

**IMPROVING NONLINEAR DIMENSIONALITY REDUCTION  
ALGORITHMS FOR HYPERSPECTRAL DATA**

by

Nicolas Rey-Villamizar

A thesis submitted in partial fulfillment of the requirements for the degree of

**MASTER OF SCIENCE**

in

**ELECTRICAL ENGINEERING**

University of Puerto Rico

Mayagüez Campus

2010

Approved by:

---

Shawn D. Hunt, Ph.D  
Member, Graduate Committee

---

Date

---

Vidya Manian, Ph.D  
Member, Graduate Committee

---

Date

---

Miguel Vélez-Reyes, Ph.D  
President, Graduate Committee

---

Date

---

Reyes M. Ortiz-Albino, Ph.D  
Representative of Graduate Studies

---

Date

---

Erick E. Aponte, Ph.D  
Chairperson of the Department

---

Date

Abstract of Dissertation Presented to the Graduate School  
of the University of Puerto Rico in Partial Fulfillment of the  
Requirements for the Degree of Master of Science

**IMPROVING NONLINEAR DIMENSIONALITY REDUCTION  
ALGORITHMS FOR HYPERSPECTRAL DATA**

By

Nicolas Rey-Villamizar

December 2010

Chair: Miguel Vélez-Reyes

Major Department: Electrical and Computer Engineering

Dimensionality reduction is a key step in hyperspectral image processing due to the large amount of data. Linear and nonlinear approaches have been proposed. The most important parameters on the majority of nonlinear dimensionality reduction algorithms (NLDR) are the number of neighbors used to construct the starting graph, and the number of dimensions of the low dimensional space where the data is embedded. This research work focuses on the influence of the first parameter on the DR. Newly proposed methods for constructing the weighted graph are used:  $k$ -VC,  $k$ -EC and  $k$ -MST which are alternatives to the classical approaches to  $k$  nearest neighbors ( $k$ -NN) and epsilon neighborhood ( $e$ -NN). This methods have the advantage that connectedness of the graph is guarantee and also update of the graph in case new data is available is computationally inexpensive compare to recalculating all the graph again as is needed on the classical algorithms. Also, a newly proposed

neighborhood selection technique called cam-weighted neighborhood is used in combination with the NLDR algorithms. Finally, a method to improve the geodesic distance estimation is explored. Experiments are carry out over hyperspectral datasets, where classification is used as the validation criteria.

Resumen de Disertación Presentado a Escuela Graduada  
de la Universidad de Puerto Rico como requisito parcial de los  
Requerimientos para el grado de Maestro en Ciencias

**MEJORAMIENTO DE ALGORITMOS NOLINEALES DE  
REDUCCIOÓN DE DIMENSIONALIDAD EN DATOS  
HYPERESPECTRALES**

Por

Nicolas Rey-Villamizar

Diciembre 2010

Consejero: Miguel Vélez-Reyes

Departamento: Ingeniería Eléctrica y Computadoras

Reducción de dimensionalidad es una etapa critica en el procesamiento de imágenes hyperspectrales debido a la gran cantidad de datos. Métodos lineales y no lineales han sido propuestos. El parámetro mas importante en la mayoría de los métodos no lineales de reducción de dimensionalidad (NLDR) es el número de vecinos usados para construir el grafo inicial, y el número de dimensiones del espacio donde la data es embebida. Este trabajo de investigación se centra en la influencia del primer parámetro en DR. Métodos recientemente propuestos para construir el grafo ponderado son usados:  $k$ -VC,  $k$ -EC y  $k$ -MST los cuales son alternativas a los métodos clásicos de  $k$  vecinos mas cercanos y vecinos dentro de una vecindad. Estos métodos tienen la ventaja de que la conectividad del grafo esta garantizada y además en caso de necesitar actualizar el grafo debido a la presencia de nueva información, el proceso es computacionalmente menor comparado a recalcular todo el grafo nuevamente como es requerido en los algoritmos clásicos. Además, un método de selección de vecinos propuesto recientemente llamado vecindad “cam-weighted”

es usado en combinación con los algoritmos de NLDR. Finalmente, un método para mejorar la estimación de la distancia geodésica es explorado. Se hicieron experimentos con datos hyperspectrales, donde la clasificación se uso como un criterio de validación.

To the great thinkers which created the fundamental ideas over which this thesis  
is build on ...

# Acknowledgments

I want to begin by expressing my most sincere gratitude and acknowledgement to my advisor, Dr. Miguel Vélez-Reyes for his patient guidance, encouragement and excellent advice throughout the course of study. Also, I am grateful to my thesis committee members for their review and helpful criticism. Thanks to all the Professors during the two years of study: Dr. Miguel Vélez-Reyes, Dr. Domingo Rodriguez, Dr. Paul Castillo, Dr. Shawn D. Hunt, Dr. Pedro Vásquez, and Dr. Rodolfo Romanach. Each one of them has left a seed that is growing in my soul. I also express my gratitude to the Electrical Engineering Department and CenSSIS staff for giving me the opportunity to carry out graduate studies at the UPRM campus. Special thanks to Sandra Montalvo and Maribel Feliciano for making me feel comfortable during the course of my studies.

Last but not least, I take this opportunity to express my profound gratitude to my family and my friends for their support during this work. Also, to all the persons who have been involved directly or indirectly during my stay at Puerto Rico.

This work was sponsored primarily by DoD, partial support was received from CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the National Science Foundation (Award Number ECE-9986821).

# Table of Contents

Abstract English . . . . .	ii
Abstract Spanish . . . . .	iv
Acknowledgments . . . . .	vii
Table of Contents . . . . .	viii
List of Tables . . . . .	x
List of Figures . . . . .	xi
List of Abbreviations . . . . .	xiv
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Justification . . . . .	1
1.2 Objectives . . . . .	2
1.3 Contribution of the Work . . . . .	3
1.4 Outline . . . . .	3
<b>2 Background . . . . .</b>	<b>5</b>
2.1 Hyperspectral Data . . . . .	5
2.1.1 Dimensionality Reduction . . . . .	6
2.1.2 Linear Dimensionality Reduction . . . . .	6
2.1.3 Nonlinear Dimensionality Reduction . . . . .	7
2.2 NLDR Algorithms . . . . .	11
2.2.1 Locally Linear Embedding . . . . .	11
2.2.2 Isometric Feature Embedding . . . . .	11
2.2.3 Kernel PCA . . . . .	14
2.2.4 Maximum Variance Unfolding . . . . .	14
2.2.5 NLDR in Hyperspectral Data . . . . .	16

2.3	Summary . . . . .	17
<b>3</b>	<b>Neighborhood Selection and Graph Construction . . . . .</b>	<b>18</b>
3.1	Spatial Information HSI . . . . .	18
3.1.1	Classical Approach for Neighborhood Selection . . . . .	19
3.1.2	Adaptive Scheme to Select Neighbors . . . . .	19
3.1.3	Cam-distance for Neighborhood Selection . . . . .	20
3.1.4	Graph Construction for Neighborhood Selection . . . . .	23
3.1.5	Short-Circuits Comparison . . . . .	26
3.1.6	Improving Geodesic Distance Estimation . . . . .	28
3.2	Performance Evaluation . . . . .	31
3.2.1	Mahalanobis Distance Classifier . . . . .	31
3.2.2	Maximum Likelihood Classifier . . . . .	31
3.3	Summary . . . . .	32
<b>4</b>	<b>Algorithms Validation and Experimental Results Using Hyper- spectral Data . . . . .</b>	<b>34</b>
4.1	Methodology . . . . .	34
4.2	Experiment 1 - Fake Leaves . . . . .	36
4.2.1	Imagery . . . . .	36
4.2.2	Results and Discussion . . . . .	39
4.3	Experiment 2 - Indian Pines . . . . .	48
4.3.1	Imagery . . . . .	48
4.3.2	Results and Discussion . . . . .	49
<b>5</b>	<b>Conclusions and Future Work . . . . .</b>	<b>59</b>
5.1	Conclusions . . . . .	59
5.2	Future Work . . . . .	60
	<b>References . . . . .</b>	<b>62</b>
	<b>Biographical Sketch . . . . .</b>	<b>68</b>

# List of Tables

2-1 Algorithms for Nonlinear Dimensionality Reduction . . . . . 9

# List of Figures

2-1	AVIRIS Hyperspectral Data Cube Over Moffett Field, CA. . . . .	5
2-2	Global Framework of Dimensionality Reduction Methods. . . . .	8
2-3	Swiss Roll. . . . .	10
2-4	Short-Circuit Represented on a Swiss Roll Manifold. . . . .	11
2-5	Illustration of the Geodesic Distance Concept [1]. . . . .	13
3-1	Cam Weighted Distance. . . . .	22
3-2	$k$ -NN. . . . .	26
3-3	$k$ -MST. . . . .	27
3-4	$k$ -EC. . . . .	27
3-5	$k$ -VC. . . . .	27
3-6	2-Chain Patch in 3D Space. . . . .	29
4-1	Process of NLDR. . . . .	35
4-2	Fake Leaves Data Set (RGB corresponds to bands 90, 68, and 29). . .	37
4-3	SOC-700 Hyperspectral Camera. . . . .	38
4-4	Testing and Training Data of the Fake Leaves Data Set. . . . .	38
4-5	Overall Accuracy Using NLE for Graph Construction, LLE for DR and Mahalanobis Distance for Classification. . . . .	39
4-6	Overall Accuracy Using NLE for Graph Construction, ISOMAP for DR and Mahalanobis Distance for Classification. . . . .	40

4-7 Overall Accuracy Using Cam-distance for Graph Construction, LLE for DR and Maximum Likelihood for Classification. . . . .	41
4-8 Overall Accuracy Using Cam-distance for Graph Construction, ISOMAP for DR and Maximum Likelihood for Classification. . . . .	42
4-9 Overall Accuracy Using Cam-distance for Graph Construction, MVU for DR and Maximum Likelihood for Classification. . . . .	42
4-10 Overall Accuracy Using the $k$ -NN Algorithm for Graph Construction, and DR Using the ISOMAP Algorithm. . . . .	45
4-11 Overall Accuracy Using the $k$ -VC Algorithm for Graph Construction, and DR Using the LLE Algorithm. . . . .	45
4-12 Overall Accuracy Using the $k$ -EC Algorithm for Graph Construction, and DR Using the MVU Algorithm. . . . .	46
4-13 Overall Accuracy Using all the Described Graph Construction Algo- rithms. . . . .	47
4-14 Overall Accuracy Improvement Using an Algorithm to Improve the Geodesic Distance Estimation. . . . .	48
4-15 Right, Indian Pines (RGB composite bands 47, 24 and 14), Left, Ground Truth. . . . .	49
4-16 Section of the Indian Pines Image (a) RGB Image (Bands 47, 24 and 14), (b) Ground Truth. . . . .	49
4-17 Overall Accuracy Using NLE for Graph Construction, LLE for Di- mensionality Reduction and Mahalanobis Distance for Classification. . . . .	50
4-18 Overall Accuracy Using NLE for Graph Construction, ISOMAP for Dimensionality Reduction and Mahalanobis Distance for Classifi- cation. . . . .	50
4-19 Overall Accuracy Using Cam-distance for Graph Construction, LLE for DR and Maximum Likelihood for Classification. . . . .	52
4-20 Overall Accuracy Using Cam-distance for Graph Construction, ISOMAP for DR and Maximum Likelihood for Classification. . . . .	52
4-21 Overall Accuracy Using Cam-distance for Graph Construction, MVU for DR and Maximum Likelihood for Classification. . . . .	53

4-22 Overall Accuracy Using the $k$ -NN Algorithm for Graph Construction, and DR Using the LLE Algorithm. . . . .	55
4-23 Overall Accuracy Using the $k$ -VC Algorithm for Graph Construction, and DR Using the LLE Algorithm. . . . .	55
4-24 Overall Accuracy Using the $k$ -NN Algorithm for Graph Construction, and DR Using the MVU Algorithm. . . . .	56
4-25 Overall Accuracy Using all the Described Graph Construction Algo- rithms. . . . .	57
4-26 Overall Accuracy Improvement Using an Algorithm to Improve the Geodesic Distance Estimation. . . . .	58

## List of Abbreviations

DR	Dimensionality reduction.
NLDR	Nonlinear dimensionality reduction.
ISOMAP	Isometric feature embedding.
LLE	Locally linear embedding.
MVU	Maximum variance unfolding.
PCA	Principal Component Algorithm.
HSI	Hyperspectral Imaging.
LLA	Locally linear assumption.

# Chapter 1

---

## Introduction

### 1.1 Justification

Hyperspectral imaging (HSI) is a remote sensing technology with great potential for extracting more accurate information, due to the discriminative capabilities of the high spectral resolution. HSI provides hundreds of images corresponding to narrow and adjacent spectral bands. Extracting the most relevant information from this large amount of data is a challenging task that requires considerable computing resources. As the number of bands -dimensions- increases, more resources are needed. With the improvement of the sensing technology, new sensors tend to include higher number of bands. Hence more computing resources is needed to process this data. Also, hyperspectral video technology is being developed and will be available soon. Although the collected data lies on a high dimensional space -hundreds of bands- it is thought that the data is intrinsically low dimensional.

Non-linear dimensionality reduction methods have been successfully used to reduce the required computing resources of processing hyperspectral images. However, a first step in this process, called neighborhood selection, accounts for most of the processing time. The most representative of these methods are  $k$ -nearest neighbors and  $\epsilon$  threshold. These methods have the disadvantage that incremental versions for updating the neighborhood are not available. That means, in case of a new band

is included or a stream of hyperspectral images is used, the neighborhood selection process needs to be repeated from scratch each time. New methods of neighborhood selection for which incremental versions have been proposed will be tested. The test will show how reliable are the results of NLDR algorithms, using this methods compared to the results obtained using the standard methods of neighborhood selection.

## 1.2 Objectives

The main objective of this research was to:

- Study and implement recently proposed neighborhood selection methods suitable to replace the widely used techniques of  $k$ -nearest neighborhood and  $\epsilon$ -neighborhood on the literature of NLDR for hyperspectral images.

Other objectives of this research were:

- Combine spatial and spectral information in hyperspectral images for the graph construction in NLDR.
- Implement and evaluate representative NLDR algorithms and select the most suitable among those studied to be used with hyperspectral images.
- Study and implement the cam weighted distance algorithm to improve the nearest neighborhood selection for nonlinear dimensionality reduction.
- Study a method to evaluate the geodesic distance estimation and their respective repercussion on NLDR for hyperspectral imagery.
- Evaluate the performance of the different NLDR methods and neighborhood selection experimentally with different hyperspectral images.

### 1.3 Contribution of the Work

The main contribution of this research is the study of recently proposed methodologies for graph construction and their applicability on NLDR of hyperspectral images. This methodologies along with a new neighborhood selection algorithm are used to improve the reliability of the nonlinear projection of the hyperspectral data into a low dimensional space. Three common studied problems of hyperspectral imagery are addressed: noise on the data that creates a not smooth distribution of the data structure, unconnected graphs that prevent some data to be “*reliable*” projected on the low dimensional space, and reducing the sensibility of the NLDR algorithms to the parameter  $k$  of the graph construction algorithm.

### 1.4 Outline

The first chapter introduces the basic concept NLDR. The objectives and main contributions of this thesis have also been discussed. Below, a brief summary of the following chapters is presented.

#### Chapter 2

This chapter gives an overview of dimensionality reduction and the most representative algorithms. Specifically, a detailed explanation of the most representative NLDR algorithms is presented. Also, previous publications that use NLDR on hyperspectral data are discussed.

#### Chapter 3

The first step of NLDR algorithms is addressed; how to construct the graph on the high dimensional space. Spatial information is included via the coherent distance

measure. A newly proposed nearest neighbor selection algorithm is presented, which takes into account the local distribution of the data on the neighborhood selection algorithm. A method to improve geodesic distance estimation, and an adaptive method NLE to select the number of nearest neighbor are presented. And, weighted graph construction algorithms that warranty connectivity of the neighborhood graph are explained.

