NTNU
Norwegian University of
Science and Technology

Dimensionality Reduction and Target Detection in Hyperspectral Remote Sensing

Sivert Bakken

Master of Science in Cybernetics and Robotics
Submission date: May 2018
Supervisor: Tor Arne Johansen, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

To Anne and Arve

Abstract

The work presented here is originally motivated by the low transfer rate possible for earth orbiting small satellites. Through different methods for dimensionality reduction, the effects of such representation on data exploration with respect to target detection have been investigated. The theory suggests that through using these methods for dimensionality reduction a considerable compression ratio could be achieved whiteout losing desired information. How these reduced space representations of the data then would affect the exploratory analysis, here represented by target detection with known signatures, was used as an example. The different dimensionality reduction methods were PCA, MNF and JADE ICA, and the different target detection algorithms were ACE, CEM, OSP, and SAM. Target detection performance has been measured using the F_1-score and Matthews correlation coefficient for binary classification performance, and visibility for robustness.

When looking purely at the compression of synthetic data all dimensionality transforms performed as expected. The MNF transform was able to achieve a higher score of restoration than the others for all cases. It should be noted that the synthetic scenes fit well with some of the assumptions made in the MNF implementation. On data from the HICO mission the MNF representation was not able to produce as good results as when performed on synthetic data or data with ground truth. It is speculated that this might be due to the limitations found in the raw data of the HICO mission and how the restoration was measured, whilst the synthetic scenes and the scenes with known ground truth were able to give consistent results for all dimensionality reduction methods, and especially for the MNF transform.

In the simulations performed MNF was in most cases able to represent the data with a high level of visibility and good results wrt. both the F_1-score and the Matthews correlation coefficient, across all detection algorithms. It was demonstrated that the MNF transform will be dependent on the estimated noise model, but that even with a lesser noise model the MNF transform was able to perform well, when compared to PCA and ICA. These results were trending across all target detection algorithms.

The different Target detection algorithms were able to produce good results, both wrt. binary classification and robustness. The ACE and SAM detection algorithms showed great promise wrt. both their ability to detect targets and their apparent robustness. The CEM algorithm often had a high detection ability, but had a tendency to be less robust. Lastly, the OSP detector did not perform as well as the others, for any of the scenes tested.

The combination of dimensionality reduction and target detection displayed positive results wrt. both detection rate and robustness. This suggests that combining the two in the proposed pipeline will not undermine the performance of the system as a whole, and potentially even strengthen the operation performed by a target detection system.

Sammendrag

(Norwegian Translation of the Abstract)
Arbeidet som presenteres her er opprinnelig motivert av den lave overføringshastigheten oppnålig for satelliter i bane rundt jorda. Det er kjent at man kan ved dimensjonalitetsreduksjon oppnå betydelig datakompresjon, uten å miste for mye viktig informasjon. Hvordan disse reduserte rom-representasjonene av dataene da vil påvirke data-utforskiningen, her representert ved signaturbasert deteksjon, ble sett på. De forskjellige dimensjonsreduksjonsmetodene var PCA, MNF og JADE ICA, og de forskjellige signaturbaserte deteksjonsalgoritmene var ACE, CEM, OSP og SAM. den spesifikke sekvensen av oprasjoner forslått i denne oppgavens prestasjonsevne har blitt målt ved hjelp av F_1-score og Matthews korrelasjonskoeffisient for binær klassifisering, videre har synlighet (visibility) blitt brukt som et mål for robusthet.

Ved kompresjonen på syntetiske data presterer alle dimensjonsreduksjonsmetodene som forventet. MNF-transformasjonen var i stand til å oppnå en høyere gjenopprettingsgrad enn de andre for alle tilfeller av støy på den syntetiske dataen. Det skal bemerkes at de syntetiske bildene passer godt med noen av antagelsene som er gjort i MNF-implementeringen. På data fra HICO-oppdraget var MNF-representasjonen ikke i stand til å produsere like gode resultater som når det ble utført på syntetiske data eller data med kjent innhold (ground truth). Dette er sannsynligvis grunnet begrensningene i de rå dataene fra HICO-oppdraget og hvordan restaureringen ble målt. For de syntetiske bildene og bildene med kjent innhold var algoritmene i stand til å gi konsistente resultater for alle dimensjonsreduksjonsmetodene.

I simuleringene var MNF i de fleste tilfeller i stand til å representere dataene med et høyt synlighetsnivå og gode resultater mtp. både F_1-score og Matthews korrelasjonskoeffisient, på tvers av alle deteksjonsalgoritmer. Det ble påvist at resultater fra MNF-transformasjonen vil være sterkt avhengig av hvor god en estimert støymodell var, men at selv med en mindre god støymodell var MNF-transformasjonen i stand til å prestere godt, sammenlignet med PCA og ICA. Det var en trend mot disse resultatene for alle de signaturebaserte deteksjonsalgoritmene på tvers av alle dimensjonsreduksrepresentasjoner.

De forskjellige signaturebaserte deteksjonsalgoritmene var i stand til å produsere gode resultater, både med hensyn på binær klassifiserings evne og robusthet. ACE og SAM deteksjonsalgoritmene viste lovende resultater, både med tanke på deres evne til å detektere mål og deres tilsynelatende gode robusthet. CEM-algoritmen hadde ofte en høy deteksjonsevne, men hadde en tendens til å være mindre robust. Til slutt må det påpekes at OSP-algoritmen ikke presterte så godt som de andre undersøkt her, som var gjeldende for alle bildene undersøkt.

Kombinasjonen av dimensionalitetsreduksjon og deteksjon viste positive resultater med hensyn på både deteksjonsevne og robusthet. Dette tyder på at kombinasjonen av de to i den foreslåtte sekvensen av oprasjoner ikke vil undergrave systemets ytelse som en helhet, og potensielt til og med styrke systemets ytelsesevne vedrørende signaturbasert deteksjon.

Preface

The work presented in this thesis has been carried out at the Department of Engineering Cybernetics, in many ways as a part of the preliminary exploratory work regarding the NTNU SmallSat project. This thesis is mainly concerned with remote sensing, a field not currently thought through the MSc. program in engineering cybernetics at NTNU. It is not a continuation of the work done in the project thesis. Furthermore, the toolboxes used was not provided for, but was discovered as part of the research for this thesis, and the limitations of these toolboxes are elaborated in the text. The datasets used in this thesis were not provided for, but also found as a part of the research. With the exception of the HICO datasets which were provided for at the beginning of February, as they also were used by other MSc. students trying to combine several forms of compression.

However, as a student it is always wonderful to learn new things, and even though it has been a trialing learning experience, it has also been a rewarding learning experience.

In the process of writing this thesis I have sought the help of far too many people to ever be listed, but naturally not all answered. However, the people who did certainly deserves praise for their valuable inputs, calming remarks, motivational talks and ambitious hopes for me and for this thesis. First of all I would like to thank my supervisor, Tor Arne Johansen, for believing in me when I don't and for giving me space to explore the potential paths to creating this thesis. Secondly, I would like to thank Milica Orlandic, as my unofficial co-supervisor she has explored the field of hyperspectral imaging with me, always been available for questions and discussions and provided me with both academic and emotional support when I needed it the most. Special thanks need to be given to Frank Westad for his invaluable insight regarding the analysis of hyperspectral data and for always giving me the time of day no matter how occupied he was or where in the world he might be.

Furthermore, my fellow MSc. students deserve a special thank for creating a social and supportive environment, where it was possible to vent one's problems both regarding the thesis work and other aspects of life.

Lastly, I would like to direct a special thank to my friends and family, even though they do not always understand what I am doing they are always supportive and they never waver in their belief in me.

Sivert Bakken
June 4, 2018

Acronyms

ACE Adaptive Cosine Estimator.

AVIRIS Airborne Visible / Infrared Imaging Spectrometer.

CEM Constrained Energy Minimization.

DR Dimensionality Reduction.

HICO Hyperspectral Imager for Coastal Ocean.

ICA Independent Component Analysis.

ISS International Space Station.

JADE Joint Approximation Diagonalization of Eigen-matrices.

MCC Matthews Correlation Coefficient.

MNF Maxmimum Noise Fraction.

NASA National Aeronautics and Space Administration (US).

NTNU Norwegian University of Science and Technology.

OSP Orthogonal Subspace Projection.

PCA Principle Component Analysis.

SAM Spectral Angle Mapper.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The vast majority of the world's surface consists of water, and the ocean plays a pivotal role in the earth's health and climate. Combined with the last half century of developments in aerospace technology, we have not just been able to explore space, but also enabled seeing ourselves and our oceans from a new perspective.

The ocean is of particular importance for the Norwegian society and economy. The Norwegian continental shelf is four times the area of the Norwegian mainland and constitutes about one-third of the European continental shelf. When taking this into consideration, Norway becomes the second largest country in Europe. The Norwegian University of Science and Technology is currently developing a small satellite program for multi-agent marine observations. This program is intended as the beginning of a dedicated constellation of ocean observing satellites.

The first mission of the NTNU SmallSat program is intended to be an ocean color satellite equipped with a hyperspectral camera [5]. A hyperspectral camera captures a broader range of wavelengths than the human eye can see. Through the information captured by the satellite, and the processing performed onboard, the system will be able to provide early warnings of oceanographic phenomena of interest, such as harmful algal blooms approaching fish farms. The SmallSat will act as an intelligent agent in a tightly controlled and coordinated autonomous mission, being supplemented in the analysis by other autonomous vehicles such as UAVs, ASVs, and AUVs.

For this mission to fulfill its objectives, innovative steps in terms of onboard data processing need to be taken. To be able to downlink the acquired data, compression needs to be performed, and to detect oceanographic phenomena of interest, exploratory data analysis will have to be executed. This thesis surveys different ways to do compression with dimensionality reduction and how this representation will affect target detection.

1.1 Dimensionality Reduction

For the NTNU SmallSat Program [5] there is a significant bottleneck in how fast the system is able to downlink a captured image. Thus the onboard processing should compress the data as much as possible without losing valuable information, before downlinking. However, what to consider as valuable information will be dependent on the application.

As the data and knowledge specifically tailored towards the system envisioned in [5] was unavailable at the time of writing, a solution specifically made for the system and its objectives were deemed unachievable in the time frame given to complete this thesis. Thus, a more general approach to dimensionality reduction was chosen, aligned with the state-of-the-art literature.

In literature and publications regarding dimensionality reduction for hyperspectral imaging, methods such as Principal Component Analysis, Maximum Noise Factoring, and Independent Component Analysis, described briefly in this thesis in section 2.2, are referenced frequently. These methods are used for exploratory analysis, compression and noise filtering within the field of hyperspectral image analysis [6, 7, 1, 8]. Even though these methods are not recent developments, their contribution to the analysis and preprocessing and processing of hyperspectral data has proven valuable, and different pipelines to improve hyperspectral image analysis utilizing these methods are still being published. There are of course modifications to these three methods, and how they have been executed in this thesis is described in chapter 3. The given methods all have their advantages and disadvantages, that will become more clear as the text progresses and their theory is elaborated upon.

1.2 Target Detection

As one of the objectives of the first satellite in the NTNU SmallSat Program [5] is to detect oceanographic phenomena of interest, an investigation of different methods for hyperspectral target detection algorithms is appropriate.

As stated in [9], an idea or clearly superior target detection algorithm may not exist, and in the quest for of achieving perfect performance from a detector, many issues are often overlooked. Some of the main concerns when developing an application specific detection algorithm raised in [9] are:

- It's challenging to select a threshold that gives a desired false alarm rate
- Sensor calibration may affect detection gravely
- The noise level experienced by the sensor will have a bad influence on detection and may change over time
- The atmospheric correction may not render a cube close to the true spectra
- The background may change a lot from the original model over time
- The signature being looked for is not necessarily close to the true target under all conditions
- Spectral variability of the target is difficult to account for

Thus, the performance gained by constructing more complex and application specific target detection algorithm may not produce desired results in practical applications.

This thesis is not the first study to compare different target detection algorithms for hyperspectral image analysis on the full space. Some of the more noteworthy publications that inspired this thesis are [10, 9, 11, 3]. Often, given the aforementioned publications, a low false alarm rate is deemed as the main objective. That is, a threshold is chosen so that as few as possible non-target pixels will be labeled as target pixels, resulting in a false alarm. As the false alarm rate may be application specific and dependent on the mission objective [9] a constant threshold was not used in this thesis.

Instead of a measure of how well a given target detection algorithm is able to separate target and background, dubbed visibility, was adopted form [10], and a description of this metric is restated in section 3.4.2.

Furthermore, the threshold used in this thesis was chosen by maximizing a given machine learning performance metric(see section 3.4). Thus, giving an indication as to which of the target detection algorithms that will give the best score for a chosen metric under good conditions.

As stated above, this is not the most common way to compare different target detection algorithms but was chosen opposed to the more common constant threshold for all target detection algorithms to show the best performance of a given target detection algorithm and the visibility chosen to illustrate the robustness of the algorithm. As a consequence of these choices, the results given in this thesis are not reliant on an externally chosen values.

1.3 Structure of This Thesis

This thesis is divided into 5 chapters and 4 appendices

- Chapter 1 is intended as a short description of the background and motivation for this thesis.
- Chapter 2 is a cursory overview of the methods used in this thesis.
- Chapter 3 covers how the different simulations were performed, and what kind of data sets were used.
- Chapter 4 is a discussion of the results
- Chapter 5 gives a final conclusion to what the work performed for this thesis accumulated into and suggestions for further work.
- Appendix A contains some selected excerpt of the code used for simulations.
- Appendix B is a short description of some of the instruments used for capturing the scenes analyzed in this thesis.
- Appendix C gives tables of different results wrt. compression on synthetic data and HICO data.
- Appendix D contains all results for all algorithms and scenes displaying their performance with respect to different metrics

Chapter 2

Background Theory

2.1 Hyperspectral Imaging and Processing

This section is mainly developed from the books *Hyperspectral Imaging Technology in Food and Agriculture* by B. Park and R. Lu [1] and *Hyperspectral Remote Sensing* by Michael T. Eismann [6], and the paper *Detection Algorithms in Hyperspectral Imaging Systems* by D. Manolakis et al. [11].

Hyperspectral Image processing is the collective term for using different types of algorithms to derive, manipulate or store information from relevant spectral bands in a hyperspectral image, often called image cube[1]. An image cube is a set of values spanning three dimensions x,y, and z, where x and y make up the spatial composition of the image and the z-direction stores the spectral information. A hyperspectral image collects and processes information from across the electromagnetic spectrum with an objective to obtain the spectrum for each pixel in the image of a scene for analysis e.g. identifying materials or detecting processes. Typically, different sources will claim that the hyperspectral domain starts when the image surpasses 30 spectral bands It is however not uncommon that a hyperspectral image contains 100 bands or more [12, 1, 13]. Most hyperspectral instruments share a high spectral resolution, or at the very least segments of high spectral resolution i.e. certain areas of spectral relevance are particularly well covered. The spectral bands most often associated with hyperspectral image processing are contained within visible to near-infrared light, often shortened down to VNIR or VISNIR. The algorithms used in hyperspectral imaging usually handle such problems as classification or detection, to better understand and characterize the scene or an object in the image. Hyperspectral image processing is in general concerned with the wavelengths between in and around the visible range as shown in figure 2.1. There are limitations to the measurements that Hyperspectral sensors may bring and figure 2.2 displays to some extent the uncertainties that hyperspectral imaging will bring as a result of illumination and reflectance angle differences. Figure 2.2 shows the spectral variance of the metal sheets in the Pavia scene depicted in figure 3.6. The red dashed line is the spectra from each pixel labeled as metal sheets, and the dark solid line is the mean of all pixels labeled as painted metal sheets.

Hyperspectral image processing is inherently a multi-disciplinary field, and many of the techniques developed for hyperspectral image processing has its origin in other fields. Chemometrics and spectroscopy have long been developed fields for treating an abundance of spectral data, but also machine learning and data analytics play an important role in hyperspectral imaging when compared to more traditional remote sensing [14, 1]. Traditional Remote sensing only uses a few spectral bands to perform analysis, and a lot of information of the scene will be lost due to the low spectral resolution. However, as the spectral resolution increases the importance of high spatial resolution increases as well. The spectral response of an object is the spectral signature that it gives from being illuminated by a given light source, e.g. the sun. As will become clear, it is not trivial to distinguish between the spectral responses of different objects. If the spectral response of several objects is captured in one pixel, distinguishing between them is even more complicated. An image consisting of hyperspectral data can inherently be considered a multidimensional or multivariate problem. Thus multivariate techniques found in exploratory statistics and chemometrics can be used to process the spectral information and to gain insights. These methods are well-established approaches within the fields of medical analysis, chemometrics, food analytics, precision agriculture and many more[15, 6, 1].

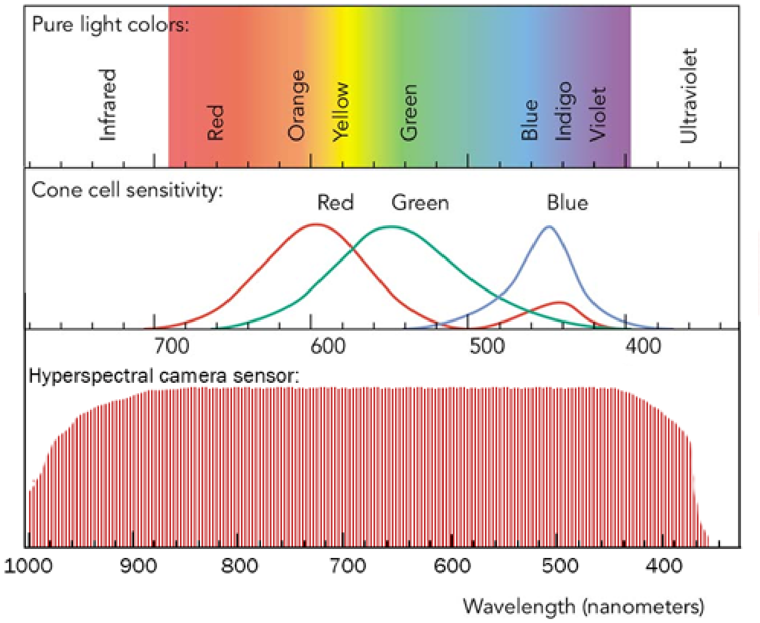


Figure 2.1: An illustration of different electromagnetic spectrum. Typically a hyperspectral image sensor will capture light in the in the ultraviolet, visible and near-infrared wavelengths. [1]. The image is taken from `https://app.griffith.edu.au/sciencesimpact/hyperspectral-imaging/`

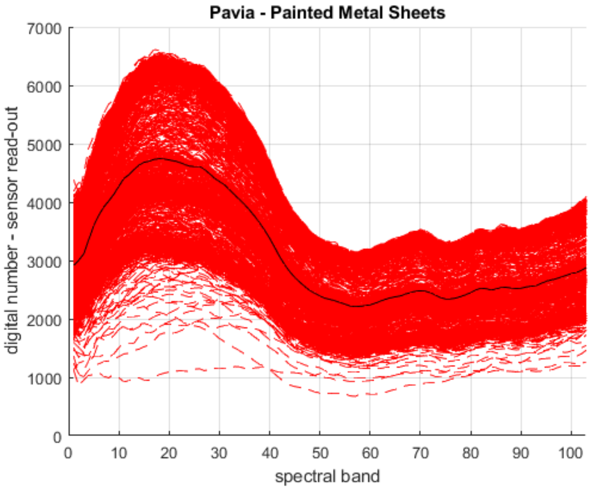


Figure 2.2: Spectral variance of the metal sheets in the Pavia scene depicted in figure 3.6.

2.1.1 Hyperspectral Image Acquisition

There exist several different ways to acquire hyperspectral data e.g. push-broom, wavelength scanning, snapshot, image slicing and Fourier transform infrared [6, 1]. However, within the field of remote sensing, the most commonly used method for acquiring a hyperspectral image is by using a pushbroom scanner. An illustration of the optical setup for a pushbroom scanner is given in figure 2.3.

A pushbroom scanner works by scanning each scene onto a slit entrance of a spectrometer. The collected light is then split into components by a dispersive element. The dispersed component is then projected onto a focal plane array, and in this array, the axis along the slit records spatial data and the axis perpendicular to the slit records the spectral data. This matrix then represents the spatial and spectral data of one line in the imaged scene. As time progresses, a pushbroom scanner will need to be in relative motion to cover a larger spatial area. Thus, through this way of operating the camera is especially well suited for applications such as airborne applications where a UAV is equipped with a hyperspectral camera. Additionally, it can be used for imaging from a USV or AUV, all relevant for the mission envisioned by The NTNU SmallSat Program [5]. Another advantage of the hyperspectral pushbroom scanner is, with a dispersive element of high quality e.g. diffraction gratings, the camera will be able to provide a high spectral resolution with high accuracy and precision.

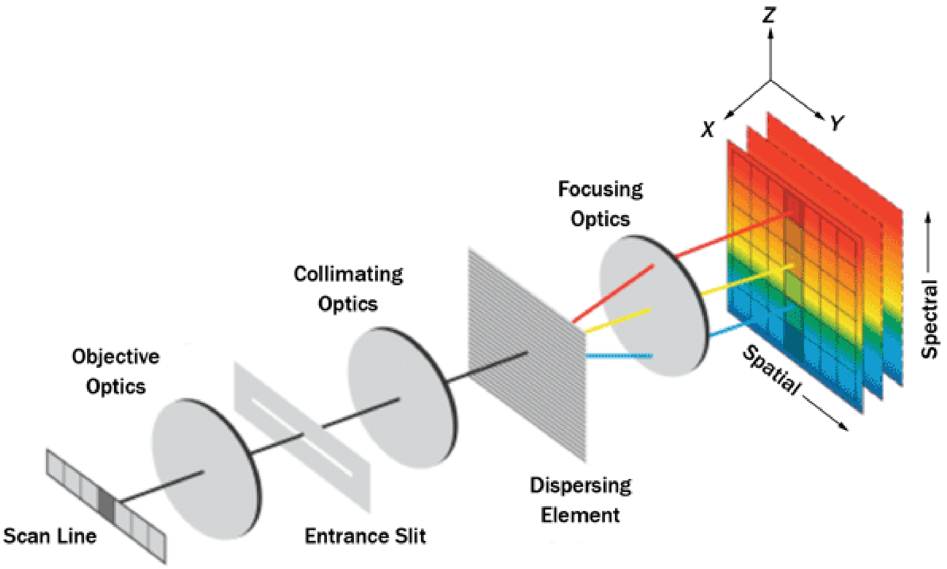


Figure 2.3: An illustration of a typical assembly of pushbroom optics. The image is taken from [2].

2.2 Data Models for Hyperspectral Data

An important element of hyperspectral imaging is data representation. This section and the following subsections is strongly influenced by [6] and [1].

A common approach to representing a hyperspectral data could be to use the original spatial extent of the image as two-dimensional, and the spectral information of the scene as a third dimension. In this way spatial information of the cube is intact, and all the data is represented in only three dimensions. It is also possible to represent the data cube as a matrix, where both spatial dimensions of an image are ordered along one dimension and the spectral information as a second dimension and the data cube can be reconstructed if the original ordering or the transformation approach of the spatial pixels is stored as metadata. presented in a more mathematical approach, the data cube, also depicted in figure 2.4:

$$\mathbf{X}_{Cube} = \begin{bmatrix} \mathbf{x_{1,1}} & \mathbf{x_{1,2}} & \cdots & \mathbf{x_{1,i}} \\ \mathbf{x_{2,1}} & \mathbf{x_{2,2}} & \cdots & \mathbf{x_{2,i}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x_{j,1}} & \mathbf{x_{j,2}} & \cdots & \mathbf{x_{j,i}} \end{bmatrix} \tag{2.1}$$

and the matrix

$$\mathbf{X}_{Matrix} = \begin{bmatrix} \mathbf{x_{1,1}} & \cdots & \mathbf{x_{1,i}} & \mathbf{x_{2,1}} & \cdots & \mathbf{x_{2,i}} & \cdots & \mathbf{x_{j,1}} \cdots & \mathbf{x_{j,i}} \end{bmatrix} \tag{2.2}$$

where i and j are the spatial extent of the data e.g. $\mathbf{x}_{50,200}$ is the pixel at row 50 column

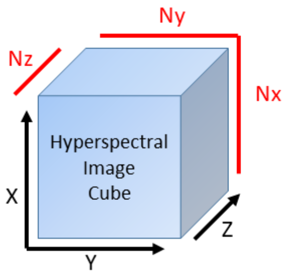


Figure 2.4: The different axis of a data cube. N_x, N_y and N_z represent the maximum number of rows, columns and spectral bands respectively.

200 or x-coordinate 50 and y-coordinate 200, and the spectral information is stored at every pixel in the following way $\mathbf{x} \in \mathbf{R}^k$, often shown as follows:

$$\mathbf{x_i} = \begin{bmatrix} L_i(\lambda_1) & L_i(\lambda_2) & \cdots & L_i(\lambda_k) \end{bmatrix}^T \tag{2.3}$$

where $L_i(\lambda_n)$ is the measurement of a spectral band centred at electromagnetic frequency λ_n. The measurement can be digital numbers, reflectance, emissivity or radiance depending on the application.

Furthermore, it is common to refer to the specific pixel in matrix representation by use of a single number, this transformation could be done as follows:

$$T_{number}(x, y) = x \times N_y + y \tag{2.4}$$

resulting in a one-dimensional indexation, the representation most often used in algorithmic descriptions, of $\mathbf{X}_{Matrix}$ as follows:

$$\mathbf{X}_{Matrix} = \begin{bmatrix} \mathbf{x_1} & \mathbf{x_2} & \cdots & \mathbf{x_N} \end{bmatrix} \tag{2.5}$$

where N is the maximum number of rows and columns multiplied, or the number of pixels in a scene.

Due to the nature of hyperspectral images containing vast amounts of data, different approaches will fit for different applications. In a general sense, there are two popular ways of representing hyperspectral data, a geometric approach, and a statistical approach.

2.2.1 Geometric Representation of Hyperspectral Data

A geometric representation of hyperspectral data is particularly useful when exploring such concepts as spectral similarity mixing and change transformation. For the geometric approach, it is natural to imagine the different spatial picture elements as vectors $\mathbf{x} \in \mathbf{R}^k$, where k is the number of spectral wavelengths that the instrument is able to perceive. From this representation of the data, it is trivial to extract certain powerful metrics and norms and to draw certain useful parallels. A widely used geometric concept to describe, characterize and compare different pixels is the Euclidean $\mathbf{L}_2$ norm:

$$d_{12}(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x_1} - \mathbf{x_2}\|_2 = \sqrt{(\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)} \tag{2.6}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are two pixels in the K-dimensional space. The resulting expression gives an indication of how far apart two points in the K-Dimensional space are.

Another useful geometrically fueled metric is the multidimensional spectral angle measurement, a geometric approach to enable angle comparison in higher dimensions than that of the spatial dimensions. It can be represented the following way:

$$\theta_{12} = \cos^{-1}(\frac{\mathbf{x}_1^T\mathbf{x}_2}{\sqrt{(\mathbf{x}_1^T\mathbf{x}_1)(\mathbf{x}_2^T\mathbf{x}_2)}}) \tag{2.7}$$

Furthermore, the geometrical representations of hyperspectral data are truly powerful when discussing endmembers and detection algorithms, the topic of section 2.3. Geometric concepts can quite effectively describe the expected transformation of the hyperspectral data by changes in environmental conditions, illumination or both. The following description is used in [6]:

$$\mathbf{x} = \mathbf{a} \circ \mathbf{p} + b \tag{2.8}$$

where $\mathbf{x}$ is the radiance measured at the sensor, $\mathbf{p}$ is the reflectance vector for the surface material, $\mathbf{a}$ represents the product of the atmospheric transmission and total dowelling radiance, $\mathbf{b}$ represents the path of radiance and $\circ$ is the Hadamard product.

This approach is best used to describe a linear mixing of constituents or endmembers, but it is still a powerful representation. It does however not perfectly hold up if the environmental or illumination changes are spatially varying, there exists nonlinear mixing as a result of adjacency effects, the temperature of the emissive data changes data changes from pixel to pixel or if the observed reflectance vector change between observations. However, it should be noted that the linear mixing model usually is a good approximation for many applications.

2.2.2 Statistical Representation of Hyperspectral Data

A statistical representation of hyperspectral data is particularly useful when modeling the inherent randomness of hyperspectral data resulting from sensor noise and the stochastic nature of real remotely sensed scenes.

In this representation, the measurement $\mathbf{x}$ is described as a multidimensional probability density function $p(\mathbf{x};\theta)$, where θ is a set of parameters saying something about the probability distribution. The $\mathbf{x}$ can be said to be composed of two main components, namely $\mathbf{s}$ and $\mathbf{n}$, which is the signal and noise respectively, as follows

$$\mathbf{x} = \mathbf{s} + \mathbf{n} \tag{2.9}$$

If $\mathbf{s}$ is a deterministic signal related to the ideal spectrum, and the noise represented by $\mathbf{n}$ can be said to be zero-mean, spatially independent normally distributed noise with wavelength independent variance, the probability density function can be written as follows:

$$p(\mathbf{x};\theta) = \frac{1}{(2\pi)^{K/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\big(- \frac{1}{2}[\mathbf{x} - \mu]^T \mathbf{\Sigma}^{-1} [\mathbf{x} - \mu]\big) \tag{2.10}$$

where the θ parameters are the mean μ and the covariance matrix $\mathbf{\Sigma} = \sigma\mathbf{I}$, given the condition described for $\mathbf{n}$. This again gives the shorthand notation of $\mathbf{x} \sim N(\mathbf{s}, \sigma^2\mathbf{I})$

However, due to the aforementioned inherent stochastic nature of real scenes as a result of man-made objects and natural background alike, the normal distribution usually does not hold to describe the complexity of a scene. The main idea of representing the data in a statistical manner is still useful, as will become apparent in the subsequent sections.

If one instead treats the probability density function parameter θ as an unknown, where the $\mathbf{x_i}$ data of an image is presumed that each spectral measurement is independent and identically distributed in a statically sense. From this, it follows that the maximum likelihood function L for a scene or a set of collected data used to determine θ can be written as follows:

$$L(\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}; \theta) = \prod_{i=1}^{N} p(\mathbf{x_i}; \theta) \tag{2.11}$$

Thus, the parameters μ and $\mathbf{\Sigma}$ can be solved analytically, and their estimations can be calculated as follows:

$$\mathbf{m} \equiv \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x_i} = \frac{1}{N}\mathbf{Xu} \tag{2.12}$$

where $\mathbf{u}$ is a vector of all ones.

$$\mathbf{C} \equiv \hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} [\mathbf{x_i} - \mathbf{m}][\mathbf{x_i} - \mathbf{m}]^T = \frac{1}{N}\mathbf{XX}^T - \mathbf{mm}^T \tag{2.13}$$

This gives the correlation matrix as follows:

$$\mathbf{R} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x_i}\mathbf{x_i}^T = \frac{1}{N}\mathbf{X}\mathbf{X}^T = \mathbf{C} + \mathbf{m}\mathbf{m}^T \tag{2.14}$$

Given a sufficiently large sample size of data the estimation for mean $\hat{\mu}$ and correlation $\hat{\mathbf{\Sigma}}$ can be regarded as unbiased.

The representation given over are useful ways to describe a scene or a collection of hyperspectral data in a statistical manner.

2.2.3 Principal Component Analysis

Principal Component Analysis or PCA is a much-used method within many fields and can be said to have its origin at the start of the 20th century [16]. It is known under several different names, and has been reinvented several times. Within the field of hyperspectral imaging, it might be known as *Karhunen-Loeve* transformation or simply PCA [6]. Regardless of the origin of the algorithm, the procedure stays the same in principle. The PCA transformation seeks to diagonalize the sample covariance matrix $\mathbf{C}$, by determining the eigenvalues of $\mathbf{C}$ as follows:

$$det(\mathbf{C} - \sigma^2\mathbf{I}) = 0 \tag{2.15}$$

Given that the $\mathbf{C}$ matrix has full rank i.e. no eigenvalue can be zero, the eigenvalues σ^2 found will represent the variance for a particular eigenvector. Furthermore, the principal direction where the spectral correlation is removed is given by

$$\mathbf{C}\mathbf{v}_j = \sigma^2\mathbf{v}_j \tag{2.16}$$

Where $\mathbf{v}_j$ is an eigenvector of unit length and the j^{th} eigenvector corresponding to the j^{th} eigenvalue. If then all the eigenvalues were ordered in a diagonal matrix $\mathbf{D}$ were the largest eigenvalue came first, and the smallest eigenvalue came last, the following relationship can be stated:

$$\mathbf{CV} = \mathbf{VD} \implies \mathbf{C} = \mathbf{VDV}^T \tag{2.17}$$

Where $\mathbf{V}$ is the corresponding eigenvectors $\mathbf{v}_j$ ordered in a descending manner as well, same as with the $\mathbf{D}$ matrix, following the fact that $\mathbf{V}$ is an orthogonal unitary matrix. Thus the transformation from the original projection into an orthogonal basis where the bands are uncorrelated and ordered according to the variance of in the original data:

$$\mathbf{Z} = \mathbf{V}^T\mathbf{X} \tag{2.18}$$

Where $\mathbf{Z}$ then represents a principal component data matrix transformation from the original coordinate space into an orthogonal coordinate space. To perform the desirable dimensionality reduction i.e. represent the original data using fewer samples or variables, with K as the full dimensional space, you would choose the $k \leq K$ largest corresponding eigenvectors and eigenvalues. The inverse transformation would then yield the original data with reduced noise ideally. Figure 2.5 is a more visual presentation of the effects of PCA on hyperspectral images. The figure shows to the left a false-color image taken by the HICO sensor over Lake Erie, the following 4 images are the top 4 Principal Components of that image, and the last 4 images are components 47 to 50.

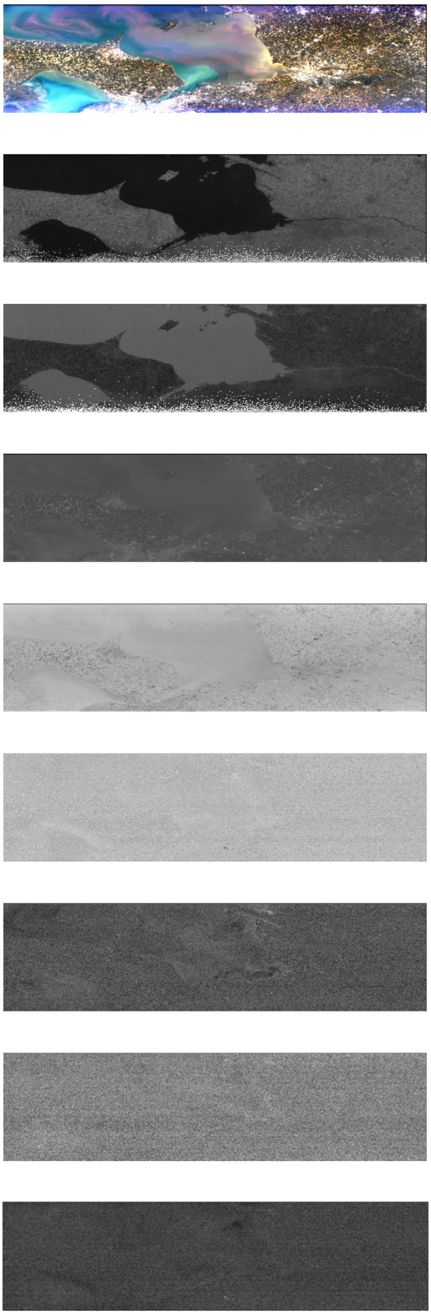


Figure 2.5: An illustration on how components propagate when using PCA.

