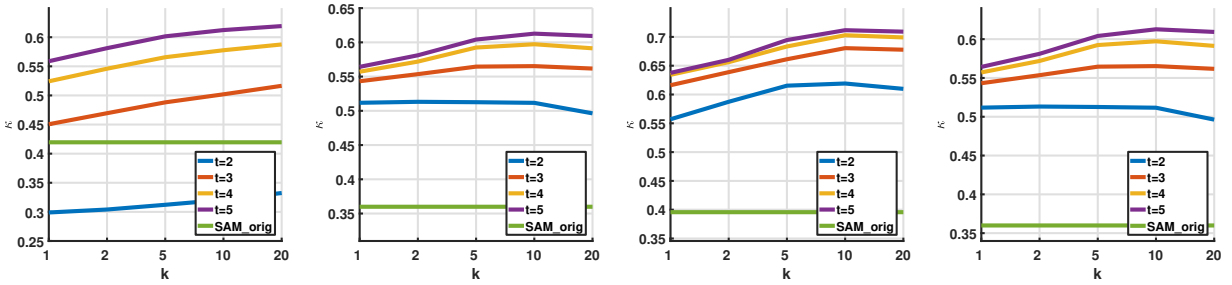


(a) KSC parameter evaluation for NFN

(b) Pavia parameter evaluation for NFN

(c) PaviaU parameter evaluation for NFN

(d) Greding parameter evaluation for NFN



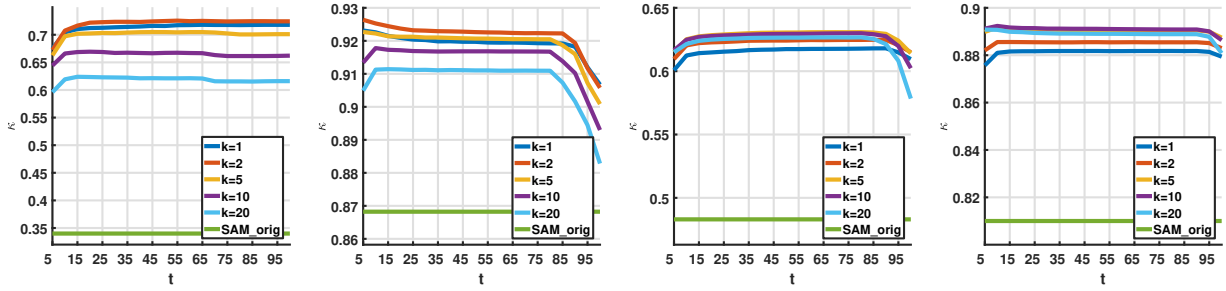
(e) Montelly parameter evaluation for NFN

(f) Prilly parameter evaluation for NFN

(g) Zurich'02 parameter evaluation for NFN

(h) Zurich'06 parameter evaluation for NFN

Figure 6.9: κ statistics for SAM classification on NFN transformed data for different parameter combinations.

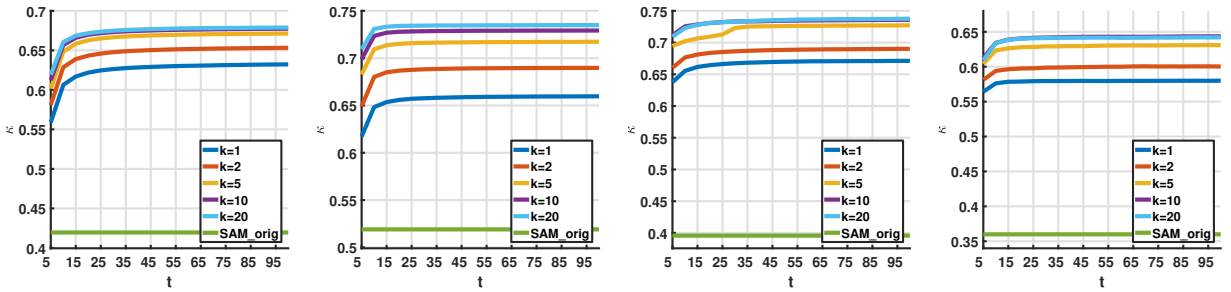


(a) KSC NFN performance for t

(b) Pavia NFN performance for t

(c) PaviaU NFN performance for t

(d) Greding NFN performance for t



(e) Montelly NFN performance for t

(f) Prilly NFN performance for t

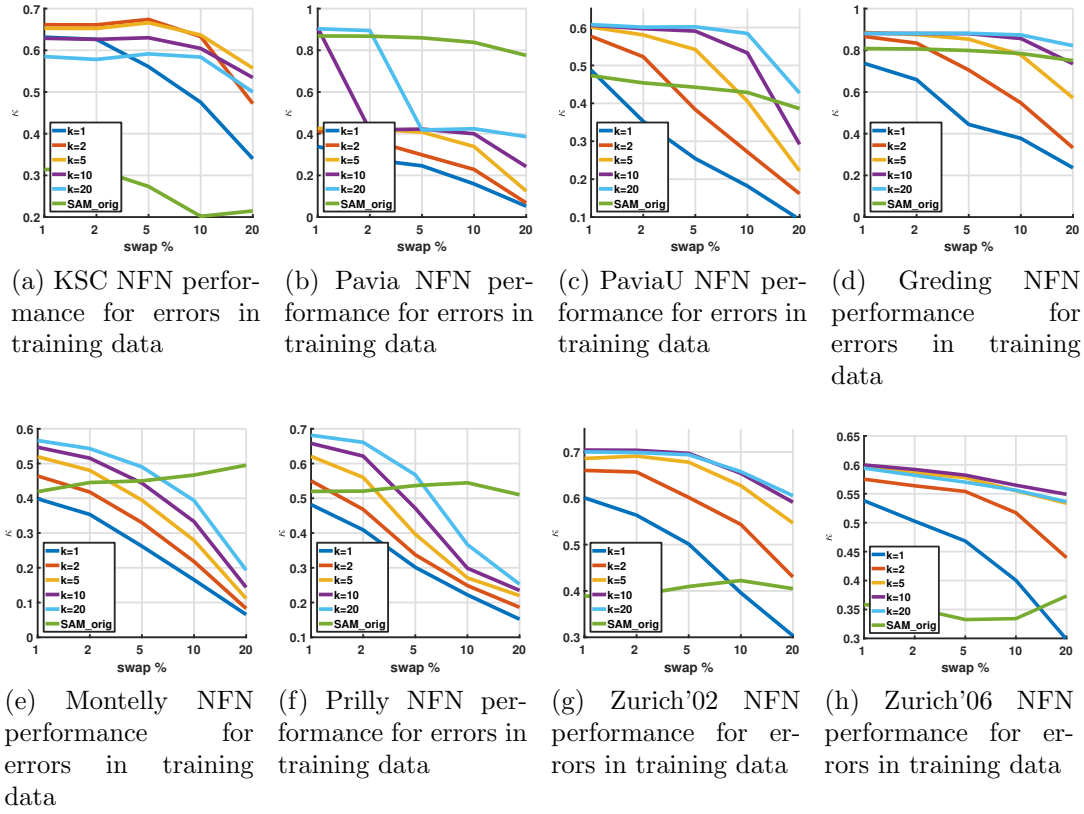
(g) Zurich'02 NFN performance for t

(h) Zurich'06 NFN performance for t

Figure 6.10: κ statistics for SAM classification on NFN transformed data for large power t of the penalty function.

results were calculated using 10 % of labeled samples for training and the remaining 90 % for the evaluation. Figure 6.11 shows the progression of the κ statistics for variable amounts of swapped training labels per class. All results were calculated using $t = 4$. The different plots are the result of a variable amount of NNs k. The general progression is very similar between the plots for different k. The monotonically decreasing behavior for an increasing number of swapped labels is expected. Here, the classification results strongly depend on the number of NNs k. Using more neighbors can mitigate the influence of a few wrong labels. Thus, $k = 10$ or $k = 20$ generally produce the best results. Only the KSC data favors smaller values for k. This can be explained by the small number of labeled samples per class. With only 10 % of the labeled samples for training, few classes have more than 20 training samples. This circumstance leads to the inclusion of wrong neighbors even with flawless training data as demonstrated in Section 6.3.2. Adding samples from other classes due to wrong labels further decreases the performance. The results on the Greding data are very stable even for high percentages of swapped labels. This fact can be attributed to the large number of available labeled samples per class. The same is true for the Zurich data to a lesser extent. The small number of $d = 8$ bands with $p = 6$ classes is responsible for the drop-off in the Montelly/Prilly results. The worst progression among the tested data is exhibited by the Pavia data set. This is again the result of the large number of water samples that account for almost 50 % of all labeled samples. This leads to large water areas being classified as another class and consequently in a decrease of κ even for small amounts of swapped labels.

The green curve in each plot is the result of a SAM classification on the original data. In contrast to the results in Section 6.3.2, the κ value changes with the variation of the swap percentage. The reference spectra per class for the SAM classification are calculated from the mean spectrum of all class labels, including the swapped labels. The results show only minor changes. Surprisingly, the κ values can even increase with higher swap percentages. This makes sense when looking at the geometry of the SAM classifier and the assembly of reference spectra in $\mathbb{R}^d$. The decision boundary of a single class for the SAM classifier is a



(a) KSC NFN performance for errors in training data

(b) Pavia NFN performance for errors in training data

(c) PaviaU NFN performance for errors in training data

(d) Greding NFN performance for errors in training data

(e) Montelly NFN performance for errors in training data

(f) Prilly NFN performance for errors in training data

(g) Zurich'02 NFN performance for errors in training data

(h) Zurich'06 NFN performance for errors in training data

Figure 6.11: κ statistics for SAM classification on NFN transformed data for variable amounts of swapped training labels.

cone centered around the reference spectrum. With correct training samples, the center of the cone is defined by the mean value of all samples. With a certain amount of swapped labels, the reference spectra are drawn closer to each other due to spectra of other classes skewing the calculation of the mean vector. Depending on the (randomly) assigned wrong labels per class there exist reference spectra that improve the classification with SAM.

Overall, NFN with SAM classification also works for training data contaminated with wrong samples. A strong decrease in classification accuracy can only be observed for $\geq$ 10 % swapped samples in most cases. Depending on the availability of training samples per class, a bigger neighborhood, e.g., $k \geq 10$, successfully mitigates a translation into the wrong direction.

6.4 NFNalign for Data Alignment and Feature Transfer

This section analyzes the performance of NFNalign for data alignment and FT. This is done by performing NFN for two different data sets, the reference X^* and the test data to be aligned X to the same basis B. After the NFN transformation the transformed data sets $\widetilde{X}^*$ and $\widetilde{X}$ are aligned in the common domain $\mathcal{C}$. NFNalign then applies the inverse NFN transformation of X^* to $\widetilde{X}$ to transform it to the original domain of X^*. The individual transformation for each sample is determined through establishing geographic or spectral pixel correspondences (see Section 5.3).
Similar to the evaluation of NFN in Section 6.3 the performance of NFNalign for different parameter variations is discussed. Since NFNalign effectively adds nonlinearities during the inverse transformation from $\mathcal{C}$ to $\mathcal{X}^*$ an SVM classification (see Section 6.2.3) trained on the reference training data L^* is used to evaluate the successful FT. Additionally, the RMSE is calculated for corresponding samples to measure the quality of the alignment. NFNalign is also evaluated when the training data L of X contains wrong samples while L^* is assumed flawless. This simulates a well-curated reference data and a hastily generated ground truth for additional test data. Finally, subsampling and interpolation of the Greding data are done to evaluate the performance of NFNalign for multimodal data alignment, i.e., when the reference and test data have different dimensionality.

6.4.1 Parameter Optimization

The goal of NFNalign is to transform the data set X from its domain $\mathcal{X}$ to the domain $\mathcal{X}^*$ where physical interpretation and further analysis is possible. Desirable properties of $\mathcal{X}^*$ can be tied to its corresponding data set X^*. The Greding1_ref data was chosen to be the reference data set for the whole Grading data set as it is atmospherically corrected to ground reflectance and has the best illumination conditions, i.e., no haze or cloud shadows. Each of the Montelly/Prilly and the Zurich images is used in turn as the reference to evaluate the

FT from their counterpart.

The results are calculated using 10 % of the available ground truth for the reference data set X^* (for the NFN transformation and training of the SVM model) and only 1 % of the ground truth in the test data sets. The training data was determined using the RICS method (see Section 6.2.1).