#### **Chapter 4**

Two hyperspectral datasets are used to test the proposed idea for improving NLDR. Classification is used as a quantitative measure of the reliability of the dimensionality reduction. Mahalanobis distance and maximum likelihood classifiers are used on the low dimensional space.

#### **Chapter 5**

Finally, conclusions on the obtained results are presented. This chapter also includes suggestions for future work.

# Chapter 2

---

## Background

### 2.1 Hyperspectral Data

We think of a hyperspectral image as gray scale images staked  $D$  times one on top of the other. In this case,  $D$  represents the spectral dimension. The difference between each image is the spectrum of light from which the image is taken. For the purpose of this research, the number of dimensions refers to the number  $D$  of different images at different wavelength. Hyperspectral data is usually illustrated as the data cube shown in Figure 2–1.

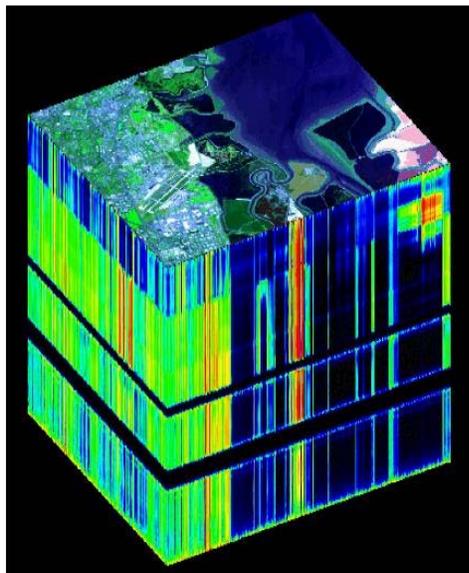


Figure 2–1: AVIRIS Hyperspectral Data Cube Over Moffett Field, CA.

Each pixel in the hyperspectral image has a particular spectral signature. One of the main goals of hyperspectral imaging is to classify the materials (pixels) by their spectral signature. The goal of DR over hyperspectral data is to reduce the number of spectral bands from  $D$  to  $d \ll D$  preserving the discriminative capabilities of the spectral signature.

### 2.1.1 Dimensionality Reduction

For classification purposes, the low sampling of the large dimensional space causes the well known “*curse of dimensionality*” problem. A way of dealing with this is to reduce the dimension of the data to a lower dimensional space thereby increasing the density of the sampling, hoping that the intrinsic structure of the data is preserved. This dimensionality reduction is achieved by creating a mapping between the high dimensional and the lower dimensional spaces.

Mathematically the problem of dimensionality reduction can be formulated as follows. Given  $n$  data samples in  $\mathbb{R}^D$ ,  $x_i$  for  $i = 1, 2, \dots, n$  the matrix  $\mathbf{X}_{D \times n}$  is formed. The goal of DR algorithms is to create a mapping of this data,  $\mathbf{X}$ , to a new dataset,  $\mathbf{Y}$ , where  $\mathbf{Y}$  is a  $d \times n$  matrix with  $d \ll D$ . This mapping can be classified in many forms, but broadly it can be classified as been of two forms: linear and nonlinear. Nonlinear mapping is the main focus of this research work.

### 2.1.2 Linear Dimensionality Reduction

Traditional methods for linear dimensionality reduction are Principal Component Analysis (PCA), Multidimensional Scaling (MDS) and Linear Discriminant Analysis (LDA). These traditional methods tend to perform poorly on nonlinear curved structures like the Swiss Roll. They perform well on flat euclidean structures. Different approaches have been proposed to create extensions of this to

non-linear methods, some examples are Probabilistic Principal Component Analysis (PPCA) [2] and self-organizing maps (SOMs) [3]. Also, kernel extensions of these algorithms have been developed, one example is kernel PCA (KPCA) [4] and kernel LDA (KLDA). Recently some new nonlinear methods have been proposed based on the idea of manifold as some studies in human cognition, suggests that the brain constructs *manifolds* when perceiving object which varies *smoothly* [5].

### 2.1.3 Nonlinear Dimensionality Reduction

Manifold learning has been widely spread since ISOMAP [1] and LLE [6] algorithms were proposed. These methods have been extensively used thanks to their geometric intuition, computational feasibility and their low dependence on parameters. Also, because they have shown to outperform their linear counterparts on nonlinear artificial datasets (Swiss Roll, Toroidal helix, Gaussian distribution, punched sphere, etc...). Figure 2–2 shows a taxonomy for dimensionality reduction methods [7]. The main focus of this work is to study some nonlinear methods based on manifold learning and their variations and applications to hyperspectral images. Table 2–1 shows a time-line of the most relevant manifold based dimensionality reduction methods that have been proposed in the literature.

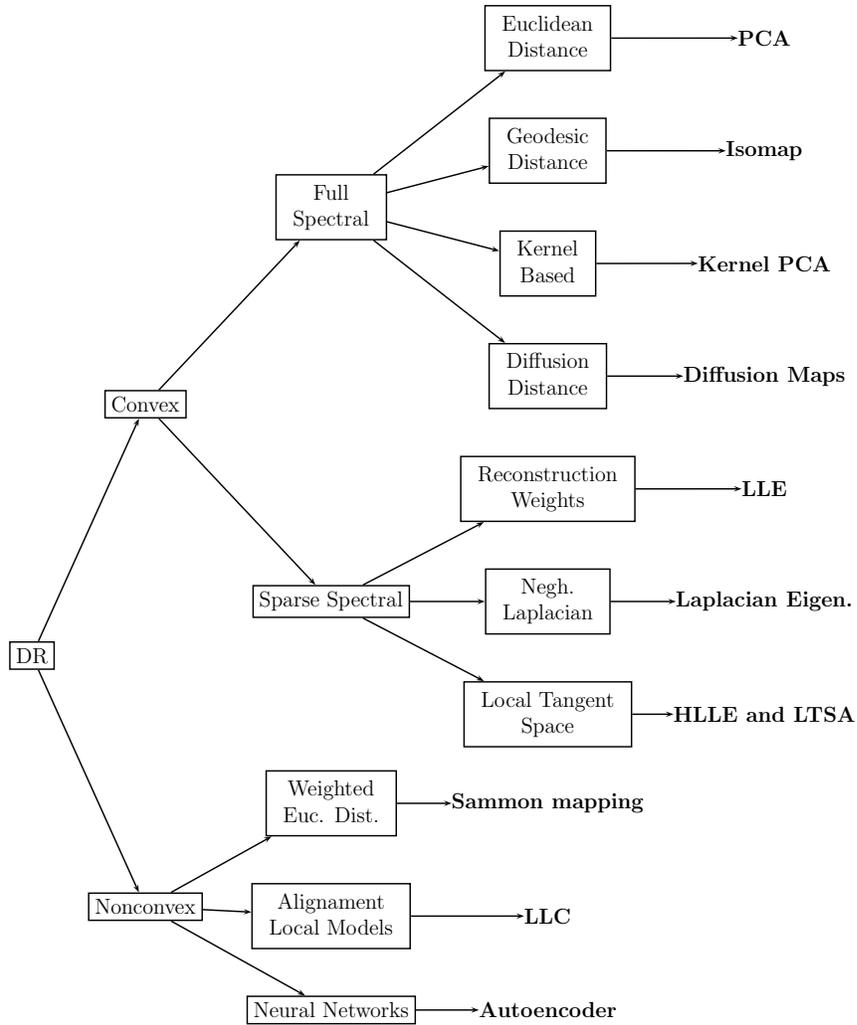


Figure 2–2: Global Framework of Dimensionality Reduction Methods.

Extensive work has been done to compare the performance of these algorithms [14] - [15] empirically. Also, theoretical relations between methods have been studied [16]. Different approaches to create a global framework for manifold learning methods have been studied. Relations between algorithms coming from different geometric intuitive ideas have been build. However a global framework has not been built due to the completely different ideas that have been the base for different algorithms. Also, the increased number of NLDR algorithms that are published each

Authors	Year	Algorithms	Property
Tenenbaum et al. [1]	2000	ISOMAP	Geodesic distance
Roweis et al. [6]	2000	LLE	Preserves linear reconstruction weights
Belkin et al. [8]	2003	Laplacian eigenmaps	Locality preserving
Donoho et al. [9]	2003	HLLE or Hessian eigenmaps	Locally isometric to an open, connected subset
Zhang et al. [10]	2004	LTSA	Minimizing the global reconstruction error
Coifman et al. [11]	2005	Diffusion maps	Preserves diffusion distance
Sha et al. [12]	2005	Conformal eigenmaps	Angle preserving embedding
Weinberger et al. [13]	2006	Maximum Variance Unfolding	Unfold the neighborhood graph

Table 2–1: Algorithms for Nonlinear Dimensionality Reduction

year is a reason for the impossibility of building a global framework -around 40 different algorithms have been proposed in the literature. For example, [17] and [18] are two new methods proposed in the last two years based on new geometric ideas.

## Manifold

Mathematically a manifold <sup>1</sup> is a space that on a sufficiently small scale resemble a euclidean space of a specific dimension [19]. A geometric idea can be intuitively seen using a toy manifold. Figure 2–3 shows a Swiss roll, this is a two dimensional manifold embedded in  $\mathbb{R}^3$ . Dimensionality reduction on this structure is to transform the data to a two dimensional space while preserving the neighborhood structure of the data and, thereby, the global structure of the data. In other words, while some

---

<sup>1</sup> Formally, a topological manifold is a second countable Hausdorff space that is locally homeomorphic to Euclidean space.

information of the data may need a 3 – *dimensional* space to be represented, the most significant information of the data can be expressed on a 2 dimensional space.

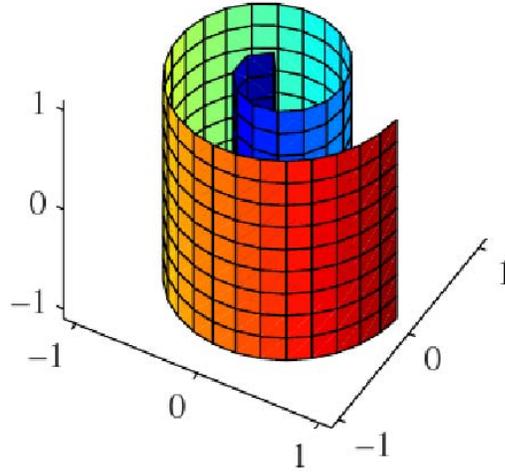


Figure 2–3: Swiss Roll.

On the literature of dimensionality reduction these artificial manifolds on 2 and 3 dimensional spaces are of particular interest because they are used to evaluate and compare the performance of different NLDR algorithms. Of particular interest are artificial manifolds like: Swiss roll, Gaussian, and toroidal helix. State-of-the-art algorithms try to reduce the short-circuits that are created during the construction of the starting graph from the original dataset, Figure 2–4 shows an example of a typically unwanted short-circuit. They are unwanted “*features*” because the reconstruction on the original manifold -optimization over the particular cost function of each NLDR algorithm- will be affected by the non-smooth connectivity of the neighbors around a particular data point.

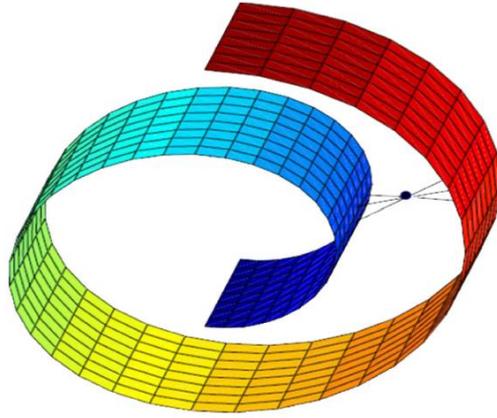


Figure 2–4: Short-Circuit Represented on a Swiss Roll Manifold.

## 2.2 NLDR Algorithms

The NLDR algorithms that will be used in this work are explained in this section:

### 2.2.1 Locally Linear Embedding

This algorithm was developed in 2000 by Saul and Roweis [6]. The idea behind this algorithm is to visualize the manifold as sufficiently smooth so that local patches are approximately linear. Then, the idea is to identify these linear patches and find the mapping that preserves the local geometry in the lower dimensional space. The main assumption is; that if the local geometry is preserved well, then, the global geometry will also be preserved. Algorithm 1 summarizes the LLE algorithm.

### 2.2.2 Isometric Feature Embedding

This algorithm was developed in 2000 by Tenenbaum *et al.* [1]. This algorithm can be viewed as an extension of Multidimensional Scaling (MDS), a classical method for embedding information in the Euclidean space. Algorithm 2 shows the

---

**Algorithm 1** Locally Linear Embedding
 

---

**Input:** Data points  $x_i$  for  $i = 1, \dots, n$  in  $\mathbb{R}^D$

**Output:** Data points  $y_i$  for  $i = 1, \dots, n$  in  $\mathbb{R}^d$

- 1: Construct a graph where each data point  $x_i$  for  $i = 1, \dots, n$  is a node of the graph
- 2: Select an appropriate neighborhood and express each data point as a linear combination of its neighbors.  $N(i) = \{x_j : j \in \{1, 2, \dots, j-1, j+1, \dots, n\} \text{ and } x_j \text{ is a neighbor of } x_i\}$ . The most common methods for performing this step are to select k-nearest neighbors or to select all the neighbors inside certain threshold  $\epsilon$ . As studied in this work the correct selection of this neighborhood is crucial, since all the subsequent steps rely on this selection.
- 3: Minimize the cost function in Eq. (2.1) that projects each data point onto the space spanned by its neighbors. Constraints need to be imposed such that translations and rotations are avoided.

$$e(\mathbf{W}) = \sum_{i=1}^n \left| x_i - \sum_{j \in N(i)} w_j^{(i)} x_{N(j)} \right| \quad (2.1)$$

- 4: After the weights are computed, the high dimensional data is mapped to a low-dimensional space preserving the local structure (weights) of the manifold. This is done by minimizing the function Eq. (2.2) with fixed weights. In this case the optimal low-dimensional coordinates  $\mathbf{Y}_i \in \mathbb{R}^d$  are found, where  $d$  is the dimension of the new space.

$$e(\mathbf{Y}) = \sum_{i=1}^n \left| y_i - \sum_{j \in N(i)} w_j^{(i)} y_{N(j)} \right| \quad (2.2)$$


---

ISOMAP algorithm and Algorithm 3 shows the classical MDS algorithm. ISOMAP seeks to preserve the intrinsic structure of the data as captured in the estimation of the geodesic distance between all data points. The geodesic distance's concept is illustrated in Figure 2–5. This distance is estimated by adding shortest distances between all data points. The approximation is done by using a shortest path algorithm over the constructed Euclidean graph. In this work, methods to construct this Euclidean Graph are studied.

---

**Algorithm 2** ISOMAP

---

**Input:** Data points  $x_i$  for  $i = 1, \dots, n$  in  $\mathbb{R}^D$  and a parameter  $k$

**Output:** Data points  $y_i$  for  $i = 1, \dots, n$  in  $\mathbb{R}^d$

- 1: Find  $k$ -nearest neighbors  $\mathbf{X}_i = x_{i1}, x_{i2}, \dots, x_{ik}$ , of each  $\mathbf{X}_i \in \mathbf{D}$  by comparing the Euclidean distances between all neighbor points and the query point.
  - 2: Compute the shortest paths between all data points (geodesic distance) using Floyd's or Dijkstra's algorithm and store the square of this in the matrix  $P$ .
  - 3: Compute the classical MDS algorithm to find the low dimensional embedding.
- 

---

**Algorithm 3** MDS

---

**Input:**  $\mathbf{P} \in \mathbb{R}^{n \times n}$  with  $\mathbf{P}_{ii} = 0$  and  $\mathbf{P}_{ij} \geq 0$

**Output:**  $\mathbf{X} \in \mathbb{R}^{n \times d}$

- 1: Set  $\mathbf{B} = -\frac{1}{2}\mathbf{H}\mathbf{P}\mathbf{H}$ , where  $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$  is the centering matrix
  - 2: Compute the spectral decomposition of  $\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ .
  - 3: Form  $\mathbf{\Lambda}_+$  by setting  $[\mathbf{\Lambda}_+]_{ij} = \max(\mathbf{\Lambda}_{ij}, 0)$
  - 4: Set  $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}_+^{1/2}$
  - 5: Take  $[\mathbf{X}]_{n \times d}$  as the low dimensional embedding.
- 

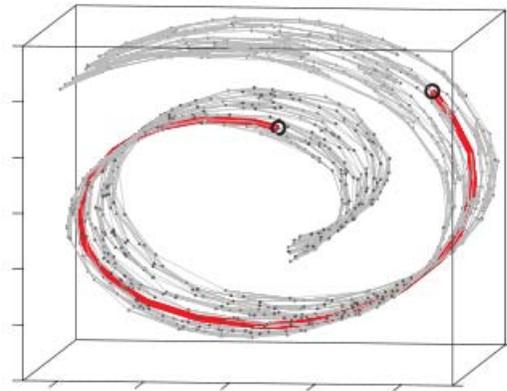


Figure 2–5: Illustration of the Geodesic Distance Concept [1].