2.2.4 Maximum Noise Fraction

Maximum noise fraction, also known as minimum noise fraction or noise-adjusted principal component analysis was first described in [17], and has since been used in many hyperspectral and multispectral image analysis approaches, and the principal idea has been developed many times over since the initial publication. The idea of MNF is to sort the components based on noise, or rather estimated noise, than variance as is the objective of PCA. At the baseline it is common to assume independence between the signal covariance and the noise covariance as such:

$$\mathbf{\Sigma} = \mathbf{\Sigma}_s + \mathbf{\Sigma}_n \tag{2.19}$$

where $\mathbf{\Sigma}$ denotes the total covariance of the samples, whilst $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_n$ are the covariance of the actual desired signal and the noise, respectively. Thus the objective of MNF could then be described as maximizing the SNR value, whilst still decorrelating the covariance matrix $\mathbf{\Sigma}$, as follows:

$$SNR = \frac{\mathbf{v}_j^T \mathbf{\Sigma}_s \mathbf{v}_j}{\mathbf{v}_j^T \mathbf{\Sigma}_n \mathbf{v}_j} = \frac{\mathbf{v}_j^T \mathbf{\Sigma} \mathbf{v}_j}{\mathbf{v}_j^T \mathbf{\Sigma}_n \mathbf{v}_j} - 1, \qquad \mathbf{v}_i^T \mathbf{\Sigma} \mathbf{v}_j = 0, i \neq j \tag{2.20}$$

As the noise cannot be assumed to be known beforehand, it has to be estimated. There are several approaches to perform this estimation [17, 18, 19], and how this estimation is performed is the main difference between MNF methods. The below is one of the original ones described in [17].

Green estimates the noise under the assumption that all the data or the portion of the data used for noise estimation is of a homogeneous region. Thus by calculating the difference between signals, using notation found in (2.5), as follows:

$$\mathbf{X}_n = \begin{bmatrix} \mathbf{x}_1 - \mathbf{x}_2, & \mathbf{x}_2 - \mathbf{x}_3, & \cdots, & \mathbf{x}_{N-1} - \mathbf{x}_N \end{bmatrix} \tag{2.21}$$

The covariance and its singular value decomposition of the noise can be found in the following way:

$$\mathbf{C}_n = \mathbf{X}_n^T \mathbf{X}_n \qquad \mathbf{C}_n = \text{svd}(\mathbf{C}_n) = \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T \tag{2.22}$$

It is important to decorrelate the noise data, also known as whitening. The process of whitening or making the covariance uncorrelated and of unit variance can be performed as follows:

$$\mathbf{X}_w = \mathbf{X} \mathbf{U}_n \mathbf{S}_n^{-1/2} \qquad \mathbf{X}_w = \text{svd}(\mathbf{X}_w) = \mathbf{U}_w \mathbf{S}_w \mathbf{V}_w^T \tag{2.23}$$

Which again results in the following transformation matrix:

$$\mathbf{V}_g = \mathbf{U}_n \mathbf{S}_n^{-1/2} \mathbf{V}_w \tag{2.24}$$

Lastly yielding the following relationship:

$$\mathbf{Z}_g = \mathbf{V}_g^T \mathbf{X} \tag{2.25}$$

Figure 2.5 is a more visual presentation of the effects of MNF, using Green's method for noise estimation on hyperspectral images from HICO.

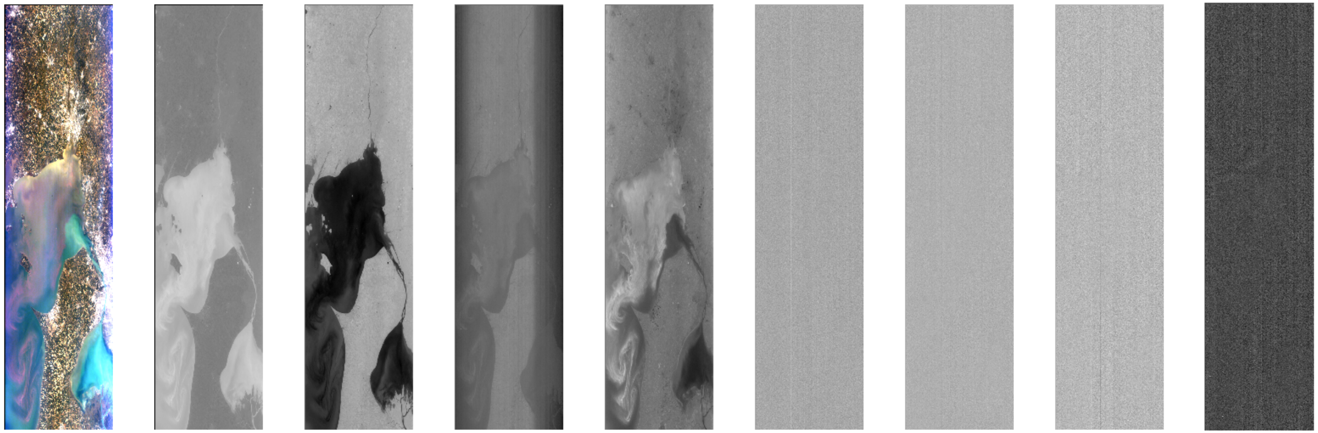


Figure 2.6: This figure shows to the left a false color image taken by the HICO sensor over lake Erie, the subsequent 4 images are the top 4 noise adjusted Principal Components of that image, the last 4 images are components 47 to 50. This figure is intended as an illustration on how components propagate when using MNF

2.2.5 Independent Component Analysis

Independent component analysis or ICA is an alternative to MNF and PCA for dimensionality reduction. Whilst PCA and MNF are based on first and second order statistics i.e. mean and variance, ICA uses higher-order statistics to find the underlying structures of the data. Through higher-order statistics the ICA algorithms are able to portray nonlinearities in the data, obtaining in a more deterministic way the same insight that can be provided by using multi-layer neural networks [20]. With ICA one seeks a linear decomposition of the spectra, on a set of normalized basis vectors, such that the resulting transformed images are uncorrelated and statistically independent. Thus, this approach should fit well when the normal distribution assumption in PCA is no longer valid[6]. ICA, as it is described here, is based on the descriptions found in [20, 6, 21].

The principle behind ICA can be described similarly to how PCA is performed. Using the same notation as previously we have the following linear decomposition:

$$\mathbf{x} = \mathbf{V}_i \mathbf{z} \tag{2.26}$$

Where $\mathbf{V}_i$ is a transformation matrix with the principal vector directions and $\mathbf{z}$ is a vector in the transformed space, where all the bands are statistically independent. With respect to PCA the columns of $\mathbf{V}_i$ are the eigenvectors found from the covariance matrix as described above. In the case of ICA, equation 2.26 can be rewritten as

$$\mathbf{z} = \mathbf{W}\mathbf{x} \tag{2.27}$$

Where $\mathbf{W}$ is the reverse transformation of $\mathbf{V}_i$, and is referred to as the separating matrix. In this reverse transformation, the $\mathbf{W}$ would represent all the different unique signals in the dataset, ideally. As ICA is used in many different fields, a unique signal could be defined as different sound sources in sound processing, features within feature extraction, and to separate different endmember sources in a hyperspectral image [20]. The linear combination of unique signals that would be contained in the transformed variable $\mathbf{z}$ would be statistically independent. That is, the joint probability density function, p_{12}, and the marginal probability density functions, p_1 and p_2, can be written in the following manner.

$$p_{12}(z_1, z_2) = p_1(z_1)p_2(z_2) \tag{2.28}$$

A common way to measure the dependence of K random variables is mutual information, often denoted as I

$$I(z_1, z_2, \cdots, z_K) = \sum_{k=1}^{K} H(z_k) - H(\mathbf{z}) \tag{2.29}$$

In equation 2.29 $H(\mathbf{z})$ is the differential entropy, defined as follows

$$H(\mathbf{z}) = -\int p(\mathbf{z}) \log p(\mathbf{z}) d\mathbf{z} \tag{2.30}$$

Thus, from the linear representation given in equation 2.27, equation 2.29 can be rewritten as follows:

$$I(z_1, z_2, \ddots, z_K) = \sum_{k=1}^{K} H(x_k) - H(\mathbf{z}) - \log|\text{det}\mathbf{W}| \tag{2.31}$$

Furthermore, if the subspace representation is such that the z_k components can be said to be decorrelated and scaled to unit variance, the final form of equation 2.29 is given as:

$$I(z_1, z_2, \ddots, z_K) = C_k - \sum_{k=1}^{K} J(z_k) \tag{2.32}$$

Where C is a constant and $J(\mathbf{z})$ is the negentropy i.e. the difference in entropy between the estimated distribution and the Gaussian distribution, given that the two distributions have the same mean and variance. The negentropy, $J(\mathbf{z})$ with $\mathbf{z}_{normal}$ as a multivariate normal random vector process with the same covariance matrix as $\mathbf{z}$, is defined as:

$$J(\mathbf{z}) = H(\mathbf{z}_{normal}) - H(\mathbf{z}) \tag{2.33}$$

From equation 2.32 it can be deducted [6] that, as the objective is to maximize the negentropy function when searching for a set of orthogonal vector directions for the separation matrix $\mathbf{W}$, the resulting transformation minimizes the statistical dependence between components.

There are several methods on how to perform this numerically with real-world data. The three most common can be said to be Fixed-point ICA (FastICA), Infomax and Joint Approximate Diagonalization of Eigenmatrices (JADE). Each method has different advantages and drawback, and thus they are suited for different types of applications [20].

The implementation used in this thesis is the JADE approach. The JADE ICA estimation approach is a tensorial method for calculating the components [20]. As a preliminary step the data is whitened by principal component transform i.e. JADE is performed on uncorrelated data that have a unit variance. Whiting is an important prepossessing step in independent component analysis [6]. The JADE algorithm was selected due to the fact that it does not have convergence problems, as it does not rely on gradient searches. Memory and time usage was not considered when implementing ICA for the analysis. The four first cumulants as defined for tensorial probability mathematics are mean, variance, skew and kurtosis, respectively. The JADE algorithm will seek to optimize the second and fourth cumulants from the data, that is the variance σ^2 and the kurtosis κ of the data, defined as follows:

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mu})^2 \qquad \kappa = \frac{1}{N}\sum_{i=1}^{N}\big(\frac{\mathbf{x}_i - \hat{\mu}}{\sigma}\big)^4 - 3 \tag{2.34}$$

Whilst the variance σ^2 gives a measurement of how much the data varies from the mean, the kurtosis κ gives an indication of how the extremities of the data are distributed. The constant number 3 is the kurtosis of a perfectly normally distributed data set.

The following description of the JADE algorithm is strongly influenced by [20]. The cumulant tensor $cum(\cdot,\cdot,\cdot,\cdot)$, a matrix space consisting of four vectors representing the four different cumulants. This cumulant tensor will have K entries, where K in the hyperspectral domain will correspond to the number of wavelengths. with the $cum(\cdot,\cdot,\cdot,\cdot)$ defined as follows for the zero mean case:

$$\begin{aligned} cum(x_i, x_j, x_k, x_l) &= E[x_i x_j x_k x_l] \\ -E[x_i x_j]E[x_k x_l] \\ -E[x_i x_k]E[x_j x_l] \\ -E[x_i x_l]E[x_j x_k] \end{aligned} \tag{2.35}$$

The matrix space then consists of $K \times K$ matrices, instead of a space of K-dimensional vectors.

Given the assumptions made in the ICA model the rows of the separating matrix $\mathbf{W}$, denoted $\mathbf{w}_m$ for row m, form the eigenmatrices $\mathbf{M} = \mathbf{w}_m \mathbf{w}_m^T$. A natural choice for the matrix $\mathbf{M}$ is the eigenmatrices of the cumulant tensor.

Furthermore, The space of matrices will then be a linear space of dimension $K \times K$, and a transformation matrix $\mathbf{F}_{ij}$, with m_{kl} being the elements of $\mathbf{M}$ being transformed, could be defined as follows

$$\mathbf{F}_{ij}(\mathbf{M}) = \sum_{kl} cum(\mathbf{x}_\mu, \mathbf{x}_\sigma, \mathbf{x}_\gamma, \mathbf{x}_\kappa) \tag{2.36}$$

The ordering of the variables that are sent into equation 2.35 does not change the outcome, thus the space is a symmetric linear operator, the cumulant tensor has an eigenvalue decomposition, and an eigenmatrix $\mathbf{M}$ of the tensor is defined as follows

$$\mathbf{F}_{ij}(\mathbf{M}) = \lambda \mathbf{M}_{ij} \tag{2.37}$$

Thus, we have a set of just K matrices with all the relevant information on the cumulants, as they span the same subspace as the cumulant tensor. Assuming the ICA model the matrix $\mathbf{F}$ will be a linear combination of terms in the form of $\mathbf{w}_i \mathbf{w}_i^T$. Thus, in the ideal case, $\mathbf{W}\mathbf{F}(\mathbf{M}_i)\mathbf{W}^T$ is a diagonal matrix. JADE is in principle performed by maximization of the expression in equation 2.38, where $||\text{diag}(\cdot)||^2$ is the sum of squares of the diagonal

$$\mathcal{J}_{JADE}(\mathbf{W}) = \sum_i ||\text{diag}(\mathbf{W}\mathbf{F}(\mathbf{M}_i)\mathbf{W}^T)||^2 \tag{2.38}$$

Figure 2.7 is a more visual presentation of the effects of ICA on hyperspectral images.

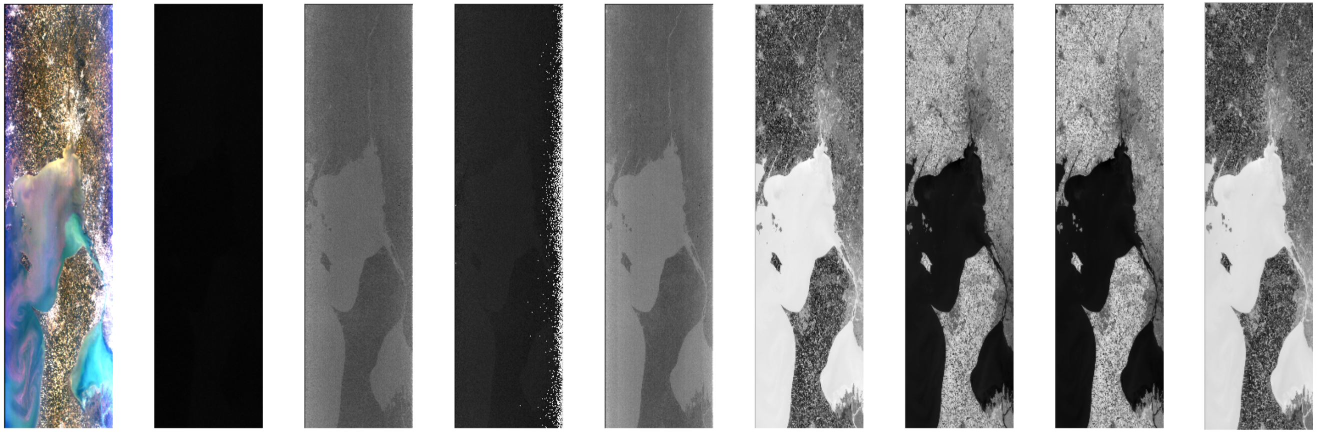


Figure 2.7: This figure shows to the left a false color image taken by the HICO sensor over lake Erie, the subsequent 4 images are the top 4 Independent Components of that image, the last 4 images are components 47 to 50. This figure is intended as an illustration on how components propagate when using ICA

2.3 Target Detection Algorithms

There are several ways to utilize acquired hyperspectral images in many different fields. Within the field of remote sensing, there are usually two approaches on how to gather information from the captured scenes[6]. Classification i.e. labeling the different pixels of an image to be of a certain constituent, and Detection i.e. determining whether or not something in the scene can be regarded as an anomaly or if a certain constituent is in the scene or not.

This thesis will focus on hyperspectral detection algorithms. These algorithms focus on detection of a known signature in a given image where a single pixel does not necessarily contain the signature of one constituent. The section is mainly based on [6, 9, 3]. To successfully perform target detection in hyperspectral images it is important to process the image in a way that makes the signature source and the image cube be in the same reference space wrt. magnitude, response, etc. That is, an image cube given with reflectance values should use a signature given in reflectance values to perform detection. From the high spectral resolution typically found in hyperspectral imaging, it is often possible to identify spatially resolved and unresolved elements of interest based on a known spectral signature [9].

The methods discussed in this thesis fall within the category of signature matched detection. The following descriptions will assume a single deterministic signature $\mathbf{s}$ to be the spectrum that represents the signature normalized to the data units in the given image or cube under inspection to be looked for.

in the following sections H_0 will represent a pixel which does not contain the signature being looked for, and H_1 will represent the signature being looked for. In this context a false alarm is defined as a false positive i.e. $p(H_1|H_0)$. All the detection algorithms will give a probability of target, and in practical applications, a detection threshold needs to be selected. This threshold should ensure that a desired rate of false alarms for a given application is achieved.

In figure 2.8 a depiction of threshold selection, false alarms and probability distribution is given. From the illustration it is clear that to select a threshold may be an iterative process, and is very dependent on the cost of false alarms versus missed targets. If the sample space does not contain an abundance of targets, a lower threshold would be natural, and vice versa.

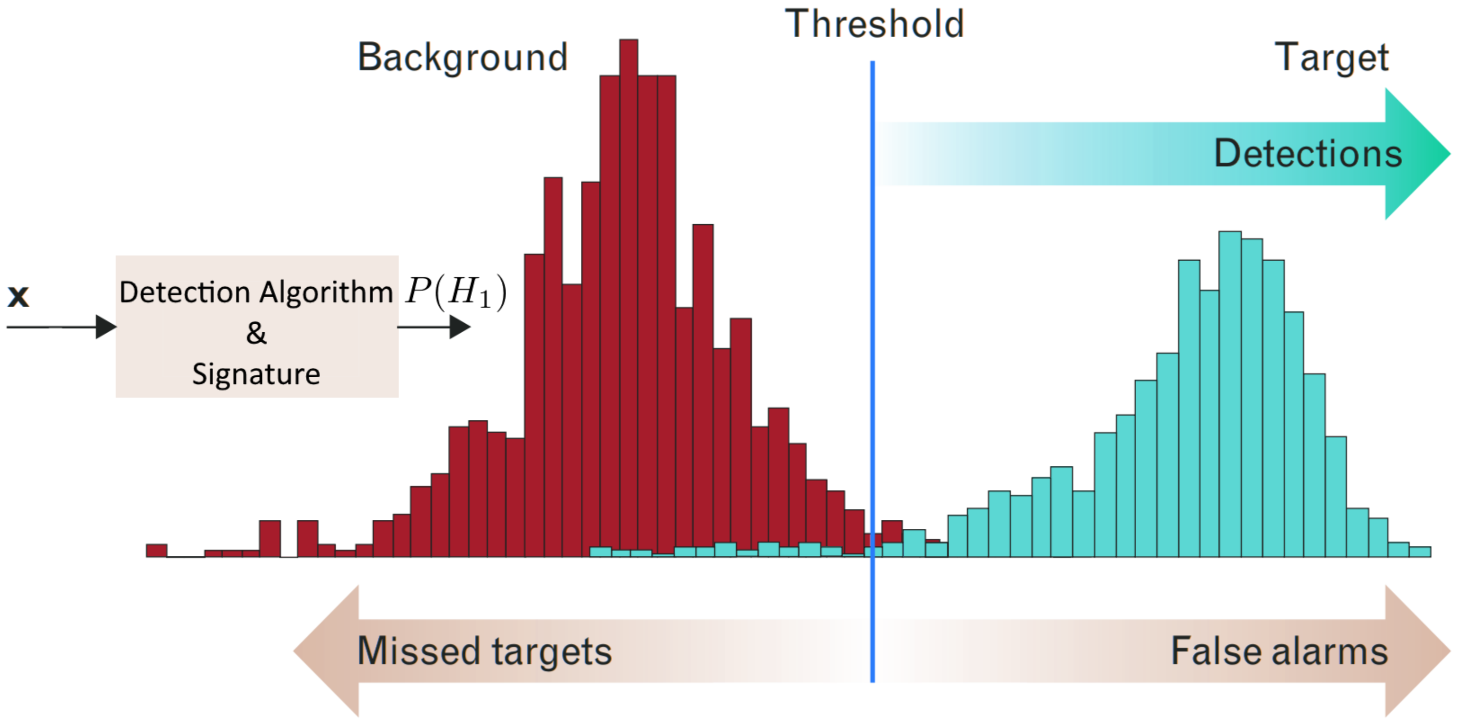


Figure 2.8: The figure is adapted and modified from [3]. It is a visualization of threshold selection wrt. detection statistics.

2.3.1 Spectral Angle Mapper

The detection problem can be represented in the following way:

$$H_0 : \mathbf{x} = \mathbf{b} \qquad H_1 : \mathbf{x} = \alpha\mathbf{s} + \mathbf{b} \tag{2.39}$$

where $\mathbf{b}$ is both the noise and the background clutter combined, α is an unknown parameter that represents the strength of the sought after signal $\mathbf{s}$. Variation in α can due to illumination variation or subpixel mixing between the target and the background clutter. The background model i.e. the model for the parameter $\mathbf{b}$ can take many forms, but in the following the descriptions, unless otherwise is stated, will be assumed to have a zero-mean, be normally distributed and variance σ^2. That is $\mathbf{b} \sim N(0, \sigma^2\mathbf{I})$.

Thus the unknown parameters to be estimated are σ and α. From a maximum likelihood estimation point of view, the normalized projection for α is given as:

$$\hat{\alpha} = \frac{\mathbf{s}^T\mathbf{x}}{\mathbf{s}^T\mathbf{s}} \tag{2.40}$$

The logarithmic form of the generalized likelihood test can be deemed from equation (2.40) as follows:

$$r(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{x}}{\sigma^2} - \frac{(\mathbf{x} - \hat{\alpha}\mathbf{s})^T(\mathbf{x} - \hat{\alpha}\mathbf{s})}{\sigma^2} = \frac{2\hat{\alpha}\mathbf{s}^T\mathbf{x} - \hat{\alpha}^2\mathbf{s}^T\mathbf{s}}{\sigma^2} = \frac{1}{\sigma^2}\frac{(\mathbf{s}^T\mathbf{x})^2}{(\mathbf{s}^T\mathbf{s})} \tag{2.41}$$

Then to be able to obtain the desired false alarm rate the noise or clutter, depicted here as σ needs to be well estimated. σ is assumed to vary spatially across the image, and a simple much-used approach is to estimate σ as the square of the spectrum magnitude i.e. $\mathbf{x}^T\mathbf{x}$. With this applied to equation (2.41) the following metric for similarity is obtained:

$$r_{sam}(\mathbf{x}) = \frac{(\mathbf{s}^T\mathbf{x})^2}{(\mathbf{s}^T\mathbf{s})(\mathbf{x}^T\mathbf{x})} \tag{2.42}$$

2.3.2 Constrained Energy Minimization

As the spectral angle mapper has its limitations in terms of how the variance in the data is assumed i.e. the normal model is an underlying assumption for background noise and clutter, there may be performance improvements to gather from a better characterization of the noise and clutter in a given image [3].

The Constrained Energy Minimization approach is a method used to tackle this problem. It tries to utilize the data available in an image to estimate the given noise and clutter of the background.

As a metric to estimate the probability of target can be a linear operation, the following equation is a starting point:

$$r(\mathbf{x}) = \mathbf{h}^T\mathbf{x} \tag{2.43}$$

where $r(\mathbf{x})$ is the detector and $\mathbf{h}^T$ is the linear operation to be performed to better separate background and target.

It has proven to be a valuable approach to try to optimize the target detection performance by minimizing the background energy. Given that the background energy E is defined in the following way:

$$\begin{aligned} E &= \frac{1}{N}\sum_{i=1}^{N} r^2(\mathbf{x}_i) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{h} \\ &= \mathbf{h}^T(\frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i\mathbf{x}_i^T)\mathbf{h} \\ &= \mathbf{h}^T\mathbf{R}\mathbf{h} \end{aligned} \tag{2.44}$$

where $\mathbf{R}$ is the sample correlation matrix derived in equation (2.14). Furthermore, that the perfect signature results in unity i.e. $\mathbf{x} = \mathbf{s}$ gives $\mathbf{h}^T\mathbf{s} = 1$. Thus the optimization problem can be formulated as follows

$$\min_{\mathbf{h}} \mathbf{h}^T\mathbf{R}^{-1}\mathbf{h} \qquad \text{s.t.}\mathbf{h}^T\mathbf{s} = 1 \tag{2.45}$$

which leads to the following solution:

$$\mathbf{h} = \frac{\mathbf{R}^{-1}\mathbf{s}}{\mathbf{s}^T\mathbf{R}^{-1}\mathbf{s}} \tag{2.46}$$

and when inserting equation 2.46 into equation 2.43 we get the following detection algorithm:

$$r_{cem}(\mathbf{x}) = \frac{\mathbf{s}^T\mathbf{R}^{-1}\mathbf{x}}{\mathbf{s}^T\mathbf{R}^{-1}\mathbf{s}} \tag{2.47}$$

2.3.3 Adaptive Cosine Estimator

The additive model depicted in equation (2.39) is limited due to how the background clutter $\mathbf{b}$ is modeled, and an alternative model, used in the derivation of the ACE detector, is given in equation (2.48)

$$H_0 : \mathbf{x} = \beta\mathbf{b} \qquad H_1 : \mathbf{x} = \alpha\mathbf{s} + \beta\mathbf{b} \tag{2.48}$$

The newly introduced parameter β is an unknown scaling factor of the background noise and clutter. This model better compensates for subpixels i.e. when the desired target is only a part of the pixel. The background clutter and noise, before whitening and mean centering, takes the form of $\mathbf{b} \sim N(\mu, \sigma)$. If one were to mean center and whiten the data, and in the process also mean center the signature, the $\mathbf{b}$ parameter would then be a zero mean, white random vector process.

Thus, by maximizing the joint probability density function of the data under inspection, the parameters for the null and one hypothesis in equation 2.48 would then become:

$$\hat{\mathbf{\Sigma}}_0 = \frac{1}{N+1}\big(\frac{1}{\beta_0^2}\mathbf{x}\mathbf{x}^T + N\hat{\mathbf{\Sigma}}\big) \tag{2.49}$$

$$\hat{\mathbf{\Sigma}}_1 = \frac{1}{N+1}\big(\frac{1}{\beta_1^2}(\mathbf{x} - \alpha\mathbf{s})(\mathbf{x} - \alpha\mathbf{s})^T + N\hat{\mathbf{\Sigma}}\big) \tag{2.50}$$

$$\hat{\beta}_0^2 = \frac{N-K+1}{NK}\big(\mathbf{x}^T\hat{\mathbf{\Sigma}}^{-1}\mathbf{x}\big) \tag{2.51}$$

$$\hat{\beta}_1^2 = \frac{N-K+1}{NK}\big((\mathbf{x} - \alpha\mathbf{s})^T\hat{\mathbf{\Sigma}}^{-1}(\mathbf{x} - \alpha\mathbf{s})\big) \tag{2.52}$$

The $\hat{\mathbf{\Sigma}}$ is the same as the sample covariance matrix given in equation (2.13) with N training spectra. This detector is known as the Adaptive Cosine Estimator or ACE for short.

$$r_{ACE}(\mathbf{x}) = \frac{(\mathbf{s}^T\hat{\mathbf{\Sigma}}^{-1}\mathbf{x})^2}{(\mathbf{s}^T\hat{\mathbf{\Sigma}}^{-1}\mathbf{s})(\mathbf{x}^T\hat{\mathbf{\Sigma}}^{-1}\mathbf{x})} \tag{2.53}$$

2.3.4 Orthogonal Subspace Projection

The Orthogonal Subspace Projection algorithm (OSP), is based on a subspace model for the background data as opposed to a statistical model The detection problem hypothesis is derived as follows:

$$H_0 : \mathbf{x} = \mathbf{B}\beta + \mathbf{n} \qquad H_1 : \mathbf{x} = \alpha\mathbf{s} + \mathbf{B}\beta + \mathbf{n} \tag{2.54}$$

In this model the $\mathbf{B}$ represent the background subspace, β is the background basis coefficient, α is an unknown scalar similar to the one found in equation (2.40), and $\mathbf{n}$ is a noise vector process with zero-mean that is spectrally independent, normally distributed and has a standard deviation of σ_n.

If $\mathbf{B}$ is computed from the data under inspection, and $\mathbf{s}$ is the known signature we are looking for, the subspace projection $\mathbf{P_B}$, which can then be formulated as:

$$\mathbf{P_B} = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T \tag{2.55}$$

The orthogonal component of equation (2.55) then becomes

$$\mathbf{P}_\mathbf{B}^\perp = \mathbf{I} - \mathbf{P_B} \tag{2.56}$$

From this, it can be shown that the scalar $\hat{\alpha}$ found in equation (2.40) based on the maximum likelihood estimation takes the following form.

$$\hat{\alpha} = \frac{\mathbf{s}^T\mathbf{P}_\mathbf{B}^\perp\mathbf{x}}{\mathbf{s}^T\mathbf{P}_\mathbf{B}^\perp\mathbf{s}} \tag{2.57}$$

From this result, it is trivial to derive the Orthogonal Subspace Projection detection algorithm as:

$$r_{OSP}(\mathbf{x}) = \mathbf{s}^T\mathbf{P}_\mathbf{B}^\perp\mathbf{x} \tag{2.58}$$

The N-FINDR Endmember Extraction Algorithm

The OSP detector needs an estimate of the background subspace $\mathbf{B}$ in addition to the known signature $\mathbf{s}$ and the pixel under inspection $\mathbf{x}$. In this thesis, the background has been estimated using an endmember extraction algorithm known as N-FINDR [22]. There exist many further developments on the original method, but in the following work, the algorithm as it is described in [22].

The N-FINDR algorithm assumes that if an image consists of k_{bands} spectral bands the total number of pure pixels, constituents or endmembers in a scene k_{dim} will be lower than the number of bands i.e. $k_{dim} < k_{bands}$. To estimate these k_{dim} number of endmembers the algorithm expands a simplex of the virtual estimated endmember space inside the data, initially starting with a set of random potential endmembers. For each pixel and each endmember, the endmember is replaced with the spectrum of the pixel and the volume is calculated again. If the volume increases, the spectrum of endmember will be replaced by the new pixel. This procedure is repeated until convergence.

Displayed in more mathematical terms, as depicted in [22], the algorithm assumes that each pixel consists of a linear combination of endmember spectra

$$\mathbf{x}_{ij} = \sum_k e_{ik} c_{kj} + \mathbf{n}, \qquad \sum_k c_{kj} = 1 \tag{2.59}$$

where $\mathbf{x}_{ij}$ is the i-th band and the j-th pixel, e_{ik} is the i-th band of the k-th endmember, whilst c_{kj} is the mixing proportions of the j-th pixel from the k-th endmember, and lastly $\mathbf{n}$ is, of course, a normally distributed vector process representing noise and to be of a rather small magnitude compared to the rest of the signal. As the pure pixels i.e. pixels that mainly consists of one endmember $c_{kj} \approx 1$, define the vertices in the simplex created inside the data. In other words, the algorithms assume that there exists at least one pure pixel for every endmember to be found in the dataset.

In order for the volume of the simplex to be determined the scene must be reduced to be one less dimension than that of the number of endmembers to be found. In this thesis, the principal component analysis is used to reduce the dimension when applying the N-FINDR algorithm. According to [22] theoretically the optimal algorithms to perform this task should be maximum noise factoring, but in practice, the two different approaches appear to work equally well.

The volume is determined in the following way

$$V(\mathbf{E}) = \frac{1}{(l-1)!} abs(|\mathbf{E}|) \qquad \mathbf{E} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vec{e}_1 & \vec{e}_2 & \cdots & \vec{e}_l \end{bmatrix} \tag{2.60}$$

where V is the volume of the simplex formed by the endmember estimates, $(l-1)$ is the number of dimensions used by the data, and $\vec{e}_i$ is a column vector containing the spectra of the i-th endmember. Each Pixel in the scene is tried as an endmember and if the volume increases the endmembers are replaced. In a perfect dataset, the data would be of a convex form and there would be no local maximum, however as most real-world data cannot be said to be perfect the algorithms usually runs several times and returns the results that gave the largest volume.

With $\mathbf{C}$ as the endmember composition of pixel $\mathbf{P}$, the resulting unmixing of a pixel is then performed as follows

$$\mathbf{C} = (\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T\mathbf{P} \tag{2.61}$$

In the implementation of the OSP detector N-FINDR, in combination with SAM to remove the endmember most similar to the signature, was used to estimate the background endmembers. This is one of several ways to estimate the background endmembers needed in the OSP algorithm. Other popular methods used to estimate the background are ICA and Automatic Target Generation Procedure. The results would undoubtedly be different with a different method for estimating the background. To what extent this choice has affected the performance of the OSP detector is difficult to say, and it is not emphasized in the analysis.

Chapter 3

Data Description & Methods

3.1 Instruments

In this thesis three different scenes with ground truth are discussed. On all three scenes a different hyperspectral imager has been used. below follows a short description of the instruments. Their respective documentation is given in appendix B.

Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) is a well-known instrument within the remote sensing community, used for several reference scenes, in addition to the one given here. The sensor captures 224 well calibrated contiguous spectral bands with wavelengths ranging from 400 to 2500 nm AVIRIS has been flown on four different aircraft platforms as of 2018. The instrument has been used with success over areas such as North America, Europe, portions of South America, and Argentina. The main goal of the AVIRIS project is to identify, measure, and monitor constituents on the Earth's surface and its atmosphere. The research conducted with the AVIRIS instrument is mainly concerned with better understanding natural phenomena such as the global environment and climate change. Both the Salinas and the Indian Pines scene were acquired using a version of the AVIRIS sensor [23].

Reflective optics system imaging spectrometer (ROSIS-3) is an imaging spectrometer designed specifically for monitoring of ocean color and other oceanographic phenomena. With its high spectral resolution of more than 5 nm, the instrument is also useful wrt. too many new airborne applications Such application as vegetation monitoring, crop health observation, monitoring of the atmosphere and many more. The instrument was jointly developed by MBB GKSS and DFVLR, and the instrument concept and the scope of applications were developed in cooperation with ESA earth observation program and the NASA. The Pavia university scene, is captured using the ROSIS-03 sensor, which, in the given scene, had a spectral range from 430 nm to 860 nm, spread across 115 spectral bands. In the publicly available data set of the Pavia university scene, 12 bands are removed due to excessive noise [24].

The Ecotone scientific UHI is a hyperspectral imager designed for use underwater. This is able to give a greater detail in remote sensing of the seabed, previously only available through airborne platforms. With the Ecotone scientific UHI, the field of hyperspectral imaging is becoming more accessible for subsea applications. The instrument is capable of capturing wavelengths in the entire visible color spectrum, according to the provider. Data from the UHI can in return be used to detect biological and chemical of interest that occur on the seafloor. The ECOTONE scientific UHI used here has 86 spectral bands evenly distributed over the range 400 to 700 nm.

3.2 Datasets

In this thesis, several different types of data sets were chosen to perform the analysis. The different environmental settings are used to validate the proposed methods, and to illuminate their shortcomings. In this section, the basis for how these datasets were chosen will be discussed, and their origin will be made clear.

3.2.1 Public Datasets with Known Ground Truth

In this thesis, the performance of different target detection algorithms, exposed to different types of dimensionality reduction procedures, were studied. Originally the Indian Pines scene was used in the analysis. This scene was dropped late in the development of this thesis due to too strong similarities with the Salinas scene. These scenes were selected due to the fact that they are often used in publications. More relevant oceanographic scenes from instruments such as the HICO have unsatisfactory uncertainty wrt. ground truth [25]. However, a more oceanographic relevant scene, with signatures and distributions closer to what one can expect from imaging the ocean is found in the Hopavågen scene.

In table 3.1 an overview of the different endmembers and scenes are given. The table shows the total number of pixel labeled as a given endmember and what percentage of the total scene the endmember makes up.

Endmember	scene	pixels	percentage
Lettuce Romaine 4 weeks	Salinas	1068	0.96
Lettuce Romaine 5 weeks	Salinas	1927	1.73
Lettuce Romaine 7 weeks	Salinas	1070	0.69
Asphalt	Pavia	6631	3.19
Painted Metal Sheets	Pavia	3064	1.47
Trees	Pavia	1345	0.64
Carolline algae	Hopavågen	9130	20.4
Fucus Erratus	Hopavågen	10276	23.0
Green algae	Hopavågen	3021	6.78

Table 3.1: Overview of different endmembers in different scenes.