The discrepancy between the amounts of training data was chosen to simulate the process of aligning multiple newly acquired data sets with scarce ground truth to a well-curated reference, e.g., to compute fast results directly after a new campaign. The evaluation compares the quality of the data alignment in the form of the RMSE for all corresponding samples after alignment. The FT is analyzed by applying a multi-class SVM classification trained on the reference data set to the aligned data.

Table 6.3 contains the RMSE and κ statistics for each data set. It shows the mean RMSE for all corresponding samples before and after NFNalign was computed. The reflectance data set Greding1_ref was used as a reference, and the radiance data of the other data sets were aligned. Since the range is different between the levels of pre-processing, RMSE was calculated from the atmospherically corrected reflectance data to warrant a fair comparison. The results indicate a reduction of RMSE by factors between 4-10. An exception is the Montelly/Prilly results. This can be explained by the calculation of the pixel correspondences. The data sets come from different areas. Thus, correspondences based on geographic location had to be replaced by spectral similarities in the common domain (see Section 5.3.2). This results in the minimal possible alignment error after inversion. However, it is not advised to use this method for data sets where geographical correspondences are possible as comparing two spectra from the same object can give further information about possible changes in spectral features between the data sets.

The SVM results are generally slightly worse when the data is classified in the new domain with its native model. However, the goal was not to improve the classification accuracy, but to successfully transfer data between domains. SVM is only used to evaluate how well NFNalign can transfer characteristic features necessary for classification from one domain

to another, even if the underlying data sets come from different stages in the pre-processing chain like the Greding data. The reduction in κ can be explained by using a model for SVM that was calculated on a reference with good illumination conditions. Thus, the cloud shadow statistics are not well represented in the model. This explains the drop-off for the Greding3_rad data classification for example. Additionally, this effect is also a sign that some spectral features are preserved during NFNalign. An idea of how to improve the preservation of important features is given in Section 7.2.1.

Figure 6.12 shows the progression of NFNalign for different t and k for all tested combinations of data sets. Looking at the ranges of κ reveals that the results are robust in the face of parameter variations. It makes sense that higher values of t produce better results as the data set becomes more similar to the reference. However, especially the Zurich'06 data has the best result for $t = 1$. When looking at the corresponding RMSE progression in Figure 6.13(f) the correlation between small RMSE and high κ values can be seen. Regarding neighborhood size these results are not conclusive, e.g., the Greding data works best using only a single neighbor, while the Zurich'06 data has the best result for $k = 10$.

Figure 6.13 shows the RMSE for the same parameter combinations. Compared to the original RMSE in Table refTab:NFNalign the variations are insignificant as they are orders of magnitude lower. For $k = 1$ the inverse transformation is based on the transformation of the closest pixel. For $k > 1$, the inverse transformation is based on the mean distance between $\widetilde{\boldsymbol{x}}$ and the k closest samples of $\widetilde{X}^*$ in the common space. This distance is by definition greater or equal to the distance to the closest point. When evaluating the SVM classification using the reference training data L^* on the aligned data set X_{align}, values for $k \geq 5$ generally gave better results. For bigger k one sample $\boldsymbol{x}$ becomes more similar to multiple other samples (and not necessarily its assigned correspondence) during the transformation process. However, larger k reduces the effect of noise and outliers.

The power of the penalty function t, however, is difficult to optimize. Besides the results given here, it is important to keep in mind that higher t also reduce the individual features per sample. For $t \to \infty$ the data set X will be identified with the basis B in the common

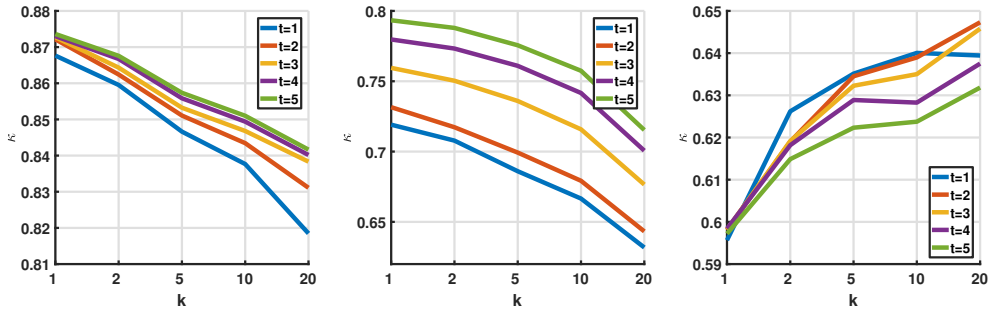
domain. The inversion then maps each sample to its correspondence. This leads to higher κ for larger t simply because the test data set becomes identical to the reference in the common domain. In this case, all individual characteristic features are lost in the process. Thus, small t are preferred for preserving characteristic features during NFNalign. If the result using the SVM trained on the reference is worse, it is a sign that the training data does not encompass all necessary features.

Overall, the results show that NFNalign successfully performs data alignment and FT. It allows the use of an already trained SVM model to classify data sets from different (but related) domains. This is especially impressive when considering the Greding data, where radiance data were aligned to reflectance data without any additional information about an underlying physical model, e.g., for atmospheric correction. NFNalign also improves the RMSE in all test cases. To visualize the initial situation and the results of the NFNalign calculation Figure 6.14 shows mosaics built from the original Greding1-3 reflectance data in Figure 6.14(a) and the Greding1_ref data aligned to Greding2-3_rad data in Figure 6.14(b). The original data depict strong illumination differences on top of the individual effects like cloud shadows and angular effects. After NFNalign was executed, the borders between the data sets are only visible in areas where the co-registration is not perfect, e.g., at the border between the grass and soil area in the bottom left corner. Figure 6.14(c) shows the RMSE per pixel for the original Greding reflectance data while Figure 6.14(d) shows the results after NFNalign on the same scale. The biggest errors appear in areas between two classes and when objects like cars etc. were present in one scene, but not in another.

6.4.2 Robustness

In this section, the robustness of NFNalign is evaluated. Two experiments are performed on the Greding data set.

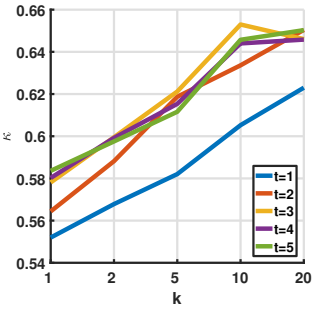
Increase of the noise level: To simulate a sensor with a smaller SNR or worse overall illumination conditions varying degrees of Gaussian white noise are added to the test data sets before alignment.



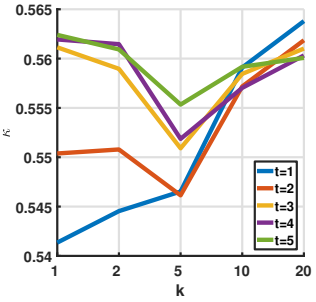
(a) SVM results for NFNalign of Greding2_rad to Greding1_ref data

(b) SVM results for NFNalign of Greding3_rad to Greding1_ref data

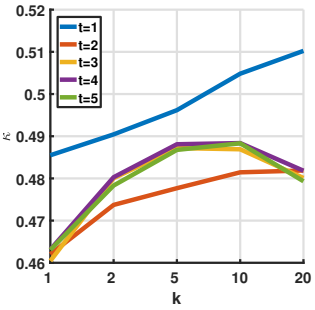
(c) SVM results for NFNalign of Montelly to Prilly data



(d) SVM results for NFNalign of Prilly to Montelly data

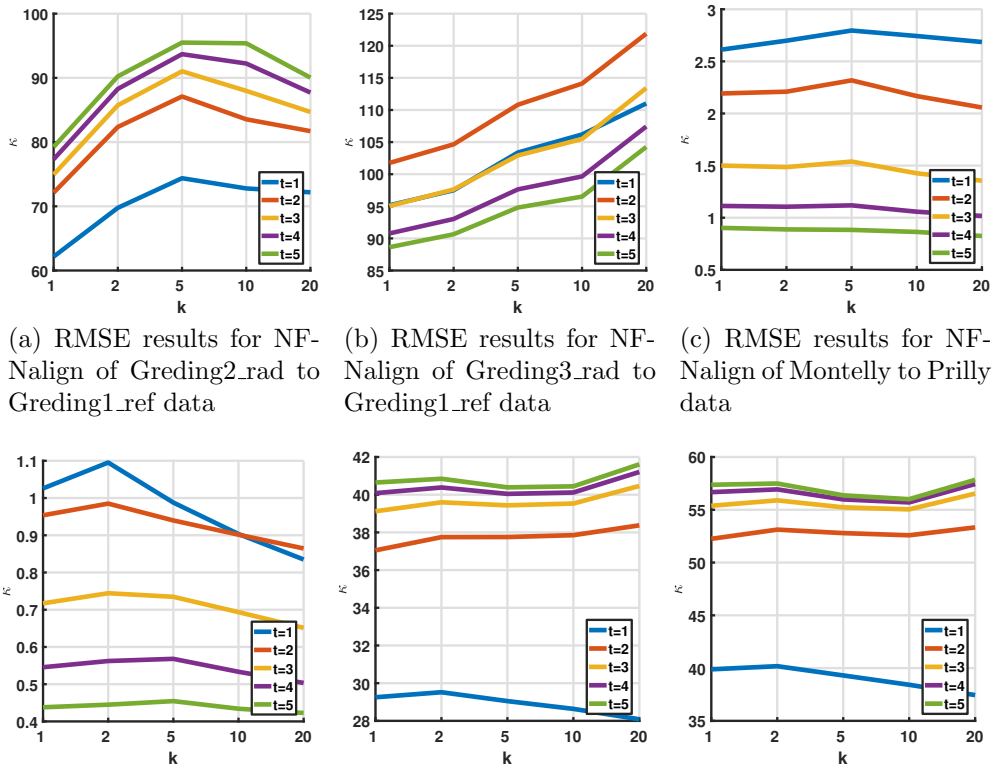


(e) SVM results for NFNalign of Zurich'02 to Zurich'06 data



(f) SVM results for NFNalign of Zurich'06 to Zurich'02 data

Figure 6.12: κ statistics for SVM classification of NFNaligned data for variable amounts of NNs k.



(a) RMSE results for NF-Nalign of Greding2_rad to Greding1_ref data

(b) RMSE results for NF-Nalign of Greding3_rad to Greding1_ref data

(c) RMSE results for NF-Nalign of Montelly to Prilly data

(d) RMSE results for NF-Nalign of Prilly to Montelly data

(e) RMSE results for NF-Nalign of Zurich'02 to Zurich'06 data

(f) RMSE results for NF-Nalign of Zurich'06 to Zurich'02 data

Figure 6.13: Mean RMSE of corresponding samples after data alignment with NFNaligne for variable amounts of NNs k.

Table 6.3: Comparison of RMSE and SVM classification results. The RMSE is calculated as the mean RMSE of all corresponding samples. $SVM_{NFNalign}$ is computed using the SVM model trained exclusively on the training data in the reference domain. NFNalign was calculated with 10 % available ground truth per class for L^* and 1 % for L. SVM_{orig} is computed on the original test data X with the specified amount of training data.

	$RMSE_{orig}$	$RMSE_{align}$ 10 % L^*, 1% L	κ SVM_{orig} 1 % training	κ $SVM_{NFNalign}$ 10 % L^*, 1% L
Greding2_rad	273.4	79.2	0.883	0.874
Greding3_rad	583.6	88.6	0.926	0.793
Montelly	95.1	2.1	0.629	0.647
Prilly	114.2	0.7	0.671	0.654
Zurich'02	139.8	28.1	0.578	0.564
Zurich'06	139.8	37.4	0.572	0.510

Wrong labels: Similar to Section 6.3.3 varying percentages of the training samples in the test data per class are randomly assigned a different class label.