### 2.2.3 Kernel PCA

The well known Principal Component Algorithm (PCA) algorithm for linear dimensionality reduction can be extended into a nonlinear version using the *kernel trick*. This algorithm computes the principal eigenvectors of the kernel matrix, instead the covariance matrix as is done in the traditional PCA. The formulation of Kernel PCA is straightforward, as in [4], a complete formulation of the method can be found.

### 2.2.4 Maximum Variance Unfolding

This algorithm was developed in 2006 by Weinberger and Saul [13] and is inspired by a simple idea. Imagine the data-points in the high dimensional space connected to its  $k$  nearest neighbor. The algorithm will try to pull each data-points as far as possible while preserving the pairwise distance of each data-point with its neighbors. This “*pulling*” of the data into the low dimensional space is what gives the name “Maximum variance unfolding”.

First, the usual weighted graph is constructed. The unweighted representation of this graph is the one where the edge  $\delta_{ij} \in \{0, 1\}$  denotes whether data points  $\vec{x}_i$  and  $\vec{x}_j$  are  $k$ -nearest neighbor connected. This unfolding can be formulated as a quadratic program (QP), where the outputs  $\vec{y}_i$  are those that solves the optimization problem.

$$\begin{aligned}
& \underset{\vec{y}_i}{\text{maximize}} && \sum_{ij} \|\vec{y}_i - \vec{y}_j\|^2 \\
& \text{subject to} && \sum_{ij} \|\vec{y}_i - \vec{y}_j\|^2 = \sum_{ij} \|\vec{x}_i - \vec{x}_j\|^2 \text{ for all } (i, j) \text{ with } \delta_{ij} = 1. \\
& && \sum_i \vec{y}_i = 0
\end{aligned} \tag{2.3}$$

The objective function tries to “unfold” the data point. The first restriction tries to preserve the distance between neighbor points and the second restriction centers the solution in the origin to avoid rotations. This optimization problem is nonconvex, which means that local spurious solutions are present. By defining the inner product matrix  $\mathbf{K}_{ij} = \vec{y}_i \cdot \vec{y}_j$  this problem can be reformulated as a semidefinite program (SDP) over the matrix  $\mathbf{K}$ .

$$\begin{aligned}
& \underset{\vec{y}_i}{\text{maximize}} && \text{trace}(\mathbf{K}) \\
& \text{subject to} && \mathbf{K}_{ii} - 2\mathbf{K}_{ij} + \mathbf{K}_{jj} = \sum_{ij} \|\vec{x}_i - \vec{x}_j\|^2 \text{ for all } (i, j) \text{ with } \delta_{ij} = 1. \\
& && \sum_{ij} K_{ij} = 0 \\
& && \mathbf{K} \leq 0
\end{aligned} \tag{2.4}$$

The solution  $\mathbf{K}$  of the SDP is the kernel matrix that is used as the input for the Kernel PCA. The low dimensional embedding is obtained by performing an eigenvector decomposition of the kernel matrix  $\mathbf{K}$ .

### 2.2.5 NLDR in Hyperspectral Data

The nonlinear nature of hyperspectral data is based in the idea of the multi-scattering between the photons and the different materials present in the scene. Plaza et al. [20] described sequences of morphological transformations for dimensionality reduction. Harsanyi et al. [21] investigated hyperspectral dimensionality reduction and posterior image classification by using an orthogonal subspace projection approach. Also, Bruce et al. [22] have used Wavelet transforms for hyperspectral dimensionality reduction.

Special attention has been paid on the three particular methods used in this work: LLE, ISOMAP, and MVU. Chang et al. proposed in [23] the RLLE (Robust Locally Linear Embedding) which modifies the basic LLE algorithm for the case of a strong presence of outliers as is the case of hyperspectral data. In [24], Chen et al. used another approach to improve the LLE algorithm by including a spacial neighborhood window. Finally, in [25], the LLE algorithm is combined with the Laplacian Eigenmaps algorithm, and the authors claim that the method better preserves the local topology of the manifold.

Bachmann et al. [26] developed a hybrid technique to circumvent ISOMAP's computational cost by dividing the scene into a set of smaller tiles. Hou et al. [27] proposed RMVU (Relaxed MVU) to try to deal with problem that causes the short-circuits in the NLDR of hyperspectral data. Finally, in [28], Samaniego et al. proposed a modified  $k$ -NN algorithm suitable to be used with NLDR algorithms, which tries to suppress the known shortcomings of the classical  $k$ -NN algorithm.

Based on the literature there is a great interest of HSI related to NLDR. First, to solve the problem of high computational requirement of NLDR algorithms, which is not the focus of this research work. Second, to use classical NLDR algorithms

in combination with graph construction methods, better than the classical  $k$ -NN, to improve the results on the NLDR algorithms. Also, effectively using the spatial information present in the hyperspectral images for NLDR is of particular interest.

## 2.3 Summary

Four nonlinear dimensionality reduction algorithms have been revised in this chapter. The KPCA algorithm is presented only to give a mathematical context to another NLDR algorithm -Maximum variance unfolding. These three methods share in common the first step of graph construction, which classically is done by using the  $k$ -NN algorithm. In the subsequent sections, different algorithms for graph construction will be presented, these algorithms can be used in combination with any of the NLDR algorithms presented here.

## Chapter 3

---

# Neighborhood Selection and Graph Construction

### 3.1 Spatial Information HSI

Classical algorithms used in hyperspectral images like spectral unmixing and target detection rely mainly on the spectral content of the images. However, some approaches have been proposed to include the spatial information in the analysis. These approaches can be divided in four basic groups based on: image segmentation [29], stacking vector approach [30], filtering (gabor filters) [31] and Markov random fields (MRF) [32], [33].

In hyperspectral images, each pixel is represented as a vector in  $\mathbb{R}^D$ . However image exploitation should take full advantage of spatial information. For example, if a particular pixel belongs to a particular class, the probability that its neighbors belong to the same class is high. In [34], a patch (neighbors of a pixel) is proposed to be used for neighborhood selection. Hence, each component (or band) in each vector is replaced by it's  $n \times n$  immediate neighbors. Equation (3.1) shows how to calculate the distance between two pixels using this concept, where  $\vec{X}_i$  represents the pixels,  $d_t$  refers to the total distance and  $d_P$  represent the distance between patches.

$$d_t(\vec{X}_i, \vec{X}_j) = \sqrt{\sum_k d_P^2(\vec{X}_i(k), \vec{X}_j(k))} \quad (3.1)$$

In [34], rotations of the patches were not included to consider changes in orientation. In this research work, this is included. This means that a transition from one type of material to another may occur in the opposite direction. The rotation of the patch tries to account for this possibility. In the rest of this document, this is the distance measure used.

### 3.1.1 Classical Approach for Neighborhood Selection

Currently the most widely used neighborhood selection techniques are  $k$ -nearest neighbors and  $\epsilon$ -neighbor. The first approach connects each point to its  $k$  nearest neighbors and the second approach connects each point to all points within a predefined Euclidean distance  $\epsilon$ . One of the major problems of these methods is that they do not guarantee the connectivity of the resulting neighborhood graph. This is especially problematic when the data is not evenly sampled. Also, incremental versions are not available.

### 3.1.2 Adaptive Scheme to Select Neighbors

A method called NLE [35] was proposed to try to reduce the influence on the performance of the NLDR algorithms on the selection of the parameter  $k$ . Selection of  $k$  is a trade-off between the redundancy present in the connections of the graph and the isolated nodes. Thus this method tries to adaptively select  $k$  in a more appropriate form. This algorithm is popular on the computer vision literature.

This adaptive scheme selects the neighbors according to the inherent properties of the input data. First, define  $d_{ij}$  as the Euclidean distance from node  $x_j$  to  $x_i$  and

$N_i$  the set containing all the neighbors of  $x_i$ , the neighbor finding procedure of NLE for a given point  $x_i$  can be summarized as follows, in [35], a complete formulation of the method can be found.:

- if  $d_{ij} = \min\{d_{im}\}$  for all  $m \in \{1, 2, \dots, N\}$  then  $x_j$  is regarded as a neighbor of the node  $x_i$ , initially  $S_i = x_j$
- if  $d_{ik} = 2_{ed}\min\{d_{im}\}$ , for all  $m \in \{1, 2, \dots, N\}$  then  $x_k$  is regard as a neighbor of node  $x_i$  if  $d_{jk} > d_{ik}$ .
- when  $S_i$  contains two or more elements, for all  $m \in S_i$ , if  $d_{jm} > d_{ji}$  and  $d_{jm} > d_{mi}$  are satisfy, then  $S_i = S_i \cup \{x_m\}$

### 3.1.3 Cam-distance for Neighborhood Selection

A first approach to improve the performance of the classical approach of  $k$ -NN and  $\epsilon$ -NN, is through the use of a distance measure called “cam-distance”. This distance measure has already been used as preprocessing for the LLE algorithm, and the result has been called “Weigthed locally linear embedding” [36]. This preprocessing step however can be used as the preprocessing to any of the existing NLDR that relies on the same of idea of building a neighborhood graph as the first step of the algorithm. In this research work, this distance measure will be combined with NLDR such as LLE, ISOMAP, and MVU.

This is a distance measure proposed for improving nearest neighbor classification. This new measure tries to take into account the deformation on the distribution of the data due to the attraction and repulsion that each sample point receives from its neighbors. This measure gives deformed cam contours for equal-distance contours. A simple transformation

$$\mathbf{X} = (a + b\mathbf{Y}'\tau/\|\mathbf{Y}\|) \quad (3.2)$$

is used to simulate the deformation of the data.  $\mathbf{Y}$  denotes the original distribution and  $\tau$  is a normalized vector denoting the deformation orientation,  $a$  and  $b$  are parameters to be estimated. The samples used to estimate the parameters  $a, b$ , and  $\tau$  are the  $k$ -nearest neighbors. Then, an inverse transformation is performed

$$\mathbf{Y} = \mathbf{X}/(a + b \cos \theta) \quad (3.3)$$

to eliminate the deformation. Here  $\theta$ , is the angle between vectors  $\mathbf{Y}$  and  $\tau$ .

**Definition 1.** (*Cam distribution*) [37] Consider a  $p$ -dimensional random vector  $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_p)'$  that takes a  $p$ -dimensional Gaussian distribution

$$f(y) = \frac{1}{(2\pi)^{p/2}} e^{-(y'y)/2} \quad (3.4)$$

Let  $\mathbf{X}$  be a random vector defined by

$$\mathbf{X} = \left( a + b \frac{\mathbf{Y}'\tau}{\|\mathbf{Y}\|} \right) \mathbf{Y} \text{ or } \mathbf{X} = (a + b \cos \theta) \mathbf{Y} \quad (3.5)$$

where  $a > b \geq 0$ ,  $\tau$  is a normalized vector,  $\|\mathbf{Y}\| = \sqrt{\mathbf{Y}'\mathbf{Y}}$  and  $\theta$  is the included angle of vectors  $\mathbf{Y}$  and  $\tau$ . Then, the distribution of  $\mathbf{X}$  is called the cam distribution with parameters  $a$  and  $b$  in the direction  $\tau$ , denoted  $\mathbf{X} \sim \text{Cam}_p(a, b, \tau)$ .

**Theorem 1.** [37] If a random vector  $\mathbf{X} \sim \text{Cam}_p(a, b, \tau)$ , then

$$E(\mathbf{X}) = c_1 b \tau \quad (3.6)$$

and

$$E(\|\mathbf{X}\|) = c_2 a \quad (3.7)$$

where  $c_1$  and  $c_2$  are constants

$$c_1 = 2^{1/2} \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} / p \quad (3.8)$$

$$c_2 = 2^{1/2} \frac{\Gamma((p+1)/2)}{\Gamma(p/2)} \quad (3.9)$$

The new distance is denoted  $\text{CamDist}(x_o, x) = \|x - x_o\| / (a + b \cos \theta)$ . Figure 3–1 shows two examples of cam distributions  $\text{Cam}_2(1, 0, [0.8, 0.6])$  and  $\text{Cam}_2(1, 0.4, [0.8, 0.6])$ . In these figures, what we have is an equidistant surface modified by the density of the distribution of the original data.

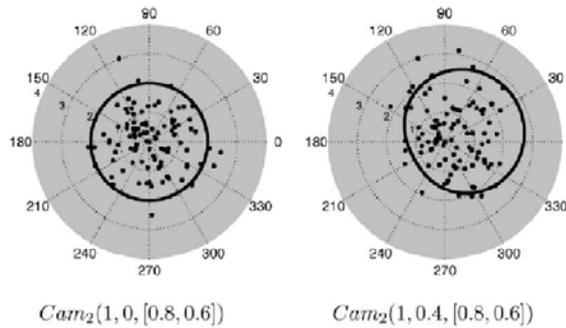


Figure 3–1: Cam Weighted Distance.

---

**Algorithm 4** Cam-Distance

---

**Input:** Data points  $F = \{x_i\}$  for  $i = 1, \dots, n$ , with  $x_i \in \mathbb{R}^D$  and parameters  $k_1$  and  $k_2$

**Output:** Minimum  $k_2$  nearest neighbor for each data point

- 1: Find  $k_1$ -nearest neighbors  $X_i = x_{i1}, x_{i2}, \dots, x_{ik_1}$ ,  $X_i \in F$  by comparing the Euclidean distance.
  - 2: Estimate the parameters  $a$ ,  $b$  and  $\tau$  using the  $k_1$ -nearest neighbors.
  - 3: Find the  $k_2$ -nearest neighbor for each data point using the deformed distance  $\text{CamDist}(x_o, x) = \|x - x_o\| / (a + b \cos \theta)$
-

Algorithm 4 shows the resulting algorithm. It should be noted that two nearest neighbors have to be used namely  $k_1$  and  $k_2$ . In the subsequent analysis, a parameter called  $\alpha$  will be used, which will represent the factor between the  $k_1$  and  $k_2$  values. This algorithm can also be extended to define a particular value of  $k$  for each data point, if some prior knowledge of the distribution of the data is available. For example, which parts of the data contains more noise than others, in that case the noisy parts should include a greater value of  $k_2$  to try to reduce the variability of the data. It should be clear that, when  $k_1 = 0$ , the cam-distance algorithm is equivalent to a regular  $k$ -NN algorithm since the distance measure is not modified by the distribution of the data around the data point.

#### 3.1.4 Graph Construction for Neighborhood Selection

A neighborhood graph should be constructed so that its connectivity is guaranteed to assure that the true structure of the data is preserved and that all the data points are included. Also, multiple edge connections are expected between any partition of the graph. These connections are expected to better reflect the geodesic structure of the data. A good choice for such a graph is a  $k$ -connected or  $k$ -edge-connected spanning subgraph of the complete Euclidean graph. As geodesic distances should follow smoothly the structure of the data, such a graph should have a minimum sum of edge lengths. Unfortunately, the problem of finding such a minimum spanning subgraph for  $k \geq 2$  is NP-complete [38].