Salinas

The scene depicted in figure 3.1 is of Salinas Valley in California, USA. The cube has a width of 512 pixels and a height of 217 pixels and as aforementioned 224 bands. In several publications [24] the spectral bands associated with water absorption is removed, but in the simulations conducted in this thesis, they are kept. As can be seen from figure 3.1, the scene contains 16 classes, which mainly are different crops The scene has large homogeneous regions, which is an important feature of the scene. The scene is also known to have a very high spatial resolution. For comparison, the spatial resolution of the Salinas scene was 3.7 meters per pixel, whereas the Indian pines scene has a spatial resolution of 20 meters per pixel.

In figure 3.2 you can see an excerpt of some of the pixels in the scene, showing the trending of the scene. Figure 3.3, 3.4 and 3.5 are the endmember used in the analysis from this scene. In the figures all the pixels labeled as a given endmember is given in red, and the mean is given in black.

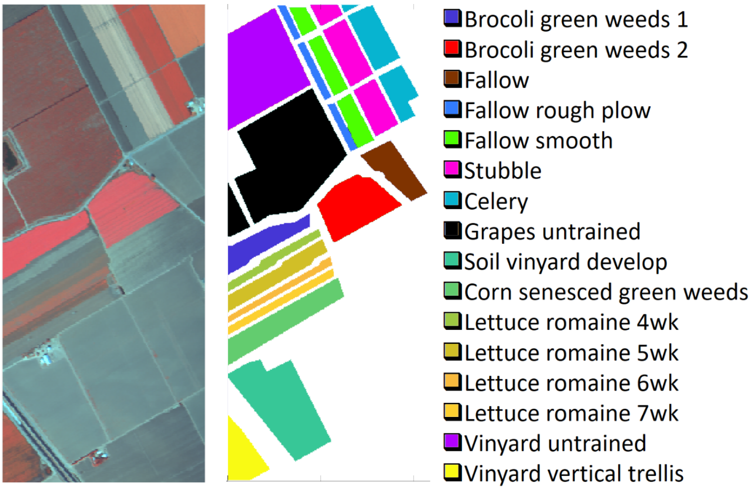


Figure 3.1: Salinas test site in California, USA.

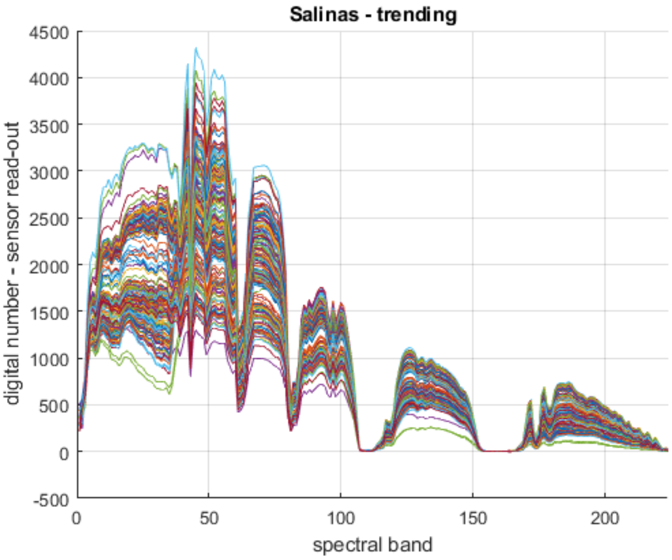


Figure 3.2: Trending of the Salinas scene. The image shows the sensor reading for the first 217 pixels diagonally

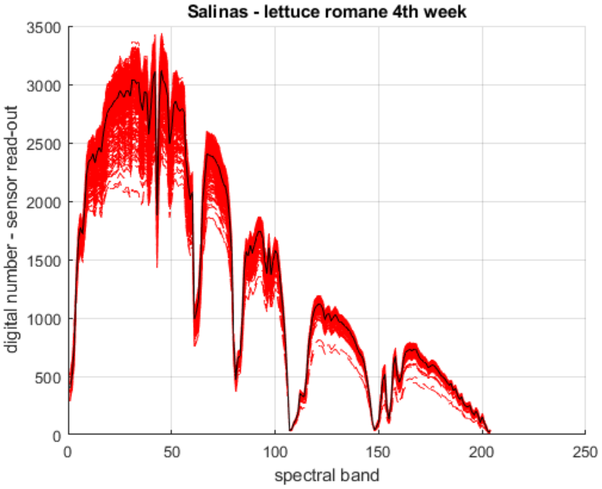


Figure 3.3: Lettuce romaine 4th week from the Salinas scene.

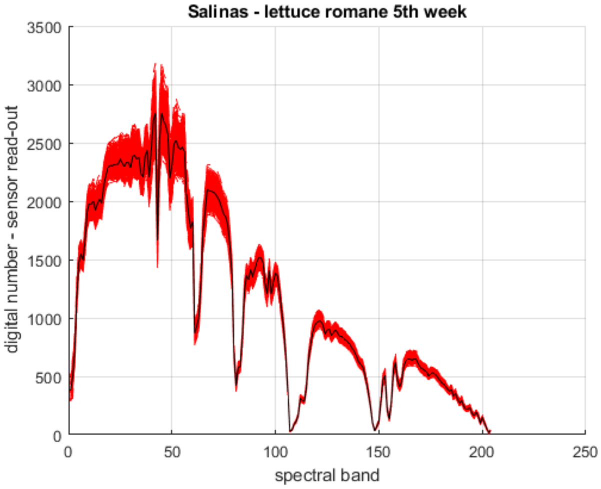


Figure 3.4: Lettuce romaine 5^{th} week from the Salinas scene.

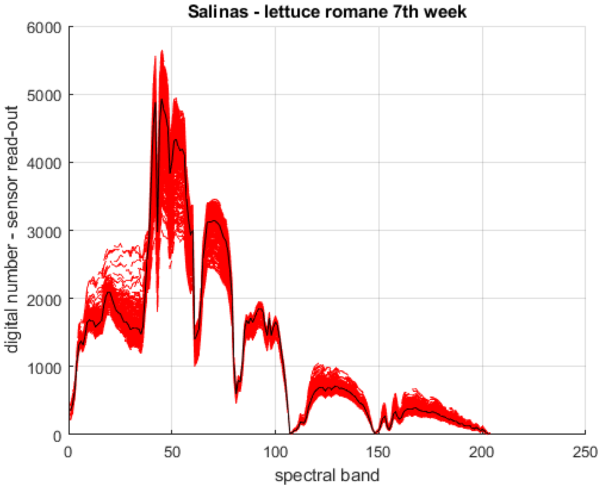


Figure 3.5: Lettuce romaine 7^{th} week from the Salinas scene.

Pavia University

The Pavia university scene is taken over the campus. This dataset differs from the other ground truth sets in several ways. It does not contain an abundance of agricultural or natural phenomena, but rather a lot of man-made structures as can be deduced from figure 3.6. With a spatial resolution of 1.3 meters per pixel, the scene has a high spatial resolution as well. The scene used in the simulations in this thesis is 610 by 340 pixels spatially and has as mentioned 103 bands. The bands containing noise are kept out as the publicly available data set does not contain them, making it impossible to include them [24].

In figure 3.7 you can see an excerpt of some of the pixels in the scene, showing the trending of the scene. Figure 3.8, 3.9 and 3.10 are the endmember used in the analysis from this scene. In the figures all the pixels labeled as a given endmember is given in red, and the mean is given in black.

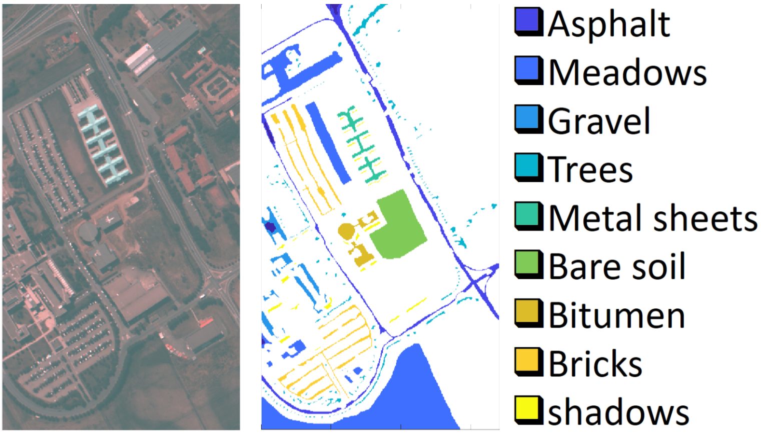


Figure 3.6: Pavia university site in northern Italy captured by the ROSIS-3 sensor.

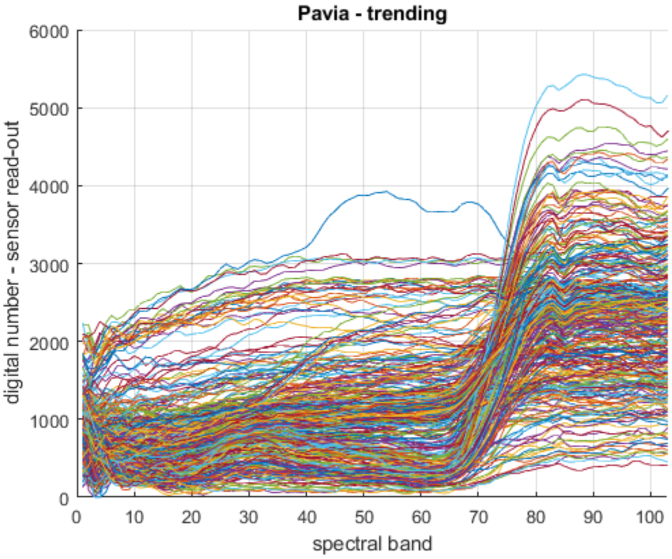


Figure 3.7: Trending of the Pavia university scene.

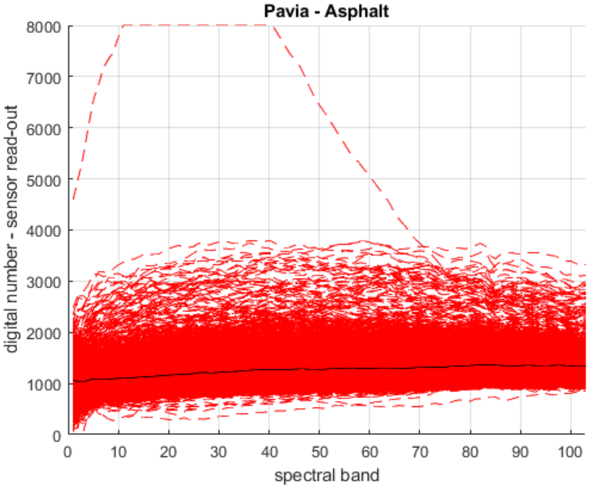


Figure 3.8: Asphalt from the Pavia scene.

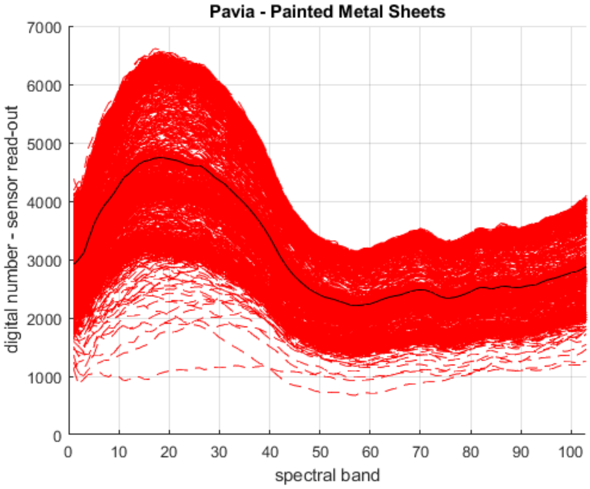


Figure 3.9: Painted metal sheets from the Pavia scene.

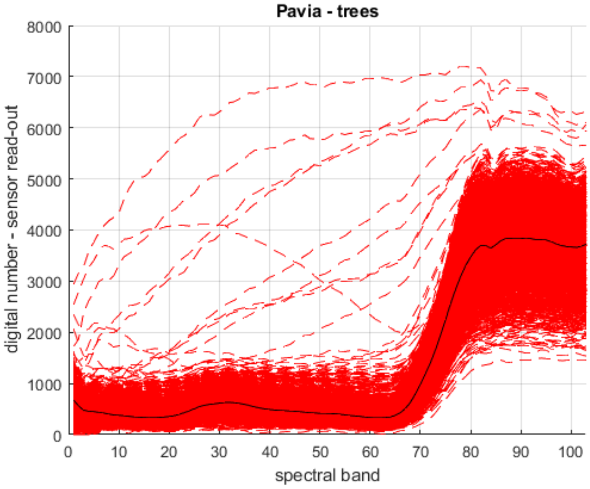


Figure 3.10: Trees from the Pavia scene.

3.2.2 Hopavågen

On the 22nd of March 2018, a joint test was performed by Trondheim Biological Station and the Department of Engineering Cybernetics (both affiliated with NTNU), using one of the prototypes for the Hyperspectral SmallSat mission on a UAV, and a commercial hyperspectral camera for underwater imaging produced by Ecotone (Ecotone Scientific UHI) [26]. There were several objectives and motivations for performing this test. With respect to this thesis, the most useful product of the test was the image depicted in figure 3.11, captured by the UHI instrument. The scene is 158 by 282 pixels spatially. This is a hyperspectral image with relevant oceanographic signatures with relatively low uncertainties. The image was classified and made available on the 9th of may 2018. The scene was classified by Ph.D. candidate Aksel Alstad Mogstad and is not publicly available.

In figure 3.12 you can see an excerpt of some of the pixels in the scene, showing the trending of the scene. Figure 3.13, 3.14 and 3.15 are the endmember used in the analysis from this scene. In the figures all the pixels labeled as a given endmember is given in red, and the mean is given in black.

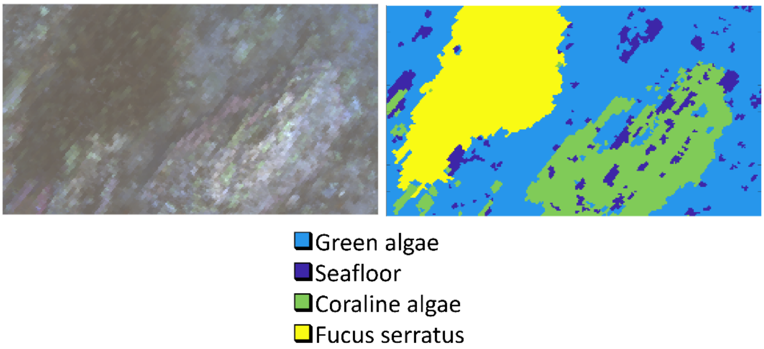


Figure 3.11: Hopavågen scene in Trondheim captured by the ECOTONE Scientific UHI, described in appendix B.

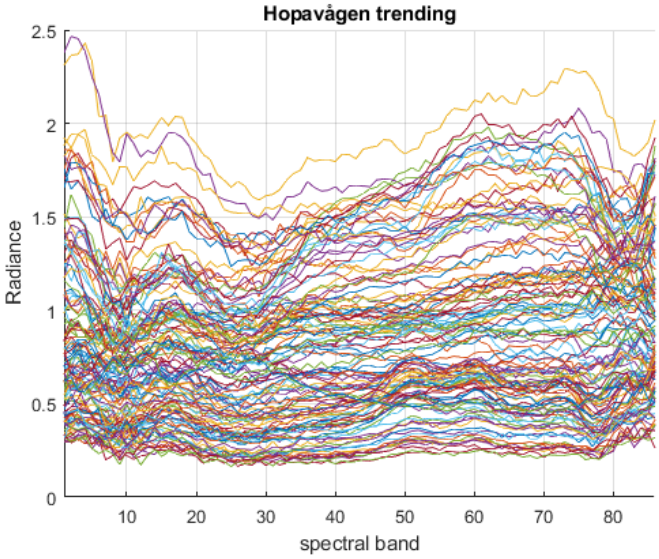


Figure 3.12: Trending of the Hopavågen scene.

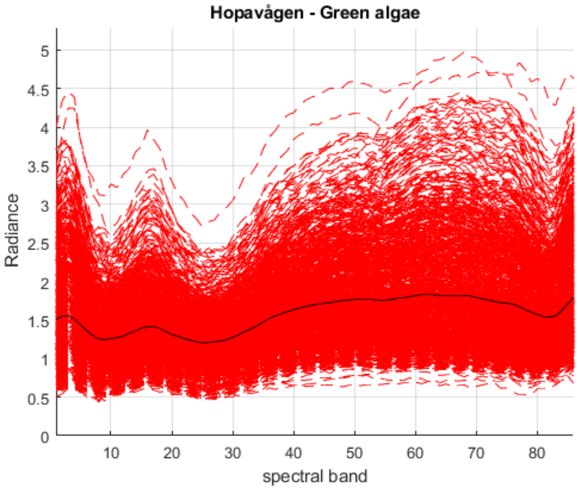


Figure 3.13: Green algae from the Hopavågen scene

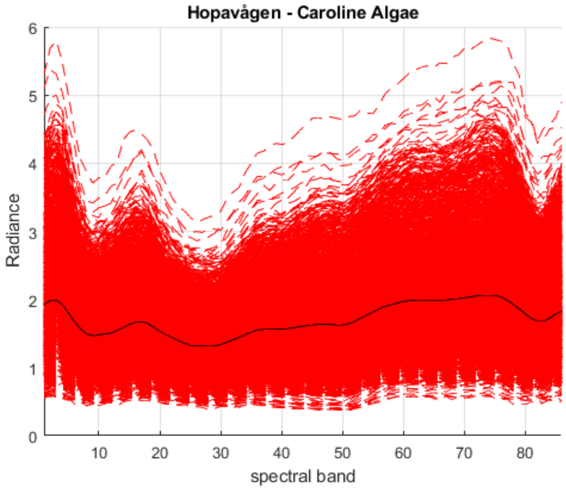


Figure 3.14: Coralline algae from the Hopavågen scene

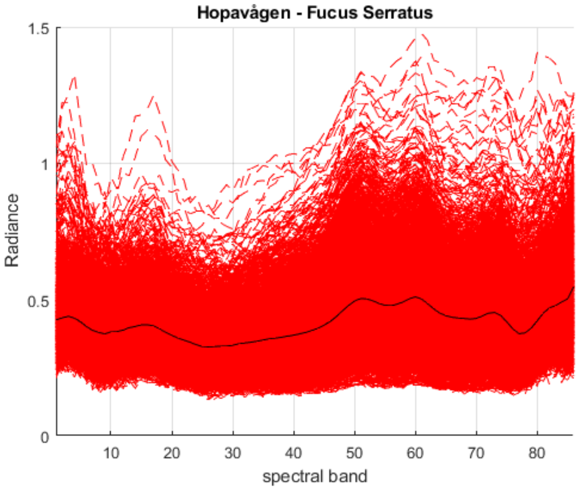


Figure 3.15: Fucus serratus from the Hopavågen scene

3.2.3 Synthetically Generated Datasets

A synthetic data set was generated using software provided by Grupo de Inteligencia Computacional [27] and utilizing the spectral library of USGS [28]. The synthetic data set was used to solely characterize the effects of dimensionality reduction, and not used to check the performance of target detection.

As the data from both HICO and all other hyperspectral sensors are contaminated with the noise it is difficult to know whether or not the dimensionality reduction is removing signal or removing noise. That is when comparing the reconstructed image with the original noise contaminated image. Thus the synthetic data set was generated, to have better control of the noise propagation in a given scene.

From the USGS spectra gathered, there were originally 2151 spectral measurements for each endmember ranging from 350 nm to 2.5 μm, having a spectral resolution of 1 nm. Only the wavelengths from 400 nm to 900 nm was used, with a spectral resolution of 5 nm. This was done in an effort to closer resemble the hyperspectral imager envisioned for the NTNU SmallSat Project [5]. A false-color image, i.e. an image showing only the bands associated with the colors red, green and blue, of one of the three scenes generated is given in figure 3.16. For the synthetic dataset wavelengths at 640, 550, 460 nm was used to represent red, green and blue, respectively.

As the motivation for creating these synthetic images was to better control the noise in a given image and to have an original noise-free image to compare towards. The synthetic image was introduced to four different types of noise. The types were Gaussian, Poisson, Salt & pepper and ”speck” noise. All four noise models can be found in the MATLAB implementation of the noise function *imnoise*. How they affect the image is listed below and can be seen in figure 3.16 and 3.17.

What kind of endmembers that were used for the image and how the selected endmembers are distributed can be seen in figure 3.18. The brighter yellow color indicates higher concentrations of the constituent in question. On the lower part of this figure you see the different endmember signatures and their name in the USGS library.

- Gaussian noise added a zero-mean noise with a variance of 1e-6
- Poisson noise was added in a way best described by the MATLAB implementation [29] for double precision numbers
- Salt & pepper noise was added in a way so that it would affect 1 $\%$ of the data in the image.
- Speck noise added multiplicative noise with the model $J = I + nI$, where n is uniformly distributed random noise with zero mean and variance of $5\text{e}{-4}$

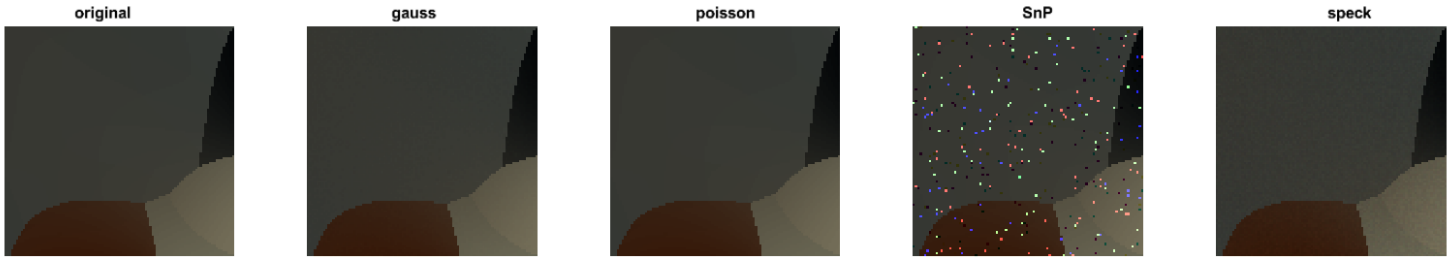


Figure 3.16: False color image of synthetic dataset.

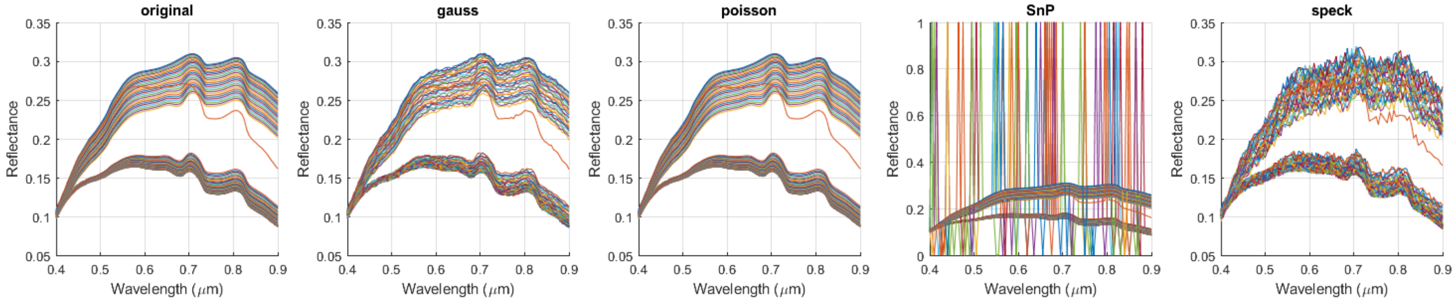


Figure 3.17: False color image of synthetic data set using wavelengths at 640, 550, 460 nm for red green and blue color representation respectively, with all the added noise representations as well. From the left there is the orginal image, Gaussian, Poisson, Salt & Pepper and Speck.

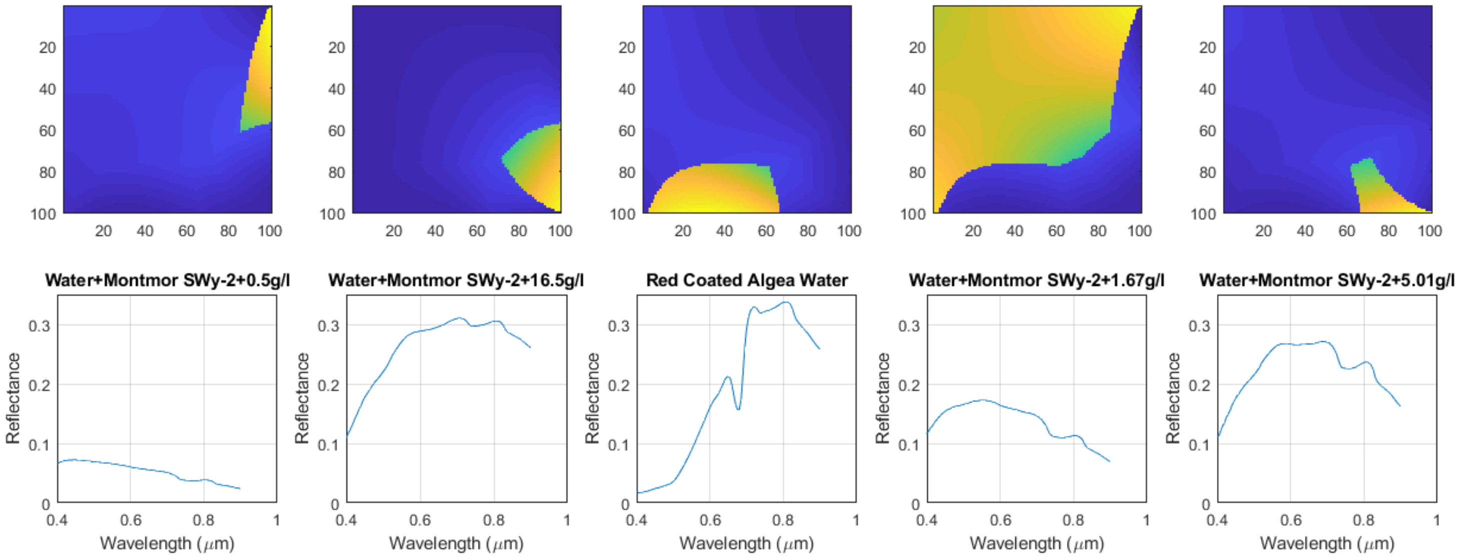


Figure 3.18: Abundance plot for the given synthetic image in figure 3.16.

3.2.4 HICO Datasets

The original intention of this thesis was to explore the effects and accuracy of different combinations of dimensionality reduction and target detection methods with respect to oceanographic phenomena for hyperspectral images. Thus, data acquired from the HICO mission was a natural choice [30]. Images from the HICO mission was used as they were the most relevant when considering the characteristics of the spectral data. The HICO was situated at the international space station and had several problems that were never solved during its lifetime. The mission was active for 5 years from September 2009 and captured several high-quality hyperspectral images of coastal regions all over the world. The HICO scenes are unable to provide a ground truth with satisfactory levels of certainty, and the images are contaminated by noise. The noise makes it difficult to see whether or not the dimensionality reduction removes signal or noise, or both. Still, the HICO scenes represent the closest publicly available set of data compared to what is to be expected from the NTNU SmallSat Program [5].

The scenes used in this thesis are depicted in figure 3.19. The dimensionality reduction described in this thesis was not used solely for target detection. The onboard processing team surveying and researching methods to use in the NTNU SmallSat Program also utilized the reduced space to see the effects of different compression algorithms, and how much there is to gain on performing dimensionality reduction wrt. compression ratio. The scenes were this selected based on the fact that they are often referenced in different papers, and again that they closely resemble what to be expected from the Hyperspectral NTNU SmallSat Satellite [5].

For a more thorough discussion of the HICO instrument and operations see [30, 4].

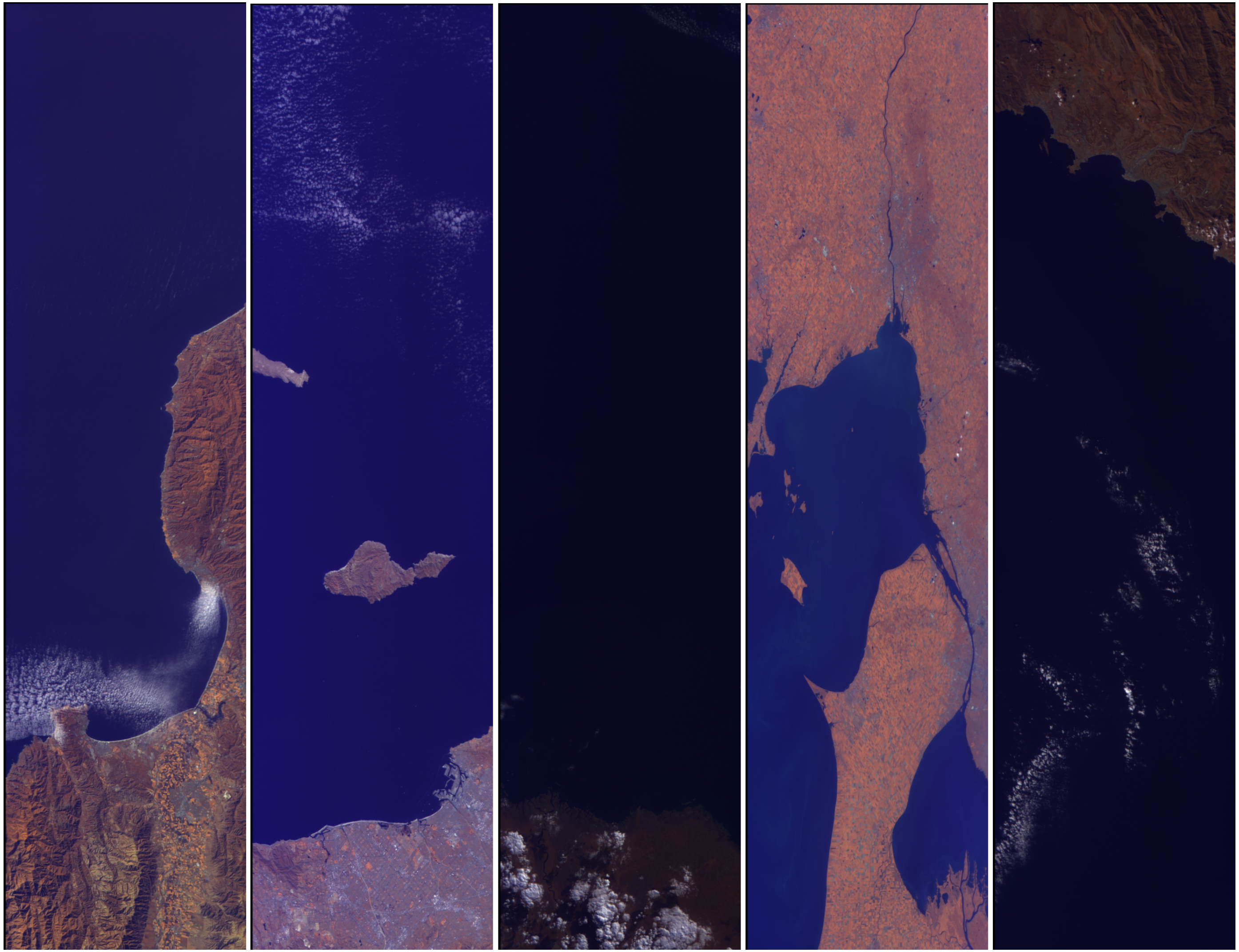


Figure 3.19: A collection of all the HICO images used in their L1B format (see [4]) The fourth scene from the left is known as the lake eerie scene, or in this thesis L1B_4. The others are simply known here as L1B accompanied by a number. Their results wrt. compression can be found in appendix C.1.

3.3 MATLAB Toolboxes and Scripts

In this section, the choice of programming language will be defended and the origin of the previously made code, both in form of toolboxes and stand-alone scripts will be made more clear.

As the objective of this thesis was to survey the effects of different target detection algorithms, and how they were affected by different dimensionality reduction methods, a fast implementation and large standard library were important factors in the decision of programming language. Furthermore, MATLAB is excellent for matrix operations and linear algebra e.g. eigenvalues, -decomposition and -vectors. All important mathematical concepts within the field of hyperspectral image analysis.

Other popular alternatives within hyperspectral image analysis are Python and IDL. Python has the advantage of having one of the largest communities of any programming language, several modules and packages both in the sense of general numerical analysis, but also modules geared specifically towards hyperspectral image analysis. To navigate between all these modules and their different versions can be a complicated task, as well as how these packages depend on each other. IDL is another popular choice for hyperspectral data analysis. Some of its advantages lie in the software package ENVI, and its execution time is faster, when compared to the aforementioned languages. Some of the difficulties regarding IDL is its simplicity i.e. it is difficult to create large programs There have also been reports of IDL exhibiting many irregularities requiring specific workarounds.

It should also be noted that the author of this thesis had a higher level of familiarity with the MATLAB programming language compared to other alternatives. Thus the natural choice was initially MATLAB, and after reviewing the alternatives that notion did not change. Excerpts of some of some selected parts of the code used in this thesis are given in the appendix A

3.3.1 Dimensionality Reduction Implementation

The implementations of PCA, MNF, and ICA have a different origin. The implementation for PCA given in the toolbox from section 3.3.2 gave unexpected results, but a pull request was submitted and accepted to help alleviate the problem. Still, for the simulations utilizing PCA, the built-in MATLAB version of the algorithm was used as is. The MNF implementation is made by the author of this thesis and closely following the descriptions given in A. Green [17] by the use of singular value decomposition as shown in section 2.2.4. The SVD is also part of the built-in MATLAB functions. There was an attempt at using the MNF as it was implemented in the toolbox mentioned in section 3.3.2, and also to directly use the implementations given in [19], but none of these gave expected results. Lastly, the implementation of JADE ICA was provided by professor Frank Westad, and how the code was used was deiced through discussion with him.

3.3.2 HyperSpectral Toolbox by Isaac Gerg

A toolbox publicly available through GITHUB, under GNU General Public License version 2.0 (GPLv2) have been used in this thesis [31]. The toolbox was originally made by Dr. Isaac Gerg and developed further by David Kun under the guidance of Dr. Isaac Gerg.

> The open source MATLAB Hyperspectral Toolbox is a MATLAB toolbox containing various hyperspectral exploitation algorithms. The toolbox is meant to be a concise repository of current state-of-the-art exploitation algorithms for learning and research purposes. The toolbox includes functions for Target detection, Endmember Extraction, Material abundance map (MAM) generation, Spectral Comparison, Anomaly Detectors, Spectral unmixing, Automated processing, Change detection, Visualization and Reading/writing files.
> *-Dr. Isaac Gerg, December 2016*

From this toolbox implementation for target detection, 3D-2D conversion of cubes and data sets, as well as the implementation of the N-FINDR algorithm has been used. The different implementations given in the toolbox have been cross-checked with the descriptions given in Hyperspectral Remote Sensing by M.T. Eismann [6]. Some of the implementations in the toolbox gave unexpected results or they were implemented in an unexpected way, and these implementations were of course not used.

3.3.3 Other Notes on Programming

The data sets with ground truth were either given directly as .mat files, or manually processed to be of the desired format, and read directly by MATLAB. The data sets from HICO were given as BSQ files. To read BSQ files, the built-in MATLAB function *multibandread* was used with the parameters given by the data provider. All images given in this thesis were generated using the same MATLAB function *create_rgb_from_bil*, given in appendix A. For different images different levels of saturation and contrast were chosen. The wavelengths selected to represent red, green and blue were approximately 640, 550 and 460 nm respectively, for all scenes.