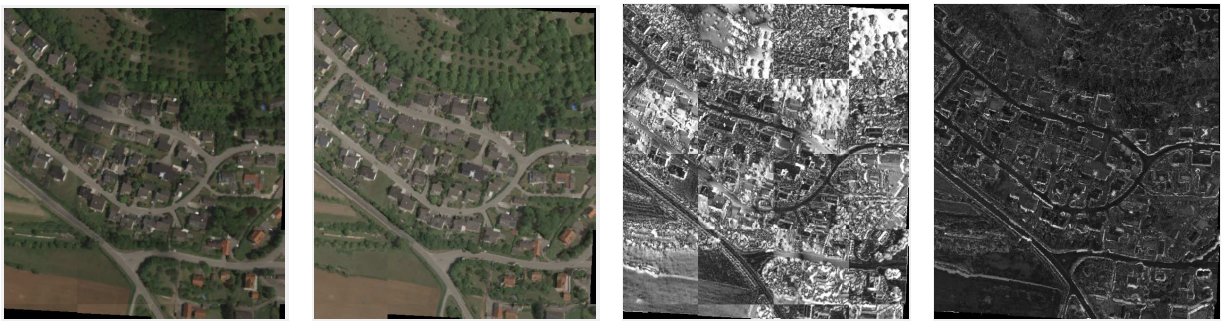
These experiments again follow the assumption that the reference data represents the ideal condition and has been extensively preprocessed to reach this status. Thus, only the test data and its ground truth are degraded.

To evaluate the robustness of NFNalign when the data is compromised by noise the following model is used for artificially degrading the test data set X. The amount of Gaussian noise is determined by the standard deviation per band over the whole scene. Each sample $\boldsymbol{x}$ of the test data is then modified by

$$\boldsymbol{x}_{noise} = \boldsymbol{x} + \alpha \sqrt{\frac{\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})^2}{n-1}}, \tag{6.9}$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ stands for the vector of mean values per band and $\alpha = 0.02, 0.04 \ldots, 0.2$ is the noise scaling factor. Fig. 6.15 shows the influence of noise in Greding2/3_rad data for NFNalign.

The original variance in each band is already high as the scene depicts multiple classes



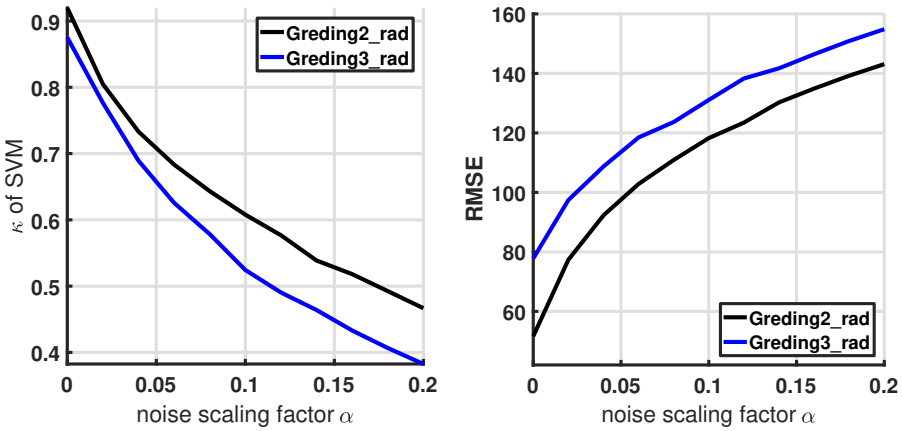
(a) RGB mosaic of Greding reflectance data

(b) RGB mosaic of Greding1_ref and NFNalign Greding2_rad & Greding3_rad

(c) RMSE mosaic of Greding reflectance data, scaled to RMSE values $\in [0, 400]$.

(d) RMSE mosaic of Greding1_ref and NFNalign Greding2_rad & Greding3_rad, scaled to RMSE values $\in [0, 400]$.

Figure 6.14: Mosaics of original and NFNalign transformed Greding data allow visual evaluation of improvement of illumination and feature adjustment. Most notable errors after NFNalign transformation can be found in areas between two classes and on moving objects. Residual errors from cloud correction are not noticeable.



(a) Progression of κ with increasing noise variance

(b) Progression of RMSE with increasing noise variance

Figure 6.15: Results after the test data was contaminated by varying degrees of Gaussian noise. The noise variance was calculated as the standard deviation per band and adjusted with a factor for different noise levels.

containing dark materials like Residential and Road, but also bright areas like Meadow and Residential (red). The RMSE increases and the classification accuracy drops with the increasing noise level. NFNalign can handle small amounts of noise, but it appears that for higher noise the training data becomes insufficient. Also, the risk of violating the class uniqueness assumption from Section 4.1 increases.

The other aspect of this section deals with the difficulties in ground truth data selection. Generating accurate ground truth can be very time-consuming. Thus, the performance of NFNalign is evaluated when the training data contains wrong labels similar to Section 6.3.3. For this, NFNalign is performed using 10 % labeled samples in reference and test data, a power of $t = 4$ of the penalty function, and a varying amount of training samples are randomly swapped to another training set in the test data. The reference ground truth is again considered flawless.

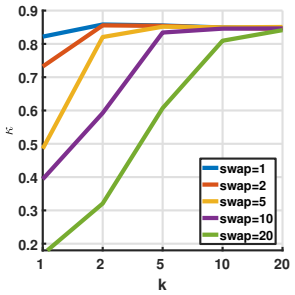
The results are given in Table 6.4. The experiments were carried out for swap percentages of $1, 2, 5, 10$, and 20 percent per class. Larger neighborhoods can mitigate the effect of wrong labels for the alignment of Greding2_rad almost completely and with only a minor decrease in κ for Greding3_rad data. This makes sense as on average most of the neighbors belong to the correct class. A few wrong labels are easier to compensate with a larger neighborhood. The same holds for the RMSE. The results here don't show the characteristic drop-off for higher swap percentages that could be observed for the similar experiment regarding NFN in Section 6.3.3. This can be explained by the accurate correspondences between the data sets. This means that the inversion to the reference domain with a sample of the correct label is sufficient for FT. This emphasizes the requirement of carefully calculated correspondences and requires further investigation in the future.

6.5 Comparison to other approaches

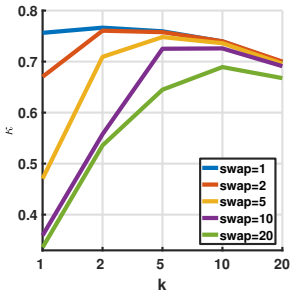
Comparing transfer learning with NFNalign to other methods is difficult, as neither of the commonly available codes offers a functioning inversion to the reference domain. For that

Table 6.4: RMS and κ of SVM classification after NFNalign transformation for variable amount of swapped training labels per class.

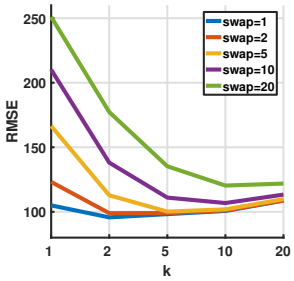
	Greding2_rad		Greding3_rad	
swap %	RMSE	κ SVM	RMSE	κ SVM
1	94.3	0.858	95.7	0.767
2	98.8	0.856	98.9	0.761
5	92.5	0.854	100.2	0.748
10	87.0	0.846	106.8	0.726
20	90.3	0.841	120.3	0.689



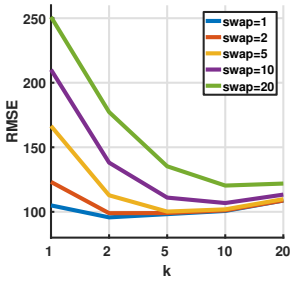
(a) SVM results for NFNalign of Greding2_rad to Greding1_ref data with variable amount of wrong training data



(b) SVM results for NFNalign of Greding3_rad to Greding1_ref data with variable amount of wrong training data



(c) RMSE results for NFNalign of Greding2_rad to Greding1_ref data with variable amount of wrong training data



(d) RMSE results for NFNalign of Greding3_rad to Greding1_ref data with variable amount of wrong training data

Figure 6.16: SVM results and RMSE for NFNalign with variable amounts of swapped training labels.

reason, it is only possible to compare the performance of different methods in the common domain. NFNalign is compared to the GFK [42]) algorithm and the KEMA [119]). Additionally, two other approaches were tested that are not directly related to MA or FT, namely *Histogram Matching* (HM) and *Individual Pixel Re-scaling* (IPR).
These approaches produce good results but have their limitations. Only NFNalign was capable of transforming radiance to reflectance data while maintaining physical interpretability of the results.

6.5.1 Histogram Matching

HM between corresponding bands of reference and test data can be used to equalize the image statistics [99]. The reflectance data is used as the reference to adjust the radiance statistics. This is performed for each band individually. Mean value and variance of the test and the reference data are equalized allowing the transfer of the SVM model. This approach does not differentiate locally, meaning that shadow areas before HM will still be darker than their surrounding areas after the procedure.
The results for different bin numbers are printed in Table 6.5. Increasing the number of bins refines the alignment until it converges for the given test case around 10000 bins. Surprisingly, the RMSE becomes worse after HM is applied. Even for a large number of bins, the original RMSE of 273.4 for Greding2_ref and 583.6 for Greding3_ref is slightly better than the HM result. Classification of Greding2-3_ref data using the SVM model of the Greding1_ref data does not work. After HM the SVM classification results are considerably worse than the results after applying NFNalign. Overall, HM can be used for FT, but data alignment is not improved in that process.

6.5.2 Individual Pixel Re-scaling

When two data sets show different effects after correction to reflectance, like the Greding data including cloud shadows, adjusting the magnitude of individual samples can be done by multiplying the spectrum with a scalar value. This was done in Section 2.3.2 to mitigate

Table 6.5: RMSE and κ of SVM classification after performing HM for different number of bins.

		# bins					
		original	200	500	1000	10000	50000
Greding2_ref	RMSE	273.4	309.7	289.1	285.8	284.7	284.7
	$\kappa\ SVM_{10\%}$	0.002	0.337	0.662	0.696	0.705	0.704
Greding3_ref	RMSE	583.6	603.4	592.3	590.6	590.1	590.1
	$\kappa\ SVM_{10\%}$	0.003	0.388	0.592	0.611	0.640	0.640

the effect of shadows. Instead of applying a constant scaling factor to a set number of samples given by the shadow mask, it is possible to exploit the pixel correspondences to calculate individual scaling factors for corresponding samples. Two things are important to remember. First, the linear scaling of radiance to reflectance is not possible due to atmospheric absorptions. This means the experiment had to be restricted to data from the same domain. In this case, the Greding reflectance data was used. For other data sets the assumption that at least a crude pre-processing was performed. The second problem is that linear re-scaling does not correct nonlinearities. This can lead to re-scaled samples outside the trained model that can not be classified correctly.

Table 6.6 shows the RMSE and the κ of the SVM classification for IPR and NFNalign. Compared to the original RMSE the alignment by IPR is better by a factor of$\approx 3-5$. The SVM classification using a model learned on the training data of the reference data set is very similar to the classification of the NFNaligned data. The results are almost identical for Greding2 and a little worse for Greding3. This can be explained by the large shadow area that is not accurately classified in the IPR case due to its nonlinear distribution.

Overall, it is important to remember that IPR had to be calculated on the Greding2-3_ref reflectance data. This means for IPR to be viable at least a crude pre-processing of all data sets to the same domain is required. The NFNalign results were calculated by aligning radiance to reflectance data. This requires only the reference to be from the desired domain.

Table 6.6: RMSE and κ of SVM classification after IPR of Greding reflectance data and comparison to NFNalign using radiance data.

	RMSE of IPR	RMSE of NFNalign	$\kappa\ SVM_{10\%}$ of IPR	$\kappa\ SVM_{10\%}$ of NFNalign
Greding2	98.0	51.7	0.901	0.920
Greding3	119.9	77.8	0.833	0.875

6.5.3 Geodesic Flow Kernel

The GFK approach for MA uses the kernel trick to align multiple data sets in a high-dimensional space with explicitly defined dimensionality. It is described in Section 3.3.3 and more details can be found in [42, 41]. An advantage of GFK is that no pixel correspondences are required to calculate the alignment. Unfortunately, it is not possible to invert the results back to the domain of the reference data set. There exists no universally valid framework for the inversion and applying a simple approach based on the pseudo-inverse of the transformation matrix does not produce results resembling spectral signatures. This is likely due to the pre-image problem (see Section 3.4, [69, 132]). Therefore, physical interpretation of spectral features in the common domain is not possible.

The MATLAB code used for this experiment is part of the 'A domain adaptation toolbox', which also contains kernel-PCA and other useful methods introduced in Chapter 3.[3].