Four approaches,  $k$ -MST [39], Min- $k$ -ST [40],  $k$ -EC [41] and  $k$ -VC [42] have been proposed to overcome the problem of disconnected neighborhood graphs. These methods build  $k$ -edge-connected or  $k$ -connected neighborhood graphs.  $k$ -MST works by repeatedly extracting  $k$  minimum spanning trees from the original Euclidean graph. Min- $k$ -ST extracts  $k$  spanning trees for which the sum of the total lengths

is minimum.  $k$ -EC builds a  $k$ -edge connected neighborhood graph by adding each edge, in a non-decreasing order of edge length, to a partially formed graph if end vertexes of the edge to be added are not yet  $k$ -edge-connected in the graph.  $k$ -VC works in a similar way as  $k$ -EC except that it adds each edge to the graph if end vertexes of the edge to be added are not yet  $k$ -vertex-connected. As a result,  $k$ -VC builds a  $k$ -connected neighborhood graph instead of a  $k$ -edge-connected one.

### **k-MST**

Construct a  $k$ -edge connected neighborhood graph by repeatedly extracting the MST of the complete euclidean graph as described in Algorithm 5.

---

#### **Algorithm 5** $k$ -MST

---

**Input:** Given graph  $G = (V, E)$

**Output:**  $k$ -edge-connected neighborhood graph

```

1: for  $i = 1, \dots, k$  do
2:   if  $i = 1$  then
3:      $MST_1 = mst(V, E)$ 
4:   else
5:      $MST_i = mst(V, E - \cup_{j=1}^{i-1} MST_j)$ 
6:   end if
7: end for

```

---

### $k$ -EC

Use a greedy algorithm to construct a  $k$ -edge-connected neighborhood graph as shown in Algorithm 6.

### $k$ -VC

Construct a  $k$ -connected neighborhood graph by adding each edge in a non-decreasing order of edge length, to a partially formed neighborhood graph if end vertexes of the edge are not yet  $k$ -connected as described in Algorithm 7.

---

**Algorithm 6**  $k$ -EC
 

---

**Input:** Given graph  $G = (V, E)$

**Output:**  $k$ -edge-connected neighborhood graph

- 1:  $E_k = \emptyset$ ;  $nc = |V|$ ; Assign each  $v \in V$  a unique component;
  - 2: Sort edges in  $E$  in non-decreasing order for edge length;
  - 3: **while**  $nc > 1$  **do**
  - 4: Get the next edge  $(u, v) \in E$  in the order;
  - 5: **if**  $u$  and  $v$  belong to different components **then**
  - 6: **if**  $u$  is not  $k$ -edge-connected to  $v$  in  $G_k$  **then**
  - 7:  $E_k = E_k \cup \{(u, v)\}$
  - 8: **else**
  - 9: Union the components  $u$  and  $v$  belong to;
  - 10:  $nc = nc - 1$
  - 11: **end if**
  - 12: **end if**
  - 13: **end while**
- 

---

**Algorithm 7**  $k$ -VC
 

---

**Input:** Given graph  $G = (V, E)$

**Output:**  $k$ -connected neighborhood graph

- 1:  $E_k = \emptyset$ ;  $\forall v \in V$ ,  $\text{Block}_v = \emptyset$
  - 2: Sort edges in  $E$  in nondecreasing order of edge length
  - 3: **for** each  $(a, b) \in E$  in the order **do**
  - 4: **if**  $|\text{Block}_a \cap \text{Block}_b| < k$  and the vertices  $a$  and  $b$  are not  $k$ -connected in  $G_k$  **then**
  - 5:  $E_k = E_k \cup \{(a, b)\}$
  - 6: **else**
  - 7:  $\text{Block}_a = \text{Block}_a \cup \{b\}$ ;  $\text{Block}_b = \text{Block}_b \cup \{a\}$ ;
  - 8: **end if**
  - 9: **end for**
-

A great advantage of these methods is the fact that an incremental version have been created [43]. Classical methods for neighborhood selection need all the data available to perform the analysis. When new data is available, the whole process has to be repeated to include the new data and no incremental version is available -although some heuristics have been proposed. In the case of the methods introduced here, incremental versions [43] are possible with little effort to include the new data. In the context of hyperspectral images, when on-line real-time stream of images are acquired, these incremental versions are of fundamental importance in the performance of NLDR algorithms.

### 3.1.5 Short-Circuits Comparison

As explained before three dimensional manifolds are used to test and compare the different NLDR algorithms and also neighborhood selection algorithms. An example using the Swiss Roll data set will be given where the presence of the short-circuits can be compared between the classical  $k$ -NN algorithms and the proposed neighborhood selection algorithms. Figure 3–2 to Figure 3–5 show the different results obtained for the Swiss Roll data set with 8 neighbors.

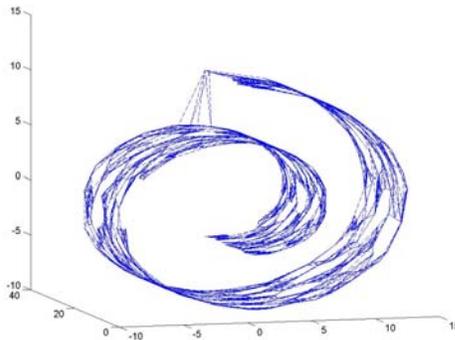
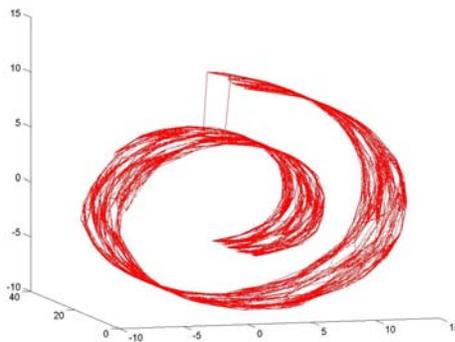
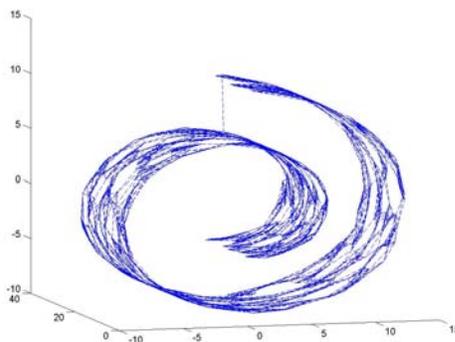
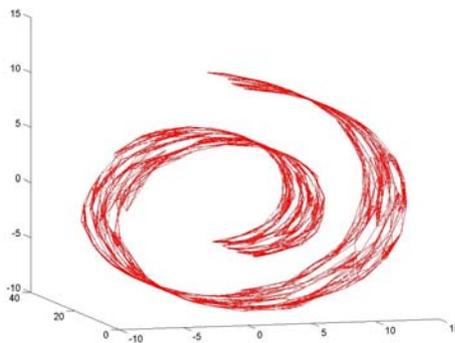


Figure 3–2:  $k$ -NN.

Figure 3-3:  $k$ -MST.Figure 3-4:  $k$ -EC.Figure 3-5:  $k$ -VC.

For any NLDR algorithm using the  $k$ -NN algorithm for the graph construction, the performance will be significantly deteriorated because of the presence of short-circuits. It should be noted that after the graph is constructed, the rest of the

NLDR process can be considered as a black box with only few tunable parameters, depending on the selected method. However the performance of the methods rely heavily on graph construction.

### 3.1.6 Improving Geodesic Distance Estimation

One of the most famous NLDR algorithms relies on the idea of geodesic distance, this method is called ISOMAP. Other methods which also relies on this idea are Curvilinear distance analysis [44] and GeoNLM [45]. It has been claimed that since geodesic distance reflects the underlying structure of the data, geodesic distance based methods are expected to correctly unfold highly nonlinear manifolds. As can be deduced of the ISOMAP algorithm explained in Subsection 2.2.2, the performance of this method depends heavily on two factors: the correct construction of the neighborhood graph, and the correct calculation of the shortest path between data points, which represent the geodesic distance. Thus accurately calculating the geodesic distance has become an important issue in improving the computation of the so called geodesic distance-based methods and is the direction towards which this research goes.

In [46], a heuristic method aiming to improve the precision of the geodesic distance estimation is presented. The main principle of this heuristic is the same in which the ISOMAP algorithms relies, which is the local linear assumption of the manifold. The geodesic distance estimation is improved by solving a convex optimization problem.

The *locally linear assumption* can be better explained by the following definition [46]:

**Definition 2.** The locally linear patch corresponding to a data point  $x_i$  is defined as:

$$\Omega_i = \left\{ \sum_{k=1}^{N_i} \lambda_{ik} x_{ik}, \sum_{k=1}^{N_i} \lambda_{ik} = 1, 0 \leq \lambda_{ik} \leq 1, k = 1, \dots, N_i \right\} \quad (3.10)$$

where  $x_{ik}$  are the neighborhood vertexes of  $x_i$ , and  $N_i$  is the number of vertexes in the neighborhood.

The locally linear assumption (LLA) assumes that any locally linear patch  $\Omega_i$  approximately resides on the manifold. As can be seen, the the geodesic distance calculation between two adjacent patches (two common vertex) will pass throughout one of those vortex. But a more truly geodesic distance can be taught as passing through a point (virtual vertex) between the two common vertexes. To facilitate the discussion, a definition is given [46]:

**Definition 3.** A 2-chain patch is an ordered list of 2 locally linear patches in which adjacent patches share common vertex. The 2-chain patch set, is the collection of all 2-chain patches.

Figure 3–6 shows an example of a 2-chain patch. The geodesic distance calculation can be improved by solving the following convex optimization problem:

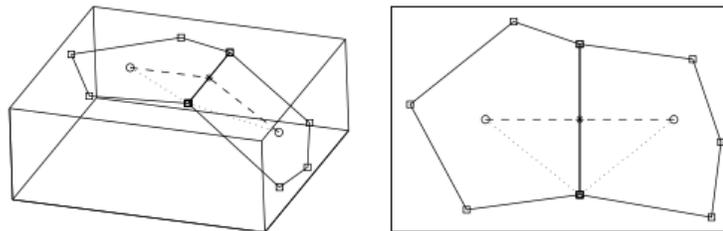


Figure 3–6: 2-Chain Patch in 3D Space.

$$\begin{aligned}
& \underset{x}{\text{minimize}} && f(x) = \|x_i - x\| + \|x_j - x\| \\
& \text{subject to} && x = \sum_{k=1}^{N_i} \lambda_k x_{ik}, \sum_{k=1}^{N_i} \lambda_k = 1 \\
& && x = \sum_{k=1}^{N_i} \mu_k x_{jk}, \sum_{k=1}^{N_i} \mu_k = 1 \\
& && 1 \geq \lambda_k \geq 0, k = 1, \dots, N_i \\
& && 1 \geq \mu_k \geq 0, k = 1, \dots, N_j
\end{aligned} \tag{3.11}$$

This is a strictly convex programming problem which can be efficiently solved. A unique optimal solution  $x^*$  can be obtained. Accordingly, the path from  $x_i$  to  $x_j$  through  $x^*$  can be added to the original neighborhood graph to more accurately reflect the geodesic distance from  $x_i$  to  $x_j$ .

Based on this analysis, Algorithm 8 the proposed geodesic distance calculation is formulated. This algorithm is called the 2-chain geodesic distance Algorithm 2.2.2 [46].

---

**Algorithm 8** 2-chain geodesic distance

---

**Input:** Data set  $X = \{x_i\}_{i=1}^l$

**Output:** Estimated geodesic distance matrix  $D = D_{i,j,l \times l}$

- 1: Apply any neighborhood graph construction method to build a graph  $G = (V, E)$ , where  $V$  is the vertex set  $X$  and  $E$  is the neighborhood edge set.
  - 2: **for** All 2-chain patches  $\Omega_i$  and  $\Omega_j$  in the 2-chain patch set  $\Lambda_2$  **do**
  - 3:   Obtain the solution  $x^*$  by solving the optimization problem. Let  $E = E \cup (x_i, x_j)$ , where the edge  $(x_i, x_j)$  is with weight  $f(x^*)$ .
  - 4: **end for**
  - 5: Compute the length of the shortest path between any data pair  $x_i$  and  $x_j$  in the neighborhood graph  $G = (V, E)$  and take it as the approximate geodesic distance  $D_{i,j}$  between the pair.
- 

As was pointed earlier, this algorithm can be used as the first step for any NLDR algorithm based on geodesic distance calculation. For this research work, the only algorithm which relies on this distance measure is the ISOMAP algorithm.

## 3.2 Performance Evaluation

The evaluation of the above proposed algorithms will be carried out using classification results on the low dimensional space.

### 3.2.1 Mahalanobis Distance Classifier

We want to obtain a rule that assigns each pixel into one of the  $\mathbf{C}$  classes. The parameter  $\mathbf{C}$  refers to the number of classes where a given pixel can be classified. This parameter depends on the particular characteristic of the image being studied. Mahalanobis distance is used in analyzing cases in discriminant analysis. The Mahalanobis distance, Eq. (3.12) is the distance between the test point and the centroid of each group (class). A given test point (pixel) has a given distance to each centroid's group, it is classified as belonging to the group for which its Mahalanobis distance is the smallest. Thus, the smaller the Mahalanobis distance, the closer the test point (pixel) to the group centroid, the more likely it is to be classified as belonging to that group. If  $\mathbf{X}$  is the vector pixel,  $\mathbf{M}_i$  is the mean vector of the class  $i$ ,  $\sum_i$  the covariance of the class  $i$ , and  $N$  the number of bands.

$$g_i(x) = -(\mathbf{X} - \mathbf{M}_i)^T \left( \sum_i \right)^{-1} (\mathbf{X} - \mathbf{M}_i) \quad (3.12)$$

### 3.2.2 Maximum Likelihood Classifier

This classifier is based on statistical information. Assuming that the vector pixel  $\mathbf{X}$  is normally distributed with mean and covariance unknown, the likelihood function becomes:

$$g_i(x) = -\frac{1}{2} \ln \sum_i -(\mathbf{X} - \mathbf{M}_i)^T \sum_i^{-1} (\mathbf{X} - \mathbf{M}_i) \quad (3.13)$$

The vector pixel  $\mathbf{X}$  belongs to the class that has the function with the largest  $g_i(x)$ . The mean  $\mathbf{M}_i$  and covariance matrix  $\sum_i$  are estimated from the training data by the following estimators:

$$\mathbf{M}_i = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathbf{X}_k \quad (3.14)$$

$$\sum_i = \frac{1}{n_i - 1} \sum_{k=1}^{n_i} (\mathbf{X}_k - \mathbf{M}_i)(\mathbf{X}_k - \mathbf{M}_i)^T \quad (3.15)$$

### 3.3 Summary

$k$ -NN algorithm does not guarantee the connectivity of the graph. In this case, there is a trade off between a small and a large value of  $k$ . A small value of  $k$  tends to create disconnected graphs which reduces the performance of criterion selected in this research, overall classification accuracy. The extreme case  $k = 0$  all points are totally separated. On the other hand, a large value of  $k$  increases the redundancy and overlapping of the connections. This also tends to lower the classification accuracy as the data loses its discriminative capabilities. The extreme case with  $k = N - 1$  illustrates how the graph is transformed into a cluster where all the points are connected.

Some graph construction algorithms will be used in the context of hyperspectral data. The NLE algorithm is proposed as an alternative to the  $k$ -NN algorithm without the need of the parameter  $k$ . The cam-distance algorithm is proposed as

an alternative that is able to take into account the local variability in the distribution of the data, to better perform the nearest neighborhood selection. The graph construction algorithms  $k$ -MST,  $k$ -EC and  $k$ -VC have the advantage of creating a connected graph and have been proven to outperform the classical  $k$ -NN in artificial datasets.

Finally, an algorithm to generate better estimates of the geodesic distance is described. In this research, the only algorithm that relies on geodesic distance is the ISOMAP algorithm. In the next chapter, tests will be performed using real hyperspectral datasets.

## Chapter 4

---

# Algorithms Validation and Experimental Results Using Hyperspectral Data

### 4.1 Methodology

Two different datasets were used to evaluate the performance of NLDR algorithms. The process for dimensionality reduction is divided into three parts, Figure 4-1 shows a description of the process. The first part is graph construction. For this part seven different approaches will be used: classical  $k$ -NN,  $k$ -MST,  $k$ -EC,  $k$ -VC, cam-distance, and finally, for the ISOMAP algorithm, the 2-chain algorithm to improve the geodesic distance estimation will be used. These approaches were explained in Chapter 3.