3.4 Target Detection Performance Metrics

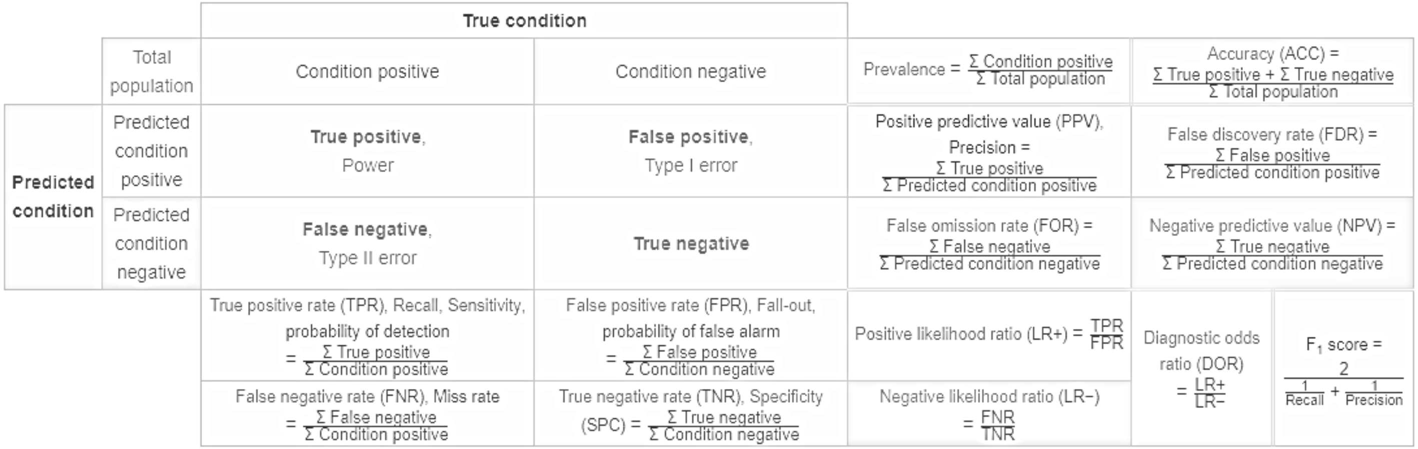
In this thesis, two approaches measuring the performance of a binary classification and one to measure the robustness of the target detection algorithms were chosen. A target detection algorithm can be viewed as a binary classification in the sense that you have the two classes target and background. The first presented here takes inspiration from machine learning i.e. it weights how well the detection algorithm would perform given the best possible threshold for a given image and target. In other words, the machine learning metric does not take into account how well background and target are separated, but rather how well a given algorithm could perform under ideal threshold values. To give an indication of how well the algorithm is able to separate the target and background a measure of visibility is chosen. This measure takes into account the response value or probability of target and is not dependent on a threshold.

In all three cases with ground truth, the given ground truth being looked for was set as a true positive, and all other pixels were viewed as true negative. Thus not accounting for pixels that were not mutually exclusive, that is consisting of more than a single endmember. This is by the nature of the datasets and not by choice. Hopefully, the people labeling the original images were thorough in their work and able to accurately determine which endmember dominated each and every pixel.

3.4.1 Machine Learning Metrics

The performance of the target detection algorithms wrt. machine learning metrics were measured in two ways. The first metric is the F_1-score [32], a metric that takes into account the precision and recall to measure the performance, The second was the Matthews correlation coefficient (MCC) [33, 32], metric which uses the entire confusion matrix. The threshold for each algorithm was chosen based on what gave the highest score for these two metrics by iterating over potential thresholds from 0 to 1.0 with a step size of 0.0001. The subsequent plots and analysis are based on the assumption that an ideal threshold was chosen. An ideal threshold is the one that gave the highest possible score wrt. the chosen machine learning metric.

A confusion matrix is a systematic representation of different ways to view binary classification and its potential outcomes, given in figure 3.20. From this the two machine learning metrics are derived, and terms such as precision and recall are defined.



		True condition				
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**, Power	**False positive**, Type I error	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Predicted condition positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Predicted condition positive}}$	
	Predicted condition negative	**False negative**, Type II error	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Predicted condition negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Predicted condition negative}}$	
		True positive rate (TPR), Recall, Sensitivity, probability of detection $= \frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out, probability of false alarm $= \frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) $= \frac{\text{LR+}}{\text{LR−}}$	F_1 score = $\frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}}$
		False negative rate (FNR), Miss rate $= \frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) $= \frac{\text{FNR}}{\text{TNR}}$		

Figure 3.20: A schematic drawing of a confusion matrix and some important derivations and definitions. The figure is directly adapted from the Wikipedia article about the confusion matrix.

The F_1-score [34, 32], is a performance metric designed for binary classification. With tp as true positives, fp as false positives and fn as false negatives the F_1-score is given as:

$$F_1 = \frac{precision \times recall}{precision + recall} = \frac{2tp}{2tp + fp + fn} \tag{3.1}$$

As mentioned the F_1-score is combination of precision and recall, defined in figure 3.20. Precision is the percentage of true positives divided by the total predicted positive conditions. By itself, precision is limited in its ability to accurately measure the performance of a binary classification. Given that a perfect precision score could be achieved if only one sample was labeled correctly and all other positive conditions were labeled as false negatives This is due to the fact that precision does not account for false negatives or true negatives outcomes. Recall, or true positives divided by the total positive conditions, falls short in the case where your algorithm labels all pixels as positive and achieves a perfect score, as the measure does not account for true negatives and false positives. As the F_1-score is a combination of these, one would get a measure that focuses on having a high number of correct guesses with as few false alarms or false positives as possible. The measure has a minimum at 0 and a maximum at 1, where 1 is a perfect score.

The Matthews correlation coefficient was chosen as metric due to the claim that it harbors certain advantages over alternative metrics for binary classification [33, 32]. As F_1-score does not account for the true negatives, whilst the MCC uses the entire confusion matrix to measure the performance of a given classification. Another significant difference for the MCC is that it ranges from -1 to 1 rather than 0 to 1. For the MCC a score close to 0 would indicate that the detection algorithm is random, and a score closer to 1 would be a better classifier. A score closer to -1 indicates that the classifier always gives the opposite class.

With tp as true positives, fp as false positives, tn as true negatives and fn as false negatives the MCC-score is given as:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \tag{3.2}$$

3.4.2 Visibility Metric

It is important for a target detection algorithm to be able to separate between background and target. A higher separability between target and background makes it easier to chose a detection threshold. This in return will suppress more undesirable false alarms. In this thesis the evaluation metric visibility, taken form [10], is used as a measure of this separability Visibility is given as follows:

$$vis = \frac{|\bar{T_t} - \bar{T_b}|}{max(T) - min(T)} \tag{3.3}$$

where the $\bar{T_t}$ is the average response value or probability of target for target pixels, and the $\bar{T_b}$ is the average response value or probability of target for non-target pixels. furthermore, $max(T)$ and $min(T)$, are the maximum and minimum probability of target in the scene for a given algorithm. The maximum score of visibility is 1, and the lowest score is 0.

Thus, a higher visibility score indicates a higher separability between target and background, a higher level of robustness.

3.5 Dimensionality Reduction and Target Detection

The thesis proposes a pipeline combining dimensionality reduction and target detection, and a simple schematic drawing is given in figure 3.21. This section describes how dimensionality reduction was performed to prepare for the execution of the different target detection algorithms. Only the OSP algorithm needs to estimate the background. For the other target detection algorithms the new space created by the dimensionality reduction is used directly in the computation. This is done with the idea of saving both time and power for the satellite on matrix operations.

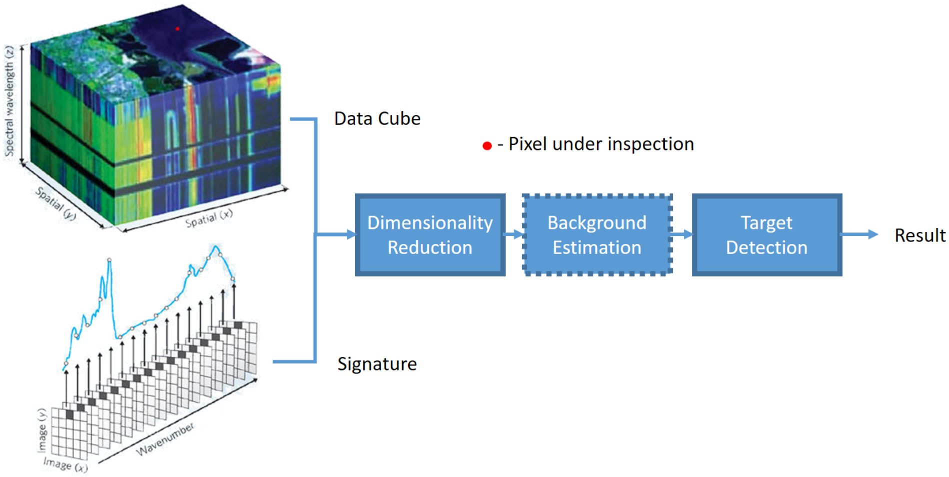


Figure 3.21: The pipeline combining dimensionality reduction and target detection.

3.5.1 Preprocessing and Preparing the Datasets

Three data sets Salinas, Pavia, and Hopavågen, were given in the format of a data cube and a ground truth with known labels. The endmember signatures, the abundance of samples, and other auxiliary data are generated from these cubes and ground truth maps. As the cubes were kept in their most noisy form, not excluding any bands, the images are closest to what an unsupervised system would experience, and most importantly the information amid the noise is not lost. The signatures were calculated as the mean of the signatures given a label. If a signature was given a ground truth label of x, all the pixels that were labeled x were averaged to produce the signature. In other words, in all the simulations performed in this thesis, the in-scene measurements were used, without any form of correction or adjustments prior to performing target detection. All the endmembers used in the simulations are given in section 3.2.1 and the black average line is then the signature being looked for in the simulations.

3.5.2 Cubes and signatures in new space

Listings 3.1, 3.2 and 3.3 show how the three different target detection algorithms were performed, and how the signatures were projected into the new space. In all of these listings M is the image cube and *end_sign* is the endmember signature in the original space. For the PCA listing 3.1, *coeff* is the transformation matrix. For the MNF listing 3.2, *Mg_mnf* is the transformed cube sorted bye noise, and *Ag_mnf* is the inverse of the transformationn matrix. For the ICA listing 3.2, *IC* is the independent components and *Tica* is the inverse transformation matrix.

Furthermore, the code depicted in listings A.10 and A.12, show the full context of how the results were generated and what was stored from each simulation, with M as the scene, *end_sign* as the signature and q as the number of components.

Listing 3.1: PCA transformation

```
            % do pca
            [h,w,d] = size(M);
            M_2d_pca = hyperConvert2d(M);
            M_2d_pca = M_2d_pca';
            [coeff,~,~,~,explained,~] = pca(M_2d_pca);

            V = coeff(:,1:q);
            M_pct = transpose(M_2d_pca*V);
            M_new_pca = hyperConvert3d(M_pct,h,w,q);

            end_sign_new = end_sign*V;
```

Listing 3.2: MNF transformation

```

            % do mnf
            [h,w,d] = size(M);

            [Mg_mnf, Ag_mnf] = GreenMNF(M);
            Ag_mnf_inv = inv(Ag_mnf);

            M_new_mnf = Mg_mnf(:,:,1:q);
            T = Ag_mnf_inv(:,1:q);

            end_sign_new = end_sign*transpose(pinv(T));
```

Listing 3.3: ICA transformation

```
% do ica
[h,w,d] = size(M);

M_2d_pca = hyperConvert2d(M);
M_2d_pca = M_2d_pca';
[coeff,~,~,~,explained,~] = pca(M_2d_pca);

P = coeff(:,1:q);
[IC, ~, ~] = jadeica(P',q);
Tica = M_2d_pca*IC'/(IC*IC');
M_new_ica = hyperConvert3d(Tica',h,w,q);
end_sign_new = end_sign*pinv(IC);
```

Listings 3.4 and 3.5 show computation of how the two different binary classification metrics, and computation of visibility, respectively. Utilizing the functional properties of MATLAB, the function named *metric(tp,tn,fp,fn)* in listing 3.4 was overloaded with the desired metric (MCC or the F_1-score). However, the chosen binary classification metric could easily be overloaded with a different metric deemed more relevant for a given application. Furthermore, as may be evident from looking at listings A.12, the target detection algorithms in question can easily be exchanged as well.

Listing 3.4: Computation of machine learning metric.

```
% Measure score
xx = size(gt,1); yy = size(gt,2);

tp = 0; tn = 0;
fp = 0; fn = 0;

P_FA = 0.0001:0.0001:1.0000;
met = zeros(length(P_FA),3);

for i = 1:length(P_FA)
    for j = 1:xx
        for k = 1:yy
            if and(gt(j,k) == id, b1(j,k) >=
                P_FA(i))
                tp = tp + 1;
            elseif and(gt(j,k) == id, b1(j,k) <
                P_FA(i))
                fn = fn + 1;
            elseif b1(j,k) >= P_FA(i)
                fp = fp + 1;
            else
                tn = tn + 1;
            end
        end
    end

    met(i,:) = [metric(tp,tn,fp,fn); tp; fp;];
    tp = 0; tn = 0;
    fp = 0; fn = 0;
end
```

Listing 3.5: Computation of visibility metric.

```
% Calculate Visibility
T_t_sum = 0; T_t_count = 0;
T_b_sum = 0; T_b_count = 0;

for j = 1:xx
    for k = 1:yy
        if gt(j,k) == id
            T_t_sum     = T_t_sum + b1(j,k);
            T_t_count   = T_t_count +1;
        else
            T_b_sum     = T_b_sum + b1(j,k);
            T_b_count   = T_b_count +1;

        end
    end
end
T_t_avg = T_t_sum / T_t_count;
T_b_avg = T_b_sum / T_b_count;

T_max = max(b1(:)); T_min = min(b1(:));

vis = norm(T_t_avg - T_b_avg)/(T_max - T_min);
```

3.6 Limitation of Methods

Below is a list of some potential concerns regarding how this thesis was conducted, and some further shortcomings of the methods used and how they were implemented.

- The scenes given does not necessarily consist of mutually exclusive pixels, and a compensation for sub pixel signatures or mixed pixels is not incorporated in the analysis.

- The target detection analysis covers 7 different representations for 3 different scenes with 3 specific endmembers using 4 different target detection algorithms, resulting in 252 different cases to be analyzed. As this is an exhaustive number of cases the analysis was performed on the average of results for specific scenes, with some exceptions. By performing the analysis in this way the results discussed gives more of a qualitative conclusion, even though many of the specific cases point in the same direction.

- The MNF algorithm, as it is implemented for this thesis, uses the entire image to estimate the noise model and not specific hand-selected homogeneous areas. This is not a good way to estimate the noise model when the scene is non-homogeneous and was intended as a demonstration of the effect of sub-par noise model estimation. As a consequence, it also undermines the performance given by the MNF algorithm in the less homogeneous scenes. Unfortunately, a good noise model for the UHI used in the Hopavågen was unavailable at the time of writing, and the MNF based DR suffers from this in this particular scene non-intentionally.

- The implementations given in the toolbox were, to the extent permitted, cross-referenced with papers and textbooks covering hyperspectral algorithms[6, 11] , but that is not the same as to say that the implementations are without error.

- The JADE ICA methods is one of many methods for performing ICA, it was chosen as it is known to converge well, but this is not a cost-effective way to perform ICA, and the implementation in this thesis does not use and automatic cross-validation or similar approaches [21] to extract the most relevant components. This may not give the optimal result for the JADE ICA DR.

- In this thesis, considerations regarding atmospheric effects have been neglected. For the Hopavågen scene it is illogical to perform atmospheric correction, but it is relevant for both the Salinas and Pavia scene. The idea behind neglecting atmospheric effects is that target detection in hyperspectral imaging is not directly dependent on the signatures obtained after atmospheric correction i.e. the spectral variance observed should be of similar variance before and after the atmospheric correction, and the signature searched for should follow a similar transformation. This assumption is potentially naive, as the outcome of atmospheric correction is very dependent on how the correction is performed [35].

- A lot of figures and graphs generated for the analysis was kept out of this thesis, as the number of pages versus information per page was unjustifiable. All the information used to make the figures and graphs can be found in appendix D, in a table that is not easily interpretable. The figures and graphs have been made available at `http://folk.ntnu.no/sivertba/master_thesis/`, along with some other files deemed relevant.

- When measuring the target detection performance wrt. visibility of a proposed pipeline it was assumed that the average probability of target for target pixels would be higher than the average probability of target for non-target pixels. That is $\bar{T}_t > \bar{T}_b$, given equation (3.3), for all cases.

- To what extent the implementation of the different DR methods in and of itself affects these results is hard to say. The MNF method uses singular value decomposition rather than the inverse matrix computation. How the inverse of a matrix has been performed in MATLAB has not been considered widely in this thesis, but that is not the same as to say that the beneficial results obtained by the MNF transform are not caused by the implementation. That is, what kind of numerical advantages taking the inverse based on singular value decomposition rather than the inverse using the MATLAB implementation gives, has not be accounted for. In principle, the difference should be rather small.

Chapter 4

Analysis

In this chapter the results and findings will be discussed. First the compression through dimensionality reduction will be analyzed for the HICO data and the synthetic data, without considering target detection. Secondly, the effect of combining dimensionality reduction and target detection will be analyzed for the scenes given in section 3.2.1.

4.1 Compression results

The main objective of the first satellite in the NTNU SmallSat program is to capture oceanographic scenes with a hyperspectral imager [5]. One of the major bottlenecks for a small satellite transferring data to earth is the data rate. Thus a smart compression scheme, one that keeps the most valuable information for a given application, is necessary to be able to produce the desired amount of data.

Several parallel efforts were made, using methods such as JPEG2000 and CCSDS-123, to survey how to best compress the data. The different DR representations were used as a prepossessing step before performing the lossless compression. Through combining dimensionality reduction and loss-less compression high compression ratios was achieved. This is not discussed further in this thesis,

The algorithms used in this section and how they operate is covered in chapter 2, and how they were implemented can be seen in appendix A.

4.1.1 Compression on HICO Data

For the compression on HICO:

- The effects of PCA is visualized in figure 4.1 and 4.2, and figure 2.5. The figures 4.1 and 4.2 show a false color restoration of the image and arbitrarily chosen spectral responses for four pixels using the reduced space and a transformation matrix.
- The effects of MNF is visualized in figure 4.3 and 4.4, and figure 2.6. The figures 4.3 and 4.4 show a false color restoration of the image and arbitrarily chosen spectral responses for four pixels using the reduced space and a transformation matrix.
- The effects of ICA is visualized in figure 4.5 and 4.6, and figure 2.7. The figures 4.5 and 4.6 show a false color restoration of the image and arbitrarily chosen spectral responses for four pixels using the reduced space and a transformation matrix.

Furthermore, the results of the analyzed scenes are given in appendix C.1.

The metrics measured when analyzing the HICO scenes were Peak Signal to Noise Ratio (PSNR) and Compression Ratio (CR). The PSNR was computed as the peak value of the original cube divided by the mean square error difference between the original cube and the restored cube. The Compression ratio was computed as the number of spectral bands originally in the image divided by the number of components. Not surprisingly, all the DR methods with the same number of components give identical compression ratio in the preprocessing step that is DR. However, for all the HICO scenes analyzed the PCA transform gives the highest score wrt. PSNR. The ICA transform follows the PCA transform closely but does not achieve the same performance wrt. PSNR. Lastly, the MNF transform is decidedly worse than the other methods for all cases.

For the HICO scenes, the PCA transform seems to be superior when compared to both MNF and ICA. From figure 2.5, it is apparent that the PCA transform will at an early stage start to model structures that originates from sensor noise, but this is beneficial in the sense that the raw HICO image is compared to the restored one directly. Thus, if one were to recreate the raw image, the PCA transform will not discard information based on a noise model such as the MNF transform, and the resulting restored image will then be more similar to the raw image to a larger extent than with MNF and ICA.

From figure 4.1 and 4.2, the ability of the PCA transform to recreate the original signal with relatively few component is compared with relatively many components. From these two figures, it is clear that even with few components a lot of the data from the raw image is still kept. With 50 components the image is more or less completely restored, with only a few pixels where the PCA model falls short. An optimal number of components wrt. compression and data integrity would be further application and implementation specific. That is, how many components to use form the PCA model should be decided based on whether or not masking of undesired pixels is performed, and if the PCA model should be static or not.

The results from performing PCA on the HICO data suggests that, if one does not wish to noise filter the data as part of the compression, the PCA transform have some advantages when compared to MNF and ICA.

The low performance displayed by the MNF transform may be due to how the noise model was estimated. The HICO images have some pixels that are saturated and *dead* i.e. the image sensor fails to sense the light levels correctly for some pixels. In none of the analyzed HICO scenes were dead or saturated pixels removed, thus including them both in the covariance matrix computation and in the noise model estimation. From this, it is safe to say that the computation of the covariance wrt. to all DR methods, and the noise model estimation in MNF, are both sub-optimal. A masking or automatic detection of undesired pixels, e.g. dead and saturated pixels or non-oceanographic pixels, would be beneficial as it would help the DR methods to generate better models. With this filtration of pixels, a more consistent data set to compute the different models would emerge. From figure 2.6 it can be observed that the lower bands contain a lot less information regarding the spatial structures found in the original image, when compared to the top bands in PCA and ICA, given in figure 2.5 and 2.7.

From figure 4.3 and 4.4 we can see that the MNF transform struggles with recreating the image with few bands, and with more components, the MNF transform is still unable to recreate the lower and upper bands of the HICO sensor. This is expected, as these bands are known to be the noisiest bands in the HICO instrument [30], suggesting that the MNF transform is filtering the noise in these bands.

Thus, by trying to filter away the noise, the MNF transform is filtering away some of the signal contained in the raw HICO image, which the restoration is compared towards. Such that, by removing noise from the image, the MNF transform is effectively removing signal that the PSNR metric, given how it is performed for the HICO images, sees as a signal. This may be some of the reason for the abysmal performance observed for the MNF transform on HICO data.

By looking at figure 2.7 it is clear that the ICA transform operates differently than the PCA or MNF transform, given how they sort components. This is as to be expected as the ICA transform is not implemented with methods to validate or sort the given independent components that it finds. As the PCA and MNF find the components that give an orthogonal data set, the ICA transform finds independent components that in the best possible way span the data set. This is visualized in the first few components given in figure 2.7, by the fact that they are very different than the first few components found in PCA and MNF, given in figure 2.5 and 2.6. The ICA transform is able to find certain patterns wrt. how noise propagation in the cross-track direction (given the third component in figure 2.7), and the natural choice would be to remove this component completely. This would, however, result in a lesser score wrt. PSNR for the HICO scenes given how it is measured.

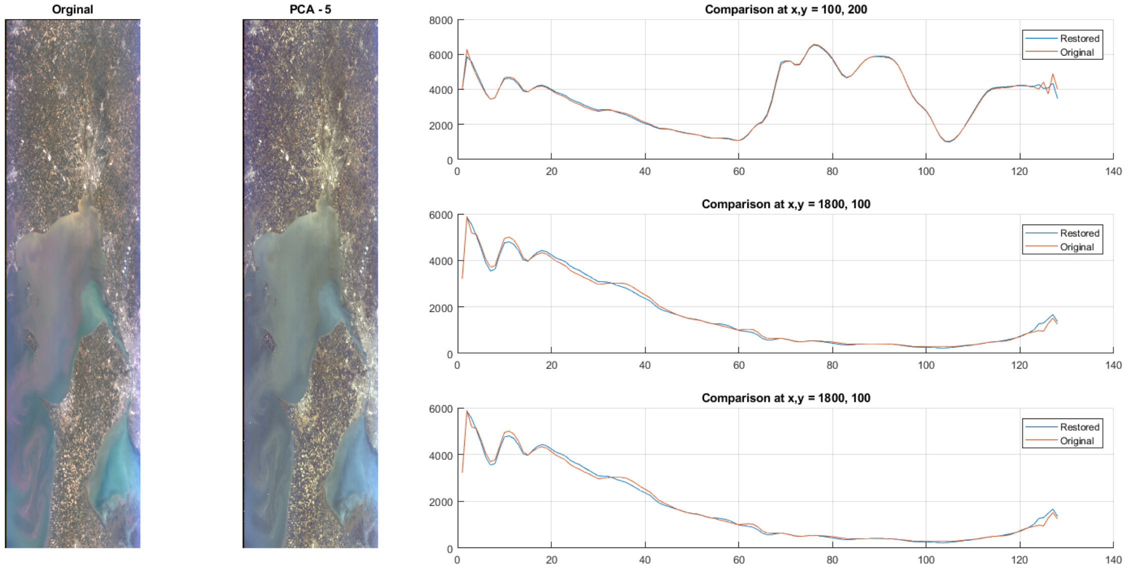


Figure 4.1: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using the 5 most significant components in PCA.

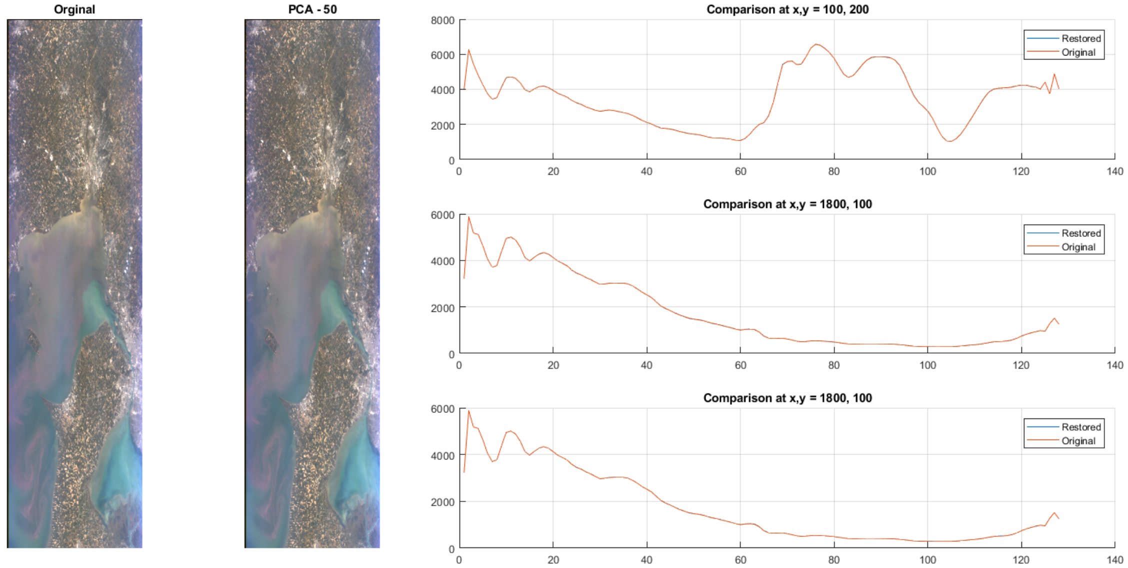


Figure 4.2: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using the 50 most significant components in PCA.

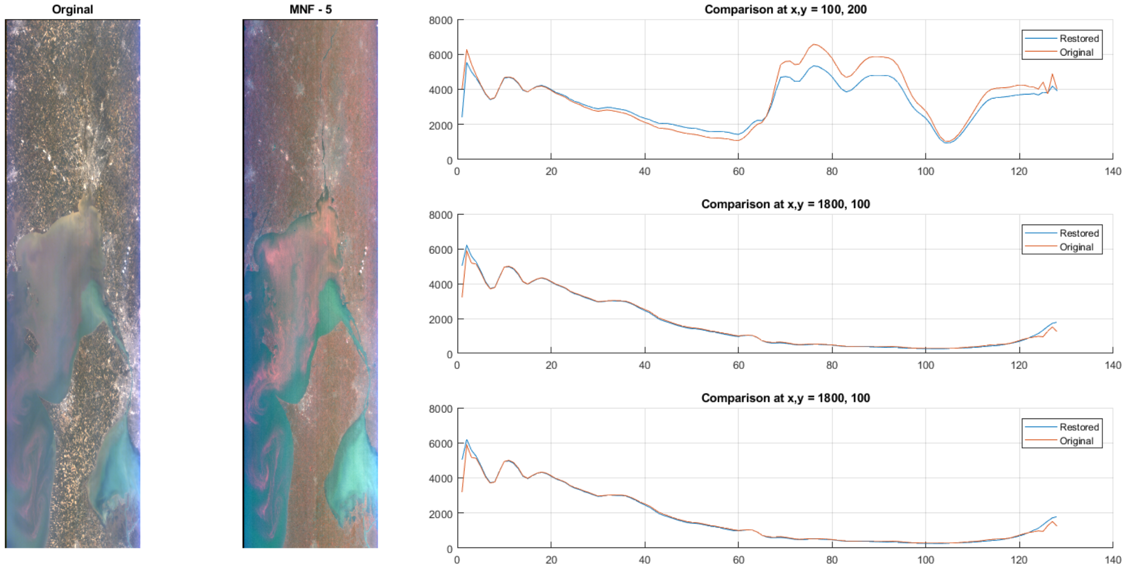


Figure 4.3: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using the 5 most significant components in MNF.

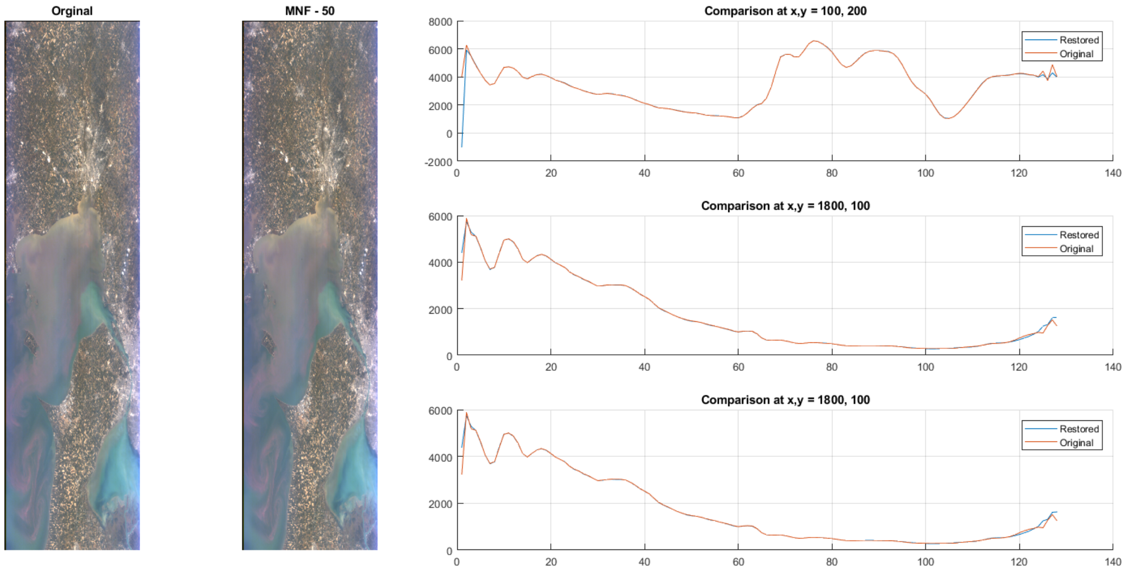


Figure 4.4: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using the 50 most significant components in MNF.

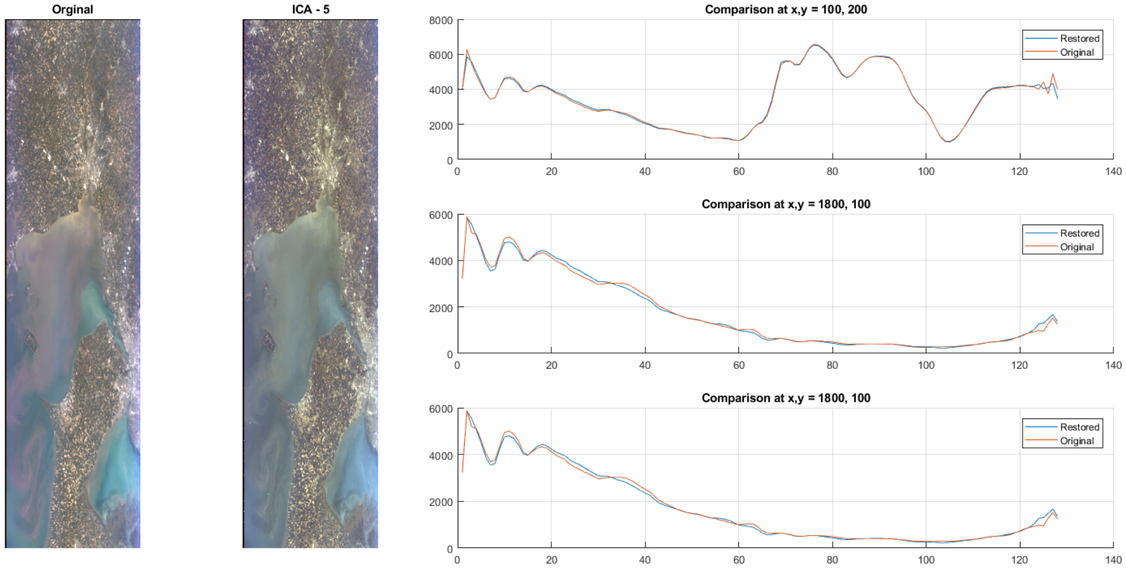


Figure 4.5: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using 5 components in ICA.

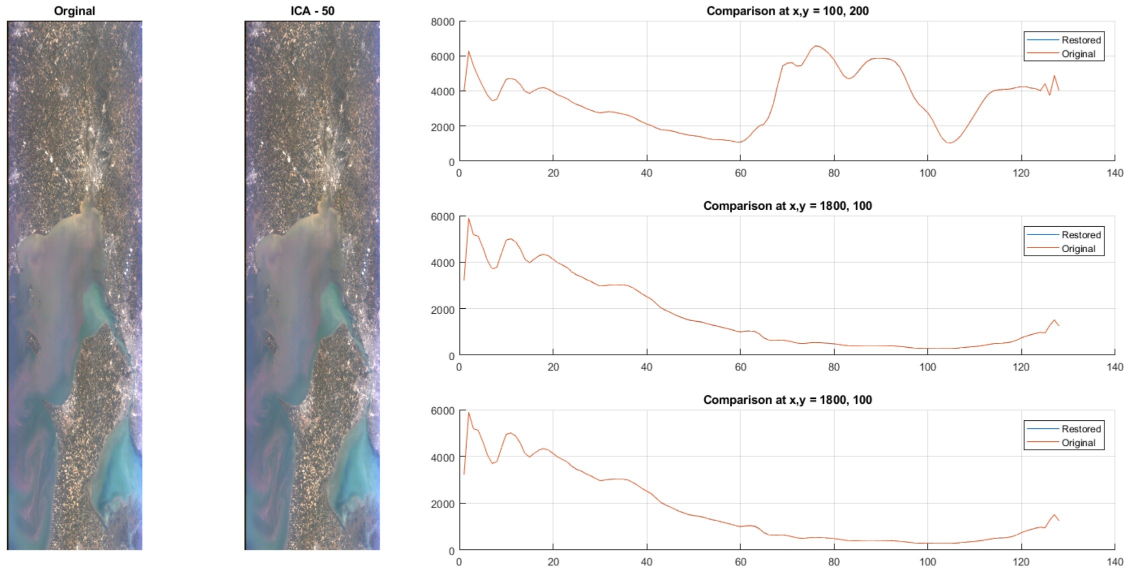


Figure 4.6: An image of Lake Eerie in the U.S. An image that has been widely used due to the high activity as can even be seen in the true color image above. In this image the reconstruction was performed by using 50 components in ICA.

4.1.2 Compression on Synthetic Images

The synthetic data sets were created to better control how the noise propagated in the image. How the images were created and what kind of noise they were exposed to is described in section 3.2.3. The results from the different DR representation is given in appendix C.

The first thing to observe from the results given in appendix C regarding the synthetic data sets, is that the images are homogeneous in how they are spatially distributed. Thus, assumptions made in the implementation of the MNF transform are well met, even with an in-scene based estimation of the noise model. This is the case across all the different noise models and component representations for all three files used in the simulations.