Table 6.7 shows the results of GFK alignment for the Greding2-3_rad data to Greding1_ref and its comparison to NFNalign. Since the alignment happens in a space of different dimensionality, it is not possible to compare the alignment error in terms of RMSE. The results of the SVM classification are slightly better than the IPR approach. This is especially valuable since radiance data was used as test data in the GFK calculation. Although it was mentioned in [119] that GFK has problems with nonlinear effects, this is not represented in the results here. However, NFNalign results are still superior regarding κ while simultaneously allowing the physical interpretation of the aligned samples.

Overall, GFK is useful for classification, especially when no pixel correspondences are avail-

[3]The toolbox can be downloaded at `https://de.mathworks.com/matlabcentral/fileexchange/56704-a-domain-adaptation-toolbox`

Table 6.7: RMSE and κ of SVM classification after GFK alignment of Greding radiance and reflectance data, and comparison to NFNalign.

	RMSE of GFK	RMSE of NFNalign	$\kappa\ SVM_{10\%}$ of GFK	$\kappa\ SVM_{10\%}$ of NFNalign
Greding2	-	51.7	0.861	0.920
Greding3	-	77.8	0.861	0.875

able, e.g., for data sets of different areas. In this case, NFNalign requires the calculation of spectral correspondences which can be very time-consuming.

6.5.4 Kernel Manifold Alignment

The KEMA is the extension of the Semi-Supervised MA with a suitable kernel. This method allows data of different dimensionalities and from different domains to be aligned in a high-dimensional common space. This approach was shortly introduced in Section 3.3.5, further information can be found in the corresponding papers [119, 125]. Similar to GFK this approach does not require pixel correspondences. An advantage over GFK is the automatic calculation of a suitable dimensionality for the common space and the ability to deal with strong nonlinearities. Additionally, KEMA can align data from different domains, regardless of their dimensionality. It also works well even when the data contain strong nonlinearities.[4] The inversion from the common domain to the target domain is not included. Following the approach from the paper to calculate the inverse projection matrix and applying it to the aligned source data does not yield results that can be physically interpreted. Thus, similar to the GFK experiment, no valid RMSE can be given.

Table 6.8 shows the results of GFK alignment for the Greding2-3_rad data to Greding1_ref and its comparison to NFNalign. The SVM classification results are slightly superior to the results calculated on NFNaligned data. However, no physical interpretation of the aligned spectra in the common domain is possible, and inversion of the projection did not produce acceptable results.

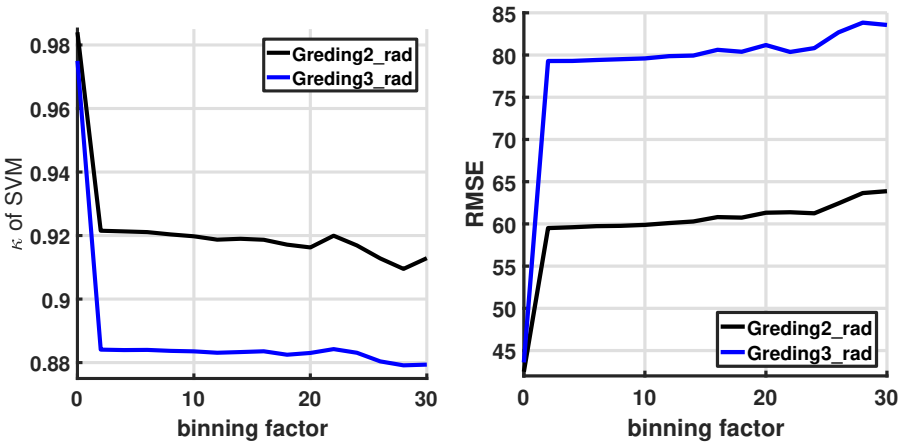
[4]The MATLAB code used for this experiment can be downloaded at `https://github.com/dtuia/KEMA`.

Table 6.8: RMSE and κ of SVM classification after KEMA alignment of Greding radiance and reflectance data, and comparison to NFNalign.

	RMSE of KEMA	RMSE of NFNalign	$\kappa\ SVM_{10\%}$ of KEMA	$\kappa\ SVM_{10\%}$ of NFNalign
Greding2	-	51.7	0.921	0.920
Greding3	-	77.8	0.882	0.875

6.5.5 Multimodal Data Alignment

The capability of KEMA to align data sets of different dimensionalities is a powerful asset when different sensors are used in conjunction. NFNalign is not inherently capable of handling data sets with different dimensions due to the alignment in the common space of the same dimensionality. Otherwise, it would have to deal with an ill-posed problem during the inversion procedure as the other MA approaches from Chapter 3. In this section, the adaptability of NFNalign to data sets of different dimensionality via interpolation is evaluated.
This approach can be applied to data recorded with two sensors that have different parameters like spatial and spectral resolution, different SNR, etc. However, this experiment is limited to simulated spectral subsets calculated via binning of neighboring bands. The Greding data set was selected for this purpose as the available data sets from different sensors neither share the same class labels nor recording location.
Greding1_ref is used as the reference X^* and remains unchanged. Greding2_rad and Greding3_rad are spectrally binned by a factor of $2, 4, \ldots, 30$ to compute the test data X. The first step to perform NFNalign on lower dimensional data X, L is to increase the dimensionality reasonably. For this, the test data X and its corresponding training data L are linearly interpolated. The center wavelengths of X^* are used as sampling points to generate the test data $\widehat{X}$ and training data $\widehat{L}$ with the same dimension as X^*. After that, NFNalign is calculated on $\widehat{X}$ and RMSE, as well as SVM classification results, are computed on the aligned data. Again, only the reference training data L^* is used to train the SVM model. RMSE and κ statistics for the SVM are plotted as a function of the binning factor in Figure 6.17. Both RMSE and κ follow the expected progression after the initial drop-off in performance going



(a) Progression of κ with increasing binning factor

(b) Progression of RMSE with increasing binning factor

Figure 6.17: Spectral binning followed by linear interpolation affects neither RMSE nor κ significantly for the Greding data.

from the original data to a binning by 2. With increasing binning factor RMSE increases and κ decreases. However, the absolute difference in classification accuracy between a binning factor of 2 and 30 is less than 0.02, and for RMSE it is approximately 6.

This result indicates that the alignment of multimodal data is theoretically possible when NFNalign is combined with linear interpolation. However, it is difficult to estimate the practical performance on real multimodal data, e.g., recorded by different sensors over the same area without actual data. A likely situation where this approach will fail is when the training data of the reference hyperspectral data set contains several very similar classes. If the dimensionality of the test data is too low to sample the required features inherently, interpolation will not work to generate them. However, this poses more of a mathematical limitation than a flaw of the method.

Chapter 7

Conclusion and Future Work

This thesis has investigated different approaches for data alignment and FT between domains. Researched methods include ML and MA algorithms, reduction of individual nonlinear effects and data pre-processing to convert multiple data sets to a common domain for evaluation. The main contribution is the NFNalign method. It constitutes a complete framework for data alignment and FT that uses training data affiliated with different classes to first align the data sets to a common basis and then invert the translation to the target domain using pixel correspondences. The NFNalign algorithm calculates an alignment of data sets in a domain that allows physical interpretation. The quality of the FT could be shown by applying SVM classification trained on one data set to another after alignment without loss of accuracy. This chapter summarizes the contributions of this thesis and suggests future research topics based on the lessons learned.

7.1 Summary of Contributions

The contributions of this thesis can be divided into two main areas. The NFN algorithm, introduced in Chapter 4, transforms a data set to a new arbitrary basis and successfully mitigates nonlinearities in the process. The NFNalign algorithm, discussed in Chapter 5, combines multiple NFN calculations and its inversion to transform whole data sets from

their original domain to a target domain.

The NFN is a data-driven approach for mitigating nonlinear effects in hyperspectral data that only requires some training samples per class in the data set. This algorithm allows the correction of nonlinear effects including changes in illumination, hard and partial shadows, and transmission/multiple reflections by objects in the scene as well as anisotropic effects for 3D objects.

Usually, these effects are dealt with separately by using complex physical models that require additional information. The transformation is defined by a set of training samples per class. If the training data represents a good sampling of the high dimensional manifold for each class, the transformation is invariant to the underlying physical effects leading to the nonlinear behavior. It has been shown that the new basis defined by one reference vector per class can be arbitrarily chosen. This approach can be used for dimensionality reduction by selecting the first p canonical unit vectors or to transform a radiance data set to reflectance using spectrally subsampled laboratory spectra. The successful mitigation was shown by comparing SAM classification before and after NFN transformation with a considerable increase in classification accuracy for the transformed data. A major advantage over the correction of individual effects is the ease of use. Manually selecting small training areas per class including potential areas with nonlinearities, e.g., slanted roofs, objects in the shadow of a tree, etc., is enough to produce good results. Only two parameters are required in the present form, one for the strictness of the penalty function and one for the robustness towards errors in the training data. The experiments have shown that good results are achieved almost regardless of parameter selection and optimization improves the outcome only by a few percents. The robustness of the NFN algorithm when the training data is contaminated with samples from other classes was tested, and even errors of 20 % could be dealt with by increasing the neighborhood size. Additionally, NFN can be computed very fast, scales linearly with image size and could even be optimized for the real-time application. In contrast to other MA approaches no global adjacency matrices are calculated alleviating the restriction to smaller data sets or specific memory management techniques. It could be

shown that NFN successfully mitigates nonlinearities for all tested data sets by comparing the SAM classification results before and after the transformation. While the NFN algorithm was able to improve linear classification for all tested data sets, it remains to be analyzed if it also performs well when classes have very similar spectral signatures or classification relies on distinct features that only constitute to a couple of bands.

The arbitrary basis selection and the mitigation of nonlinearities are the two important features that have lead to the development of the NFNalign algorithm. The former allows the transformation of multiple data sets to the same basis in a common domain while the latter guarantees that the data structure in the common domain is similar up to range for all data sets, i.e., radiance data in [116383] and reflectance data in [01]. By calculating correspondences between samples of different data sets, it is possible to directly apply the inverse translation of a sample in one data set to its correspondence in another data set. This inversion transforms the samples from the common domain to the original domain of its correspondence. Practical approaches for calculating the correspondences between co-registered and between spatially disconnected data sets are discussed in Section 5.3. They were evaluated successfully on data sets of the same geographic location as well as spatially disjoint data sets. Since the dimension of the domains is not changed during the transformation, everything is calculated analytically. In contrast to other MA approaches, this has the benefit that no ill-posed problem has to be solved for the inversion. Since the dimensionality of the data is not changed by NFNalign, physical interpretability of the aligned data is possible by using the features from the reference domain. The algorithm was successfully applied to data pre-processing by aligning radiance data to a reference reflectance data set, skipping the model-based atmospheric correction completely. The quality of the alignment was evaluated by classifying the aligned data with an SVM model learned on the reference data alone. The results are comparable to classifying the data in their original domains, meaning FT was successful. The RMSE before and after the transformation with NFNalign were compared and show a highly improved conformity between the aligned data sets. Comparing NFNalign to MA approaches revealed that data alignment in the

reference domain was not possible without further consideration of the pre-image problem. Thus, only FT to a common domain could be evaluated. The results were similar to the results from NFNalign. However, physical interpretability was completely lost. The only other approach that produced acceptable results was an individual rescaling of the samples using the correspondences between samples. The results were slightly worse compared to NFNalign. Finally, the alignment of data sets with different dimensionalities was tested by linearly interpolating the missing bands from their neighbors. The results show that even an interpolation from a five-band image to a 127 band image is possible.

7.2 Future Works

Although NFN and NFNalign produce very good results for all tested data sets, there are still some characteristics that need further research or improvement. The most important characteristic is the preservation of features during the alignment process. Spectral features can be specific for a given material, caused by unwanted effects or be a sign of a physical process that can be analyzed to gain new insight. Usually, only the unwanted effects should be removed during the transformation. Another important topic is the training data selection. Some methods can simplify the selection or at least support the researcher in the process. Finally, one interesting property of the NFN algorithm is the calculation of the weights for each individual translation step. This intermediate result can potentially be used to compute a nonlinear spectral unmixing of the data set.

7.2.1 Specific Feature Preservation

This topic is not discussed in any of the MA papers as the goal is usually to minimize the alignment error under specific deformations. However, certain effects may not be caused by unwanted nonlinear effects but rather by natural changes. An example for these changes are vegetation parameters that can hint at plant stress due to malnutrition or pest infestation. The preservation of these features is an important topic for precision farming. Losing them

during the alignment must be prevented.