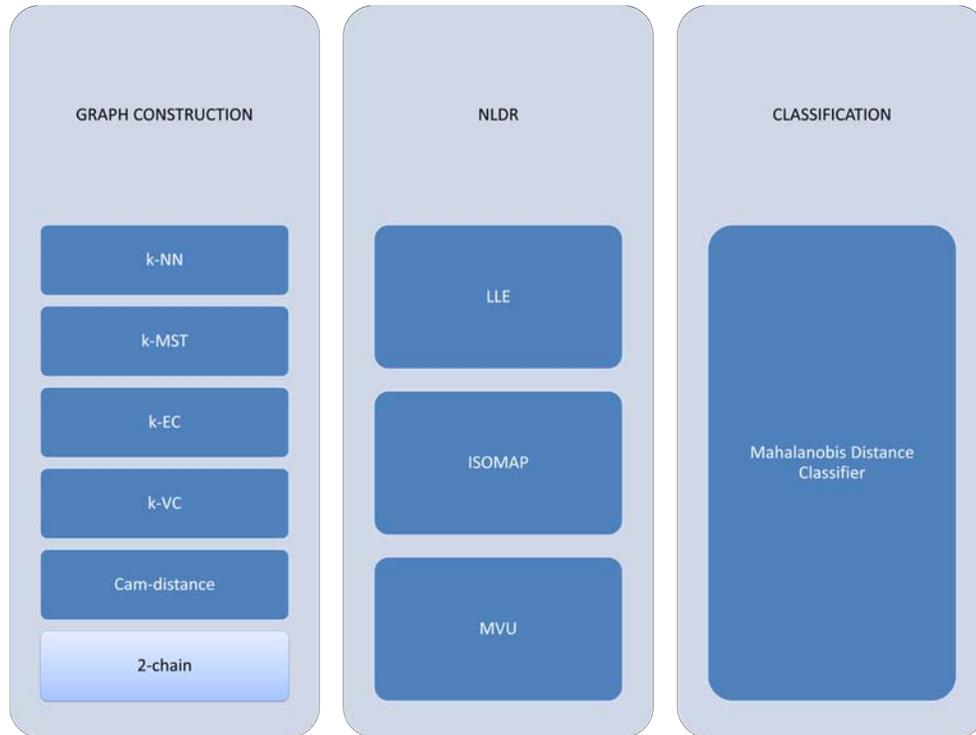


Figure 4-1: Process of NLDR.

The second part, is the nonlinear projection of the data into the low dimensional space. This projection is performed using the NLDR algorithms explained in Chapter 2. The algorithms used are LLE, ISOMAP and MVU, and the parameter used is  $d$  which represents the number of lower dimensions of the space where the data is projected. Finally, for the classification the well known Mahalanobis distance and maximum likelihood classifiers were used. Figure 4-1 shows a sketch of all the algorithms that are used for each part of the process of NLDR and the classifier used to evaluate the performance.

Testing for each of the dataset will be divided in 6 parts. Each part will be used to set some parameter of the algorithms in some of the cases, and in the other cases will be used to evaluate the performance of a particular algorithms reducing the number of parameters been varied. The evaluation of all possible combination of

neighborhood selection algorithms with all the NLDR algorithms is huge, we focused on few combinations.

First, the the NLE algorithm will be tested for hyperpectral images. This algorithm for neighborhood selection has been extensively used in the literature of computer vision as a successful attempt to construct the graphs independent of the  $k$  parameter. Second, the cam-distance algorithm will be tested with different ratios between the number of neighbors of the first and second iteration of the algorithm. Third, the advantage of using graph construction algorithms that produce connected graphs, will be shown by testing the  $k$ -NN algorithm with the lowest  $k$  that produces a connected graph. Fourth, the coherent distance approach will be tested to validate its usefulness for hyperspectral images. After performing these initials steps the best performing algorithms for graph construction and a correct approximation for the value of  $\alpha$  on the cam-distance algorithm can be set.

The fifth step is to test changing different parameters, and using the above algorithms mentioned to validate which of the algorithms outperforms the classical  $k$ -NN. The sixth, and final step is to choose some of the graph construction methods with the ISOMAP, the only Algorithm that depends on the geodesic distance. This is done to evaluate how the optimization over the geodesic distance mentioned above will perform using a real high dimensional dataset.

## 4.2 Experiment 1 - Fake Leaves

### 4.2.1 Imagery

The Fake Leaves dataset shown in Figure 4–2 is provided by Surface Optics Corporation (SOC) as a testing data set of their SOC-700 hyperspectral camera Figure 4–3. The classes in this image are true leaves, false leaves, wall, a jar, a

metallic case, a flowerpot, plastic label, paper label, and lens cover of the SOC-700 hyperspectral camera that appears in the image. Only the first five classes were used in the experiments. This dataset has 110 bands between 430 and 900 nanometers, with a spectral resolution of 4 nanometers. A subset of  $619 \times 553$  pixels was used. Figure 4-4 shows the different areas of the image, white label corresponds to the training area, and gray label corresponds to the testing area.

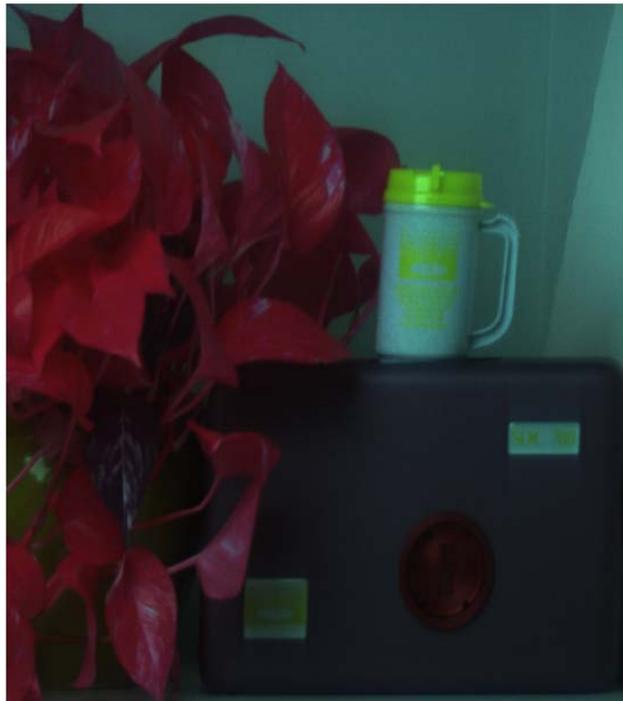


Figure 4-2: Fake Leaves Data Set (RGB corresponds to bands 90, 68, and 29).



Figure 4-3: SOC-700 Hyperspectral Camera.

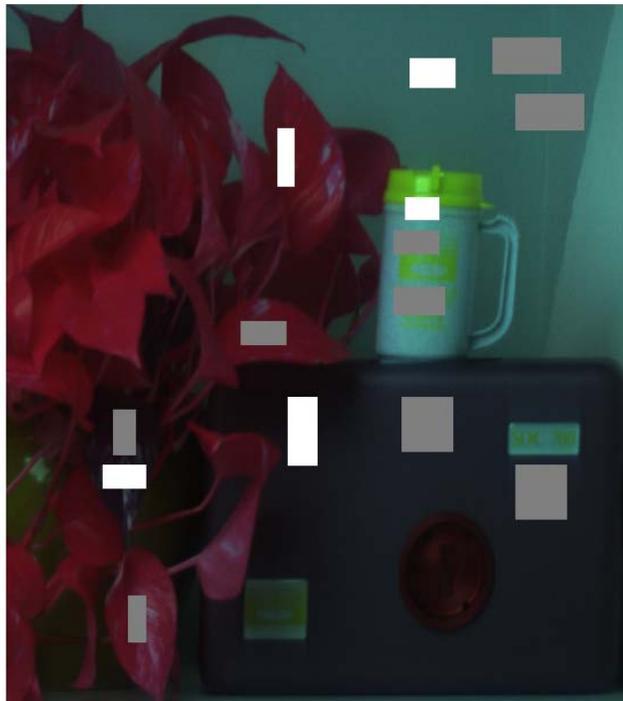


Figure 4-4: Testing and Training Data of the Fake Leaves Data Set.

### 4.2.2 Results and Discussion

The NLE algorithm was used to construct the graph, the results are compared to the performance using the  $k$ -NN algorithm. As the NLE algorithm does not depend on the parameter  $k$ , the testing in this case is done by setting parameter  $k$  to 15, and changing the number of dimensions of the lower space, where the data is projected. The parameter  $k$  was set based on different test to evaluate the behavior of the nonlinear projection when using the different algorithms. The overall classification accuracy obtained using the LLE algorithm is shown in Figure 4–5, and using the ISOMAP algorithm in Figure 4–6.

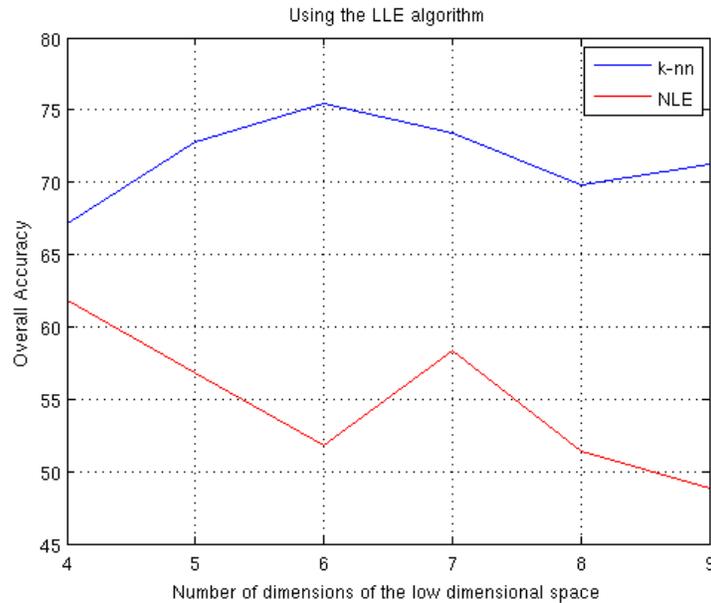


Figure 4–5: Overall Accuracy Using NLE for Graph Construction, LLE for DR and Mahalanobis Distance for Classification.

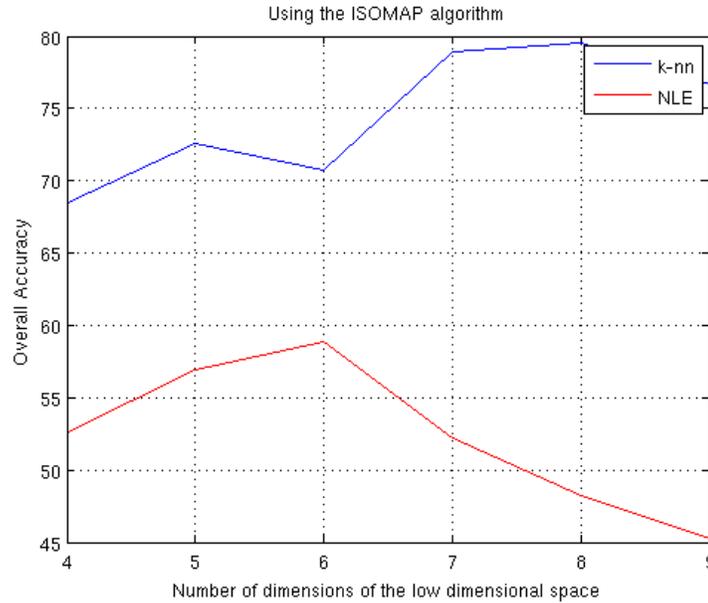


Figure 4–6: Overall Accuracy Using NLE for Graph Construction, ISOMAP for DR and Mahalanobis Distance for Classification.

This algorithm has the drawback that it is too strict on reducing the redundancy of the neighborhood construction, hence usually few neighbors are selected. In the ideal case of a dense smooth sample manifold, this algorithm could perform outstandingly well. However in this real hyperspectral data set the results were poor. The mean value of neighbors found by this algorithm was 9.4, which shows that the connectivity of the graph was too low. The MVU algorithm did not converge in many cases due to the low connectivity of the graph, and for that reason those results are not reported, only the results obtained with the LLE and ISOMAP algorithms are reported.

Now the cam-distance algorithm with different ratios of the number of neighbors in the first part of the algorithm with respect to the actual desirable neighbors will be tested. For this purpose, the three NLDR algorithms will be used. Classification will be done using the maximum likelihood classifier. The number of dimension where the data is embedded is set to 12, and the number of neighbors is set to 15.

The overall accuracy using the  $k$ -NN algorithm using  $d = 10$  and  $k = 15$  is plotted in the subsequent figures showing the results. This accuracy is the same for all the cases, as the only parameter being changed is the ratio of the two numbers of neighbors needed for the cam-distance algorithm.

The results for the LLE, ISOMAP and MVU algorithms are shown in Figure 4–7 to Figure 4–9 respectively. It should be noted that the results for the  $k$ -NN algorithm is represented by a straight line, since the variation is over the factor  $\alpha$  for the cam-distance algorithm. Also, it should be noted that the actual value for  $k_1$ , the first number of neighbors in the cam-distance, was the nearest integer to the value  $\alpha \times 15$ .

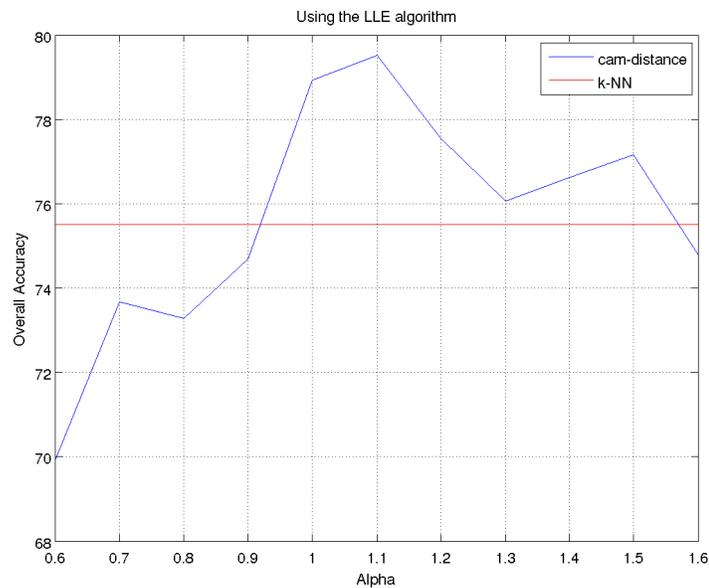


Figure 4–7: Overall Accuracy Using Cam-distance for Graph Construction, LLE for DR and Maximum Likelihood for Classification.

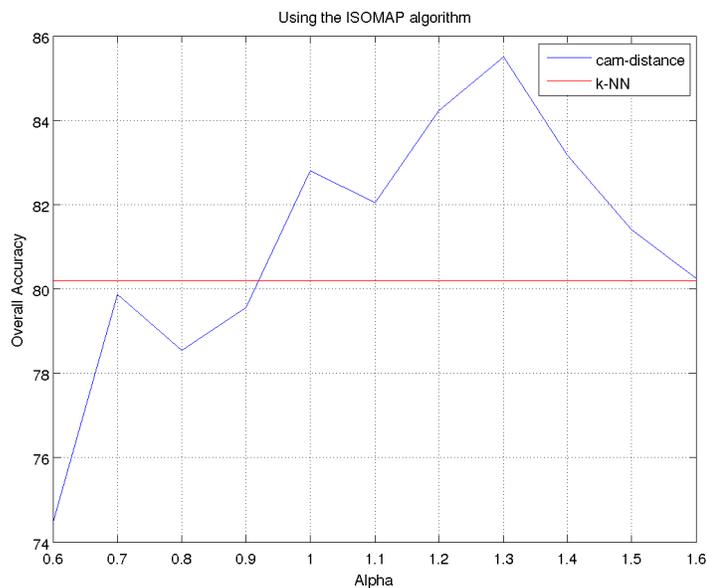


Figure 4–8: Overall Accuracy Using Cam-distance for Graph Construction, ISOMAP for DR and Maximum Likelihood for Classification.