In the synthetic scene PSNR and SNR were measures computed as the peak and mean value of the original cube divided by the mean square error difference between the original cube and the restored cube, respectively. The increased performance wrt. to PSNR and SNR demonstrated for the original image, given the MNF representation, is noticeably higher than for PCA and ICA. The images were synthesized using the software discussed in section 3.2.3, which simply multiplies the abundance with the endmember signature for a given pixel. This result suggests once again that the given spatial distribution of data is particularly well suited for the MNF algorithm.

The differences between the three methods are less acute when Gaussian noise is added. For the PCA transform, adding more components seem to enable a better recreation of the original image i.e. in the range researched the model has yet to start modeling noise to a large extent, and the same can be observed for the ICA transform.

The MNF transform on the other hand does not draw a large advantage of increasing the number of components, and for some of the files, the SNR becomes lower with an increased number of components. This suggests that the transformation is modeling noise at an earlier component than that of the PCA and ICA.

Furthermore, the SNR of the restored image after being exposed to Gaussian noise is far lower for the MNF representation, and at the same time the PCA and ICA representation barley experience an effect of the distortion wrt. restoration of the original data cube. Suggesting that the PCA an ICA transform is best suited for images where the Gaussian noise distribution dominates.

This corresponds well with the results found in [7], where they concluded that the PCA transform was beneficial compared to the MNF transform when the added noise was of a Gaussian character in AVIRIS images.

When the noise added is Poisson noise, the MNF transform performs noticeably better than both the PCA and ICA transform. Across all representations of the data, an increase in the number of components increases the ability to reproduce the original image. The change in SNR is lower with more components for the PCA and ICA than for the MNF. This suggests that the PCA and ICA methods are not able to filter away this specific type of noise, whilst the MNF method tries to make a model of the noise based on the sampled data with better success.

As the scene is well suited for the type of noise model estimation performed by the MNF implementation, the noise model created produces better results than when simply assuming Gaussian distribution.

From this, it seems like the MNF method, as a preprocessing step, is able to better suppress non-Gaussian noise than PCA and ICA, a welcome result as this also corresponds well with the conclusion and result found in [7].

The salt pepper noise is easy to detect as an error in sensor reading, and not something that one would want to keep for analysis. It still may occur in a physical system, thus how this will affect the restoration of the original image, i.e. the image that should have been captured, still gives useful insight.

As opposed to the results regarding Gaussian and Poisson noise, the added noise, in this case, does not obtain a better SNR with an increase in the number of components. In addition to not improving the SNR with more components, the SNR is initially rather low, when compared to the effects from other models. Thus indicating that none of the aforementioned methods are able to filter away the noise. The model for the scene created by the methods are dominated by the effects of the noise, and the modeling is then unable to recreate an image close to the original image.

The speck noise model produced by the *imnoise* function in MATLAB closely resembles the noise one can find in real applications. For the PCA and ICA implementation, the addition of more components does not seem to affect the restoration of the original image in any way. Thus, suggesting that the model created by PCA and ICA model the signal and noise from one of the most significant components.

The MNF transform on the other hand start off with a much higher SNR than both PCA and ICA but also start modeling noise when adding more components. This indicates that the MNF transform once again is better at estimating the noise model, but at some point, the components will start to model noise rather than signal.

From this it is clear that the different transformations will be able to suppress certain types of noise, some better than others. As these methods are often used as a preprocessing step on raw data, it is expected of them to be able to extract what is relevant wrt. data exploration. The MNF transform seems to be superior in most cases, and this may be due to the assumptions made in the implementation fits the scenes well. Thus indicating that with an optimal noise model the MNF transform is able to filter away undesirable noise, and at the same time keep the desired signal. How the different methods have been implemented and if this has skewed the results in a particular direction should be investigated further.

4.2 Target Detection Results

In this section, the results generated by the proposed pipeline in section 3.5, depicted in figure 3.21, will be discussed. The different dimensionality reduction methods will be compared towards each other, in combination with different target detection algorithms. The analysis will only be performed on the scenes with ground truth described in section 3.2.1. The simulations are conducted on the endmembers given in section 3.2.1. A more detailed table of the individual results for each scene, endmember, algorithm and dimensionality reduction is given in appendix D.

As can be seen from the column $|F_1 - MCC|$ in appendix D, the F_1-score and the MCC-score rarely differ substantially, or they follow one-another as the DR representations progress. This is a welcome result as it suggests that the algorithms have consistent precision and recall, whilst still maintaining a stable rate of true negatives. The analysis is discussed here only for the MCC-score. This was done due to the small difference between the F_1-score and the MCC-score, and how the results closely followed each other across all DR representations and target detection algorithms.

4.2.1 Target Detection on Salinas

The Salinas scene is well characterized and has a lot of homogeneous areas, and this is among the more important features that make it different from the two other scenes presented in this section. This scene contains 16 different signatures and large unclassified regions (see figure 3.1). Because of this the number of components to represent all the relevant information in the image should at the very least be higher than 17. The endmembers looked at are of very similar spectral characteristics, being three similar crops seeded in different weeks. This is intended to represent the case of a homogeneous scene, where the target signature shares some of the characteristics of the background or surrounding regions.

In figure 4.7 it's clear that the three detectors ACE, CEM, and SAM are able to achieve a relatively high MCC-score, across all DR representations. The DR representations do not alter the performance wrt. MCC-score substantially, but the MNF representations seems to have the least negative impact compared to the full space representation. On the other hand CEM, OSP and SAM all have similar, and relatively low levels of visibility, whilst the ACE detector is able to achieve a much higher visibility for all DR representations, strongly suggesting that it is a more robust alternative given the characteristics of the scene.

The DR representations do not decrease the performance substantially for all detectors. In the Salinas scene, the original image has a spectral space that consists of 224 bands, thus a reduction to 50 and 20 components represents a compression ratio of about 4.48 and 11.2 respectively. This is a substantial rate of compression. Furthermore, in some of the cases shown in appendix D, and as indicated by figures 4.7 and 4.8, the dimensionality reduction may introduce an improvement wrt. both binary classification performance and robustness.

To look at a more specific case, the ACE detector with the target being Lettuce Romaine 4th week and the MNF representation with 20 components were selected. The results from this particular scenario is given in table 4.1, taken from appendix D. In figure 4.9 a spatial representation of the visibility can be seen. The MNF representation gives a higher probability of target to a lot more non-target pixels but at the same time a higher probability of target for target pixels. This pipeline gives an MCC-score and visibility of 0,888 and 0.826 respectively. When performing the target detection on the full space an MCC-score and visibility of 0.848 and 0.568 are obtained. The improvement in MCC-score may be due to random variations as it is a factor of less than 5%. However, wrt. the visibility score an increase of 45% was achieved. This argues that the MNF representation is able to better separate the background and target for this particular scene, for this particular endmember. However, similar improvements can be found for all detection algorithms for the MNF representation in this scene for all the investigated endmembers as well (see appendix D).

Table 4.1: Results from Salinas scene looking for Lettuce Romaine 4th week with ACE

DR	F1	MCC	$\lvert F_1\text{-MCC}\rvert$	Recall	Visbility
FULLDIM	0,849	0,848	0,001	0,840	0,568
PCA50	0,879	0,878	0,001	0,874	0,638
MNF50	0,887	0,885	0,001	0,889	0,636
ICA50	0,879	0,878	0,001	0,874	0,638
PCA20	0,818	0,818	0,000	0,772	0,698
MNF20	0,889	0,888	0,001	0,884	0,826
ICA20	0,818	0,818	0,000	0,772	0,698

As the Salinas scene is particularly homogeneous, the assumptions made in the implementation of MNF is well met. Figure 4.7 and 4.8 show that on average both the performance and visibility is highest for the MNF representation, regardless of the number of components. So for the Salinas scene, on average, the best performance and visibility was achieved by using the ACE detector in combination with an MNF-based dimensionality reduction with a low number of components. Suggesting that a pipeline using the MNF representation in combination with the ACE detector is well suited for homogeneous scenes where the spectral signatures are of similar characteristics.

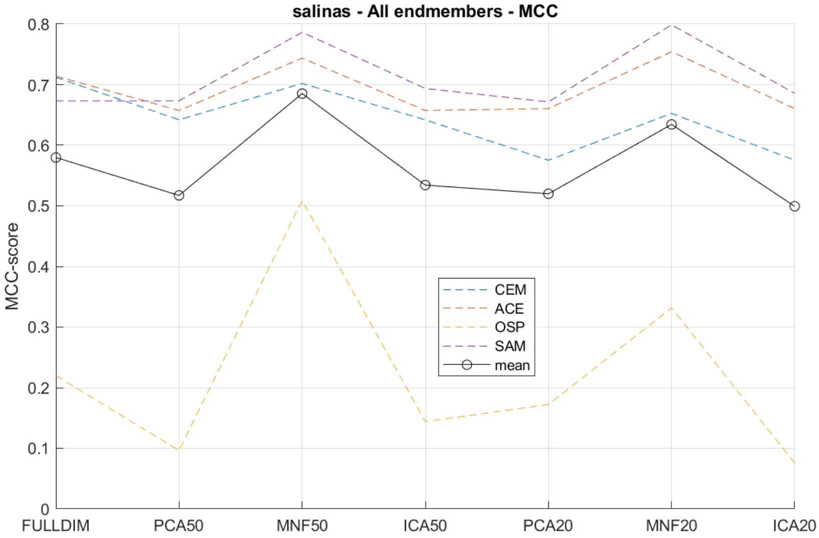


Figure 4.7: Average MCC-score for all target detection algorithms in the Salinas scene

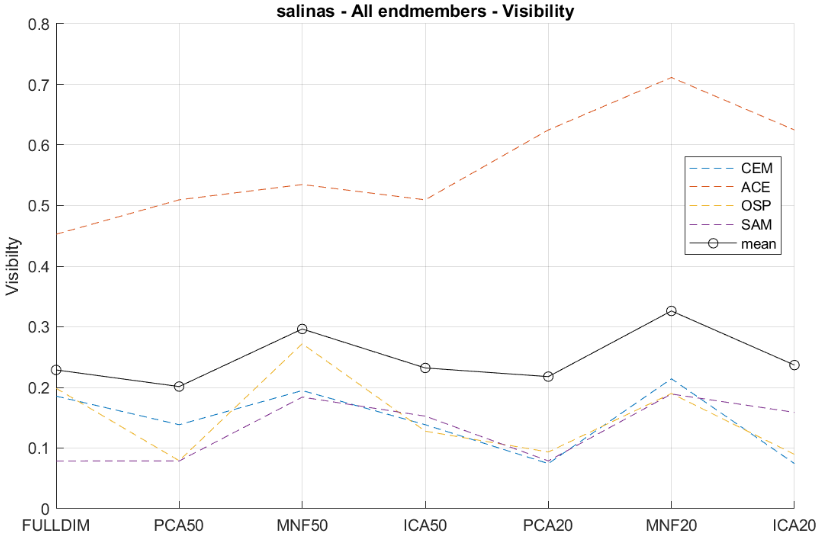


Figure 4.8: Average visibility score for all target detection algorithms in the Salinas scene

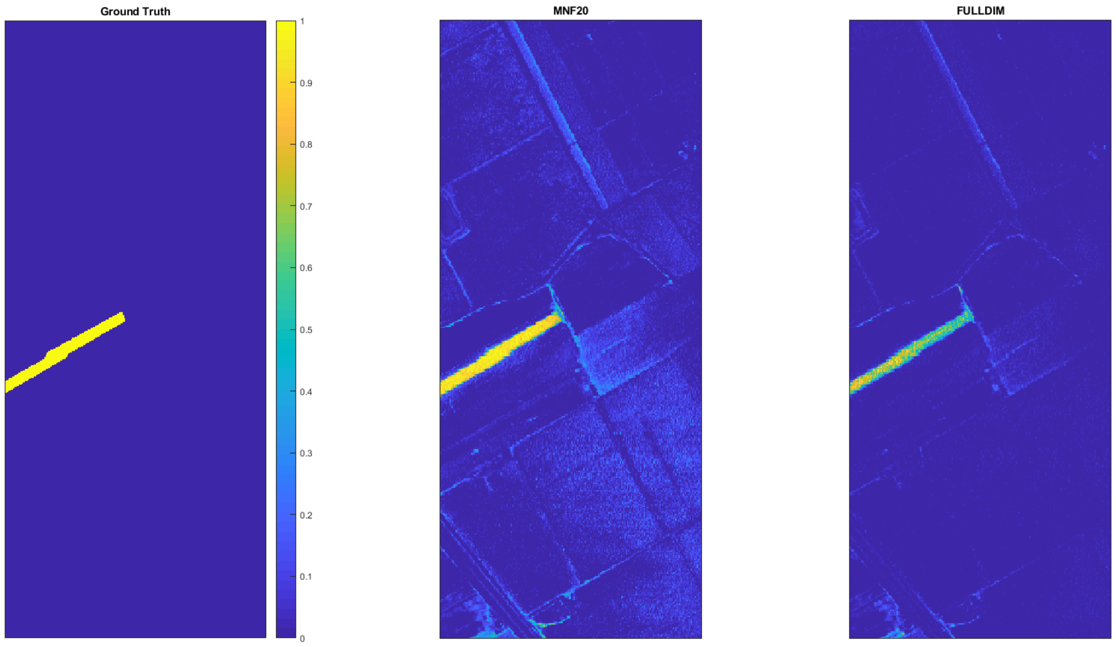


Figure 4.9: Visualization of the visibility produced by the ACE algorithm in Salinas scene for Lettuce Romaine 4th week.

4.2.2 Target Detection on Pavia

The Pavia scene is well characterized with a high spatial resolution, and a lot of man-made rigid structures that creates a lot of *edges* (see figure 3.6). This scene contains 9 different signatures and large unclassified regions. Because of this the number of components to represent all the information in the image should at the very least be higher than 10, but as all the scenes have been treated the same it is exposed to the same types of transformations as the Salinas and Hopavågen scene. Due to the aforementioned edges, in an image processing sense, the scene could be described as non-homogeneous. Thus, for the MNF representation, this is intended as a challenge, revealing the shortcomings of a subpar noise model, wrt. how the noise model estimation is implemented. As the other DR representations do not use the image to estimate any parameters beyond the components, the change of scene should not affect the performance of the algorithm in that sense. Furthermore, the full spectral range of this scene contains 103 bands, and a DR representation with 50 and 20 components would give a compression ratio of 2.06 and 5.15. This is not an equally large compression ratio as compared to the Salinas scene that contained originally 224 spectral bands.

When looking at the average MCC-score for all 4 target detection algorithms in figure 4.10 the results does not follow a clear pattern across all detection algorithms. Both the ACE and CEM detector experience a decrease in performance when exposed to the MNF representation, but is not much affected by the other representations. The SAM detector have similar performance as ACE and CEM detector but does not suffer from the different DR representations in the same pattern. The OSP detector is also affected in a different pattern than the other algorithms, but cannot be said to perform with a similar MCC-score. A representation utilizing fewer components does not seem to decrease the performance further wrt. MCC-score for any of the 4 algorithms.

This suggests that the transformed covariance and correlation matrix, on a scene as Pavia represents, is not as straightforward to compute, especially so for the DR representations. This is expected as the scene is a non-homogeneous dataset. As both the covariance and correlation matrices are computed based on the sampled data, on a per image basis, and the DR representations can be said to remove commonalities shared between pixels, the resulting estimate for the matrices may be insufficient. Thus, indicating that a DR representation of a non-homogeneous dataset might be more demanding when considering the MCC-score. Furthermore, as expected, the estimated noise model does not seem to perform well under these circumstances either.

The average visibility of the different target detection algorithms in the Pavia scene is given in figure 4.12. None of the detection algorithms obtain an especially high average visibility score, irrespective of the chosen representation. The figure also shows that even though the CEM detector performed well wrt. MCC-score, it does not display robustness for this particular scene, having the lowest visibility score for all representations. Relatively speaking the ACE and SAM detector achieves a higher visibility score, with the ACE detector once again showing the highest visibility with the fewest components. The SAM detector seem to experience a non-trivial increase in visibility with the MNF representation, independent of the number of components used. This suggests that even with a poor noise model, irrelevant information is filtered away, and the MNF representation better separates the data. As the SAM detector does not utilize covariance nor correlation, this promotes the idea that a sample matrix should be calculated with a more consistent data set for better results.

To look at a more specific case, the ACE detector with the target being painted metal sheets and the MNF representation with 20 components were selected. The results from this particular scenario is given in table 4.2, taken from appendix D. In figure 4.12 a spatial representation of the visibility can be seen. once again, the MNF representation gives a higher probability of target to some non-target pixels, but at the same time a higher probability of target for target pixels. By closer inspection of the image titled MNF 20 and FULLDIM in figure 4.12, both representations seem to give a high probability of target for pixels of nearby positives when compared to the ground truth. It is not clear if this is due to poor classification of the original image, or poor performance of the detection of the proposed pipeline and the detection on the full spectral range.

Using the MNF representation with 20 components gives an MCC-score and visibility of 0,817 and 0,818 respectively. This is not the optimal score wrt. to either MCC or visibility obtained for this specific scene and endmember, but it differs from the optimal score with less than 0.13% for the MCC-score, and less than 2.4% for the visibility score. These are small variations when taking into account the disadvantage that this specific scene poses for the MNF representation. Furthermore, when performing the target detection on the full space an MCC-score and visibility of 0,815 and 0,619 are achieved. The improvement experienced wrt. MCC-score is probably due to random variations as it is a factor of less than 0.3%. When compared with the full spectral representation the visibility score increased with 32% for the MNF representation with 20 components. Even though this is not an equally large difference as the one experienced in the Salinas scene, it is still noteworthy. This argues that the MNF representation, even when presented with a lesser noise model estimate, is able to better separate the background and target for this particular scene, for this particular endmember, when compared to the full spectral space representation. Similar trends can be found for all detection algorithms for the MNF representation in this scene for all the investigated endmembers as well (see appendix D).

Table 4.2: Results from Pavia scene looking for painted metal sheets with ACE

DR	F1	MCC	$\|F_1$-MCC$\|$	Recall	Visbility
FULLDIM	0,808	0,815	0,008	0,944	0,619
PCA50	0,805	0,814	0,009	0,952	0,712
MNF50	0,809	0,818	0,009	0,946	0,664
ICA50	0,805	0,814	0,009	0,952	0,712
PCA20	0,810	0,817	0,007	0,946	0,838
MNF20	0,808	0,817	0,008	0,935	0,818
ICA20	0,810	0,817	0,007	0,946	0,838

The Pavia scene showcased the effects of a suboptimal noise model wrt. the MNF representation, and how the other DR representations are not as affected by the lack of a priori knowledge. The ICA and PCA representation did not experience a significant change in performance, both wrt. MCC-score and visibility. An expected result in the sense that none of the assumptions made in their implementation was violated. Both the ACE and the SAM algorithm performs rather well in this scene for the MNF representation with 20 components. The SAM detector narrowly outperforms the ACE detector on average, both wrt. MCC-score and visibility for this particular scene.

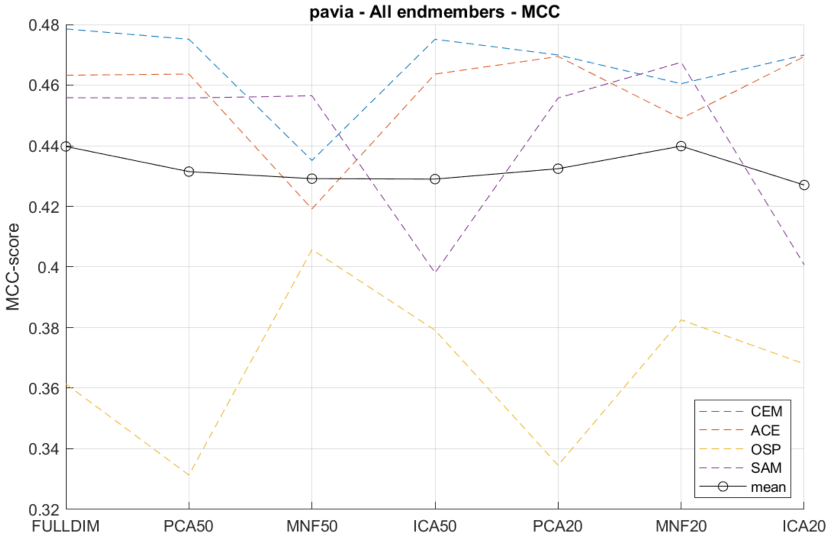


Figure 4.10: Average MCC-score for all target detection algorithms in the Pavia scene

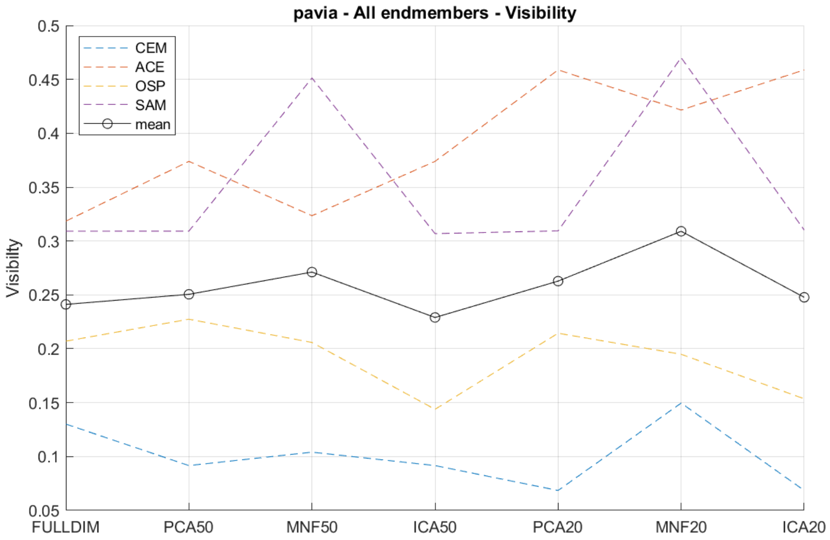


Figure 4.11: Average visibility score for all target detection algorithms in the Pavia scene

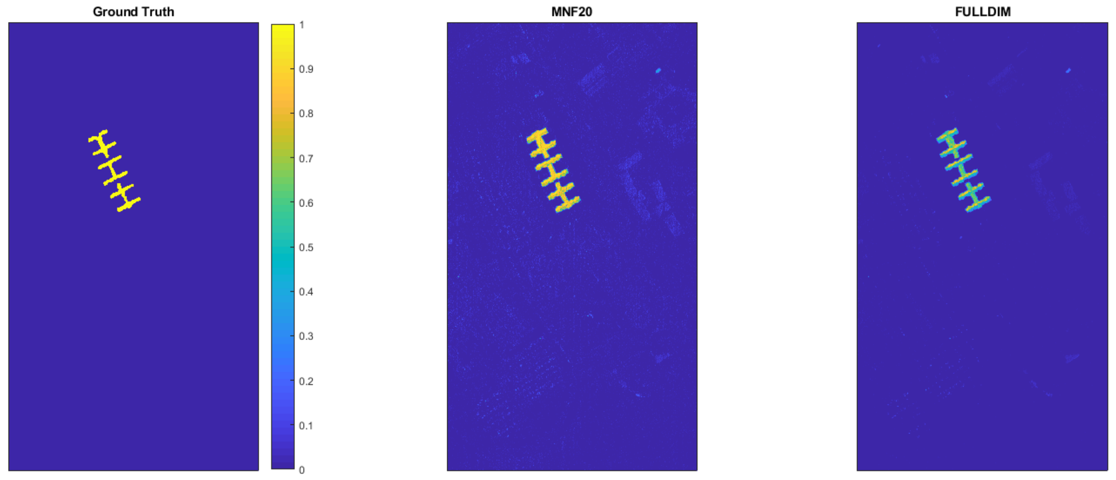


Figure 4.12: Visualization of the visibility produced by the ACE algorithm in the Pavia scene for painted metal sheets.

4.2.3 Target Detection on Hopavågen

The Hopavågen scene is the most interesting scene with respect to the NTNU SmallSat Program [5], as it has signatures closest resembling relevant signatures. It should be noted that an approach similar to the SAM detector was used to compare spectral signatures when manually classifying this scene. The image is given in figure 3.11 shows that the scene is somewhat homogeneous, but the assumption of large homogeneous areas wrt. neighboring pixels i.e. the MNF implementation, may not be optimal in this scene either. Furthermore, the scene was classified assuming that it contained 4 distinct classes, which in theory suggests that it should need at least 4 components to classify the image. However, for all the scenes given, the same number of components was used i.e. 20 and 50 components, for all the different DR representations. The endmember looked for in this scene is given in section 3.2.1, with a visual representation of their respective spectral variance.

In the Hopavågen scene, the original image has a spectral space that consists of 86 bands, thus a reduction to 50 and 20 components represents a compression ratio of about 1.72 and 4.3 respectively. This is the lowest level of compression ratio for any of the scenes given, but still, a noteworthy rate of compression when compared to what can be achieved on the full space directly.

In figure 4.13 it is clear that all detectors perform, on average, well wrt. MCC-score. The CEM detector outperforms all other detectors for all representations in this given scene, and it does not seem to be very affected by the different DR representations. The ACE detector shows a similar trend wrt. the different DR representations, but does not have the same performance wrt. MCC-score. The OSP and SAM detectors have the lowest performance wrt. MCC-score and they are both more affected by different DR representation. Even though the ACE and CEM detector is not very much affected of the different representations on average, both the OSP and SAM detectors experience a positive effect from the MNF representation when compared to the full space.

All of the different target detection algorithms experience a low level of visibility, on average, in this scene, as can be seen from figure 4.14. No combination of DR representation and detection algorithm gives a very high visibility score, as can be seen in appendix D, for the Hopavågen scene. The CEM detector is virtually unaffected by the different DR representations, but the ACE detector seems to experience a small improvement in performance with a lower number of components. Both the OSP and SAM detector is more affected by the different representation, with the SAM detector experiencing a substantial performance increase with MNF.

Thus, this scene also implies that target detection performed on the reduced space potentially will not harm the performance, while at the same time enabling a lower amount of data to be downlinked. That is, based on the averaged results, but the same trends can be found for the individual cases given in appendix D.

For the Hopavågen scene, the specific case of looking for Fucus serratus was selected, and the spatial visualization of this case, with the ACE detector and the MNF representation of 20 components, is given in figure 4.15. The results from this particular scenario is given in table 4.3, taken from appendix D. The MNF representation with 20 components seem to give a higher probability of target for non-targets, but an even higher probability of target for targets. Which has been observed in the other scenes as well, and also in this case the visibility score increased.

This pipeline gives an MCC-score and visibility of 0,913 and 0.341 respectively, with the ACE detector. When performing the target detection on the full space an MCC-score and visibility of 0.905 and 0.268 were obtained. The improvement in MCC-score may be due to random variations as it is a factor of less than 1 %. The MNF representation gave an increase of 27% wrt. visibility, but as the visibility score, for both full space and the MNF representation, was rather low it is hard to quantify the value of this improvement.

When conferring with appendix D, it is clear that the MNF representation with the SAM detector outperforms the MNF representation using the ACE detector. This difference is most noticeable when considering visibility. As the ACE detector does not seem to be very affected by representation, the SAM detector, on the other hand, is affected by the different representations, for this particular scene.

This specific case suggests that the MNF representation is an adequate approach for DR for compression, with an acceptable change in performance wrt. MCC-score or visibility, as they are both improved.

Table 4.3: Results from Hopavågen scene looking for Fucus Serratus with ACE

DR	F1	MCC	$\lvert F_1\text{-MCC}\rvert$	Recall	Visbility
FULLDIM	0,905	0,877	0,028	0,902	0,268
PCA50	0,900	0,870	0,030	0,890	0,304
MNF50	0,905	0,877	0,028	0,898	0,291
ICA50	0,900	0,870	0,030	0,890	0,304
PCA20	0,893	0,862	0,032	0,896	0,339
MNF20	0,913	0,887	0,026	0,903	0,341
ICA20	0,893	0,862	0,032	0,896	0,339

From figure 4.13 and 4.14, in combination with the specific case, it can be inferred that for this scene the proposed DR methods as a preliminary step before performing target detection may not decrease performance to an unsatisfactory degree. How well the noise model in the MNF representation is estimated, given how it is implemented, is difficult to quantify. The model assumptions are not well met in the scene, but at the same time, it is better suited than the Pavia scene. independent of this, the MNF representation does not decrease the performance of the detection a lot, independent of which detection algorithm is used.

The high visibility experienced by the SAM detector may be due to how the classification of pixels was performed.

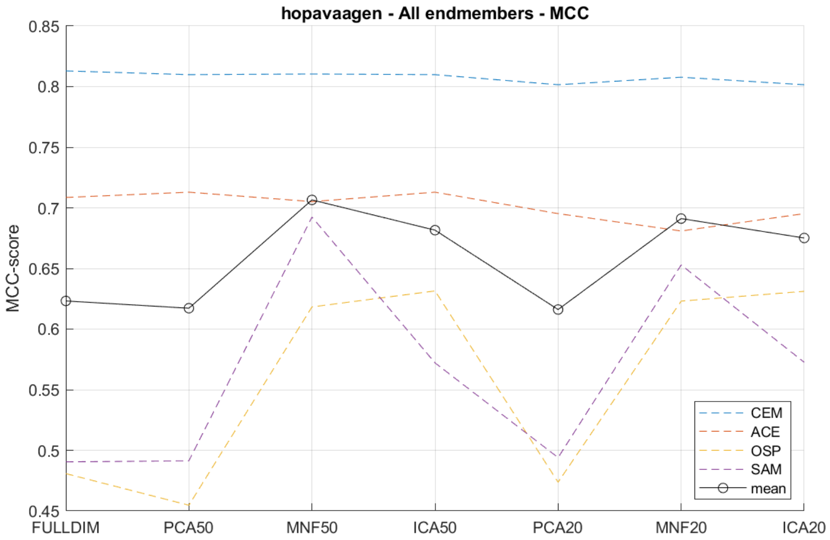


Figure 4.13: Average MCC-score for all target detection algorithms in the Hopavågen scene

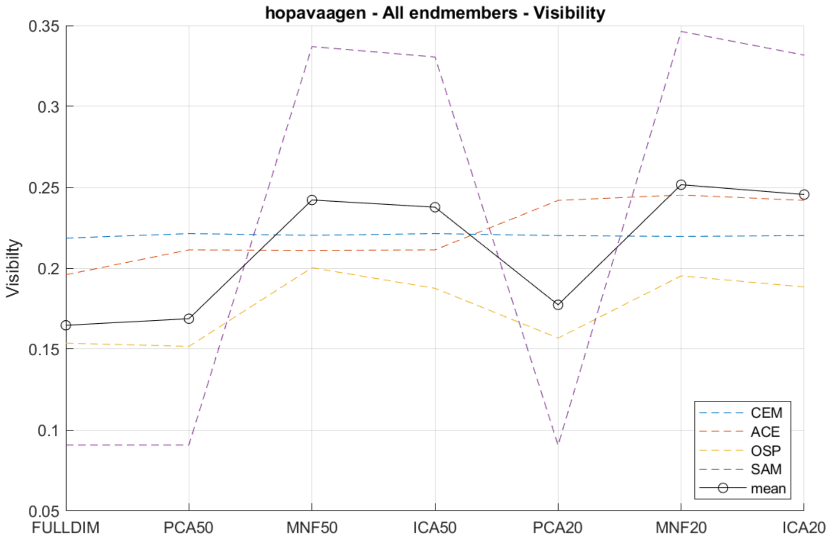


Figure 4.14: Average visibility score for all target detection algorithms in the Hopavågen scene

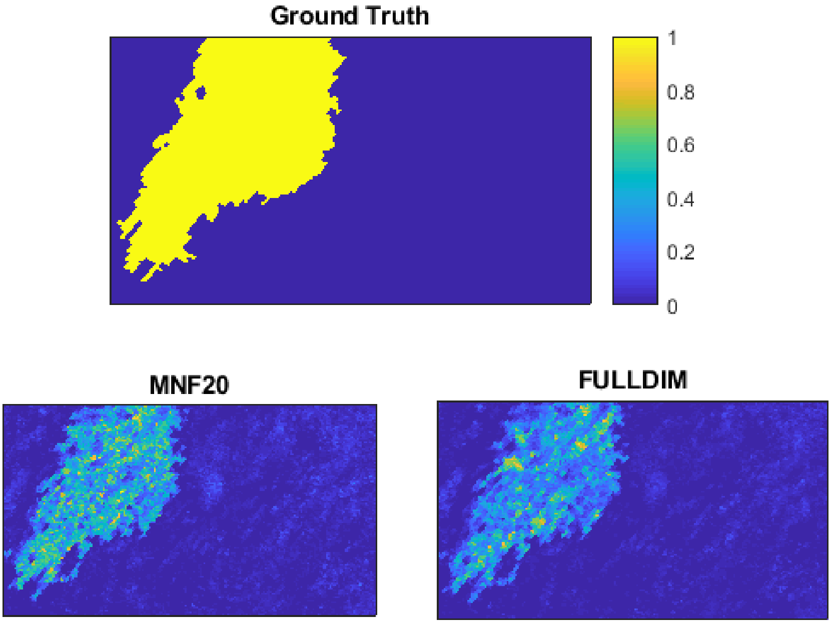


Figure 4.15: Visualization of the visibility produced by the ACE algorithm in the Hopavågen scene for Fucus serratus

Chapter 5

Conclusion and Further Work

This thesis has investigated the effects of dimensionality reduction methods such as PCA, MNF and JADE ICA on hyperspectral data, and how these representations would affect target detection. A proposed pipeline, utilizing the filtering obtained by reduced space representations, was analyzed for target detection algorithms ACE, CEM, OSP, and SAM. The proposed pipelines were tested on 3 different scenes with ground truth. The scenes were of different types in hopes to better display the general behavior to be expected from the proposed pipeline.

All the discussed DR representation gave an expected compression ratio, and all are well suited to be performed before performing other methods for compression.

The overall robustness with respect to visibility found for the ACE detector corresponds well with the favorable results found in other publications comparing detection algorithms [9, 10, 3, 11, 36]. The MNF representation performed well on the Salinas scene, where the assumptions made in the implementation were well met. With a better noise model, one not necessarily based on a per scene basis, the MNF representation should be able to further suppress undesired noise from a scene. As has been demonstrated, this may improve both the detection rate and robustness of a given application.

The simplicity of a potential implementation on an onboard processing unit for a small-sat system should also be taken into consideration. The most computationally demanding and complex of the surveyed target detection algorithms is the OSP detector, which also gave relatively poor results. The ACE and CEM detector are of similar complexity and gave similar results, but the ACE detector tended to be more robust wrt. visibility. Lastly, the SAM detector is the simplest detector and responded well to the MNF representation.

To better tailor a target detection solution to the specific application of ocean color observation, more data is needed. These preliminary results suggest that a good approach to investigate once an acceptable abundance of data is available is the MNF transform in combination with the ACE detector, or the SAM detector for simplicity.

5.1 Further Work

It could be beneficial for the system to have the covariance, and other significant statistical entities, based on a large dataset with relevant pixels specters calculated beforehand e.g. the DR transformation is not calculated on a per scene basis. This would save both time and power for the satellite, and could even be beneficial wrt. to performance. How well the transformation would be projected is difficult to say with the current level of relevant data available, and a thorough investigation of the cost and benefit of such an approach should be performed.

To mask out land-regions or regions that are of no interest would give a more consistent data set with a lower variance for the dimensionality reduction to be performed on. In return, this would better project the characteristics relevant for ocean color applications, such as the NTNU SmallSat program, and the subtle differences would emerge in the data. That is, techniques to label pixels on-the-fly could be useful for any system. Thus using a minimum of resources on uninteresting data. However, if the satellite were to be used for other applications, such as terrestrial purposes the sample data representing the basis for covariance and other significant statistical entities would need to be changed.