The results form Chapter 6 indicate that NFN and NFNalign at least reduce the spectral features to mitigate the nonlinearities or to align the test data to a given reference. However, finding ways to determine whether a feature is simply caused by nonlinearities or if it could be useful to preserve during the transformation can boost the usefulness of NFNalign.

Using the Euclidean distance between a sample and the training data per class during the NFN transformation mitigates the discrepancy evenly for all bands. In theory, while the nonlinearities in the data are multidimensional, it is generally assumed that they each have a low-dimensional embedding in the original domain of the data set [125, 123, 119].

A possible approach to feature preservation is to change the penalty function and the translation of the samples from uniform operations to a band- or feature-dependent operation, e.g., by customizable band weights. Thus, specific features can be targeted and preserved during the computation.

A specific known feature can be preserved by simply reducing the translation in the bands where the feature is present to zero. If this is done, it will affect all other translations as well, even if that feature is not present. Since it is unlikely that all important features are known a priori, a less absolute approach is required.

Another idea is to compute the PCA of the training data. By definition, the magnitude of the eigenvalues determines the encoded variance in that specific direction. These vectors point in the direction of highest variance as a sample that can be encoded by the eigenvectors belonging to the largest eigenvalues is collinear to the nonlinear training data. Such a vector has a high likelihood of containing only redundant information. Thus, little information is lost during the NFN transformation. If a sample cannot be encoded by the largest eigenvectors, this means that it has at least some components orthogonal to the principal direction of the variance of the training data. By performing NFN with a too strict penalty function, this sample would collapse into one of the classes and the orthogonal part would be lost, or at least severely diminished.

During the transformation, the goal is then to perform the translation only along the di-

rection of encoded variances while simultaneously suppress the translation in an orthogonal direction. The exact procedure has to be researched and contains many different components, e.g., choosing the correct number of significant eigenvectors, introducing another penalty function based on the orthogonal distance, etc.
If the feature preservation works for the NFN transformation, it will automatically transition to the NFNalign algorithm as well. It was already shown in Section 5.4.1 that the inversion in NFNalign retains the distance (and direction) to the corresponding reference sample in the common domain after the inversion due to parallel shift.

7.2.2 Active Learning for Training Data Selection

Manual selection of comprehensive ground truth is time-consuming and error-prone. Using automatic approaches for training data selection can be beneficial. Possible solutions include segmentation followed by manual refinement or (semi-)supervised learning algorithms [33, 13]. A simple approach is to perform segmentation with a high number of classes. A common effect is that the same material is assigned different labels based on its location and state due to nonlinearities, e.g., meadow in the sun and meadow in the shadow of a tree. This intermediate results can be used to switch from a pixel-based ground truth selection to a region based selection by identifying multiple labels with a single material. Supervised Learning approaches can be used to do this iteratively. They have the benefit that they usually give out a hint about image regions that are not yet represented properly in the labeled data by comparing the discrepancy between individual regions and a selected model. Thus, selection of proper training data can be supported.

7.2.3 Spectral Unmixing with NFN

Spectral Unmixing is a technique that assumes all spectra in a hyperspectral data set are composed of mixtures by pure materials, so-called endmembers. These endmembers are the components that are mixed following a specific model to build the spectral signatures. For

most data sets the linear mixture model, which tries to reconstruct the data with linear combinations of endmembers and corresponding weights, is used [10]. Data sets with strong nonlinearities, however, require more complex models. One approach is to use multiple endmembers per pure material to encode the nonlinearities [59]. This is similar to the idea behind NFN to use training samples per class for the transformation. During the calculation of the NFN, one step is to determine the distance to the nearest training sample(s) per class. Normalizing these distances with $\|\cdot\|_1$ gives a ratio of the distance to the classes. This can also be understood as material similarity. It is difficult to determine whether this hypothesis holds as no established benchmark data is available at the moment.

If this theory holds, it can lead to a refined version of the NFN algorithm where the penalty function is adapted to the abundances, i.e., only transform samples where the number of significant abundances is high. Small abundances are usually a sign that an endmember was included to minimize the error function of the unmixing model and large abundances in more than three endmembers are geometrically unlikely [146].

Similar to the requirements for the NFN transformation gaps in the training data can lead to errors in the unmixing result. A first test was performed on a data set of mixed rock and mineral mixtures with known abundances [48]. However, only the pure materials were available for training, and consequently, abundances between, e.g., the dark basalt and the bright rhyolite were not representative of the actual mixture coefficients. To alleviate this problem where the training data does not contain a lot of nonlinear variations principal curves [27] could be used to model the assumed trend. This is, however, a difficult topic as the general behavior of different nonlinear effects are not known. Using it to bridge small gaps might be possible, but completely model the behavior from only a few samples that are close to each other is ambitious.

Appendices

Appendix A

Comprehensive Listing of Data Set Information

Table A.1: Ground truth information of Greding1_rad and Greding1_ref data

class	samples	color
Dark roof	21307	ocher
Red roof	2749	red
Asphalt	21386	gray
Soil	23978	brown
Grass	25870	green
Tree	32398	dark green
Total	127688	

Table A.2: Ground truth information of Greding2_rad and Greding2_ref data

class	samples	color
Dark roof	18421	ocher
Red roof	2193	red
Asphalt	17110	gray
Soil	19841	brown
Grass	21636	green
Tree	25072	dark green
Total	104273	

Table A.3: Ground truth information of Greding3_rad and Greding3_ref data

class	samples	color
Dark roof	18015	ocher
Red roof	2543	red
Asphalt	17008	gray
Soil	20235	brown
Grass	21877	green
Tree	25977	dark green
Total	105655	

Table A.4: Ground truth information of Pavia City Center data

class	samples	color
Water	65971	blue
Tree	7598	dark green
Asphalt	3090	gray
Self-blocking bricks	2685	ocher
Bitumen	6584	red
Tiles	9248	pink
Shadow	7287	light blue
Meadow	42826	green
Bare soil	2863	brown
Total	148152	

Table A.5: Ground truth information of Pavia University data

class	samples	color
Asphalt	6631	gray
Meadow	18649	green
Gravel	2099	yellow
Tree	3064	dark green
Painted metal sheets	1345	pink
Bare soil	5029	brown
Bitumen	1330	red
Self-blocking bricks	3682	ocher
Shadow	947	light blue
Total	42776	

Table A.6: Ground truth information of Kennedy Space Center data

class	samples	color
Scrub	761	ocher
Willow swamp	243	medium blue
Cabbage palm hammock	256	green
Cabbage palm / oak hammock	252	light green
Slash pine	161	dark green
Oak / broadleaf hammock	229	yellow
Hardwood swamp	105	light blue
Graminoid marsh	431	pink
Spartina marsh	520	medium pink
Cattail marsh	404	light pink
Salt marsh	419	red
Mud flats	503	brown
Water	927	blue
Total	5211	

Table A.7: Ground truth information of Montelly/Prilly data; only classes common to both images are used.

class	Montelly samples	Prilly samples	color
Residential	38725	29907	ocher
Meadow	47865	148604	green
Tree	177203	116343	darkgreen
Road	104582	141353	gray
Shadow	218189	39404	light blue
Commercial	140970	121364	red
Total	727534	596975	

Table A.8: Ground truth information of Zurich'02/Zurich'06 data

class	Zurich'02 samples	Zurich'06 samples	color
Residential	6746	3837	ocher
Commercial	5277	4992	red
Vegetation	13123	7642	green
Harvested Vegetation	2523	2883	brown
Gravel	3822	1408	light pink
Road	6158	4365	gray
Sand	269	165	yellow
Parking lot	1749	1671	medium pink
Water	1095	1812	blue
Total	40762	28775	

Appendix B

Linearization results with accurate training data

In this section the classification results of the linear SAM with and without SIMA transformation to get rid of nonlinear effects are compared. Additionally, linear ACE and nonlinear SVM results are given. The range for the power of the penalty function was $1, \ldots, 5$, while the tested numbers of nearest neighbors were $k = [1, 2, 5, 10, 20]$. Only the best result in terms of Cohen's κ is printed with the corresponding parameter combination for training percentages $pct = [1, 2, 5, 10, 20, 50]$.

Table B.1: Greding1_rad

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.928	0.744	0.968	0.966
2	5	2	0.921	0.790	0.974	0.966
5	5	2	0.925	0.809	0.983	0.974
10	5	2	0.924	0.820	0.986	0.978
20	5	2	0.925	0.823	0.989	0.981
50	5	2	0.924	0.824	0.994	0.988

Table B.2: Greding1_refl

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.907	0.803	0.960	0.962
2	5	5	0.903	0.840	0.965	0.963
5	5	5	0.906	0.859	0.973	0.969
10	5	2	0.905	0.868	0.978	0.974
20	5	2	0.905	0.871	0.982	0.977
50	5	2	0.905	0.872	0.986	0.985

Table B.3: Greding2_rad

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.924	0.782	0.976	0.969
2	5	2	0.926	0.814	0.978	0.972
5	5	2	0.922	0.827	0.985	0.978
10	5	2	0.923	0.838	0.988	0.981
20	5	2	0.921	0.840	0.991	0.984
50	5	2	0.922	0.842	0.995	0.990

Table B.4: Greding2_refl

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.912	0.836	0.966	0.963
2	5	2	0.913	0.859	0.963	0.967
5	5	2	0.911	0.881	0.977	0.974
10	5	2	0.911	0.894	0.981	0.977
20	5	2	0.910	0.898	0.985	0.980
50	5	2	0.911	0.899	0.989	0.986

Table B.5: Greding3_rad

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.893	0.798	0.979	0.966
2	5	1	0.899	0.811	0.981	0.970
5	5	2	0.895	0.826	0.988	0.978
10	5	2	0.895	0.829	0.991	0.981
20	5	2	0.895	0.829	0.994	0.985
50	5	2	0.895	0.831	0.996	0.990

Table B.6: Greding3_refl

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.891	0.787	0.972	0.959
2	5	2	0.896	0.833	0.976	0.965
5	5	2	0.893	0.857	0.982	0.974
10	5	2	0.893	0.864	0.984	0.977
20	5	2	0.892	0.867	0.987	0.981
50	5	2	0.893	0.870	0.991	0.988

Table B.7: Kennedy Space Center

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	1	0.567	0.430	0.675	0.604
2	5	1	0.636	0.520	0.768	0.683
5	5	1	0.591	0.679	0.843	0.760
10	5	2	0.479	0.748	0.871	0.787
20	5	2	0.420	0.798	0.906	0.828
50	5	2	0.376	0.828	0.930	0.827

Table B.8: Pavia City

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.887	0.800	0.946	0.955
2	5	2	0.877	0.831	0.952	0.961
5	5	2	0.881	0.859	0.965	0.965
10	5	2	0.883	0.875	0.972	0.970
20	5	5	0.882	0.886	0.975	0.973
50	5	5	0.881	0.888	0.981	0.975

Table B.9: Pavia University

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.435	0.318	0.763	0.713
2	5	2	0.475	0.396	0.787	0.776
5	5	2	0.479	0.470	0.831	0.807
10	5	2	0.479	0.510	0.852	0.830
20	5	5	0.478	0.525	0.870	0.845
50	5	2	0.482	0.548	0.885	0.866

Table B.10: Montelly

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	20	0.429	0.459	0.722	0.651
2	5	20	0.418	0.457	0.727	0.654
5	5	20	0.420	0.463	0.736	0.657
10	5	10	0.425	0.467	0.739	0.664
20	5	10	0.424	0.465	0.742	0.669
50	5	5	0.424	0.465	0.746	0.682

Table B.11: Prilly

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	10	0.581	0.419	0.764	0.711
2	5	10	0.571	0.412	0.769	0.734
5	5	10	0.575	0.413	0.773	0.734
10	5	5	0.574	0.415	0.775	0.741
20	5	5	0.575	0.415	0.777	0.751
50	5	5	0.575	0.415	0.779	0.767

Table B.12: Zurich'02

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.574	0.429	0.687	0.739
2	5	5	0.517	0.437	0.723	0.767
5	5	10	0.526	0.437	0.725	0.773
10	5	5	0.524	0.438	0.730	0.782
20	5	10	0.537	0.432	0.738	0.791
50	5	10	0.534	0.429	0.736	0.798

Table B.13: Zurich'06

pct	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.444	0.405	0.628	0.659
2	5	5	0.470	0.396	0.627	0.679
5	5	5	0.424	0.397	0.649	0.696
10	5	10	0.435	0.405	0.645	0.707
20	5	20	0.433	0.407	0.648	0.711
50	5	20	0.437	0.403	0.650	0.715

Appendix C

Linearization results with inaccurate training data

In this section the classification results of the linear SAM with and without SIMA transformation are compared when the training data contains a percentage of $swap = [1, 2, 5, 10, 20]$ wrong labels. Additionally, linear ACE and nonlinear SVM results are given. The range for the power of the penalty function was $1, \ldots, 5$, while the tested numbers of nearest neighbors were $k = [1, 2, 5, 10, 20]$. Only the best result in terms of Cohen's κ is printed with the corresponding parameter combination for a training percentage of $pct = 10$.