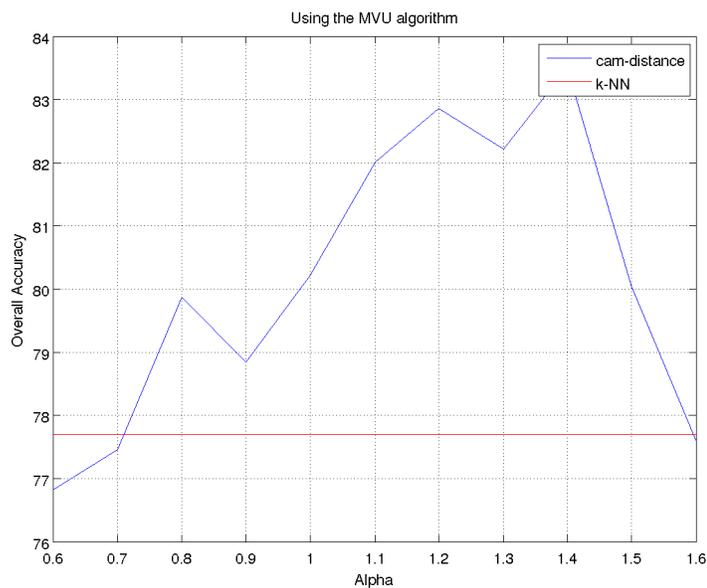


Figure 4–9: Overall Accuracy Using Cam-distance for Graph Construction, MVU for DR and Maximum Likelihood for Classification.

From these figures, obtained by changing the parameter  $\alpha$ , it should be noted that the results using the ISOMAP algorithm are the ones that present lower variability. This can be related to how the geodesic distance is calculated, and how some

incorrectly selected neighbors do not affect significantly how the geodesic distance is estimated. It can be concluded that a range between 1.1 and 1.4 works correctly for the three algorithms.

As mentioned before, the classical  $k$ -NN algorithm has the disadvantage that a trade off is present. This trade off is between a small  $k$  which tends to create a disconnected graph and a large  $k$  which generate redundancy in the connectivity of the graph. For this particular dataset, the smallest value of  $k$  which generates a connected graph, is  $k = 27$ .

NLRD Algorithm	LLE	ISOMAP	MVU
Classification Accuracy	60%	57%	48%

The table above shows the classification accuracy obtained when the data is projected into the lower dimensional space using the value of  $k$  and  $d = 8$  with the Mahalanobis distance classifier. This table shows how the performance is significantly degraded due to the presence of redundancy in the connectivity of the graph. From this test, it should be clear also that for any value of  $k$  below 26, the algorithm  $k$ -NN always produces a non connected graph. This means that the mapping from some data points to the lower space is unknown and for that reason the classification accuracy is reduced. On the other hand, the graph construction methods  $k$ -MST,  $k$ -VC and  $k$ -EC always generates connected graphs. Also it should notice that the behavior in the classification accuracy when using the  $k$ -NN for graph construction is expected to produce a bell shape as the number of neighbors goes from 1 to  $N$ , disjoint points and a cluster where every point is connected to each other respectively.

Coherent distance (see Subsection 3.1), is an algorithm that includes the spatial information present in the hyperspectral image by modifying the distance measure

when constructing the graph. For this part of the test, the parameter  $d$  is fixed to 8, the number of nearest neighbors is varied from 9 to 18 and the maximum likelihood classifier is used. The cam distance algorithms is tested using a value of  $\alpha = 1.2$ . The overall classification accuracy tends to improve for most of the combination of graph-construction algorithm and NLDR. To present a summary of the results, only the two best and worst cases will be presented in full detail. It should also be noted that as shown in [29] the classical  $k$ -NN algorithm was the one which presented a better improvement in the performance for most cases.

On the other hand, the  $k$ -EC algorithm tends to reduce the performance when the coherent distance is used. Figure 4–10 shows the result using the  $k$ -NN algorithm and the ISOMAP algorithm. Figure 4–11 shows the result when using the  $k$ -VC algorithm and the LLE algorithm. These combinations of algorithms were the ones which presented the best improvement in the performance with and without using the coherent distance. Finally, the Figure 4–12 shows result when using the  $k$ -EC algorithm together with the MVU algorithm. In this case the performance is actually reduced by a significant amount around 6% in the worst case.

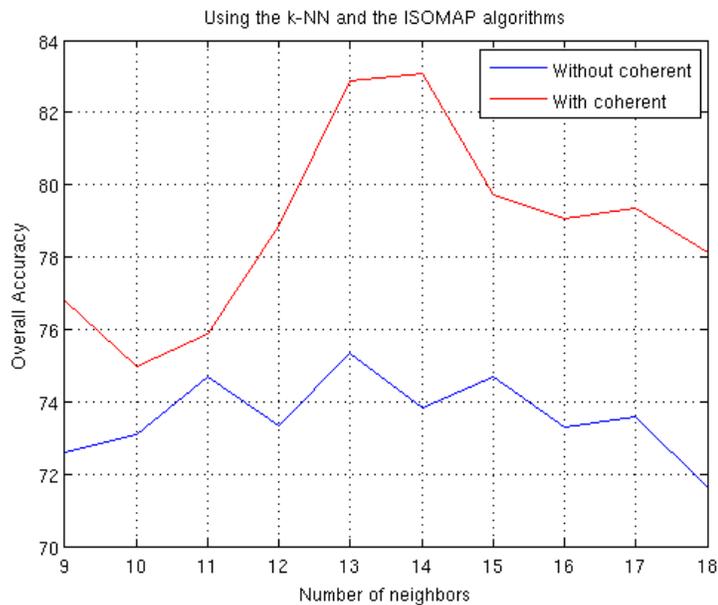


Figure 4–10: Overall Accuracy Using the  $k$ -NN Algorithm for Graph Construction, and DR Using the ISOMAP Algorithm.

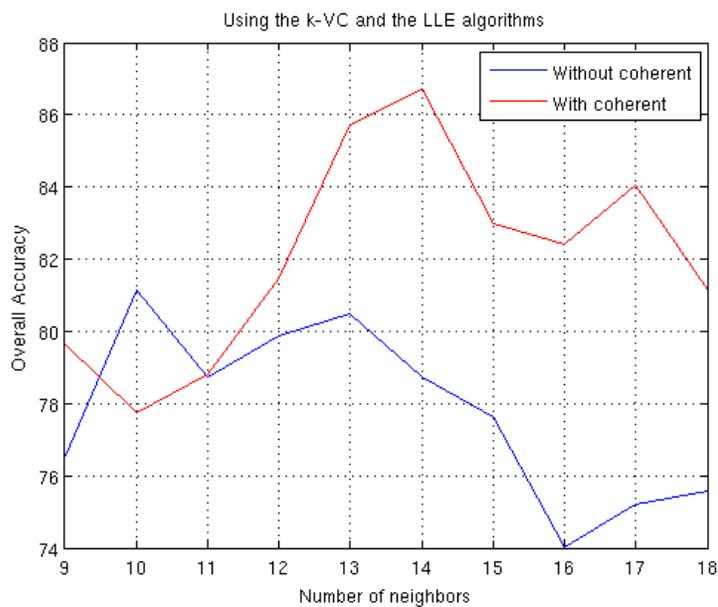


Figure 4–11: Overall Accuracy Using the  $k$ -VC Algorithm for Graph Construction, and DR Using the LLE Algorithm.

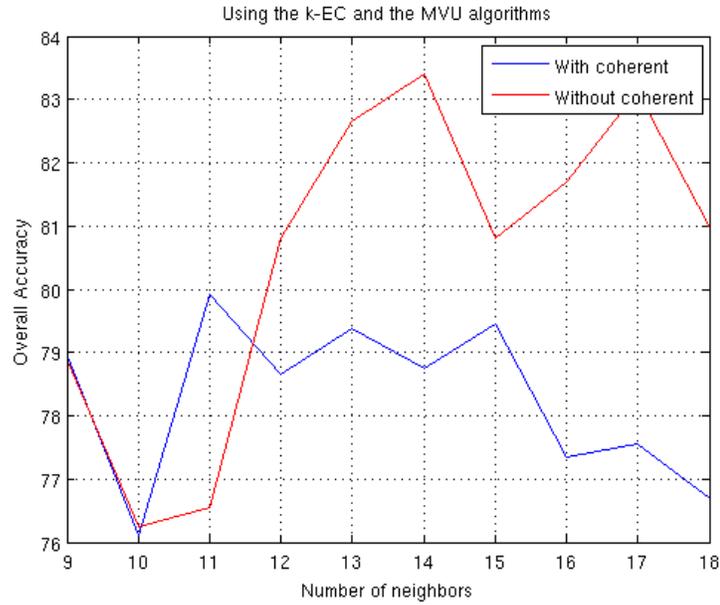


Figure 4–12: Overall Accuracy Using the  $k$ -EC Algorithm for Graph Construction, and DR Using the MVU Algorithm.

At this point, we are ready to perform a comparison in the performance of the different graph construction algorithms. The NLE algorithm will not be used as this algorithm performs poorly on this data set. The parameter for the cam distance algorithm will be  $\alpha = 1.2$ , and the parameter  $d$  will be equal to 8. The coherent distance will be used as it improves the performance for most of the algorithms. Figure 4–13 shows the results when using all the graph construction algorithms.

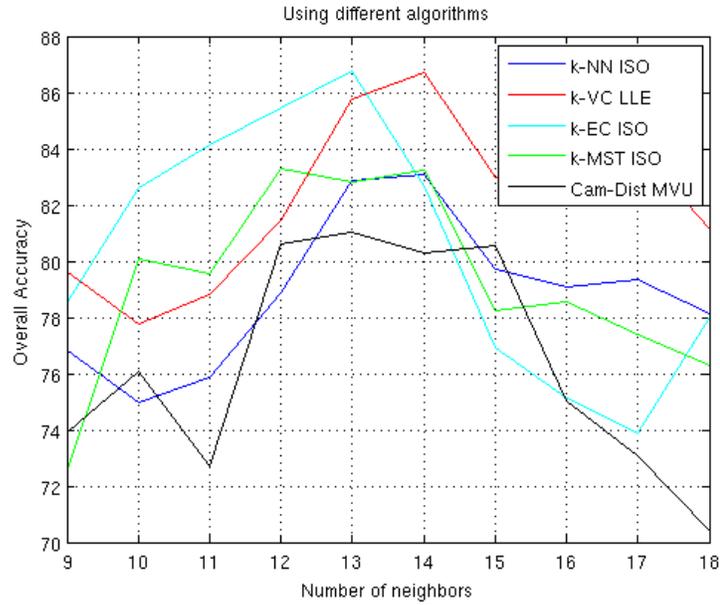


Figure 4–13: Overall Accuracy Using all the Described Graph Construction Algorithms.

For the final test, the best performing graph construction algorithm when combined with the ISOMAP algorithm was selected. The geodesic improvement estimation algorithm described in Subsection 3.1.6 was applied to this case. The result using the  $k$ -VC and the ISOMAP algorithm is shown in Figure 4–14. On 3% of the cases, the optimization algorithm did not converge and then the optimization algorithm was not used, instead the original geodesic estimation was used. Figure 4–14 shows the improvement in the overall accuracy.

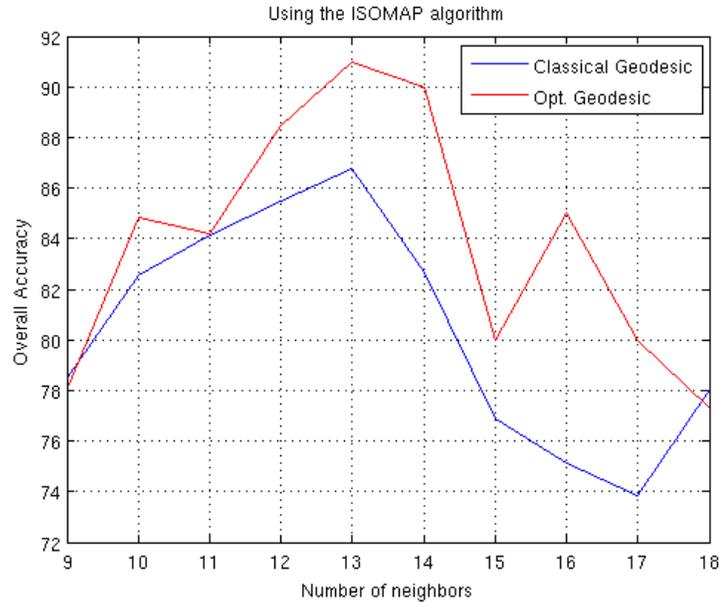


Figure 4–14: Overall Accuracy Improvement Using an Algorithm to Improve the Geodesic Distance Estimation.

## 4.3 Experiment 2 - Indian Pines

### 4.3.1 Imagery

An image taken with the AVIRIS (Airbone Visible/Infrared Imaging Spectrometer) sensor [47], flown by NASA/Ames on June 12, 1992, over an area 6 miles west of West Lafayette, Indiana. This image contains  $145 \times 145$  pixels and 220 spectral bands in the 400-2500 nm range, for which ground truth exists and is shown in Figure 4–15. Bands 1-3, 58, 74-79, 101-112, 144-168, and 218-220 were removed from the original image. Because of the memory requirement of the NLDR algorithms, only a portion of the Indian Pines data set was used, this portion is shown in Figure 4–16 this portion contains four different classes. For this image, 15% of the data in each class was used as training data.

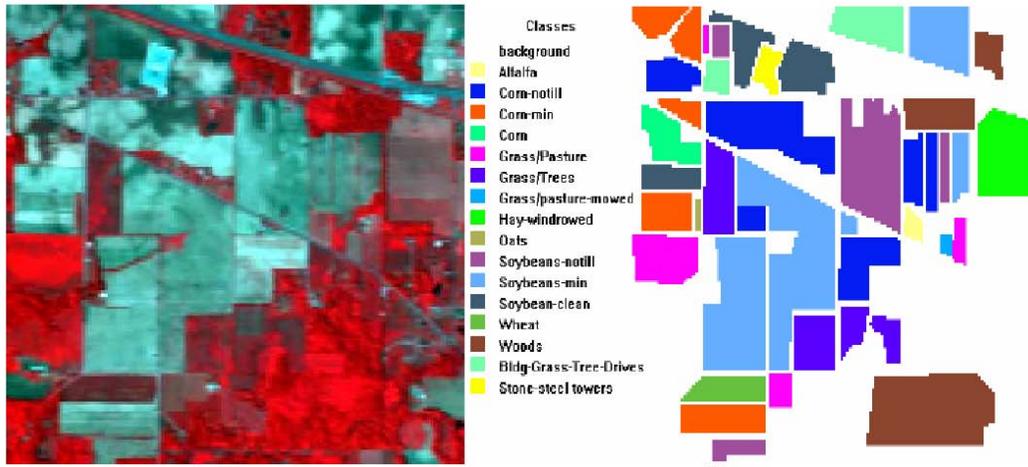


Figure 4–15: Right, Indian Pines (RGB composite bands 47, 24 and 14), Left, Ground Truth.