In the work performed in this thesis a static number of components was used. There exist methods to statistically estimate an optimal number of components. Such methods could be used to dynamically change the number of components used to better fit a given scene. An investigation of how well the methods estimate the dimensionality of the data, and how this would affect both compression and target detection could give useful insight.

Lastly, methods to better utilize the dimensionality of data, i.e. viewing the data not as a matrix of values but as a cube of values, should be investigated. A popular method in the literature concerning such multi-way methods could be PARAFAC[37]. An investigation comparing such methods to the two-dimensional approach taken in this thesis could give interesting insight.

Bibliography

[1] Bosoon Park and Renfu Lu. *Hyperspectral imaging Technology in Food and Agriculture*. 2015.

[2] Francesco DellEndice, Jens Nieke, Benjamin Koetz, Michael E Schaepman, and Klaus Itten. Improving radiometry of imaging spectrometers by using programmable spectral regions of interest. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(6):632–639, 2009.

[3] Dimitris Manolakis, David Marden, and Gary a Shaw. Hyperspectral Image Processing for Automatic Target Detection Applications. *Lincoln Laboratory Journal*, 14(1):79–116, 2003.

[4] project thesis as of fall 2017 by sivert bakken. `http://folk.ntnu.no/sivertba/project_thesis.pdf`. (Accessed on 11-04-2018).

[5] Mariusz E Grøtte, F Fortuna, Roger Birkeland, Julian Veisdal, Milica Orlandic, Harald Martens, J Tommy Gravdahl, Fred Sigernes, Tor A Johansen, Jan Otto Reberg, Geir Johnsen, and Kanna Rajan. Hyper-spectral imaging Small satellite in multi-agent marine observation system. *Internal Document*, 1(Non-published):1–16, 2017.

[6] Michael T. Eismann. *Hyperspectral Remote Sensing*. SPIE, 2012.

[7] Guangchun Luo, Guangyi Chen, Ling Tian, Ke Qin, and Shen-En Qian. Minimum Noise Fraction versus Principal Component Analysis as a Preprocessing Step for Hyperspectral Imagery Denoising. *Canadian Journal of Remote Sensing*, 42:106–116, 2016.

[8] Jing Wang and Chein I. Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 2006.

[9] D. Manolakis, R. Lockwood, T. Cooley, and J. Jacobson. Is there a best hyperspectral detection algorithm? *International Society for Optics and Photonics*, 7334(XV):733402, 2009.

[10] Xiaoying Jin, Scott Paswaters, and Harold Cline. A comparative study of target detection algorithms for hyperspectral imagery. *International Society for Optics and Photonics*, 7334(XV), 2009.

[11] Dimitris Manolakis, Eric Truslow, Michael Pieper, Thomas Cooley, and Michael Brueggeman. Detection algorithms in hyperspectral imaging systems: An overview of practical algorithms. *IEEE Signal Processing Magazine*, 2014.

[12] Ocean optics web book a collaborative web-based book on optical oceanography. `http://www.oceanopticsbook.info/`. (Accessed on 08-09-2017).

[13] Liguo Wang Chunhui Zhao and Hyperspectral. *Hyperspectral Image Processing*. 2016.

[14] M. Martens H. Martens. *Multivariate Analysis of Quality: An Introduction*. Wiley, 1 edition, 2000.

[15] Harald Martens, Tormod Næs, Lennart Eriksson, and Torbjrn Lundstedt. Multivariate calibration. I. Concepts and distinctions. *Trends in Analytical Chemistry*, 3(4):204–210, 2001.

[16] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 1901.

[17] Andrew A. Green, Mark Berman, Paul Switzer, and Maurice D. Craig. A Transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal. *IEEE Transactions on Geoscience and Remote Sensing*, 1988.

[18] Chein I. Chang. Interference and noise-adjusted principal components analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 1999.

[19] Chein-I Chang. *Appendix: Algorithm Compendium*, pages 997–1051. John Wiley & Sons, Inc., 2013.

[20] Aapo Hyvarinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. JOHN WILEY & SONS, INC, 2001.

[21] Frank Westad and Martin Kermit. Cross validation and uncertainty estimates in independent component analysis. In *Analytica Chimica Acta*, 2003.

[22] Michael E. Winter. N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data&lt;/title&gt;. *International Society for Optics and Photonics*, 3753(V), 1999.

[23] Marion F. Baumgardner, Larry L. Biehl, and David A. Landgrebe. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3, Sep 2015.

[24] Xudong Kang, Shutao Li, Leyuan Fang, Meixiu Li, and Jo Atli Benediktsson. Extended random walker-based classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 2015.

[25] Z-P. Lee. *Remote Sensing of Inherent Optical Properties: Fundamentals, Tests of Algorithms, and Applications*, volume No. 5 of *Reports of the International Ocean Colour Coordinating Group*. IOCCG, Dartmouth, Canada, 2006.

[26] About ecotone ecotone. `https://ecotone.com/om-oss/?lang=en`. (Accessed on 08-05-2018).

[27] Hyperspectral imagery synthesis tools for matlab - grupo de inteligencia computacional (gic). `http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB`. (Accessed on 26-03-2018).

[28] Usgs spectral library version 7 data - sciencebase-catalog. `https://www.sciencebase.gov/catalog/item/5807a2a2e4b0841e59e3a18d`. (Accessed on 26-03-2018).

[29] Add noise to image - matlab imnoise - mathworks nordic. `https://se.mathworks.com/help/images/ref/imnoise.html`. (Accessed on 05-05-2018).

[30] Robert L. Lucke, Michael Corson, Norman R. McGlothlin, Steve D. Butcher, Daniel L. Wood, Daniel R. Korwan, Rong R. Li, Willliam A. Snyder, Curt O. Davis, and Davidson T. Chen. Hyperspectral Imager for the Coastal Ocean: instrument description and first images. *Applied Optics*, 50(11):1501, 2011.

[31] Hyperspectral toolbox by davidkun. `http://davidkun.github.io/HyperSpectralToolbox/`. (Accessed on 07-02-2018).

[32] David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Flinders University*, 2011.

[33] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, 12:e0177678, 2017.

[34] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 6 2006.

[35] M. Wang. *Atmospheric Correction for Remotely-Sensed Ocean-Colour Products*, volume No. 10 of *Reports of the International Ocean Colour Coordinating Group*. IOCCG, Dartmouth, Canada, 2010.

[36] D Manolakis, M Pieper, E Truslow, T Cooley, M Brueggeman, and S Lipson. The remarkable success of adaptive cosine estimator in hyperspectral target detection. *Proc.SPIE*, 8743, 2013.

[37] Rasmus Bro. PARAFAC. Tutorial and applications. *Chemometrics and Intelligent Laboratory Systems*, 38(2):149–171, 10 1997.

Appendices

Appendix A

MATLAB code

A.1 Code regarding dimensionality reduction

Listing A.1: implementation of Maximum Noise Factoring

```matlab
function [M_new,A] = GreenMNF(M)

% Input   M: data set M is a hyperpsectral datacube.
% Output  M_new: 3D basis cube
%         A: Transformation matrix

[h,w,d] = size(M);
M = transpose(hyperConvert2d(M));
[m, n] = size(M);

% 1. Estimate the covariance of the noise.
dX = zeros(m-1,n);
for i=1:(m-1)
    dX(i,:) = M(i,:) - M(i+1,:);
end

% Take the eigenvector expansion of the covariance of dX
[U1,S1,V1] = svd(dX'*dX);

% Whiten the original data
wX = M*U1*inv(sqrt(S1));

% Compute the eigenvector expansion of the covariance of wX
[U2,S2,V2] = svd(wX'*wX);

```

```
% Define transformation matrix
A = U1*inv(sqrt(S1))*V2;

% Compute the Maximum noise fraction basis vectors
M_new = M*A;
A = A';
M_new = hyperConvert3d(M_new',h,w,d);
```

Listing A.2: implementation of ICA

```
function [Tica,IC] = myICA_CV(M,NumComponents)

[coeff,~,~,~,~,~] = pca(M);

P = coeff(:,1:NumComponents);

%    T = M*P;

nIC = NumComponents;
[IC, ~, ~] = jadeica(P',nIC);
Tica = M*IC'/(IC*IC');

function [icasig, A, W] = jadeica(x,m)

if (nargin < 2)
     A = matlabjader(x);
     W = A^(-1);
else
     A = matlabjader(x,m);
     W = pinv(A);
end


icasig = A*x;

return;
```

Listing A.3: implementation of JADE ICA

```
function [icasig, A, W] = jadeica(x,m);

if (nargin < 2),
    A = matlabjader(x);
    W = A^(-1);
else
    A = matlabjader(x,m);
    W = pinv(A);
end


icasig = A*x;

return;
```

Listing A.4: implementation of JADE ICA - provided code

```
function B =  jadeR(X,m)
% Blind separation of real signals with JADE.  Version 1.5 Dec. 1997.
%
% Usage:
%   * If X is an nxT data matrix (n sensors, T samples) then
%     B=jadeR(X) is a nxn separating matrix such that S=B*X is an nxT
%     matrix of estimated source signals.
%   * If B=jadeR(X,m), then B has size mxn so that only m sources are
%     extracted.  This is done by restricting the operation of jadeR
%     to the m first principal components.
%   * Also, the rows of B are ordered such that the columns of pinv(B)
%     are in order of decreasing norm; this has the effect that the
%     `most energetically significant' components appear first in the
%     rows of S=B*X.
%
% Quick notes (more at the end of this file)
%
%  o this code is for REAL-valued signals.  An implementation of JADE
%    for both real and complex signals is also available from
%    http://sig.enst.fr/~cardoso/stuff.html
%
%  o This algorithm differs from the first released implementations of
%    JADE in that it has been optimized to deal more efficiently
%    1) with real signals (as opposed to complex)
%    2) with the case when the ICA model does not necessarily hold.
%
%  o There is a practical limit to the number of independent
%    components that can be extracted with this implementation.  Note
%    that the first step of JADE amounts to a PCA with
```

```
   dimensionality
%    reduction from n to m (which defaults to n).  In practice m
%    cannot be `very large' (more than 40, 50, 60... depending on
%    available memory)
%
%  o See more notes, references and revision history at the end of
%    this file and more stuff on the WEB
%    http://sig.enst.fr/~cardoso/stuff.html
%
%  o This code is supposed to do a good job!  Please report any
%    problem to cardoso@sig.enst.fr


% Copyright : Jean-Francois Cardoso.  cardoso@sig.enst.fr

verbose = 0 ;   % Set to 0 for quiet operation


% Finding the number of sources
[n,T]   = size(X);
if nargin==1, m=n ; end;         % Number of sources defaults to # of sensors
if m>n ,    fprintf('jade -> Do not ask more sources than sensors here!!!\n'), return,end
if verbose, fprintf('jade -> Looking for %d sources\n',m); end ;



% Self-commenting code
%=====================
if verbose, fprintf('jade -> Removing the mean value\n'); end
X       = X - mean(X')' * ones(1,T);


%%% whitening & projection onto signal subspace
%   ===========================================
if verbose, fprintf('jade -> Whitening the data\n'); end
 [U,D]          = eig((X*X')/T) ;
 [puiss,k]      = sort(diag(D)) ;
 rangeW          = n-m+1:n                    ; % indices to the m most significant directions
 scales          = sqrt(puiss(rangeW))        ; % scales
  W              = diag(1./scales)  * U(1:n,k(rangeW))'   ; % whitener
 iW              = U(1:n,k(rangeW)) * diag(scales)        ; % its pseudo-inverse
 X               = W*X;


%%% Estimation of the cumulant matrices.
%   ====================================
 if verbose, fprintf('jade -> Estimating cumulant matrices\n'); end

dimsymm          = (m*(m+1))/2;  % Dim. of the space of real symm matrices
nbcm             = dimsymm  ;    % number of cumulant matrices
CM               = zeros(m,m*nbcm);  % Storage for cumulant matrices
R                = eye(m);       %%
 Qij             = zeros(m);     % Temp for a cum. matrix
Xim              = zeros(1,m);   % Temp
Xjm              = zeros(1,m);   % Temp
 scale           = ones(m,1)/T ; % for convenience



%% I am using a symmetry trick to save storage.  I should write a
%% short note one of these days explaining what is going on here.
%%
Range            = 1:m ; % will index the columns of CM where to store the cum. mats.
 for im = 1:m
        Xim = X(im,:)  ;
        Qij = ((scale* (Xim.*Xim)) .* X ) * X'  - R - 2 * R(:,im)*R(:,im)'  ;
        CM(:,Range)      = Qij  ;
        Range            = Range + m  ;
    for jm = 1:im-1
        Xjm = X(jm,:)  ;
        Qij = ((scale * (Xim.*Xjm) ) .*X ) * X' - R(:,im)*R
```

```
            (:,jm)' - R(:,jm)*R(:,im)' ;
        CM(:,Range)    = sqrt(2)*Qij ;
        Range          = Range + m ;
    end ;
end;

%%% joint diagonalization of the cumulant matrices
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Init
if 1,   %% Init by diagonalizing a *single* cumulant matrix. It seems to save
        %% some computation time `sometimes'. Not clear if initialization is
        %% a good idea since Jacobi rotations are very efficient.

        if verbose, fprintf('jade -> Initialization of the diagonalization\n'); end
        [V,D]   = eig(CM(:,1:m));       % For instance, this one
        for u=1:m:m*nbcm,               % updating accordingly the cumulant set given the init
                CM(:,u:u+m-1) = CM(:,u:u+m-1)*V ;
        end;
        CM      = V'*CM;

else,   %% The dont-try-to-be-smart init
        V       = eye(m) ; % la rotation initiale
end;

seuil   = 1/sqrt(T)/100; % A statistically significant threshold
encore  = 1;
sweep   = 0;
updates = 0;
g       = zeros(2,nbcm);
gg      = zeros(2,2);
G       = zeros(2,2);
c       = 0 ;
s       = 0 ;
ton     = 0 ;
toff    = 0 ;
theta   = 0 ;

```

```
%% Joint diagonalization proper
 if verbose, fprintf('jade -> Contrast optimization by joint diagonalization\n'); end

 while encore, encore=0;

   if verbose, fprintf('jade -> Sweep #%d\n',sweep); end
   sweep=sweep+1;

  for p=1:m-1,
   for q=p+1:m,

        Ip = p:m:m*nbcm ;
        Iq = q:m:m*nbcm ;

        %%% computation of Givens angle
        g       = [ CM(p,Ip)-CM(q,Iq) ; CM(p,Iq)+CM(q,Ip) ];
        gg      = g*g';
        ton     = gg(1,1)-gg(2,2);
        toff    = gg(1,2)+gg(2,1);
        theta   = 0.5*atan2( toff , ton+sqrt(ton*ton+toff*toff) );

        %%% Givens update
        if abs(theta) > seuil,  encore = 1 ;
                updates = updates + 1;
                c       = cos(theta);
                s       = sin(theta);
                G       = [ c -s ; s c ] ;

                pair            = [p;q] ;
                V(:,pair)       = V(:,pair)*G ;
                CM(pair,:)      = G' * CM(pair,:) ;
                CM(:,[Ip Iq])   = [ c*CM(:,Ip)+s*CM(:,Iq) -s*CM(:,Ip)+c*CM(:,Iq) ] ;

                %% fprintf('jade -> %3d %3d %12.8f\n',p,q,s);

        end%%of the if
   end%%of the loop on q
  end%%of the loop on p
end%%of the while loop
 if verbose, fprintf('jade -> Total of %d Givens rotations\n
',updates); end

%%% A separating matrix
%   ===================
B       = V'*W ;

%%% We permut its rows to get the most energetic components first.
%%% Here the **signals** are normalized to unit variance. Therefore,
%%% the sort is according to the norm of the columns of A = pinv(B)

if verbose, fprintf('jade -> Sorting the components\n',updates); end
A                 = iW*V ;
[vars,keys]       = sort(sum(A.*A)) ;
B                 = B(keys,:);
B                 = B(m:-1:1,:) ; % Is this smart ?

% Signs are fixed by forcing the first column of B to have
% non-negative entries.
if verbose, fprintf('jade -> Fixing the signs\n',updates); end
b       = B(:,1) ;
signs   = sign(sign(b)+0.1) ; % just a trick to deal with sign=0
B       = diag(signs)*B ;


return ;

% To do.
%   - Implement a cheaper/simpler whitening (is it worth it?)
%
% Revision history:
%
%-  V1.5, Dec. 24 1997
%   - The sign of each row of B is determined by letting the first
%     element be positive.
%
%-  V1.4, Dec. 23 1997
%   - Minor clean up.
%   - Added a verbose switch
%   - Added the sorting of the rows of B in order to fix in some
%     reasonable way the permutation indetermination.  See note 2)
%     below.
%
%- V1.3, Nov.  2 1997
%   - Some clean up.  Released in the public domain.
%
%- V1.2, Oct.  5 1997
%   - Changed random picking of the cumulant matrix used for
%     initialization to a deterministic choice.  This is not because
%     of a better rationale but to make the ouput (almost surely)
%     deterministic.
%   - Rewrote the joint diag. to take more advantage of Matlab's
%     tricks.
%   - Created more dummy variables to combat Matlab's loose memory
%     management.
%
%- V1.1, Oct. 29 1997.
%    Made the estimation of the cumulant matrices more regular. This
%    also corrects a buglet...
%
%- V1.0, Sept. 9 1997. Created.
%
% Main reference:
% @article{CS-iee-94,
%  title         = "Blind beamforming for non {G}aussian signals",
%  author        = "Jean-Fran\c{c}ois Cardoso and Antoine Souloumiac",
%  HTML          = "ftp://sig.enst.fr/pub/jfc/Papers/iee.ps.gz",
%  journal       = "IEE Proceedings-F",
%  month = dec, number = 6, pages = {362-370}, volume = 140, year = 1993}
%
%  Notes:
```

```
% =======
%
% Note 1)
%
% The original Jade algorithm/code deals with complex signals in
% Gaussian noise white and exploits an underlying assumption that the
% model of independent components actually holds.  This is a
% reasonable assumption when dealing with some narrowband signals.
% In this context, one may i) seriously consider dealing precisely
% with the noise in the whitening process and ii) expect to use the
% small number of significant eigenmatrices to efficiently summarize
% all the 4th-order information.  All this is done in the JADE
% algorithm.
%
% In this implementation, we deal with real-valued signals and we do
% NOT expect the ICA model to hold exactly.  Therefore, it is
% pointless to try to deal precisely with the additive noise and it
% is very unlikely that the cumulant tensor can be accurately
% summarized by its first n eigen-matrices. Therefore, we consider
% the joint diagonalization of the whole set of eigen-matrices.
% However, in such a case, it is not necessary to compute the
% eigenmatrices at all because one may equivalently use `parallel
% slices' of the cumulant tensor.  This part (computing the
% eigen-matrices) of the computation can be saved: it suffices to
% jointly diagonalize a set of cumulant matrices.  Also, since we are
% dealing with reals signals, it becomes easier to exploit
```

```
the
% symmetries of the cumulants to further reduce the number of
% matrices to be diagonalized.  These considerations, together with
% other cheap tricks lead to this version of JADE which is optimized
% (again) to deal with real mixtures and to work `outside the model'.
% As the original JADE algorithm, it works by minimizing a `good set'
% of cumulants.
%
% Note 2)
%
% The rows of the separating matrix B are resorted in such a way that
% the columns of the corresponding mixing matrix A=pinv(B) are in
% decreasing order of (Euclidian) norm.  This is a simple, `almost
% canonical' way of fixing the indetermination of permutation.  It
% has the effect that the first rows of the recovered signals (ie the
% first rows of B*X) correspond to the most energetic *components*.
% Recall however that the source signals in S=B*X have unit variance.
% Therefore, when we say that the observations are unmixed in order
% of decreasing energy, the energetic signature is found directly as
% the norm of the columns of A=pinv(B).
%
% Note 3)
%
% In experiments where JADE is run as B=jadeR(X,m) with m varying in
% range of values, it is nice to be able to test the stability of the
% decomposition.  In order to help in such a test, the rows of B can
% be sorted as described above. We have also decided to fix the sign
%  of each row in some arbitrary but fixed way.  The convention is
%  that the first element of each row of B is positive.
%
%
%  Note 4)
%
%  Contrary to many other ICA algorithms, JADE (or least this version)
%  does not operate on the data themselves but on a statistic (the
%  full set of 4th order cumulant).  This is represented by the matrix
% CM below, whose size grows as m^2 x m^2 where m is the number of
%  sources to be extracted (m could be much smaller than n).  As a
%  consequence, (this version of) JADE will probably choke on a
%  `large' number of sources.  Here `large' depends mainly on the
%  available memory and could be something like 40 or so.  One of
%  these days, I will prepare a version of JADE taking the `data'
%  option rather than the `statistic' option.
%
%



% JadeR.m ends here.
```

A.2 Code regarding target detection

Listing A.5: Implementation of Adaptive Cosine Estimator

```
function [results] = hyperAce(M, S)
% HYPERACE Performs the adaptive cosin/coherent estimator algorithm
%   Performs the adaptive cosin/coherent estimator algorithm for target
% detection.
%
% Usage
%   [results] = hyperAce(M, S)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   S - 2d matrix of target endmembers (p x q)
% Outputs
%   results - vector of detector output (N x 1)
%
% References
%   X Jin, S Paswater, H Cline.  "A Comparative Study of Target Detection
% Algorithms for Hyperspectral Imagery."  SPIE Algorithms and Technologies
% for Multispectral, Hyperspectral, and Ultraspectral Imagery XV.  Vol
% 7334.  2009.


[p, N] = size(M);
% Remove mean from data
u = mean(M.').';
M = M - repmat(u, 1, N);
S = S - repmat(u, 1, size(S,2));

R_hat = hyperCov(M);
G = inv(R_hat);

results = zeros(1, N);
% From Broadwater's paper
%tmp = G*S*inv(S.'*G*S)*S.'*G;
tmp = (S.'*G*S);
for k=1:N
    x = M(:,k);
    % From Broadwater's paper
    %results(k) = (x.'*tmp*x) / (x.'*G*x);
    results(k) = (S.'*G*x)^2 / (tmp*(x.'*G*x));
end
```

Listing A.6: Implementation of Constrained Energy Minimization

```matlab
function [results] = hyperCem(M, target)
% HYPERCEM Performs constrained energy minimization (CEM) algorithm
%   Performs the constrained energy minimization algorithm for target
% detection.
%
% Usage
%   [results] = hyperCem(M, target)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   target - target of interest (p x 1)
% Outputs
%   results - vector of detector output (N x 1)
%
% References
%   Qian Du, Hsuan Ren, and Chein-I Cheng. A Comparative Study of
% Orthogonal Subspace Projection and Constrained Energy Minimization.
% IEEE TGRS. Volume 41. Number 6. June 2003.

% Check dimensions
if ndims(M) ~= 2
    error('Input image must be p x N.');
end

p = size(M,1);

if ~isequal(size(target), [p,1])
    error('Input target must be p x 1.');
end

% CEM uses the correlation matrix, NOT the covariance matrix. Therefore,
% don't remove the mean from the data.
R_hat = hyperCorr(M);

% Equation 6 : w = inv( target'*inv(R)*target ) * inv(R)*target
invRtarget = R_hat\target; % inv(R)*target
weights    = ( target'*invRtarget ) \ invRtarget;

results    = weights'*M;
```

Listing A.7: Implementation of Spectral Angle Mapper

```
function [results] = mySam(M, target)
% mySam Performs Spectral angle mapper algorithm
%
%
% Usage
%   [results] = mySam(M, target)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   target - target of interest (p x 1)
% Outputs
%   results - vector of detector output (N x 1)
%
% References

% Check dimensions
if ndims(M) ~= 2
    error('Input image must be p x N.');
end

p = size(M,1);

if ~isequal(size(target), [p,1])
    error('Input target must be p x 1.');
end

results = zeros(size(M,2),1);

r_sam = @(x,s) (s'*x)^2 / ((s'*s)*(x'*x));

for i = 1:length(results)
    results(i,:) = r_sam(M(:,i),target);
end
```

Listing A.8: Implementation of Orthogonal Subspace Projection

```
function [nu] = hyperOsp(M, U, target)
% HYPEROSP Performs the othogonal subspace projection (OSP) algorithm
%   Performs the othogonal subspace projection algorithm for target
% detection.
%
% Usage
%   [results] = hyperOsp(M, U, target)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   U - 2d matrix of background endmebers (p x q)
%   target - target of interest (p x 1)
% Outputs
%   results - vector of detector output (N x 1)
%
% References
%   Qian Du, Hsuan Ren, and Chein-I Cheng. ”A Comparative Study of
% Orthogonal Subspace Projection and Constrained Energy Minimization.”
% IEEE TGRS. Volume 41. Number 6. June 2003.

[p, N] = size(M);

% Equation 3
P_U = eye(p) - U * pinv(U);

% For abundance estimation
% Equation 4
%w_osp = inv(target.'*P_U*target) * P_U * target;

tmp = target'*P_U*target;
nu = zeros(N, 1);
for k=1:N
    nu(k) = (target'*P_U*M(:,k))/tmp;
end
```

Listing A.9: Implementation of N-FINDR

```
function [U] = hyperNfindr(M, q)
% HYPERNFINDR Performs the N-FINDR (endmember extraction) algorithm
%   Performs the N-FINDR algorithm to find q endmembers. If only M is
%  given as input, this function calls hyperHfcVd to estimate the number
%  of endmembers (q) and then hyperPct to reduce dimensionality to (q-1).
%
% Usage
%   [U] = hyperNfindr(M)
%   [U] = hyperNfindr(M, q)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   q - Number of endmembers to find
%       -- if not given, q is obtained from hyperHfcVd(M, 10^-3)
% Outputs
%   U - Recovered endmembers (p x q)
%
% References
%   M. Winter, "N-findr: an algorithm for fast autonomous
% spectral endmember determination in hyperspectral data," SPIE s
% International Symposium on Optical Science, Engineering, and
% Instrumentation, pages 266 275. International Society for Optics
% and Photonics, 1999.

% Error trapping
if ndims(M) ~= 2
    warning('WarnTests:dim', ...
            'Input image must be p x N.\n',...
            'Converting with hyperConvert2d.\n')
    M = hyperConvert2d(M);
end

M_orig = M;
[p, N] = size(M);

if nargin == 1
    fprintf('Implementing hyperHfcVd to determine the
        number of endmembers.\n')
    q = hyperHfcVd(M_orig, [10^-3]);
    fprintf('Reducing dimensionality to (q-1) using hyperPct.\n')
    M = hyperPct(M, q-1);
elseif q < p+1
    warning('WarnTests:dim', ...
    strcat('N-FINDR requires (q-1) spectral bands.\n',...
           'Performing PCA to reduce dimensionality.\n'))
    M = hyperPct(M, q-1);
elseif q > p+1
    warning('WarnTests:dim', ...
    strcat('N-FINDR requires (q-1) spectral bands.\n',...
           'Performing PCA to reduce dimensionality.\n'))
    error('ErrTests:dim', ...
        strcat('N-FINDR cannot find more than (p+1) endmembers (q),\n', ...
            'where p is the number of available spectral bands.\n'))
end

% Initialize
U_idx = randperm(N,q); % Random endmember selection
E     = M(:,U_idx);    % Endmember matrix
V     = abs(det([ones(1,q); E])) / factorial(q-1); % Simplex volume
vols  = zeros(q,1);

% Search for maximum volume simplex
for j = 1:N;
    % Replace each column of E with sample vector M(:,j)
    % and compute the volume for each
    for k = 1:q;
        E_tmp      = E;
        E_tmp(:,k) = M(:,j);
        vols(k)    = abs(det([ones(1,q); E_tmp])) / factorial(q-1);
    end
    % If max volume is greater than previous V, update E and V
    [V_tmp,k_idx] = max(vols);
    if V_tmp > V
        V           = V_tmp;
        E(:,k_idx)  = M(:,j);
        U_idx(k_idx) = j;
    end
end

% Return endmembers
U = M_orig(:, U_idx);
```

A.3 Code used in simulations

Listing A.10: How the simulations were ran, only line 36 to the comment ”calculate score” was changed between the different dimensionality reduction methods

```
 1 td_test_init;
 2
 3 %% test 1 - loop
 4 score_csv = 'Name;score;c/a;w/a;\n';
 5 score_struct = struct;
 6
 7 for s = 1:length(scenes)
 8     for temp = 1:size(ids,2)
 9         for a = 1:length(td_algs)
10             id = ids(s,temp);
11
12             %set params
13             scene = gt_data_set.(scenes(s));
14             M                    = scene.cube;
15             %M                    = scene.corrected;
16             gt                   = scene.gt;
17             end_name     = scene.endmembers(id);
18             end_sign     = scene.signatures(id,:);
19             %end_sign     = scene.signatures_corrected(id,:);
20             end_index    = id;
21             abundance    = scene.abundance(id);
22             td_alg               = td_algs(a);
23             %end set params
24
25             %Calculate score
26             if td_alg == "hyperOsp"
27                 numEndmembers = length(scene.endmembers);
28                 B = getBackground(M, numEndmembers,
                       end_sign);
29                 [a1, b1, ~] = td_results(M, gt, end_name,
                       end_sign,...
30                     end_index, abundance, scenes(s), td_alg
                           , B);
31
32             else
33                 [a1, b1, ~] = td_results(M, gt, end_name,
                       end_sign,...
34                     end_index, abundance, scenes(s), td_alg
                           );
35             end
36
```

```matlab
        b1 = normalize(b1);
        % Measure score
        xx = size(gt,1); yy = size(gt,2);

        tp = 0; tn = 0;
        fp = 0; fn = 0;

        P_FA = 0.0001:0.0001:1.0000;
        met = zeros(length(P_FA),3);

        for i = 1:length(P_FA)
            for j = 1:xx
                for k = 1:yy
                    if and(gt(j,k) == id, b1(j,k) >= P_FA(i))
                        tp = tp + 1;
                    elseif and(gt(j,k) == id, b1(j,k) < P_FA(i))
                        fn = fn + 1;
                    elseif b1(j,k) >= P_FA(i)
                        fp = fp + 1;
                    else
                        tn = tn + 1;
                   end
               end
           end

           met(i,:) = [metric(tp,tn,fp,fn); tp; fp;];
           tp = 0; tn = 0;
           fp = 0; fn = 0;
        end

        % Calculate Visibility
        T_t_sum = 0; T_t_count = 0;
        T_b_sum = 0; T_b_count = 0;

        for j = 1:xx
            for k = 1:yy
                if gt(j,k) == id
                    T_t_sum      = T_t_sum + b1(j,k);
                    T_t_count    = T_t_count +1;
                else
                    T_b_sum      = T_b_sum + b1(j,k);
                    T_b_count    = T_b_count +1;

                    end
                end
            end
            T_t_avg = T_t_sum / T_t_count;
            T_b_avg = T_b_sum / T_b_count;

            T_max = max(b1(:)); T_min = min(b1(:));

            vis = norm(T_t_avg - T_b_avg)/(T_max - T_min);

            % store score
            [max_score, max_index] = max(met(:,1));
            p_c_a = met(max_index,2)/abundance;
            p_f_a = met(max_index,3)/abundance;

            name = replace(join([erase(td_alg,'hyper')
                replace(scenes(s),'_','') end_name]),' ','_'
                );
            result = sprintf('%s;%d;%d;%d\n',name,max_score
                ,p_c_a,p_f_a);
            score_csv = join([score_csv result]);

            end_name_alt = replace(end_name,' ','_');
            end_name_alt = replace(end_name_alt,'-','_');
            alg_name = replace(td_alg,'hyper','');

            result_struct.(scenes(s)).(end_name_alt).(
                alg_name).('ca') = p_c_a;
            result_struct.(scenes(s)).(end_name_alt).(
                alg_name).('wa') = p_f_a;
            result_struct.(scenes(s)).(end_name_alt).(
                alg_name).('score') = max_score;
            result_struct.(scenes(s)).(end_name_alt).(
                alg_name).('abundance') = abundance;
            result_struct.(scenes(s)).(end_name_alt).(
                alg_name).('visibility') = vis;

        end
    end
end

fid = fopen('result_full.csv','a');
fprintf(fid, score_csv);
fprintf(fid, '\n');
fclose(fid);
```

```

disp('done!');
```

Listing A.11: The same initialization was used across all simulations

```
clc; clear; close all;
%% load
[file, path] = uigetfile();
load([path file]);

%% parameters
%assuming the gt_data_set struct is used
td_algs = ["hyperAce", "hyperOsp", "hyperCem", "mySam"];

if string(file) == "hopavaagen_data.mat"
    gt_data_set.hopavaagen = hopavaagen;
    scenes = ["hopavaagen"];
    ids = [ 1 3 4;
            3 0 0;
            4 0 0];
else
    scenes = ["pavia", "salinas", ... "indian_pines",
                                  "hopavaagen"];
    ids = ...
        [1 4 5;     % scene 1
        11 12 14;   % scene 2
%         1 7 9;     % scene 3
        1 3 4];     % scene 4
end

result_struct = struct;

q = 50;

%metric for performance
mcc = @(tp,tn,fp,fn) (tp*tn - fp*fn)/(sqrt((tp+fp)*(tp+fn)*(tn+fp)*(tn+fn)));
f1  = @(tp,tn,fp,fn) (2*tp)/(2*tp+fp+fn);

metric = mcc;
```

Listing A.12: How the different target detection algorithms were tested.

```matlab
function [fig, res_img, res_prob] = td_results(M, gt, end_name, end_sign, end_index, abundance, scene_name, td_alg, background)

% M,          A 3 dimensional datacube
% gt,         a spatial 2d representation of the different classes in the image
% end_name,   the name of the endmember to be detected
% end_sign,   the signature of the endmember to be detected
% end_index,  the index of the endmember to be detected
% abundance,  the number of target pixels in the image
% td_alg,     the function name of the algorithms used to do detection
% background the estimated backgroudn if relevant for the detection algorithm

%number_of_bins = round(0.25*abundance);

switch nargin
    case 8
        background = -1;
    case 9
        %Do nothing;
    otherwise
        disp('Wrong number of arguments');
end

[h,w,d] = size(M);
from_3d_to_2d = @(M) reshape(M, w*h, d).';
from_2d_to_3d = @(M,h,w,d) reshape(M.', h, w, d);

end_sign = end_sign';

M2d = from_3d_to_2d(M);

if background == -1
    res_2d = feval(td_alg,M2d, end_sign);
else
    res_2d = feval(td_alg, M2d, background, end_sign);
end

res_prob = zeros(1,abundance);
res_img = from_2d_to_3d(res_2d,h,w,1);
[rows, columns] = find(gt == end_index);
```

```

for i = length(rows)
    res_prob(i) = res_img(rows(i),columns(i));
end

fig = histogram(res_prob,20);
title([end_name erase(td_alg,'hyper') replace(scene_name,'_',' ')]);

fig = 3;

end
```

A.4 Other utility code

Listing A.13: Implementation of false color image function

```
function [x_img] = create_rgb_from_bil(x_bil,R,G,B,FRA, OS)
% Create RGB Image
x_img = x_bil(:,:,[R G B]);


% The variable FRA roughly controls the contrast.
% Increase the value of FRA to make the image brighter.
switch nargin
    case 4
        FRA = 0.02;
        OS = 0.5;
    case 5
        OS = 0.5;
end

% scale the radiances from 0 to 1
RA = (1.0*x_img(:,:,1))/max(max(x_img(:,:,1)));
GA = (1.0*x_img(:,:,2))/max(max(x_img(:,:,2)));
BA = (1.0*x_img(:,:,3))/max(max(x_img(:,:,3)));

RA = OS+(FRA/std(std(RA))*(RA - mean(mean(RA))));
GA = OS+(FRA/std(std(GA))*(GA - mean(mean(GA))));
BA = OS+(FRA/std(std(BA))*(BA - mean(mean(BA))));

RA(RA>1) = 1; RA(RA<0) = 0;
GA(GA>1) = 1; GA(GA<0) = 0;
BA(BA>1) = 1; BA(BA<0) = 0;

x_img(:,:,1) = RA;
x_img(:,:,2) = GA;
x_img(:,:,3) = BA;

%figure;
imagesc(x_img); axis off; axis image;

end
```

Listing A.14: Implementation of the N-FINDR algorithm as found in the toolbox described in section 3.3.2

```
function [U] = hyperNfindr(M, q)
% HYPERNFINDR Performs the N-FINDR (endmember extraction) algorithm
%   Performs the N-FINDR algorithm to find q endmembers. If only M is
%  given as input, this function calls hyperHfcVd to estimate the number
%  of endmembers (q) and then hyperPct to reduce dimensionality to (q-1).
%
% Usage
%   [U] = hyperNfindr(M)
%   [U] = hyperNfindr(M, q)
% Inputs
%   M - 2d matrix of HSI data (p x N)
%   q - Number of endmembers to find
%       -- if not given, q is obtained from hyperHfcVd(M, 10^-3)
% Outputs
%   U - Recovered endmembers (p x q)
%
% References
%   M. Winter, "N-findr: an algorithm for fast autonomous
% spectral endmember determination in hyperspectral data," SPIE s
% International Symposium on Optical Science, Engineering, and
% Instrumentation, pages 266 275. International Society for Optics
% and Photonics, 1999.

% Error trapping
if ndims(M) ~= 2
    warning('WarnTests:dim', ...
            'Input image must be p x N.\n',...
            'Converting with hyperConvert2d.\n')
    M = hyperConvert2d(M);
end

M_orig = M;
[p, N] = size(M);

if nargin == 1
```

```
    fprintf('Implementing hyperHfcVd to determine the number of endmembers.\n')
    q = hyperHfcVd(M_orig, [10^-3]);
    fprintf('Reducing dimensionality to (q-1) using hyperPct.\n')
    M = hyperPct(M, q-1);
elseif q < p+1
    warning('WarnTests:dim', ...
    strcat('N-FINDR requires (q-1) spectral bands.\n',...
          'Performing PCA to reduce dimensionality.\n'))
    M = hyperPct(M, q-1);
elseif q > p+1
    warning('WarnTests:dim', ...
    strcat('N-FINDR requires (q-1) spectral bands.\n',...
          'Performing PCA to reduce dimensionality.\n'))
    error('ErrTests:dim', ...
        strcat('N-FINDR cannot find more than (p+1) endmembers (q),\n', ...
            'where p is the number of available spectral bands.\n'))
end

% Initialize
U_idx = randperm(N,q); % Random endmember selection
E     = M(:,U_idx);    % Endmember matrix
V     = abs(det([ones(1,q); E])) / factorial(q-1); % Simplex volume
vols  = zeros(q,1);

% Search for maximum volume simplex
for j = 1:N;
    % Replace each column of E with sample vector M(:,j)
    % and compute the volume for each
    for k = 1:q;
        E_tmp       = E;
        E_tmp(:,k)  = M(:,j);
        vols(k)     = abs(det([ones(1,q); E_tmp])) / factorial(q-1);
    end
    % If max volume is greater than previous V, update E and V
    [V_tmp,k_idx] = max(vols);
    if V_tmp > V
        V             = V_tmp;
        E(:,k_idx)    = M(:,j);
        U_idx(k_idx) = j;
    end
end

% Return endmembers
U = M_orig(:, U_idx);
```

Appendix B

Instrument Specifications

In this part of the appendix a short descriptions of the different instruments are given. Unfortunately they do not follow the same structure, but most of the useful information is there.

Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)

Robert O. Green, Michael L. Eastwood, Charles M. Sarture, Thomas G. Chrien, Mikael Aronsson, Bruce J. Chippendale, Jessica A. Faust, Betina E. Pavri, Christopher J. Chovit, Manuel Solis, Martin R. Olah, and Orlesa Williams

Imaging spectroscopy is of growing interest as a new approach to Earth remote sensing. The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) was the first imaging sensor to measure the solar reflected spectrum from 400 nm to 2500 nm at 10 nm intervals. The calibration accuracy and signal-to-noise of AVIRIS remain unique. The AVIRIS system as well as the science research and applications have evolved significantly in recent years. The initial design and upgraded characteristics of the AVIRIS system are described in terms of the sensor, calibration, data system, and flight operation. This update on the characteristics of AVIRIS provides the context for the science research and applications that use AVIRIS data acquired in the past several years. Recent science research and applications are reviewed spanning investigations of atmospheric correction, ecology and vegetation, geology and soils, inland and coastal waters, the atmosphere, snow and ice hydrology, biomass burning, environmental hazards, satellite simulation and calibration, commercial applications, spectral algorithms, human infrastructure, as well as spectral modeling. ©*Elsevier Science Inc., 1998*

INTRODUCTION

Spectroscopy is used in the laboratory in the disciplines of physics, chemistry, and biology to investigate material properties based on the interaction of electromagnetic radiation with matter. Imaging spectroscopy in the solar reflected spectrum was conceived for the same objective, but from the Earth looking and regional perspective (Fig. 1). Molecules and particles of the land, water and atmosphere environments interact with solar energy in the 400–2500 nm spectral region through absorption, reflection, and scattering processes. Imaging spectrometers in the solar reflected spectrum are developed to measure spectra as images in some or all of this portion of this spectrum. These spectral measurements are used to determine constituent composition through the physics and chemistry of spectroscopy for science research and applications over the regional scale of the image.

To pursue the objective of imaging spectroscopy, the Jet Propulsion Laboratory proposed to design and develop the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) in 1983. AVIRIS first measured spectral images in 1987 and was the first imaging spectrometer to measure the solar reflected spectrum from 400 nm to 2500 nm (Fig. 2). AVIRIS measures upwelling radiance through 224 contiguous spectral channels at 10 nm intervals across the spectrum. These radiance spectra are measured as images of 11 km width and up to 800 km length with 20 m spatial resolution. AVIRIS spectral images are acquired from the Q-bay of a NASA ER-2 aircraft from an altitude of 20,000 m. The spectral, radiometric, and spatial calibration of AVIRIS is determined in laboratory and monitored inflight each year. More than 4 TB of AVIRIS data have been acquired, and the requested data has been calibrated and distributed to investigators since the initial flights.

AVIRIS has measured spectral images for science research and applications in every year since 1987. More than 250 papers and abstracts have been written for the

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California

Address correspondence to R. O. Green, JPL Mail-Stop 306-438, 4800 Oak Grove Dr., Pasadena, CA 91109-8099. E-mail: rog@gomez.jpl.nasa.gov

Received 24 June 1998; accepted 8 July 1998.

REMOTE SENS. ENVIRON. 65:227–248 (1998)
©Elsevier Science Inc., 1998
655 Avenue of the Americas, New York, NY 10010

0034-4257/98/$19.00
PII S0034-4257(98)00064-9

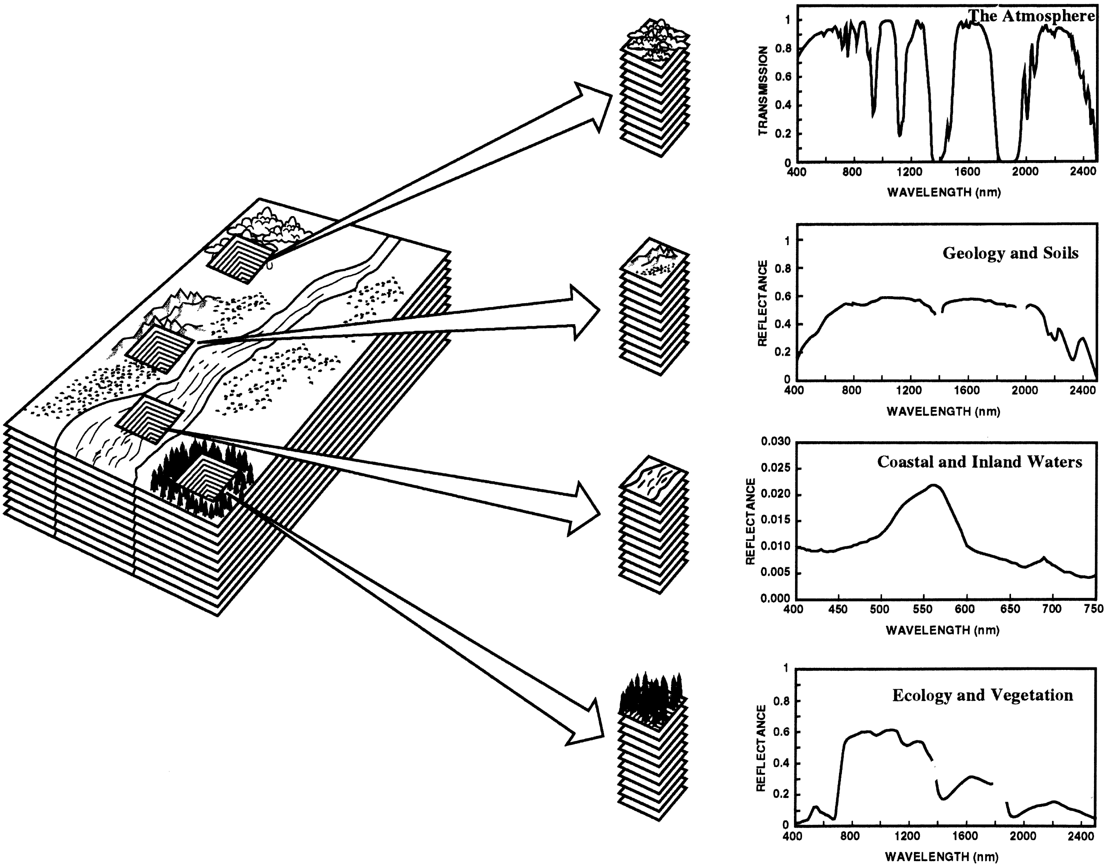


Figure 1. The concept of imaging spectroscopy is shown with a spectrum measured for each spatial element in a image. The spectra are analyzed for science research and applications in a range of disciplines.

AVIRIS workshops (Vane, 1988, Green, 1990a; 1991; 1992; 1993; 1995; 1996a). The workshop documents and additional information are maintained on the AVIRIS website (http://makalu.jpl.nasa.gov/AVIRIS.html). In the past 10 years, there have been a comparable number of AVIRIS papers written for other workshops, symposia, and conferences. A previous special journal issue related to AVIRIS has been published (Vane, 1993). There are additional AVIRIS related articles and papers throughout the remote sensing literature.

The AVIRIS system has been upgraded and improved in a continuous effort to meet the requirements of investigators using AVIRIS spectral images for science research and applications. These improvements have been directed towards the AVIRIS sensor, calibration, data system, and flight operations. In parallel with the sensor, the science research and applications pursued with AVIRIS have diversified and evolved. This article describes the characteristics and recent improvements to the AVIRIS system. The article also reviews a range of science research and applications results and objectives pursued with AVIRIS and provides a context for the accompanying articles in this journal special issue.

SENSOR

AVIRIS is a sophisticated and complex optical sensor system involving a number of major subsystems, components, and characteristics (Table 1). Taken together, these result in the AVIRIS data characteristics (Table 2).

The AVIRIS sensor receives white light in the foreoptics, disperses the light into the spectrum, converts the photons to electrons, amplifies the signal, digitizes the signal and records the data to high density tape. The major subsystems of the sensor are the: scan mirror, foreoptics, spectrometers, detectors, onboard calibrator, and electronic signal chain (Fig. 3). The initial design contained many opportunities for improvements with advances in

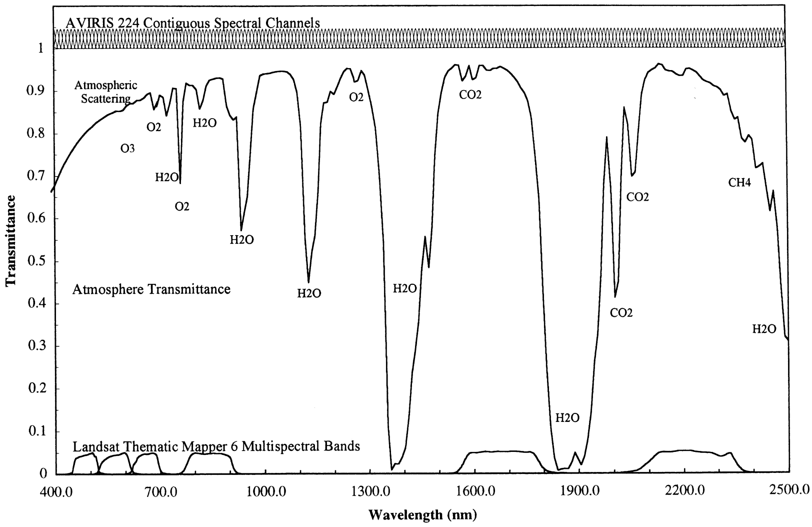


Figure 2. The 224 spectral channels of AVIRIS are shown with a transmittance spectrum of the atmosphere as well as the six solar reflected bands of the Landsat Thematic Mapper.

technology. The initial design, subsequent improvements, and the signal path through AVIRIS are described.

The upwelling radiance arriving at AVIRIS in flight (Fig. 4) enters through at 2.54-cm-thick hatch window in the ER-2 aircraft. This hatch window is composed of low-water fused silica and has an average transmittance of better than 0.95 across the spectrum. An antireflection coating was applied to the hatch window to further improve transmittance. Light arriving from the hatch window passes through the AVIRIS aperture door and is reflected into the foreoptics by an oscillating whiskbroom scan mirror. The aperture door is closed to protect the scan mirror and foreoptics and automatically opened only when spectral images are being measured. The door replaces a protective aperture window that reduced throughput and introduced a small amount of scattered light into the AVIRIS foreoptics. The scan mirror is silver-coated and triangular in cross section with two primary facets giving a 10 cm by 20 cm equivalent area. The oscillating scan drive sweeps the mirror across the active 30° field of view of the Earth at a rate of 12 Hz. Significant engineering effort was required to develop a scan drive to sweep linearly across the 30° field-of-view and then fly back at nearly twice the speed to start the next imaging scan (Miller, 1987). The AVIRIS scan drive approaches 70% efficiency. The great advantage of a whiskbroom scanner is that the light for every spatial element passes through the same path of the optical system. This gives exceptional uniformity for the 614 cross-track spatial samples in each AVIRIS image scan line.

After reflection from the scan mirror the light from the two facets is recombined using a set of fold flats and then focused using a paraboliod and elliptical mirror. A final fold flat is used to direct the focused light to an array of optical fibers. The first three mirrors are coated with high-reflectivity silver, the last two mirrors are aluminum. The 200 mm effective focal length of the foreoptics and the 200 μm diameter entrance aperture of the fibers define the 1 milliradian instantaneous field of view (IFOV) of the AVIRIS sensor. A mechanical shutter is

Table 1. AVIRIS Sensor Characteristics

Imager type	Whiskbroom scanner
Scan rate	12 Hz
Dispersion	Four grating spectrometers (A,B,C,D)
Detectors	224 detectors (32, 64, 64, 64) Si and InSb
Digitization	12 bits
Data rate	20.4 mbits/s
Spectrum rate	7300 spectra/s
Data capacity	>10 GB (>8000 km^2)

Table 2. Nominal AVIRIS Data Characteristics

Spectral	
Wavelength range	400–2500 nm
Sampling	10 nm
Spectral response (fwhm)	10 nm
Calibration accuracy	<1 nm
Radiometric	
Radiometric range	0 to maximum lambertian radiance
Sampling	~1 DN noise RMS
Absolute calibration	⩾96%
Inter flight stability	⩾98%
Signal-to-noise	Exceeding 100:1 requirement
Polarization sensitivity	⩽1%
Spatial (at 20 km altitude)	
Field of view	30 degrees (11 km)
Instantaneous FOV	1.0 mrad (20 m)
Calibration accuracy	⩽0.1 mrad
Flight line length	800 km total

ROSIS - An imaging spectrometer enviromental research

Heinz van der Piepen
Institute for Optoelectronic, DFVLR , D-8031 Oberpfaffenhofen, FRG

Roland Doerffer
Institute for Physics, GKSS, D-2054 Geesthacht, FRG.

Bernd Kunkel
Space Systems Group, MBB, D-8012 Ottobrunn, FRG

Abstract

The Reflective optics system imaging spectrometer (ROSIS) is a compact programmable imaging spectrometer based on a CCD matrix detector array . The instruction has been designed specially for the monitoring for the of water color and of natural chlorophyll fluorescence in order to quantitatively derive pigments suspended matter and yellow substance distributions in the marine environment how ever its high spectral resolution of ≤ 5 nm also permits many new air borne new application in vegetation monitoring and in atmospheric physics an air borne prototype IS ROSIS IS jointly developed at present by MBB GKSS and DFVLR the instrument concept the scope of applications and the relationship to ESA earth observation programme and to NASA earth observation system is discussed.

Imaging spectroscopy

The future observation system by NASA and the earth observation programme by ESA rely to a large extent on the imaging spectrometers like the high resolution imaging spectrometer the moderate resolution imaging spectrometer or the medium resolution imaging spectrometer in addition to other operational sensors these are expect to largely improve the monitoring capabilities for climatology and environment studies in regard to ocean land and atmospheric parameters.

The main advantage of imaging spectrometers in comparison to conventional optional mechanical multi spectral scanners is the availability of a large number of narrow band width spectral channel combined with the positively of an applications specific selection of a few channels for data recording or transmission this permits the use of a single for a variety of monitoring tasks sensor alternatives focus instead on either medium of high geometric resolution (HIRIS, HRIS).

The Narrow band width channels permit a detailed analysis of spectral fine structures which are present in many signatures related to marine biology pigment fluorescence water pollution vegetation stress land use geology atmospheric absorption features etc. As a result these sensors will permit an essential improvement of data interpretation for environmental monitoring.

In preparation of NASA and ESA future space programmers a few air borne prototypes of such sensors have been developed and used during the past few years Typical examples are the air borne imaging spectrometer from the U.S jet propulsion Laboratory or the Fluorescence line manager from the Canadian Department of Fishers and oceans while the former was designed mainly for the monitoring of mineralogical features the design driver for the latter was based on earlier experience with modern CCD technology a new sensor for water color monitoring from space was studied during 1986/87 by the company MBB on observation mission with ESA's European Retrievable Carrier an air borne prototype of this under construction the instrument conception its future applications are discussed in the following sections .

ROSIS

1. **Cooperation agreement**
   Based on the EURECA / ROSIS experience an agreement between the organizations GKSSS MBB and DFVLR was signed in 1987 with the aim to develop jointly an aircraft prototype of ROSIS .In order to meet future requirements in regard to ESA's polar platform missions there aircrafts version was to be developed as closely as possible to the spacecraft version studied already before. And after through tests in to laboratory the instrument will be ready for the first flight tests in early 1990.

2. **Optical system**
   In order to meet the stringent radiometric requirements of a water color florescence sensor throughout the relevant spectral range the optical system incorporates reflectance components only (Figure 1):

   i. The image on ground is relayed through a baffle via a lift mirror the purpose of which is to shift the scan line either forward or aft for sun glint avoidance, onto the

   ii. two telescope mirrors, which focus the image on to the entrance slit of the spectrometer (the entrance slit represents the actual scan line on ground by cutting off the rest FOV);

iii. the focused scan line image is expanded and paralcllized through a collimator system (two spherical mirrors) for

iv. dispersion by means of a reflective grating (bottom of the housing);

v. the collimating system (using again the same spherical mirrors) subsequently focuses the beam via a small deflection mirror onto the CCD detector (top of housing);

vi. the trigger and read-out electronics is arranged on top of the array so as to avoid long connections

The front-end tilt mirror also serves as a means to reflect diffuse sun light into the system for calibration purposes(the reverse slide is covered by a diffuser).Further more, the same mirror is used in an intermediate position as shutter to enable dark current measurements.

The off-axis system compensates partially for the slit curvature in the focal plane. The optical performance data are summarized in Table1.

Figure 1. ROSIS optical scheme.

3. **Dectator array**

A matrix detector array of the type Thomson CSF model THX 31156 is incorporated in to the design (Table2).Thisallows1024 picture elements to be used across the scan perpendicular to the flight direction 85 spectral channel corresponding to the spectral range from 430 to 850 nm can be used in the spectral mode. all the other detector elements on the elements on the array are masked or used as intermediate storage.

However since the above mentioned detector will become only available later the present ROSIS instrument will be temporarily operated with the detector model TH 7884 which permits the use of only 500 detector elements across the scan line this can be mounted either off axis for tests of the system or alternatively in

Table 1. ROSIS optical performance data

Total FOV	±16 degrees
IFOV	0.56 mrad
F Number	3.6 Distortion
	≤ 2 %
Grating constant	n = 55
Blaze angle	1.01 degrees
Spectral angle	430 - 860 nm
Spectral range	5 nm / detector element
Tilt	± 20 degrees

Table 2. ROSIS detector array

Present Type	Thomson CSF TH 7884
Lines / Columns	512 X 500
Element size	23.5 x 18.5 μm
Dynamic range	3300 :1
operation mode	frame transfer
Future Type	Thomson CSF THX 31156
Lines / Columns	1024 X 1024
Element size	19.0 x 19.0 μm
Dynamic range	5000 : 1
operation mode	frame transfer

the center for flight operations with FOV of 16 degrees this detector will be replaced once the large one will become commercially available .

4. **Operation modes**

Similar to the FLI the air borne ROSIS can be operated either in the spatial or alternatively in the spectral mode (Table 3).

The spatial mode allows the full geometric resolution to be recorded in up to 32spectral channels .the spatial mode allows all 85 spectral channels to be recorded simultaneously at a reduced spatial resolution the center wave length can be adjusted electronically in 1 mm steps.

5. **Data recording**
The control of ROSIS, the real-time quick look and the recording of data is done by means of a multi processors based VME bus system using OS/9 as real time operating system all memories on processor and inter face modules are dual-ported.

Table 3. Rosis operation modes

Imaging mode:

500 pixel across track (with detector TH 7884
any of 65 wavelength selectable upon command
recording of up to 32 selected channels.

Spectral mode:

85 adjacent spectral channels (430 850 nm)
every third pixel or across track.

The data from ROSIS are transferred in to system as a block of one frame with up to 16 k 12 bit words. data are accepted from the aircraft inertial navigation system and from other instruments via an ancillary data inter face processor the final frame is built up in the main memory and then transformation to the disc controller which packs the information and writes it on the storage medium this will be a 51 /4 erasable will once optical disc with 1 GB storage capacity.

The control processor checks the dynamic range of the radiance data and the environmental of the instrument it allows the operator to display the numeric or graphic form and to program the sampling mode adjustment a protocol of all actions is recorded on floppy disc the quick look image processor with color display provides the operator with a continuously updated image of the data.

The data rate is at present maximum 85 frames per second with 2.2 MB/s.

6. **Future developments**
After tests flights to be performed by GKSS and DFVLR it is a planned to incorporate a wider user community in to the ROSIS data and application by means of an extensive flight programme.

The present aircraft prototype of ROSIS will eventually be modified as indicated with the larger detector array so as to permit imaging of the full 32 degree FOV for which the optical system has been designed further developments may include an extension of the special range in to the short wave infrared so to be compactable with sensors like the thematic mapped or similar.

In regard to space flights further studies are presently performed so as investigate in more detail the possibilities of applying the ROSIS concept to the requirements of MERIS and MODIS T incorporated in ESA EOP and NASA EOS as part of the Columns space programmed

The rapidly increasing demand for environmental monitoring may also lead to a combined mission for monitoring the atmosphere plus and the coastal environmental vegetation stress water pollution and Biomass through during a period after NASA Upper Atmospheric research Satellite Mission and prior to the EOP /EOS programme i.e in the middle of the next decade investigations are taking asoas the determine how ROSIS could be modified to include the ultra spectral range for measuring solar back scattered radiation around nadir.

Applications
Monitoring of the water color fluorescence permits applications in the fields of marine biology and ecology water pollution and sediment transport citatory and monitoring of dynamic features.

It is expected that the narrow spectral bands of ROSIS in combination with the programmable channel selection bands of ROSIS in water color monitoring in parameters in case II waters where the presence of components such as yellow substance or sediments restrict the use of color ratios fore a quantities interpretation of spectral .

The possibility to adapt the spectral channels to the pigments of different populations will improve the sensitivity and specify of the system for monitoring also exceptional restricts blooms by means of color ratios or by inverse modeling of spectral .

In additions it will permit a precise selection of several atmosphere correction channels for the aerosol determination as well as suitable channels for establishing a base for an evaluations of the fluorescence signal.

Future land applications in context with NASA EOS will include especially the advanced monitoring to tropical rain forest vegetation index stress plant diseases and land use. it is expected that the high spectral resolution will also permit an analysis of the red edge shift of the pant albedo associated with stress features .

The precise positioning of narrow band channels is expected to open addition measurements also in the field of atmospheric physics which sofar could be dealt with if at all only with active optical or micro wave sensors. These aspects include

- Cloud height atmospheric pressure determination by comparative analysis of radiances originating from the surface and cloud tops.
- Optical depth of clouds and droplet size determination by means of comparison of different liquid water absorption bands.
- Water vapour column content through a comparison of relative differences of radiance ratios with in out side water vapour rotation bands.
- acrosol type and concentration determination from radiances especially in the near infrared region.

Acknowledgements
The authors wish to express their particular acknowledgements to Prof..H. GRABI MPI, Mr. W. Cordes and DR. J. FischeR, GKSS, Dr. D. Beran, MR. M.Mooshuber Mr. W.W. Schrooder, DFLVIR as well as to the MBB team (Mr. F. Blechinger, DR. R. Buschner, MR. Herbig, DR. R. Lutz, MR. D. Vichmann and MR. H. Wolter, MR. R.Ziegler) for their effort and dedications in the planning developing and test of the prototype of ROSIS.

References

- NASA, "MODIS instrument panel report" Earth observation system VOL. 11b (1986).
- NASA "HIRIS instrument panel report" Earth obdservation system VOL HC (1987).
- ESA, "Imaging spectrometry fror land applications," Proc. of an workshop held at Frascati, 19-21 APRIL 1988, ESA SP-1101, OCT.1988.
- C.O. Davis, "Future U.S. ocean coor missions - OCI, MODIS, HIRIS," in Occanpography from space, J.F.R. Gower ed., Advances in space research 7 (2), 3-9 (1987)
- A.B. Hollinger, L.H. Gray J.F.R. Gower and H,R. Edwl, "The fluorescence line imager: An imaging spectrometer for ocean and land remote sensing" Proc. SPIE. 834, 2-11(1987).
- MBB "Chlorphlyll mapping of the ocean by means of detector array sensors" (Feasibility and definition study) BMFT contract No. 01 QS 8508 8, Final Report (1988).
- GKSS, "The use of chlorophyll fluorescence measurements from space for separating constituments water," ESA contract No. RFQ 3-5059 / 84 / NL / MD, Vol I (Summary Report) and vol II (Appendices (1986).
- R. Doerffer V. Amann and H van der Piepen, "Remote sensing of exceptional plankton blooms, "Proc. ICES Special Meeting on causes, Dynamic and Effects of exceptional Marine Blooms and related evrnts, Copenhagen. 1984

ecotone

Scientific UHI

Underwater hyperspectral imaging system

Hardware

The UHI is a hyperspectral imager optimized for underwater use. This opens up an entirely new layer of detail in remote sensing previously only available for airborne-based platforms. With the UHI, the field of hyperspectral imaging is now available for subsea applications. The UHI is capable of acquiring information in the entire visible color spectrum (370-800nm - adjustable). Data from the UHI can thus be used to detect biological and chemical characteristics of seafloor habitats, as well.

The UHI-system integrates with most modern ROV-systems using off-the-shelf connectors and Ethernet-connection. It can also be independently operated on other underwater platforms. Mounted with a clear view of the seafloor and flanked by dedicated lights, the UHI will acquire data frame-by-frame as the platform moves. The data archives (hyperspectral cube, navigation data, video*) is stored locally and can be transferred topside

Hyperspectral imaging

A hyperspectral imager can record the full electromagnetic spectrum in each image pixel. This is a representation of the light energy reflected and absorbed off of an object, and can be used to detect and discriminate unique characteristics of materials and features. Objects of different colors and materials will have different spectral response in the image, and the identifying features can be used as an “optical fingerprint” pertaining to that particular object.

Software

The unit is delivered complete with control software necessary to acquire and preprocess data. All hyperspectral and ancillary data is stored in a structured archive file. The control software enables communication with the deployed unit, and provides the following functions:

- Live view and acquisition of hyperspectral data (RGB and spectrum) and video stream
- Adjust spatial/spectral binning
- Adjust framerate/integration time and gain
- Parsing, synchronizing and storing of navigation data from ROV and GPS.

ecotone

UHI Technical Specifications	
UHI model	UHI4
Depth rating	3000 m (Other depth options on request)
Imager Specifications	
Imaging range	0.2m to 5m typical (Depends on illumination)
Imager update rate:	Adjustable up to 65 hz
Field of view transverse:	~ 60° standard (Spec. upon order)
Field of view longitudinal:	~ 0.4° standard (Spec. upon order)
Camera resolution:	1920 spatial pixels (resolution adjustable via spatial binning)
Data Range	12-bit
Spectral Range	370-800 nm (resolution adjustable via spectral binning)
Focus	Manual adjustable
Interface Specifications	
Communication:	Ethernet (standard)
Input voltage:	12 to 36 VDC
Power consumption:	Max. 35 W (20W typical)
Integrated computer	
Integrated computer	Intel i7, 8 GB ram, Linux OS
Internal data storage:	1TB M.2 SSD
System startup time:	30 sec
Software Specifications	
Data acquisition and preprocessing	“Immersion”, acquisition software for hyperspectral data, video and navigation data
File compatibility	Compatible with ENVI and other spectral analysis applications.
Environmental Specifications	
Operation temperature (in water):	-5°C to +30°C
Operation temperature (in air):	-5°C to +20°C
Storage temperature:	-20°C to +50°C
Mechanical Specifications	
Weight without mounting bracket (air/water)	Approx. 11/6 kg (3000m version)
Size (Length x Diameter)	Approx. 355 x 135 mm (3000m version)
Connector Type:	Subconn DFCR2013M (standard)
Housing Material:	Titanium GR 5
Front port material:	Fused silica
Delivery Specifications	
Included in standard delivery	• UHI and software • Connector dust cap • Pelicase for transport and storage • Power supply
Optional equipment/services	
Upgraded internal data storage	One additional 1TB SSD
Quick mounting bracket	Flexible mounting bracket for integration on an underwater platform. Carrying handle included.
Integrated IMU	Integrated Inertial Measurement Unit (IMU) for high-resolution attitude information (pitch, roll, yaw) of the UHI.
Integrated video camera	5 MP (2592 x 1944) camera, providing live view and video recording of the same field of view as the hyperspectral camera.
Contact Information	
	support@ecotone.com

Table 1. HICO Parameters

	Value	Comment
ISS		
Orbital altitude	≈350 km	May rise to 400 km over a 1–3 year mission
Inclination	51.6°	HICO's latitude range ≈ + 54°/ − 53°; it covers 80% of the Earth's surface
Period	91.5 min	15.7 orbits/day; the HICO can do one scene per orbit
Ground speed	7.0 km/s	Speed of sub-ISS point on ground, includes Earth rotation
HICO Hardware Lens		
FL	67.16 mm	Gives 83 m nadir IFOV for 16 μm pixels and 350 km altitude
Clear aperture	19 mm	$f/3.5$ gives desired radiometrics
FOV	6.92° (±3.46°)	Covers 512 cross-track pixels (see Subsection 2.C)
Spectrometer		
Slit width	16 μm	Matches CCD pixel size
Blaze wavelength of grating	400 nm	High efficiency in first-order blue light
Blazed fraction of groove	≈90% of groove spacing	High efficiency
CCD		CCD97 from e2v Technologies
Pixel size	16 μm	Gives 83 m nadir IFOV for 16 μm pixels and 350 km altitude
IFOV	0.01365°	See Eq. (2)
Array size, total used	512 × 512 512 (spatial) × 384 (spectral)	One spatial pixel = 83 m at 350 km altitude; one spectral pixel = 1.91 nm
Spectral sample width (normal)	5.73 nm	Three-pixel on-chip binning
Spectral sample width (HR mode)	1.91 nm	On-orbit wavelength calibration check (no binning)
Digitization	14 bits	Maximum DN = 16,383
Dark noise	≈3.8 DN	Dynamic range: 16,000/3.8 = 4200
Photoelectrons/DN	26 ± 2	Photon-limited SNR = $\sqrt{26 \times \mathrm{DN}}$
Operations		
Stable exposure time	12.64 ms (28.94 ms in HR mode)	See Subsection 2.A and Appendix A
Frame transfer time	1.11 ms	See Subsection 2.A and Appendix A
Exposure time (total frame time)	13.75 ms (12.64 + 1.11) (30.05 ms in HR mode)	Along-track GSD = 96 m (210 m in HR mode)
Scene time	275 s (0.01375 × 2000) 15 s (0.03 × 500) in HR mode	Scene = 2000 frames in normal mode and 500 frames in HR mode
Data volume normal mode		
Data bytes/frame	512 × 128 × 2 = 131 KB	BIL format
Frames per observation	200 dark + 2000 scene + dark = 2400 total	(13.75 ms/frame) × (2000 frames) ⇒ 2.75 s to record scene
Data bytes/scene	131 KB × 2400 = 315 MB	—
Data volume HR (hi-res.) mode		
Data bytes/frame	512 × 384 × 2 = 393 KB	BIL format
Frames per observation	100 dark + 500 scene + 100 dark = 700	(30.05 ms/frame) × (500 frames) ⇒ 15 s to record scene
Data bytes/scene	393 KB × 700 = 275 MB	—
Performance		
Off-nadir pointing (normal flight mode)	45° to port (north), 30° to starboard (south)	Accessible ground swath ≈550 km
Observed ground swath width	42 km (nadir, 350 km altitude)	Up to 92 km at 45° off nadir
GSD	83 m cross-track (nadir, 350 km altitude)	Proportional to altitude
GSD	96 m along track (210 m in HR mode)	Proportional to frame time
Scene length	192 km (2000 × 96 m) (105 km in HR mode)	Maximum, can be shorter
Scene area	(42 km wide) × (192 km long) ≈8000 km^2 (nadir view)	Encompasses the scale of coastal dynamics
Spectral range	Nominal: 350–1080 nm; best data: 400–900 nm	—
Spatial FWHM	1.6 pixels typical (130 m at nadir, 350 km)	See Fig. 3
Spectral FWHM	1.9 pixels typical (3.6 nm)	See Fig. 5
Keystone	<0.2 pixels (0.4 nm), mostly	Minimal spatial smearing; see Fig. 4
Spectral tilt	<0.15 pixels (0.3 nm), mostly	Minimal spectral smearing; see Fig. 6
Spectral smile	Negligible	See Fig. 6
Polarization sensitivity	<4% for most wavelengths	See Fig. 7
SNR	>200:1 over 400–600 nm for shallow water (see Fig. 13)	For 11.46 nm spectral bins, allows removal of atmospheric effects

Appendix C

Compression on HICO and Synthetic Data

C.1 Compression on HICO

On the following page the results discussed in section 4.1.1 is given. The file names corresponds with figure 3.19 from left to right in numbering. The names were given by the on-board processing group. The ”missing” fifth file was not available when the simulation was performed. It was processed by the NASA SeaDAS processing software[12], and is thus not an example of unprocessed data.