Table C.1: Greding1_rad

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	5	0.920	0.819	0.984	0.976
2	5	10	0.912	0.819	0.984	0.974
5	5	10	0.893	0.817	0.984	0.974
10	5	20	0.860	0.813	0.983	0.969
20	5	20	0.779	0.801	0.978	0.646

Table C.2: Greding1_refl

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	5	0.902	0.867	0.978	0.973
2	5	5	0.896	0.868	0.978	0.972
5	5	10	0.881	0.865	0.976	0.971
10	5	20	0.855	0.860	0.974	0.966
20	5	20	0.796	0.850	0.972	0.857

Table C.3: Greding2_rad

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	5	0.917	0.838	0.988	0.979
2	5	5	0.912	0.837	0.987	0.979
5	5	10	0.893	0.836	0.984	0.976
10	5	20	0.863	0.831	0.984	0.973
20	5	20	0.781	0.818	0.981	0.758

Table C.4: Greding2_refl

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	5	0.906	0.893	0.981	0.975
2	5	5	0.901	0.894	0.981	0.975
5	5	10	0.884	0.893	0.981	0.973
10	5	20	0.858	0.888	0.979	0.970
20	5	20	0.788	0.873	0.973	0.910

Table C.5: Greding3_rad

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.891	0.829	0.991	0.981
2	4	5	0.888	0.829	0.991	0.981
5	5	5	0.874	0.826	0.987	0.980
10	5	10	0.851	0.822	0.987	0.978
20	5	20	0.793	0.813	0.984	0.952

Table C.6: Greding3_refl

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.889	0.863	0.984	0.978
2	3	5	0.884	0.863	0.983	0.978
5	5	5	0.870	0.860	0.982	0.979
10	5	10	0.847	0.856	0.980	0.975
20	5	20	0.797	0.845	0.976	0.970

Table C.7: Kennedy Space Center

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.360	0.745	0.858	0.760
2	5	2	0.356	0.746	0.857	0.756
5	5	2	0.343	0.740	0.849	0.766
10	5	2	0.316	0.733	0.848	0.740
20	5	2	0.265	0.709	0.834	0.714

Table C.8: Pavia City

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	10	0.885	0.875	0.970	0.964
2	5	20	0.886	0.874	0.970	0.964
5	5	10	0.881	0.871	0.968	0.492
10	5	20	0.872	0.862	0.967	0.488
20	5	20	0.837	0.827	0.964	0.480

Table C.9: Pavia University

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	5	0.461	0.510	0.851	0.822
2	5	5	0.455	0.506	0.849	0.823
5	5	10	0.449	0.501	0.846	0.801
10	5	20	0.421	0.495	0.838	0.774
20	5	20	0.384	0.468	0.812	0.755

Table C.10: Montelly

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	20	0.417	0.468	0.738	0.658
2	5	20	0.420	0.467	0.737	0.648
5	5	20	0.455	0.473	0.736	0.622
10	5	20	0.466	0.477	0.733	0.565
20	5	20	0.504	0.472	0.729	0.415

Table C.11: Prilly

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	10	0.581	0.419	0.764	0.711
2	5	10	0.571	0.412	0.769	0.734
5	5	10	0.575	0.413	0.773	0.734
10	5	5	0.574	0.415	0.775	0.741
20	5	5	0.575	0.415	0.777	0.751
50	5	5	0.575	0.415	0.779	0.767

Table C.12: Zurich'02

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.574	0.429	0.687	0.739
2	5	5	0.517	0.437	0.723	0.767
5	5	10	0.526	0.437	0.725	0.773
10	5	5	0.524	0.438	0.730	0.782
20	5	10	0.537	0.432	0.738	0.791
50	5	10	0.534	0.429	0.736	0.798

Table C.13: Zurich'06

swap	t	k	κ_{SAM}	κ_{ACE}	κ_{SVM}	$\kappa_{SAM-SIMA}$
1	5	2	0.444	0.405	0.628	0.659
2	5	5	0.470	0.396	0.627	0.679
5	5	5	0.424	0.397	0.649	0.696
10	5	10	0.435	0.405	0.645	0.707
20	5	20	0.433	0.407	0.648	0.711
50	5	20	0.437	0.403	0.650	0.715

Appendix D

Data alignment

This chapter contains the combined results of all data alignment experiments.

D.1 Greding

The Greding alignment is performed using Greding1_refl data as a reference and aligning the Greding2_rad and Greding3_rad data using a spatial correspondence map. The results are given in terms of RMS per pixel and band, and as Cohen's κ of the SVM classification of the aligned data sets. The SVM was trained only on the original Greding1_refl training data. The selection of training data was performed using different sampling techniques. Systematic sampling selects evenly spaced training samples from the ground truth per class according to the training percentage. Cluster sampling select spatially close training samples per class, according to the training percentage. Horizontal sampling selects spatially close training samples in horizontal direction per class. This is done to include spectral variations across track.

Table D.3: Performance evaluation of Greding data alignment with variable amounts of incorrect training data $swap = [1, 2, 5, 10, 20]$ using systematic sampling. Training data was set to 10 % per class.

	Greding2_rad				Greding3_rad			
swap	t	k	RMS	SVM	t	k	RMS	SVM
1	5	10	43.6	0.969	5	5	42.5	0.959
2	5	10	44.1	0.968	5	5	45.9	0.958
5	5	20	45.8	0.965	5	10	49.5	0.956
10	5	20	54.5	0.958	5	20	54.5	0.949
20	5	20	82.3	0.876	5	20	82.3	0.922

Table D.1: Alignment of Greding radiance data to Greding1_refl data using systematic sampling

	Greding2_rad				Greding3_rad			
pct	t	k	RMS	SVM	t	k	RMS	SVM
1	4	2	50.3	0.951	2	2	48.4	0.944
2	2	5	46.5	0.961	2	2	47.4	0.950
5	4	10	44.5	0.966	4	5	45.3	0.956
10	4	5	42.7	0.969	4	2	45.7	0.959
20	4	5	43.6	0.971	4	5	43.6	0.961

Table D.2: Comparing sampling strategies for alignment of Greding radiance data to Greding1_refl data using 10 % ground truth data for training per class and data set.

	Greding2_rad				Greding3_rad			
sampling	t	k	RMS	SVM	t	k	RMS	SVM
systematic	4	5	42.7	0.969	4	2	45.7	0.959
cluster	4	20	47.8	0.810	1	1	50.1	0.715
horizontal	4	20	55.2	0.865	4	1	65.8	0.710

Table D.4: Performance evaluation of Greding data alignment with variable amounts of incorrect training data $swap = [1, 2, 5, 10, 20]$ using cluster sampling. Training data was set to 10 % per class.

	Greding2_rad				Greding3_rad			
swap	t	k	RMS	SVM	t	k	RMS	SVM
1	5	20	47.4	0.809	5	5	54.1	0.679
2	5	20	47.4	0.804	5	2	80.3	0.670
5	5	20	49.0	0.786	5	10	59.9	0.665
10	5	20	61.9	0.761	5	20	61.9	0.641
20	5	20	89.9	0.680	5	20	89.9	0.593

Table D.5: Performance evaluation of Greding data alignment with variable amounts of incorrect training data $swap = [1, 2, 5, 10, 20]$ using horizontal sampling. Training data was set to 10 % per class.

	Greding2_rad				Greding3_rad			
swap	t	k	RMS	SVM	t	k	RMS	SVM
1	4	20	55.3	0.865	5	10	57.7	0.706
2	4	20	55.4	0.864	5	10	57.7	0.705
5	3	20	65.2	0.867	5	20	57.5	0.692
10	5	20	70.6	0.836	5	20	70.6	0.686
20	5	20	100.6	0.702	5	20	100.6	0.617

Table D.6: Performance evaluation of Greding data alignment using 10 % training data in the reference and $pct = [0.5, 1.0, 1.5, 2.0]$ in the test data set selected with systematic sampling.

	Greding2_rad				Greding3_rad			
pct	t	k	RMS	SVM	t	k	RMS	SVM
0.5	4	2	65.1	0.946	4	1	68.2	0.942
1.0	4	2	60.3	0.956	4	2	60.3	0.946
1.5	4	2	47.9	0.957	4	2	47.9	0.950
2.0	4	2	49.1	0.960	4	2	49.1	0.951

Table D.7: Performance evaluation of Greding data alignment using 10 % training data in the reference and $pct = [0.5, 1.0, 1.5, 2.0]$ in the test data set selected with horizontal sampling.

	Greding2_rad				Greding3_rad			
pct	t	k	RMS	SVM	t	k	RMS	SVM
0.5	4	10	64.4	0.847	4	1	70.4	0.602
1.0	3	10	61.1	0.861	4	10	59.7	0.584
1.5	3	10	60.0	0.858	4	1	63.6	0.599
2.0	3	20	60.3	0.861	4	1	63.4	0.603

D.2 Montelly / Prilly

This section lists the results of the Montelly and Prilly data sets.

Table D.8: Alignment evaluation of Montelly and Prilly data using spectral similarity to calculate the correspondence map in the common domain. The associated reference data set is stated in the table.

	Montelly				Prilly			
pct	t	k	RMS	SVM	t	k	RMS	SVM
1	4	5	186.3	0.539	4	20	135.2	0.548
2	4	20	190.3	0.489	4	10	160.3	0.545
5	4	2	185.2	0.501	3	2	138.1	0.462
10	4	2	190.6	0.509	4	5	138.2	0.570
20	4	2	189.2	0.527	4	20	135.6	0.569

D.3 Zurich'02 / Zurich'06

This section lists the results of the Zurich'02 and Zurich'06 data sets.

Table D.9: Alignment evaluation of Zurich'02 and Zurich'06 data. The associated reference data set is stated in the table.

	Zurich'02				Zurich'06			
pct	t	k	RMS	SVM	t	k	RMS	SVM
1	1	5	35.8	0.622	1	10	22.2	0.609
2	1	20	31.4	0.674	1	2	29.1	0.612
5	1	10	39.4	0.664	1	20	26.4	0.623
10	1	20	39.3	0.680	1	10	28.6	0.635
20	1	10	41.7	0.685	1	20	28.8	0.649

Table D.10: Alignment evaluation of Zurich'02 and Zurich'06 data with variable amounts of incorrect training data $swap = [1, 2, 5, 10, 20]$. Training data was set to 10 % per class. The associated reference data set is stated in the table.

	Zurich'02				Zurich'06			
swap	t	k	RMS	SVM	t	k	RMS	SVM
1	1	20	38.8	0.679	1	20	27.4	0.633
2	1	5	38.5	0.683	1	20	26.8	0.649
5	1	20	36.0	0.696	1	20	25.6	0.641
10	1	10	30.6	0.699	1	20	22.9	0.636
20	1	10	21.2	0.707	1	20	18.1	0.632

Bibliography

[1] M. A. Armstrong. *Basic topology*. Springer Science & Business Media, 2013.

[2] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE transactions on Geoscience and Remote Sensing*, 43(3):441–454, 2005.

[3] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina. Exploiting manifold geometry in hyperspectral imagery. *IEEE transactions on Geoscience and Remote Sensing*, 43(3):441–454, 2005.

[4] C. M. Bachmann, T. L. Ainsworth, and R. A. Fusina. Improved manifold coordinate representations of large-scale hyperspectral scenes. *IEEE Transactions on Geoscience and Remote Sensing*, 44(10):2786–2803, 2006.