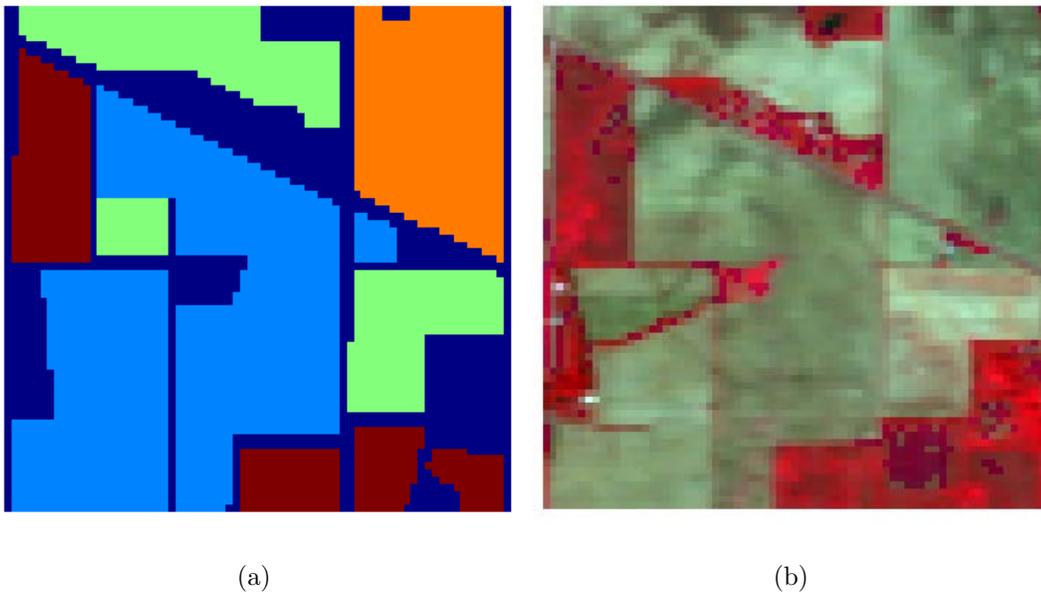


Figure 4–16: Section of the Indian Pines Image (a) RGB Image (Bands 47, 24 and 14), (b) Ground Truth.

### 4.3.2 Results and Discussion

The NLE algorithm was used to construct the graph, the results are compared to the performance using the  $k$ -NN algorithm. The testing in this case is done by setting parameter  $k$  to 12, and changing the number of dimensions of the lower space where the data is projected. The parameter  $k$  was set based on different tests to

evaluate the behavior of the nonlinear projection when using the NLDR algorithms. The overall classification accuracy obtained using the LLE algorithm is shown in Figure 4–18, and using the ISOMAP algorithm in Figure 4–17.

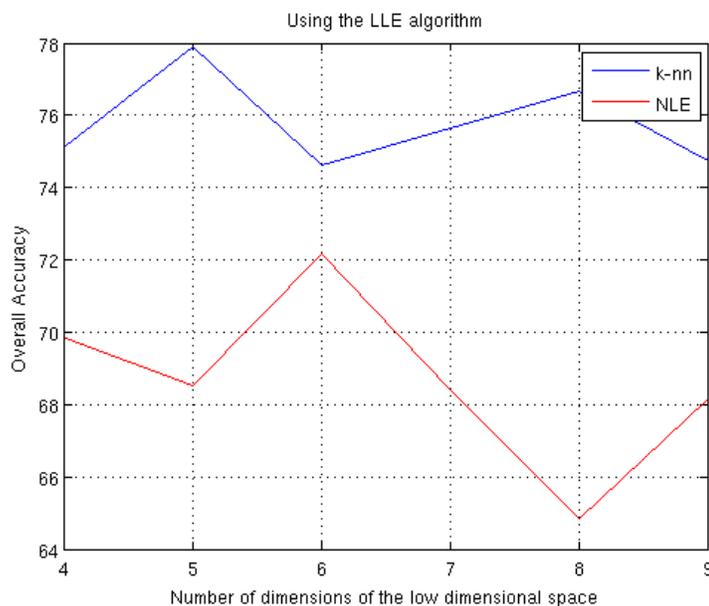


Figure 4–17: Overall Accuracy Using NLE for Graph Construction, LLE for Dimensionality Reduction and Mahalanobis Distance for Classification.

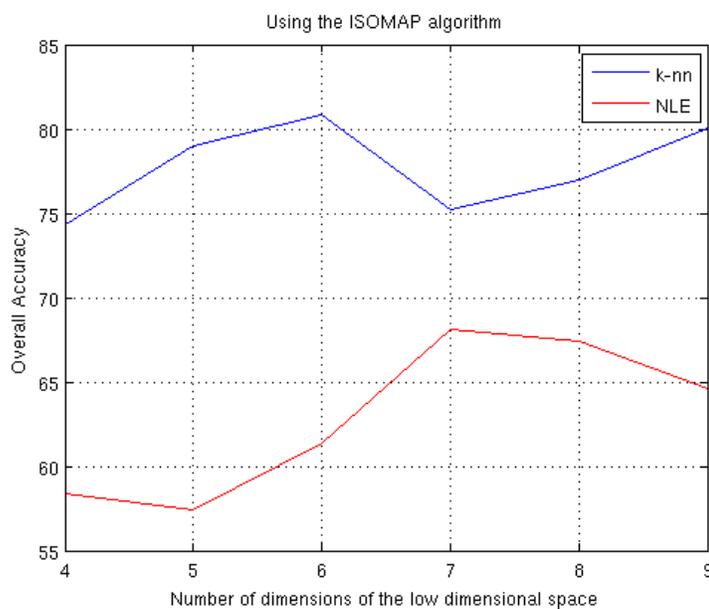


Figure 4–18: Overall Accuracy Using NLE for Graph Construction, ISOMAP for Dimensionality Reduction and Mahalanobis Distance for Classification.

As mentioned before, the drawback of this algorithm is that few neighbors are selected. In the ideal case of a dense smooth sampled manifold, this algorithm could potentially perform outstandingly well, however in this real hyperspectral data set the results were poor. The mean value of number of neighbors found by this algorithm was 8.31, which shows that the connectivity of the graph was too low. The MVU algorithm did not converge in many cases. For that reason, those results are not reported, only the results obtained with the LLE and ISOMAP algorithms are reported.

Now the cam-distance algorithm with different values of  $\alpha$  will be tested. For this purpose, the three NLDR algorithms will be used. Classification will be done using the maximum likelihood classifier. The number of dimension where the data is embedded is set to 10, and the number of neighbors is set to 12. The overall accuracy using the  $k$ -NN algorithm using  $d = 10$  and  $k = 12$  is plotted in the subsequent figures showing the results. This accuracy is the same for all the cases, as the only parameter being changed is the ratio of the two numbers of neighbors needed for the cam-distance algorithm. The result with the  $k$ -NN algorithm is included only for comparison purposes.

Results for the LLE, ISOMAP and MVU algorithms are shown in Figure 4–19, Figure 4–20 and Figure 4–21 respectively. It should be noted that the actual value for  $k_1$ , the first number of neighbors in the cam-distance, was the nearest integer to the value  $\alpha \times 12$ .

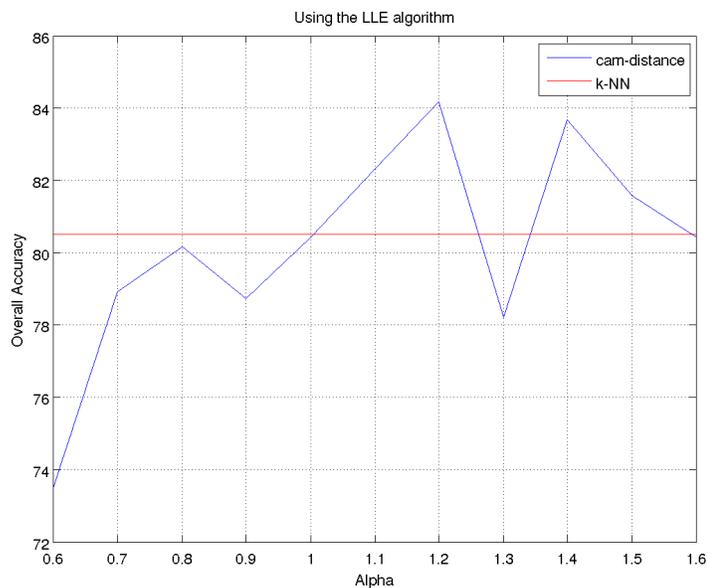


Figure 4–19: Overall Accuracy Using Cam-distance for Graph Construction, LLE for DR and Maximum Likelihood for Classification.

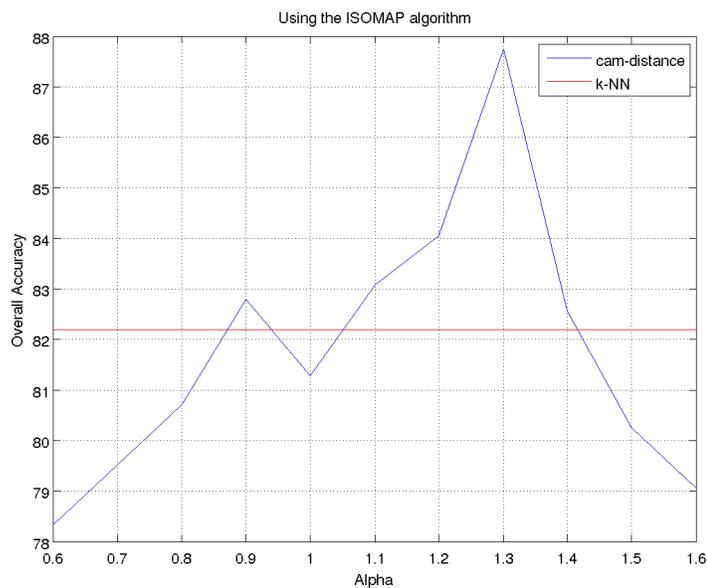


Figure 4–20: Overall Accuracy Using Cam-distance for Graph Construction, ISOMAP for DR and Maximum Likelihood for Classification.

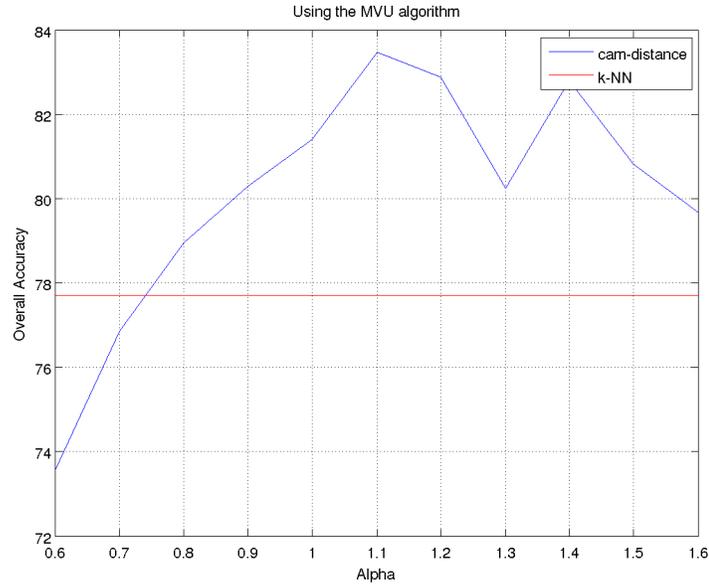


Figure 4–21: Overall Accuracy Using Cam-distance for Graph Construction, MVU for DR and Maximum Likelihood for Classification.

From the figures obtained by changing the parameter  $\alpha$ , it can be inferred that the results using the MVU algorithm are the ones that present the greatest increment in the performance. It can be concluded, that a range between 1.3 and 1.5 works correctly for the three algorithms.

As mentioned before, the classical  $k$ -NN algorithm has the disadvantage that a trade off is present. This trade off is between a small  $k$  which tends to create a disconnected graph and a large  $k$  which generate redundancy in the connectivity of the graph. For this particular dataset, the smallest value of  $k$  which generates a connected graph, is  $k = 30$ .

NLRD Algorithm	LLE	ISOMAP	MVU
Classification Accuracy	55%	63%	58%

The table above shows the classification accuracy obtained when the data is projected into the lower dimensional space using the value of  $k = 30$  and  $d = 8$

with the Mahalanobis distance classifier. This table shows how the performance is significantly degraded due to the presence of redundancy in the connectivity of the graph. From this test, it should be clear also that for any value of  $k$  below 30 the algorithm  $k$ -NN always produces a disconnected graph. This means, the mapping from some data points to the lower space is unknown and for that reason the classification accuracy is reduced. On the other hand, the graph construction methods  $k$ -MST,  $k$ -VC and  $k$ -EC always generates connected graphs. Also, it should be noticed that the behavior in the classification accuracy, when using the  $k$ -NN for graph construction, is expected to produce a bell shape as the number of neighbors goes from 1 to  $N$ , disjoint points and a cluster where every point is connected to each other respectively.

As before Coherent distance will be used. For this part of the test, the parameter  $d$  is fixed to 10, the number of nearest neighbors is varied from 12 to 19 and the maximum likelihood classifier is used. The cam-distance algorithms is tested using a value of  $\alpha = 1.4$ . The overall classification accuracy tends to improve for most of the combination of graph-construction algorithm and NLDR. In summary, only the two best and the worst cases are presented in full detail. It should also be noted that, as shown in [29], the classical  $k$ -NN algorithm was the one which presented highest improvement in performance for most cases.

On the other hand, for the  $k$ -EC algorithm, performance worsens when the coherent distance is used. Figure 4-22 shows the result using the  $k$ -NN algorithm and the ISOMAP algorithm. Figure 4-23 shows the result when using the  $k$ -VC algorithm and the LLE algorithm. These combinations of algorithms were the ones which presented the best improvement in the performance with and without

using the coherent distance. Finally Figure 4–24 shows results when using the cam-distance algorithm together with the MVU algorithm. In this case, the accuracy is actually reduced by 7% in the worst case (12 neighbors).

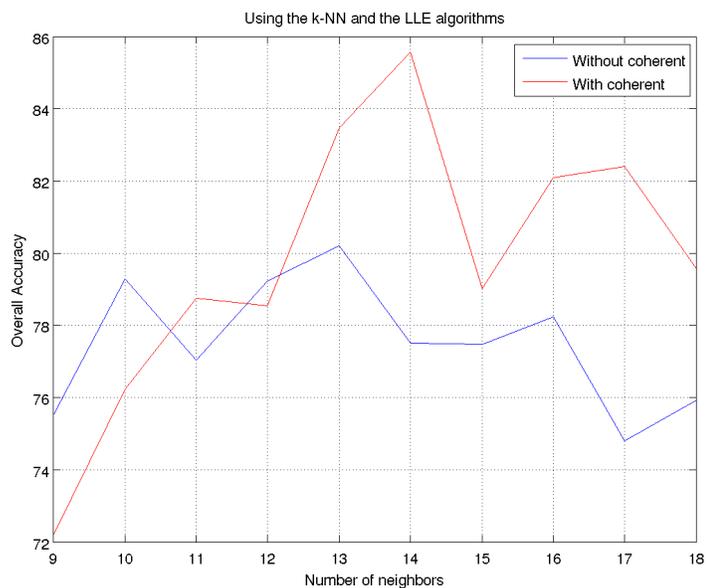


Figure 4–22: Overall Accuracy Using the  $k$ -NN Algorithm for Graph Construction, and DR Using the LLE Algorithm.

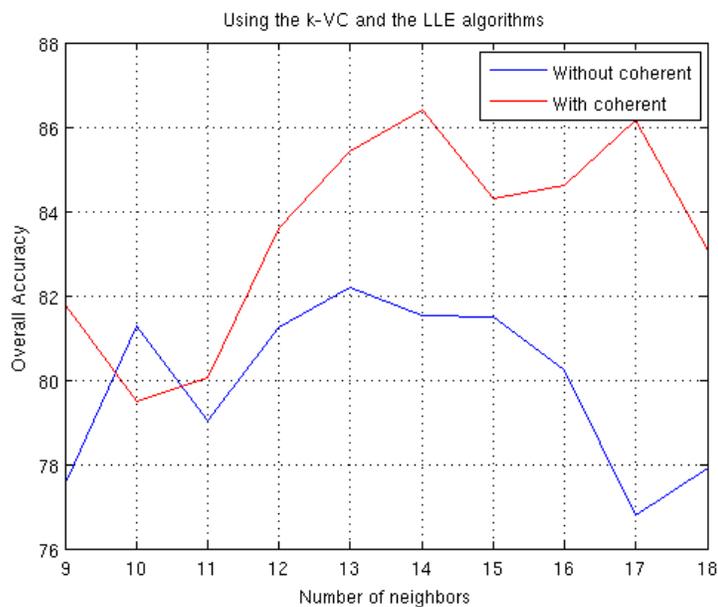


Figure 4–23: Overall Accuracy Using the  $k$ -VC Algorithm for Graph Construction, and DR Using the LLE Algorithm.