Results from compression on HICO

Files	HICO L1B_1		HICO L1B_2		HICO L1B_3		HICO L1B_4		HICO L1B_6	
Methods	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR	PSNR	CR
PCA 5	36.33	25.60	35.50	25.60	38.31	25.60	54.33	25.60	35.50	25.60
PCA 10	48.54	12.80	60.25	12.80	54.38	12.80	58.95	12.80	62.66	12.80
PCA 20	60.58	6.40	73.60	6.40	69.91	6.40	72.67	6.40	74.77	6.40
PCA 30	75.58	4.27	81.05	4.27	80.18	4.27	78.55	4.27	81.25	4.27
PCA 40	83.46	3.20	84.62	3.20	84.34	3.20	81.93	3.20	84.50	3.20
PCA 50	86.42	2.56	87.21	2.56	87.01	2.56	84.85	2.56	87.22	2.56
MNF 5	29.48	25.60	31.42	25.60	28.64	25.60	36.42	25.60	29.88	25.60
MNF 10	29.59	12.80	32.72	12.80	29.50	12.80	37.88	12.80	31.24	12.80
MNF 20	30.32	6.40	33.34	6.40	31.32	6.40	38.38	6.40	32.42	6.40
MNF 30	32.73	4.27	37.61	4.27	33.42	4.27	38.97	4.27	36.74	4.27
MNF 40	36.35	3.20	39.80	3.20	34.17	3.20	45.76	3.20	38.21	3.20
MNF 50	36.95	2.56	40.36	2.56	35.38	2.56	48.69	2.56	39.37	2.56
JADE ICA 5	41.57	25.60	40.65	25.60	38.69	25.60	54.44	25.60	39.25	25.60
JADE ICA 10	52.49	12.80	61.37	12.80	57.32	12.80	58.92	12.80	59.68	12.80
JADE ICA 20	60.61	6.40	70.20	6.40	63.36	6.40	63.37	6.40	64.92	6.40
JADE ICA 30	66.27	4.27	71.76	4.27	66.15	4.27	66.91	4.27	67.27	4.27
JADE ICA 40	69.32	3.20	72.21	3.20	68.28	3.20	67.44	3.20	67.88	3.20
JADE ICA 50	70.36	2.56	73.13	2.56	69.32	2.56	67.88	2.56	68.75	2.56

C.2 Compression on Synthetic Data Sets

In figure C.1 all the abundance plot of the synthetic files are given.

On the following page the results discussed in section 4.1.2 is given. Figures from this simulation can be found at `http://folk.ntnu.no/sivertba/master_thesis/`.

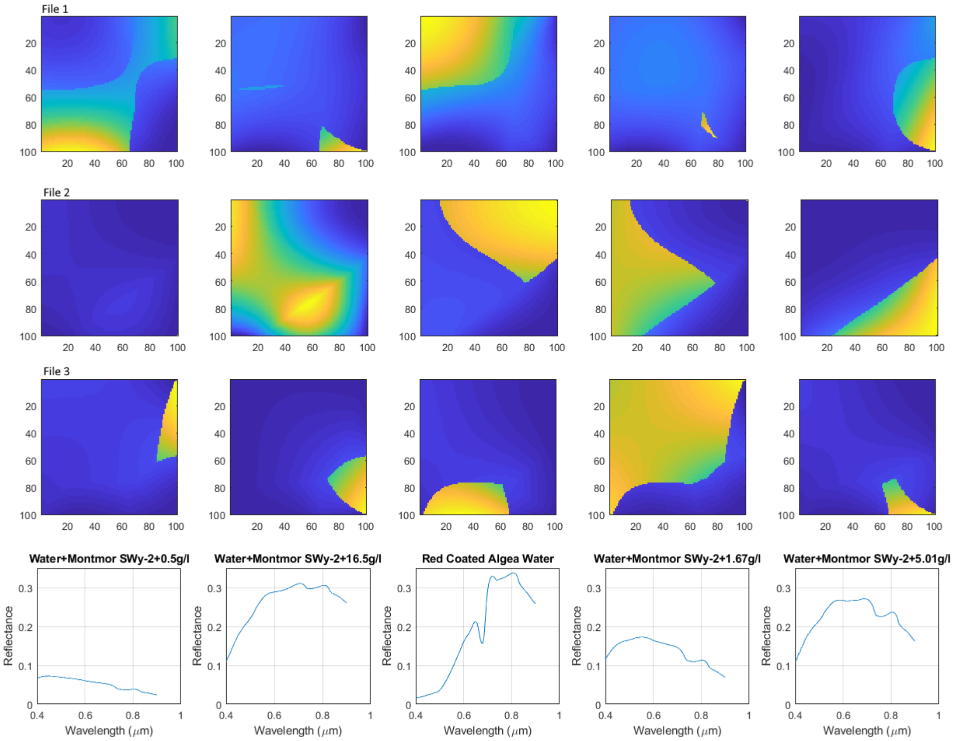


Figure C.1: The synthetic figures discussed in section 4.1.2

	File 1				File 2				File 3			
Name	mse	PSNR	mean SNR	CR	mse	PSNR	mean SNR	CR	mse	PSNR	mean SNR	CR
original PCA_5	8,39E-06	40,60	34,66	20,20	8,38E-06	41,34	35,62	20,20	8,37E-06	41,34	34,48	20,20
original MNFg_5	8,43E-11	90,58	84,64	20,20	4,42E-11	94,12	88,39	20,20	6,66E-12	102,34	95,47	20,20
original ICA_5	8,39E-06	40,60	34,66	20,20	8,38E-06	41,34	35,62	20,20	8,37E-06	41,34	34,48	20,20
original PCA_10	5,76E-06	42,23	36,29	10,10	4,39E-06	44,15	38,43	10,10	7,09E-06	42,07	35,20	10,10
original MNFg_10	6,17E-11	91,93	85,99	10,10	3,95E-11	94,60	88,88	10,10	2,15E-12	107,25	100,38	10,10
original ICA_10	5,76E-06	42,23	36,29	10,10	4,39E-06	44,15	38,43	10,10	7,09E-06	42,07	35,20	10,10
original PCA_20	4,68E-06	43,13	37,19	5,05	3,52E-06	45,10	39,38	5,05	5,75E-06	42,97	36,11	5,05
original MNFg_20	7,65E-12	101,00	95,06	5,05	3,48E-11	95,15	89,43	5,05	2,02E-12	107,52	100,65	5,05
original ICA_20	4,68E-06	43,13	37,19	5,05	3,52E-06	45,10	39,38	5,05	5,75E-06	42,97	36,11	5,05
original PCA_30	4,50E-06	43,30	37,37	3,37	2,91E-06	45,93	40,21	3,37	5,61E-06	43,09	36,22	3,37
original MNFg_30	7,62E-12	101,01	95,08	3,37	2,57E-11	96,48	90,76	3,37	2,01E-12	107,54	100,67	3,37
original ICA_30	4,50E-06	43,30	37,37	3,37	2,91E-06	45,93	40,21	3,37	5,61E-06	43,09	36,22	3,37
original PCA_40	4,11E-06	43,70	37,77	2,53	2,74E-06	46,19	40,47	2,53	5,32E-06	43,32	36,45	2,53
original MNFg_40	7,59E-12	101,03	95,10	2,53	1,96E-11	97,64	91,92	2,53	2,00E-12	107,56	100,69	2,53
original ICA_40	4,11E-06	43,70	37,77	2,53	2,74E-06	46,19	40,47	2,53	5,32E-06	43,32	36,45	2,53
original PCA_50	3,77E-06	44,07	38,13	2,02	1,87E-06	47,85	42,13	2,02	5,10E-06	43,50	36,63	2,02
original MNFg_50	7,58E-12	101,04	95,10	2,02	1,83E-11	97,96	92,23	2,02	2,00E-12	107,56	100,69	2,02
original ICA_50	3,77E-06	44,07	38,13	2,02	1,87E-06	47,85	42,13	2,02	5,10E-06	43,50	36,63	2,02
gauss PCA_5	9,02E-06	40,28	34,35	20,20	1,58E-05	38,57	32,85	20,20	8,34E-06	41,36	34,49	20,20
gauss MNFg_5	1,24E-06	48,89	42,96	20,20	1,35E-06	49,27	43,55	20,20	1,20E-06	49,78	42,92	20,20
gauss ICA_5	9,02E-06	40,28	34,35	20,20	1,58E-05	38,57	32,85	20,20	8,34E-06	41,36	34,49	20,20
gauss PCA_10	8,50E-06	40,54	34,61	10,10	1,55E-05	38,66	32,93	10,10	8,02E-06	41,53	34,66	10,10
gauss MNFg_10	1,23E-06	48,93	43,00	10,10	1,30E-06	49,42	43,69	10,10	1,17E-06	49,88	43,01	10,10
gauss ICA_10	8,50E-06	40,54	34,61	10,10	1,55E-05	38,66	32,93	10,10	8,02E-06	41,53	34,66	10,10
gauss PCA_20	7,43E-06	41,13	35,19	5,05	1,49E-05	38,83	33,11	5,05	7,25E-06	41,97	35,10	5,05
gauss MNFg_20	1,19E-06	49,10	43,16	5,05	1,25E-06	49,59	43,86	5,05	1,18E-06	49,84	42,97	5,05
gauss ICA_20	7,43E-06	41,13	35,19	5,05	1,49E-05	38,83	33,11	5,05	7,25E-06	41,97	35,10	5,05
gauss PCA_30	6,43E-06	41,76	35,82	3,37	1,41E-05	39,08	33,35	3,37	6,75E-06	42,28	35,41	3,37
gauss MNFg_30	1,16E-06	49,21	43,27	3,37	1,26E-06	49,56	43,83	3,37	9,80E-07	50,66	43,79	3,37
gauss ICA_30	6,43E-06	41,76	35,82	3,37	1,41E-05	39,08	33,35	3,37	6,75E-06	42,28	35,41	3,37
gauss PCA_40	5,74E-06	42,25	36,31	2,53	1,25E-05	39,62	33,89	2,53	5,83E-06	42,92	36,05	2,53
gauss MNFg_40	1,11E-06	49,37	43,43	2,53	1,23E-06	49,68	43,96	2,53	9,93E-07	50,60	43,73	2,53
gauss ICA_40	5,74E-06	42,25	36,31	2,53	1,25E-05	39,62	33,89	2,53	5,83E-06	42,92	36,05	2,53
gauss PCA_50	4,65E-06	43,16	37,22	2,02	9,40E-06	40,84	35,12	2,02	5,02E-06	43,56	36,69	2,02
gauss MNFg_50	9,95E-07	49,86	43,92	2,02	1,25E-06	49,60	43,87	2,02	9,59E-07	50,76	43,89	2,02
gauss ICA_50	4,65E-06	43,16	37,22	2,02	9,40E-06	40,84	35,12	2,02	5,02E-06	43,56	36,69	2,02
poisson PCA_5	8,39E-06	40,60	34,66	20,20	8,41E-06	41,32	35,60	20,20	8,41E-06	41,32	34,46	20,20
poisson MNFg_5	8,94E-07	50,32	44,39	20,20	8,91E-07	51,08	45,35	20,20	1,60E-06	48,53	41,66	20,20
poisson ICA_5	8,39E-06	40,60	34,66	20,20	8,41E-06	41,32	35,60	20,20	8,41E-06	41,32	34,46	20,20
poisson PCA_10	8,32E-06	40,63	34,70	10,10	8,20E-06	41,43	35,71	10,10	8,38E-06	41,34	34,47	10,10
poisson MNFg_10	8,64E-07	50,47	44,54	10,10	8,59E-07	51,23	45,51	10,10	1,40E-06	49,11	42,24	10,10

poisson ICA_10	8,32E-06	40,63	34,70	10,10	8,20E-06	41,43	35,71	10,10	8,38E-06	41,34	34,47	10,10
poisson PCA_20	8,03E-06	40,79	34,86	5,05	7,97E-06	41,56	35,83	5,05	8,26E-06	41,41	34,54	5,05
poisson MNFg_20	6,83E-07	51,49	45,55	5,05	7,33E-07	51,92	46,19	5,05	1,26E-06	49,58	42,71	5,05
poisson ICA_20	8,03E-06	40,79	34,86	5,05	7,97E-06	41,56	35,83	5,05	8,26E-06	41,41	34,54	5,05
poisson PCA_30	7,68E-06	40,98	35,04	3,37	7,86E-06	41,62	35,90	3,37	7,92E-06	41,59	34,72	3,37
poisson MNFg_30	5,90E-07	52,13	46,19	3,37	6,54E-07	52,42	46,69	3,37	1,12E-06	50,09	43,22	3,37
poisson ICA_30	7,68E-06	40,98	35,04	3,37	7,86E-06	41,62	35,90	3,37	7,92E-06	41,59	34,72	3,37
poisson PCA_40	7,28E-06	41,22	35,28	2,53	7,62E-06	41,75	36,03	2,53	7,62E-06	41,76	34,89	2,53
poisson MNFg_40	4,77E-07	53,05	47,12	2,53	5,83E-07	52,92	47,20	2,53	8,43E-07	51,31	44,45	2,53
poisson ICA_40	7,28E-06	41,22	35,28	2,53	7,62E-06	41,75	36,03	2,53	7,62E-06	41,76	34,89	2,53
poisson PCA_50	6,89E-06	41,45	35,52	2,02	7,36E-06	41,91	36,18	2,02	6,94E-06	42,16	35,29	2,02
poisson MNFg_50	4,15E-07	53,65	47,71	2,02	4,81E-07	53,75	48,03	2,02	6,84E-07	52,22	45,35	2,02
poisson ICA_50	6,89E-06	41,45	35,52	2,02	7,36E-06	41,91	36,18	2,02	6,94E-06	42,16	35,29	2,02
SnP PCA_5	4,45E-04	23,35	17,41	20,20	1,05E-03	20,38	14,66	20,20	5,62E-04	23,08	16,21	20,20
SnP MNFg_5	2,04E-04	26,75	20,81	20,20	1,99E-04	27,59	21,87	20,20	1,97E-04	27,63	20,76	20,20
SnP ICA_5	4,45E-04	23,35	17,41	20,20	1,05E-03	20,38	14,66	20,20	5,62E-04	23,08	16,21	20,20
SnP PCA_10	6,70E-04	21,57	15,64	10,10	1,14E-03	20,00	14,27	10,10	7,57E-04	21,78	14,91	10,10
SnP MNFg_10	3,90E-04	23,93	17,99	10,10	3,84E-04	24,73	19,00	10,10	3,79E-04	24,79	17,92	10,10
SnP ICA_10	6,70E-04	21,57	15,64	10,10	1,14E-03	20,00	14,27	10,10	7,57E-04	21,78	14,91	10,10
SnP PCA_20	1,10E-03	19,42	13,49	5,05	1,51E-03	18,78	13,06	5,05	1,14E-03	19,99	13,12	5,05
SnP MNFg_20	7,59E-04	21,03	15,10	5,05	7,55E-04	21,79	16,07	5,05	7,41E-04	21,87	15,00	5,05
SnP ICA_20	1,10E-03	19,42	13,49	5,05	1,51E-03	18,78	13,06	5,05	1,14E-03	19,99	13,12	5,05
SnP PCA_30	1,48E-03	18,13	12,20	3,37	1,85E-03	17,91	12,19	3,37	1,54E-03	18,68	11,82	3,37
SnP MNFg_30	1,13E-03	19,30	13,37	3,37	1,12E-03	20,07	14,34	3,37	1,11E-03	20,12	13,25	3,37
SnP ICA_30	1,48E-03	18,13	12,20	3,37	1,85E-03	17,91	12,19	3,37	1,54E-03	18,68	11,82	3,37
SnP PCA_40	1,87E-03	17,11	11,18	2,53	2,15E-03	17,25	11,52	2,53	1,90E-03	17,78	10,91	2,53
SnP MNFg_40	1,49E-03	18,09	12,15	2,53	1,49E-03	18,84	13,12	2,53	1,48E-03	18,88	12,01	2,53
SnP ICA_40	1,87E-03	17,11	11,18	2,53	2,15E-03	17,25	11,52	2,53	1,90E-03	17,78	10,91	2,53
SnP PCA_50	2,23E-03	16,35	10,42	2,02	2,45E-03	16,69	10,96	2,02	2,23E-03	17,09	10,22	2,02
SnP MNFg_50	1,87E-03	17,12	11,19	2,02	1,86E-03	17,89	12,16	2,02	1,84E-03	17,92	11,06	2,02
SnP ICA_50	2,23E-03	16,35	10,42	2,02	2,45E-03	16,69	10,96	2,02	2,23E-03	17,09	10,22	2,02
speck PCA_5	8,44E-05	30,57	24,64	20,20	8,68E-04	21,19	15,47	20,20	1,12E-05	40,07	33,21	20,20
speck MNFg_5	1,81E-06	47,25	41,31	20,20	1,82E-06	47,97	42,24	20,20	2,25E-06	47,04	40,18	20,20
speck ICA_5	8,44E-05	30,57	24,64	20,20	8,68E-04	21,19	15,47	20,20	1,12E-05	40,07	33,21	20,20
speck PCA_10	6,57E-05	31,66	25,72	10,10	8,66E-04	21,20	15,47	10,10	1,20E-05	39,77	32,90	10,10
speck MNFg_10	2,43E-06	45,98	40,04	10,10	2,57E-06	46,48	40,75	10,10	2,84E-06	46,03	39,17	10,10
speck ICA_10	6,57E-05	31,66	25,72	10,10	8,66E-04	21,20	15,47	10,10	1,20E-05	39,77	32,90	10,10
speck PCA_20	5,78E-05	32,22	26,28	5,05	8,58E-04	21,24	15,51	5,05	1,35E-05	39,27	32,40	5,05
speck MNFg_20	3,65E-06	44,21	38,28	5,05	4,16E-06	44,39	38,66	5,05	3,95E-06	44,60	37,73	5,05
speck ICA_20	5,78E-05	32,22	26,28	5,05	8,58E-04	21,24	15,51	5,05	1,35E-05	39,27	32,40	5,05
speck PCA_30	5,38E-05	32,52	26,59	3,37	8,27E-04	21,40	15,67	3,37	1,48E-05	38,88	32,01	3,37
speck MNFg_30	4,96E-06	42,88	36,95	3,37	5,83E-06	42,92	37,19	3,37	5,07E-06	43,52	36,65	3,37
speck ICA_30	5,38E-05	32,52	26,59	3,37	8,27E-04	21,40	15,67	3,37	1,48E-05	38,88	32,01	3,37

speck PCA_40	5,34E-05	32,56	26,62	2,53	7,97E-04	21,56	15,84	2,53	1,62E-05	38,48	31,61	2,53
speck MNFg_40	6,17E-06	41,94	36,00	2,53	7,38E-06	41,89	36,17	2,53	6,29E-06	42,59	35,72	2,53
speck ICA_40	5,34E-05	32,56	26,62	2,53	7,97E-04	21,56	15,84	2,53	1,62E-05	38,48	31,61	2,53
speck PCA_50	5,26E-05	32,63	26,69	2,02	7,34E-04	21,92	16,19	2,02	1,72E-05	38,21	31,34	2,02
speck MNFg_50	7,35E-06	41,17	35,24	2,02	9,11E-06	40,98	35,25	2,02	7,34E-06	41,91	35,05	2,02
speck ICA_50	5,26E-05	32,63	26,69	2,02	7,34E-04	21,92	16,19	2,02	1,72E-05	38,21	31,34	2,02

Appendix D

Performance of Target Detection Algorithms

On the subsequent pages the results from all the different scenes, DR representations, and Target detection algorithms are given. In the tables you can find the F_1-score, the MCC-score, the distance between the two binary classification scores, recall and visibility. The format in the table is the format of the name of the scene - endmember - target detection algorithms, the results for that given scenario follows.

Dimensionality Reduction and Target Detection Results

DR	F1	MCC	|F1-MCC|	Recall	Visibility
Pavia - Asphalt - CEM					
FULLDIM	0,214	0,234	0,020	0,485	0,039
PCA50	0,202	0,228	0,026	0,511	0,012
MNF50	0,124	0,116	0,008	0,371	0,015
ICA50	0,202	0,228	0,026	0,511	0,012
PCA20	0,196	0,224	0,029	0,507	0,010
MNF20	0,174	0,196	0,022	0,483	0,028
ICA20	0,196	0,224	0,029	0,507	0,010
Pavia - Asphalt - ACE					
FULLDIM	0,156	0,183	0,026	0,588	0,077
PCA50	0,164	0,191	0,027	0,532	0,108
MNF50	0,100	0,065	0,034	0,213	0,038
ICA50	0,164	0,191	0,027	0,532	0,108
PCA20	0,197	0,210	0,013	0,452	0,142
MNF20	0,173	0,167	0,007	0,381	0,127
ICA20	0,197	0,210	0,013	0,452	0,142
Pavia - Asphalt - OSP					
FULLDIM	0,065	0,129	0,064	0,995	0,074
PCA50	0,065	0,039	0,026	0,995	0,074
MNF50	0,073	0,066	0,007	0,917	0,001
ICA50	0,070	0,055	0,015	0,896	0,005
PCA20	0,065	0,037	0,027	0,995	0,071
MNF20	0,069	0,083	0,014	0,934	0,004
ICA20	0,066	0,032	0,035	0,890	0,007
Pavia - Asphalt - SAM					
FULLDIM	0,249	0,266	0,017	0,395	0,253
PCA50	0,250	0,266	0,016	0,396	0,253
MNF50	0,159	0,196	0,037	0,561	0,172
ICA50	0,086	0,062	0,023	0,482	0,018
PCA20	0,254	0,266	0,012	0,395	0,254
MNF20	0,208	0,248	0,040	0,598	0,246
ICA20	0,091	0,072	0,019	0,500	0,030
Pavia - Trees - CEM					
FULLDIM	0,372	0,388	0,017	0,500	0,175
PCA50	0,365	0,386	0,021	0,490	0,072
MNF50	0,359	0,378	0,019	0,484	0,103
ICA50	0,365	0,386	0,021	0,490	0,072
PCA20	0,364	0,381	0,017	0,493	0,058
MNF20	0,355	0,375	0,020	0,498	0,214
ICA20	0,364	0,381	0,017	0,493	0,058
Pavia - Trees - ACE					
FULLDIM	0,367	0,392	0,025	0,551	0,259
PCA50	0,367	0,386	0,019	0,493	0,302
MNF50	0,356	0,374	0,019	0,452	0,269

ICA50	0,367	0,386	0,019	0,493	0,302
PCA20	0,368	0,381	0,013	0,465	0,396
MNF20	0,352	0,364	0,012	0,454	0,319
ICA20	0,368	0,381	0,013	0,465	0,396
Pavia - Trees - OSP					
FULLDIM	0,255	0,302	0,046	0,476	0,263
PCA50	0,255	0,302	0,047	0,476	0,262
MNF50	0,341	0,367	0,025	0,406	0,264
ICA50	0,261	0,307	0,047	0,462	0,221
PCA20	0,259	0,305	0,046	0,442	0,229
MNF20	0,328	0,353	0,025	0,430	0,223
ICA20	0,256	0,304	0,048	0,463	0,199
Pavia - Trees - SAM					
FULLDIM	0,241	0,287	0,046	0,430	0,243
PCA50	0,241	0,287	0,046	0,430	0,243
MNF50	0,332	0,349	0,017	0,539	0,414
ICA50	0,296	0,318	0,022	0,491	0,287
PCA20	0,241	0,287	0,047	0,449	0,243
MNF20	0,307	0,329	0,023	0,436	0,404
ICA20	0,295	0,316	0,021	0,485	0,285
Pavia - Painted_metal_sheets - CEM					
FULLDIM	0,804	0,813	0,009	0,956	0,177
PCA50	0,802	0,812	0,010	0,941	0,191
MNF50	0,800	0,812	0,011	0,972	0,194
ICA50	0,802	0,812	0,010	0,941	0,191
PCA20	0,795	0,805	0,010	0,950	0,137
MNF20	0,801	0,811	0,009	0,955	0,207
ICA20	0,795	0,805	0,010	0,950	0,137
Pavia - Painted_metal_sheets - ACE					
FULLDIM	0,808	0,815	0,008	0,944	0,619
PCA50	0,805	0,814	0,009	0,952	0,712
MNF50	0,809	0,818	0,009	0,946	0,664
ICA50	0,805	0,814	0,009	0,952	0,712
PCA20	0,810	0,817	0,007	0,946	0,838
MNF20	0,808	0,817	0,008	0,935	0,818
ICA20	0,810	0,817	0,007	0,946	0,838
Pavia - Painted_metal_sheets - OSP					
FULLDIM	0,640	0,653	0,014	0,704	0,346
PCA50	0,661	0,653	0,008	0,726	0,191
MNF50	0,770	0,785	0,015	0,894	0,324
ICA50	0,762	0,775	0,013	0,934	0,212
PCA20	0,654	0,661	0,007	0,718	0,343
MNF20	0,773	0,711	0,062	0,955	0,329
ICA20	0,755	0,769	0,014	0,932	0,254
Pavia - Painted_metal_sheets - SAM					
FULLDIM	0,807	0,814	0,008	0,946	0,431
PCA50	0,807	0,814	0,008	0,946	0,431
MNF50	0,816	0,824	0,009	0,965	0,767
ICA50	0,811	0,814	0,003	0,907	0,615
PCA20	0,807	0,814	0,008	0,946	0,431

MNF20	0,817	0,825	0,008	0,962	0,760
ICA20	0,810	0,814	0,004	0,908	0,614
Salinas - Lettuce_romaine_4wk - CEM					
FULLDIM	0,842	0,840	0,001	0,857	0,154
PCA50	0,816	0,814	0,002	0,800	0,099
MNF50	0,842	0,840	0,002	0,845	0,148
ICA50	0,816	0,814	0,002	0,800	0,099
PCA20	0,680	0,681	0,001	0,761	0,027
MNF20	0,841	0,839	0,001	0,854	0,172
ICA20	0,680	0,681	0,001	0,761	0,027
Salinas - Lettuce_romaine_4wk - ACE					
FULLDIM	0,849	0,848	0,001	0,840	0,568
PCA50	0,879	0,878	0,001	0,874	0,638
MNF50	0,887	0,885	0,001	0,889	0,636
ICA50	0,879	0,878	0,001	0,874	0,638
PCA20	0,818	0,818	0,000	0,772	0,698
MNF20	0,889	0,888	0,001	0,884	0,826
ICA20	0,818	0,818	0,000	0,772	0,698
Salinas - Lettuce_romaine_4wk - OSP					
FULLDIM	0,058	0,181	0,123	0,611	0,220
PCA50	0,036	0,126	0,090	0,914	0,123
MNF50	0,783	0,676	0,107	0,789	0,306
ICA50	0,033	0,088	0,055	0,973	0,082
PCA20	0,030	0,022	0,007	0,893	0,069
MNF20	0,435	0,348	0,087	0,493	0,262
ICA20	0,040	0,054	0,014	0,914	0,132
Salinas - Lettuce_romaine_4wk - SAM					
FULLDIM	0,658	0,671	0,013	0,542	0,072
PCA50	0,657	0,670	0,013	0,542	0,072
MNF50	0,871	0,870	0,001	0,889	0,193
ICA50	0,738	0,738	0,000	0,680	0,136
PCA20	0,654	0,665	0,012	0,543	0,072
MNF20	0,872	0,871	0,001	0,893	0,194
ICA20	0,717	0,718	0,001	0,657	0,144
Salinas - Lettuce_romaine_5wk - CEM					
FULLDIM	0,571	0,563	0,007	0,598	0,071
PCA50	0,369	0,429	0,060	0,776	0,088
MNF50	0,538	0,538	0,000	0,657	0,088
ICA50	0,369	0,429	0,060	0,776	0,088
PCA20	0,333	0,381	0,048	0,677	0,026
MNF20	0,361	0,436	0,074	0,895	0,151
ICA20	0,333	0,381	0,048	0,677	0,026
Salinas - Lettuce_romaine_5wk - ACE					
FULLDIM	0,563	0,555	0,008	0,572	0,272
PCA50	0,411	0,432	0,021	0,647	0,287
MNF50	0,627	0,621	0,006	0,646	0,350
ICA50	0,411	0,432	0,021	0,647	0,287
PCA20	0,525	0,524	0,001	0,565	0,418
MNF20	0,708	0,703	0,005	0,714	0,505
ICA20	0,525	0,524	0,001	0,565	0,418

Salinas - Lettuce_romaine_5wk - OSP					
FULLDIM	0,056	0,140	0,084	0,983	0,008
PCA50	0,093	0,102	0,010	0,663	0,105
MNF50	0,088	0,184	0,096	0,950	0,175
ICA50	0,075	0,167	0,092	0,967	0,185
PCA20	0,075	0,095	0,021	0,963	0,151
MNF20	0,078	0,140	0,062	0,917	0,094
ICA20	0,088	0,154	0,066	0,752	0,163
Salinas - Lettuce_romaine_5wk - SAM					
FULLDIM	0,684	0,679	0,005	0,728	0,065
PCA50	0,685	0,680	0,005	0,728	0,065
MNF50	0,781	0,778	0,004	0,804	0,143
ICA50	0,656	0,656	0,000	0,766	0,118
PCA20	0,684	0,680	0,005	0,726	0,065
MNF20	0,811	0,808	0,003	0,810	0,152
ICA20	0,655	0,654	0,001	0,754	0,125
Salinas - Lettuce_romaine_7wk - CEM					
FULLDIM	0,734	0,732	0,002	0,779	0,332
PCA50	0,685	0,682	0,002	0,723	0,229
MNF50	0,729	0,728	0,001	0,780	0,348
ICA50	0,685	0,682	0,002	0,723	0,229
PCA20	0,667	0,664	0,003	0,693	0,170
MNF20	0,685	0,682	0,003	0,700	0,320
ICA20	0,667	0,664	0,003	0,693	0,170
Salinas - Lettuce_romaine_7wk - ACE					
FULLDIM	0,741	0,739	0,001	0,790	0,519
PCA50	0,661	0,662	0,001	0,747	0,604
MNF50	0,724	0,725	0,000	0,785	0,619
ICA50	0,661	0,662	0,001	0,747	0,604
PCA20	0,640	0,640	0,000	0,717	0,758
MNF20	0,673	0,672	0,001	0,734	0,803
ICA20	0,640	0,640	0,000	0,717	0,758
Salinas - Lettuce_romaine_7wk - OSP					
FULLDIM	0,030	0,338	0,308	0,875	0,029
PCA50	0,027	0,061	0,035	0,961	0,005
MNF50	0,579	0,667	0,088	0,651	0,265
ICA50	0,117	0,177	0,059	0,508	0,186
PCA20	0,021	0,399	0,379	0,983	0,068
MNF20	0,608	0,507	0,101	0,681	0,252
ICA20	0,082	0,020	0,062	0,682	0,062
Salinas - Lettuce_romaine_7wk - SAM					
FULLDIM	0,669	0,669	0,000	0,744	0,099
PCA50	0,669	0,669	0,000	0,744	0,099
MNF50	0,714	0,711	0,003	0,731	0,216
ICA50	0,685	0,686	0,001	0,774	0,204
PCA20	0,669	0,669	0,000	0,744	0,099
MNF20	0,719	0,717	0,002	0,753	0,221
ICA20	0,683	0,685	0,002	0,777	0,208
Hopavågen - Green_algae - CEM					
FULLDIM	0,782	0,767	0,015	0,836	0,197

PCA50	0,780	0,764	0,016	0,804	0,193
MNF50	0,779	0,763	0,016	0,791	0,198
ICA50	0,780	0,764	0,016	0,804	0,193
PCA20	0,780	0,764	0,016	0,811	0,194
MNF20	0,772	0,756	0,017	0,777	0,198
ICA20	0,780	0,764	0,016	0,811	0,194
Hopavågen - Green_algae - ACE					
FULLDIM	0,762	0,745	0,017	0,755	0,182
PCA50	0,768	0,753	0,015	0,739	0,193
MNF50	0,759	0,744	0,015	0,736	0,203
ICA50	0,768	0,753	0,015	0,739	0,193
PCA20	0,744	0,726	0,019	0,744	0,260
MNF20	0,736	0,723	0,013	0,669	0,257
ICA20	0,744	0,726	0,019	0,744	0,260
Hopavågen - Green_algae - OSP					
FULLDIM	0,380	0,452	0,072	0,675	0,189
PCA50	0,461	0,375	0,087	0,720	0,196
MNF50	0,579	0,547	0,031	0,596	0,191
ICA50	0,529	0,514	0,015	0,545	0,161
PCA20	0,453	0,440	0,013	0,649	0,205
MNF20	0,584	0,557	0,027	0,562	0,188
ICA20	0,534	0,516	0,018	0,552	0,165
Hopavågen - Green_algae - SAM					
FULLDIM	0,387	0,341	0,046	0,402	0,081
PCA50	0,384	0,339	0,045	0,407	0,081
MNF50	0,578	0,547	0,031	0,577	0,251
ICA50	0,442	0,432	0,010	0,380	0,277
PCA20	0,377	0,333	0,044	0,395	0,079
MNF20	0,519	0,483	0,036	0,567	0,236
ICA20	0,442	0,429	0,013	0,379	0,275
Hopavågen - Coralline_algae - CEM					
FULLDIM	0,836	0,794	0,043	0,841	0,253
PCA50	0,833	0,789	0,044	0,842	0,266
MNF50	0,835	0,793	0,042	0,828	0,256
ICA50	0,833	0,789	0,044	0,842	0,266
PCA20	0,823	0,776	0,047	0,867	0,271
MNF20	0,834	0,791	0,043	0,854	0,257
ICA20	0,823	0,776	0,047	0,867	0,271
Hopavågen - Coralline_algae - ACE					
FULLDIM	0,561	0,504	0,057	0,485	0,137
PCA50	0,567	0,516	0,051	0,478	0,137
MNF50	0,550	0,495	0,055	0,508	0,139
ICA50	0,567	0,516	0,051	0,478	0,137
PCA20	0,553	0,498	0,054	0,504	0,127
MNF20	0,515	0,432	0,082	0,478	0,137
ICA20	0,553	0,498	0,054	0,504	0,127
Hopavågen - Coralline_algae - OSP					
FULLDIM	0,725	0,651	0,074	0,806	0,207
PCA50	0,724	0,650	0,074	0,805	0,207
MNF50	0,722	0,647	0,075	0,774	0,234

ICA50	0,749	0,682	0,067	0,774	0,217
PCA20	0,721	0,646	0,075	0,825	0,209
MNF20	0,716	0,639	0,077	0,773	0,218
ICA20	0,748	0,682	0,067	0,769	0,217
Hopavågen - Coralline_algae - SAM					
FULLDIM	0,662	0,574	0,089	0,694	0,111
PCA50	0,661	0,571	0,090	0,686	0,110
MNF50	0,700	0,622	0,078	0,710	0,247
ICA50	0,671	0,591	0,080	0,658	0,381
PCA20	0,655	0,565	0,091	0,699	0,108
MNF20	0,648	0,558	0,090	0,647	0,247
ICA20	0,668	0,585	0,083	0,714	0,377
Hopavågen - Fucus_serratus - CEM					
FULLDIM	0,906	0,878	0,028	0,948	0,206
PCA50	0,905	0,876	0,029	0,942	0,205
MNF50	0,904	0,876	0,029	0,946	0,206
ICA50	0,905	0,876	0,029	0,942	0,205
PCA20	0,896	0,864	0,032	0,925	0,195
MNF20	0,905	0,877	0,029	0,936	0,203
ICA20	0,896	0,864	0,032	0,925	0,195
Hopavågen - Fucus_serratus - ACE					
FULLDIM	0,905	0,877	0,028	0,902	0,268
PCA50	0,900	0,870	0,030	0,890	0,304
MNF50	0,905	0,877	0,028	0,898	0,291
ICA50	0,900	0,870	0,030	0,890	0,304
PCA20	0,893	0,862	0,032	0,896	0,339
MNF20	0,913	0,887	0,026	0,903	0,341
ICA20	0,893	0,862	0,032	0,896	0,339
Hopavågen - Fucus_serratus - OSP					
FULLDIM	0,510	0,339	0,170	0,815	0,058
PCA50	0,510	0,339	0,170	0,815	0,058
MNF50	0,741	0,660	0,081	0,783	0,176
ICA50	0,769	0,698	0,071	0,873	0,184
PCA20	0,704	0,334	0,369	0,836	0,110
MNF20	0,752	0,674	0,078	0,787	0,179
ICA20	0,767	0,695	0,072	0,871	0,184
Hopavågen - Fucus_serratus - SAM					
FULLDIM	0,629	0,557	0,073	0,566	0,081
PCA50	0,636	0,564	0,071	0,568	0,081
MNF50	0,929	0,908	0,021	0,943	0,514
ICA50	0,736	0,693	0,043	0,622	0,333
PCA20	0,653	0,585	0,068	0,591	0,084
MNF20	0,937	0,918	0,019	0,950	0,557
ICA20	0,749	0,703	0,046	0,655	0,343