[5] M. Balasubramanian and E. L. Schwartz. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[6] G. Bareth, H. Aasen, J. Bendig, M. L. Gnyp, A. Bolten, A. Jung, R. Michels, and J. Soukkamäki. Low-weight and uav-based hyperspectral full-frame cameras for monitoring crops: Spectral comparison with portable spectroradiometer measurements. *Photogrammetrie-Fernerkundung-Geoinformation*, 2015(1):69–79, 2015.

[7] A. Baumgartner, P. Gege, C. Köhler, K. Lenhard, and T. Schwarzmaier. Characterisation methods for the hyperspectral sensor hyspex at dlr's calibration home base. In *Sensors, Systems, and Next-Generation Satellites XVI*, volume 8533, page 85331H. International Society for Optics and Photonics, 2012.

[8] A. Berk, L. S. Bernstein, and D. C. Robertson. Modtran: A moderate resolution model for lowtran. Technical report, Spectral Sciences Inc Burlington MA, 1987.

[9] M. Bernstein, V. De Silva, J. C. Langford, and J. B. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report, Technical report, Department of Psychology, Stanford University, 2000.

[10] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5(2):354–379, 2012.

[11] M. Brell, K. Segl, L. Guanter, and B. Bookhagen. Hyperspectral and lidar intensity data fusion: A framework for the rigorous correction of illumination, anisotropic effects, and cross calibration. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2799–2810, 2017.

[12] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2010.

[13] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou. Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3044–3054, 2007.

[14] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[15] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

[16] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[17] Y. Chen, M. Crawford, and J. Ghosh. Improved nonlinear manifold learning for land cover classification via intelligent landmark selection. In *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, pages 545–548. IEEE, 2006.

[18] Y. Chen, M. M. Crawford, and J. Ghosh. Applying nonlinear manifold learning to hyperspectral data for land cover classification. In *Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International*, volume 6, pages 4311–4314. IEEE, 2005.

[19] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[20] S. Collings, P. Caccetta, N. Campbell, and X. Wu. Techniques for brdf correction of hyperspectral mosaics. *IEEE Transactions on Geoscience and Remote Sensing*, 48(10):3733–3746, 2010.

[21] J.-P. Combe, S. Le Mouélic, C. Sotin, A. Gendrin, J. Mustard, L. Le Deit, P. Launeau, J.-P. Bibring, B. Gondet, Y. Langevin, et al. Analysis of omega/mars express data hyperspectral data using a multiple-endmember linear spectral unmixing model (melsum): Methodology and first results. *Planetary and Space Science*, 56(7):951–975, 2008.

[22] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. Introduction to algorithms third edition. *MIT Press. ISBN 0-262-03384-4. Section*, 23:631–638, 2009.

[23] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[24] N. Cristianini, J. Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[25] Z. Cui, S. Shan, H. Zhang, S. Lao, and X. Chen. Image sets alignment for video-based face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2626–2633. IEEE, 2012.

[26] C. De Vries, T. Danaher, R. Denham, P. Scarth, and S. Phinn. An operational radiometric calibration procedure for the landsat sensors based on pseudo-invariant target sites. *Remote Sensing of Environment*, 107(3):414–429, 2007.

[27] D. Dong and T. J. McAvoy. Nonlinear principal component analysis—based on principal curves and neural networks. *Computers & Chemical Engineering*, 20(1):65–78, 1996.

[28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[29] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[30] W. H. Farrand, R. B. Singer, and E. Merényi. Retrieval of apparent surface reflectance from aviris data: A comparison of empirical line, radiative transfer, and spectral mixture methods. *Remote Sensing of Environment*, 47(3):311–321, 1994.

[31] M. Fauvel, J. Chanussot, and J. A. Benediktsson. Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas. *EURASIP Journal on Advances in Signal Processing*, 2009:11, 2009.

[32] G. M. Foody. Thematic map comparison. *Photogrammetric Engineering & Remote Sensing*, 70(5):627–633, 2004.

[33] G. M. Foody and A. Mathur. Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification. *Remote Sensing of Environment*, 93(1):107–117, 2004.

[34] J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[35] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[36] C.-S. Fuh and P. Maragos. Motion displacement estimation using an affine model for image matching. *Optical Engineering*, 30(7):881–888, 1991.

[37] B.-C. Gao, M. J. Montes, C. O. Davis, and A. F. Goetz. Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocean. *Remote Sensing of Environment*, 113:S17–S24, 2009.

[38] Y. Gao, Y. Zhang, and K. Chen. Development of a real-time single-frequency precise point positioning system and test results. In *Proceedings of Ion GNSS*, pages 26–29. Citeseer, 2006.

[39] M. Gleicher. Projective registration with difference decomposition. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 331–337. IEEE, 1997.

[40] L. Gómez-Chova, G. Camps-Valls, L. Bruzzone, and J. Calpe-Maravilla. Mean map kernel methods for semisupervised cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*, 48(1):207–220, 2010.

[41] B. Gong, K. Grauman, and F. Sha. Learning kernels for unsupervised domain adaptation with applications to visual object recognition. *International Journal of Computer Vision*, 109(1-2):3–27, 2014.

[42] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2066–2073. IEEE, 2012.

[43] D. G. Goodenough, A. Dyk, K. O. Niemann, J. S. Pearlman, H. Chen, T. Han, M. Murdoch, and C. West. Processing hyperion and ali for forest classification. *IEEE transactions on geoscience and remote sensing*, 41(6):1321–1331, 2003.

[44] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 999–1006. IEEE, 2011.

[45] M. K. Griffin and H.-h. K. Burke. Compensation of hyperspectral data for atmospheric effects. *Lincoln Laboratory Journal*, 14(1):29–54, 2003.

[46] W. Gross, J. Boehler, H. Schilling, W. Middelmann, J. Weyermann, P. Wellig, R. Oechslin, and M. Kneubuehler. Assessment of target detection limits in hyperspectral data. *Proc. SPIE Secur.+ Def*, 9653, 2015.

[47] W. Gross, J. Böhler, K. Twizer, B. Kedem, A. Lenz, M. Kneubuehler, P. Wellig, R. Oechslin, H. Schilling, S. Rotman, et al. Determination of target detection limits in hyperspectral data

using band selection and dimensionality reduction. In *Target and Background Signatures II*, volume 9997, page 99970H. International Society for Optics and Photonics, 2016.

[48] W. Gross, S. Borchardt, and W. Middelmann. Evaluation of spectral unmixing using nonnegative matrix factorization on stationary hyperspectral sensor data of specifically prepared rock and mineral mixtures. In *Proc. OCM 2013–Optical Characterization of Materials*, volume 1, pages 169–178, 2013.

[49] W. Gross, N. Espinosa, M. Becker, S. Schreiner, and W. Middelmann. Improving linear classification using semi-supervised invertible manifold alignment. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 3551–3554. IEEE, 2018.

[50] W. Gross, G. Keskin, H. Schilling, A. Lenz, and W. Middelmann. Automatic modeling of nonlinear signal source variations in hyperspectral data. In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 2965–2968. IEEE, 2014.

[51] W. Gross, D. Tuia, U. Soergel, and W. Middelmann. Nonlinear feature normalization for hyperspectral domain adaptation and mitigation of nonlinear effects. *IEEE Transactions on Geoscience and Remote Sensing*, 2019.

[52] W. Gross, S. Wuttke, and W. Middelmann. Transformation of hyperspectral data to improve classification by mitigating nonlinear effects. In *7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, WHISPERS*, 2015.

[53] N. Haala, D. Fritsch, D. Stallmann, and M. Cramer. On the performance of digital airborne pushbroom cameras for photogrammetric data processing-a case study. *International Archives of Photogrammetry and Remote Sensing*, 33(B4/1; PART 4):324–331, 2000.

[54] G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE transactions on pattern analysis and machine intelligence*, 20(10):1025–1039, 1998.

[55] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh. Investigation of the random forest framework for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):492–501, 2005.

[56] J. Ham, D. D. Lee, and L. K. Saul. Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127, 2005.

[57] R. Hänsch, A. Ley, and O. Hellwich. Correct and still wrong: The relationship between sampling strategies and the estimation of the generalization error. In *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*, pages 3672–3675. IEEE, 2017.

[58] R. Heylen, M. Parente, and P. Gader. A review of nonlinear hyperspectral unmixing methods. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):1844–1868, 2014.

[59] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza. Sparse unmixing of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2014–2039, 2011.

[60] E. Izquierdo-Verdiguier, V. Laparra, L. Gomez-Chova, and G. Camps-Valls. Encoding invariances in remote sensing image classification with svm. *IEEE Geoscience and Remote Sensing Letters*, 10(5):981–985, 2013.

[61] X. Jia and J. A. Richards. Segmented principal components transformation for efficient hyperspectral remote-sensing image display and classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37(1):538–542, 1999.

[62] G. Jun and J. Ghosh. Spatially adaptive semi-supervised learning with gaussian processes for hyperspectral data analysis. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(4):358–371, 2011.

[63] W. Kim and M. M. Crawford. Adaptive classification for hyperspectral image data using manifold regularization kernel machines. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4110–4121, 2010.

[64] D. Knuth. A generalization of dijkstra's algorithm. In *Inf. Proc. Letters*, volume 6, pages 1–7, 1977.

[65] R. F. Kokaly, R. N. Clark, G. A. Swayze, K. E. Livo, T. M. Hoefen, N. C. Pearson, R. A. Wise, W. M. Benzel, H. A. Lowers, R. L. Driscoll, et al. Usgs spectral library version 7. 2017.

[66] E. Kreyszig. *Introduction to differential geometry and Riemannian geometry*. University of Toronto Press, 1968.

[67] F. A. Kruse, A. Lefkoff, J. Boardman, K. Heidebrecht, A. Shapiro, P. Barloon, and A. Goetz. The spectral image processing system (sips)—interactive visualization and analysis of imaging spectrometer data. *Remote sensing of environment*, 44(2-3):145–163, 1993.

[68] F. Kühn, K. Oppermann, and B. Hörig. Hydrocarbon index–an algorithm for hyperspectral detection of hydrocarbons. *International Journal of Remote Sensing*, 25(12):2467–2473, 2004.

[69] J.-Y. Kwok and I.-H. Tsang. The pre-image problem in kernel methods. *IEEE transactions on neural networks*, 15(6):1517–1525, 2004.

[70] C. A. Laben and B. V. Brower. Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening, Jan. 4 2000. US Patent 6,011,875.

[71] P. L. Lai and C. Fyfe. Kernel and nonlinear canonical correlation analysis. *International Journal of Neural Systems*, 10(05):365–377, 2000.

[72] M. Langhans, S. van der Linden, A. Damm, and P. Hostert. The influence of bidirectional reflectance in airborne hyperspectral data on spectral angle mapping and linear spectral mixture analysis. 2007.

[73] K. C. Lawrence, B. Park, W. R. Windham, and C. Mao. Calibration of a pushbroom hyperspectral imaging system for agricultural inspection. *Transactions of the ASAE*, 46(2):513, 2003.

[74] P. Leite, J. M. Teixeira, T. Farias, B. Reis, V. Teichrieb, and J. Kelner. Nearest neighbor searches on the gpu. *International Journal of Parallel Programming*, 40(3):313–330, 2012.

[75] A. Lenz, H. Schilling, D. Perpeet, S. Wuttke, W. Gross, and W. Middelmann. Automatic in-flight boresight calibration considering topography for hyperspectral pushbroom sensors.

In *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, pages 2981–2984. IEEE, 2014.

[76] Y. Liu, J. Bioucas-Dias, J. Li, and A. Plaza. Hyperspectral cloud shadow removal based on linear unmixing. In *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*, pages 1000–1003. IEEE, 2017.

[77] X. Ma and N. Zabaras. Kernel principal component analysis for stochastic input model generation. *Journal of Computational Physics*, 230(19):7311–7331, 2011.

[78] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, volume 2, page 3, 2014.

[79] M. Maier, U. V. Luxburg, and M. Hein. Influence of graph construction on graph-based clustering measures. In *Advances in neural information processing systems*, pages 1025–1032, 2009.

[80] G. Matasci, D. Tuia, and M. Kanevski. Svm-based boosting of active learning strategies for efficient domain adaptation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(5):1335–1343, 2012.

[81] G. Matasci, M. Volpi, M. Kanevski, L. Bruzzone, and D. Tuia. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7):3550–3564, 2015.