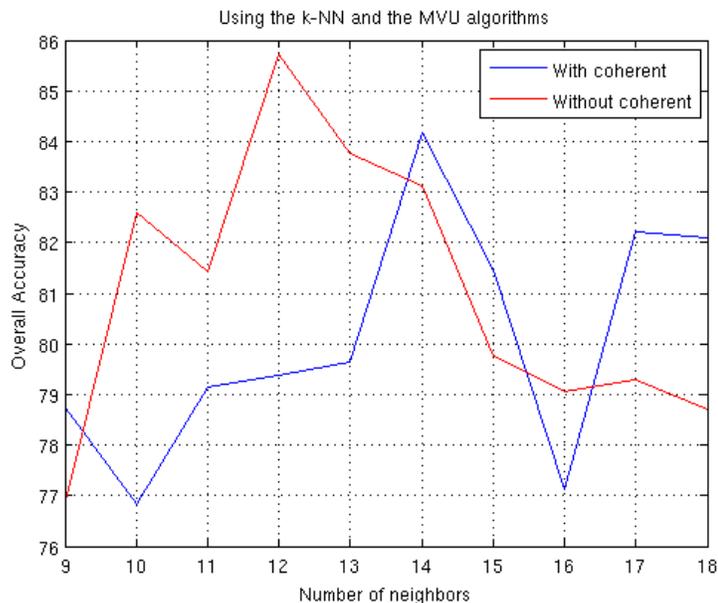


Figure 4–24: Overall Accuracy Using the  $k$ -NN Algorithm for Graph Construction, and DR Using the MVU Algorithm.

Now a comparison in the performance of the different graph construction algorithms is done. The NLE algorithm will not be used as this algorithm shows to perform poorly on this data set as was evaluated before. The parameter for the cam distance algorithm will be  $\alpha = 1.4$ , and the parameter  $d$  will be equal to 10. The coherent distance is used as this distance shows to improve the performance for most of the algorithms combinations. Figure 4–25 shows the results when using all the graph construction algorithms.

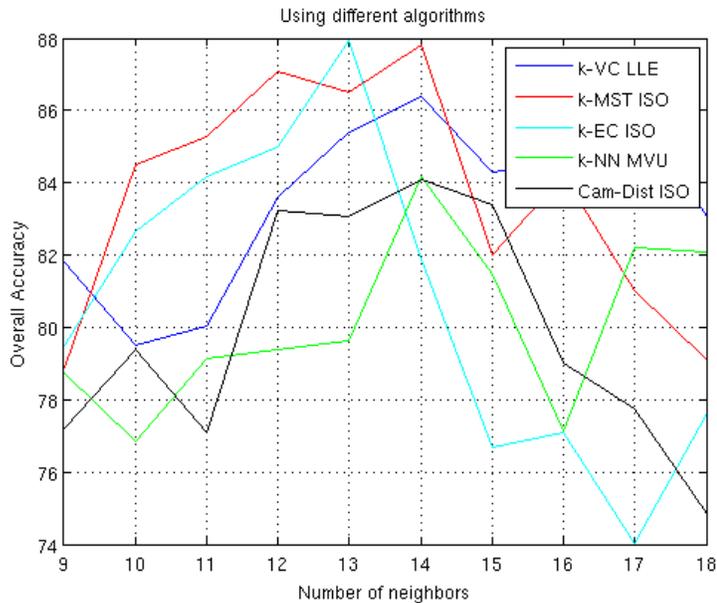


Figure 4–25: Overall Accuracy Using all the Described Graph Construction Algorithms.

For the final test the best performing graph construction algorithm when combined with the ISOMAP algorithm was selected. The geodesic improvement estimation algorithm explained in Subsection 3.1.6 was applied to this case. The result using the  $k$ -MST and the ISOMAP algorithm is shown in Figure 4–26. On 7% of the cases the optimization algorithm did not converge and then the optimization algorithm was not used, instead the original geodesic estimation was used. Figure 4–26 shows the resulting improvement in the overall accuracy.

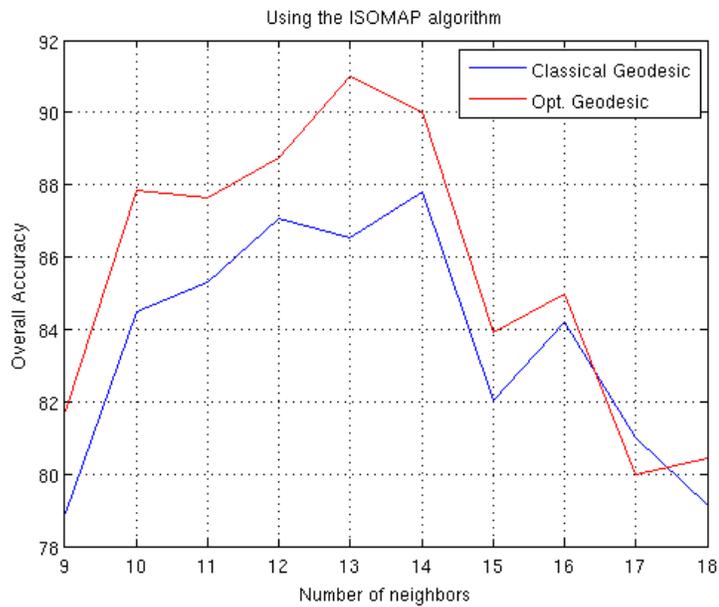


Figure 4–26: Overall Accuracy Improvement Using an Algorithm to Improve the Geodesic Distance Estimation.

# Chapter 5

---

## Conclusions and Future Work

### 5.1 Conclusions

In this research work, six graph construction algorithms were evaluated together with three nonlinear dimensionality reduction algorithms. Also, an algorithm to include spatial information in the DR process was included and an algorithm to improve the geodesic distance estimation for NLDR was explored. Few combinations of the mentioned algorithms were selected to get the best performance and a fair comparison. A synthetic data test was performed where the classical  $k$ -NN approach has the disadvantage of generating short-circuits for small values of  $k$ , which in high-dimensional data, is reflected as a misrepresentation of the global structure of the data. The Laplacian Eigenmap algorithm was used but the performance was comparatively lower to the NLDR algorithm reported in this report.

The performance of the algorithms was evaluated by using Mahalanobis distance classifier and maximum likelihood classifier. Two hyperspectral datasets were used. The results show that the NLE algorithm did a poor job compared to the other algorithms. The  $k$ -MST,  $k$ -EC and  $k$ -VC neighborhood graph algorithms show improvement in extracting the intrinsic structure of the hyperspectral data, this is evidenced indirectly, by the better results obtained on the supervised classification performed on the low dimensional space when using the different NLDR algorithms.

The same result was seen when including spatial information in the construction of the graph, which helps on capturing the intrinsic structure of the data, as evidenced by the improvement in the overall accuracy.

Whether or not the high dimensional data really lies on a manifold is an open problem. Theoretical work have been done [19] to try to answer this question, however no definitive answer has been given. In this work an algorithm that improves the geodesic distance estimation, which is build under the assumption that the data lies in a manifold, was used and the overall accuracy was improved. This result suggest that the data may lie on a manifold.

Disconnected graph turn out to be a problem when we want all the data to be embedded into the new low dimensional space classification application, contrary to what it is expected on anomaly detection applications. Strictly speaking the classical graph construction algorithms are sensitive to spurious data, since getting a connected graph might require a high value of  $k$  which turns out to misrepresent the structure of the data. In the contrary,  $k$ -MST,  $k$ -EC and  $k$ -VC algorithms guarantee connectivity of the graph which is a great advantage. Because high values of  $k$  tend to misinterpret the structure of the data, and small values of  $k$  can be used for these applications.

## 5.2 Future Work

- Study the performance of the Min- $k$ -MST algorithm under the same experimental conditions. This is a robust algorithm to construct a connected graph. Is an extension over the  $k$ -MST,  $k$ -EC and  $k$ -VC algorithms presented here. A computational performance comparison should also be done to evaluate the advantage of the incremental versions proposed.

- Test these algorithms with more NLDR algorithms like KPCA or Local Tangent Space Alignment (LTSA). KPCA is of particular interest since this is a natural extension of the linear PCA algorithm. However, new parameters come into play when selecting the correct kernel function with its respective parameters. LTSA have been successfully used in hyperspectral data, showing promising results.
- A combination of the cam-distance algorithm with any other algorithm for graph construction can be done. In this case it was combined with the classical  $k$  nearest neighborhood. But for example, the cam-distance can be used together with the  $k$ -MST,  $k$ -EC and  $k$ -VC algorithms studied here. However, this will increment the complexity of the graph construction process and more knowledge of the data would be expected.
- Test the performance of these algorithms when noise is added to the images. Of particular interest will be to evaluate the cam-distance algorithm which is expected to overcome this problem up to a high level. Also, it will be expected that the algorithm to improve the geodesic distance estimation performs good under this conditions, since this algorithm is theoretically build under the idea to overcome this situation.

## References

- [1] Silva V. Tenenbaum, J. and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, Dec 2000.
- [2] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.
- [3] T. Kohonen. *Self-Organizing Maps*. Springer, 2001.
- [4] Christopher K.I. Williams. On a connection between kernel pca and metric multidimensional scaling. In *Advances in Neural Information Processing Systems 13*, pages 675–681. MIT Press, 2001.
- [5] Sebastian H. Seung and Daniel D. Lee. Cognition: The manifold ways of perception. *Science*, 290(5500):2268–2269, December 2000.
- [6] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.
- [7] L.J.P. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review, 2008.
- [8] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591. MIT Press, 2001.
- [9] David L. Donoho and Carrie Grimes. Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data, 2003.
- [10] Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimension reduction via local tangent space alignment. *SIAM Journal of Scientific Computing*, 26:313–338, 2002.

- [11] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.
- [12] Fei Sha and Lawrence K. Saul. Analysis and extension of spectral methods for nonlinear dimensionality reduction. In *In Proceedings of the Twenty Second International Conference on Machine Learning (ICML-05)*, pages 785–792, 2005.
- [13] Kilian Weinberger and Lawrence Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70:77–90, 2006. 10.1007/s11263-005-4939-z.
- [14] Silva, V. and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction, 2003.
- [15] Lawrence K. Saul, Kilian Q. Weinberger, Fei Sha, Jihun Ham, and Daniel D. Lee. *Spectral Methods for Dimensionality Reduction*. MIT Press, 2006.
- [16] Shuicheng Yan, Dong Xu, Benyu Zhang, and Hong-Jiang Zhang. Graph embedding: a general framework for dimensionality reduction. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 830–837 vol. 2, June 2005.
- [17] Shiming Xiang, Feiping Nie, Changshui Zhang, and Chunxia Zhang. Nonlinear dimensionality reduction with local spline embedding. *Knowledge and Data Engineering, IEEE Transactions on*, 21(9):1285–1298, Sept. 2009.
- [18] Li Yang. Alignment of overlapping locally scaled patches for multidimensional scaling and dimensionality reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):438–450, March 2008.
- [19] David L. Donoho and Carrie Grimes. Image manifolds which are isometric to euclidean space. *Journal of Mathematical Imaging and Vision*, 23:5–24, July

- 2005.
- [20] A. Plaza, P. Martinez, J. Plaza, and R. Perez. Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(3):466 – 479, 2005.
  - [21] J.C. Harsanyi and C.-I. Chang. Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach. *Geoscience and Remote Sensing, IEEE Transactions on*, 32(4):779 –785, July 1994.
  - [22] L.M. Bruce, C.H. Koger, and Jiang Li. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *Geoscience and Remote Sensing, IEEE Transactions on*, 40(10):2331 – 2338, October 2002.
  - [23] Hong Chang and Dit yan Yeung. Robust locally linear embedding, 2005.
  - [24] Guangyi Chen and Shen-En Qian. Dimensionality reduction of hyperspectral imagery using improved locally linear embedding. *Journal of Applied Remote Sensing*, 1:013509, 2007.
  - [25] Shen-En Qian and Guangyi Chen. A new nonlinear dimensionality reduction method with application to hyperspectral image analysis. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, pages 270 –273, 2007.
  - [26] C.M. Bachmann, T.L. Ainsworth, and R.A. Fusina. Exploiting manifold geometry in hyperspectral imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(3):441 – 454, 2005.
  - [27] Chenping Hou, Yuanyuan Jiao, Yi Wu, and Dongyun Yi. Relaxed maximum-variance unfolding. *Optical Engineering*, 47(7):077202, 2008.
  - [28] L. Samaniego, A. Bardossy, and K. Schulz. Supervised classification of remotely sensed imagery using a modified k -nn technique. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(7):2112 –2125, 2008.

- [29] Luis O. Jiménez, J.L. Rivera-Medina, E. Rodríguez-Díaz, E. Arzuaga-Cruz, and M. Ramírez-Vélez. Integration of spatial and spectral information by means of unsupervised extraction and classification for homogenous objects applied to multispectral and hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(4):844 – 851, April 2005.
- [30] Shanmugam K. Dinstein Its'Hak Haralick, Robert M. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610 –621, Nov. 1973.
- [31] G. Miaohong Shi, Healey. Hyperspectral texture recognition using a multiscale opponent representation. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(5):1090 – 1095, May 2003.
- [32] Elliott Howard Derin, Haluk. Modeling and segmentation of noisy and textured images using gibbs random fields. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(1):39 –55, Jan. 1987.
- [33] Healey G. Sarkar, S. Hyperspectral texture classification using generalized markov fields. volume 1, pages I-429 – I-434 Vol.1, June-2 July 2004.
- [34] Sapiro G. Bosch E. Mohan, A. Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images. *Geoscience and Remote Sensing Letters, IEEE*, 4(2):206 –210, April 2007.
- [35] S.S. Ge, F. Guan, A.R. Loh, and C.H. Fua. Feature representation based on intrinsic structure discovery in high dimensional space. *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 3399 –3404, may. 2006.
- [36] Yaozhang Pan, Shuzhi Sam Ge, and Abdullah Al Mamun. Weighted locally linear embedding for dimension reduction. *Pattern Recognition*, 42(5):798 – 811, 2009.

- [37] Chang Yin Zhou and Yan Qiu Chen. Improving nearest neighbor classification with cam weighted distance. *Pattern Recognition*, 39(4):635 – 645, 2006. Graph-based Representations.
- [38] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, January 1979.
- [39] Li Yang. K-edge connected neighborhood graph for geodesic distance estimation and nonlinear data projection. volume 1, pages 196 – 199 Vol.1, Aug. 2004.
- [40] Li Yang. Building k edge-disjoint spanning trees of minimum total length for isometric data embedding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1680 –1683, Oct. 2005.
- [41] Li Yang. Building k-edge-connected neighborhood graph for distance-based data projection. volume 26, pages 2015 – 2021, 2005.
- [42] Li Yang. Building k-connected neighborhood graphs for isometric data embedding. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(5):827 –831, May 2006.
- [43] Li Yang Dongfang Zhao. Incremental isometric embedding of high-dimensional data using connected neighborhood graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):86 –98, Jan. 2009.
- [44] John Aldo Lee, Amaury Lendasse, and Michel Verleysen. Curvilinear distance analysis versus isomap. In *Proceedings of ESANNâ2002, 10th European Symposium on Artificial Neural Networks*, pages 185–192, 2000.
- [45] Li Yang. Sammon’s nonlinear mapping using geodesic distances. *Pattern Recognition, International Conference on*, 2:303–306, 2004.
- [46] Deyu Meng, Yee Leung, Zongben Xu, Tung Fung, and Qingfu Zhang. Improving geodesic distance estimation based on locally linear assumption. *Pattern Recognition Letters*, 29(7):862 – 870, 2008.

- [47] David Landgrebe. Documentation for multispec. <https://engineering.purdue.edu/~biehl/MultiSpec/documentation.html>, November 2009.

## Biographical Sketch

Nicolas Rey was born in Jan. 28 of 1986 in Bucaramanga, Colombia. On March of 2008 he received his B.S.E.E degree, *Cum Laude* from the Universidad Industrial de Santander, Colombia. During his undergraduate studies he took the ECAES test organized annually nationwide to evaluate higher education in Colombia and ranked tenth. On August of 2008 he started his M.Sc. studies at the University of Puerto Rico at Mayagüez.