[82] S. Matteoli, E. J. Ientilucci, and J. P. Kerekes. Operational and performance considerations of radiative-transfer modeling in hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 49(4):1343–1355, 2011.

[83] M. W. Matthew, S. M. Adler-Golden, A. Berk, G. Felde, G. P. Anderson, D. Gorodetzky, S. Paswaters, and M. Shippert. Atmospheric correction of spectral imagery: evaluation of the

flaash algorithm with aviris data. In *Applied Imagery Pattern Recognition Workshop, 2002. Proceedings. 31st*, pages 157–163. IEEE, 2002.

[84] G. J. Michon. Fixed-pattern noise correction circuitry for solid-state imager, May 31 1994. US Patent 5,317,407.

[85] P. Misra and P. Enge. Global positioning system: Signals, measurements and performance second edition. *Massachusetts: Ganga-Jamuna Press*, 2006.

[86] T. Mori, R. Taketa, S. Hiura, and K. Sato. Photometric linearization by robust pca for shadow and specular removal. In *Computer Vision, Imaging and Computer Graphics. Theory and Application*, pages 211–224. Springer, 2013.

[87] M. Mostafa, J. Hutton, B. Reid, et al. Gps/imu products–the applanix approach. In *Photogrammetric Week*, volume 1, pages 63–83, 2001.

[88] K. T. Mueller, P. V. Loomis, R. M. Kalafus, and L. Sheynblat. Networked differential gps system, June 21 1994. US Patent 5,323,322.

[89] H. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International journal of computer vision*, 14(1):5–24, 1995.

[90] O. Mutanga and A. K. Skidmore. Hyperspectral band depth analysis for a better estimation of grass biomass (cenchrus ciliaris) measured under controlled laboratory conditions. *International journal of applied earth observation and geoinformation*, 5(2):87–96, 2004.

[91] E. Naesset. Effects of differential single-and dual-frequency gps and glonass observations on point accuracy under forest canopies. *Photogrammetric Engineering and Remote Sensing*, 67(9):1021–1026, 2001.

[92] A. A. Nielsen. Multiset canonical correlations analysis and multispectral, truly multitemporal remote sensing data. *IEEE transactions on image processing*, 11(3):293–305, 2002.

[93] C. W. O'Dell. Acceleration of multiple-scattering, hyperspectral radiative transfer calculations via low-streams interpolation. *Journal of Geophysical Research: Atmospheres*, 115(D10), 2010.

[94] F. Omruuzun, D. O. Baskurt, H. Daglayan, and Y. Y. Cetin. Shadow removal from vnir hyperspectral remote sensing imagery with endmember signature analysis. In *Next-Generation Spectroscopic Technologies VIII*, volume 9482, page 94821F. International Society for Optics and Photonics, 2015.

[95] M. K. Pal and A. Porwal. Evaluation and modelling across-track illumination variation in hyperion images using quadratic regression. *International Journal of Remote Sensing*, 38(23):6790–6815, 2017.

[96] N. Paragios, Y. Chen, and O. D. Faugeras. *Handbook of mathematical models in computer vision*. Springer Science & Business Media, 2006.

[97] Y. Pei, F. Huang, F. Shi, and H. Zha. Unsupervised image matching based on manifold alignment. *IEEE transactions on pattern analysis and machine intelligence*, 34(8):1658–1664, 2012.

[98] T. Perkins, S. M. Adler-Golden, M. W. Matthew, A. Berk, L. S. Bernstein, J. Lee, and M. Fox. Speed and accuracy improvements in flaash atmospheric correction of hyperspectral imagery. *Optical Engineering*, 51(11):111707, 2012.

[99] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[100] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gaultieri, et al. Advanced processing of hyperspectral images. In *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, pages 1974–1978. IEEE, 2006.

[101] G. Rabatel, N. Al Makdessi, M. Ecarnot, and P. Roumet. A spectral correction method for multi-scattering effects in close range hyperspectral imagery of vegetation scenes: application to nitrogen content assessment in wheat. *Advances in Animal Biosciences*, 8(2):353–358, 2017.

[102] M. Reed and B. Simon. *Methods of Modern Mathematical Physics: Functional Analysis.-1972.-(RU-idnr: M103448034)*. Academic Press, 1972.

[103] R. Richter and D. Schläpfer. Geo-atmospheric processing of airborne imaging spectrometry data. part 2: atmospheric/topographic correction. *International Journal of Remote Sensing*, 23(13):2631–2649, 2002.

[104] R. Richter and D. Schläpfer. Atmospheric/topographic correction for airborne imagery. *ATCOR-4 user guide*, 2011.

[105] F. Riesz and B. S. Nagy. Functional analysis. translated from the 2nd french edition by leo f. boron. *Dover Publications Inc., New York*, 2(4):2–2, 1990.

[106] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.

[107] S. Schiefer, P. Hostert, and A. Damm. Correcting brightness gradients in hyperspectral data from urban areas. *Remote sensing of environment*, 101(1):25–37, 2006.

[108] D. Schläpfer and R. Richter. Evaluation of brefcor brdf effects correction for hyspex, casi, apex imaging spectroscopy data. *Proc. 6th IEEE WHISPERS*, page 4, 2014.

[109] D. Schläpfer and R. Richter. Recent developments in atcor for atmospheric compensation and radiometric processing of imaging spectroscopy data. *EARSeL eProceedings*, 10(2015):1, 2015.

[110] D. Schläpfer, R. Richter, and A. Damm. Correction of shadowing in imaging spectroscopy data by quantification of the proportion of diffuse illumination. In *Proc. 8th SIG-EARSeL Imag. Spectrosc. Workshop Nantes*, page 10, 2013.

[111] D. Schläpfer, R. Richter, and T. Feingersh. Operational brdf effects correction for wide-field-of-view optical scanners (brefcor). *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):1855–1864, 2015.

[112] D. Schläpfer, R. Richter, and A. Hueni. Recent developments in operational atmospheric and radiometric correction of hyperspectral imagery. *EARSeL SIG IS*, 2009.

[113] B. Scholkopf, S. Mika, C. J. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A. J. Smola. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5):1000–1017, 1999.

[114] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. In *Panoramic vision*, pages 227–268. Springer, 2001.

[115] V. D. Silva and J. B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in neural information processing systems*, pages 721–728, 2003.

[116] G. W. Staben, K. Pfitzner, R. Bartolo, and A. Lucieer. Empirical line calibration of worldview-2 satellite imagery to reflectance data: Using quadratic prediction equations. *Remote sensing letters*, 3(6):521–530, 2012.

[117] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[118] N. Thorstensen. *Manifold learning and applications to shape and image processing*. PhD thesis, Ecole des Ponts ParisTech, 2009.

[119] D. Tuia and G. Camps-Valls. Kernel manifold alignment for domain adaptation. *PloS one*, 11(2):e0148655, 2016.

[120] D. Tuia, D. Marcos, and G. Camps-Valls. Multi-temporal and multi-source remote sensing image classification by nonlinear relative normalization. *ISPRS Journal of Photogrammetry and Remote Sensing*, 120:1–12, 2016.

[121] D. Tuia, J. Munoz-Mari, L. Gomez-Chova, and J. Malo. Graph matching for adaptation in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1):329–341, 2013.

[122] D. Tuia, E. Pasolli, and W. J. Emery. Using active learning to adapt remote sensing image classifiers. *Remote Sensing of Environment*, 115(9):2232–2242, 2011.

[123] D. Tuia, C. Persello, and L. Bruzzone. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE geoscience and remote sensing magazine*, 4(2):41–57, 2016.

[124] D. Tuia and M. Trolliet. Multisource alignment of image manifolds. In *IEEE International Geoscience and Remote Sensing Symposium, IGARSS*, number EPFL-CONF-197375, 2013.

[125] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls. Semisupervised manifold alignment of multimodal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12):7708–7720, 2014.

[126] D. Tuia, M. Volpi, M. Trolliet, and G. Camps-Valls. Semisupervised manifold alignment of multimodal remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 52(12):7708–7720, 2014.

[127] W. Verhoef and H. Bach. Simulation of hyperspectral and directional radiance images using coupled biophysical and atmospheric radiative transfer models. *Remote sensing of environment*, 87(1):23–41, 2003.

[128] C. Wang and S. Mahadevan. A general framework for manifold alignment. In *AAAI Fall Symposium: Manifold Learning and Its Applications*, 2009.

[129] C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *IJCAI*, volume 2, page 3, 2009.

[130] C. Wang and S. Mahadevan. Heterogeneous domain adaptation using manifold alignment. In *IJCAI proceedings-international joint conference on artificial intelligence*, volume 22, page 1541, 2011.

[131] Z. Wang and X. Xue. Multi-class support vector machine. In *Support Vector Machines Applications*, pages 23–48. Springer, 2014.

[132] J. Weston, B. Schölkopf, and G. H. Bakir. Learning to find pre-images. In *Advances in neural information processing systems*, pages 449–456, 2004.

[133] J. Weston and C. Watkins. Multi-class support vector machines. Technical report, Citeseer, 1998.

[134] J. Weyermann, D. Schläpfer, A. Hueni, M. Kneubühler, and M. Schaepman. Spectral angle mapper (sam) for anisotropy class indexing in imaging spectrometry data. *Imaging Spectrometry XIV*, 7457:74570B, 2009.

[135] J.-L. Widlowski, B. Pinty, T. Lavergne, M. M. Verstraete, and N. Gobron. Using 1-d models to interpret the reflectance anisotropy of 3-d canopy targets: Issues and caveats. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2008–2017, 2005.

[136] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 7.

[137] C. Wu, Z. Niu, Q. Tang, and W. Huang. Estimating chlorophyll content from hyperspectral vegetation indices: Modeling and validation. *Agricultural and forest meteorology*, 148(8):1230–1241, 2008.

[138] S. Wuttke, H. Schilling, and W. Middelmann. Reduction of training costs using active classification in fused hyperspectral and lidar data. In *Image and Signal Processing for Remote Sensing XVIII*, volume 8537, page 85370M. International Society for Optics and Photonics, 2012.

[139] H. L. Yang and M. M. Crawford. Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):51–64, 2016.

[140] P. Zarco-Tejada, J. Miller, G. Mohammed, T. Noland, and P. Sampson. Vegetation stress detection through chlorophyll+ estimation and fluorescence effects on hyperspectral imagery. *Journal of environmental quality*, 31(5):1433–1441, 2002.

[141] Z.-Q. Zeng, H.-B. Yu, H.-R. Xu, Y.-Q. Xie, and J. Gao. Fast training support vector machines using parallel sequential minimal optimization. In *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, volume 1, pages 997–1001. IEEE, 2008.

[142] Q. Zhang, V. P. Pauca, R. J. Plemmons, and D. D. Nikic. Detecting objects under shadows by fusion of hyperspectral and lidar data: A physical model approach. In *Proc. 5th Workshop Hyperspectral Image Signal Process.: Evol. Remote Sens*, pages 1–4, 2013.

[143] T. Zhang, X. Li, D. Tao, and J. Yang. Local coordinates alignment (lca): a novel manifold learning approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(04):667–690, 2008.

[144] Z. Zhang and J. Wang. Mlle: Modified locally linear embedding using multiple weights. In *Advances in neural information processing systems*, pages 1593–1600, 2007.

[145] F. Zhu, P. Honeine, and M. Kallas. Kernel nonnegative matrix factorization without the pre-image problem. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, pages 1–6. IEEE, 2014.

[146] F. Zhu, Y. Wang, S. Xiang, B. Fan, and C. Pan. Structured sparse method for hyperspectral unmixing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 88:101–118, 2014.

Curriculum Vitae

Wolfgang Johannes Groß

PERSONAL DETAILS

Birth	March 1, 1985
Address	Seestraße 30d, 76275 Ettlingen
Mail	`wolfgang.gross@iosb.fraunhofer.de`

EDUCATION

Diplom Technomathematik 2004-2011
Karlsruhe Institute of Technology KIT

WORK EXPERIENCE

Research Fellow 2011-present
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB, Full-time

Project management, planning and conducting measurement campaigns, research, and publication of results. Current research topics are the development of algorithms for hyperspectral analysis regarding data-driven automatic pre-processing, classification and manifold alignment for data alignment.

SKILLS

Languages	German (mother tongue) English (fluent)
Software	MATLAB, LaTeX, ENVI, ERDAS IMAGINE,C++